ЎЗБЕКИСТОН АЛОҚА ВА АХБОРОТЛАШТИРИШ АГЕНТЛИГИ
ФАН-ТЕХНИКА ВА МАРКЕТИНГ ТАДҚИҚОТЛАРИ МАРКАЗИ

АКБАРОВ ДАВЛАТАЛИ ЕГИТАЛИЕВИЧ

АХБОРОТ ХАВФСИЗЛИГИНИ ТАЪМИНЛАШНИНГ КРИПТОГРАФИК УСУЛЛАРИ ВА УЛАРНИНГ ҚЎЛЛАНИЛИШИ

ТОШКЕНТ
«ЎЗБЕКИСТОН МАРКАСИ» НАШРИЁТИ
2009

УДК 681.3

Маъсул муҳаррир:
т.ф.д., проф. С.С. Қосимов

Тақризчилар:
т.ф.д., проф. П.Ф. Хасанов, т.ф.д., проф. М.М. Каримов

Акбаров Давлатали Егиталиевич
Ахборот хавфсизлигини таъминлашнинг криптографик усуллари ва уларнинг қўлланилиши – Тошкент, «Ўзбекистон маркаси» нашриёти, 2009 – 432 бет.

Ушбу китобда криптографияни фан сифатида шаклланиш даврлари, унинг илмий асослари, ахборот хавфсизлигини таъминлашнинг асосий масалалари, бу масалалар ечимларининг криптографик воситалари ва услублари, уларни ахборот-коммуникация тизимларида қўлланилиши илмий асосда ёритилган. Ахборот хавфсизлигини таъминлашнинг криптографик воситалари бўлган: шифрлаш, хэш-функция ва электрон рақамли имзонинг мавжуд стандарт алгоритмлари таҳлил қилиниб, аппарат-техник ва аппарат-дастурий таъминотлари қулай бўлган янги, самарали ва криптобардошли алгоритмлар таклиф этилган. Криптоалгоритмлар учун бардошли калитлар ишлаб чиқиш ва уларни бошқариш масалаларининг мавжуд ечимлари таҳлил қилиниб, янги криптобардошли калитлар генерация қилиш алгоритмлари ва калитларни бошқариш протоколлари таклиф этилган.

Китоб криптология фанини ўрганувчилар ва бу соҳада илмий изланишлар олиб борувчилар учун ўқув қўлланма сифатида тавсия этилиши мумкин.

ISBN 978-9943-356-05-4

СЎЗ БОШИ

Ахборот-коммуникация тизимларида маълумотлар алмашинувини самарали амалга оширишни ташкил этиш бугунги ривожланган жамиятда катта аҳамият касб этади. Ахборот технологияларининг жадал ривожланиб бориши, жамият фаолиятининг кенг соҳасида турли ахборот хизматларининг вужудга келишига олиб келди. Айниқса банк ва бошқа тўлов тизимларида, давлат ва жамият манфаатлари билан боғлиқ муҳим маълумотларни алмашиш ҳамда таҳлил қилишда, тез ва ишончли маълумот алмашинуви талаб этиладиган тизимларда ахборот муҳофазаси масалалари долзарб ҳисобланади. Ҳақиқатан ҳам, ҳар қандай маълумот у ёки бу маънода ахборот-коммуникация тизими фойдаланувчиларининг манфаати билан боғлиқ. Ахборот муҳофазасини таъминлаш: ҳуқуқий-меъёрий ҳужжатлар, техник воситалар ва криптографик алгоритмлар ҳамда протоколлар негизида яратилган дастурий, аппарат-дастурий ва аппарат-техник воситаларнинг биргаликда қўллаш билан самарали амалга оширилади.

Ўзбекистон Республикаси Президенти И.А. Каримов ва Ўзбекистон Республикаси Вазирлар Маҳкамасининг қатор фармон ва қарорларида Республикамизда ахборот технологияларини ривожлантиришнинг аниқ йўналишлари белгилаб берилиб, бу соҳа мутахассисларига фаоллик кўрсатиш учун шарт-шароитлар яратилиб берилмоқда. Бу соҳада Ўзбекистон алоқа ва ахборотлаштириш агентлигининг Фан-техника ва маркетинг тадқиқотлари марказининг мутахассис ходимлари фаоллик кўрсатиб келмоқда.

Ушбу китоб муаллифнинг Ўзбекистон алоқа ва ахборотлаштириш агентлиги Тошкент ахборот технологиялари университети магистрлари билан олиб борилган ўқув ва илмий тадқиқот машғулотлари материаллари ҳамда Фан-техника ва маркетинг тадқиқотлари марказининг мутахассис ходимлари билан қилинган илмий ҳамкорликлари натижалари асосида ёзилган. Китобда криптологиянинг фан сифатида шаклланиш даврлари, унинг илмий асослари, ахборот хавфсизлигини таъминлашнинг асосий масалалари, бу масалалар ечимларининг криптографик услублар ва алгоритмлари, уларни ахборот-

коммуникация тизимларида қўлланишлари илмий асосда ёритилган. Муаллиф ахборот хавфсизлигини таъминлашнинг криптографик воситалари бўлган: шифрлаш, хэш-функция ва электрон рақамли имзо мавжуд стандарт алгоритмларини таҳлил қилиб ўз илмий изланишлари маҳсули бўлган бардошлилиги илмий асосланган бир нечта алгоритмларни таклиф этган. Криптоалгоритмлар учун бардошли калитлар ишлаб чиқиш ва уларни бошқариш масалаларининг мавжуд ечимлари таҳлил қилиниб, янги криптобардошли калитлар генерация қилиш алгоритмлари ва калитларни бошқариш протоколлари таклиф этилган. Келтирилган маълумотларни таҳлил ва баён қилиш усуллари ўқувчида криптология соҳасида илмий изланишлар олиб боришга ишонч уйғотади.

Китоб криптология фанини ўрганувчилар ва бу соҳада илмий изланишлар олиб борувчилар учун қўлланма сифатида тавсия этилади.

Фан-техника ва маркетинг
тадқиқотлари маркази директори
т.ф.н. М. Махмудов

КИРИШ

Бугунги жамият тараққиёти инсоният тафаккурининг маҳсули бўлган ривожланган илм-фан ютуқларига асосланган техника ва технологиялар билан бир қаторда, кенг маънода, ахборотларнинг муҳим аҳамиятга эгалиги орқали ҳам белгиланади. Инсон тафаккури ривожининг манбаи эса маълумотлар (ахборотлар) мажмуидан иборатдир. Шак-шубҳасиз ўз вақтида олинган тўла ва ишончли маълумот, шу маълумот билан боғлиқ бўлган ҳолатдан келиб чиқадиган амалий фаолиятларнинг мақсадли кечишларини мувофиқлаштиришда муҳим аҳамият касб этади. Фаолият мақсадларининг турлича бўлиши табиий равишда ахборотлардан турли мақсадларда фойдаланиш асосларига сабаб бўлади. Шунинг учун бугунги, ахборотларни сақлаш ва узатиш тизимлари бир томондан такомиллашиб мураккаблашган ва иккинчи томондан ахборотлардан фойдаланувчилар учун кенг қулайликлар вужудга келган даврда, ахборотларни мақсадли бошқаришнинг қатор муҳим масалалари келиб чиқади. Бундай масалалар қаторига катта ҳажмдаги ахборотларнинг тез ва сифатли узатиш ҳамда қабул қилиш, ахборотларни ишончлилигини таъминлаш, ахборотлар тизимида ахборотларни бегона шахслардан (кенг маънода) муҳофаза қилиш каби кўплаб бошқа масалалар киради. Ахборот ва ахборот тизимидан фойдаланиш инсоният фаолиятининг барча соҳаларига кириб бориб, муҳим аҳамият касб этиб, ривожланиб бораётган бугунги жамиятда ахборотларни мақсадли бошқариш фаоллашмоқда. Компьютерлар ва компьютер тизимлари ахборот тизимининг муҳим бўғимидир. ИНТЕРНЕТ тармоқлари жамият фаолиятининг барча соҳаларини қамраб олиб, ахборотни тез ва сифатли алмашинувини таъминлаш технологияларининг ривожланишига ижобий манба бўлиб келмоқда. Юқоридаги келтирилган асосли мулоҳазалардан келиб чиқиб, ахборотларни асли ҳолидан ўзгартирилган ҳолда, яъни шифрланган ҳолда, сақлаш ва узатиш масалаларининг муҳим эканлигига шубҳа йўқдир. Ахборотларнинг муҳофазасини таъминлаш масалалари инсоният жамиятида қадимдан муҳим бўлиб келган. Айтиш мумкин-

ки, ахборотни муҳофаза қилиш услублари жамиятда дастлабки пайдо бўлган муомала тили ва ёзуви билан узвий боғлиқ ҳамда тенгдошдир. Ҳақиқатдан ҳам, қадимда ёзув муомала воситасидан фақат айрим юқори табақадаги жамият аъзоларигина фойдаланганлар. Қадимий Миср ва Ҳиндистоннинг илоҳий китоблари бунга мисол бўла олади. Эрамиздан аввалги бешинчи асрда яшаб ўтган грек олими Геродотнинг хабар беришича, қадимий Мисрда шифрланган ахборотлар ролини, жрецлар, яъни юқори табақадаги етук фикрли кишилар томонидан яратилган муомала тили бажарган. Бунда учта алифбо асосланилган: ёзув, илоҳий ва махфий. Ёзув алифбоси оддий ўзаро муомалада қўлланилган, илоҳий алифбо диний муомала воситаси сифатида қўлланилган, махфий алифбо эса маълумотларнинг асл маъносини бегоналардан муҳофаза қилишда ва астриологлар томонидан қўлланилган.

Турли ёзув алифболарининг вужудга келиши ва ривожланиши натижасида *криптография* мустақил йўналишда ривожлана борди.

Криптография – ахборотни аслидан ўзгартирилган ҳолатга акслантириш услубларини топиш ва такомиллаштириш билан шуғулланади. Дастлабки тизимлашган криптографик услублар эрамиз бошида, Юлий Цезарнинг иш юритиш ёзишмаларида учрайди. У, бирор маълумотни махфий ҳолда бирор кишига етказмоқчи бўлса, алифбонинг биринчи ҳарфини тўртинчи ҳарфи билан, иккинчи ҳарфини бешинчиси билан ва ҳоказо тартибда алмаштириб матнни асл ҳолатидан шифрланган матн ҳолатига ўтказган.

Криптографик тизимлар йўналишидаги изланишлар айниқса биринчи ва иккинчи жаҳон уруши йиллари даврида муҳим аҳамият касб этди ва жадал ривожланди. Урушдан кейинги йилларда, ҳисоблаш техникаларининг яратилиши, уларнинг такомиллашиб, инсоният фаолиятининг барча соҳаларига чуқур ва кенг маънода кириб бориши, криптографик услубларни табиий равишда ривожланиб ва такомиллашиб боришини таъқозо этмоқда.

Криптографик услубларнинг ахборот тизими муҳофазаси масалаларида қўлланилиши, айниқса, ҳозирги кунда фаоллашиб бормоқда. Ҳақиқатан ҳам, бир томондан компьютер тизимларида ИНТЕРНЕТ тармоқларидан фойдаланган ҳолда катта ҳажмдаги давлат ва ҳарбий аҳамиятга эга бўлган, ҳамда, иқтисодий, шахсий ва бошқа турдаги ахборотни тез ва сифатли узатиш, қабул қилиш кенгайиб бормоқда. Иккинчи томондан эса бундай ахборотларнинг муҳофаза қилинишини таъминлаш масалалари муҳимлашиб бормоқда.

Ахборотни муҳофаза қилиш масалалари билан *криптология* (kryptos- махфий, logos-илм) фани шуғулланади. Криптология мақсадлари ўзаро қарама-қарши иккита йўналишига эга бўлган – *криптография* ва *криптотаҳлил.*

Криптографиянинг очиқ маълумотларни шифрлаш масалаларининг математик услублари билан шуғулланиши тўғрисида юқорида айтиб ўтилди.

Криптотаҳлил эса шифрлаш услуби (калити ёки алгоритми)ни билмаган ҳолда шифрланган маълумотни асл ҳолатини (мос келувчи очиқ маълумотни) топиш масалаларини ечиш билан шуғулланади.

Ҳозирги замон криптографияси қуйидаги тўртта бўлимни ўз ичига олади:

Симметрик криптотизимлар.

Очиқ услубга ёки яна бошқача айтганда очиқ калитлар алгоритмига асосланган криптотизимлар.

Электрон рақамли имзо криптографик тизимлари.

Криптотизимлар учун криптобардошли калитларни ишлаб чиқиш ва улардан фойдаланишни бошқариш.

Криптографик услублардан фойдаланишнинг асосий йўналишлари: ёпиқ маълумотларни очиқ алоқа канали бўйича муҳофазаланган ҳолда узатиш, уларнинг ҳақиқийлигини таъминлаш, ахборотларни (электрон ҳужжатларни, электрон маълумотлар жамғармасини) компьютерлар тизими хотирасида шифрланган ҳолда сақлаш ва шу каби масалаларнинг ечимларини ўз ичига олади.

Таъкидлаш жоизки, криптография узоқ вақт давомида давлат органлари алоқа тармоқларида алмашинадиган маълумотлар муҳофазасининг таъминланишида қўлланиб келинди. Компьютер тармоқлари ва электрон ҳужжат алмашинуви технологияларининг ривожланиши молия, банк ишлари, савдо-сотиқ каби соҳаларда қўлланилиши ахборот муҳофазасининг криптографик усулларини умумжамият фаолиятининг турли соҳаларига кенг кириб боришига сабаб бўлди. Ҳақиқатан ҳам, алоқа тармоқларида ахборотни муҳофаза қилиниши, криптографик усулда таъминлаш умумжамият тараққиётининг ривожланиш босқичлари билан боғлиқ бўлган узоқ тарихий манбаларига эга бўлиб, умуминсоният жамиятига хизмат қилмаслиги (яъни криптографик усулларни кенг омма томонидан фойдаланилишининг чекланиши) таажжубланарли ҳолат бўлар эди.

I БОБ
КРИПТОГРАФИЯ ФАНИНИНГ ШАКЛЛАНИШИ ВА УНИНГ АСОСИЙ МАСАЛАЛАРИ

§1.1. Асосий тушунчалар

Ахборот муҳофазасининг криптографик услублари очиқ маълумотларни ўзгартириб, фақат калит маълум бўлгандагина уни асл ҳолатига қайтариш имкониятини беради.

Шифрлаш ва дешифрлаш масалаларига тегишли бўлган, маълум бир алифбода тузилган маълумотлар *матнларни* ташкил этади.

Алифбо – ахборотни кодлаш учун фойдаланиладиган чекли сондаги белгилар тўплами. Мисол сифатида:

- ўттиз олтита белгидан (ҳарфдан) иборат ўзбек тили алифбоси;
- ўттиз иккита белгидан (ҳарфдан) иборат рус тили алифбоси;
- йигирма саккизта белгидан (ҳарфдан) иборат лотин алифбоси;
- икки юз эллик олтита белгидан иборат ASCII ва КОИ–8 стандарт компьютер кодларининг алифбоси;
- бинар алифбо, яъни 0 ва 1 белгилардан иборат алифбо;
- саккизлик ва ўн олтилик саноқ тизимлари белгиларидан иборат алифболарни келтириш мумкин.

Матн – алифбонинг элементларидан (белгиларидан) ташкил топган тартибланган тузилма.

Шифрлаш – очиқ матн деб аталувчи *дастлабки маълумотни шифрланган маълумот (криптограмма)* ҳолатига ўтказиш жараёни.

Дешифрлаш – шифрлашга тескари бўлган жараён, яъни калит ёрдамида шифрланган маълумотни дастлабки ҳолатга ўтказиш.

Калит – дастлабки маълумотни бевосита шифрлаш ва дешифрлаш учун зарур манба.

Криптографик тизим – очиқ маълумотни шифрлаш ва дешифрлаш жараёнини ташкил этувчи амаллар мажмуи бўлиб, алифбо белгиларини алмаштириш кетма-кетлигидан иборат.

Криптотизимлар икки қисмга бўлинади: *симметрик* ва *асимметрик – очиқ калитли.*

Симметрик криптотизимларда шифрлаш учун ҳам ва дешифрлаш учун ҳам бир хил калитдан фойдаланилади.

Очиқ калитли криптотизимларда иккита калитдан фойдаланилади – ўзаро математик жиҳатдан боғлиқ бўлган *очиқ* ва *ёпиқ* калитлардан. Бунда маълумотлар маълумот юборилаётган шахснинг ҳаммага

маълум бўлган очиқ калити билан шифрланади ва фақат маълумот юборилаётган шахснинг ўзигагина маълум бўлган ёпиқ калит билан дешифрланади.

Калитларни тақсимлаш ва *бошқариш* – криптобардошли калитларни ишлаб чиқиш (ёки яратиш), уларни сақлаш, ҳамда калитларни фойдаланувчилар орасида муҳофазаланган ҳолда тақсимлаш жараёнларини ўз ичига олади.

Электрон рақамли имзо – электрон матнга илова қилинадиган криптографик алмаштиришдан иборат бўлиб, шу матн жўнатилган шахсга қабул қилинган электрон матннинг ва матнни рақамли имзоловчининг ҳақиқий ёки ноҳақиқий эмаслигини аниқлаш имконини беради.

Криптобардошлилик – шифрлаш калити номаълум бўлган ҳолда шифрланган маълумотни дешифрлашнинг қийинлик даражасини белгилайди. Криптобардошлиликни белгиловчи бир нечта кўрсаткичлар мавжуд, булардан:

- дешифрлаш учун қидирилаётган калитларнинг мумкин бўлган барча имкониятлари сони;
- дешифрлаш учун зарур бўлган ўртача вақт.

Ахборотни муҳофаза қилиш мақсадида шифрлаш сифати калитнинг махфий сақланиши ва шифрлашнинг криптобардошлилик даражасига боғлиқ.

§ 1.2. Криптологиянинг фан сифатида шаклланиши

Қадимги шифрлаш услублари ҳар-хил жадвалларга асосланган бўлиб, бу жадваллар маълумотлар матнидаги алифбо белгиларининг маълум тартибдаги ўрин алмаштиришларини ифодаловчи оддий амаллардан иборат бўлган. Бунда калит вазифасини жадвалнинг ўлчами, алифбо белгиларининг алмаштирилишини таминловчи бирор аниқ жумла ёки жадвалнинг ўрин алмаштиришларини тартибловчи алоҳидалик хусусияти ва шу кабилар ўтаган.

Мисол учун,

АНГЛАШИЛМОВЧИЛИК ТУШИНАРСИЗЛИККА ОЛИБ КЕЛДИ

жумла устунларининг сони 5 та ва сатрларининг сони 8 та бўлган жадвалнинг устунлари бўйича ёзиб чиқилса, сўнгра шу жадвалнинг сатрлари бўйича гуруҳланса:

А М Т И Л, Н О У З И, Г В Ш Л Б, Л Ч И И К,
А И Н К Е, Ш Л А К Л, И И Р А Д, Л К С О И

каби шифрланган сўзлар ҳосил бўлади.

Маълумотларни шифрлаб муҳофаза қилишнинг турли мақсадларда қўлланиб ривожланиб бориши, шифрлаш услубларининг фойдаланувчилар томонидан алоқа тармоқларида қўллаш учун қулай бўлишини талаб қилиниши билан бирга, унинг бардошлилигига бўлган талабнинг ҳам кучайишига олиб келди. XIX асрда алоқа коммуникацияларининг ривожланиб бориши, табиий равишда, шифрлаш жараёнларини автоматлаштирилишини талаб эта бошлади. Телеграф алоқа тизимлари вужудга келди ва улар ҳам, ўз навбатида, маълумотларни шифрлашни талаб эта бошлади. Махсус ғилдирак кўринишидаги, сонли шифрлаш қурилмаси 1790 йилда Америка қўшма Штатларининг (АҚШ) давлат котиби, кейинчалик эса АҚШнинг учинчи Президенти Томас Жефферсон томонидан яратилган ва шунга ўхшаш сонли шифрлаш қурилмалари иккинчи жаҳон уруши йилларидан кейин ҳам АҚШ қуролли кучларида қўлланилиб келинган. Бундай қурилмаларнинг ишлаши, етарли даражада узун берилган калит бўйича маълумотлар матнини кўп алифболи алмаштиришга асосланган бўлиб, арифмометрнинг ишлаш асосларига ўхшашдир. Калитнинг (даврий) узунлиги шифрлаш қурилмасининг махсус ғилдиракларини бир марта тўла айланишларининг умумий даври билан аниқланади. Масалан, мос ҳолда 13, 15, 17, ва 19 даврий айланишларга эга бўлган махсус тўртта ғилдиракли қурилма 62985 (даврий) узунликка эга бўлган калитни беради. Яъни, қурилма ғилдираклари бирор аниқ ҳолатда турган бўлса, мана шу ҳолатга қайтадан кетма-кет 1-ғилдиракни 13 марта, 2-ғилдиракни 15 марта, 3-ғилдиракни 17 марта, 4-ғилдиракни 19 марта айлантириш билан эришилади [1].

Ҳозирги замон криптографик машиналари асосини, 1917 йилда Эдвард Хеберн томонидан яратилган, «Enigma – Энигма» («Жумбоқ» маъносини англатувчи) деб аталувчи роторли криптографик машинанинг ишлаш тамойиллари ташкил этади [2]. «Энигма» машиналарининг саноат наъмуналари Siemens фирмаси томонидан ишлаб чиқилиб, дастлаб битта ўққа ўрнатилган тўртта айланувчи ғилдиракдан иборат бўлиб, бирор аниқ ҳолатнинг, оддий алмаштиришлар ёрдамида, миллиондан ортиқ шифрланган ҳолатини олиш имконини берган. Ҳар бир ғилдиракнинг иккала томонида 25 тадан (лотин алифбосининг белгилари (ҳарфлари) сонича) электр боғланиш тугунлари жойлашган бўлиб, ғилдираклар айланганда электр боғланиш импульслари рўй бериб, ҳарфларнинг алмашув жараёни юзага келади. Шифрлаш жараёни бошланиши олдидан ғилдираклар калитни белгиловчи сўзни аниқлаш ҳолатига ўрнатилади. Алифбо ҳарфларини белгиларини бошқа белгилар билан алмаштириб шифрлаш жараёни, шифрлани-

ши керак бўлган белгининг тугмачаларини босиш натижасида амалга оширилган. Бунда шифрланиши керак бўлган белгининг тугмачаларини босиш натижасида, аввал 1-ғилдирак, сўнгра 2-ғилдирак бир қадамга бурилган ва ҳоказо. Натижада, калитнинг узунлиги очиқ матн узунлигига нисбатан узун бўлган. Масалан, чап ва ўнг томондаги ғилдиракларнинг U белгисига мос келувчи электр боғланиш тугунлари ғилдиракларнинг бошқа томонидаги F белгига мос келувчи электр боғланиш тугунлари билан боғланган. Агар ғилдирак бир қадамга бурилса, бу ҳолат U белгидан кейинги V белгини F белгидан кейинги G белгига алмаштириш жараёнини ифодалайди. Тўрт ғилдиракли криптографик машиналарда алиффо белгиларини шифрлаш жараёнида ҳамма белги ҳар бир ғилдиракда ўзгариш жараёнидан ўтиб, тўрт карра шифрланади. Дешифрлаш жараёнини мураккаблаштириш мақсадида ғилдиракларнинг ўрни вақти-вақти билан алмаштирилиб турилган. Кейинчалик эса ғилдиракларнинг сони 5 ва 6 тага кўпайтирилиб, уларнинг ҳаракатининг маълум маънода тартибсиз бўлиши таъминланган. Бу қурилманинг ҳажми катта бўлмаганлиги ҳамда ундан фойдаланиш мураккаб эмаслиги сабабли, оддий алоқа хизматчилари ҳам ишлата олганлар. Шу даврга келиб, маълумотларни ишончли шифрлаш масаласи тўла ҳал қилингандек эди. Лекин, Англия криптографик хизматининг хизматчилари Лондондан 80 км шимолда жойлашган «Блетчли Боғ» қароргоҳида иккинчи жаҳон уруши йиллари давомида немислар шифрмаълумотларини ўқиб боришга муваффақ бўлганлар. Бунга Польша разведка хизмати томонидан 1939 йилда қўлга киритилган «Энигма» криптографик машинасининг чизмалари асос бўлди. Гитлерчиларнинг Польшага хужумидан сўнг машина чизмалари Англиянинг тегишли хизмат ташкилотларига берилди. Тез орада, Англия криптотаҳлил хизмати ходимлари, «Энигма» машинасининг шифрлаш калитини билиш учун, машина махсус ғилдиракларидаги электр боғланиш тугунларининг схемасини билиш кераклигини аниқладилар. Шундан сўнг, «Энигма» машинасининг қурилма наъмунасини қўлга киритиш учун ҳаракатлар бошланди. Биринчи намунани Германиянинг жанубий-шарқий қисмида жойлашган заводдан олинишига эришилди, иккинчиси Норвегия ҳаво ҳудудларида уриб туширилган немис ҳужумчи самолётларидан, учинчиси эса Франция учун бўлган жангларда аср тушган немис ҳарбий алоқачи аскаридан олинган. Кейинги намуналар эса ғоввослар махсус қисмлари томонидан немис сув ости кемаларидан олинган. 1942 йилда Алан Тьюринг томонидан махсус электрон ҳисоблаш машинаси яратилгунга қадар, «Энигма» шифрларини дешифрлаш анча мураккаб бўлди. Алан Тьюрингнинг дешифрлаш учун

махсус яратган ва «Колосс» деб номланган ушбу машинаси инсоният дунёсида биринчи тез ишловчи электрон ҳисоблаш машинаси (ЭҲМ) эди. Шундан сўнг Англия криптотаҳлилчилари, намуналари олинган «Энигма» машиналарининг ғилдиракларидан фойдаланиб, қисқа вақт ичида мумкин бўлган барча калитларни танлаб чиқиб, дешифрлаш масаласини ҳал эта бошладилар. Немислар эса дешифрлашда ЭҲМнинг қўлланилишини ҳисобга олмаган эдилар. Шуни ҳам айтиб ўтиш керакки, 1930 йилда немис криптотаҳлилчиси Георг Шредернинг «Энигма» криптографик машинасининг шифрлаш услубига ишончсизлик билдириб келтирган далиллари кўпчилик масъул мутахассислар назаридан четда қолган. 1926 йилда Америка телефон ва телеграф компанияларидан бирининг маҳандиси Г.С. Вернам ўзининг иккилик саноқ тизимси асосида яратган шифрлаш алгоритмини эълон қилди [3]. Вернамнинг шифрлаш алгоритми Цезарнинг шифрлаш алгоритмига ўхшаш бўлиб, у қуйидаги $y=x\oplus z$, (1.1)

тенглама билан ифодаланади ва бунда x, y, z ўзгарувчилар иккилик саноқ тизимси алифболарида қийматлар қабул қилади, $\oplus$ белги эса 2 модуль бўйича қўшиш амалини билдиради, яъни: $0\oplus0=0$, $0\oplus1=1$, $1\oplus0=1$, $1\oplus1=0$. Бу алгоритмнинг моҳияти дешифлаш калитининг фақат бир марта ишлатилишига асосланган бўлиб, бунда шифрлаш ҳар сафар янги тасодифий битлардан иборат калит билан амалга оширилади. Бундай шифрлаш услубидан кўриниб турибдики, шифрлаш ва дешифрлаш учун очиқ матн узунлиги билан тенг бўлган битта калитдан фойдаланилади, ҳамда бу калитнинг фойдаланувчига муҳофазаланган алоқа канали орқали узатилиши талаб этилади. Бундан ташқари, шу усул билан шифрланган матнни дешифрлаш имконияти мураккаб бўлиб, бу унинг муаллифи Г.С. Вернам томонидан ҳам таъкидлаган бўлсада, исботи келтирмаган. Криптология соҳасидаги илмий ишларнинг муаллифлари, 1949 йилгача бўлган даврни қатъий исботсиз – фақат интуиция ва «ишончга» асосланган – илмий асосланмаган криптология даври, деб атайдилар. Таъкидлаб ўтиш жоизки, *Англия криптология хизмати иккинчи жаҳон уруши йиллари даврида математиклар криптологиянинг ривожланишига ўзларининг катта ҳиссаларини қўшишлари мумкинлигига иқрор бўлдилар.* Алан Тьюринг ҳам криптология хизмати мутахассисларидан бири бўлган. К.Э. Шеннонниг 1949 йилда чоп этилган «Махфий тизимларда алоқа назарияси» [4], деб номланган илмий мақоласи илмий асосланган *махфий калитли криптография даврини* бошлаб берди. Шеннон ўзининг электротехника ва математикага оид билимларидан келиб чиқиб, махфий алоқа тизими назариясининг асосини 1948

йилда эълон қилинган – ахборотлар назариясига бағишланган илмий мақоласи асосида яратди [5]. Шеннон ўзининг бу илмий мақолаларида Вернам услубида шифрлашнинг ишончлилиги даражасига тўхталиб, дешифрлаш максимал мураккабликка эгалигини, ҳамда шу услубда шифрлашдан фойдаланувчига махфий алоқа канали орқали узатиладиган махфий калит ҳажми (узунлиги) учун аниқ қуйи чегаранинг қандай бўлишини илмий асосда исботлаб берди. Шенноннинг 1948 йилда эълон қилинган илмий мақоласи криптология соҳасидаги илмий мақолаларнинг пайдо бўлишига олиб келди. Унинг томонидан 1948–1949 йилларда эълон қилинган илмий мақолалари катта аҳамиятли бўлсада, криптология соҳасидаги илмий мақолаларнинг сезиларли кўпайишига олиб келмади. Бунга сабаб, эҳтимолки, Шенноннинг махфий тизимларда алоқа назариясининг махфий калитга асосланган бўлиб, махфий калитни фойдаланувчига етказиш масалалари ечимининг мураккаблиги билан боғлиқлигидадир. 1976 йилда У. Диффи ва М.Е. Хеллманнинг «Криптографияда янги йуналиш» [6], деб номланган мақоласининг эълон қилиниши шу соҳадаги очиқ илмий ишлар ривожининг жуда юқори поғонага кўтарилишига сабаб бўлди. Улар ушбу ишлари орқали, махфий алоқа тизимларида маълумотларни шифрлаш ва дешифрлашда махфий калитнинг тизим фойдаланувчилари орасида махсус муҳофазаланган алоқа канали орқали узатилиши ва қабул қилинишига ҳожат бўлмайдиган илмий-амалий услуб асосларини яратиб, бугунги кунда ҳам ривожланиб ва долзарблашиб бораётган *очиқ (махфий бўлмаган) калитли криптография даврини* бошлаб бердилар. Таъкидлаб ўтиш жоизки, Р. К. Мерклининг У. Диффи ва М.Е. Хеллманга боғлиқ бўлмаган ҳолда, лекин улар билан деярли бир пайтда бошқа илмий журналга берган мақоласида [7] ҳам маълумотларни *очиқ калитли шифрлаш* ғоясининг асослари келтирилган. Аммо Р. К. Меркли мақоласининг нашриётда узоқ вақт эълон қилинмай тўхтаб қолиши, уни муаллифлик ҳуқуқидан деярли маҳрум этди.

§ 1.3. Криптотизимларга қўйиладиган талаблар

Маълумотларни муҳофаза қилиш муҳим масалалари билан бевосита муносабатда бўлмаган кишилар, ахборот-коммуникация тизимида маълумотларни муҳофаза қилиш қоидаларини бузиши мумкин бўлган сабаблар сон ва сифат жиҳатдан серқирралигини, табийки, маълум бир қолипда тасаввур қила олмайди. Қуйида кўп учрайдиган ва нисбатан яққолроқ хис қилиш мумкин бўлган муҳофаза қилиш қоидаларининг баъзи бузилиш сабаблари келтирилган.

Маълумотларни муҳофаза қилиш қоидаларини бузувчининг мақсади ва уни амалга ошириш услублари:

Рухсат этилмаган маълумотларни беруҳсат олиш ва унга эга бўлиш, яъни маълумотларнинг сақланиш қоидаларини бузиш.

Ахборотлар тизимида фойдаланувчиларнинг бирор маълумот юзасидан ўзини жавобгарликдан (маъсулликдан) халос этиш учун ўзини бошқа фойдаланувчи сифатида ифодалаш ёки бошқа фойдаланувчининг ваколатидан фойдаланиш мақсадида:

а) ёлғон маълумотларни ташкиллаштириш;

б) ҳақиқий (қонуний) маълумотларни ўзгартириш;

в) рухсат этилмаган маълумотни олиш учун ўзини шу маълумотни олишга ваколати бўлган шахс сифатида ифодалаш;

г) ёлғон маълумотларни ахборот-коммуникация тизимига тушишига йўл қуйиб бериш ёки ёлғон маълумотларни тасдиқлаш.

Мавжуд бўлган маълумотларни ташкиллаштиришни рад этиш.

Қоидабузар томонидан ёлғон маълумотлар ташкиллаштирилиб, уни ахборот-коммуникация тизимининг бошқа бир фойдаланувчиси томонидан ташкиллаштирилган, деб ифодалаш.

Бирор аниқ кўрсатилган вақтда маълумот олувчига юборилмаган маълумотни юборилган, деб ифодалаш ёки маълумотни юборилган вақтини ёлғон кўрсатиш.

Ҳақиқатан ҳам олинган маълумотларни олинганлигини рад этиш ёки маълумотларнинг ҳақиқий олинган вақтини сохталаштириш.

Ахборотлар тизимидан фойдаланувчиларнинг ўзларига берилган ваколатланган маълумотларни ташкиллаштириш, узатиш, тарқатиш ва бошқа йўналишларда рухсат этилмаган ҳолда кенгайтириш.

Фойдаланувчиларнинг ваколатларини рухсат этилмаган тарзда ўзгартириш.

Конфединциал маълумотни конфединциал бўлмаган маълумотлар каби ифодалаш.

Алоқа тизими фойдаланувчиларининг ўзаро алоқа шаҳобчаларига рухсат этилмаган ҳолда боғланиб, ундан олинган маълумотларни бошқа алоқа тизимларига мунтазам равишда тарқатиб туриш.

Алоқа каналидаги маълумотлар оқимини таҳлил қилиб, маълумотлар жамғармасининг тузилиш тартибига қараб, дастурий таъминот ва бошқа хосликларга кўра, фойдаланувчилар томонидан қандай маълумотлар қачон олинишини ғаразли мақсадларда ўрганиш.

Протокол (маълум тартиб ҳамда қоида) бўйича ҳар қандай ҳолларда ҳам конфединциал қолиши керак бўлган маълумотни конфединциаллигига путур етказган ҳолда, ушбу протокол маълумотлари софлигига шубҳа билан қараш.

Бирор яққол сезилмайдиган муолажа (процедура) билан маълумотларни муҳофаза қилиш алгоритми дастурига ўзгартиришлар киритиш.

Бошқа фойдаланувчиларни ёлғон маълумотлар асосида муҳофаза протоколини бузишга ундаш.

Протоколни бузиш билан ушбу муҳофаза протоколига ишончни йўқотишга олиб келадиган очиқдан-очиқ ҳатти-ҳаракатлар.

Ахборот тизимининг бошқа фойдаланувчиларига маълумотларни сифатли узатилишига, хусусан узатилаётган маълумотга яққол сезилмайдиган техник, дастурий ва бошқа услублар билан ҳалақит берган ҳолда, узатилган маълумотнинг ҳақиқийлигини (аутентификациясини) рад этишга олиб келадиган ҳатти-ҳаракатлар.

Юқорида келтирилган муносабатлар (айниқса келишмовчилик) муаммоларини сабабларини мантиқан таҳлил қилишда ҳам асос қилиб олиниши мумкин. Месси ўзининг «Ҳозирги замон криптология фанига кириш», деб номланган илмий мақоласида қанчалик ишончли криптобардошли алгоритм яратилмасин, бари-бир ўз ечимини кутаётган бошқа криптографик масалалар келиб чиқиши мумкинлигини таъкидлаб, фойдаланувчиларга протокол бўйича ўз вазифаларини бажаришлари учун махфий калитни қандай узатиш ва олинган маълумотлар ҳақиқийлигига ишонч ҳосил қилиш масалалари тўғрисида тўхталади. Шундай масаланинг қўйилиши, калитларни ахборотлар тизими фойдаланувчиларига тақсимлашда келиб чиқадиган муаммоларни ҳал этувчи, очиқ калитли криптография йўналишининг вужудга келишига сабаб бўлди. Бундан ташқари, тизим фойдаланувчиларининг ҳар бири бутун тизим протоколи ичида ўзларининг қисм протоколи бўйича фаолият кўрсатаётганлигига ҳамда бошқа фойдаланувчиларнинг ҳам умумий тизим протоколини бузмаган ҳолда фаолият кўрсатаётганлигига ишонч ҳосил қилиниши, яъни умумий тизим протоколининг бардошлилик даражасига ишонч масалалари ҳам муҳим аҳамият касб этади. Содда қилиб ифодалаганда, ахборот-коммуникация тизимидаги ҳар бир фойдаланувчи шахсий калитининг муҳофазасини таъминлаш долзарб масаладир.

Криптографик алгоритмнинг бардошлилик даражаси қанчалик мустаҳкам бўлишидан қатъий назар, ахборотлар тизимининг фаолият жараёнларини бузиш усуллари мавжуд бўлиб, бу усуллар криптографик алгоритмнинг бардошлилик даражасига боғлиқ эмас. Масалан, калитларни тақсимлаш жараёни протоколининг камчилиги билан, бир нечта фойдаланувчилар ўз калитларини бир-бирларига ошкор қилган ҳолларда, криптографик алгоритимнинг махфийлигига зарар етказилиши мумкин. Умуман олганда ахборот-коммуникация тизими

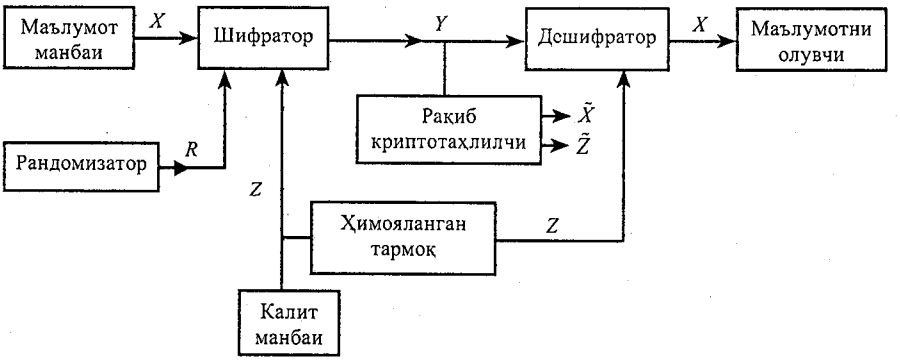
фойдаланувчиларининг протокол бўйича ишлаш жараёни камчилиги криптоалгоритм бардошлилик даражасининг сунъий равишда пасайишига олиб келади. Бундай камчиликларни олдини олишда протоколнинг бир қисми иккинчи қолган қисми тўғрисидаги маълумотни муҳофазаланган ҳолда ахборотлар тизими бўйича очиқ алоқа тармоғи орқали узатилишини таъминлаш имконини бериши керак.

Маълумотларни криптографик услублар билан муҳофазалаш жараёнлари алгоритмик тиллар билан махсус криптобардошли алгоритмларни дастурлаш орқали ёки махсус техник аппаратлар ёрдамида амалга оширилади. Бунда дастурлаш услублари ўзининг қўлланилиши жиҳатидан қулайлиги билан ажралиб туради. Техник аппаратлардан фойдаланиш услублари катта қийматдаги моддий маблағни талаб қилсада, ўзининг самарадорлиги, қулайлиги, ишончлилиги ва шу каби хусусиятлари билан фарқланади.

Ахборотлар тизими муҳофазасининг замонавий криптографик услубларига қуйидаги умумий талаблар қўйилади:

- шифрланган маълумотни асл нусхасига эга бўлиш имконияти фақат дешифрлаш калити маълум бўлгандагина мумкин бўлсин;
- фойдаланилган шифрлаш калитини шифрматннинг бирор маълум қисми бўйича ёки унга мос келувчи очиқ қисми бўйича аниқлаш учун, бажарилиши зарур бўлган амаллар сони калитни аниқ топиш учун бажарилиши керак бўлган барча амаллар сонидан кам бўлмаслиги керак, яъни калит танлаб олиниши керак бўлган тўплам элементларининг сонидан кам бўлмаслиги керак;
- шифрлаш алгоритмининг маълумлиги унинг бардошлилигига салбий таъсир кўрсатмаслиги керак;
- калитнинг ҳар қандай даражадаги (озми, кўпми) ўзгариши шифрланган маълумотнинг жиддий ўзгаришига олиб келиши керак;
- шифрлаш алгоритми таркибидаги элементлар ўзгармас бўлиши керак;
- шифрлаш жараёни давомида маълумотларга киритиладиган қўшимча битлар (элементлар) шифрланган матнда (маълумотда) тўла ва ишончли холда қўлланилган бўлиши керак;
- шифрлаш жараёнида қўлланиладиган калитлар орасида содда ва осонлик билан ўрнатиладиган боғлиқликлар бўлмаслиги керак;

– калитлар таркиби тўпламидан олинган ихтиёрий калит ахборотнинг ишончли муҳофазасини таъминлаши керак;
– криптоалгоритм дастурий ҳамда техник жиҳатдан амалий қўлланишга қулай бўлиб, калит узунлигининг ўзгариши шифрлаш алгоритмининг сифатсизлигига олиб келмаслиги керак.



1.1- расм.

Юқорида келтирилган расмдаги схема Шенноннинг 1949 йилдаги илмий ишларида [4,5] келтирилган «Умумий махфий алоқа тизими схемаси» дан шифрлаш жараёнига «рандомизатор» киритилганлиги билан фарқланади. Рандомизация аутентификация масалаларида муҳим ўрин тутади. Рандомизатор ва рандомизация жараёнларининг маъносини тушунтириб ўтамиз.

Инглиз тилидаги сўзларда «е» ҳарфи бошқа ҳарфларга нисбатан кўпроқ такрорланади. Инглиз тилидаги маълумотларнинг ҳарфларини инглиз тили алифбоси белгиларидан кенг бўлган белгили (ҳарфли) алифбо белгиларига эга бўлган алифбодаги маълумот билан алмаштириб ва бунда, хусусан, «е» ҳарфи ва шу каби бошқа кўп такрорланувчи ҳарфларни кенг бўлган алифбонинг тасодифий белгилари (ҳарфлари) билан алмаштирилиб, кенг алифбо белгилари бир хил (текис) тақсимланган (бир хил частота билан такрорланадиган) шифрмаълумот олиш жараёни рандомизациялаш деб аталади. Бундай рандомизация қилинган шифрмаълумотлар, бирор аниқ тил алифбосида ифодаланган маълумотлардаги сўзларда алоҳида олинган ҳарфларнинг такрорланиш частотасига асосланган ҳолда дешифрлаш услубларига бардошли бўлади. Яъни, рандомизация қилинган шифрмаълумотларни алифбо ҳарфларининг сўзларда такрорланиш частотасига асосланган ҳолда дешифрлаш имкониятлари йўқ. Аммо, маълумотни ҳақиқий олиши керак бўлган томон, олинган шифрмаъ-

лумотни калит билан очиб, маълумотни олгандан сўнг, рандомизация жараёнида қўлланилган тасодифий белгиларни кўп такрорланувчи ҳарфлар билан алмаштириб, ҳақиқий очиқ маълумотни олади. Рандомизация қилинган шифрмаълумотлар «кўпкаррали алмаштиришли шифрмаълумотлар» ёки «тенг частотали шифрмаълумотлар» дейилади. Буюк математик Гаусс кўпкаррали алмаштиришли шифрлаш услубидан фойдаланиш дешифрлаш мумкин бўлмаган шифрмаълумотларни беради, деб хатога йўл қўйган. Чунки, 1.1-расмда келтирилган рандомизаторли схема Шенноннинг «Умумий махфий алоқа тизими схемаси» га фақатгина (вазифаси юқорида баён қилиб ўтилган) рандомизаторнинг қўшилгани билан фарқ қилади.

Шу нарса муҳимки, X, Y ва Z – миқдорларни тасодифий деб тушунмоқ керак. Бунда очиқ матннинг статистик хоссалари маълумотлар манбаи билан аниқланади, махфий калит Z ва рандомизациялаш кетма-кетлиги R миқдорларнинг статистик хоссалари криптографга маълум. 1 – расмда келтирилган схемага асосан, X, Z ва R миқдорлар статистик нуқтаи назардан боғлиқ эмас. Рақиб томон криптотаҳлилчиси фақат узатилаётган маълумотнинг Y – криптограммасига эга бўлган ҳолда X – очиқ маълумотни тиклашга ҳаракат қилиб, Z – махфий калитнинг бирор $\tilde{Z}$ кўринишдаги ҳолатини (баҳосини) олади ва унга кўра очиқ маълумотнинг бирор $\tilde{X}$ кўринишини (баҳосини) олади.

§ 1.4. Криптографик тизимларнинг назарий ва амалий бардошлилиги

Шеннон криптографик тизимларнинг (тизимларнинг) бардошлилиги масаласига икки хил нуқтаи назар билан қаради. Биринчидан, назарий бардошлилик масаласини кўрди: «Рақиб криптотаҳлилчиси криптографик тизимнинг криптотаҳлили учун етали даражадаги техник ва бошқа керакли воситаларга эга бўлса ҳамда криптотаҳлил муддати чегараланмаган бўлса, ушбу криптографик тизимнинг бардошлилиги қандай?» ([5], с. 360). Криптографик тизимнинг назарий бардошлилиги тушунчаси криптографик тизимларни баҳолашга аниқлик киритади, лекин бардошлилиги юқори бўлган криптотизимларнинг яратилиши нуқтаи назардан тушкунликка олиб келади. Амалда кўплаб ҳолларда назарий бардошли криптотизимларнинг яратилиши махфий калит ҳажмининг чексиз катта бўлиб кетиши масаласи билан боғлиқ. Шунинг учун Шеннон криптотизимларнинг амалий бардошлилиги масаласини ҳам кўрди, агар рақиб криптотаҳлилчиси

криптотаҳлил учун етарли даражадаги воситалар билан таъминланмаган бўлса ва анализ муддати чекланган бўлса криптотизимнинг бардошлилиги қандай? Шу ерда алоҳида таъкидлаб ўтамизки, очиқ калитли криптотизимлар амалий бардошли бўлиб, назарий бардошли бўлишлари шарт эмас.

§ 1.5. Шеннонниng мутлақо махфийлик назарияси

Шеннон криптотизимларнинг назарий бардошлилик масалаларида қуйидаги қоидаларни қабул қилди:

- махфий калитдан фақат бир марта фойдаланилади, яъни X – очиқ маълумотнинг M та белгисини шифрлагандан сўнг Z – махфий шифрлаш калитини ва R – рандомизаторини алмаштириш керак;
- рақиб криптотаҳлилчиси Y – шифрмаълумотга эга ва шунинг учун фақат шифрмаълумотга асосланган ҳолда криптотаҳлил услубларидан фойдаланиб очиқ маълумотнинг $\tilde{X}$ – баҳосини ҳамда махфий калитнинг $\tilde{Z}$ – баҳосини олиши мумкин.

Шеннон *мутлақо махфийлик* тушунчасининг таърифини: X – очиқ маълумот ва Y – шифрмаълумот статистик боғлиқ эмас, яъни ихтиёрий очиқ маълумот ва шифрмаълумот учун $P(X=x/Y=y)=P(X=x)$, деб берди. Яна ҳам бошқача қилиб айтганда мутлақо махфийлик, криптотаҳлилчи очиқ маълумотнинг мос шифрмаълумотига эга бўлган ҳолда, очиқ маълумот баҳосининг аниқлигини барча дешифрлаш воситалари ва вақти чегарланмаганлиги имконияти мавжуд бўлганда ҳам тўла ҳолда баҳолай олмайди, деган маънони англатади. Шеннон мутлақо махфийликнинг мана шундай аниқ (математик) таърифини бериб, мутлақо махфий криптографик тизимларнинг мавжудлигини кўрсатди. Ҳақиқатдан ҳам: белгилари шартли равишда $\{0, 1, ..., L-1\}$ тўпламдан (элементлари сони L та бўлган) иборат бўлган, махфий калит узунлиги K ва шифрмаълумот узунлиги N очиқ маълумот узунлиги M билан тенг бўлган, яъни $K=N=M$ бўлган, рандомизация қилинмаган шифрмаълумотни кўрайлик. Шифрлаш жараёни модуль бўйича қўшиш амалига асосланган бўлсин, яъни

$$Y_i=X_i \oplus Z_i, \quad i=1,2, ..., M. \tag{1.2}$$

Комбинаторика курсидан маълумки, L та белгидан фойдаланиб узунлиги M га тенг бўлган, яъни белгилари сони M та бўлган барча мумкин бўлган (шартли) сўзлар тўпламининг элементлари сони L^M та бўлади. Шунинг учун мумкин бўлган барча калитлар тўпламидан

бирор элементнинг (калитнинг) таҳлил қилинаётган шифрмаълумотнинг (криптограмманинг), юқорида айтиб ўтилган параметрли калити эканлиги эҳтимоли $P(Z=z)=L^{-M}$ бўлади. Мос равишда $x_i \in X_i$ ва $y_i \in Y_i$ бўлган элементлар учун (1.2) тенгликни қаноатлантирувчи ягона $z_i \in Z_i$ элемент мавжуд. Шунинг учун нинг статистик хоссаларига боғлиқ бўлмаган ҳолда ихтиёрий ва элементлар учун $P(Y=y/X=x)=L^{-M}$ тенглик ўринли бўлади. Шундай қилиб, ва статистик боғлиқ эмас ҳамда L модуль бўйича қўшиш амалига асосланган Вернам криптоалгоритми Шеннонниng мутлақо махфийлик таърифи шартларини қаноатлантиради. Вернам криптоалгоритми «шифр-блокнот» номи билан иккинчи жаҳон уруши ва ундан кейинги даврларда ҳам разведка хизмати ва бошқа шу каби махсус хизмат ходимлари томонидан кенг қўлланилган. Разведка ходимларига бирор (тасодифий) махфий калитли, яъни узунлиги шифрланаётган маълумотнинг узунлигига (криптограмма узунлигига) тенг бўлган 0 дан L гача бўлган сонлардан тузилган бирор тасодифий сонлар кетма-кетлигидан иборат «шифр-блокнот» берилиб, ундан маълумотни шифрлаш учун фақат бир марта фойдаланиш мумкинлиги таъкидланган. Криптография соҳаси мутахассислари бундай усулнинг мутлақо бардошлилигига тўла ишонч ҳосил қилган бўлсаларда, бу усулнинг ҳақиқатан ҳам тўла бардошлилигини, биринчи бўлиб Шеннон назарий жиҳатдан исботлаб берди. Вернам криптотизимидан фойдаланишда махфий калитнинг битта белгисини, шифрланиши керак бўлган маълумотнинг ҳам фақат битта белгисига боғлиқлиги, бу криптотизимни махфийлиги юқори даражада таъминланиши керак бўлган кичик ҳажмдаги матнларгагина яроқли бўлиб (мисол учун Тошкент-Вашингтон алоқа тизимида), катта ҳажмдаги маълумотларга қўлланиш имкониятларини баъзан чегаралаб қўяди. Чунки катта ҳажмдаги маълумотни муҳофаза қилишда Вернам криптотизимининг қўлланилиши махфий калитнинг ҳам ҳажмини очиқ маълумот ҳажми даражасида катта бўлишини талаб этади.

§ 1.6. Мутлақо махфийлик мисни таъминловчи криптотизимларнинг калитларига қўйиладиган талаблар

Назарий бардошлилик масалаларига тегишли бўлган саволлар билан шуғилланишда Шеннонниng ахборотлар назариясига олиб кирган асосий сонли қиймат – «*ноаниқлик*» ёки «*энтропия*» деб аталувчи тушунчадан фойдаланамиз. Ноаниқлик – тасдифий миқдорлар эҳтимоллигига мос келувчи тақсимот функцияси логарифимининг

манфий ишора билан олинган ифодасини математик кутилмасидан иборат. Яъни $H(X/Y)$ (бу ифода бирор аниқ Y миқдорда номаълум X миқдорнинг баҳосининг ноаниқлиги, деб ўқилади) ноаниқлик $P(X=x/Y=y)$ эҳтимоллик қиймати логарифмининг математик кутилмаси орқали қуйидаги тенглик билан ифодаланади:

$$H(X/Y)=\sum_x\sum_y P(X=x/Y=y)[-\log P(X=x/Y=y)]$$

бу ерда: йиғиндилар X, Y – тасодифий миқдорларнинг барча мумкин бўлган қийматлари бўйича ҳисобланади. Ноаниқликлар, табиий бўлган қуйидаги

$$H(X/Y)=H(X)+H(Y/X)$$

қоидани қаноатлантиради. Ноаниқликлар тушунчаси орқали берилган қуйидаги ифодалар:

$$H(Y/X, Z, R)=0, \tag{1.3}$$

$$H(X/Y, Z)=0, \tag{1.4}$$

мос равишда қуйидагича тушунилади: (1.3) тенглик ўринли бўлади шунда ва фақат шунда, қачонки, X, Y, Z миқдорлар биргаликда миқдорни бир қийматли аниқласа ва (1.4) тенглик ўринли бўлади, қачонки, Y ва Z миқдорлар биргаликда миқдорни бир қийматли аниқласа. Мутлақо махфийлик таърифини

$$H(X/Y)=H(X), \tag{1.5}$$

кўринишида ифодалаш мумкин. Чунки, бу охирги тенглик фақат X ва Y миқдорлар статистик нуқтаи назардан боғлиқ бўлмагандагина ўринли бўлади.

Махфий калитли криптотизимлар учун ушбу тенгсизлик

$$H(X/Y) \le H(X, Z/Y)=H(Z/Y)+H(X/Y, Z)=H(Z/Y) \le H(Z), \tag{1.6}$$

ўринли бўлади. Бу ерда (1.4) тенгликдан ва маълум малумотларнинг ҳажмини қисқариши табиий ҳолда ноаниқлик қийматининг ўсишига олиб келишишидан фойдаланилган. Агарда криптотизим ахборотларнинг мутлақо махфийлигини таъминласа, у ҳолда (1.5) ва (1.6) тенгликлардан ушбу

$$H(Z) \ge H(X) \tag{1.7}$$

тенгсизлик келиб чиқади.

Юқоридаги (1.6) тенгсизлик – мутлақо махфий тизимлар учун Шеннон чегарасини аниқлайди, яъни махфий калитнинг ноаниқлик

қиймати шу калит билан шифрланадиган маълумотнинг ноаниқлик қийматидан кичик бўлмаслиги керак. Агарда махфий калит элементлари сони L_z бўлган алифбонинг белгиларидан тузилган бўлиб, унинг ҳажми (узунлиги) га тенг (яъни калитни ташкил этувчи белгиларнинг умумий сони га тенг) бўлса, у ҳолда махфий калитнинг ноаниқлик қиймати баҳосини ифодаловчи ушбу

$$H(Z) \le \log(L_Z^K) = K \log L_Z, \qquad (1.8)$$

тенгсизликда тенглик фақат махфий калит мутлақо тасодифий бўлгандагина бажарилади. Худди шу каби очиқ маълумот элементлари сони L_x бўлган алифбонинг белгиларидан тузилган бўлиб, уни ташкил этувчи белгиларнинг умумий сони M бўлса, у ҳолда очиқ маълумот қуйидаги ноаниқлик қийматининг баҳосида

$$H(X) \le M \log L_X, \qquad (1.9)$$

тенглик фақат очиқ маълумот мутлақо тасодифий бўлгандагина бажарилади. Шундай қилиб, агарда $L_x = L_z$ бўлиб, очиқ маълумот бутунлай тасодифий бўлса, охирги (1.8) ва (1.9) муносабатлардан Шеннон чегарасини аниқловчи (1.7) муносабатдан $K \ge M$ тенгсизликка эга бўламиз. Бу муносабат эса калитнинг ҳажми (узунлиги) очиқ маълумот ҳажмидан кам бўлмаслиги кераклигини кўрсатади. Калит узунлигининг қуйи чегарасига Вернам шифрлаш криптотизимидан фойдаланилганда эришилади ва бунда $K = M$ бўлиб, калит узунлиги очиқ маълумот узунлигига тенг бўлади.

§ 1.7. Мукаммал бўлмаган шифрларни очиш

Муккамал бўлмаган шифрларни очиш деганда, бирор берилган (эга бўлинган) шифрмаълумотга (криптграммага) асосан криптотаҳлил услубларидан фойдаланиб, шу берилган криптограммага мос келувчи очиқ маълумотни тиклаб, шифрлаш алгоритимини топиш жараёни тушунилади. Шеннон, криптотаҳлилчи томонидан мукаммал бўлмаган шифрларнинг очилиши, назарий жиҳатдан мумкинлиги масалаларини кўриб чиқди. Бунинг учун у калитнинг ишончсизлиги (бардошсизлиги) функциясини

$$f(n) = H(Z/Y_1, Y_2, ..., Y_n) \qquad (1.10)$$

криптограмманинг дастлабки та белгисига асосан таҳлил қилинаётган шифрмаълумот (криптограмма) учун калитнинг ноаниқлик ўлчови сифатида киритди. Бундан ташқари, Шеннон таҳлил қилинаётган

криптограмманинг Z-калитини бир қийматли аниқловчи дастлабки n та белгисидан иборат (Y_1, Y_2, ..., Y_n) белгилар тўпламининг энг кичик ҳажмда бўлганининг элементлари сонини, яъни $f(n)=0$ тенгликни қаноатлантирувчи энг кичик сонини ягоналик масофаси сифатида аниқлади. Агарда шифрланган маълумотнинг n тадан кам бўлмаган миқдордаги белгилари ҳам маълум бўлса (таҳлил қилинаётган шифрмаълумот u та ҳар хил белгиларнинг комбинацияларидан иборат бўлса), у ҳолда Y_1, Y_2, ..., Y_n белгиларга асосан махфий калитнинг фақат битта қийматини топиш мумкин, яъни етарли даражада вақт ва бошқа керакли воситалар билан таъминланган криптотаҳлилчи шу таҳлил қилинаётган криптограмманинг махфий калитини топиб шифрни оча олади. Бирор берилган «тасодифий шифр» учун Шеннон муносабат

$$u=\frac{H(Z)}{r\log L_y}, \tag{1.11}$$

ўринли эканлигини кўрсатди. Бу ерда:

$$r=1-\frac{H(X)}{N\log L_y}, \tag{1.12}$$

тенглик билан аниқланувчи сон r – белгилари сони L_y бўлган алифбода тузилган, ҳажми N бўлган (яъни криптограммани ташкил этувчи белгиларнинг умумий сони N бўлган) криптограммада алифбо барча белгиларининг такрорланишини (даврларининг ўртача қийматини) аниқлайди. Кўплаб криптотизимларда $N=M$ ва $L_x=L_y$, бундай ҳолда инглиз тилидаги очиқ матнлар учун $r=3/4$. Агарда $L_x=L_y$ бўлса ва калитнинг ягоналик масофаси мутлақо тасодифий бўлса, у ҳолда (1.8) ифодани ҳисобга олиб (1.11) ифодага кўра ушбу $u\approx\frac{K}{r}$ муносабатга эга бўламиз.

Шундай қилиб, агарда инглиз тилидаги маълумотни шифрлаш криптотизимида $L_x=L_y=L_z$, муносабатлар ўринли бўлса, у ҳолда бундай криптограммани шифрмаълумотнинг $N=(4/3)K$ та бўлган (ёки $u=(4/3)K$ тадан кам бўлмаган) белгиларидан фойдаланиб очиш мумкин. Мисол учун, хажми 56 битдан (ASCII кодида саккизта 7 битли белгилардан) иборат бўлган махфий калитни шифрмаълумотнинг дастлабки 11 та 7 битли белгисини таҳлил қилиш билан тиклаш мумкин.

Криптотизимларнинг махфий калитларини ягоналик масофаларини баҳолашда Шеннон келтирган (1.11) тенглик билан аниқланувчи ифодадан кенг фойдаланилади.

Ҳақли равишда, бўлган ҳолда (1.11) ва (1.12) ифодаларнинг мазмуни қандай бўлиши тўғрисида савол туғилади, яъни $N = M$, $L_x = L_y$, бўлиб, очиқ маълумот мутлақо тасодифий, $H(X) = M\log L_x = N\log L_y$, бўлган ҳолда. Амалда мана шундай ҳолатга кўпроқ дуч келинади. Юқорида қўйилган саволга қуйидагича жавоб берилади: бир томондан $r \to 0$ бўлса (1.14) тенгликда $u \to \infty$ бўлиб, калитнинг узунлиги K очиқ маълумот узунлиги M га нисбатан жуда кичик бўлганда ҳам, яъни $K < M$ бўлганда ҳам криптотаҳлилчи ҳеч қачон криптотизим асосини ташкил этувчи алгоритмни оча олмайди;

иккинчи томондан эса $K < M$ бўлиши (1.10) тенгсизликка зид бўлиб, бундай криптотизим мутлақо махфийликни таъминлай олмайди. Бундай парадоксни (юзаки қараганда зиддиятли ҳолатни): мутлақо махфий криптотизимда криптограмма (шифрланган маълумот) Y – очиқ маълумот X ҳақида ҳеч қандай маълумотни ўз ичига олмаслиги, яъни Y ва X миқдорларнинг статистик нуқтаи назардан боғлиқ эмаслиги ҳамда X миқдорни Y миқдор бир қийматли аниқлаши учун Y миқдорнинг X миқдорга нисбатан боғлиқлиги ҳақида мумкин қадар кўпроқ маълумот бўлиши талаб этилиши билан тушунтирилади. Ҳақиқатан ҳам мутлақо тасодифий бўлган X-очиқ маълумотнинг Y-криптограммасини очиш учун Z – махфий калит мутлақо тасодифий танланади. У ҳолда ҳар бир Y – криптограммага X – очиқ маълумотнинг ва Z – махфий калитнинг L_Z^K тадан мумкин бўлган ифодалари мос келади ва бу ифодаларнинг ҳар бири бир хил эҳтимоллик билан Y – криптограммага мос келади. Маълум бўлган Y – криптограммага ҳақиқатан ҳам мос бўлган очиқ маълумот X ва махфий калит Z ларни, юқорида айтилган, мумкин бўлган барча тенг эҳтимолли L_Z^K та мос ифодалар ичидан танлаб олиш учун эса криптотаҳлилчи ҳеч қандай қўшимча маълумотга эга эмас. Худди мана шу ҳолатдан Шеннон табиий равишда: маълумотларни сиқиш, яъни маълумотларни шифрлашда криптограмманинг узунлиги (ҳажми) очиқ маълумот узунлигидан (ҳажмидан) кичик бўлиши – криптографияда фойдали восита, деб тўғри хулоса чиқарди. Шундай қилиб, маълумотларнинг ҳажмини сиқишнинг мукаммал алгоритми маълумотлар манбаини мутлақо тасодифий маълумотлар манбаига айлантиради. Аммо шу пайтгача маълумотлар манбаи учун бир пайтнинг ўзида мукаммал ва амалий жиҳатдан қулай бўлган маълумотларни сиқиш алгоритми яратилган эмас. Шундай бўлсада, маълумотларни сиқишнинг мукаммал бўлмаган алгоритмлари ҳам миқдорнинг сезиларли камайишига ва бунинг

натижасида ягоналик масофаси миқдорининг ўсишига олиб келади. Дастлаб, маълумотлар техник воситаларсиз таҳлил қилиниб келинган даврларда ҳам криптографлар очиқ матндан маълумотни қабул килувчи томонидан осон тикланиши мумкин бўлган алифбо белгиларини чиқариб ташлаганлар. Мисол учун: СҚШГАМСОЛ.

Ягоналик масофасининг (1.11) ифодасини Шеннон рандомизациялашни ҳисобга олмаган ҳолда келтириб чиқарган. Бу ифода рандомизациялаштирилган шифрмаълумотлар учун ҳам ўринли бўлиши учун (1.12) ифодада миқдорни миқдор билан алмаштириб қараш лозим. Бундан эса рандомизациялаш ҳам миқдорнинг камайишига олиб келади. Тажрибалар криптографлар ишида очиқ матнни ташкил этувчи алифбо белгиларининг статистик хоссаларини асли ҳолатини яширишда (ўзгартиришда) берилган матнга қўшимча белгилар киритишни қулай усул эканлигини кўрсатди. Мисол учун:

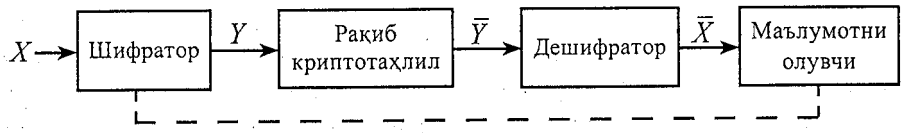
ҚЎШХИМХЧАХКХРИХТИШГАХМХИСОЛХ

ифода фикримизнинг далили бўла олади.

§ 1.8. Ишончлилик ва алдов

Биз юқорида криптографиянинг мақсади – ахборот ёки маълумотларнинг махфийлигини ва уларнинг ҳақиқийлигини таъминлашдан иборат эканлигини бир неча бор таъкидладик. Аммо, ахборот ёки маълумотларни конфиденциаллиги ва уларнинг ҳақиқийлиги масалалари алоҳида хоссаларга эга. Ҳақиқатан ҳам криптограммани олиб ва уни дешифрлаб керакли очиқ маълумотни олгандан сўнг, бу криптограммани махфий калитга ваколатсиз эга бўлган шахс томонидан юборилмаганлигига, яъни махфий калитга эгаликка ваколати бўлган шахс томонидан юборилганлигига кандай тўла ишонч ҳосил қилиш мумкин? Мана шундай йўналишдаги масалаларнинг ечими билан, яъни аутентификация масалалари билан Г. Дж. Симмонс шуғулланиб, Шеннонниng махфий алока назарияси каби, ўзининг *аутентификация* назариясини яратди [12].

Аутентификация тизимининг назарий бардошлилиги ҳақидаги масалани Симмонс, криптотаҳлилчи 1.1 – расмда бўлган ҳолатидан қулайроқ бўлган 1.2 – расм ҳолатида бўлганда, яъни криптотаҳ лилчи сохталаштирилган $\bar{Y}$ – криптограммани (маълумотни) олиши керак бўлган шахснинг дешифраторига юбориш имкониятига эга бўлган ҳолатдан келиб чиқиб ўрганди.



1.2- расм.

Бундай криптограмманинг сохталиги аниқланиб, у маълумотни олувчи шахсга жўнатилмайди. Шунинг учун 1.2-расмда шифратор билан маълумотни олувчи шахс орасидаги алоқа штрихли чизиқлар билан белгиланган.

Симмонс, Шеннон каби, махфий калит Z (ҳақиқий бўлган) криптограммани яратишда фақат бир марта фойдаланилади, деб қабул қилди. Шундай бўлганда ҳам криптотаҳлилчи қуйидаги мухим имкониятларга эга эканлигини, Симмонс назардан қочирмади:

Рақиб криптотаҳлилчиси ҳақиқий бўлган Y – криптограмманинг келишини кутмай сохта $\bar{Y}$ – криптограммани ҳақиқий криптограмма юборилиши керак бўлган шахсга етказиши мумкин (бундай ҳолат *иммитация* қилиш ҳолати дейилади) ва шунинг учун 1.2-расмда маълумот манбаи шифратори билан криптотаҳлилчи орасидаги алоқа штрих чизиқлар билан кўрсатилган. Агар ҳақиқий криптограммани олиши керак бўлган шахснинг дешифратори сохта $\bar{Y}$ – криптограммани ҳақиқий криптограмма Y – сифатида қабул қилса иммитация қилиш муваффақиятли кечган ҳисобланади (хаттоки, бунда кейинроқ сохта $\bar{Y}$ – криптограмма билан ҳақиқий Y – криптограмма мос тушса ҳам).

1. Агарда сохта криптограмма $\bar{Y}$ дешифратор томонидан ҳақиқий криптограмма Y сифатида қабул қилиниб, $\bar{Y}$ га мос келувчи дешифрланган очиқ матн $\bar{X}$ ҳақиқий криптограмма Y дешифрлангандаги очиқ матн X га мос келмаса, яъни $\bar{X} \neq X$ бўлса, рақиб криптотаҳлилчиси сохта $\bar{Y}$ – криптограммани ҳақиқий Y – криптограммадан кейин ҳам етказиши мумкин (бундай ҳолат *алмаштириш* дейилади) ва алмаштириш жараёни муваффақиятли кечган ҳисобланади. Иммитация ва алмаштириш жараёнларини муваффақиятли кечиши эҳтимолликларининг энг юқори қийматларини мос равишда P_I ва P_S деб белгилаймиз. Симмонс, миқдорни *алдов эҳтимоллиги* миқдори сифатида киритди, яъни криптотаҳлилчи рақиб томонни мана шундай эҳтимоллик билан алдай олади. Аутентификация назариясининг масалалари кўп қиррали ва ўзига хос хусусиятларга эга бўлиб, алоҳида илмий изланишларни талаб этади.

§1.9. Амалий бардошлилик

Юқорида кўриб ўтганимиздек, бирор аниқ чекли ҳажмдаги калит билан шифрлаш криптотизимлари (яъни $K<H(X)$ бўлган) ягоналик масофаси миқдори чексиз катта қийматга эга бўлиши мумкин ва натижада, бундай криптотизимлар мутлақо махфийликни таъминлайди. Шеннон шундай тизимларни идеал шифрлаш криптотизимлари деб атади ва бундай тизимларни яратишда ҳал қилиб бўлмайдиган табиий тўсиқлар келиб чиқишини ҳам таъкидлади. Амалдаги кўплаб криптотизимларнинг бардошлилиги уларнинг шифрлаш алгоритмининг калитини назарий жиҳатдан топиш масаласининг ечиб бўлмаслигига эмас, балки, алгоритм калитини топиш масаласининг амалий жиҳатдан мураккаблигига асосланган бўлиши керак. Ҳақиқатдан ҳам, Шеннон шифрмаълумот ва унинг ташкил этувчи белгиларига асосан махфий шифрлаш калитини топиш учун энг замонавий воситалардан фойдаланган ҳолда сарф бўладиган ўртача вақтни шифрнинг (яъни шу шифрлаш криптотизими алгоритмининг) иш характеристикаси деб атади. Ҳажми n бўлган шифрнинг иш характеристикасини $W(n)$ деб белгилаймиз. $W(n)$ катталикни $n \to \infty$ бўлгандаги қиймати муҳим бўлиб, бу қийматни чексиз ҳажмдаги шифрмаълумотни очиш учун бажарилиши зарур бўладиган ўртача ишни (ёки иш вақтини) ифодалайди. Шундай қилиб, бирор криптотизим учун $W(n)$ миқдорни аниқлашда қуйидаги алоҳида мураккабликка эга бўлган масала келиб чиқади – шифрни очишнинг энг яхши услубини топиш, яъни оптимал (самарали) таҳлил билан $W(n)$ миқдорнинг қуйи қийматини аниқлаш. Ҳозирда криптотизимлар учун $W(n)$ миқдор $n \to \infty$ даги қийматининг қуйи чегарасини аниқловчи бирор умумий илмий асосланган услуб мавжуд эмас. Амалдаги криптотизимлар одатда иш характеристикасининг эришилган баҳоси деб аталувчи $W_h(\infty)$ миқдор билан баҳоланади. $W_h(\infty)$ миқдор шифрланган маълумотнинг n та белгисига асосланиб, берилган шифрни энг самарали воситалардан фойдаланган ҳолда таҳлил қилиб, шифрлаш алгоритмининг калитини топиш зарур бўлган, яъни сарфланиши керак бўлган ўртача вақт миқдори. Криптотаҳлил услубларининг такомиллашиб бориши $W(n)$ ва $W_h(\infty)$ миқдорларнинг тегишли криптотизимлар учун мумкин қадар аниқ қийматларини топиш имкониятларининг кенгайиб боришини таъминлайди.

§ 1.10. Мутлақо бардошли амалий шифрлаш алгоритмларининг мавжудлиги

Шеннон ўзининг «Махфий тизимларда алоқа назарияси» деб номланган илмий ишида шифрлаш алгоритмларининг амалий бардошлилиги масалаларини таҳлил қилиб, бу борадаги натижаларни математик теоремалар кўринишида ифодалаш учун керак бўладиган тушунчаларни (аксиомаларга ўхшаш бўлган дастлабки асосларни) аниқлаш мушкул эканлигини таъкидлаган. Иш характеристикаси $W(n)$ ёки унинг $n \to \infty$ бўлгандаги $W_n(\infty)$ қийматларини ҳисоблашнинг бирор аниқ умумий қоидасини яратиш масаласи ечилган эмас. K та битли калит билан ахборот матни белгиларини группалаш ва рандомизациялаш билан шифрлаш криптотизими учун $n \to \infty \approx 2^{K/2}$ эканлиги маълум бўлиб, бунда иш характеристикаси бирлиги натижаси 0 ва 1 бўлган оддий имкониятларни танлашдан иборатдир. Ахборотни қабул қилиб олувчи шахс олинган криптограммани дешифрлашга киришиши учун 2^K та битнинг алоқа тизими воситалари орқали етиб келишини кутишга мажбурдир. Агарда биз дешифрлаш имконияти туғилгунча миллион йил кутишга рози бўлсак, у ҳолда ишонч билан айтишимиз мумкинки, криптотаҳлилчи бу криптограммани очиши учун минг йил керак бўлади.

Юқоридаги фикр ва мулоҳазаларда ахборот-коммуникация тармоқларида маълумотлар алмашинуви технологияларининг бугунги ривожланган инсоният жамиятининг турли соҳаларига кенг ва чуқур кириб бориб, ахборотлар мажмуаси – маълумотлар тўплами барча кундалик фаолият жараёнларини мақсадли режалаштиришнинг муҳим омили эканлигини эътиборга олиб, у ёки бу соҳага тегишли бўлган муҳим ахборотлар мажмуасини муҳофазасини таъминлаш масалалари ва уларнинг ечимлари ҳақида баён қилинди. Қуйида, ахборот-коммуникация тармоқларида ахборотларнинг криптографик муҳофазасини таъминлашнинг асосий масалалари келтирилади, булар:

– ахборотнинг конфиденциаллигини таъминлаш;

– ахборотнинг тўлалигини (ўзгармаганлигини) таъминлаш;

– ахборотнинг аутентификациясини (маълумот субъектларини ҳақиқийлигини) таъминлаш;

– ахборотнинг муаллифини ва муаллифликдан бош тортмаслигини таъминлаш;

– криптографик алгоритмлар учун криптобардошли калитлар ишлаб чиқариш ва уларни тармоқ фойдаланувчиларига муҳофазаланган ҳолда тарқатилишини бошқариш.

Ахборот муҳофазасининг санаб ўтилган масалаларини криптографик усуллар билан ечиш воситаси шифрлаш алгоритмларидир.

Ахборот конфиденциаллигини таъминлашнинг асосий мақсади очиқ алоқа тармоғида конфиденциалликни таъминлаган ҳолда конфиденциал маълумотларни алмашинуви масалаласини ечишдан иборат. Ахборот-коммуникация тизимлари очиқ алоқа тармоғи фойдаланувчиларининг маълумотлар мажмуасидан турли мақсадларни, баъзан эса ўзаро қарама-қарши мақсадларни назарда тутиши, конфиденциаллигини кафолатли таъминлаган ҳолда маълумотлар алмашинувини амалга оширишни тақазо этади. Ўзаро қарама-қарши мақсадларни назарда тутувчи томонлар криптотаҳлилчилари бугунги ривожланган ахборот технологиялари ютуқларидан фойдаланиб, алоқа тармоғига боғланиб, маълумотлар алмашинувини кузатиш (мониторингини олиб бориш), шифрлаш алгоритмларини қўллаш билан махфийлиги таъминланган маълумотларга эга бўлиш, уларни дешифрлаш чора – тадбирларини амалга оширишга ҳаракат қилиш имкониятларига эга. Бундай ҳатти-ҳаракатлар (ҳужумлар) икки турда бўлади: *фаол* (*актив*) ва *фаол бўлмаган* (*пассив*). Фаол бўлмаган ҳужумлар эшитиш, алоқа тармоғида алмашинаётган маълумотларни мазмунини кузатиш ва таҳлил қилиш, шифрланган маълумотларни ёзиб олиш ва дешифрлаш каби хатти-ҳаракатлар билан боғлиқ. Фаол ҳужумлар маълумотлар алмашинуви жараёнига тўсқинлик қилиш, узатилаётган маълумотлар мазмунини ўзгартириш каби хатти-ҳаракатларни ўз ичига олади.

Очиқ маълумот M, шифрланган маълумот C, шифрлаш алгоритми E ва калити k_1, дешифрлаш алгоритми D ва калити k_2, деб белгиланса, шифрлаш жараёни $E_{k_1}(M)=C$, дешифрлаш жараёни $D_{k_2}(C)=D_{k_2}(E_{k_1}(M))=M$ кўринишда ифодаланади.

Турли хусусиятли – ҳужжатли, овозли, тасвирли маълумотларнинг барчасини шифрлаб, алоқа тармоғида узатилиши ва қабул қилинишини кафолатли мухофаза қилинишини самарали кечишини таъминловчи ягона криптографик алгоритм мавжуд эмас. Чунки, криптографик воситалар маълумотларнинг физик хусусиятлари, уларнинг конфиденциаллик даражаси, ҳажми, сигнал кўринишида ифодаланиш усули, алоқа тармоғида узатилиш технологиялари хусусиятлари, қўлланиладиган техник қурилмаларнинг қиймати, фойдаланишга қулайлиги каби хосликларни **ҳисобга** олган ҳолда танланади.

Ахборот тўлалигини таъминлашнинг асосий мақсади очиқ алоқа тармоғида маълумотлар алмашинуви **жараёнларида рақиб** томоннинг

алмашинаётган маълумотларни ўз манфаатидан келиб чиққан ҳолда ўзгартиришларини аниқлашнинг имконини берувчи криптографик воситаларни (алгоритмларни) яратишдан иборат. Бунинг учун узатилаётган маълумотга, уни қабул қилувчи томон учун, маълумотни ўзгарган ёки ўзгармаганлигини текшириш имконини берувчи, махсус алгоритм билан ҳисобланадиган – *назорат йиғиндиси* ёки *маълумотнинг аутентификация коди*, деб аталувчи қўшимча қўшилади. Бундай қўшимча қўшиш усулининг кодлаштириш усулидан фарқи, назорат йиғиндиси ҳисобланадиган криптографик алгоритмнинг махфий калитга боғлиқлигидадир. Махфий калитни билмаган ҳолда узатилаётган маълумотга рақиб томонидан ўзгартириш киритиш эҳтимоллиги деярли йўқ. Шундай эҳтимоллик ўлчови *шифрнинг имитобардошлилиги* – фаол ҳужумларга бардошлилик ўлчови дейилади. Берилган M – маълумотни аутентификациясини (ҳақиқийлигини) текшириш имконини берувчи қайд қилинган узунликдаги қиймат қабул қиладиган назорат йиғиндисини ҳисоблаш алгоритмининг калит деб аталувчи махфий k – параметрга ва M – маълумотга боғлиқ функцияси $h_k(M)=S$ – хэшлаш функцияси деб юритилади. Хэш-функцияга қуйдаги талаблар қўйилади:

– калитни билмаган ҳолда берилган M – маълумотнинг $h_k(M)=S$ қийматини ҳисоблаш мумкин эмас;

– берилган M – маълумот ва унинг хэш – функция қийматини $h_k(M)=S$ билган ҳолда шу M – маълумотдан фарқли $M \neq M_1$, лекин хэш-функция қиймати тенг $h_k(M)=h_k(M_1)=S$ бўлган M_1 – маълумотни топиш имкони йўқ.

Келтирилган биринчи талаб маълумотнинг қалбакилаштирилишига йўл қўймасликни таъминлайди, иккинчи талаб эса бирор маълумотни бошқа маълумот билан алмаштириш имкониятини чекланишини таъминлайди.

Ахборотнинг аутентификацисини (маълумот субъектларининг ҳақиқийлигини) таъминлашнинг мақсади ахборот алмашинуви тўғри ўрнатилганлигини, томонларнинг ҳақиқийлигини, маълумот ва унинг муаллифи каби субъектларнинг ҳақиқийлигини текширишни таъминлашдан иборат.

Ахборот алмашинуви жараёни (сеанси) тўғри ўрнатилганлигини аутентификациси: тармоқ бўғинлари боғланишларининг тўғри амалга оширилганлигини текширишни, рақиб томонидан маълумотларни қайта узатиш имконияти йўқлигини ва маълумотлар алмашинувининг ўз вақтида кечишини таъминлаш каби тадбир-

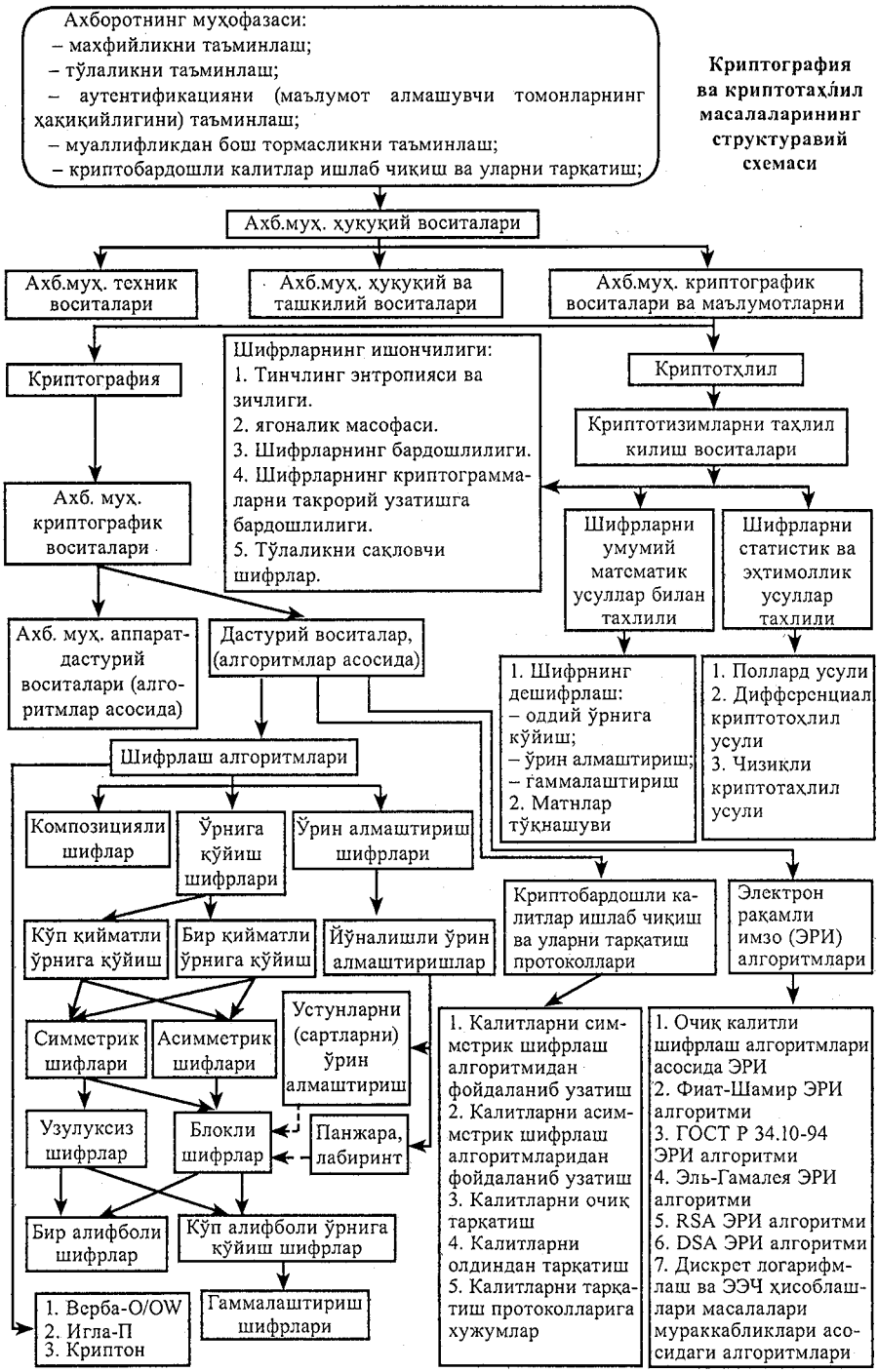
ларни ўз ичига олади. Бунинг учун узатилаётган маълумотларга осон текшириладиган қўшимча параметрлар киритишдан фойдаланилади.

Ахборот муаллифлигини ва муаллифликдан бош тортмаслигини таъминлашнинг мақсади бир-бирига ишонмайдиган томонларнинг маълумотлар алмашинуви жараёнларида жўнатувчи маълумотни юборганлигини рад этиб, бу маълумотни олувчининг ўзи тузганлигини даъво қилиши ёки ҳақиқатан ҳам, олувчи ўзи қабул қилиб олган маълумотни ўзгартириши, қалбакилаштириши ва янги маълумот тузиши, сўнгра бу маълумотни жўнатувчидан олганлигини даъво қилиши мумкин бўлган ҳолатларда келиб чиқадиган муаммо ва низоларни тўғри ҳал этишдан иборат. Бундай муаммо ва низоларни ҳал этишнинг фундаментал механизми электрон рақамли имзо (ЭРИ) ҳисобланади. ЭРИ жўнатилаётган маълумотни ташкил этувчиларига ва жўнатувчининг махфий калитига боғлиқ ҳолда ҳисобланиб, жўнатилаётган маълумотга илова қилинадиган рақамли кетма-кетликдан ташкил топади. Маълумотни қабул қилувчи томон қабул қилинган маълумотни бу рақамлар кетма-кетлиги ва жўнатувчининг очиқ калитига боғлиқ ҳисоблашларни бажариб, маълумотнинг аутентификацисини амалга оширади. Шундай қилиб ЭРИ жарёни алгоритми икки қисмдан рақамли имзони ҳисоблаш (шакллантириш) ва рақамли имзони текширишдан иборат. Рақамли имзони ҳисоблаш махфий калитга боғлиқ бўлгани учун ҳам уни фақат маълумотни жўнатувчи (яъни маълумотнинг ҳақиқий муаллифи) тўғри шакллантира олади. Рақамли имзони текшириш очиқ калит орқали амалга оширилади, яъни исталган томон учун унинг тўғрилигини текшира олиш имконияти мавжуд.

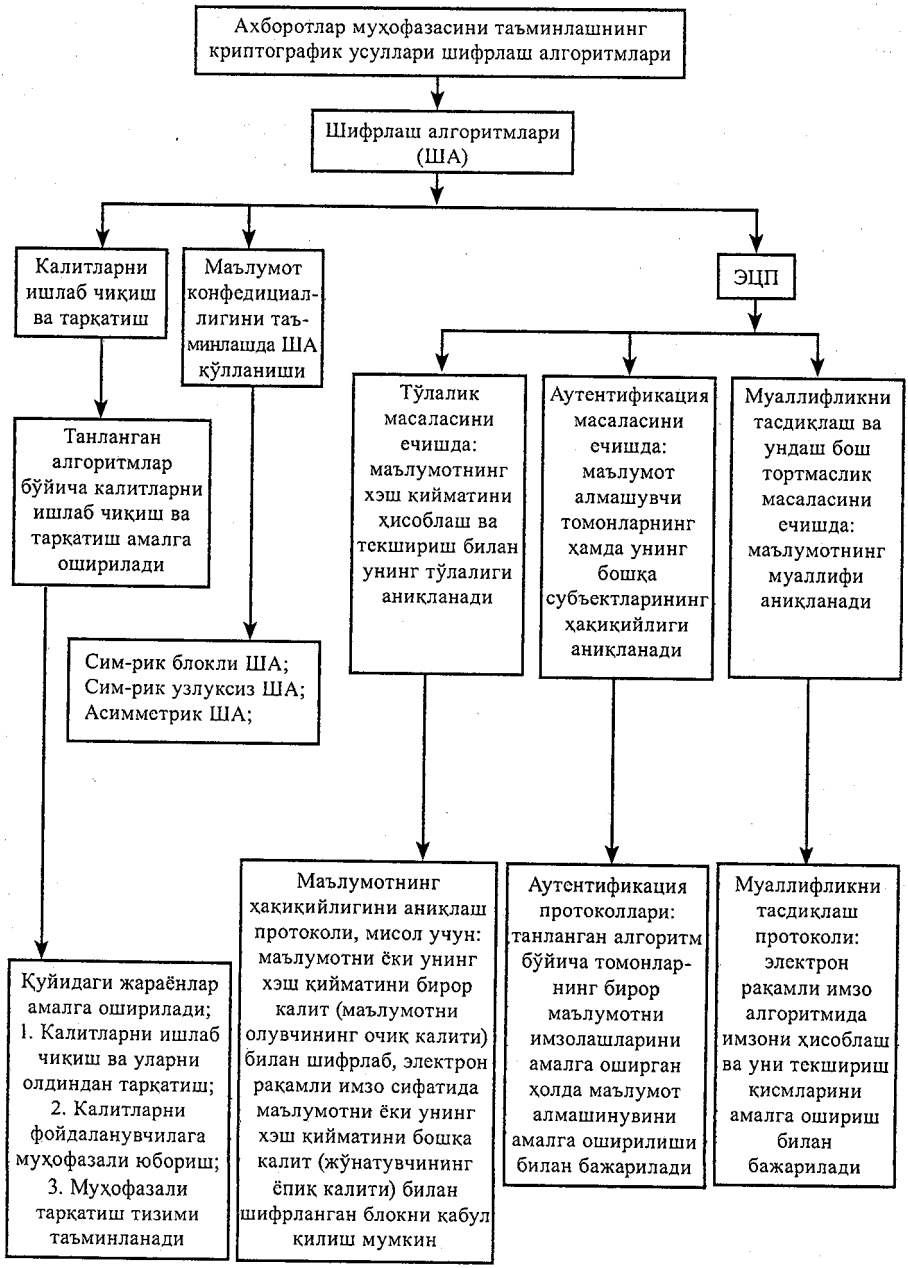
Криптобардошли калитлар ишлаб чиқариш мақсади калит блокини ташкил этувчи элементлар (битлар ёки байтларнинг) тасодифийлигини таъминлашдан иборат.

Ахборотнинг муҳофазасини таъминлашнинг санаб ўтилган масалалари ва уларнинг криптографик ечимлари ҳамда очиқ ва шифрланган маълумотларни таҳлил қилиш усулларининг боғлиқликларини қуйидагича ифодалаш мумкин:

Криптография ва криптотаҳлил масалаларининг структуравий схемаси



Ахборот муҳофазасини криптографик усуллар билан таъминлаш воситалари асосини шифрлаш алгоритмлари ташкил этади. Бу фикр қуйидаги схемада ўз аксини топган:



Шифрлаш алгоритмлари махфий параметрларга асосланган – симметрик калитли ва қўлланиш протоколи билан аниқланувчи – махфий ҳамда очиқ параметрларга асосланган – асимметрик шифрлаш криптоалгоритмларидан иборат.

Ахборот-коммуникация тармоқларида ахборот муҳофазасини таъминлашнинг криптографик воситалари: криптографик алгоритмларнинг дастурий таъминоти ва аппарат-дастурий қурилмаларидан иборат бўлади. Нисбатан содда, аммо криптобардошли бўлган алгоритмларнинг аппарат-техник қурилмалари самарали қўлланилади.

Шифрлаш алгоритмларининг асосий криптографик хусусиятларга эга бўлган математик моделларда ифодаланувчи акслантиришлар билан аниқланади.

1-боб бўйича хулосалар

Ушбу бобда:

1. Криптологиянинг асосий тушунчалари: *алифбо, очиқ матн (очиқ маълумот), шифрлаш, шифрматн, шифрлаш ва дешифрлаш калитлари, криптотизим, симметрик ва асимметрик шифрлаш алгоритмлари, хэш-функция, электрон рақамли имзо, криптобардошлилик* ва шу кабилар ҳақида сўз юритилди.

2. Криптологиянинг фан сифатида шаклланиши:

– 1949 йилгача – фақат интуиция ва «ишончга» асосланган – исботсиз ва илмий асосланмаган криптология даврини;

– 1949 йилдаги К.Э. Шенноннинг «Махфий тизимларда алоқа назарияси», деб номланган илмий мақоласи чоп этилгандан сўнг – илмий асосланган *махфий калитли криптография* даврини;

– 1976 йилда У. Диффи ва М.Е. Хеллманнинг «Криптографияда янги йўналиш», деб номланган мақоласининг эълон қилиниши – илмий асосланган *очиқ калитли критография* даврини ўз ичига олиши баён қилинди.

3. Криптотизимлар ва уларнинг калитларига қўйиладиган талабларнинг моҳиятлари илмий асосланган ҳолда ёритилди.

Ахборот хавфсизлигини таъминлашнинг асосий масалалари кетирилиб, улар ечимларининг криптографик воситаларини таркиби баён этилди.

II БОБ
КРИПТОЛОГИЯДА ҚЎЛЛАНИЛАДИГАН БАЪЗИ МАТЕМАТИК ТУШУНЧА ВА ТАСДИҚЛАР

Илмий тадқиқ қилинаётган объектлар математик моделларининг сифати даражаси (адекватлиги) улар билан боғлиқ бўлган жараёнларни қанчалик тўлиқ ва аниқ ифодаланиши билан белгиланади.

Математик модел бошланғич фикр ва мулохазалар асосида ўтказилган тажрибалар натижаларини солиштириш ҳамда тадқиқ қилинаётган объект хусусиятларини белгиловчи параметрларнинг табиий боғлиқлиги қонуниятларини ифодаловчи тенглик, тенгсизлик ва тегишлилик муносабатлари билан аниқланади. Криптология бирор чекли сондаги алифбо белгиларининг кетма-кетлиги орқали ифодаланган маълумотни ва унинг ўзгаришлари (акслантирилишлари) билан боғлиқ жараёнларни тадқиқ қилади. Криптографик акслантиришлар математиканинг: тўпламлар ва функциялар назарияси, алгебра, дискрет математика, сонлар назарияси, эҳтимоллар назарияси, ҳақиқий ва комплекс ўзгарувчи функциялар назаряси, мураккаблик назарияси, ахборотлар назарияси ва шу каби бўлимларига тегишли бўлган математик моделлардан иборат. Криптографик моделларнинг математик асослари билан чукурроқ танишишни истаганлар адабиётлар рўйхатида келтирилган [2–5, 7–13, 15–17, 19–20] манбалардан фойдаланишлари мумкин. Ушбу бобда криптографик алгоритмларнинг математик моделлари ифодаларини ёритишда фойдаланиладиган тушунчалар, таърифлар, тасдиқлар ва шу каби баъзи маълумотлар келтирилган.

§2.1. Тўпламнинг таърифи, элементар хоссалари ва улар устидаги амаллар

Тўплам математиканинг кўплаб соҳаларида бошланғич-фундаментал тушунча ҳисобланиб, бирор белгиси, ҳусусияти ёки хоссалари каби умумийлик асосидаги бирлашмадан иборат ва бирлашмани ташкил этувчилар тўпламнинг элементлари деб юритилади.

Ушбу $x \in X$ ифода x – элементнинг X – тўпламга тегишли эканлигини билдиради, акс ҳолда $x \notin X$ ифода билан белгиланади.

Агар ҳар иккала тўплам ҳам бир хил элементлардан ташкил топган бўлса, берилган X ва Y тўпламлар тенг дейилади, акс ҳолда тенг эмас дейилади.

Мисол учун: $X=\{0;0;0\}=\{0;0;0;0\}=Y$, $X=\{0;0;0;0\}\neq\{0;0;0\}=Y$, яъни тўпламлар элементлари сони тенг эмас.

Элементлари сони чекли (чексиз) бўлган тўплам чекли (чексиз) тўплам дейилади.

Берилган X ва Y тўпламлар ўзаро бир қийматли (биектив) φ – мосликка эга дейилади, агарда ҳар бир олинган $x\in X$ элементга битта $\varphi(x)\in Y$ элемент мос келиб, ҳар бир олинган $y\in Y$ элементга $\varphi(x)=y$ тенгликни қаноатлантирувчи $x\in X$ элемент мос келса. Бундай биектив мослик $\varphi: X\leftrightarrow Y$ кўринишда ифодаланади. Умуман олганда «φ – акслантириш X – тўплам элементларини Y – тўплам элементларига акслантиради» ибораси: $\varphi: X\to Y$ кўринишда ифодаланади.

Агар берилган X – чексиз тўплам, агар унинг элементларини номерлаб чиқиш мумкин бўлса, яъни X – тўплам билан N – натурал сонлар тўплами ўзаро бир қийматли мосликка эга бўлса, у саноқли дейилади. Бошқа чексиз тўпламлар саноқсиз дейилади. Мисол учун, исбот қилиш мумкинки, барча рационал сонлар тўплами саноқли, $[0; 1]$ – кесмадаги барча ҳақиқий сонлар тўплами эса саноқсиздир.

Берилган чекли тўплам элементлари сони унинг қувватини аниқлайди. Элементлари сони n та бўлган X – тўпламнинг қуввати n га тенг бўлиб, $|X|=n$, деб ифодаланади. Саноқсиз тўпламлар «континиум» қувватга эга деб ҳам юритилади.

Тўпламни аниқлаш унинг элементларини бевосита кўрсатиш билан амалга оширилади. Бундан ташқари, тўпламни, унинг элементлари хусусиятини сўзлар орқали ёритиш:

$M=\{i\in N: i$ – *натурал сон бўлиб, 2 га қолдиксиз бўлинади*$\}$

ёки формулалар билан ифодалаш (рекурсив усул):

$$M=\{i\in N: i=2k;\ k=1,\ 2,...\}$$

орқали аниқлаш мумкин.

Агарда Y – тўпламнинг ҳар бир элементи X – тўпламнинг ҳам элементи бўлса, у ҳолда Y – тўплам X – тўпламга қисм тўплам бўлади ва $Y\subseteq X$ кўринишда ифодаланади.

Агарда $Y\subseteq X$ бўлиб, $Y\neq X$ бўлса, у ҳолда $Y\subset X$ кўринишда ифодаланади ва Y – тўплам X – тўпламнинг хос қисм тўплами дейилади.

Агар $Y\subseteq X$ ва $X\subseteq Y$ бўлса, у ҳолда $Y=X$ бўлади.

Бирорта ҳам элементга эга бўлмаган тўплам бўш тўплам дейилади ва $\varnothing$ белги билан ифодаланади. Бўш тўплам $\varnothing$ ихтиёрий тўпламга қисм тўплам бўлади ва унинг қуввати нолга тенг, яъни $|\varnothing|=0$.

Ҳар қандай X ва Y – тўпламлар жуфтлиги учун қуйидаги амаллар аниқланган:

1) йиғинди $X \cup Y = \{x : x \in X$ ёки $x \in Y\}$;

2) кесишма (кўпайтма) $X \cap Y = \{x : x \in X$ ва $x \in Y\}$;

3) айирма $X \setminus Y = \{x : x \in X$ ва $x \notin Y\}$.

Бу амаллар қуйидаги хоссаларга эга:

1) коммутативлик: $X \cup Y = Y \cup X$ ва $X \cap Y = Y \cap X$;

2) ассоциативлик: $(X \cup Y) \cup Z = X \cup (Y \cup Z)$ ва $(X \cap Y) \cap Z = X \cap (Y \cap Z)$;

3) дистрибутивлик: $X \cap (Y \cap Z) = (X \cap Y) \cap (X \cap Z)$ ва $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$;

4) $(X / Y) \cup (X \cap Y) = X$.

Агар $X \subseteq U$ бўлса, у ҳолда X – тўпламнинг U – тўпламга нисбатан тўлдирувчиси деб:

$$\overline{X} = U \setminus X = \{x \in U : x \notin X \subseteq U\}$$

тўпламга айтилади.

Қуйидаги муносабатлар ўринли:

$$\overline{X \cap Y} = \overline{X} \cup \overline{Y} \text{ и } \overline{X \cap Y} = \overline{X} \cap \overline{Y}.$$

Берилган $X_1, X_2, ..., X_m$ – тўпламларнинг Декарт кўпайтмаси деб, ушбу $X = X_1 \times X_2 \times ... \times X_m = \{(x_1, x_2, ..., x_m\} = x \in X : x_i \in X_i\}$ – тўпламга айтилади. Математик индукция усулидан фойдаланиб $X_1, X_2, ..., X_m$ – тўпламлар Декарт кўпайтмасини ташкил этувчи тўпламнинг қуввати ушбу

$$|X_1 \times X_2 \times ... \times X_m| = \prod_{i=1}^{m} |X_i|$$

тенглик билан аниқланишини исбот қилиш мумкин, яъни берилган тўпламлар Декарт кўпайтмасини ташкил этувчи тўпламнинг қуввати кўпайтувчилар қувватларининг кўпайтмасидан иборат.

Берилган X – тўплам $\leq$ – муносабат билан тартибланган (чизиқли тартибланган, тўла тартибланган) дейилади, агарда $\forall a, b, c \in X$ – элементлар учун қуйидаги ҳоссалар бажарилса:

1) рефлексивлик $a \leq a$;

2) антисимметриклик – агар $a \leq b$ ва $b \leq a$ бўлса, у ҳолда $a = b$;

3) транзитивлик – агар $a \leq b$ ва $b \leq c$ бўлса, у ҳолда $a \leq c$;

4) чизиқлилик – ёки $a \leq b$ ёки $b \leq a$.

Агар $\forall a, b, c \in X$ – элементлар учун (1) – (3) хоссалар бажарилса, берилган X – тўплам қисман тартибланган тўплам (қ.т.т.) дейилади.

X – қисман тартибланган тўпламнинг *диаграммаси* (*Хаас диаграммаси*) деб, шу тўплам элементлари жуфтликларининг $(a, b) \in X$ ёй (йўналтирилган кесма) билан боғланган ифодасини текисликдаги тасвирига айтилади. Графлар таърифида, X – қисман тартибланган тўплам – бу йўналишга эга бўлган граф бўлиб, унинг учлари X – тўпламдан иборат эканлиги, (a, b) – жуфтлик фақат ва фақат ушбу $a \le b$ ва $a \ne b$ – шартлар билан биргаликда a ва b элементлардан фарқли бўлган $a \le c \le b$ шартни қаноатлантирувчи $c \in X$ элемент мавжуд бўлмагандагина ёй ташкил этиши таъкидланади.

Y – тўплам берилган X – қисман тартибланган тўпламнинг қисм тўплами бўлиб, $a \in X$ бўлсин. У ҳолда $a \in X$ бўлган элемент Y – қисм тўпламнинг юқори (қуйи) чегараси дейилади, агарда барча $b \in Y$ элементлар учун $b \le a$ $(a \le b)$ шарт бажарилса. Y – тўпламнинг юқори чегараси a унинг аниқ юқори (қуйи) чегараси дейилади, агарда Y – тўпламнинг барча c – юқори (қуйи) чегаралари учун $a \le c (c \le a)$ шарт бажарилса, $a = \sup Y (a = \inf Y)$ деб белгиланади.

Агар $\forall a, b, c \in X$ элементлар учун $\sup(a, b) \in X$ ҳамда $\inf(a, b) \in X$ бўлса қ.т.т. *X панжара* дейилади.

Тўпламларнинг хоссалари билан боғлиқ бўлган криптология масаларини таҳлил қилишда қўлланиладиган тушунча ва тасдиқларни тўпламлар назариясининг амалий тадбиқлари ёритилган ўқув қўлланмаларидан топиш мумкин.

§2.2. Тўпламларни акслантириш

Акслантиришлар берилган тўпламлар устида амаллар бажариш билан уларнинг элементлари орасида мослик ўрнатиш жараёнини ифодалайди. Акслантиришларнинг хоссаларини таҳлил қилиш билан боғлиқ бўлган айрим тушунча ва таърифларни келтирамиз.

Берилган φ – акслантириш (функция) X – тўпламни Y – тўпламга *бир қийматли* акслантиради дейилади (ва $\varphi : X \to Y$ кўринишда белгиланади), агарда ҳар бир $x \in X$ элементга фақат битта $y = \varphi(x) \in Y$ элемент мос қўйилса. Бу ерда X – тўплам φ – акслантиришнинг *аниқланиш соҳаси*, X – тўплам эса *қийматлар соҳаси*, y – элемент x – элементнинг *акси*, x – элемент y – элементнинг *асли* дейилади.

Агарда берилган φ ва ψ акслантиришларнинг аниқланиш ва қийматлар соҳалари тўла устма-уст тушиб, $\forall x \in X$ элемент учун $\varphi(x) = \psi(x)$ тенглик бажарилса, бундай акслантиришлар *тенг* дейилади.

Ушбу $\varphi : X \to Y$ акслантириш берилган бўлсин, у ҳолда $\psi : X' \to Y$ акслантириш φ акслантиришнинг $X' \subseteq X$ тўпламдаги *изи* дейилади, агарда $\forall x \in X'$ учун $\varphi(x) = \psi(x)$ тенглик ўринли бўлса.

Берилган $\varphi: X \to Y$ акслантириш учун:

1) ихтиёрий $x \in X$ учун $\varphi(x) = y \in Y$ элемент мавжуд бўлиб, баъзи $y \in Y$ элементлар учун $\varphi^{-1}(y) = x$ тенгликни қаноатлантирувчи $x \in X$ элементлар мавжуд бўлмаса, бундай акслантириш *сюръектив* ёки устига акслантириш дейилади;

2) $x_1 \neq x_2$ бўлган $\forall x_1, x_2 \in X$ элементлар учун $y_1 = \varphi(x_1) \neq \varphi(x_2) = y_2$ шу каби бўлса, бундай акслантириш *инъектив* акслантириш дейилади.

3) бир пайтнинг ўзида ҳам *сюръективлик* ҳам *инъективлик* шартлари бажарилса, бундай акслантириш *биектив* ёки *ўзаро бир қийматли* акслантириш дейилади.

Ушбу $\varphi: X \to Y$ ва $\varphi: Y \to Z$ акслантиришларнинг *кўпайтмаси* (*композицияси*, *суперпозицияси*) деб, $\sigma(x) = \psi(\varphi(x))$ тенгликни қаноатлантирувчи $\sigma: X \to Y$ акслантиришга айтилади, ҳамда $\sigma = \psi \cdot \varphi$ кўринишда ифодаланади.

$\varphi: X \to Y$ акслантириш X – тўпламни *ўзини-ўзига* акслантириш дейилади.

$\forall x \in X'$ элемент учун $I(x) = x$ тенгликни қаноатлантирувчи X – тўпламни ўзини-ўзига акслантирувчи I – акслантириш *бирлик* (*айнан*) акслантириш дейилади.

Агар $\psi \cdot \varphi = \varphi \cdot \psi = I$ шарт бажарилса, берилган $\varphi: X \to Y$ ва $\psi: Y \to X$ – акслантиришлар *ўзаро тескари* акслантиришлар дейилади, ҳамда $\psi^{-1} = \varphi$, $\varphi^{-1} = \psi$ деб ёзилади.

Тескариси мавжуд бўлмаган акслантиришлар *бир томонлама* акслантиришлар дейилади

Бирор $x \in X$ элемент учун $\varphi(x) = x$ тенглик бажарилса, бу элемент φ акслантиришнинг қўзғалмас элементи дейилади.

Элементлари сони n та бўлган X – тўпламни ўзини-ўзига биектив акслантирувчи φ – акслантириш X – тўпламда n – *даражали ўрнига қўйиш* дейилади. Агарда тўплам $X = \{x_1, ..., x_n\}$ бўлса, у ҳолда φ – акслантириш қуйидагича:

$$\varphi = \begin{pmatrix} x_1, \ldots, x_n \\ \varphi(x_1), \ldots, \varphi(x_n) \end{pmatrix} = \begin{pmatrix} x_1, \ldots, x_n \\ x_1, \ldots, x_n \end{pmatrix},$$

ёзилади, бу ерда $(i_1, ..., i_n)$ индекслар (1, 2, ... n) – сонларнинг ўрин алмаштиришдан иборат.

Агарда ўрнига қўйиш акслантириши φ ушбу $\varphi^{-1} = \psi$ тенгликни қаноатлантирса, у ҳолда бу акслантириш *инволюция* дейилади

X – тўпламни ўзини-ўзига акслантирувчи φ – ўрнига қўйиш акслантириши $x_i, x_j \in X$ элементлар учун $\varphi(x_i) = x_j$ ва $\varphi(x_j) = x_i$ тенгликлар-

ни қаноатлантириб, X – тўпламнинг бошқа элементлари бу акслантиришга нисбатан қўзғалмас элементлар бўлса, бундай φ – акслантириш x_i ва x_j элементларнинг X – тўпламдаги *транспозицияси* дейилади.

§2.3. Графлар

Ушбу параграфда графларга оид баъзи тушунча ва таърифлар келтирилади.

Бирор $X=\{x_1, x_2, ...\}$ тўплам берилган бўлсин. Ушбу $X\times X$ ёки X^2 орқали бирор тартиблаш муносабати φ билан тартибланган барча (x_i, x_j) – жуфтликлар тўпламини белгилаймиз.

Агар X^2 тўпламнинг ихтиёрий $R_\varphi \subseteq X^2$ қисм тўпламида: $\forall(x_i, x_j)\in R_\varphi$ элементлар жуфтлиги φ муносабатни $x_i\varphi x_j$ қаноатлантириб, бу муносабат натижаси $x_i\varphi x_j=(x_i, x_j)\in R_\varphi$ бўлса $X=\{x_1, x_2, ...\}$ – тўплам элементлари устида аниқланган тартиблаш муносабати φ бинар муносабат дейилади,

Бинар муносабат φ аниқланган тўплам $X=\{x_1, x_2, ...\}$ граф дейилади ва $G=(X, \varphi)$ деб белгиланади. Бунда, $X=\{x_1, x_2, ...\}$ тўплам элементлари G – графнинг *уч(чўққи)лари*, $(x_i, x_j)=x_i\varphi x_j\in R_\varphi$ – жуфтликлар G – графнинг *қирралари* дейилади.

Агарда қирра $(x_i, x_j)\in R_\varphi$ учун $x_i\neq x_j$ тенгсизлик ўринли бўлса, у ҳолда x_i ва x_j *ёндош учлар* ёки қирра (x_i, x_i) x_i ва x_j учларни туташтиради дейилади.

Агарда $i=j$ бўлса, у ҳолда қирра (x_i, x_i) *сиртмоқ* дейилади.

Қирра (x_i, x_i) x_i ва x_j учларга *инцидент* (ёки x_i ва x_j учларга туташ) дейилади. Битта учга инцидиент (туташ) бўлган (ёки битта учда туташувчи) иккита қирра *ёндош қирралар* дейилади. Ҳеч бир қиррага инцидиент (туташ) бўлмаган (ёки ҳеч бир қиррага эга бўлмаган) уч *ажралган* (*изоляцияланган*) дейилади. Битта қиррага инцидиент (туташ) бўлган уч ва шу қирра *охирги* (*тугал*) дейилади.

Бир хил учлар жуфтлигига мос қўйилган қирралар *каррали* ёки *паралел* дейилади.

Ихтиёрий иккита ҳар хил учлари ёндош бўлиб, каррали қирраларсиз (ва сиртмоқсиз) граф *тўла* дейилади.

Агарда ҳар бир элементлар жуфтлиги $(x_i, x_j)\in G$ тартиблашган бўлса, у ҳолда G *йўналтирилган (ориентирланган) граф* ёки *орграф*, унинг қирралари *ёй* деб аталади. Акси ҳолда *йўналтирилмаган (ориентирилмаган)* граф дейилади.

Агарда графнинг G *учлари* (*қирралари, ёйлари*) бирор тўпламнинг элементлари билан белгиланган бўлса, у *белгиланган учли* (*қиррали, ёйли*) дейилади.

Граф $G=(X, \varphi)$ икки қисмли дейилади, агарда $X = X_1 \cup X_2$ бўлиб, $(x_i, x_j) \in R$ жуфтликни ташкил этувчи элементлар x_i ва x_j ҳар хил қисмларга тегишли бўлса.

Агарда X ва R_φ тўпламлар чекли бўлса, у ҳолда граф G *чекли* дейилади.

Йўналтирилмаган (ориентирилмаган) графнинг i учига инцидиент (туташ) бўлган қирралар сони i *учнинг даражаси* дейилади, бунда сиртмоқ икки марта ҳисобга олинади.

Йўналтирилган (ориентирланган) графнинг ҳар бир i учи унга кирувчи қирраларнинг сони p_i чиқувчи қирраларнинг сони q_i билан ифодаланучи (p_i, q_i) сонлар жуфтлиги билан характерланади. Бунда, p_i ва q_i сонлари мос равишда i учга *киришнинг ярим даражаси* ва *чиқишнинг ярим даражаси* дейилади.

§2.4. Мураккаблик назарияси

Мураккаблик назарияси криптографик алгоритмларнинг ҳисоблаш мураккабликларини таҳлил қилиш услубини беради. Ҳар хил криптографик алгоритмларнинг ҳисоблаш мураккабликларини солиштириб, уларнинг ишончлилик – бардошлилик даражаси аниқланади.

Алгоритмнинг мураккаблиги. Алгоритмнинг мураккаблиги, шу алгоритмни тўла амалга ошириш учун бажарилиши назарда тутилган барча амаллар сони билан аниқланади. Алгоритмнинг ҳисоблаш мураккаблиги одатда иккита параметр – алгоритмда кўрсатилган амалларни бажаришга сарфланадиган *вақт билан аниқланадиган мураккаблик* T ва ҳисоблаш қурилмасида алгоритм параметлари устида амаллар бажаришда керак бўладиган регистрлар сони билан аниқланадиган – *ҳисоблаш қурилмаси хотираси ҳажми билан боғлиқ бўлган мураккаблик* S билан аниқланади.

Бу T ва S параметрлар алгоритм хусусиятларидан келиб чиқиб бошланғич қийматларнинг n ўлчамига боғлиқ ҳолда яъни $T=f(n)$ ва $S\varphi(n)=$функциялар билан аниқланади.

Алгоритмнинг ҳисоблаш мураккаблиги одатда ҳисоблаш мураккаблиги қийматининг тартибини кўрсатувчи «O» деб аталувчи белги билан ифодаланади ҳамда бу белги n – параметр қийматининг ортиши билан мураккаблик функцияси ифодаси ичида қиймати энг тез ўсадиган ҳадни ифодалаб, бошқа ҳадларни ҳисобга олмайди. Масалан, алгоритмнинг вақт билан аниқланадиган мураккаблиги

$T=f(n)=5n^2+6n+11$ бўлса, у ҳолда унинг n^2 тартибли ҳисоблаш мураккаблиги $O(n^2)$ кўринишда ифодаланади.

Ҳисоблаш мураккаблиги баҳолари бошланғич қийматларни, алгоритмнинг хусусиятларидан келиб чиққан ҳолда, алгоритмни амалга ошириш учун сарфланадиган вақт ва ҳисоблаш курилмаси хотирасига қўйиладиган талабларни яққол намоён этади. Масалан, $T=O(n)$ бўлса, бошланғич қиймат ўлчамининг икки марта ўсиши вақтнинг ҳам икки марта ўсишига олиб келади; агарда $T=O(2^n)$ бўлса, бошланғич қиймат ўлчамига битта битнинг қўшилиши алгоритмни амалга ошириш учун сарфланадиган вақтни икки баравар ортиришни билдиради.

Алгоритмлар вақт ва ҳисоблаш мураккабликларига кўра қуйидагича классификацияланади (синфларга ажратилади), таснифланади:

1. Алгоритм *доимий* дейилади, агарда унинг мураккаблик қиймати бошланғич қиймат ўлчамига боғлиқ бўлмаса, яъни $O(1)$;

2. Алгоритм *чизиқли* дейилади, агарда унинг мураккаблики қийматининг тартиби $O(n)$ бўлса;

3. Алгоритм *полиномиал* дейилади, агарда унинг мураккаблики қийматининг тартиби $O(n^m)$ (бу ерда $m>1$) бўлса;

4. Алгоритм *экспоненциал* дейилади, агарда унинг мураккаблики қийматининг тартиби $O(t^{f(n)})$ (бу ерда $const=t>1$ ва $f(n)$ – бошланғич қиймат ўлчами n га нисбатан полиномиал функция) бўлса;

5. Мураккаблики қийматининг тартиби $O(t^{f(n)})$ бўлган *экспоненциал* алгоритмлар тўпламига қисм тўплам бўладиган алгоритмлар *суперполиномиал* дейилади, агарда $f(n)$ – полиномиал функция t ўзгармасга нисбатан тезроқ, лекин чизиқли функцияга нисбатан секинроқ ўсса, мисол учун: $O(t^{\sqrt{n}})$, $1<t<\sqrt{n}$ бўлса.

Шу ерда таъкидлаш жоизки, криптоалгоритмлар натижасига кўра унинг номаълум параметрларини топишнинг мавжуд алгоритмлари суперполиномиал мураккабликка эга бўлиб, уларнинг полиномиал мураккабликка эга бўлган алгоритмларини топиш мумкин эмаслиги исбот қилинмаган. Яъни бирор алгоритмнинг номаълум параметрини полиномиал мураккабликка эга бўлган алгоритмларини топиш мумкинлиги унинг криптобардошсиз бўлиб қолганлигигни билдиради.

Масаланинг мураккаблиги. Бирор масаланинг ечилишини алгоритмининг мураккаблигидан ташқари масалани ўзининг мураккаблиги тушунчаси ҳам мавжуд. Масаланинг мураккаблиги назарияси энг ечилиши мураккаб бўлган масалани **Тьюринг машинаси** деб аталувчи – **назарий компьютерда** ечиш учун сарфланадиган минимал вақт ва хотира ҳажмини баҳолаш масалалари билан

шуғулланади. Тьюринг машинаси – ўқиш ва ёзиш учун чексиз хотирага эга бўлган чекли сондаги амалларни бажарувчи ҳисоблаш қурилмасидан иборат.

Полиномиал мураккабликка эга бўлган алгоритмлар билан ечиладиган масалалар, *ечиш мумкин бўлган* масалалар дейилади, яъни бошланғич киритиладиган қийматларнинг бирор чекли n-ўлчамида қониқарли вақт бирлиги ичида ечилиши мумкин полинамиал мураккабликка эга бўлган масалалар. Полиномиал вақт бирлиги ичида ечилмайдиган масалалар *қийин ечиладиган* ёки *қийин* масалалар дейилади, яъни бошланғич киритиладиган қийматларнинг бирор етарли кичик чекли n-ўлчамидан бошлаб ечиш учун бажарилиши керак бўлган амаллар сонининг етарли даражада тез ўсиб кетишига олиб келиб, бу амалларнинг барчасини амалга ошириш имконини бермайдиган масалалар. Бошланғич киритиладиган қийматларнинг нисбатан етарли кичик чекли n – ўлчамида супер полиномиал мураккабликка эга бўлган алгоритмлар билан ечиладиган масалалар *ҳисобланиши қийин* бўлган масалалар дейилади.

Ечиш алгоритмлари яратилмаган (ёки қандай яратилиш асослари замонавий илм-фан ютуқларига мантиқан маълум бўлмаган) масалалар – *ечилмайдиган* масалалар дейилади.

Иккилик соноқ тизимининг сўзлари деб аталувчи, иккилик саноқ тизимининг алифбо белгиларидан {0; 1} дан иборат барча:

0; 1; 00; 01; 10; 11; 000; 001; … ; 111; …; 00…0; 00…1; …; 11…1; …

чекли сондаги 0 ва 1 *белгилар кетма-кетликлар блокларидан (векторларидан)* тузилган тўпламни Σ деб белгилаймиз. Барча ўлчами n га тенг бўлган иккилик саноқ тизимининг сўзлари тўпламини Σ^n деб белгилаймиз. Мураккаблик назариясида Σ – тўпламга қисм бўлган тўпламлар $L \in \Sigma$ – тиллар дейилади деб қабул қилинган.

Агар Тьюринг машинаси M, да у ихтиёрий чекли n – ўлчамли бошланғич кириш қийматига (сўзига) боғлиқ бўлган $p(n)$ – кўпҳаднинг (максимал) қийматидан кўп бўлмаган амаллар бажаргандан сўнг тўхтаса, у полиномиал вақт бирлиги ичида ишлайди (ёки полиномиал) дейилади.

M – Тьюринг машинаси L – тилни тушунади (қабул қилади) дейилади, агарда у L – тилга тегишли бўлган ихтиёрий кириш сўзида, яъни $\forall x \in L$ бўлганда, амаллар бажариб, яна қабул қилиш ҳолатида, ҳамда, $\forall x \notin L$ бўлганда амаллар бажариб рад этиш ҳолатида тўхтаса.

Полиномиал вақт бирлиги ичида ишлайдиган Тьюринг машинаси M қабул қиладиган барча тиллар синфи P – синф деб белгиланади.

Агарда функция f учун полиномиал Тьюринг машинаси мавжуд бўлиб, бошланғич қиймат – кириш $x \in \Sigma$ сўзида амаллар бажариб, тўхтаганда $f(x)$ қийматни берса, у $f: \Sigma \to \Sigma$ полиномиал вақт бирлиги ичида ҳисобланади дейилади.

Агарда полиномиал вақт бирлиги ичида ҳисобланадиган $P(x, y)$: $: \Sigma \times \Sigma \to \{0,1\}$ – функция (предикат) мавжуд бўлиб, бошланғич кириш қийматлари ўлчовига нисбатан аниқланувчи мураккаблик полиноми $p \in L = \{x \mid \exists\, y\ P(x, y)\ \&\ |y| \le p(|x|)\}$ бўлса, L – тил NP – тўлиқ синфга тегишли бўлади. Яъни L –тил NP –тўлиқ синфга тегишли бўлади, агарда ихтиёрий n – ўлчами $x \in L$ сўз учун унга мос $p(|x|) = p(n)$ – полиномиал узунликка эга бўлган y сатрни кўрсатиш мумкин бўлиб, кўрсатилган сатрни тўғри ёки нотўғрилигини $P(x, y)$ – предикат орқали аниқланади.

Юқоридаги фикр ва мулоҳазалардан $P \subseteq NP$ эканлиги келиб чиқади. *Бу тегишлилик муносабати қатъий, яъни*: $P \subset NP$ *ва* $P \neq NP$ *эканлиги тўғрисида ҳозирги кунда бирор исбот қилинган далил мавжуд эмас.*

NP – тўлиқ синфдан максимал полиномиал мураккабликка эга бўлган тилларнинг қисм синфи ажратилган, яъни ихтиёрий $L \in NP$ – тўлиқ – тил полиномиал вақт бирлиги ичида тушунилиши (қабул қилиниши) учун $P = NP$ бўлиши зарур ва етарли.

Юқорида киритилган *Тьюринг машинаси* тушунчасидан ташқари *Тьюрингнинг эҳтимоллик машинаси* тушунчаси ҳам мавжуд. Бу тушунчаларнинг фарқи қуйидагича. Тьюринг машинасининг кейинги (янги) ҳолати унинг бундан олдинги ҳолати билан тўлиқ аниқланади. Тьюрингнинг эҳтимоллик машинасининг кейинги (янги) ҳолати унинг бундан олдинги ҳолати ва яна 0 ҳамда 1 қийматларни ½ эҳтимоллик билан қабул қилувчи тасодифий миқдорнинг қиймати билан биргаликда аниқланади. Яъни Тьюрингнинг эҳтимоллик машинаси унинг ҳолатини ифодаловчи қўшимча тасодифий миқдорнинг 0 ва 1 қийматлари чексиз кетма – кетлиги сатрининг ҳолатига ҳам боғлиқ.

Табиий равишда савол туғилади: ушбу $P \neq NP$ тенгсизлик бардошли криптографик тизимлар мавжудлигини зарурий ва етарлилик шартини ифодалайдими?

Ҳақиқатдан ҳам бу шартнинг зарурийлиги – бардошли криптотизимлар учун $P \neq NP$ шартининг бажарилишига бевосита ишонч ҳосил қилиш мумкинлигидадир. Юқорида кўрилган мисолга қайтган ҳолда, ушбу

$L = \{(k_1, d, i) \mid \exists$ маълумот m: $d = E_{k_1}(m)$ и $m_i = 1\}$

тилни аниқлаймиз. Яъни тўплам $L \subset \Sigma^n$ бирор n – ўлчамли барча $m=(m_1, m_2, ..., m_i, m_n) \in \Sigma^n$ сўзлардан, i – бити 1 га тенг $m_i=1$ бўлганлари бўлиб (уларнинг сони 2^{n-1} та), уларни k_1 – калит билан E – бир томонламалик хусусиятига эга бўлган алгоритмдан фойдаланган холда шифрланганда $d=E_{k_1}(m)$ тенгликни қаноатлантиради. Ушбу k_1 ва d параметрларни ҳамда E – алгоритмни илган ҳолда $d=E_{k_1}(m)$ ва $m_i=1$ тенгликларни қаноатлантирувчи барча $m=(m_1, m_2, ..., m_i, m_n) \in L \subset \Sigma^n$ топиш экспоненциал мураккабликка эга. Бундай аниқланган тил $L \in NP$ бўлиб, экспоненциал вақт бирлиги ичида бу тилда шундай m матнларни кўрсатиш мумкинки, бу матнлар учун $d=E_{k_1}(m)$ ва унинг (m нинг) i – бити 1 га тенг, яъни $m_i=1$. Агар шундай бўлса, кириш сўзи (k_1, d, i) қабул қилинади, акси ҳолда рад этилади.

Агарда $P=NP$ деб фараз қилинса, L – тилни тушинувчи (қабул қилувчи) полиномиал мураккабликка эга бўлган E – алгоритм мавжуд бўлиб, k_1 ва d параметрларни билган ҳолда, бу алгоритмдан фойдаланган ҳолда $d=E_{k_1}(m)$ ва $m_i=1$ шартларни қаноатлантирувчи $m=(m_1, m_2, ..., m_i, m_n) \in L \subset \Sigma^n$ очиқ матнларни ҳисоблаш мумкин. Бундай хусусиятга эга бўлган алгоритмлар криптобардошсиз бўлади.

Ушбу $P \neq NP$ тенгсизлик ўринли бўлганда, NP–тўлиқ масала асосида яратилган ҳар қандай алгоритми махфий параметрларини аниқлаш ҳар доим ҳам NP – тўлиқ масала бўладими, яъни экспоненциал мураккабликка эга бўладими? Бундай саволга жавоблар асимметрик криптографик алгоритмларни таҳлил қилиш орқали қидирилган. Ҳамда, NP–тўлиқ масала асосида яратилган ҳар қандай криптоалгоритм махфий параметрларини аниқлаш ҳар доим ҳам NP–тўлиқ масала бўлавермаслигига ишонч ҳосил қилинган. NP–тўлиқ масала фақат унга бошланғич киритиладиган қийматларнинг бирор чекли n – ўлчами бирор қийматдан кичик бўлмагандагина қийин ечиладиган масала бўлишлиги аниқланган. Бундан келиб чиқадики, $P \neq NP$ шартнинг бажарилиши криптобардошлилик учун етарли эмас. Шунинг учун ҳам криптобардошли алгоритмлар асосида бир томонламалик хусусиятига эга бўлган акслантиришлар ётади.

§2.5. Сонлар назарияси

Натурал сонлар тўпламини $N=\{1, 2, 3, ...\}$ ва бутун сонлар тўпламини $Z=\{0, \pm 1, \pm 2, \pm 3, ...\}$ кўринишда белгилаймиз.

Нолдан фарқли бўлган a сони ва $в$ сонлар Z –тўпламга тегишли, яъни

$a, b \in Z$ бўлиб, $a \neq 0$ бўлсин., агарда шундай c сони мавжуд бўлиб, $в = ac$ тенглик бажарилса, у ҳолда, a сони $в$ сонини бўлади дейилади.

Берилган a ва $в$ сонларни бўлувчи бутун сон, уларнинг *умумий бўлувчиси* дейилади. Умумий бўлувчилар ичида энг каттаси *энг катта умумий бўлувчи* (ЭКУБ) дейилади ва $(a, в)$ кўринишда белгиланади. Агарда a ва $в$ сонларнинг энг катта умумий бўлучиси 1, $(a, в) = 1$ бўлса, a ва $в$ сонлар *ўзаро туб* дейилади. Энг катта умумий бўлувчиларни топишга оид тасдиқларни келтирамиз.

2.1-лемма. Агар $в$ сони a сонини бўлса, у ҳолда бу сонларнинг энг катта умумий бўлувчиси $(a, в) = в$, яъни a сонининг умумий бўлувчилари тўплами $в$ сонининг умумий бўлувчилари тўплами билан устма-уст тушади.

2.2–лемма. Агар $a = bq + c$ бўлса, у ҳолда a ва $в$ сонларининг энг катта умумий бўлувчиси $в$ ва c сонларининг энг катта умумий бўлувчиси билан устма-уст тушади, яъни $(a, b) = (b, c)$: a ва $в$ сонларининг умумий бўлувчилари тўплами $в$ ва c сонларининг умумий бўлувчилари тўплами билан устма-уст тушади.

Юқорида келтирилган ламмалардан ЭКУБ топиш – Евклид алгоритми келиб чиқади.

Ҳақиқатан ҳам қуйидаги бўлиш амалларини бажарамиз:

$$\begin{aligned}
&a = bq_1 + r_1, && 0 \le r_1 < b,\\
&b = r_1 q_2 + r_2, && 0 \le r_2 < r_1,\\
&\ldots\ldots\ldots, && \ldots\ldots\ldots,\\
&r_{n-2} = r_{n-1}\, q_n + r_n, && 0 \le r_n < r_{n-1},\\
&r_{n-1} = r_n q_{n+1}. &&
\end{aligned}$$

У ҳолда $(a, b) = (b, r_1) = \ldots = (r_{n-2}, r_{n-1}) = r_n$.

Берилган натурал сон $p > 1$ туб дейилади, агарда бу сон ўзи p ва 1 дан бошқа натурал сонга бўлинмаса. Мисол учун: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ..., туб сонлар, улар саноқли ва чексиз қувватли тўпламни ташкил этади.

Келгусида, барча бутун сонларни *модуль* (*характеристика*) деб аталувчи бирор фиксирланган натурал n сонига бўлганда қоладиган қолдиқлар билан боғлиқ ҳолда қараймиз. Бунда чексиз қувватли (элементлари сони чексиз) бўлган барча бутун сонлар тўпламига, 0 дан n–1 гача бўлган бутун сонларни ўз ичига оладиган чекли, қуввати n га тенг бўлган $\{0; 1; 2; 3; \ldots; n–1\}$ – тўплам мос қўйилади. Бу қуйидагича амалга оширилади: a ва n – натурал сонлар бўлса, a сонини n сонига қолдиқ билан бўлиш», деганда ушбу шартни қаноатлантирувчи

$$a = qn + r, \text{ бу ерда } 0 \le r < n,$$

натурал q ва r сонларини топиш тушунилади. Бу охирги тенгликда қолдиқ деб аталувчи r сони нолга тенг бўлса $r = 0$, натурал a сони n сонига бўлинади ёки n сони a сонининг бўлувчиси дейилади.

Бутун a ва b сонлари *модуль n бўйича таққосланадиган* дейилади, агарда уларни n га бўлганда қоладиган қолдиқлари тенг бўлса, ҳамда,

$$a \equiv b (\text{mod } n)$$

деб ёзилади. Бундан эса a ва b сонлар айирмасининг n га қолдиқсиз бўлиниши келиб чиқади.

Қолдиқни ифодалаш учун ушбу

$$b = a \bmod n$$

тенгликдан фойдаланилади, ҳамда $b = a \bmod n$ тенгликни қаноатлантирувчи b сонини топиш *а сонини модуль n бўйича келтириш* дейилади.

Ихтиёрий бутун b сони учун ушбу

$$M = \{a_0, a_1, \ldots, a_{n-1} \in Z : 0 \le a_k \le n-1;\ k = 0,1, \ldots, n-1\}$$

тўпламга тегишли $a_k \equiv b \pmod n$ муносабатни қаноатлантирувчи сон a_k, $k \in \{0,1,\ldots, n-1\}$ мавжуд бўлса, тўплам M модуль n бўйича *тўлиқ чегирмалар тизими* дейилади. Кўриниб турибдики, тўлиқ чегирмалар тизими

$$M = \{a_0, a_1, \ldots, a_{n-1} \in Z : 0 \le a_k \le n-1;\ k = 0, 1, \ldots, n-1\} = \{0, 1, \ldots, n-1\}.$$

Бирор n модул бўйича қўшиш, айириш ва кўпайтириш амалларига нисбатан қуйидаги коммутативлик, ассоциативлик ва дистрибутивлик муносабатлари ўринли:

$$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n,$$

$$(a - b) \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n,$$

$$(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n,$$

$$(a \cdot (b + c)) \bmod n = (((a \cdot b) \bmod n) + (a \cdot c) \bmod n)) \bmod n.$$

2.1- теорема. Бутун a ва $в$ сонлари ўзаро туб бўлади, қачонки шундай бутун u ва v сонлари топилсаки, улар учун $au + bv = 1$ тенглик ўринли бўлса.

Бу келтирилган теоремани қуйидагича ҳам ифодалаш мумкин: *бутун а ва в сонлари ўзаро туб бўлиши учун, бутун бўлган и ва v*

сонлари топилиб, улар учун $au+bv=1$ *тенгликнинг бажарилиши зарур ва етарли.*

Агарда бутун a ва $в$ сонлари ўзаро туб бўлса, яъни $(a, n)=1$ бўлса, у ҳолда ушбу $a\,a' \equiv 1 (\mathrm{mod}\ n)$ муносабатни қаноатлантирувчи бутун a' сони мавжуд бўлиб, бу a' *сон* a *сонига модул* n *бўйича тескари* дейилади, ҳамда, $a' \equiv a^{-1} (\mathrm{mod}\ n)$ деб белгиланади. Тескари a' элементни a ва n сонларининг чизиқли комбинациясидан иборат бўлган уларнинг ЭКУБ ифодасидан $au+bn=1$ фойдаланган ҳолда, бу тенгликнинг ҳар иккала томонини модул n бўйича келтириш (ҳисоблаш) билан $a' \equiv u$ $(\mathrm{mod}\ n)$ эканлиги топилади.

Қуйида тескари элементни ҳисоблашнинг яна бир усули келтирилади.

Берилган n сони билан ўзаро туб бўлган $(1; n)$ ораликдаги барча элементларнинг сони билан аниқланувчи $\varphi(n)$ функцияга *Эйлер функцияси* дейилади:

$\varphi(n)=|M|$, бу ерда: $|M|$ M – тўпламнинг қуввати, $M=\{m_i \in N: 1 \le m_i \le n;\ (m_i, n)=1\}$.

Агарда $n = p_1^{k_1} \ldots p_1^{k_t}$ бўлиб, $p_1, \ldots, p_t$ – ҳар хил туб сонлар бўлса, у ҳолда Эйлер функциясининг қиймати $\varphi(n) = \prod_{i=1}^{t} (p_j - 1) \cdot p_j^{k_j - 1}$ ифода билан ҳисобланади.

Ферманинг кичик теоремаси деб аталувчи ушбу тасдиқ ўринли, агар n – туб сон бўлса, $a^{n-1} \equiv 1 (\mathrm{mod}\ n)$ ўринли.

Эйлер томонидан олинган, *Ферманинг кичик теоремасининг умумлашгани* деб аталувчи ушбу тасдиқ ўринли, агар n – туб сон бўлса, $a^{\varphi(n)} \equiv 1 (\mathrm{mod}\ n)$ муносабат бажарилади.

Юқоридагилардан келиб чиққан ҳолда $a^{-1} \equiv a^{\varphi(n)-1} (\mathrm{mod}\, n)$ муносабатнинг ўринлигига ишонч ҳосил қилинади.

Агар n – туб сон бўлса, у ҳолда $\varphi(n)=n-1$. Агар $n=pq$ бўлиб, p ва q – туб сонлар бўлса, у ҳолда $\varphi(n)=(n-1)(q-1)$. Бу каби хоссалардан очиқ калитли криптоалгоритмлар яратишда фойдаланилади. Масалан, қандай сон модуль 7 бўйича 5 сонига тескари эканлигини топайлик. Бу ерда, 7 сони туб бўлгани учун, унинг Эйлер функцияси $\varphi(7)=7-1=6$, модул 7 бўйича 5 сонига тескари сон эса $a^{-1} \equiv a^{\varphi(n)-1} (\mathrm{mod}\, n)$ формулага кўра $5^{-1}=5^{6-1} \bmod 7=5^5 \bmod 7=3125 \bmod 7=3$. Ҳақиқатан ҳам, $5 \cdot 3 \bmod 7=15 \bmod 7=1 \bmod 7=1$. Бирор модул бўйича берилган сонга тескари бўлган сон ҳар доим ҳам мавжуд бўлавермайди. Мисол учун, 5 сонига модул 14 бўйича тескари сон $3:5 \cdot 2 \bmod 14=15 \bmod 14=1 \bmod 14=1$. Аммо, 2 сонининг модул 14 бўйича тескариси мавжуд эмас, яъни $2x \equiv 1 (\mathrm{mod}\ 14)$ ёки $2x=14k+1$ тенглама x ва k но-

маълумларнинг бутун қийматларида ечимга эга эмас, чунки, x ва k номаълумларнинг бутун қийматларида, ҳар доим, тенгликнинг чап томонида жуфт сон, ўнг томонида эса тоқ сон хосил бўлади.

Умумий ҳолда, агар, a ва n сонлари ўзаро туб бўлса, тенглама $a^{-1} \equiv x \bmod n$ ягона ечимга эга бўлади; агар, a ва n сонлари ўзаро туб бўлмаса, тенглама $a^{-1} \equiv x \pmod n$ ечимга эга эмас. Бевосита ҳисоблашлар асосида, ушбу $(a \cdot x) \bmod n = b$ тенглама a, n, b – сонларининг қандай қийматлар қабул қилишига қараб, ёки бир нечта ечимларга эга бўлиши мумкинлигига, ёки битта ҳам ечимга эга бўлмаслигига ишонч хосил қилиш мумкин.

Қуйидагиларни таъкидлаш жоиз: агар a сони M сонини бўлса ва b сони ҳам M сонини бўлса, у ҳолда бу $M \in N$ сони $a, b \in Z$ сонларнинг *умумий бўлинувчиси* (*карралиси*) дейилади. Умумий бўлинувчилар ичида энг кичиги *энг кичик умумий бўлинувчи* дейилади, ҳамда, $[a, b]$ деб белгиланади.

2.2–теорема. Агар $M \in N$ сон $a, b \in Z$ сонларнинг умумий бўлинувчиси бўлса, у ҳолда M сони бу сонларнинг энг кичик бўлинувчиси $[a, b]$ га ҳам бўлинади.

2.3–теорема. Ушбу $[a, b] = b/(a, b)$ муносабат ўринли.

2.4 – теорема (Қолдиқлар ҳақида Хитой теоремаси). Агарда n сонининг туб кўпайтувчилари $p_1, p_2, \ldots, p_t$ сонлардан иборат (яъни $n = p_1^{k_1} \ldots p_1^{k_t}$) бўлса, у ҳолда

$$x(\bmod p_i) \equiv a_i, \; i = 1,2, \ldots, t;$$

тенгламалар тизимси ягона $x < n$ ечимга эга бўлади.

Квадратик чегирмалар. Агар p – туб сон ва $0 < a < p$ бўлиб, ушбу

$$x^2 \equiv a(\bmod p)$$

муносабатни қаноатлантирувчи x – номаълумнинг қийматлари мавжуд бўлса, у ҳолда a сони модуль p бўйича квадратик чегирма ҳисобланади.

Мисол учун, $p = 7$ бўлса, квадратик чегирма ташкил этувчилар: 1, 2 ва 4 сонларидан иборат, яъни $a = 1$, $a = 2$ ва $a = 4$ қийматларда, ушбу таққосламалар

$1^2 = 1 \equiv 1(\bmod 7)$; $2^2 = 4 \equiv 4(\bmod 7)$; $3^2 = 9 \equiv 2(\bmod 7)$; $4^2 = 16 \equiv 2(\bmod 7)$;

$5^2 = 25 \equiv 4(\bmod 7)$; $6^2 = 36 \equiv 1(\bmod 7)$;

ўринли.

Номаълум x нинг қуйидаги муносабатларни:

$$x^2 \equiv 3(\bmod 7); \; x^2 \equiv 5(\bmod 7); \; x^2 \equiv 6(\bmod 7),$$

қаноатлантирувчи қийматлари мавжуд эмас, шунинг учун $a=3$, $a=5$ ва $a=6$ сонлари модул 7 бўйича квадиратик чегирма эмас, яъни берилган квадратик таққосламалар ечимга эга эмас.

Модул p жуфт бўлса, у холда $(p-1)/2$ та квадратик чегирма мавжуд ва шунча квадратик чегирма мавжуд эмас, яъни ушбу

$$x^2 \equiv a(\mathrm{mod}\ p)$$

муносабатни қаноатлантирувчи x – номаълум мавжуд бўладиган

a – параметрнинг мумкин бўлган қийматлари сони $(p-1)/2$ та, бу муносабатни қаноатлантирувчи x – номаълум мавжуд бўлмайдиган a – параметрнинг мумкин бўлган қийматлари сони ҳам $(p-1)/2$ та. Бундан ташқари, агарда a сони модул p бўйича квадратик чегирма бўлса, у ҳолда a учун иккита квадрат илдиз мавжуд бўлиб, улардан бири $[0;\ (p-1)/2]$ ораликда, иккинчиси $[(p-1)/2;\ p-1]$ ораликда, шу билан бирга улардан бири модул p бўйича квадратик чегирма бўлади ва у *бош квадратик илдиз* дейилади.

Ясовчи (Тузувчи). Берилган p – туб сон ва $g < p$ учун, g – ясовчи (*тузувчи*) ёки модуль p бўйича *примитив илдиз* дейилади, агарда $1 \le b \le p-1$ шартни қаноатлантирувчи ҳар бир b сони учун, ушбу $g^a \equiv b(\mathrm{mod}\ p)$ муносабатни қаноатлантирувчи a сони мавжуд бўлса. Мисол учун, бевосита ҳисоблаш билан, 2 сони модуль $p = 11$ бўйича ясовчи эканлигига ишонч ҳосил қилиш мумкин, яъни 1 дан 10 гача бўлган барча натурал сонларни a сонининг бирор қийматида 2^a $(\mathrm{mod}\ p)$ кўринишда ифодалаш мумкин. Модуль $p = 11$ бўйича: 2, 6, 7, 8 – сонлари ясовчи бўла олади, 3, 4, 5, 9, 10 – сонлари ясовчи бўла олмайди.

Умумий ҳолда, берилган g сони бирор модул p бўйича ясовчи бўлиши ёки бўлмаслигини текшириш учун, $p-1$ сони туб кўпайтувчиларга ажратилиб, яъни

$p-1 = q_1^{\alpha_1} q_2^{\alpha_2} \ldots q_n^{\alpha_n}$ ифодадан q_1, q_2, ..., q_n – сонлари аниқланиб, $g^{(p-1)/q_i} \bmod p = P_i$ сонлари ҳисобланади. Агар барча $P_i \neq 1 (i = 1, 2, \ldots, n;)$ бўлса, у ҳолда g сони модуль p бўйича ясовчи бўлади, акс ҳолда: бирор $P_l = 1 (1 \le l \le n)$ бўлса, у ҳолда g сони модуль p бўйича ясовчи бўлмайди.

Галуа майдони. Агар p – туб сон бўлса, у ҳолда 0 дан $p-1$ гача бўлган барча натурал сонлар тўплами $\{0, 1, 2, \ldots, p-1\}$ элементлари сони p та бўлган чекли майдон ташкил этади ва бу майдон *Галуа майдони* деб юритилади, ҳамда, $GF(p)$ кўринишда белгиланади. Галуа майдонида қўшиш, айириш, кўпайтириш ва нолдан фарқли бўлган элементга бўлиш амаллари аниқланган. Ҳамда, бу майдонда: қўшиш амалига нисбатан ихтиёрий $a \in GF(p)$ учун $a+0=a$ тенгликни

қаноатлантирувчи 0 (ноль) элемент мавжуд; кўпайтириш амалига нисбатан ихтиёрий $a \in GF(p)$ учун $a \cdot 1 = a$ тенгликни қаноатлантирувчи 1 (бир) элемент мавжуд; ихтиёрий нолдан фарқли $a \in GF(p)$ учун $(a \cdot b) \bmod p = 1$ тенгликни қаноатлантирувчи $b \in GF(p)$ элемент мавжуд бўлиб, бу b элемент a элементга *тескари* элемент дейилади ва a^{-1} деб белгиланади; аниқланган амаллар коммутативлик, ассоциативлик ва дистрибутивлик хоссаларига эга.

Галуа майдони билан боғлиқ бўлган математик тушунча ва тасдиқлар криптографияда кенг қўлланилади.

Криптографик масалаларни яна ҳам мураккаблаштириш мақсадида келтирилмайдиган (кўпайтувчиларга ажрамайдиган), коэффициентлари ушбу {0, 1, …, $q-1$} (бу ерда q – туб сон) тўпламдан бўлган барча n – тартибгача кўпҳадлар тўпламидан фойдаланилади. Бу кўпҳадлар майдони $GF(q^n)$ деб белгиланади. Ҳамма амаллар характеристикаси n – тартибли келтирилмайдиган кўпҳад $p(x) \in GF(q^n)$ билан аниқланадиган майдонда бажарилади. Мисол учун, $GF(2^3)$ майдон ушбу: 0, 1, x, $x+1$, x^2 , x^2+1, x^2+x, x^2+x+1 элементларни ўз ичига олади.

Берилган $GF(q^n)$ майдондан олинган $p(x)$ кўпҳаднинг коэффициентлари ўзаро туб бўлса, бу кўпҳад *ясовчи* бўлади, ҳамда, *примитив* (*содда*) дейилади. Примитив кўпҳадлар *чизиқли тескари боғлиқликка эга* бўлган силжиш регистрлари билан узвий боғлиқликка эга, яъни $q=2$ бўлганда $GF(2^n)$ майдонда бажариладиган амалларни чизиқли тескари боғлиқликка эга бўлган силжитиш регистрларининг аппарат-техник қурилмалари ёрдамида тез бажариш мумкин. Ҳақиқатан ҳам, даражага кўтариш амалини бажариш $GF(q^n)_{q<2}$ майдондагидан кўра $GF(2^n)$ майдонда самаралидир. Бундан эса $GF(2^n)$ майдонда дискрет логарифмларни ҳисоблашнинг ҳам самарали эканлиги келиб чиқади.

Коэффициентлари иккилик саноқ тизимси элементларидан иборат n – тартибгача бўлган барча кўпҳадлар тўплами $GF(2^n)$ – Галуа майдонида модуль – майдон характеристикаси сифатида $p(x)=x^n+x+1$ кўринишдаги уч ҳаддан иборат бўлган n – тартибли примитив кўпҳад олинади. Майдон характеристикасининг бундай танлаб олиниши, яъни x^n ва x ҳадлар оралиғидаги: x^{n-1}, x^{n-2}, …, x^2 ҳадларнинг йўқлиги модуль бўйича кўпайтириш амалининг самарали бажарилишини таъминлайди. Майдон характеристикасини ифодаловчи $p(x)=x^n+x+1$ кўпҳад примитив бўлмаса, амалларнинг бажарилиши мураккаблашади ҳамда криптографик самарадорликка эришилмайди. Мисол учун, бевосита ҳисоблаш натижасида, n нинг 1000 дан кичик бўлган: 1, 3, 4, 6, 9, 15, 22, 28, 30, 46, 60, 63, 127, 153, 172, 303, 471, 532, 865, 900 [11],

293-бет) қийматларида $p(x)=x^n+x+1$ кўпҳад примитивлик хоссасига эга бўлади.

Туб кўпайтувчиларга ажратиш. Берилган сонни кўпайтувчиларга ажратиш деганда, унинг туб кўпайтувчиларини топиш тушунилади.

Мисол учун:

100 сони 2, 2, 5 ва 5 туб сонларидан иборат кўпайтувчиларга эга, яъни $100=2\cdot2\cdot5\cdot5$;

6279 сони 3, 7,13 ва 23 туб сонларидан иборат кўпайтувчиларга эга, яъни $6279=3\cdot7\cdot13\cdot23$.

Берилган сонни кўпайтувчиларга ажратиш сонлар назариясининг энг дастлабки масалаларидан бири ҳисобланади. Берилган сонни (ёки тўпламни) бирор амал ёки хусусиятга кўра унинг ташкил этувчилари орқали ифодаланиши, шу сонни (ёки тўпламни) факторлаш (ажратиш) дейилади. Сонни кўпайтувчиларга ажратиш қийин жараён эмас, аммо кўпайтувчиларга ажратилиши керак бўлган соннинг қиймати катталашиб бориши билан, уни кўпайтувчиларга ажратиш жараёнига сарфланадиган вақт ҳам кўпайиб боради. Шундай бўлсада, кўпайтувчиларга ажратиш жараёнини тезлаштирувчи қуйидаги алгоритмлар мавжуд [11], 294-бет):

1. *Сонли майдон умумий ғалвир усули* –ўнлик саноқ тизимсида 110 та ва ундан кўп разрядли (рақамли) сонларни кўпайтувчиларга ажратишнинг маълум бўлган энг самарали (тез, кам вақт сарфланадиган) алгоритми;

2. *Квадратик ғалвир усули* – ўнлик саноқ тизимсида 110 тадан кам бўлмаган разрядли (рақамли) сонларни кўпайтувчиларга ажратишнинг маълум бўлган энг самарали (тез ва кам вақт сарфланадиган) алгоритми;

3. *Эллиптик эгри чизиқ усули* – ўнлик саноқ тизимида туб кўпайтувчиларининг разряди (рақамлари сони) 43 тадан кўп бўлмаган сонларни кўпайтувчиларга ажратишда фойдаланилган;

4. *Поллардниng Монте-Карло усули* – амалда кам ишлатилади;

5. *Узулуксиз касрлар усули* – қўллашга кўп вақт сарфланади;

6. *Танлаб бўлиш усули* – энг дастлабки усуллардан бўлиб, кўпайтувчиларга ажратилиши керак бўлган (берилган) соннинг квадрат илдизига тенг ва ундан кичик бўлган ҳар бир туб сонни берилган сонни қолдиқсиз бўлиши ёки бўлмаслиги текшириб чиқилиши натижасида, берилган соннинг туб кўпайтувчилари аниқланади.

Модуль *n* бўйича квадрат илдиз. Агарда майдон характеристикасини ифодаловчи n сони иккита туб соннинг кўпайтмасидан иборат бўлса,

у ҳолда соннинг квадрат илдизини модуль n бўйича топиш масаласини ечиш n сонини кўпайтувчиларга ажратиш масаласини ечиш ҳисоблаш нуқтаи назаридан тенг кучли масалалар ҳисобланади. Яъни, майдон характеристикасини ифодаловчи n сонининг кўпайтувчилари маълум бўлса, берилган ихтиёрий соннинг квадрат илдизини модул n бўйича ҳисоблаш қийинчилик туғдирмайди, акс ҳолда ҳисоблашлар n сонининг туб кўпайтувчиларини топиш масаласи каби мураккабликларни ўз ичига олади. Майдон характеристикаси етарлича катта бўлганда криптобардошлилиги квадрат илдизни ҳисоблаш масаласининг мураккаблигига асосланган очиқ калитли криптоалгоритмлар мавжуд.

Туб сонлар генерацияси (ишлаб чиқариш). Очиқ калитли криптоалгоритмлар асослари яратилишида туб сонларнинг хоссаларидан фойдаланилади. Бирор берилган сонни туб кўпайтувчиларга ажратиш, уни туб ёки туб эмаслигини аниқлашга нисбатан мураккаб бўлган масала. Етарли катта разряддаги тоқ сонни тасодифий танлаб олиб, уни кўпайтувчиларга ажратиш билан туб ёки туб эмаслигини аниқлашдан кўра, уни тублигини бирор мавжуд усул билан текшириш осонроқ. Бунинг учун турли эҳтимоллик тестлари мавжуд бўлиб [11], 97-бет), соннинг тублигини берилган даражадаги ишонч билан аниқлаб беради. Криптобардошлилиги етарли даражада катта разрядли сонни туб кўпайтувчиларга ажратиш масаласининг мураккаблигига асосланган очиқ калитли криптоалгоритмлар мавжуд.

Чекли майдонларда дискрет логарифмлаш. Криптографияда бир томонлама (тескариси йўқ) функция сифатида бирор модуль n бўйича даражага кўтариш амалини ҳисоблашдан фойдаланилади:

$$y = a^x \bmod n.$$

Бу функциянинг y –қийматини x –аргументнинг берилган қиймати бўйича ҳисоблаш қийинчилик туғдирмайди. Аммо, y нинг қийматини билган ҳолда x нинг қийматини топиш мураккаб масала ҳисобланади. Умуман олганда,

$$a^x \equiv b (\bmod\ n)$$

муносабатни қаноатлантирувчи x – номаълумнинг бутун қийматлари ҳар қандай n лар учун ҳам мавжуд бўлавермайди. Мисол учун, ушбу

$$3^x \equiv 7 (\bmod\ 13)$$

муносабат x нинг ҳеч бир бутун қийматида бажарилмайди. a, b, n – параметрларнинг етарли катта қийматларида юқорида келтирилган масаланинг ечими яна ҳам мураккаблашади.

Криптографияда асимметрик шифрлаш алгоритмларининг асослари билан боғлиқ бўлган қуйидаги:

- туб сонлар майдонида $GF(p)$ дискрет логарифмлаш;
- характеристикаси асоси 2 бўлган $GF(2^n)$ майдонда дискрет логарифимлаш;
- эллиптик эгри чизиқ нуқталари устида бажариладиган амалларни бирор чекли F майдонда амалга ошириш масалаларини ечишнинг мураккаблиги билан боғлиқ бўлган муаммолар асосида иш кўрилади.

Криптобардошлилиги дискрет логарифимлаш масаласининг мураккаблигига асосланган кўплаб очиқ калитли криптоалгоритмлар мавжуд.

§2.6. Бул функциялари

Фикрлар алгебраси ва улар устида амаллар

Логика сўзи грекча «**логос**» сўзидан олинган бўлиб, «**фикр**», «**сўз**», «**ибора**», «**тушунча**» деган маъноларни билдиради. Логика фан сифатида эса фикрлашни ўрганади. Логика фани тўғри фикрлашнинг қонун ва қоидаларини ўрганувчи фандир [16–18].

Математик логика эса анъанавий логикадан ажралиб чиққан бўлиб, тўғри фикрлашнинг тузилишига математик методларни қўллаб ўрганувчи фандир. Шу билан бирга математик логика илмий назария ва тасдиқларни исботлаш жараёнини ҳам ўз ичига олади.

Математик логиканинг вазифаси – формулалар ёрдамида ёзилган маълум фикрлардан бошқа фикрларни келтириб чиқаришга имкон берадиган мулоҳаза юритишнинг умумий методларини баён қилишдан иборат.

Дискретлик лотинча **discretus** – узлукли, узлуксизликка қарама-қарши тушунча бўлиб, берилган тўплам элементлари орасида «сакраш», «оралиқ» деган маънони билдиради [16–18].

Фикрлар алгебраси – мавжуд бўлган элементар фикрлардан янги фикрлар қуриш ва бундай қуришларнинг усулларини ўрганувчи фан.

Фикр деганда, одатда унинг чинлиги ёки ёлғонлиги ҳақида гапириш, ўйлаш ва маънога эга бўлиши мумкин бўлган бирор даъво тушинилади. Фикр чин ёки ёлғон бўлиб, у бир вақтда ҳам чин, ҳам ёлғон бўлолмайди.

Барча фикрлар тўпламини мавжуд деб ҳисоблаб, бу тўпламнинг элементларини элементар фикрлар деб атаймиз. Фикрлар одатда ло-

тин алифбосининг бош ҳарфлари A, B, $C...X$, Y, Z билан белгиланади. Бунда, индексли белгилардан ҳам фойдаланиш мумкин, яъни A_1, A_2, ..., B_1, B_2... .

Масалан:

A_1: «Тошкент – Ўзбекистоннинг пойтахти»

A_2: «Фарғона шаҳри Амударё бўйида жойлашган»

A_3: «Инсон умри абадий эмас»

A_4: «А. Навоий – ғазал мулкининг султони»

A_5: « Етти бешдан кичик»

A_6: «Сирдарё орол денгизига қуйилади»

A_7: «Пушкин – буюк рус математиги»

A_8: «Қор- оппоқ»

Рост фикрга «1» белги, ёлғон фикрга эса «0» белги мос қўйилади:

$$A = \begin{cases} 0, \text{ агар } A - \text{ёлғон бўлса;} \\ 1, \text{ агар } A - \textit{рост} \text{ бўлса;} \end{cases}$$

у ҳолда, келтирилган фикрлар қийматлари: $A_1 = 1$, $A_2 = 0$, $A_3 = 1$, $A_4 = 1$, $A_5 = 0$, $A_6 = 1$, $A_7 = 0$ ва $A_8 = 1$ бўлади.

Нутқда фикрлар ўртасида «ва», «ёки» ва ҳоказо боғловчилардан фойдаланилади. Бу боғловчилар ҳар хил фикрларни ўзаро боғлаб, янги мураккаб фикрлар тузишга имкон беради.

Мисоллар.

1. «3 сони 15 нинг бўлувчиси» – деган фикр ва «3 – туб сон» деган фикр берилган бўлса, биз бу икки рост фикрдан «ва» – боғловчиси орқали қуйидагича янги рост фикрни қуришимиз мумкин: « 3 сони 15 нинг бўлувчиси ва 3 – туб сон»;

2. «Агар $\sqrt{2}$ – иррационал сон бўлса, у ҳолда $\sqrt{2}+1$ сони ҳам иррационал бўлади» – деган фикр икки фикрни «агар ... бўлса, у ҳолда ... бўлади» – боғловчиси билан боғлаш натижасида ҳосил қилинган.

Шунингдек, берилган фикрни «инкор» қилиш орқали ҳам янги фикр қуриш мумкин. Элементар фикрлар орқали боғловчилар ёрдамида мураккаб бўлган фикрлар ҳосил қилишимиз мумкин. Яъни, фикрлар устида қуйидаги логик амалларни бажаришимиз мумкин.

2.1-таъриф (фикрнинг инкори): A фикрнинг инкори деб, A рост бўлганда ёлғон бўладиган, A ёлғон бўлганда рост бўладиган ва $\neg A$ (ёки $\overline{A}$) каби белгиланувчи бошқа бир фикрга айтилади.

Бу таърифни тушинтириш учун фикр инкорининг чинлик жадвали деб аталувчи қуйидаги жадвални киритамиз:

A	⌝A (ёки $\bar{A}$)
0	1
1	0

Фикрнинг инкори

2.2-таъриф (икки фикрнинг конъюнкцияси) А ва В фикрларнинг конъюкцияси деб, шундай янги фикрга айтиладики, бу фикр берилган иккала фикр бир пайтда рост бўлгандагина рост бўлиб, қолган холларда ёлғон бўлади, ҳамда, А∧В ёки А&В каби белгиланади.

Бу таърифни тушинтириш учун икки фикрнинг конъюнкцияси чинлик жадвали деб аталувчи жадвални киритамиз:

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

Икки фикрнинг конъюнкцияси

Берилган А ва В фикрларнинг конъюкцияси «А ва В», «ҳам А, ҳам В», «А хамда В» кўринишида курилади.

2.3-таъриф (икки мулоҳазанинг дизъюнкцияси): А ва В фикрларнинг дизъюнкцияси деб, шундай янги фикрга айтиладики, бу фикр А ва В фикрлардан ақалли биттаси рост бўлгандагина рост бўлади. Унинг белгиланиши А∨В кўринишда бўлиб, унинг чинлик жадвали:

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

Икки мулохазанинг дизъюнкцияси

Берилган А ва В фикрларнинг дизъюнкцияси «А ёки В», «ё А, ё В» каби курилади.

2.4-таъриф (икки фикрнинг импликацияси): А ва В фикрларнинг импликацияси деб, шундай янги фикрга айтиладики, А рост бўлиб, В

ёлғон бўлганда ёлғон, қолган холларда рост бўлади, ҳамда, белгиланиши А→В кўринишда бўлиб, унинг чинлик жадвали:

А	В	А→В
0	0	1
0	1	1
1	0	0
1	1	1

Икки фикрнинг импликацияси

«Агар А бўлса, у ҳолда В», «А дан В келиб чикади», «А – фикр, В – фикр учун етарли», «В – фикр, А – фикр учун зарур».

Келтирилган шартли фикрда А – асос, В – эса хулоса дейилади. Импликация хулоса чиқаришда муҳим роль ўйнайди ва теоремалар, ҳар хил тушунчалар таърифлари шаклланади.

2.5-таъриф (икки мулоҳазанинг эквивалентлиги): А ва В фикрларнинг эквивалентлиги деб, шундай янги фикрга айтиладики, бу фикр бир пайтда рост ёки бир пайтда ёлғон бўлгандагина рост фикр, қолган ҳолларда ёлғон бўлади ҳамда белгиланиши А↔В кўринишда бўлиб, унинг чинлик жадвали:

А	В	А↔В
0	0	1
0	1	0
1	0	0
1	1	1

Икки фикрнинг эквивалентлиги (эквиваленцияси)

«А фикр В фикрга эквивалент», «А фикр В фикрга тенг кучли» «А фикр В фикр учун зарур ва етарли».

Фикрлар алгебрасида формулаларнинг эквивалентлиги тушунчаси муҳим роль ўйнайди.

2.6-таъриф. Агарда А← →В бўлса (агар А – фикрнинг чинлигидан В – фикрнинг чинлиги келиб чиқса ва В – фикрнинг чинлигидан А – фикрнинг чинлиги келиб чиқса, у ҳолда А ва В формулалар эквивалент ёки тенг кучли дейилади ва А~В кўринишда белгиланади.

А ва В формулалар эквивалент бўлса, у ҳолда чинлик функциялари бўйича улар тенг. Формулаларнинг эквивалентлиги тушунчаси

(← →) – эквиваленция амали орқали киритилади, аммо эквивалент тушунчаси эквиваленция амалидан фарқ қилади.

Формулаларни ўзгартириб, ўзига эквивалент, лекин соддароқ ёки маълум бир масалаларни ечиш учун қулайроқ бўлган бўлган формулалар ҳосил қилиш мумкин.

Айрим айниятлар алгебраик амалларнинг асосий хоссаларини ифодалагани каби, фикрлар алгебрасида айрим эквивалентликлар, амалларнинг муҳим хоссаларини ифодалайди.

Инкор, конъюнкция, дизъюнкция амалларининг таърифидан бевосита қуйидаги эквивалентликлар келиб чиқади:

1) ⌉⌉A← →A;
2) A∧B← →B∧A;
3) A∧(B∧C)← →(A∧B)∧C;
4) A∨B← →B∨A;
5) A∨(B∨C)← →(A∨B)∨C.

Биринчи эквивалентлик «кўш инкор» қонуни деб аталади. 2) ва 5) – эквивалентликлар мос равишда конъюнкция ва дизъюнкция амалларининг коммутативлик ва ассоциативлик қонунларини ифодалайди.

Қуйидаги икки амал ўзаро дистрибутивлик қонунлари билан боғланган:

6) A∨B∧C← →(A∨B)∧(A∨C);
7) A∧B∨C← →(A∧B)∨(A∧C).

Шундай қуйидагича эквивалентликлар (Морген қонунлари ҳам деб юритилади) ўринли:

8) ⌉(A∨B)← →⌉A∨⌉B
9) ⌉(A∧B)← →⌉A∧⌉B

Айний чин фикрни «Ч» орқали ва айний ёлғон фикрни «Ё» орқали белгиланса, қуйидаги эквивалентликлар ўринли бўлади:

10) A∧Ч← →A;
11) A∧Ё← →Ё;
12) A∨Ч← →Ч;
13) A∨Ё← →A;
14) A∧A← →A;
15) A∧A← →A.

Қуйидаги эквивалентликлар «импликация» амалига тегишли:

16) A→B← →⌉A∨B;
17) (A→B)← →(⌉B→⌉A).

Импликация ва эквиваленция амалларини боғловчи эквивалентлик ҳам мавжуд:

18) (A↔B)← →(A→B)∧(B→A)

Келтирилган 16) ва 18) – эквивалентликлардан эса ушбу:

19) $(A \leftrightarrow B) \leftarrow \rightarrow (\neg A \vee B) \wedge (A \vee B)$

эквивалентлик келиб чиқади.

Қўшимча қилиб мантиқий формулаларни соддалаштиришда қўлланиладиган қуйидаги эквивалентликларни келтириш мумкин:

21) $A \vee (A \wedge B) \sim A$;

22) $A \wedge (A \vee B) \sim A$;

23) $A \vee (\neg A \wedge B) \sim A \vee B$;

24) $\neg A \vee (A \wedge B) \sim \neg A \vee B$

25) $A \wedge (\vee A \vee B) \sim A \wedge B$;

26) $\neg A \wedge (A \vee B) \sim \neg A \wedge B$;

27) $A \wedge B \sim \neg A \vee \neg B$

Юқорида кўриб ўтилган бешта жадвал фикрлар устида киритилган бешта амалнинг таърифидир. Бу амаллар «логик амаллар» ёки «мантиқий амаллар» деб аталади. Бу жадваллардан кўринадаки, A ва B – фикрлар бўлса, у ҳолда $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ ифодалар ҳам фикрлардир. Яъни, ҳар қандай мураккаб гапнинг ўзи ҳам фикрдир.

Мисол. A, B, C, D – фикрларнинг чин қийматлари мос равишда Ч, Ё, Ё, Ч – бўлса, улар орқали ифодаланган $((A \vee B) \rightarrow (C \leftrightarrow \neg D))$ – мураккаб фикрнинг Ч ёки Ё эканлиги аниқлансин.

Ечиш: Мураккаб фикрнинг қисмлари бўлган: $A \vee B$ – фикрнинг қиймати Ч, $\neg D$ – нинг қиймати Ё, $C \leftrightarrow \neg D$ – нинг қиймати Ч бўлиб, берилган фикрнинг чин қиймати Ч эканлиги келиб чиқади.

Қавслардан камроқ фойдаланиш мақсадида, сонлар алгебрасидаги каби келишиб олинади. Ташқи қавслар тушириб қолдирилади. Қавслар билан кўрсатилмаган бўлса, аввал $\neg$(ёки) – инкор амали бажарилади, сўнгра $\wedge$ – амали, ундан кейин эса $\vee$ – амали, улардан кейин $\rightarrow$ – амали, охирида $\leftrightarrow$ – амали бажарилади. Яъни, таъсир соҳасининг кенглиги бўйича энг кучли амал эквиваленция ундан кейин импликация, дизъюнкция, конъюнкция ва инкор амаллари келади. Бир хил амал бир неча марта кетма-кет келган бўлса, амалларни бажаришда чапдан ўнгга силжиймиз.

Мисол. 1) $(((A_1 \vee A_2) \rightarrow A_3) \wedge (A_1 \vee A_3)) \Leftrightarrow (A_1 \vee A_2 \rightarrow A_3) \wedge (A_1 \vee A_3)$;

2) $((A_2 \vee A_3) \leftrightarrow (A_2 \vee A_4)) \Leftrightarrow A_2 \vee A_3 \leftrightarrow A_2 \vee A_4$.

2.7-таъриф. Фикрий ўзгарувчиларнинг қабул қилиши мумкин бўлган барча қийматлари учун формула фақат чин қийматларни қабул қилса, бундай формулага айнан чин формула ёки тавтология дейилади.

Масалан,

1) $A \vee \neg A$ – формула, А нинг барча қабул қилиши мумкин бўлган қийматларида:

$$А=Ч : Ч \vee \neg Ч = Ч \vee Ё = Ч;$$
$$А=Ё : Ё \vee \neg Ё = Ё \vee Ч = Ч;$$

фақат чин қийматларни қабул қилади, демак бу формула тавтология.

Худди шундай:

2) $А \rightarrow А$; $\neg(А \wedge \neg А)$ –формулалар ҳам тавтология бўлишини текшириб ишонч ҳосил қилиш мумкин.

2.8-теорема. Агарда А ва $А \rightarrow В$ формулалар тавтология бўлса, у ҳолда В –формула ҳам тавтология бўлади.

Фикрлар алгебрасида формулалар эквивалентлиги тушунчаси катта аҳамиятга эга, чунки формулаларнинг бу ерда ўрганиладиган асосий хусусиятлари эквивалент формулаларга ўтилганда сақланади. Шу сабабли ҳар бир формула учун унга эквивалент бўлган, лекин мумкин қадар содда бўлган формулалар тузиш жуда муҳимдир. Фикрий ўзгарувчилардан $\neg$, $\wedge$, $\vee$ – амаллар ёрдамида махсус тарзда тузилган, дизъюнктив нормал формадаги ёки конъюктив нормал формадаги формулалар деб аталувчи формулалар ана шундай турдаги формулалардир. Қуйида ушбу нормал формалар билан танишилади.

2.9-таъриф. $X_1, X_2, ..., X_n$ – ўзгарувчиларнинг дизъюнктив бир ҳади деб, бу ўзгарувчиларнинг ёки уларнинг инкорларини дизъюнкциясига айтилади, яъни:

$$X_1 \vee X_2 \vee X_3 \vee X_4 \text{ ёки } X_1 \vee \neg X_1 \vee X_2 \vee \neg X_3.$$

2.10-таъриф. Дизъюнктив нормал форма (ДНФ) деб, конъюктив бир ҳадларнинг дизъюнкциясига айтилади, яъни:

$$K_1 \vee K_2 \vee \ldots \vee K_p,$$

бу ерда: K_i конъюктив бир хадлар бўлиб, баъзилари бир бири билан устма-уст тушиши ҳам мумкин.

2.11-таъриф. Конъюктив нормал форма (КНФ) деб, дизъюнктив бир ҳадларнинг конъюкциясига айтилади, яъни:

$$D_1 \wedge D_2 \wedge \ldots \wedge D_q,$$

бу ерда D_i – дизъюнктив бир хадлар бўлиб, баъзилари бир-бири билан устма-уст тушиши ҳам мумкин.

Бу таърифларда $p = 1$, $q = 1$ бўлиши мумкин, яъни конъюктив бир ҳад ёки дизъюнктив бир ҳаднинг ўзи ҳам нормал форма дейилиши мумкин.

Ҳар қандай формулаларни ҳам конъюктив, ҳам дизъюнктив нормал формулаларга келтириш мумкин, масалан:

$$P \vee Q \cong \neg(\neg P \wedge \neg Q) \text{ ёки } P \wedge Q \cong \neg(\neg P \vee \neg Q).$$

Берилган F формулани ДНФ ва КНФ кўринишларига келтиришнинг бир нечта усуллар орқали амалга ошириш мумкин. Улардан баъзилари «мураккаб» кўринишда, баъзи бирлари эса «соддароқ» кўринишда бўлади.

Фикрлар алгебрасидаги F – формула дизъюнктив (конъюктив) нормал форма формулаларининг ичида энг «соддароқ» бўлган ягона бир кўриниши мавжуд. Бу кўриниш одатда мукаммал дизъюнктив (конъюктив) нормал форма деб юритилади [16–18].

2.12-таъриф. Агар X_i ва $\neg X_i$ $(i = \overline{1, n})$ жуфтликлардан албатта фақат биттаси қатнашса $X_1, X_2, ..., X_n$ – ўзгарувчилар қатнашган конъюктив (дизъюнктив) бир ҳади мукаммал дейилади.

2.13-таъриф. $X_1, X_2, ..., X_n$ – ўзгарувчилар қатнашган нормал форма (дизъюнктив ёки конъюктив) мукаммал дейилади, унда фақат мукаммал бир ҳадлар қатнашган бўлса, масалан:

$$F(X_1, X_2, X_3) \equiv (\neg X_1 \wedge \neg X_2 \wedge \neg X_3) \wedge (X_1 \wedge X_2 \wedge \neg X_3) \wedge (X_1, X_2, X_3) - \text{МКНФ};$$
$$F(X_1, X_2, X_3) \equiv (\neg X_1 \wedge \neg X_2 \wedge X_3) \vee (X_1 \wedge X_2 \wedge \neg X_3) \vee \wedge (X_1 \vee X_2 \vee X_3) - \text{МДНФ}.$$

Қуйида фикрлар алгебраси формулаларининг мукаммал дизъюнктив нормал форма (МДНФ) кўринишида ифодаланиши кўриб ўтилган.

Кейинги фикрлашлар учун қуйидаги белгилашлардан фойдаланилди:

$$X^{\alpha} = \begin{cases} X, & \text{агар } \alpha = 1; \\ \overline{X}, & \text{агар } \alpha = 0. \end{cases}$$

Хусусий ҳолда қуйидагилар ўринли:

$$0^0 = 1; \quad 0^1 = 0; 1^0 = 0; \quad 1^1 = 1.$$

Демак, $X = \alpha$ бўлганда $X^{\alpha} = 1$, ҳамда, $X \neq \alpha$ бўлганда $X^{\alpha} = 0$ бўлади, яъни:

$$X^0 = \neg X, \qquad X^1 = X.$$

Бундан ташқари қуйидаги белгилашларни киритамиз: $X_1 \vee X_2 \vee ... \vee X_n = \bigvee_{i=1}^{n} X_i$, хусусий ҳолда $\bigvee_{\alpha 1, \alpha 2, ..., \alpha n} H(X_1, X_2, ..., X_n, \alpha_1, \alpha_2, ..., \alpha_n)$ белгилаш $H(X_1, X_2, ..., X_n, \alpha_1, \alpha_2, ..., \alpha_n)$ ифоданинг мумкин бўлган барча $(\alpha_1, \alpha_2, ..., \alpha_n)$ танланмалар бўйича дизъюнкцияси тушунилади, бу ерда: $\alpha_i \in \{0, 1\}$, $i = \overline{1, n}$

Масалан $\bigvee_{\alpha_1,\alpha_2}(x_1^{\alpha_1}\wedge x_2^{\alpha_2})=(x_1^0\wedge x_2^1)\vee(x_1^1\wedge x_2^0)\vee(x_1^1\wedge x_2^1)=(\overline{X}_1\wedge\overline{X}_2)\vee$

$\vee(\overline{X}_1\wedge X_2)\vee(X_1\wedge\overline{X}_2)\vee(X_1\wedge X_2);$

$F(X_1, X_2, ..., X_n)\cong\bigvee_{\alpha_1,\alpha_2,...,\alpha_n}(F(\alpha_1, \alpha_2, ..., \alpha_n)\wedge x_1^{\alpha_1}\wedge x_2^{\alpha_2}\wedge...\wedge x_n^{\alpha_n}).$

2.1-лемма Фикрлар алгебрасидаги ихтиёрий $F(X_1, X_2, ..., X_n)$ формула учун, ушбу

$$F(X_1, X_2, ..., X_n)\cong\bigvee_{\alpha_1,\alpha_2,...,\alpha_n}(F(\alpha_1, \alpha_2, ..., \alpha_n)\wedge x_1^{\alpha_1}\wedge x_2^{\alpha_2}\wedge...\wedge x_n^{\alpha_n}).$$

ёйилма ҳар доим ўринли.

Мисол $F(X, Y, Z)=(X\Leftrightarrow Y)\wedge Z$ функциянинг чинлик жадвали келтирилган:

X	Y	Z	$X\Rightarrow Y$	$(X\Rightarrow Y)\wedge Z$
0	0	0	1	0
0	0	1	1	1
0	1	0	1	0
0	1	1	1	1

Чинлик жадвали

$(F(0,0,0)\wedge x^0\wedge y^0\wedge z^0)\vee(F(0,0,1)\wedge x^0\wedge y^0\wedge z^1)\vee(\mathrm{F}(0,1,0)x^0\wedge y^1\wedge z^0)\vee$

$\vee(\mathrm{F}(0,1,1)\wedge x^0\wedge y^1\wedge z^1)\vee(F(1,0,0)\wedge x^1\wedge y^0\wedge z^0)\vee(\mathrm{F}(1,0,1)\vee x^0\wedge y^0\wedge z^1)\vee$

$\vee(\mathrm{F}(1,1,0)\wedge x^1\wedge y^1\wedge z^0)\vee(F(1,1,1)\wedge x^1\wedge y^1\wedge z^1)\cong(\overline{x}\wedge\overline{y}\wedge z)\vee(\overline{x}\wedge y\wedge z)\vee(x\wedge y\wedge z)$

Келтирилган леммага асосланган қуйидаги теорема ўринли.

2.6-теорема. Фикрлар алгебрасидаги ҳар қандай айнан ёлғон бўлмаган формулалар конъюктив бир ҳадларнинг ўрни алмашиши аниқлигида ягона кўринишдаги МДНФ билан ифодаланиши мумкин.

Демак, юқоридаги теоремага асосан берилган формулалар бўйича МДНФни аниқлаш учун бу формулаларнинг ростлик жадвалини тузиб оламиз ва формула бир қийматга эришадиган танланмаларга мос келган конъюктив бир ҳадларнинг дизъюнкциясини оламиз.

Энди фикрлар алгебрасидаги формулаларни мукаммал конъюктив нормал форма (МКНФ)ларда ифодалашни кўриб чиқилади.

Қуйидаги белгилашлар киритилади:

$$x^{\beta}=\begin{cases}\overline{x}, & \text{агар } \beta=1;\\ x, & \text{агар } \beta=0.\end{cases}$$

бу ерда: хусусий ҳолда: $0^0 = 0$, $0^1 = 1$, $1^0 = 1$, $1^1 = 0$. Демак, $x \neq \beta$ бўлганда, $x^\beta = 1$ бўлади, акс ҳолда $x^\beta = 0$. Бундан ташқари қуйидаги белгилаш киритилади:

$$X_1 \wedge X_2 \wedge ... X_n = \bigwedge_{i=1}^{n} X_i$$

Хусусий $\bigwedge_{\beta_1, \beta_2, ..., \beta_n} H(X_1, X_2, ..., X_n, \beta_1, \beta_2, ..., \beta_n)$ ҳолда ифода учун мумкин бўлган $(\beta_1, \beta_2, ..., \beta_n)$ танланмалар бўйича конъюкцияси тушунилади, $\beta_i = \{0, 1\}$, $(i = \overline{1, n})$. Масалан,

$$\bigwedge_{\beta_1, \beta_2} (x_1^{\beta_1} \wedge x_2^{\beta_2}) = (X_1^0 \vee X_1^0) \wedge (X_1^0 \vee X_2^1) \wedge (X_1^1 \vee X_2^0) \wedge (X_1^1 \vee X_2^1) =$$

$$= (X_1 \vee X_2) \vee (X_1 \vee \neg X_2) \vee (\neg X_1 \vee X_2) \vee (\neg X_1 \vee \neg X_2).$$

2.2-лемма. Фикрлар алгебрасидаги ҳар қандай учун ушбу $F(X_1, X_2, ..., X_n) \cong \bigvee_{\beta_1, \beta_2, ..., \beta_n} (F(\beta_1, \beta_2, ..., \beta_n) \wedge x_1^{\beta_1} \wedge x_2^{\beta_2} \wedge ... \wedge x_n^{\beta_n})$. ёйилма ҳар доим ўринли.

Бу леммага қуйидаги асосланган теорема келтирилади.

2.7-теорема. Фикрлар алгебрасидаги ҳар қандай рост бўлмаган, яъни тавтология бўлмаган формулалар дизъюнктив бир ҳадлар ўрни алмашиши аниқлигида ягона кўринишдаги МКНФда ифодаланиши мумкин.

Демак, келтирилган леммаларга асосан берилган формулаларнинг МКНФни аниқлаш учун бу формулаларнинг ростлик жадвали тузиб олинади ва формула ноль қийматга эришадиган танланмаларга мос келган дизъюнктив бир ҳадларнинг конъюкцияси олинади.

§ 2.7. Эллиптик эгри чизиқлар

Ҳозирда эллиптик эгри чизиқларнинг криптография соҳасига тадбиқи кенг қўлланилмоқда. Ушбу параграфда эллиптик эгри чизиқ ва унинг нуқталари ҳақида умумий тушунчалар, ҳамда, уларга боғлиқ бўлган амаллар билан танишилади.

2.7.1. Дастлабки тушунчалар

2.13-таъриф. Бирор K – майдонда олинган эллиптик эгри чизиқ деб, қуйидаги Вейерштрасс тенгламаси деб аталувчи тенглик орқали аниқланувчи

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

эгри чизиққа айтилади, бу ерда $a_1, a_2, a_3, a_4, a_6 \in K$.

Эллиптик эгри чизиқ одатда E ёки E/K билан белгиланади ва эллиптик эгри чизиққа тегишли нуқталар, яъни (2.1) тенглама ечимлари шу эллиптик эгри чизиқнинг **аффин нуқталари** дейилади.

2.14-таъриф. $P(x_0, y_0)\in E$ нуқта эллиптик эгри чизиқнинг силлиқ нуқтаси дейилади, агар

$$f(x_0, y_0) = y_0^2 + a_1x_0y_0 + a_3y_0 - x_0^3 - a_2x_0^2 - a_4x_0 - a_6$$

бўлиб, қуйидаги шартлардан биттаси ўринли бўлса:

$$f'_x(x_0, y_0) \neq 0 \quad \text{ёки} \quad f'_x(x_0, y_0) \neq 0 \qquad (2.2)$$

2.15-таъриф. E/K – эллиптик эгри чизик силлиқ деб аталади, агар унинг ҳар бир аффин нуқтаси силлиқ бўлса.

1-мисол. $y^2 = x^3$ эллиптик эги чизиқ учун (0; 0) нуқта силлиқ нуқта эмас эканлиги кўрсатилсин.

Ечиш.

$$f(x, y) = y^2 - x^3, \quad f'_x = -3x^2, \quad f'_x = -3x^2,$$

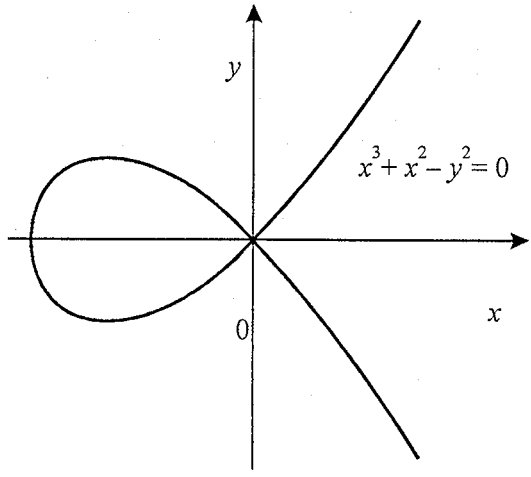
бўлиб, (2.2) шартга нисбатан зиддиятга келинади. Натижада, (0; 0) нуқтанинг ҳақиқатан ҳам силлиқ нуқта бўла олмаслиги келиб чиқади.

2-мисол. $y^2 = x^3 + x^2$ эллиптик эги чизиқ учун (0; 0) нуқта силлиқ нуқта эмас эканлиги кўрсатилсин.

Ечиш. Ҳақиқатан ҳам,

$$f(x, y) = y^2 - x^3 - x^2, \quad f'_x = -3x^2 - 2x, \quad f'_x = 2y$$

бўлиб, (2.2) шартга нисбатан зиддиятга келинади. Натижада, (0; 0) нуқтанинг ҳақиқатан ҳам силлиқ нуқта бўла олмаслиги келиб чиқади:



Қуйида эллиптик эгри чизиқларнинг умумий каноник кўриниши ҳисобланган ушбу

$$y^2=x^3+ax^2+bx+c, \tag{2.3}$$

тенглама билан иш кўрамиз, бу ерда a, b, $c\in Z$ (a, b, c – бутун сонлар) ва кўпҳад $p(x)=x^3+ax^2+bx+c$ каррали илдизга эга эмас деб қаралади.

2.7.2 Эллиптик эгри чизиқларнинг графиклари

Юқорида келтирилган (2.3) кўринишдаги эгри чизиқ графигини чизиш учун

$$y=\sqrt{x^3+ax^2+bx+c}, \tag{2.4}$$

чизиш ва Ox – ўқига нисбатан симметрик акслантириш лозим. Бу (2.4) берилган функция ғрафигини чизиш учун эса квадратсиз ҳолидаги функция

$$z=x^3+ax^2+bx+c$$

графигини чизиб олиш керак бўлади. Функция графигининг Ox – ўқи билан кесишиш нуқталарини

$$x^3+ax^2+bx+c=0$$

тенгламанинг ечимларини топиш орқали аниқланади. Бу тенгламадан,

$$v=x+\frac{a}{3}\left(x=v-\frac{a}{3}\right)$$

алмаштиришдан фойдаланиб,

$$v^3+pv+q=0$$

келтирилган тенглама олинади, бу ерда $p=\frac{3b-a^2}{3}$, $q=\frac{2a^3}{27}-\frac{ab}{3}+c$, $D=\left(\frac{p}{3}\right)^3+\left(\frac{q}{2}\right)^2$ ифода дискриминант деб аталиб, келтирилган тенгламанинг илдизлари сони дискриминант қийматининг ишорасига боғлиқ:

а) $D<0$ бўлса, битта ҳақиқий илдизга эга, яъни функция графиги Ox – ўқи билан битта нуқтада кесишади;

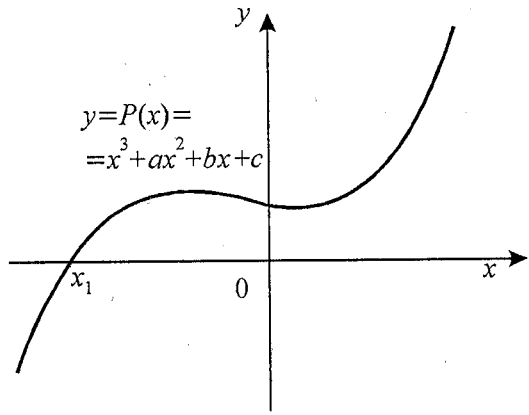
б) $D<0$ бўлса, учта ҳақиқий илдизга эга, яъни функция графиги Ox – ўқи билан учта нуқтада кесишади;

c) $D = 0$ бўлса, учта ҳақиқий илдизга эга бўлиб, уларнинг иккитаси тенг (каррали), яъни функция графиги Ox – ўқи билан иккита нуқтада кесишади.

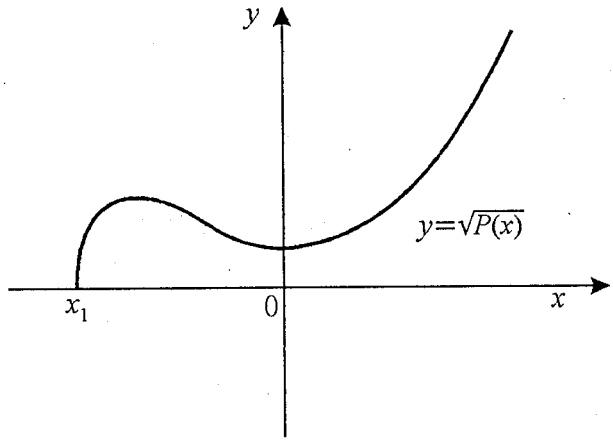
Келтирилган а) ҳол учун

$$z = x^3 + ax^2 + bx + c,$$

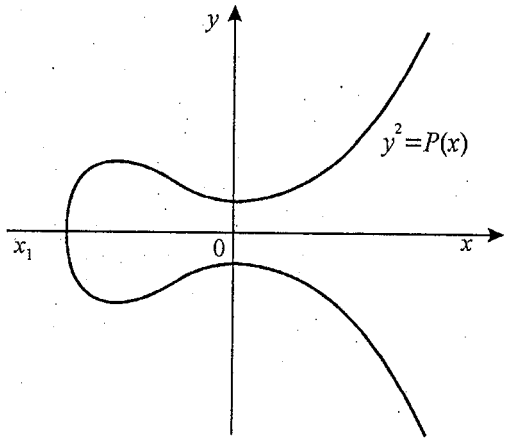
функция графиги қуйидаги кўринишга эга:



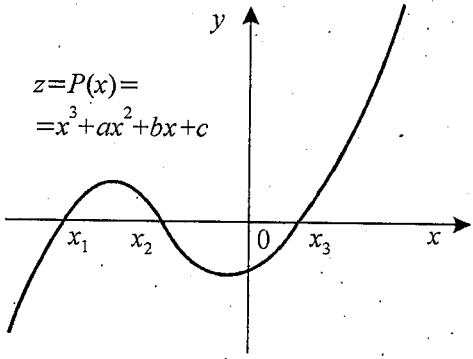
Бу графикдан (2.4) функция графигини олиш учун, квадрат илдиз остидаги ифоданинг манфий бўлмаган қийматлар соҳасига мос келувчи – аниқланиш соҳаси қисмини



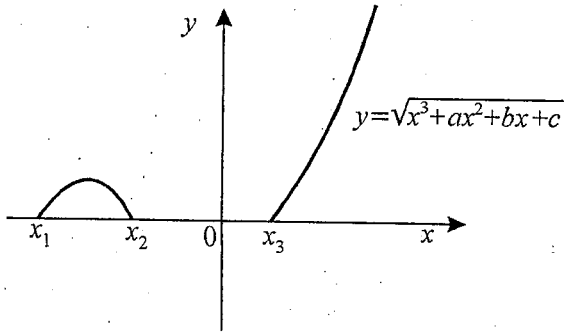
Ox – ўқига нисбатан симметрик кўчирилади, яъни:



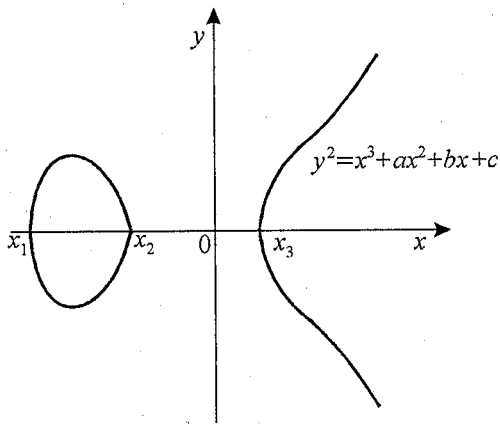
Учта ҳақиқий илдизга эга бўлган б) ҳол учун $z=x^3+ax^2+bx+c$, функция графиги қуйидаги кўринишга эга:



Худди юқоридаги фикр ва мулоҳазаларга кўра, бу графикдан (2.4) функция графигини олиш учун, квадрат илдиз остидаги ифоданинг манфий бўлмаган қийматлар соҳасига мос келувчи – аниқланиш соҳаси қисмини



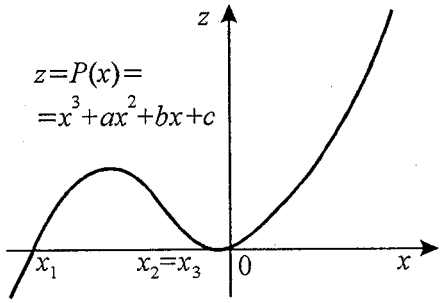
Ox – ўқига нисбатан симметрик кўчирилади, натижада график эллипс ва гиперболадан иборат бўлган иккита қисмлар билан ифодаланади:



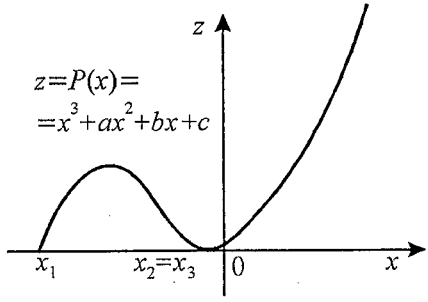
Учта ҳақиқий илдизга эга бўлиб, уларнинг иккитаси тенг (каррали) бўлган с) ҳол учун

$$z=x^3+ax^2+bx+c,$$

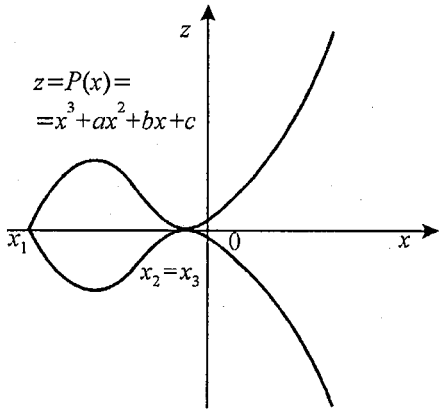
функция графиги қуйидаги кўринишга эга:



Бу графикдан (2.4) функция графигини олиш учун, квадрат илдиз остидаги ифоданинг манфий бўлмаган қийматлар соҳасига мос келувчи – аниқланиш соҳаси қисмини



Ox – ўқига нисбатан симметрик кўчирилади, натижада график умумий нуқтага эга бўлган эллипс ва гиперболадан иборат бўлган иккита қисмлар билан ифодаланади:



Амалда, $z=x^3+ax^2+bx+c$ – эллиптик эгри чизиқ коэффициенти $a=0$ бўлган $y^2=x^3+bx+c$ – эллиптик эгри чизиқнинг келтирилган кўринишидаги ифодасидан ҳамда унинг дискриминанти $D<0$ бўлиб, учта ҳақиқий илдизга эга, яъни функция графиги Ox – ўқи билан учта нуқтада кесишадиган ҳолатидан фойдаланиш, қулай ва самарали татбиқга эга.

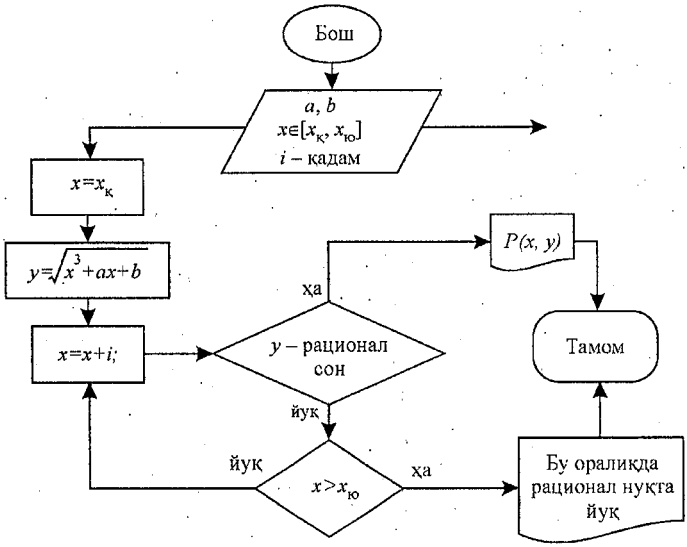
2.7.3 Эллиптик эгри чизиққа тегишли рационал нуқталарни аниқлаш усуллари

Олдиндан шуни айтиш лозимки, ҳозирги кунда $y^2=x^3+bx+c$ тенгламанинг барча рационал ечимларини топиш математикада номаълумлигича қолиб келмокда. Лекин, қуйидаги иккита усулдан фойдаланиб, рационал ечимларни топиш мумкин.

1-усул. Танланган $y^2=x^3+bx+c$ тенгламага x_i қийматлар берилиб, тенгламанинг ўнг томони тўла квадрат ташкил қилишини текширилади. Агар қандайдир x_k қийматда тўла квадрат ташкил қилса, у ҳолда тенгламага тегишли нуқта координаталарини

$$(x_k;\ y_k=\pm\sqrt{x_k^3+axk+b}) \qquad (2.5)$$

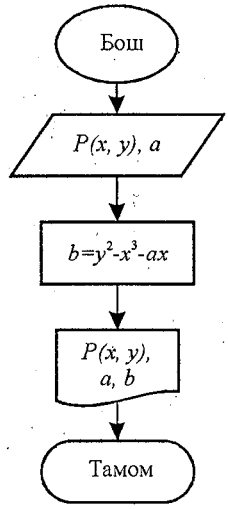
жуфтликлар билан фиксирланади. Бу усул тенглама коэффициентларига бирор шарт аввалдан берилган ҳолда яхши натижа беради. Яъни коэффицентларга мос тенглама қуриб, кейин шу тенгламага тегишли нуқта қидириш усули ҳисобланади. Қуйида ушбу усул алгоритмининг блок схемаси келтирилади:



2-усул. Бу усул топилиши керак бўлган нуқтага бирор шарт қўйилганда фойдаланиладиган усул ҳисобланади. Яъни нуқта координаталари $(x; y)$ ва тенгламанинг битта a – коэффицентини фиксирлаб: $(a; x; y \in R)$,

$$b = y^2 - x^3 - ax \qquad (2.6)$$

формула орқали b – коэффициент ҳисоблаб топилади ва унинг асосида тенглама тузилади. Қуйида ушбу усул алгоритмининг блок схемаси келтирилади:



2.7.4. Эллиптик эгри чизиқларнинг рационал нуқталарини қўшиш

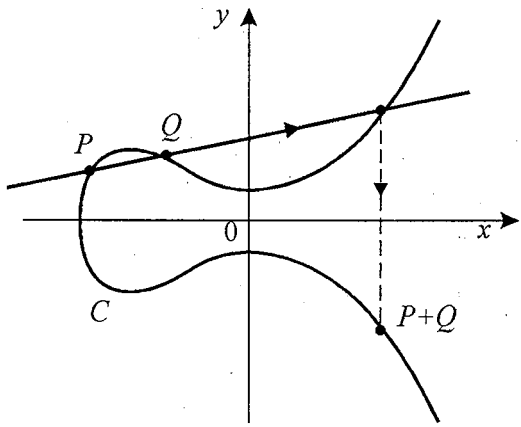
Ушбу

$$E: y = x^3 + ax^2 + bx + c,$$

эллиптик эгри чизиқда $P(x_1, y_1)$, $Q(x_2, y_2)$ нуқталар берилган бўлсин. Бу нуқталар орқали тўғри чизиқ ўтказилади. У ҳолда ўтказилган чизиқ, E – эгри чизиқни учинчи нуқтада кесиб ўтади. Бу $B(x_3, y_3)$ нуқтани Ox – ўқига симметрик кўчирилади ва ҳосил бўлган:

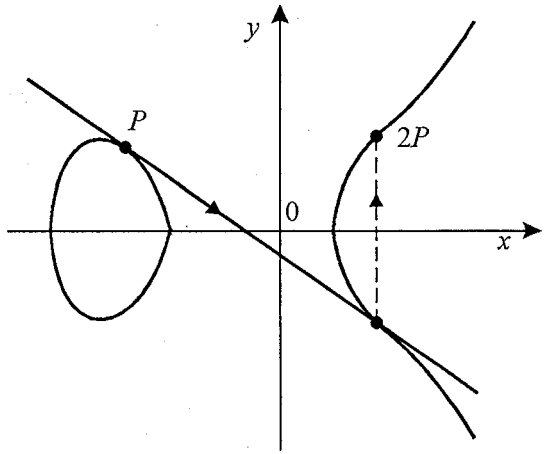
$$B(x_3 - y_3) = P(x_1, y_1) + Q(x_2, y_2)$$

нуқтани, $P(x_1, y_1)$ ва $Q(x_2, y_2)$ нуқталарнинг эллиптик эгри чизиқ устида йиғиндиси деб эълон қилинади:



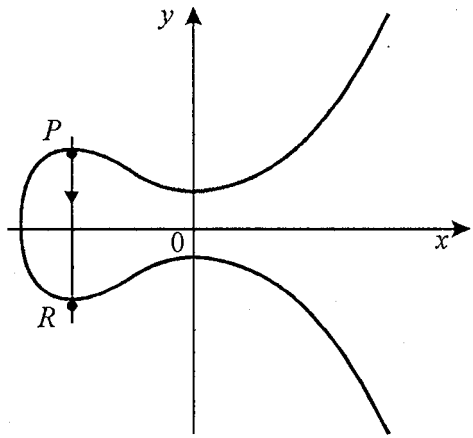
Бу график $x^3 + ax^2 + bx + c = 0$ тенглама битта ечимга эга бўлган ҳол учун келтирилди.

Юқорида эллиптик эгри чизиқда координаталари ҳар-хил бўлган, яъни $P(x_1, y_1) \neq Q(x_2, y_2)) \neq 0$ бўлган нуқталар йиғиндисини $P(x_1, y_1) + Q(x_2, y_2)$ топиш кўриб чиқилди. Энди $P + P = ?$ қандай амалга оширилиши ҳақида тўхталинади. Бунинг учун эллиптик эгри чизиқдаги P – нуқта орқали уринма тўғри чизиқ ўтказилади. Бу уринма эллиптик эгри чизиқ графигидаги иккинчи қисмни (гипербола қисмида) бирор нуқтада кесиб ўтади. Ана шу кесиб ўтган нуқтани Ox – ўқига нисбатан симметрик кўчирилади ва бу нуқта $2P$ деб эълон қилинади:



Сўнгра, $3P$ – ни топиш учун, $3P = P + 2P$, шу каби $4P = P + 3P$, $5P = 4P + P$ ва ҳоказолар амалга оширилади.

Ҳар доим ҳам $P(x_1, y_1)$ ва $Q(x_2, y_2)$ нуқталар орқали ўтувчи тўғри чизиқ эллиптик эгри чизиқни учинчи нуқтада кесиб ўтавермайди. Масалан, $P(x_1, y_1)$ ва $Q(x_1 - y_1)$ нуқталардан ўтувчи тўғри чизиқ – ўқига перпендикуляр бўлиб, у эллиптик эгри чизиқни учинчи нуқтада кесиб ўтмайди:



Бундай ҳолда ўтказилган тўғри чизиқ эллиптик эгри чизиқни чексизликда кесиб ўтади деб қабул қилиниб, чексизликдаги барча нуқталар битта ноль нуқтага бирлаштирилган деб ҳисобланади, яъни чексизликдаги барча нуқталар, эллиптик эгри чизиқ нуқталари устида

аниқланган қўшиш амалига нисбатан, ҳақиқий сонларни қўшишдаги ноль қиймати каби хоссага эга. Ҳақиқатан ҳам, $P(x_1, y_1)$ ва $Q(x_2, y_2)$ нуқталардан ўтувчи тўғри чизиқ Ox – ўқига перпендикуляр бўлиб, у эллиптик эгри чизиқни учинчи нуқтада кесиб ўтмай, чексизликдаги E – нуқтага йўналади. Чексизликдаги E – нуқта билан $P(x_1, y_1)$ – нуқтани қўшишни $E+P(x_1, y_1)$ шаклида кўриб чиқадиган бўлсак, бу нуқталардан ўтувчи тўғри чизиқ Ox – ўқига перпендикуляр бўлиб, эллиптик эгри чизиқни $Q(x_2, y_2)$ – нуқтада кесиб ўтади, сунгра $E+P(x_1, y_1)$ – йиғиндини ифодаловчи нуқтани топиш учун бу $Q(x_1-y_1)$ – нуқтани Ox – ўқига симметрик акслантирилса, $P(x_1, y_1)$ – нуқта билан устмауст тушади, яъни киритилган қўшиш амали қоидасига кўра $E+P(x_1, y_1)=P(x_1, y_1)$ тенглик ўринли бўлади. Бу E – нуқтани Ox – ўқига нисбатан акслантирилса яна қарама – қарши томон чексизлигидаги $(-E)$ – нуқтага йўналади. Аммо, чексизликдаги барча нуқталар битта ноль нуқтага бирлаштирилганда $(-E)+P(x_1, y_1)=P(x_1, y_1)$ тенгликнинг ўринли бўлишига келтирилган фикр мулоҳозалар асосида ҳам ишонч ҳосил қилиш мумкин.

Бевосита ҳисоблашлар билан кўрсатиш мумкинки, эллиптик эгри чизиқ нуқталарини қўшиш амали Абель группаси (19) ни ташкил этади, яъни эллиптик эгри чизиққа тегишли бўлган a, b, c – нуқталар учун:

1) коммутативлик $a+b=b+a$;

2) ассоциативлик $(a+b)+c=(b+c)+a$;

3) ноль элементининг мавжудлиги $a+E=a$;

4) тескари (қарама – қарши) элементнинг мавжудлиги $a+(-a)=E$ каби Абель группасининг аксиомалари ўринлидир.

2.7.5. Эллиптик эгри чизиқнинг нуқталарини қўшиш формулалари

Мазкур қисмда юқорида келтирилган нуқталарни қўшишнинг геометрик маъносидан келиб чиқиб, ушбу амаллар учун математик формулалар келтириб чиқарилган. Кўриб ўтилганларга мувофиқ, агар $P(x_1, y_1)$ ва $Q(x_2, y_2)$ – нуқталар E – эллиптик эгри чизиқда ётса, яъни $P(x_1, y_1)$, $Q(x_2, y_2)\in E$ нуқталар бўлса, унда улар орқали кесувчи тўғри чизиқ ўтказилиб, бу кесувчи тўғри чизиқ E – эллиптик эгри чизиқни бирор учунчи $R(x_3, y_3)$ нуктада кесиб ўтади.

Тасдиқ. Агар $P(x_1, y_1)$, $Q(x_2, y_2)\in E$ нуқталар рационал координатали бўлса, у ҳолда $R(x_3, y_3)$ нуқта координаталари ҳам рационал бўлади.

Исботи. $P(x_1, y_1)$, $Q(x_2, y_2) \in E$ нуқталар орқали ўтувчи тўғри чизиқнинг умумий кўриниши:

$$y = kx + d$$

ифодага эга бўлиб, бу ерда k, d – коэффициентлар $P(x_1, y_1)$ ва $Q(x_2, y_2)$ нуқталарнинг координаталари орқали ифодаланади. $P(x_1, y_1)$, $Q(x_2, y_2)$ – нуқталар $y = kx + d$ чизиққа тегишли. Бундан эса:

$$\begin{cases} y_1 = kx_1 + d, \\ y_2 = kx_2 + d, \end{cases} \qquad y_1 - y_2 = k(x_1 - x_2) \text{ ва } k = \frac{y_1 - y_2}{x_1 - x_2}$$

эканлиги келиб чиқади.

Шунингдек,

$$d = y_1 - kx_1 = y_1 - \left(\frac{y - y_2}{x - x_2} \right) \cdot x_1 = \frac{y_2 x_1 - y_1 x_2}{x_1 - x_2}$$

Шундай қилиб, $y = kx + d$ – тўғри чизиқғи тиклаб олинди. Кейинги қадамда $y = kx + d$ – ифода

$$y^2 = x^3 + ax^2 + bx + c,$$

эллиптик эгри чизиқнинг тенгламасига қўйилади, яъни

$$(kx + ad)^2 = x^3 + ax^2 + bx + c,$$
$$x^3 + (a - k^2)x^2 + (b - 2kd)x + c - d^2 = 0,$$

у ҳолда учинчи тартибли тенглама учун Виет теоремасига кўра:

$$x_1 + x_2 + x_3 = k^2 - a$$

тенглик ўринли бўлиб, бу охирги тенгликда x_1, x_2 – рационал сонлар бўлгани учун, x_3 – ҳам рационал сон бўлади. Худди шунингдек,

$$y_3 = kx_3 + d$$

ифодага кўра y_3 – сонининг ҳам рационал эканлиги келиб чиқади.

Бу келтирилган тасдиқ исботидан эса $P + Q$ йиғинди нуқта координатасини ҳисоблаш формуласини келтириб чиқариш мумкин. $P + Q$ нуқта R – нуқтани Ox – ўқига симметрик кўчиришдан ҳосил бўлар эди. Натижада, йиғинди нуқтанинг координаталари (u, v) деб белгиланса, бу координаталар қуйидаги формулалар орқали топилади:

$$u = k_2 - a - x_1 - x_2,$$
$$v = -ku - d = -(k(u - x_1) + y_1)$$

чунки $u = x_3$, $v = -y_3$. Бу формулада k – коэффициентининг қиймати ўрнига қўйилса ушбу:

$$\begin{cases} v=\frac{y_1-y_2}{x_1-x_2}(-u+x_1)-y_1, \\ u=\frac{(y_1-y_2)^2}{(x_1-x_2)^2}-(a+x_1+x_2), \end{cases} \tag{2.7}$$

тенгликларга эга бўлинади, бу ерда, $x_1 \neq x_2$.

Агар $x_1=x_2$ бўлса, у ҳолда кесувчи тўғри чизиқ ўрнига уринма ўтказилиб, қуйидаги формулалар келтирилиб чиқарилади:

$$\begin{cases} u=-2x_1-a+\frac{(3x_1^2+2ax_1+b)^2}{4y_1^2} \\ v=-y-\frac{3x_1^2+2ax_1+b}{2y_1}(u-x_1). \end{cases} \tag{2.8}$$

Шундай қилиб, ҳеч бўлмаса битта P – рационал нуқта эллиптик эгри чизиқдаги нуқта бўлса, у ҳолда (2.7), (2.8) – формулалар орқали $2P$, $3P$, $4P$,... ва ҳакозаларни топишимиз мумкин бўлади.

Шуни алоҳида таъкидлаш керакки, келтирилган (2.7) ва (2.8) формулалар (2.3) тенгламага нисбатан келтириб чиқарилди. Энди эллиптик эгри чизиқнинг криптографияда кенг қўлланиладиган тенгламаси

$$y=x^3+ax+b$$

учун рационал нуқталарини қўшиш формулалари келтириб ўтилади [24].

$$\begin{cases} u=\frac{(y_1-y_2)^2}{(x_1-x_2)^2}-x_1-x_2, \\ v=-y_1+\frac{y_1-y_2}{x_1-x_2}(x_1-u). \end{cases} \tag{2.9}$$

бу ерда: $x_1=x_2$

Агар $x_1=x_2$ бўлса, у ҳолда

$$\begin{cases} u=\frac{(3x_1^2+a)^2}{4y_1^2}-2x_1, \\ v=-y-\frac{3x_1^2+a}{2y_1}(x_1-u). \end{cases} \tag{2.10}$$

Мисол. Эллиптик эгри чизиқ $y^2=x^3-2$, унга тегишли нуқта $P(3, 5)$ берилган бўлса, бу нуқтанинг йиғиндисини ифодаловчи нуқталар топилсин: $2P=?$, $3P=?$, $4P=?$, $5P=?$

Ечиш. (2.4) формуладан фойдаланилса:

$$y^2 = x^3 + ax^2 + bx + c,\ a = 0,\ b = 0$$

$$u = -2x_1 - a + \frac{(3x_1^2 + 2ax_1 + b)^2}{4y_1^2} = \frac{129}{100}$$

$$v = -y_1 - \frac{3x_1^2 + 2ax_1 + b}{2y_1}(u - x_1) = -\frac{383}{100}$$

демак, $2P = \left(\frac{129}{100};\ -\frac{383}{100}\right)$.

Сўнгра, (2.8) формуладан фойдаланиб: $3P$, $4P$, $5P$ – нуқталар координаталарини ҳисоблаш мумкин, яъни u_n – орқали nP – нуқтанинг биринчи координатасини олсак, у ҳолда:

$$u_1 = 3,\ u_2 = \frac{129}{100},\ u_3 = \frac{164323}{29241}$$

$$u_4 = \frac{2340922881}{58675600},\ u_5 = \frac{307326105747363}{160280942564521}$$

Шу ҳисоблашларни давом эттирсак, u_1 учун 71 хоналидан катта сонлар билан ишлашга тўқнаш келинади [18].

Изоҳ. $y^2 = x^3 + ax^2 + bx + c$ – эллиптик эгри чизиқ рационал нуқталарини топишнинг эффектив усулини аниқлаш ҳозирги кунда сонлар назариясининг муаммоларидан бири ҳисоблансада, эгри чизиққа тегишли битта нуқта топилса, қолганлари (2.7), (2.8) формулалар орқали аниқланади.

Эллиптик эгри чизиқ нуқталарини қўшиш жараёнида қуйидаги иккита ҳолат бўлиши мумкин:

1. Бирор n – қада мда $nP = 0$ тенглик бажарилиши мумкин;

2. $2P$, $3P$, $4P$ ва ҳаказо nP–нуқталар ҳар хил қийматга эга бўлиши мумкин.

2.16-таъриф. Агар барча $m < n$ ҳолатларда $mP \neq 0$ бажарилиб, $nP = 0$ бўлса, у ҳолда P – нуқта n – чекли тартибга эга дейилади [8, 9, 11].

2-боб бўйича хулосалар

Ушбу бобда:

1. Криптологияда қўлланиладиган баъзи математик тушунча ва тасдиқлар, усуллар ҳамда воситалар ҳақида сўз юритилди.

2. Криптографик акслантиришлар ҳоссалари моҳиятларини илмий асослаган ҳолда исботлашда, тўпламнинг таърифи, улар устида бажариладиган амалларга таянилиши нуқтаи назаридан уларнинг баъзи хоссалари баён этилди. Тўплам, тўплам элементлари,

тўпламлар ва улар элементларини акслантиришлари орасида таркибий боғлиқликларни ифодалаш инструменти бўлган графлар ва уларнинг хоссалари келтирилди.

3. Ҳар хил криптоалгоритмларнинг ҳисоблаш мураккабликларини солиштириб ва таҳлил қилиб уларнинг ишончлилик даражаси – бардошлилик даражасини аниқлашда илмий асос бўлувчи мураккаблик назариясининг криптография масалаларини таҳлил қилишда қўлланишлари ёритилди.

4. Асимметрик криптотизимлар негизини ташкил этувчи:

– катта тоқ сонни туб кўпайтувчиларга ажратиш;

– характеристикаси етарли катта бўлган чекли майдонда дискрет логарифмлаш;

– тартиби етарлича даражада катта бўлган чизиқли тенгламалар тизимини чекли майдонларда ечиш;

– характеристикаси етарлича катта бўлган чекли майдонларда эллиптик эгри чизиқнинг рационал нуқталари устида амаллар бажариш;

– мураккабликларни ахборот мухофазасининг криптографик усулари билан ечишда қўлланиш асослари таҳлил қилиниб, амалий таклифлар билдирилди.

Навбатдаги бобда шифрлаш алгоритмларининг асосларини ташкил этувчи акслантиришларни хусусиятларига қараб синфларга ажратиш масаласи кўриб ўтилади.

III БОБ
ШИФРЛАШ АЛГОРИТМЛАРИНИНГ КЛАССИФИКАЦИЯСИ

Шифрлаш алгоритмлари асосларини очиқ маълумотни ифодаловчи алифбо белгиларини ёки белгилар бирикмаларини (уларни *шифр қийматлар* деб ҳам аталади [2]) шифрмаълумотни ифодаловчи алифбо белгиларига ёки белгилар бирикмаларига (уларни *шифрбелгилар* деб ҳам аталади [2]) акслантирувчи математик моделлар ташкил этиши юқорида таъкидланган эди. Шунинг учун ҳам шифрлаш алгоритмларини синфларга ажратишнинг бошланғич босқичи, улар негизидаги акслантириш турлари асосида амалга оширилади. Агар шифрлаш жараёнида очиқ маълумот алифбоси белгилари шифр маълумот алифбоси белгиларига алмаштирилса, бундай акслантиришга асосланган шифрлаш алгоритми *ўрнига қўйиш шифрлаш* синфига киради. Агар шифрлаш жараёнида очиқ маълумот алифбоси белгиларининг ўринлари алмаштирилса, бундай шифрлаш алгоритми *ўрин алмаштириш шифрлаш* синфига киради. Кўриниб турибдики, ўрин алмаштириш шифрлаш алгоритмларида очиқ маълумотни ташкил этувчи алифбо белгиларининг маъноси шифр маълумотда ҳам ўзгармасдан қолади. Аксинча, ўрнига қўйиш шифрлаш алгоритмларида шифрмаълумотни ташкил этувчи алифбо белгилари маъноси очиқ маълумотни ташкил этувчи алифбо белгиларининг маъноси билан бир хил бўлмайди. Шифрлаш жараёнида ўрнига қўйиш ва ўрин алмаштириш акслантиришларининг комбинацияларидан биргаликда фойдаланилса, бундай шифрлаш алгоритми *композицион шифрлаш* туркумига киради. Демак, шифрлаш алгоритмлари акслантириш турларига қараб *ўрнига қўйиш*, *ўрин алмаштириш* ва *композицион шифрлаш* синфига бўлинади.

Умумий тасаввурга кўра, ўрнига қўйиш шифрлаш алгоритмлари акслантиришларининг математик моделлари кўп қийматли функциялар билан ифодаланади. Бундай ҳолат дешифрлаш жараёнида турли ноқулайликларни келтириб чиқаради. Шунинг учун бир қийматли (тескариси мавжуд бўлган) функциялар билан ифодаланувчи акслантиришларни қўллаш қулайлик туғдиради. Шундай қилиб, табиий равишда, ўрнига қўйиш шифрлаш алгоритмлари *бир қийматли* ва *кўп қийматли шифрлаш* синфига бўлинади. Бир қийматли шифрлаш алго-

ритмларида очиқ маълумот алифбоси белгиларининг ҳар бирига шифр маълумот алифбосининг битта белгиси мос қўйилади. Кўп қийматли шифрлаш алгоритмларида очиқ маълумот алифбоси белгиларининг ҳар бирига шифр маълумот алифбосининг иккита ёки ундан ортиқ чекли сондаги белгилари мос қўйилади, яъни очиқ маълумот алифбосининг бирор x_i белгисига шифр маълумот алифбосининг чекли $\{y_{i1}, y_{i2}, ..., y_{it}\}$ тўпламдан олинган бирор y_{ij}, $(1 \le j \le t)$, белгиси мос қўйилади.

Шифрлаш алгоритмлари, калитлардан фойдаланиш турларига кўра, *симметрик* ва *асимметрик* синфларга бўлинади. Агар шифрлаш ва дешифрлаш жараёнлари бир хил калит билан амалга оширилса, бундай шифрлаш алгоритми симметрик шифрлаш алгоритми синфига киради. Агар шифрлаш жараёни бирор k_1 калит билан амалга оширилиб, дешифрлаш жараёни $k_2 \neq k_1$ бўлган k_2 калит билан амалга оширилиб, k_1 калитни билган ҳолда k_2 калитни топиш ечилиши мураккаб бўлган масала билан боғлиқ бўлса, бундай шифрлаш алгоритми асимметрик шифрлаш алгоритми синфига тааллуқли бўлади.

Шифрлаш жараёни очиқ маълумотни ифодаловчи элементар (масалан: бит, ярим байт, беш бит, байт) белгиларни шифрмаълумотни ифодаловчи элементар белгиларга акслантириш асосида амалга оширилса, бундай шифрлаш алгоритми *узлуксиз(оқимли) шифрлаш* синфтуркумига киради.

Шифрлаш жараёни очиқ маълумот алифбоси белгиларининг икки ва ундан ортиқ чекли сондаги бирикмаларини шифрмаълумот алифбоси белгиларининг бирикмаларига акслантиришга асосланган бўлса, бундай шифрлаш алгоритми *блокли шифрлаш* синфига киради.

Шифрлаш жараёнида очиқ маълумот алифбосининг бирор алоҳида олинган a_i белгиси ҳар доим шифрмаълумот алифбосининг бирор фиксирланган b_j белгисига алмаштирилса, бундай шифрлаш алгоритми *бир алифболи шифрлаш* синфига киради. Агар шифрлаш жараёнинг ҳар хил босқичларида очиқ маълумот алифбосининг бирор алоҳида олинган a_i белгиси шифрмаълумот алифбосининг ҳар хил b_j, b_l, ..., b_t белгиларига алмаштирилса, бундай шифрлаш алгоритми *кўп алифболи шифрлаш* синфига киради.

Шифрлаш жарёнида очиқ маълумот алифбоси белгилари ёки алифбо белгилари бирикмалари бирор амал бажариш билан шифрмаълумот алифбоси белгилари ёки уларнинг бирикмаларига алмаштирилса, бундай шифрлаш алгоритми *гаммалаштирилган шифрлаш* синфига киради.

Қуйида шифрлаш алгоритмларининг синфлари алоҳида-алоҳида кўриб чиқилади.

§3.1. Ўрнига қўйиш шифрлаш алгоритмлари

Шифрлаш алгоритмлари очиқ маълумот алифбоси белгиларини шифрмаълумот белгиларига акслантиришдан иборат эканлиги таъкидланди. Акслантиришлар функциялари (калит деб аталувчи номаълум) параметрга боғлиқ ҳолда: жадвал ва аналитик ифода кўринишларида берилиши мумкин. Ўрнига қўйиш шифрлаш алгоритмларининг дастлабки намуналари бўлган тарихий шифрлаш алгоритмларининг деярли ҳаммаси жадвал кўринишида ифодаланади. Улар ҳақидаги тўлиқ маълумотлар адабиётлар рўйхатида келтирилган [1, 2] манбаларда мавжуд. Ўрнига қўйиш шифрлаш алгоритмларининг умумий хусусиятини ҳисобга олиб, бу синфдаги алгоритмларни жадвал кўринишда қуйидагича ифодалаш мумкин:

Очиқ маълумот алифбоси (кириллча белгилар)	А	Б	...	...	Я
Шифрмаълумот алифбоси (иккилик саноқ тизими белгилари)	$x_0^0 x_1^0 ... x_7^0$	$x_0^1 x_1^1 x_2^1 x_3^1 x_4^1$	...	...	$x_0^{31} x_1^{31} x_2^{31} x_3^{31} x_4^{31}$

Кириллча алифбо белгилари сони 32 та, шу 32 та ҳар хил белгиларни битлар билан ифодалаш учун беш бит кифоя, яъни $2^5 = 32$. Келтирилган жадвалдан фойдаланиб, кириллча алифбода ифодаланган очиқ малумот белгиларини уларга мос келувчи иккилик саноқ тизимидаги беш битлик белгиларга алмаштириб шифрмаълумот ҳосил қилинади, яъни $x_i^j \in \{0; 1\}$. Агарда, келтирилган жадвалда очиқ маълумот алифбоси белгиларига шифрмаълумот алифбосининг қандай беш битлик белгилари мос қўйилганлиги номаълум бўлса, бу жадвал калит бўлиб, шифрмаълумотдан очиқ маълумотни тиклаш масаласи мураккаблашади. Бундай шифрлаш жараёнини ифодаловчи алгоритмнинг калитларининг умумий сони 32! бўлиб, ушбу $n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$ – Стирлинг формуласига кўра қуйидагича $32! = \left(\frac{32}{2{,}7}\right)^{32} \sqrt{2 \cdot 3{,}14 \cdot 32} > \left(\frac{32}{4}\right)^{32} \sqrt{2 \cdot 2 \cdot 32} > \left(\frac{32}{4}\right)^{32} \sqrt{2 \cdot 2 \cdot 32} = 2^{96} \cdot 2^3 \cdot \sqrt{2} > 2^{99}$ ҳисобланади. Бундай ҳолат эса калитни билмаган ҳолда дешифлаш жараёнини амалга оширишни жиддий мураккаблаштиради.

Агарда очиқ маълумот компьютердан фойдаланилган ҳолда тузилиб, стандарт ASCII коди алифбоси белгиларидан иборат бўлиб, шифрмаълумот стандарт ASCII коди алифбоси белгиларини бирини бошқаси билан алмаштиришдан иборат бўлган ўрнига қўйиш шифрлаш алгоритмини қўллаш натижасида ҳосил қилинган бўлса, у ҳолда шифрлаш жараёни асосини қуйидаги ўрнига қўйиш алмаштириш жадвали ташкил этади:

Очиқ маълумот алифбоси (стандарт ASCII коди белгилари)	ASCII	ASCI	...	...	$ASCII_{255}$
Шифрмаълумот алифбоси (иккилик саноқ тизими белгилари)	$x_0^0 x_1^0 ... x_7^0$	$x_0^1 x_1^1 x_2^1 x_3^1 x_4^1$	...	...	$x_0^{31} x_1^{31} x_2^{31} x_3^{31} x_4^{31}$

бу ерда: $x_i^j \in \{0; 1\}$ бўлиб, стандарт ASCII коди алифбоси белгиларини 256 та ҳар хил белгиларини битлар билан ифодалаш учун саккиз бит кифоя, яъни $2^8 = 256$.

Бу шифрлаш жараёнини ифодаловчи алгоритм калитларининг умумий сони 256! бўлиб, ушбу $n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$ – Стирлинг формуласига кўра қуйидагича $256! = \left(\frac{256}{2,7}\right)^{256} \sqrt{2 \cdot 3,14 \cdot 256} > \left(\frac{256}{4}\right)^{256} \sqrt{2 \cdot 2 \cdot 256} > > \left(\frac{4 \cdot 2^6}{4}\right)^{256} \sqrt{2 \cdot 2 \cdot 2^8} = 2^{6 \cdot 256} \cdot 2^5 = 2^{1541}$ ҳисобланади. Бундай ҳолат эса калитни билмаган ҳолда дешифрлаш жараёнини амалга оширишни етарли даражада мураккаблаштиради.

Бу юқорида келтирилган жадваллар ўрнига қўйиш шифирлаш алгоритмлари энг умумий кўринишлари моделини ифодалайди.

Ўрнига қўйиш шифрлаш алгоритмлари ҳам, ўз навбатида бир қийматли ва кўп қийматли шифрлаш синфига бўлинади.

Агарда очиқ маълумот компьютердан фойдаланилган ҳолда тузилиб, стандарт ASCII коди алифбоси белгиларини кенгайтирилган компьютер стандарт ANSI коди алифбоси белгиларидан иборат бўлиб, шифрмаълумот стандарт ANSI коди алифбоси белгиларини бирини бошқаси билан алмаштиришдан иборат бўлган ўрнига қўйиш шифрлаш алгоритмини қўллаш натижасида ҳосил қилинган бўлса, у ҳолда шифрлаш жараёни асосини қуйидаги ўрнига қўйиш алмаштириш жадвали ташкил этади:

Очиқ маълумот алифбоси (стандарт ANSI коди белгилари)	$ANSI_0$	$ANSI_1$	...	...	$ANSI_{2^{32}-1}$
Шифрмаълумот алифбоси (иккилик соноқ тизими белгилари)	$x_0^0 x_1^0 \ldots x_{31}^0$	$x_0^1 x_1^1 \ldots x_{31}^1$	...	...	$x_0^{2^{32}-1} x_1^{2^{32}-1} \ldots x_{31}^{2^{32}-1}$

§3.2. Бир қийматли ва кўп қийматли ўрнига қўйиш шифрлаш алгоритмлари

Ўрнига қўйиш шифрлаш алгоритмлари, уларнинг асосини ташкил этувчи акслантиришнинг бир қийматли ёки кўп қийматлилигига кўра, бир қийматли ва кўп қийматли синфларга бўлинади.

Агар ўрнига қўйиш шифрлаш алгоритмида очиқ маълумот алифбоси белгиларининг ҳар бирига шифр маълумот алифбосининг битта белгиси мос қўйилса, бундай алгоритм бир қийматли ўрнига қўйиш шифрлаш алгоритми синфига киради. Очиқ маълумот алифбоси белгилари x_1, x_2, ..., x_N, деб белгиланса, масалан лотин алифбоси белгилари учун $N=26$, кирилл алифбоси белгилари учун $N=32$, стандарт ASCII коди алифбоси белгилари учун $N=256$, ва ҳоказо. Шифрмаълумот алифбоси белгиларини: y_1, y_2, ..., y_M, деб белгиланса, у ҳолда бир қийматли ўрнига қўйиш шифрлаш алгоритмининг умумий ҳолдаги модели жадвал кўринишда қуйидагича ифодаланади:

Очиқ маълумот алифбоси белгилари		x_2	...	...	x_N
Шифрмаълумот алифбоси белгилари		y_{i_2}	...	...	y_{i_N}

бу ерда: $y_{i_j}\{y_1, y_2, ..., y_M\}$.

Мисол сифатида қуйидаги (2×26) – ўлчамли жадвални келтириш мумкин:

Очиқ маълумот алифбоси (лотинча белгилар)	A	Б	...	...	Я
Шифрмаълумот алифбоси (кириллча белгилар)	*, *d*, *n*	W, &, s, g	...	...	14, !, /, j, a

Кўп қийматли шифрлаш алгоритмларида очиқ маълумот алифбоси белгиларининг ҳар бирига шифр маълумот алифбосининг икки ёки ундан ортиқ чекли сондаги белгилари мос қўйилади, яъни очиқ маълумот алифбосининг бирор x_i белгисига шифр маълумот алифбосининг чекли $\{y_{i1}, y_{i2}, ..., y_{it}\} \subset \{y_1, y_2, ..., y_M\}$ тўпламдан олинган бирор y_{ij}, $(1 \le j \le t)$белгиси мос қўйилади. Кўп қийматли ўрнига қўйиш шифр-

лаш алгоритмининг умумий ҳолдаги модели жадвал кўринишида қуйидагича ифодаланади:

Очиқ маълумот алифбоси белгилари	x_1	x_2	...	x_N
Шифрмаълумот алифбоси белгилари	$y_{i_1^1}, y_{i_2^1}, ..., y_{i_t^1}$	$y_{i_1^2}, y_{i_2^2}, ..., y_{i_r^2}$	...	$y_{i_1^N}, y_{i_2^N}, ..., y_{i_p^N}$

бу ерда $y_{i_l^d} \in \{y_1, y_2, ..., y_M\}$

Мисол сифатида қуйидаги (2×32) – ўлчамли жадвални келтириш мумкин:

Очиқ маълумот алифбоси (кириллча белгилар)	А	Б	...	...	Я
Шифрмаълумот алифбоси (стандарт ASCII коди белгилари)	*, *d*, *n*	W, &, s, g	...	...	14, !, /, j, a

Ўрнига қўйиш шифрлаш алгоритмлари, уларнинг асосидаги акслантиришни шифрлаш жараёнида босқичма-босқич ўзгариб туришига кўра бир алифболи ва кўп алифболи шифрлаш синфларига бўлинади.

§ 3.3. Бир алифболи ва кўп алифболи ўрнига қўйиш шифрлаш алгоритмлари

Олдинги параграфларда бир қийматли ва кўп қийматли ўрнига қўйиш шифрлаш алгоритмларининг умумий моделини мос равишда сатрлари сони иккига ва устунлари сони очиқ маълумот алифбоси белгилари сонига тенг бўлган (2×N) – ўлчамли жадваллар билан ифодаланди. Бу жадваллар ўрнига қўйиш акслантиришни ифодалайди ва шифрлаш жараёнида фақат битта жадвалдан фойдаланилади, яъни очиқ маълумот алифбосининг бирор алоҳида олинган белгиси, шифрлаш жараёнида унинг неча марта такрорланишидан қатъий назар, ҳар доим жадвалнинг шифрмаълумот алифбоси белгилари сатридаги мос белгига алмаштирилади. Агарда, ўрнига қўйиш шифрлаш алгоритми акслантиришини асосини ташкил этувчи жадвалнинг шифрмаълумот алифбоси белгилари сатридаги мос белгиларининг жойлашиш тартиби шифрлаш жараёни босқичларида ўзгариб турмаса, бундай алгоритм бир алифболи ўрнига қўйиш шифрлаш алгоритми синфига киради. Аксин-

ча бўлса, яъни шифрмаълумот алифбоси белгилари сатридаги мос белгиларнинг жойлашиш тартиби шифрлаш жараёни босқичларида ўзгариб турса, бундай алгоритм кўп алифболи ўрнига қўйиш шифрлаш алгоритми синфига киради. Бундан келиб чиқадики, кўп алифболи ўрнига қўйиш шифрлаш алгоритмининг моделини ифодаловчи акслантириш жадвалининг сатрлари сони учта ва ундан ортиқ бўлади, уларнинг сони қанча кўп бўлса, мос алгоритмнинг бардошлилиги шунча юқори бўлади. Шундай қилиб, кўп алифболи ўрнига қўйиш шифрлаш алгоритмининг умумий ҳолдаги модели жадвал кўринишида қуйидагича ифодаланади:

Очиқ маълумот алифбоси белгилари	x_1	x_2	...	...	x_N
Шифрмаълумот алифбоси белгилари	$y_{i_1}^1$	$y_{i_2}^1$	...	...	$y_{i_N}^1$
Шифрмаълумот алифбоси белгилари	$y_{i_1}^2$	$y_{i_2}^2$	...	...	$y_{i_N}^2$
...	...	...	...	...	...
...	...	...	...	...	...
Шифрмаълумот алифбоси белгилари	$y_{i_1}^w$	$y_{i_2}^w$	...	...	$y_{i_N}^w$

бу ерда: $y_{i_l} \in \{y_1, y_2, ..., y_M\}$.

Мисол сифатида қуйидаги жадваллар билан ифодаланувчи кўп алифболи ўрнига қўйиш шифрлаш алгоритмларининг моделларини келтириш мумкин:

Очиқ маълумот алифбоси (лотинча белгилар)	A	B	...	...	Z
Шифрмаълумот алифбоси (кириллча белгилар)	И	Л	...	...	У
...	...	...	...	...	...
Шифрмаълумот алифбоси (кириллча белгилар)	Д	Я	...	...	З

ҳамда,

Очиқ маълумот алифбоси (лотинча белгилар)	A	B	...	...	Z
Шифрмаълумот алифбоси (кириллча белгилар)	И	Л	...	...	У

Шифрмаълумот алифбоси (стандарт ASCII коди белгилари)	*	G	…	…	&
…	…	…	…	…	…
Шифрмаълумот алифбоси (кириллча белгилар)	Д	Я	…	…	З

Юқорида ўрнига қўйиш шифрлаш жараёни модели жадваллар билан ифодаланиши мумкин бўлган алгоритмлар ҳақида сўз юритилди. Қуйида ўрнига қўйиш шифрлаш жараёни бирор амални қўллаш билан амалга ошириладиган агоритмлар синфи ҳақида сўз юритилади.

§ 3.4. Гаммалаштириш шифрлаш алгоритмлари

Шифрлаш жараёнида очиқ маълумотни ташкил этувчи мос алифбо белгилари билан «калит» деб аталувчи параметрнинг мос элементлари устида бирор амал бажариш натижасида шифрмаълумотни ташкил этувчи алифбо белгиларига акслантириш амалга оширилса, бундай шифрлаш алгоритми гаммалаштириш шифрлаш алгоритми синфига киради.

Гаммалаштириш билан шифрлаш услубининг моҳияти очиқ маълумотни (ёки шифрмаълумотни) ташкил этувчи алифбо белгилари билан, псевдотасодифий кетма – кетликнинг мос элементлари гаммасини ташкил этувчи элементлар устида бирор амал бажариш билан шифрмаълумот ҳосил қилишдан иборат. Бунда очиқ, шифрланган ва гамма маълумотларнинг алифбо белгилари битта тўпламдан олинган бўлиши зарур. Мисол учун 2 модуль бўйича қўшиш амалидан фойдаланиб, иккилик саноқ тизими алифбосида рақамли кўринишда берилган маълумотни, куйидагича шифрлаш ва дешифрлаш мумкин:

Очиқ матн:	0110011100100011...
Гамма:	1110110010110101...
Гаммалаштирилган матн:	1000101110010110...
Гамма:	1110110010110101...
Очиқ матн:	0110011100100011...

Бу мисолдан кўринадики, дешифрлаш учун калит бўйича (яъни калитни ташкил этувчи гамма элементлари бўйича) шифрмаълумотнинг мос элементларини 2 модуль бўйича қўшишдан фойдаланиб қайта гаммалаштириш кифоя.

Очиқ маълумотни ташкил этувчи алифбо кириллча 32 та белгилардан иборат бўлсин. Уларни А→0, Б→1, В→2, ..., Я→30, *бўшлиқ*

(*пробел*) мослик билан ифодалаб, калит гаммасини ушбу *ТГЯ...ЯЛ... КЗУ* кўринишдаги тасодифий кетма-кетликдан иборат деб олиб, «*гаммалаштириш*» – очиқ маълумотни шифрлашни қуйидагича амалга ошириш мумкин: $(Г+Т) \bmod 32 = (4+19) \bmod 32 = 23 \rightarrow Ц$, $(А+Г) \bmod 32 = (0+4) \bmod 32 = 4 \rightarrow Г$, $(М+З) \bmod 32 = (13+8) \bmod 32 = 21 \rightarrow Ф$, ..., $(А+Я) \bmod 32 = (0+30) \bmod 32 = (0+30) \bmod 32 = 30 \rightarrow Я$, $(Ш+Л) \bmod 32 = (25+12) \bmod 32 = 5 \rightarrow Д$, ..., $(Р+К) \bmod 32 = (17+11) \bmod 32 = 28 \rightarrow Ь$, $(И+З) \bmod 32 = (9+8) \bmod 32 = 17 \rightarrow Р$, $(Ш+У) \bmod 32 = (25+20) \bmod 32 = 13 \rightarrow М$, ва натижада «*ЦГФ...ЯД...ЬРМ*» – шифрмаълумотга эга бўламиз.

Худди юқорида келтирилгани каби, агарда очиқ маълумот компьютердан фойдаланилган ҳолда тузилиб, стандарт ASCII коди алифбоси белгиларидан иборат бўлса, у ҳолда очиқ маълумотнинг X_i – белгисини, унга мос $ASCII_i$ коди қийматига, шифрлаш жараёнида унга мос келувчи калит гаммаси $Г_j$ – элементининг $ASCII_j$ коди қийматини характеристикаси 256 бўлган чекли майдонда қўшиб, натижанинг қийматига тенг бўлган ASCII кодли Y_i белгига алмаштирилади: $(X_i + Г_j) \bmod 256 = Y_i$, ва шифрмаълумот ҳосил қилинади.

Агарда калит гаммаси қайтарилувчи даврга эга бўлган битлардан иборат бўлмаса, олинган шифрмаълумотни очиш етарли даражада қийин бўлади. Бунинг учун калит гаммасини ташкил этувчи элементлар тасодифий ўзгариши керак. Амалда калит гаммасининг даври бутун шифрмаълумот узунлигидан катта бўлиб, очиқ маълумотнинг ҳеч бир қисми маълум бўлмаса, бундай шифрмаълумотга мос келувчи очиқ маълумотни топиш мураккаб бўлади. Бундай ҳолларда шифрмаълумот фақат узунлиги унинг узунлигига тенг бўлган калит гаммасининг мумкин бўлган барча вариантларини танлаш орқали очилади.

Агарда рақиб томонга очиқ маълумотнинг бирор қисми ва унга мос келувчи шифрмаълумот маълум бўлиб қолса, у ҳолда шифрлашнинг гаммалаштириш услуби ўз кучини йўқотади. Чунки, бундай ҳолда, рақиб томон очиқ маълумотнинг маълум бўлган қисми мазмунига кўра бутун шифрмаълумотни очишга ҳаракат қилади. Бундай ҳолатларни ахборот тизими муҳофазаси криптотизимининг амалда қўлланилишида албатта ҳисобга олиш керак.

Юқорида, ўрнига қўйиш шифрлаш алгоритмларини, уларнинг айрим хусусиятларига кўра, синфлаш усуллари кўриб ўтилди. Эндиги навбатда ўрин алмаштириш шифрлаш алгоритмларини синфлаш усуллари кўриб ўтилади.

§3.5. Ўрин алмаштириш шифрлаш алгоритмлари

Ўрин алмаштириш шифрлаш алгоритмларининг асосий хусусияти очиқ маълумот ва шифрмаълумот алифбоси белгиларининг бир хиллигидадир, яъни шифрмаълумотни ташкил этувчи белгиларнинг маъноси мос келувчи очиқ маълумотдаги белгиларнинг маъноси билан бир хил бўлади. Ҳақиқатан ҳам, ўрин алмаштириш шифрлаш жараёнида очиқ маълумот алифбоси белгилари ўринлари алмаштирилиши натижасида шифрмаълумот ҳосил қилинади. Шунинг учун ҳам бундай шифрлаш алгоритмларининг калити узунлиги, умуман олганда, шифрланиши керак бўлган маълумот узунлигига, яъни очиқ маълумот ташкил этувчи алифбо белгиларининг сонига тенг. Бундан ташқари, очиқ маълумотни ташкил этувчи алифбо белгиларининг частотавий хусусиятлари тўлалигича шифрмаълумотга ўтади. Бундай ҳолатлар амалий тадбиқ имкониятларини чеклайди. Шундай бўлсада уларнинг самарали тадбиқларини таъминлашга қаратилган синфлари мавжуд. *Йўналишли ўрин алмаштириш* синфидаги шифрларнинг қўлланилиши амалда кўп тарқалган. Бундай шифрлаш алгоритмлари бирор геометрик шаклга асосланган бўлади. Очиқ маълумот блоклари геометрик шаклга бирор траектория (узлуксиз из) бўйича жойлаштирилади. Шифрмаълумот эса бошқа траектория бўйича ҳосил қилинади. Геометрик шакл сифатида ($n \times m$) ўлчамли жадвал олиб, унинг биринчи сатри бошидан бошлаб очиқ маълумот белгиларини чапдан ўнгга кетма-кет жойлаштириб, сатр тугагач иккинчи сатрга, очиқ маълумот белгиларини ўнгдан чапга кетма-кет жойлаштириб, бу сатр тамом бўлгач, кейинги сатрга олдингисига тескари йўналишда жойлаштирилади ва ҳоказо. Охирида тўлмай қолган сатр ячейкалари очиқ маълумот алифбосидан фарқли бўлган белгилар билан тўлдирилади. Сўнгра, очиқ маълумотни жойлаштириш тартибидан фарқли бўлган бирор йўналиш танлаб олиниб, шу йўналиш асосида шифрмаълумот ҳосил қилинади. Шифрмаълумот ҳосил қилиш йўналиши калит вазифасини бажаради. Мисол сифатида «*йўналишли ўрин алмаштириш шифрлаш алгоритми*» жумласини шифрлашни (4×10) – ўлчамли жадвал асосида қўйидагича амалга ошириш мумкин:

1	2	3	4	5	6	7	8	9	10
Й	*ў*	*н*	*а*	*л*	*и*	*ш*	*л*	*и*	*ў*
и	*т*	*ш*	*а*	*м*	*л*	*а*	*н*	*и*	*р*
р	*и*	*ш*	*ш*	*и*	*ф*	*р*	*л*	*а*	*ш*
...	*и*	*м*	*т*	*и*	*р*	*о*	*г*	*л*	*а*

Бу жадвал устунлари кетма-кетликларини аралаштирган ҳолда (бундай аралаштиришларнинг умумий сони 10! = 3628800 та бўлади), масалан, 72968411035 тартиб (калит) билан

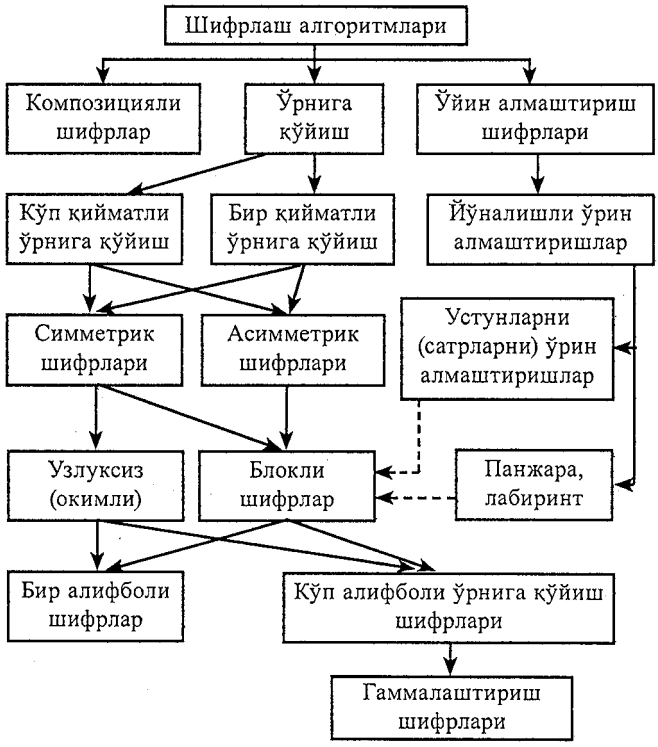
«*шароўтиииналилфрлнлгааштйир.ўршаншшмлмии*»

шифрмаълумотни ҳосил қиламиз. Шифрмаълумотни ҳосил қилиш жараёнини жадвалнинг сатрлари ўринларини ёки ҳар бир устунлари сатрларини алоҳида алмаштиришлар билан яна ҳам мураккаблаштириш мумкин. Сатрлар, устунлар ва алоҳида олинган сатр устунларини ёки алоҳида олинган устун сатрларини шифрлаш жараёни босқичларида ўзгартириб туриш билан яна ҳам мураккаб бўлган шифрлаш алгоритмларини ҳосил қилиш мумкин.

Ўрин алмаштириш шифрлаш алгоритмларининг *панжара* ва *лабиринт* синфлари ҳам мавжуд.

Ўрин алмаштириш шифрлаш алгоритмлари ҳақида тўлароқ маълумотлар адабиётлар рўйхатида келтирилган [2, 13] ўқув қўлланмаларидан топиш мумкин.

Шифрлаш алгоритмларини синфларга (туркумларга) таснифлашнинг (классификациялашнинг) умумий таркибий тузилиши қуйидагича ифодаланиши мумкин [2]:



3-боб бўйича хулосалар

Ушбу бобда:

1. Шифрлаш алгоритмлари криптографик акслантиришларининг хусусиятларига ва қўлланиладиган калитлардан фойдаланиш қоидаларига кўра, уларни синфлаш масаласи кўриб ўтилди.

2. Ўрнига қўйиш шифрлаш алгоритмларининг: бир ва кўп қийматли, бир ва кўп алифболи ҳамда гаммалаштириш синфлари батафсил муҳокама қилинди.

3. Ўрин алмаштириш шифрлаш алгоритмларининг асосий хусусиятлари ёритилди.

4. Шифрлаш алгоритмларининг умумий синфланиш таркибий тузилиши берилди.

Берилган синфлаш жадвалидаги симметрик, асимметрик ва узлуксиз шифрлаш алгоритмлари ҳақида кейинги бобларда алоҳида фикр юритилади.

IV БОБ
СИММЕТРИК БЛОКЛИ ШИФРЛАШ АЛГОРИТМЛАРИНИНГ ХОССАЛАРИ ВА УЛАРНИНГ АХБОРОТ МУҲОФАЗАСИНИ ТАЪМИНЛАШДА ҚЎЛЛАНИЛИШИ

Олдинги бобда ўрнига қўйиш ва ўрин алмаштириш шифрлаш алгоритмларини, уларнинг асосидаги акслантиришларни хусусиятларига кўра синфлашни айрим усуллари кўриб ўтилди.

Ўрнига қўйиш шифрлаш жараёнида очиқ маълумотни ташкил этувчи алифбо белгиларини айрим (алоҳида) олинган ҳолда, шифрмаълумот алифбоининг айрим (алоҳида) олинган белгиларига алмаштириш, ёки, ўрин алмаштириш шифрлаш жараёнида очиқ маълумотни ташкил этувчи алифбо белгиларини айрим (алоҳида) олинган ҳолда ўринларини алмаштириш амалга оширилган бўлсин. Бундай ҳолатда шифрлаш жараёни алгоритмининг криптобардошлилигини ошириш учун калит узунлиги шифрланиши керак бўлган маълумот узунлиги даражасида бўлиши зарур бўлади. Мисол учун, шартли равишда, бирор алифбода тузилган ушбу «$x_1x_2 \ldots x_N$» – очиқ маълумотдан, уни ташкил этувчи алифбо белгиларининг ўринларини алмаштириш натижасида «$x_{i_1}x_{i_2}\ldots x_{i_N}$» – шифрмаълумот ҳосил қилинган бўлса, у ҳолда калитни ифодаловчи $1 \to i_1$, $2 \to i_2$,...$N \to i_N$ – ўрин алмаштиришлар сони билан тенг. Худди шу каби, ўрнига қўйиш шифрлаш алгоритмларидан фойдаланишда очиқ ма'лумот частотавий хусусиятларининг шифрмаллумотга кўчмаслигини та'минлаш учун кўп алифбони шифлаш алгоритмларидан фойдаланилади, бунга эришиш учун эса, юқорида кўрсатилгандек шифрлаш жараёни босқичларида бир хил белгиларни бир хил белгиларга алмаштириш, яъни калит узунлигини ошириш зарурияти туғилади. Шифрланиши керак бўлган маълумот ҳажмининг ортиши б илан, шифрлаш жарёнини амалга оширишда қўлланиладиган алгоритмнинг калити узунлигини мос равишда ортиб бориши, криптобардошлиликни таъминлаш нуқтаи назаридан самарали бўлсада, бундай ҳолат алгоритмларнинг амалда қўлланишлари нуқтаи назаридан: калитларни сақлашда, уларни тарқатишда, аппарат-техник таъминотларни амалга оширишда ва бошқа шу каби ҳолатларда ноқулайликлар туғдиради. Шунинг учун шифрланиши керак бўлган маълумотни, уни ташкил этувчи алифбо белгиларининг маълум бир узунликдаги бирикмалари (блоклари) бирлашмаси (конкатенацияси) кўринишда ифодалаб, ана шу блокларнинг алоҳида-алоҳида самарали ва криптобардошли шифрланишини амалга ошириш масаласи келиб чиқди. Бу масала симметрик блокли шифрлаш алгоритмлари орқали амалга оширилди. Симметрик блокли шифрлаш алгоритмларининг асосини очиқ маълумот блокларини юқори даражада *аралаштириш* ва *тарқатиш* (ёйилиш, таралиш) хоссаларига эга бўлган акслантиришлар ташкил этади [2, 4,

5, 14, 15]. *Самарали аралаштириш* берувчи ($\oplus$, mod2^n, *ўрин алмаштириш жадваллари*, *циклик суришлар* ва ҳокозо) амаллар *коррелияцион иммунстлик* – шифрланиши керак бўлган ёки калит блокларини ташкил этувчи алифбо белгиларидан бирининг ўзгариши, акслантириш натижасида олинган шифрблокни ташкил этувчи алифбо белгиларининг фақат биргина мос белгиси ўзгаришига таъсир қилиб, бошқа қисмига таъсир этмаслигини таъминловчи ўрин алмаштириш шифрлаш акслантиришларидан иборат. *Самарали тарқатиш* берувчи бир алифболи ва кўп алифболи ўрнига қўйиш акслантиришларга асосланган S – блок акслатиришлари *чизиқсизликни* – шифрланиши керк бўлган ёки калит блокларини ташкил этувчи алифбо белгиларидан бирининг ўзгариши, акслантириш натижасида олинган шифрблокни ташкил этувчи алифбо белгиларининг икки ва ундан ортиқ қисмига таъсир этишини таъминловчи ўрнига қўйиш шифрлаш алгоритмлари акслантиришларидан иборат.

Аралаштирувчи акслантиришлар очиқ маълумот ва унга мос келувчи шифрмаълумот блокларининг частотавий (статистик) ва аналитик боғлиқлик хусусиятларини ўрнатишни мураккаблаштирса, тарқатувчи акслантиришлар очиқ маълумот блоки битта белгисининг ўзгаришини мос шифрмаълумот блокининг кўп белгилари ўзгаришига таъсир қилишини ифодалаб, очиқ маълумотнинг частотавий (статистик) хусусиятларини шифрмаълумотга кўчмаслигини таъминлайди.

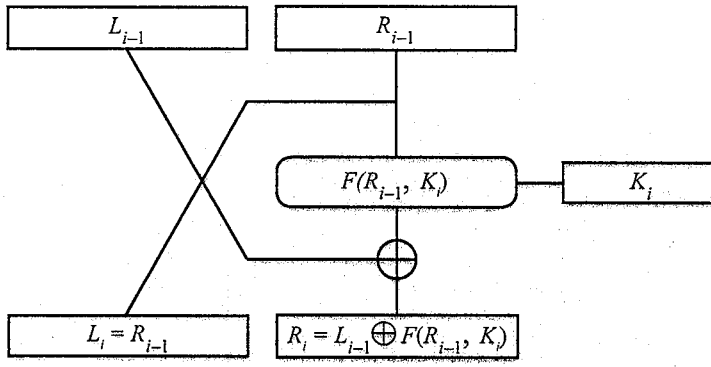
Симметрик блокли шифрлаш алгоритмлари бир нечта босқичлардан (раундлардан) иборат бўлиб, ҳар бир раунд аралаштирувчи ва тарқатувчи акслантиришлардан тузилган. Бундай асосда тузилиш тамоили, ҳар бир раунд шифрлаш жараёнини ҳар хил калитлар билан бир хил турдаги акслантиришларни амалга оширишга, ҳамда, дешифрлаш жараёнини раунд акслантиришлари ва калитларини тескари тартибда қўллашнинг самарали имконини беради. Алгоритм асосини ташкил этувчи, раунд шифрлаш жараёнини амалга оширувчи, аралаштириш ва тарқатиш хусусиятларига эга бўлган функциялар *асосий акслатиришлар* дейилади. *Асосий акслантириш*ларнинг аппарат-техник жиҳатдан қулай қўлланиш модели сифатида тескари боғлиқликка эга бўлган силжитиш регистрларини келтириш мумкин [2, 14, 15]. Бунда тарқатувчи акслантириш тескари боғлиқликни таъминловчи функция билан, аралаштирувчи акслантириш эса, регистрдаги маълумотларни силжитиш билан амалга оширилади.

Шифрланиши керак бўлган маълумот блокини силжитиш регистрларига киритиб (юклаб), регистрдаги маълумотни шартли равишда чап ва ўнг қисмблок векторларига бўлиб, улар устида ҳар хил калитлар билан бир хил турдаги акслантиришларни босқичма-босқич амалга оширишга асосланган – *Фейстел тармоғи* деб аталувчи шифрлаш жараёни функционал қурилмасига асосланган алгоритмлар кенг тарқалган.

§4.1. Фейстел тармоғига асосланган симметрик блокли шифрлаш алгоритмлари ва уларни такомиллаштириш

Фейстел тармоғининг қўлланиши кўпгина симметрик блокли шифрлаш алгоритмларида учрайди. Бу криптоалгоритмларга мисол қилиб **FEAL, LOCI, Khufu, Khafre Blowfish, Lucifer, CAST,** шунингдек, **DES, ГОСТ 28147-89** каби стандарт алгоритмларни келтириш мумкин [2, 14].

Фейстел тармоғи ғояси қуйидагича ифодаланади. Шифрланадиган блок иккита L_0, R_0 қисмларга ажратилади. Фейстел тармоғи i – раунди итератив блокли шифрлаш алмаштириши қуйидаги схема бўйича аниқланади:



4.1-расм. **Фейстел тармоғи i – раунди.**

Бу ерда: $X_i = (L_{i-1}, R_{i-1})$ – i-раунд учун ва қисмларга ажратилган кирувчи маълумот, $Y_i = (L_i, R_i)$ эса X_i ни i–раунд калити K_i билан F акслантириш натижасида ҳосил бўлган шифрмаълумот.

Фейстел тармоғи i – раундининг математик модели қуйидагича ифодаланади:

$$\begin{cases} L_i = R_{i-1}, \\ R_i = L_{i-1} \oplus F(R_{i-1}, K_i). \end{cases} \tag{4.1}$$

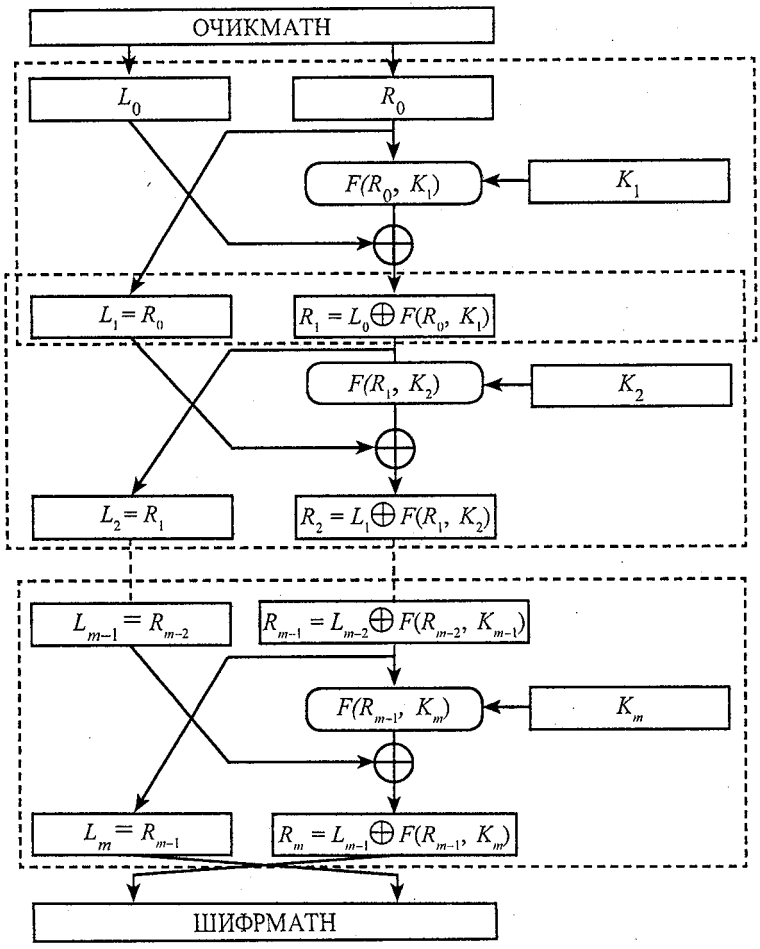
Фейстел тармоғига асосланган алгоритмлар бир неча итерациядан ташкил топган K_i калитларда шифрланадиган функциядан ташкил топади. Ҳар бир i – раунддаги шифрмаълумот $i+1$ – раунд учун кирувчи (очиқ) маълумот ҳисобланади ёки i–раунддаги кирувчи маълумот $i-1$ – раунд учун шифрмаълумот ҳисобланади. K_i раунд калитлари дастлабки K – калитдан алгоритмда кўрсатилган қоида билан ҳосил килинади.

Фейстел тармоғи акслантиришларининг асосий хоссаси шундан иборатки, F – раунд функцияси қайтмас бўлса ҳам, Фейстел тармоғи

бу акслантиришларини қайтариб беради. Ҳақиқатан ҳам, (4.1) ифодада келтирилган i – раунд математик моделида $\oplus$ – иккилик санок тизимида қўшиш амали хоссасидан фойдаланган ҳолда қуйидаги тенгликни олиш мумкин:

$$\begin{cases} R_{i-1} = L_i, \\ L_{i-1} = R_i \oplus F(L_i, K_i). \end{cases} \tag{4.2}$$

Бу охирги тенгликлар тизими Фейстел тармоғи асосида қурилган шифрлаш алгоритмларини дешифрлашининг математик моделини ифодалайди. Умумий ҳолатда m – раундли Фейстел тармоғининг функционал схемаси қуйидагича ифодаланади:

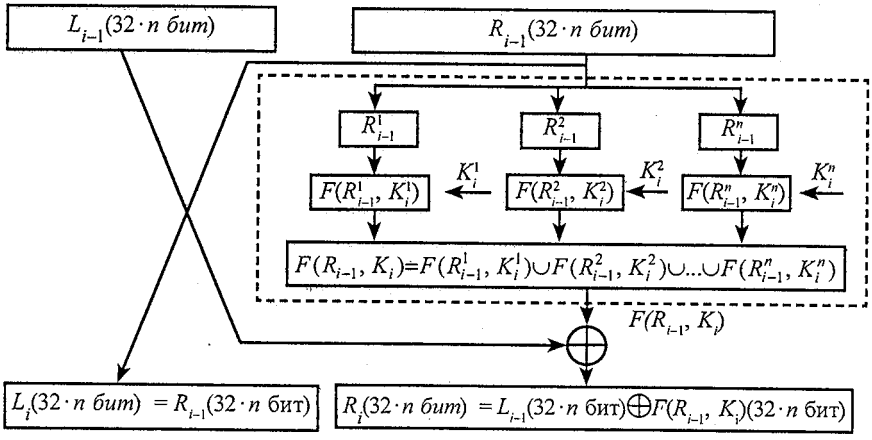


4.2-расм. ***m* – раундли Фейстел тармоғи.**

Фейстел тармоғи асосида қурилган шифрлаш алгоритмларида шифрлаш ва дешифрлаш учун бир хил алгоритмдан фойдаланилиб, фақат раунд калитларининг қўлланилиши тескарисига ўзгаради, яъни дешифрлашда 1-раундда K_m, 2-раундда K_{m-1} ва ҳакозо охирги раундда K_1 ишлатилади. $F(R_{i-1}, K_i)$ функция бир томонлама бўлса ҳам, дешифрлаш натижасида бу функция қайтади.

Ҳисоблаш техникалари қурилмаларининг такомиллашуви натижасида, бугунги кунда стандарт сифатида қўлланилиб келинаётган шифрлаш алгоритмларининг бардошлилиги, уларда қўлланиладиган акслантиришларга боғлиқ бўлмаган ҳолда, улар калитларининг узунликларига нисбатан камаяди. Юқорида санаб ўтилган Фейстел тармоғига асосланган шифрлаш алгоритмлари бугунги кунда ҳам стандат сифатида бенуқсон қўлланилиб келинаётганлиги, бундай алгоритмлар акслантиришларини сақлаб қолган ҳолда, уларнинг калитларини узайтириш масаласининг долзарблиги келиб чиқади. Қуйида Фейстел тармоғига асосланган барча шифрлаш алгоритмларини такомиллаштириш учун умумий бўлган қоида келтирилади.

Бугунги кунда кўплаб амалда қўлланилиб келинаётган компьютерлардаги арифметик амалларни бажарувчи қурилма иккилик санок тизимида 32 разряд билан ифодаланувчи сонлар учун мўлжалланган. Келажакда компьютер фойдаланувчилари учун бундан ҳам катта 64, 128 ва хоказо разрядли сонлар устида арифметик амаллар бажариш имкониятини берувчи тезкор қурилмалар яратилиши табиий ҳол. Шуларни ҳисобга олиб, Фейстел тармоғига асосланган шифрлаш алгоритмларини акслантириш асосларини сақлаб қолган ҳолда, K – калит узунликларини ошириш масаласи ечилади. Мана шундай масалани ечиш учун Фейстел тармоғи қуйидагича такомиллаштирилади:



4.3-расм. **Такомиллашган Фейстел тармоғи *i* – раунди.**

Бу ерда:

1. Шифрланиши керак бўлган очиқ маълумот блоклари узунлиги $64m$ *бит*га тенг.
2. Калит узунлиги $|K| \cdot n$ *бит*га тенг.
3. $K_i = K_i^1 K_i^2 \ldots K_i^n$ – i – раунд қисм калитлари бирлашмаси.
4. Фейстел тармоғи R – ўнг ва L – чап қисмлари узунликлари: $|L| = |R| = 32 \cdot n$ *бит*га тенг.
5. $L_{i-1}(32 \cdot n$ *бит*$)$ – i–раунд чап қисми.
6. $R_{i-1}(32$ *бит*$)$, – i – раунд ўнг қисми.
7. $L_{i-1}^1(32$ *бит*$)$, $L_{i-1}^2(32$ *бит*$)$, ..., $L_{i-1}^n(32$ *бит*$)$ i–раунд чап қисмнинг 32 *бит*лик бўлаклари.
8. $R_{i-1}^1(32$ *бит*$)$, $R_{i-1}^2(32$ *бит*$)$, ..., $R_{i-1}^n(32$ *бит*$)$ i–раунд ўнг қисмнинг 32 *бит*лик бўлаклари.
9. $F(R_{i-1}^1, K_i^1)$, $F(R_{i-1}^2, K_i^2)$, ..., $F(R_{i-1}^n, K_i^n)$ – i–раунд Фейстел функциясининг мос акслантиришлари.

Такомиллашган Фейстел тармоғи i – раунди математик модели қуйидагича ифодаланади:

$$\begin{cases} L_i(32 \cdot n \text{ бит}) = R_{i-1}(32 \cdot n \text{ бит}) \\ R_i(32 \cdot n \text{ бит}) = L_{i-1}(32 \cdot n \text{ бит}) \oplus F(R_{i-1}, K)(32 \cdot n \text{ бит}) \end{cases} \qquad (4.3)$$

Юқорида такомиллашган ва асосий Фейстел тармоғи схемасидан кўриниб турибдики, такомиллашган Фейстел тармоғида такомиллаштириш параметри n га боғлиқ бўлган ҳолда бир нечта $F(R_{i-1}^1, K_i^1)$, $F(R_{i-1}^2, K_i^2)$, ..., $F(R_{i-1}^n, K_i^n)$ Фейстел функциялари учрайди. Бу эса n га боғлиқ ҳолда бир неча Фейстел тармоғига асосланган алгоритмлар функцияларидан ёки бир неча S – блоклардан фойдаланиш имконини беради. Шунингдек, га боғлиқ равишда калит узунликлари ҳам ортиб боради, яъни $n = 1$ да калит узунлиги 256 бит бўлса, $n = 2$ да калит узунлиги 512 ва ҳаказо бўлади. Калит узунлиги ва такомиллаштириш параметри n орасида қуйидагича боғлиқлик ўрнатиш мумкин:

$$l_1 = l \cdot n$$

бу ерда: l – асосий алгоритм калити узунлиги, l_1 – такомиллашган алгоритм калити узунлиги.

Фейстел тармоғига асосланган такомиллашган ва асосий алгоритмларнинг шифрлаш ва дешифрлаш тезлиги тенг, чунки $n = 1$ да алгоритм блок узунлиги 64 га тенг бўлиб, алгоритм тезлиги 20 тактдан иборат бўлса, $n = 2$ да такомиллашган алгоритм блок узунлиги 128 бит бўлиб, тезлиги 40 тактдан иборат бўлади.

Демак, такомиллашган Фейстел тармоғи қуйидаги афзалликларга эга:

1) Такомиллаштириш параметри n га боғлиқ ҳолда шифрлаш алгоритми хоссалари ва бардошлилигини сақлаб қолган ҳолда алгоритм калити узунлигини ошириб бориш имконияти мавжуд. Бу эса, ўз навбатида, ҳисоблаш техникаси қурилмаларининг такомиллашуви натижасида алгоритм калити узунлиги тўлиқ танлаш усулига бардошсиз бўлиб қолишининг олдини олади.

2) Алгорим тезлиги такомиллаштириш параметри n га боғлиқ эмас, яъни Фейстел тармоғига асосланган такомиллашган ва асосий алгоритм тезликлари тенг. Бу хосса ўз навбатида алгоритм тезлигини сақлаб қолган ҳолда такомиллаштириш имкониятини беради.

Қуйида Фейстел тармоғига асосланган симметрик блокли шифрлаш алгоритмларига мисоллар кўриб ўтилади.

§4.2. DES стандарт симметрик блокли шифрлаш алгоритми

DES стандарт шифрлаш алгоритми Америка Қўшма Штатлари (АҚШ) «Миллий Стандартлар Бюроси» томонидан 1977 йилда эълон қилинган. 1980 йилда АҚШнинг «Стандартлар ва Технологиялар Миллий Институти» бу алгоритмни давлат ва савдо-сотиқ молияси соҳасидаги махфий бўлмаган, аммо муҳим бўлган маълумотларни руҳсат этилмаган жисмоний ва юридик шахслардан муҳофаза қилинишида шифрлаш алгоритми сифатида қўллаш стандарти деб қабул қилди.

DES алгоритмида: дастлабки 56 битли калитдан раунд калитларини ҳосил қилишнинг мураккаб эмаслиги, раунд асосий акслантиришларининг аппарат-техник ва дастурий таъминот кўринишларида қўлланилишини таъминлашнинг қулайлиги, ҳамда, улар криптографик хоссаларининг самарадорлиги – криптобардошлилигининг юқорилиги, бу алгоритмнинг асосий хусусиятларини белгилайди.

Шифрлаш жараёни 64 битли очиқ маълумот блокларини алгоритмда берилган IP – жадвал бўйича ўрин алмаштириш, унингнатижасини дастлабки 56 битли калитдан алгоритмда келтирилган жадваллар билан битларнинг ўринларини алмаштириш, циклик суриш ва баъзи битларни йўқотиш акслантиришларидан фойдаланиб ҳосил қилинадиган 48 битли раунд калитлари ҳамда асосий акслантиришлари билан 16 марта шифрлаш, шифрлаш натижаси блоки битларини берилган IP^{-1} – жадвал бўйича ўринларини алмаштиришдан иборат.

Алгоритм акслантиришларини ёритиш учун қуйидаги белгилашлар киритилади:

L_i ва R_i – ҳар бири 32 битли блоклар бўлиб, Фейстел тармоғини чап ва ўнг қисмларини ифодалайди, $i = 0,1, ..., 16$;

$\oplus$ – битлар блоклари векторлари координаталарини бўйича қўшиш;

K_i – 48 битли раунд калитлари;

F – Фейстел тармоғи асосий акслантиришлари функцияси;

IP – ўрин алмаштириш жадвали.

Навбатдаги Т – блокни шифрлаш жараёни бу блок битларини қуйидаги бошланғич IP – ўрин алмаштириш жадвали:

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

асосида акслантириш билан бошланади T – блокнинг 58-бити 1-бит ўрнига, 50-бити 2-бит ўрнига ва ҳоказо қолган битлар ҳам жадвалда кўрсатилган ўринларга ўтказилади. Сўнгра, олинган натижа иккита 32 битлик L_0 ва R_0 қисмларга ажратилиб, 16 раундлик Фейстел тармоғи асосий акслантиришлари функцияси билан ҳар хил 48 битлик калитларда шифрланади. Яъни раунд натижаси, $T_{i-1} = L_{i-1}R_{i-1}(i-1)$ деб белгиланса, у ҳолда, юқорида таъкидланганидек, i – раунд натижаси қуйидаги тенгликлар:

$$\begin{cases} L_i = R_{i-1}, \\ R_i = L_{i-1} \oplus F(R_{i-1}, K_i), \quad i = 1,2, ..., 16; \end{cases}$$

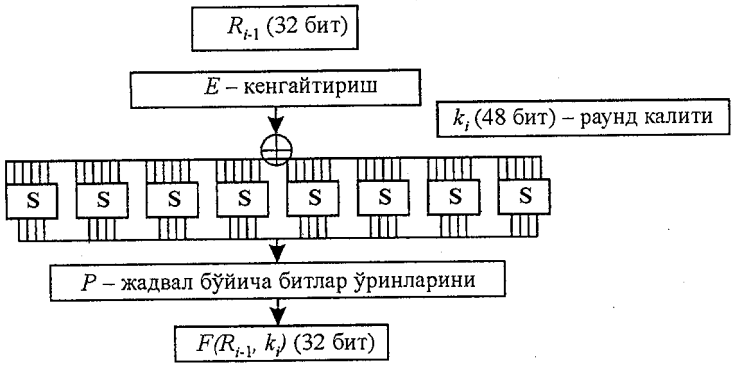
билан топилади. Бу ерда, $F(R_{i-1}, K_i)$ – 32 битли R_{i-1} ва 56 битли дастлабки калитни акслантириш натижасида олинган 48 битли K_i векторларнинг Фейтель тармоғи асосий акслантиришларининг функциясини ифодалайди. Охирги итерация-раунд натижаси $T_{16} = R_{16}L_{16}$ – блок бўлиб, бу блок битлари устида IP –жадвал бўйича IP^{-1} –тескари ўрин алмаштириш акслантириши бажарилади: T_{16} – блокнинг 1-бити 58-бит ўрнига, 2-бити 50-бит ўрнига ва хоказо қолган битлар ҳам жадвалда кўрсатилган ўринларга ўтказилади.

Дешифрлашда шифрлаш жараёнида бажарилган акслантиришлар тескари тартибда бажарилади, бунда ушбу:

$$\begin{cases} L_i = R_{i-1}, \\ R_i = L_{i-1} \oplus F(R_{i-1}, K_i), \quad i = 1,2, ..., 16; \end{cases}$$

муносабатлардан фойдаланилиб, ҳар бири 32 битли бўлган L_{16} ва R_{16} шифрмаълумот блокларини кетма-кет акслантириш, L_0 ва R_0 блоклар олиш, 64 битли блок устида IP^{-1} – акслантиришни бажариш орқали, T – очиқ маълумот блоки олинади.

DES алгоритми Фейстел тармоғи асосий акслантиришларининг $F(R_{i-1}, K_i)$– функциясини ҳисоблаш схемаси қўйидагича:



$E(R_{i-1})$ – кенгайтириш функцияси R_{i-1} – 32 битли блокнинг 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29 ва 32 – битларини икки мартадан такрорлаш, ҳамда, қуйидаги жадвал:

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

бўйича жойлаштириш натижасида ҳосил бўлган 48 битли $E(R_{i-1})$ – блокнинг ҳар бир битини K_i – 48 битли раунд калитининг мос битла-

рига $\oplus$ – XOR (mod2 бўйича) амали билан қўшилиб, натижа саккизта олти битлик B_1, …, B_8, блоклар кўринишида ифодаланади: $E(R_{i-1}) \oplus k_i = B_1 B_2 ... B_8$.

Сўнгра ҳар бир B_j – олти битлик блок S – блокнинг мос жадвалли S_j – блоки орқали акслантирилиб, тўрт битли блок билан алмаштирилади. S_j – блоклар ўлчами 4×16 бўлган саккизта ўзгармас жадвалдан иборат:

(S_1)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	4	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

(S_2)																
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

(S_3)																
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

(S_4)																
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

(S_5)

0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

(S_6)

0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	1	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

(S_7)

0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

(S_8)

0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Ҳар бир $B_j = b_1(j)\ b_2(j)\ \ldots\ b_7(j)\ b_8(j)$, $j = 1, \ldots, 8$; блоклар мос S_j– блокнинг $b_1(j)\ b_8(j)$ – битлар билан аниқланувчи сатри ва $b_2(j) \ldots b_7(j)$ – битлар билан аниқланувчи устунларининг кесишувида жойлашган соннинг иккилик саноқ тизимида ифодаланувчи $z_1(j)\ z_2(j)\ z_3(j)\ z_4(j)$ – тўрт битлик ифодаси билан алмаштирилади. Ҳосил бўлган 32 – битлик $z_1(1)\ z_2(1)\ z_3(1)\ z_4(1)\ \ldots\ z_1(8)\ z_2(8)\ z_3(8)\ z_4(8)$ – блок битлари ўринлари алгоритмда берилган P – жадвал асосида алмаштирилади, унинг кўриниши қуйидагича:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

P –жадвал

Дастлабки 56 битли блок 7 битли қисмблокларга ажратилиб, 8, 16, …, 64 позицияларга, ҳар бир байтдаги бирлар сони тоқ бўладиган қилиб битлар қўшилади. Бу битлар шифрлаш жараёнида ишлатилмайди, улар калитларни узатиш ва сақлашда хатоликларга йўл қўймасликни назорат қилиш учун ишлатилади. Шундай қилиб ҳосил қилинган 64 –битли калит блокининг 56 та битлари ўринлари алгоритмда кўрсатилган жадвал бўйича алмаштирилади. Бундай алмаштириш ифодаси қуйидаги жадвалда келтирилган.

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Бу жадвалнинг юқоридаги тўртта сатри C_0 ва кейинги тўртта сатри D_0 , деб белгиланади. Сўнгра, 56 битли C_0 D_0 – блокдан ушбу: 9, 18, 22, 25, 35, 38, 43, 54 ўринларда турган битлар ўчирилиб, 48 битли блок ҳосил қилиниб, шу блок битлари ўринларини яна алгоритмда берилган қуйидаги:

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

жадвал асосида алмаштирилиб 1-рауннинг k_1 – калити олинади.

C_i ва $Д_i$ жадваллар улардан олдинги C_{i-1} ва $Д_{i-1}$ (i = 1,2,…,16), жадваллардан алгоритмда кўрсатилган:

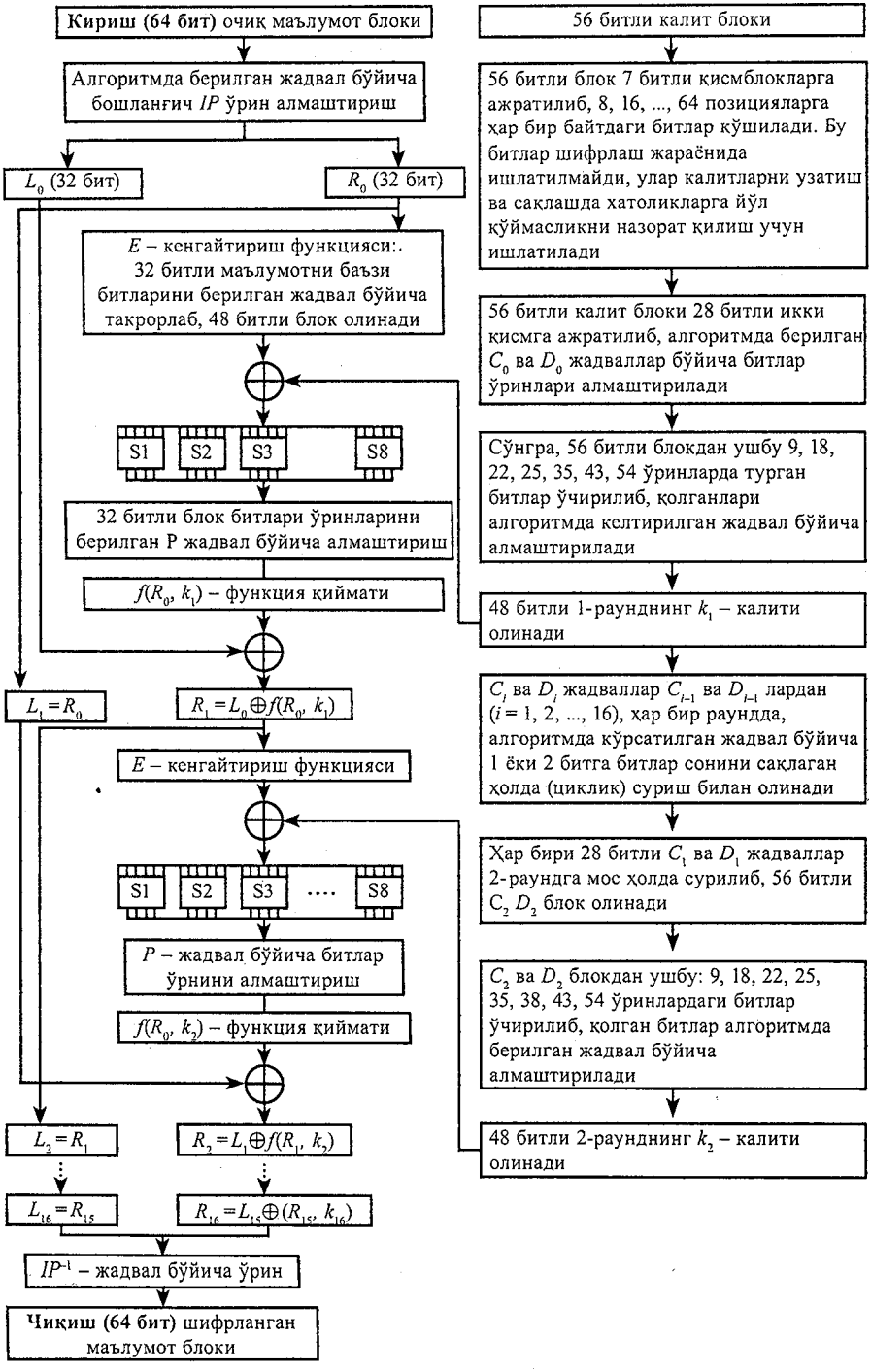
I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Силжитиш сони	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

жадвал бўйича 1 ёки 2 битга циклик суриш, ҳамда, ҳосил бўлган 56 битли блокдан ушбу: 9, 18, 22, 25, 35, 38, 43, 54 ўринларда турган битлар ўчирилиб, 48 битли блок ҳосил қилиниб, шу блок битлари ўринлари яна алгоритмда берилган

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

жадвал асосида алмаштирилиб i – рауннинг k_i – калити ҳосил қилинади.

DES шифрлаш алгоритми АҚШда 1998 йилнинг 31 декабргача стандарт шифрлаш алгоритми деб ҳисобланган. Бу алгоритмда қўлланилган акслантиришлар криптографик нуқтаи назардан бардошли, аммо дастлабки 56 – битли калитнинг узунлиги, бугунги кун ҳисоблаш техника ва технологияларининг ютуқларидан фойдаланилганда, мумкин бўлган барча 2^{56} та калитларни тўла танлаб чиқиш имкониятини сезиларли қисқартиради. Қуйида DES шифрлаш алгоритмининг блок схемаси келтирилган.


Кириш (64 бит) очиқ маълумот блоки
Алгоритмда берилган жадвал бўйича бошланғич IP ўрин алмаштириш
L_0 (32 бит)
R_0 (32 бит)
E – кенгайтириш функцияси: 32 битли маълумотни баъзи битларини берилган жадвал бўйича такрорлаб, 48 битли блок олинади
S1 S2 S3 S8
32 битли блок битлари ўринларини берилган P жадвал бўйича алмаштириш
$f(R_0, k_1)$ – функция қиймати
$L_1 = R_0$
$R_1 = L_0 \oplus f(R_0, k_1)$
E – кенгайтириш функцияси
S1 S2 S3 .... S8
P – жадвал бўйича битлар ўрнини алмаштириш
$f(R_0, k_2)$ – функция қиймати
$L_2 = R_1$
$R_2 = L_1 \oplus f(R_1, k_2)$
$L_{16} = R_{15}$
$R_{16} = L_{15} \oplus (R_{15}, k_{16})$
IP^{-1} – жадвал бўйича ўрин
Чиқиш (64 бит) шифрланган маълумот блоки
56 битли калит блоки
56 битли блок 7 битли қисмблокларга ажратилиб, 8, 16, ..., 64 позицияларга ҳар бир байтдаги битлар қўшилади. Бу битлар шифрлаш жараёнида ишлатилмайди, улар калитларни узатиш ва сақлашда хатоликларга йўл қўймасликни назорат қилиш учун ишлатилади
56 битли калит блоки 28 битли икки қисмга ажратилиб, алгоритмда берилган C_0 ва D_0 жадваллар бўйича битлар ўринлари алмаштирилади
Сўнгра, 56 битли блокдан ушбу 9, 18, 22, 25, 35, 43, 54 ўринларда турган битлар ўчирилиб, қолганлари алгоритмда келтирилган жадвал бўйича алмаштирилади
48 битли 1-раунднинг k_1 – калити олинади
C_i ва D_i жадваллар C_{i-1} ва D_{i-1} лардан (i = 1, 2, ..., 16), ҳар бир раундда, алгоритмда кўрсатилган жадвал бўйича 1 ёки 2 битга битлар сонини сақлаган ҳолда (циклик) суриш билан олинади
Ҳар бири 28 битли C_1 ва D_1 жадваллар 2-раундга мос ҳолда сурилиб, 56 битли C_2 D_2 блок олинади
C_2 ва D_2 блокдан ушбу: 9, 18, 22, 25, 35, 38, 43, 54 ўринлардаги битлар ўчирилиб, қолган битлар алгоритмда берилган жадвал бўйича алмаштирилади
48 битли 2-раунднинг k_2 – калити олинади

АҚШнинг «Стандартлар ва Технологиялар Миллий Институти» томонидан 1997 йилда янги стандарт учун конкурс эълон қилиниб, 2000 йилнинг 2 октябрида унинг ғолиби аниқланди. Бу стандарт шифрлаш алгоритми AES FIPS–197 деб номланиб, унинг асосини Фейстел тармоғи ташкил этмайди. Алгоритм ҳақида кейинги параграфларда сўз юритилади.

§4.3. ГОСТ 28147-89 стандарт симметрик блокли шифрлаш алгоритми

ГОСТ 28147-89 криптоалгоритми ҳозирда Россия Федерацияси давлат стандарт шифрлаш алгоритми ҳисобланади. Бу алгоритм аппарат ва дастурий таъминот учун мўлжалланган бўлиб, ҳимояланадиган маълумотнинг махфийлик даражасига чегаралаш йўқ . Алгоритмнинг калит узунлиги 256 битга шифрлашни 64 бит узунликдаги блокларда амалга оширади ва раундлар сони 32 га тенг. Бирор маълумотни **ГОСТ 28147-89** криптоалгоритми билан шифрлаш учун дастлаб 256 битли калитдан 32 та 32 битли рунд калитлари генерация қилинади ва очиқ маълумот 64 битли X_i, $i = 1, 2, \ldots$ блокларга бўлинади. Бу 64 битли X_i блок 32 битли чап L_i ва ўнг R_i қисмларга бўлинади $X_i = L_i \| R_i$ ва (4.1) формула ёрдамида алмаштирилади, яъни шифрланади.

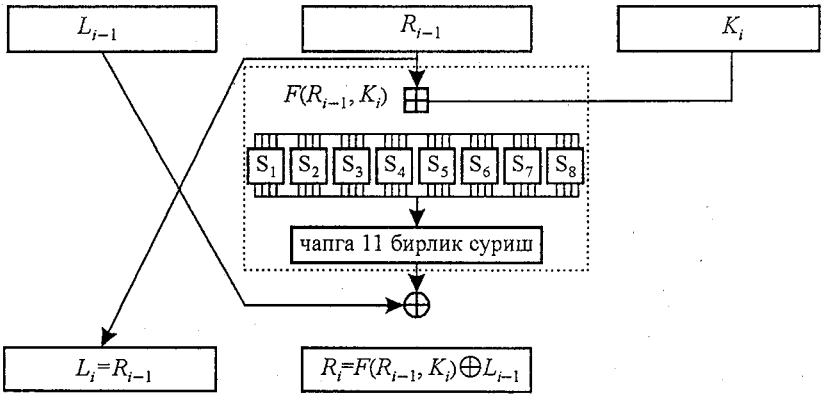
Криптоалгоритмнинг F функцияси қуйидаги амал ва алмаштиришлардан ташкил топган:

1) блокни 32 битли ўнг қисми ва 32 битли раунд калитини $\mathrm{mod}2^{32}$ бўйича қўшиш: $C_i=(R_{i-1}+K_i)\mathrm{mod}2^{32}$;

2) 32 битли C_i натижа саккизта махфий S–блокларда ўрнига қўйиш акслантириши орқали аксланади ;

3) S–блокларда чиқувчи 32 битли блок чапга 11 бирлик циклик сурилади;

Очиқ маълумот 32 раунд итератив шифрлашдан сўнг, чап L_{32} ва ўнг R_{32} қисмлар бирлаштирилади ва $Y_i=R_{32}\|L_{32}$ шифрмаълумот, яъни Y_i шифрмаълумот ҳосил қилинади.



4.4-расм. **ГОСТ 28147-89 криптоалгоритмининг *i* – раунди.**

ГОСТ 28147-89 криптоалгоритмида 8 та S–блоклар қўлланилади, S–блоклар махфий ва бу алгоритмдаги ягона чизиқли бўлмаган акслантиришдир. Бу S–блокларнинг кириш ва чиқиш битлари тўртга тенг бўлиб, нолдан ўн бешгача бўлган сонлар қатнашади. Масалан, биринчи S–блок қуйидагича бўлиши мумкин:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
11	7	13	0	7	9	14	1	6	15	3	4	10	2	5	12

Биринчи S–блокка кирувчи қиймат 4 га тенг бўлса, S–блокдан чиқувчи қиймат 7 га тенг. 4 ва 7 сонлари орасида чизиқли боғланиш мавжуд эмас.

ГОСТ 28147-89 криптоалгоритмида блокнинг 32 битли ўнг қисми R_i 32 битли раунд калити K_{i+1} га mod2^{32} амали бўйича қўшилади. Криптоалгоритм K_{i+1} раунд калити махфийлигини ҳисобга олганда, R_i ёки K_{i+1} ни битта бити ўзгариши натижанинг камида битта битини ўзгаришига олиб келади, шунингдек бу амал умумлашган тўлдириш хусусиятига эга. Бунинг учун калит билан қўшишда ҳосил бўладиган коллизияни кўрсатиш етарли. φ_x – 32 битли блокни шифрлаш акслантириши, φ_x калит акслантириши, F – шифрлаш раунд функцияси, L – чап блок, R – ўнг блок бўлсин. Тўлдириш хусусияти қуйидаги тенглик бўйича аниқланади:

$$\varphi_x(L\oplus F(R+k))=\varphi_x(L\oplus F(\varphi_x(R)+\varphi_x(k))).$$

φ_x ва F акслантиришлар тескариси ҳам ўзига тенглиги хоссасидан фойдаланса, қуйидаги шифр автоморфизмлик шарти ҳосил бўлади:

$$R+k=\varphi_x(R)+\varphi_x(k)(\text{mod } 2^{32})$$

Хусусан бу шартни $\varphi_x(X)=X+2^{31}(\text{mod}2^{32})$ ва $\varphi_k(k)=k+2^{31}(\text{mod}2^{32})$ операторлари ҳам қаноатлантиради. Бу эса катта битнинг инверсияси раунд калити ёки 32 битли блокда пайдо бўлишини билдиради.

Криптоалгоритмнинг S – блоклари махфийлиги алгоритм бардошлилигини янада оширади. Ҳар бир S – блокда 16 та бир хил бўлмаган сонлар қатнашади ва бу сонларни тўлиқ танлаш 16! ни ва саккизта S– блокларни танлаш $C_{16!}^{8}=\frac{(16!)}{8!(16!-8)!}$ ни ташкил этади. Криптоалгоритм дифференциал ва чизиқли криптотаҳлил усулларига ҳам бардошли бўлиб, бу криптотаҳлил усулларини алгоритмга қўллаш учун 2^{64}, яъни мумкин бўлган барча блоклар сонидан ҳам кўп очиқ маълумот талаб этилади. Алгоритмда S–блоклардан сўнг 11 бит чапга циклик суриш акслантириши қўлланилган. 11 сони 33 га каррали, 32 га каррали эмас ва алгоритмга кирувчи блокдаги ҳар бир элемент тўлиқ аралашишини таъминлайди, яъни алгоритмга кирувчи блокнинг бирор x_i элементи, масалан 4–ўринда x_4 бўлса, 1–раунддан сўнг 30–ўринда x_{30} бўлиб, 2–раунддан сўнг x_{17}–ўринда бўлиб ва ҳоказо ўринларда учрайди. Ҳеч қачон бирор раунддан сўнг жойлашган ўрни қайтарилмайди, яъни $x_i \neq x_j$, $i \neq j$, $1 \le i, j \le 32$.Бу стандарт шифрлаш алгоритми ҳозирги кунда ҳам кўп жиҳатдан бошқа алгоритмларга нисбатан ўзининг криптографик самарадорлигини сақлаб келмоқда.

Мисол тариқасида бугунги кунда ҳам ўзининг самарадорлиги ва бардошлилиги билан ишончли криптографик хусусиятларга эга бўлган Фейстел тармоғига асосланган **ГОСТ 28147-89** *стандарт симметрик блокли шифрлаш алгоритми такомиллашган вариантини келтирамиз.*

1. Калит узунлиги: $|k|=256 \cdot n$ *бит* $=32 \cdot n$ *байт.*

2. Блок узунлиги: $|B|=64 \cdot n$ *бит* $=8 \cdot n$ *байт.*

3. R – ўнг ва L – чап қисмлари узунликлари: $|L|=|R|=32 \cdot n$ *бит* $=4 \cdot n$ *байт.*

4. Такомиллашган алгоритм калити: $k(256 \cdot n)=k_1 \ldots k_{8 \cdot 32 \cdot n}=k_1 \ldots k_{32 \cdot n}$ $k_{32 \cdot n+1} \ldots k_{2 \cdot 32 \cdot n} k_{2 \cdot 32 \cdot n+1} \ldots k_{3 \cdot 32 \cdot n} k_{3 \cdot 32 \cdot n+1} \ldots k_{4 \cdot 32 \cdot n} k_{4 \cdot 32 \cdot n+1} \ldots k_{5 \cdot 32 \cdot n} k_{5 \cdot 32 \cdot n+1} \ldots k_{6 \cdot 32 \cdot n}$ $k_{6 \cdot 32 \cdot n+1} \ldots k_{7 \cdot 32 \cdot n} k_{7 \cdot 32 \cdot n+1} \ldots k_{8 \cdot 32 \cdot n}$.

5. Раунд калитлари: $k(i) = k_{(i-1) \cdot 32 \cdot n+1} \ldots k_{i \cdot 32n}$, $i=1, \ldots 8$.

6. S – блоклар сони: $8 \cdot n$ (дона.)

7. Раунд калитлари узунлиги: $|k_{raund}(i)|=32 \cdot n$ *бит.*

Қуйида такомиллашган **ГОСТ 28147-89** криптоалгоритмининг **C#** тилида тузилган дастурий таъминоти келтирилади:

```
public byte n; // такомиллаштириш параметри
sbyte []GOST={4, 10, 9, 2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3,
              14, 11, 4, 12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9,
              5, 8, 1, 13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11,
              7, 13, 10, 1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3, // n=1 да
              6, 12, 7, 1,5, 15, 13, 8, 4, 10, 9, 14, 0,3, 11, 2, // ишлатиладиган
              4, 11, 10, 0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14, // S–блоклар
              13, 11, 4, 1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12,
              1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12,
              //==================================
              10, 9, 2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3, 4,
              11, 4, 12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9, 14,
              8, 1, 13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11, 5, // n=2 да қўшимча
              13, 10, 1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3, 7, // ишлатиладиган
              12, 7, 1, 5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2, 6, // S–блоклар
              11, 10, 0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14, 4,
              11, 4, 1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12, 13,
              15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12, 1,
              //==================================
              9, 2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3, 4, 10,
              4, 12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9, 14, 11,
              1, 13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11, 5, 8, // n=3 да қўшимча
              10, 1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3, 7, 13, // ишлатиладиган
              7, 1, 5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2, 6, 12, // S–блоклар
              10, 0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14, 4, 11,
              4, 1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12, 13, 11,
              13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12, 1, 15,
              //==================================
              2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3, 4, 10, 9,
              12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9, 14, 11, 4,
              13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11, 5, 8, 1, // n=4 да қўшимча
              1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3, 7, 13, 10, // ишлатиладиган
              1, 5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2, 6, 12, 7, // S–блокла
              0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14, 4, 11, 10,
              1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12, 13, 11, 4,
              0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12, 1, 15, 13,
              //==================================
              13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3, 4, 10, 9, 2,
              6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9, 14, 11, 4, 12,
              10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11, 5, 8, 1, 13, // n=5 да қўшимча
```

0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3, 7, 13, 10, 1, // ишлатиладиган
5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2, 6, 12, 7, 1, // *S*–блоклар
7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14, 4, 11, 10, 0,
3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12, 13, 11, 4, 1,
5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12, 1, 15, 13, 0,
//==========

8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3, 4, 10, 9, 2, 13,
13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9, 14, 11, 4, 12, 6,
3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11, 5, 8, 1, 13, 10, // *n*=6 да қўшимча
8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3, 7, 13, 10, 1,0, // ишлатиладиган
15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2, 6, 12, 7, 1, 5, // *S*–блоклар
2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14, 4, 11, 10, 0, 7,
15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12, 13, 11, 4, 1, 3,
7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12, 1, 15, 13, 0, 5,
//==========

0, 14, 6, 11, 1, 12, 7, 15, 5, 3, 4, 10, 9, 2, 13, 8,
15, 10, 2, 3, 8, 1, 0, 7, 5, 9, 14, 11, 4, 12, 6, 13,
4, 2, 14, 15, 12, 7, 6, 0, 9, 11, 5, 8, 1, 13, 10, 3, // *n*=7 да қўшимча
9, 15, 14, 4, 6, 12, 11, 2, 5, 3, 7, 13, 10, 1, 0, 8, // ишлатиладиган
13, 8, 4, 10, 9, 14, 0, 3, 11, 2, 6, 12, 7, 1, 5, 15, // *S*–блоклар
1, 13, 3, 6, 8, 5, 9, 12, 15, 14, 4, 11, 10, 0, 7, 2,
5, 9, 0, 10, 14, 7, 6, 8, 2, 12, 13, 11, 4, 1, 3, 15,
10, 4, 9, 2, 3, 14, 6, 11, 8, 12, 1, 15, 13, 0, 5, 7,
//==========

14, 6, 11, 1, 12, 7, 15, 5, 3, 4, 10, 9, 2, 13, 8, 0,
10, 2, 3, 8, 1, 0, 7, 5, 9, 14, 11, 4, 12, 6, 13, 15,
2, 14, 15, 12, 7, 6, 0, 9, 11, 5, 8, 1, 13, 10, 3, 4, // *n*=8 да қўшимча
15, 14, 4, 6, 12, 11, 2, 5, 3, 7, 13, 10, 1, 0, 8, 9, // ишлатиладиган
8, 4, 10, 9, 14, 0, 3, 11, 2, 6, 12, 7, 1, 5, 15, 13, // *S*–блоклар
13, 3, 6, 8, 5, 9, 12, 15, 14, 4, 11, 10, 0, 7, 2, 1,
9, 0, 10, 14, 7, 6, 8, 2, 12, 13, 11, 4, 1, 3, 15, 5,
4, 9, 2, 3, 14, 6, 11, 8, 12, 1, 15, 13, 0, 5, 7, 10
//==========

6, 11, 1, 12, 7, 15, 5 ,3, 4, 10, 9, 2, 13, 8, 0, 14,
2, 3, 8, 1, 0, 7, 5, 9, 14, 11, 4, 12, 6, 13, 15, 10,
14, 15, 12, 7, 6, 0, 9, 11, 5, 8, 1, 13, 10, 3, 4, 2, // *n*=9 да қўшимча
14, 4, 6, 12, 11, 2, 5, 3, 7, 13, 10, 1, 0, 8, 9, 15, // ишлатиладиган
4, 10, 9, 14, 0, 3, 11, 2, 6, 12, 7, 1, 5, 15, 13, 8, // *S*–блоклар
3, 6, 8, 5, 9, 12, 15, 14, 4, 11, 10, 0, 7, 2, 1, 13,
0, 10, 14, 7, 6, 8, 2, 12, 13, 11, 4, 1, 3, 15, 5, 9,

```
                9, 2, 3, 14, 6, 11, 8, 12, 1, 15, 13, 0, 5, 7, 10, 4,
                //==================================
                11, 1, 12, 7, 15, 5, 3, 4, 10, 9, 2, 13, 8, 0, 14, 6,
                3, 8, 1, 0, 7, 5, 9, 14, 11, 4, 12, 6, 13, 15, 10, 2,
                15, 12, 7, 6, 0, 9, 11, 5, 8, 1, 13, 10, 3, 4, 2, 14, //
n=10 да қўшимча
4, 6, 12, 11, 2, 5, 3, 7, 13, 10, 1, 0, 8, 9, 15, 14, // ишлатиладиган
10, 9, 14, 0, 3, 11, 2, 6, 12, 7, 1, 5, 15, 13, 8, 4, //
S – блоклар 6, 8, 5, 9, 12, 15, 14, 4, 11, 10, 0, 7, 2, 1, 13, 3,
10, 14, 7, 6, 8, 2, 12, 13, 11, 4, 1, 3, 15, 5, 9, 0,
2, 3, 14, 6, 11, 8, 12, 1, 15, 13, 0, 5, 7, 10, 4, 9};
Sbyte
[]Sikl = {0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 7, 6,
5, 4, 3, 2, 1, 0};
public string KeyFileName;
public string _InFile;
public string _OutFile;
public uint [ , ]KeyRound;
public uint [] Round Key=new uint [8];
                //==================================
public void KeyGen()
KeyRound = new uint [8, n];
uint []_KeyRound = new uint [8*n];
byte [] Key_Byte = new byte [32*n];
FileStream Key_Fin = new
FileStream (KeyFileName, FileMode. Open, FileAccess. Read);
Key_Fin. Read (Key_Byte, 0,32*n);
for(int i=0; i<8*n; i++)
_KeyRound [i]=BitConverter. ToUInt32(Key_Byte,i*4);
for(int j=0; j<n; j++)
for(int i=0; i<8; i++)
KeyRound[i,j] = _KeyRound [(j*8)+i];
                //==================================
public void CRYPTFILE()
uint [] nL=new uint[n];
uint [] nR=new uint[n];
FileStream instream=new
FileStream (_InFile, FileMode. Open, FileAccess. Read, FileShare. Read);
FileStream outstream=new
FileStream (_OutFile, FileMode. Create,
```

```
FileAccess. Write, FileShare. None);
long nl =instream. Length;
long BufSize;
if((nl% (n*8))==0) BufSize=nl;
else BufSize=((nl/(n*8))+1)*(n*8);
byte [] Buffer=new byte [BufSize];
instream. Read (Buffer, 0, (int) nl);
for (long i=nl; i<BufSize; i++)
Buffer [i]=0;
byte [] cl =new byte [4];
long [] nF=new long [n];
long []_nF=new long [n];
long N=4294967296;
for(long i=0; i<(long) (BufSize/ (8*n)); i++)
int size=0;
for (int j=0; j<n; j++)
nL[j]=BitConverter. ToUInt32(Buffer, ((int)i*8*n)+(size))
size+=4;
for(int j=0; j<n; j++)
nR[j] = BitConverter. ToUInt32(Buffer, ((int)i*8*n)+(size)) size+ = 4;
for(int nSikl=0; nSikl<32; nSikl++)
for(int k=0; k<n; k++)
nF[k]=nR[k]+KeyRound [Sikl [nSikl], k];
if(nF[k]>=N) nF[k]-=N;
int ns=0;
int nVal;
uint nFS=0;
for (int ni=k*8; ni<(k+1)*8; ni++)
nVal=int)(nF [k] >>ns )&0×0F;
nVal=GOST [ni*16+nVal];
nFS<<=4;
nFS|=(uint)nVal;
ns+=4;
nF[k]=nFS;
nF[k]=nF[k];
nF[k]<<=11;
nF[k]&=0×FFFFF800;
nF[k]>>=21;
nF[k]|=_nF[k]& 0×7FF;
nF[k]^=nL[k];
```

```
nL[k]=nR[k];
nR[k]=(uint)n F[k];
for(int k1=0; k1<n; k1++)
uint a=nL [k1];
nL[k1]=nR[k1];
nR[k1]=a;
for(int j=0; j<n; j++)
c1=BitConverter. GetBytes (nL [j]);
outstream. Write(c1, 0, 4);
for(int j=0; j<n; j++)
c1=BitConverter. GetBytes (nR [j]);
outstream. Write(c1, 0, 4);
                //==================================
public void DECRYPTFILE()
uint [] nL=new uint [n];
uint [] nR=new uint [n];
FileStream instream=new
FileStream(_InFile, FileMode. Open, FileAccess. Read, FileShare. Read);
FileStream outstream=new
FileStream(_OutFile, FileMode. Create,
FileAccess. Write, FileShare. None);
long n1=instream. Length;
long BufSize;
if ((n1%(n*8))==0) BufSize=n1;
else BufSize=((n1/(n*8))+1)*(n*8);
byte [] Buffer=new byte [BufSize];
instream. Read (Buffer, 0, (int) n1);
for(long i=n1; i<BufSize; i++)
Buffer [i]=0;
byte []c1=new byte [4];
long []nF=new long [n];
long []_nF=new long [n];
long N=4294967296;
for(long i=0; i<(long) (BufSize/(8*n)); i++)
int size=0;
for(int j=0; j<n; j++)
L [j]=BitConverter. ToUInt32 (Buffer, ((int)i*8*n)+(size));
size+=4;
for(int j=0; j<n; j++)
nR [j]=BitConverter. ToUInt32(Buffer, ((int)i*8*n)+(size))
```

```
size+=4;
for(int nSikl=0; nSikl<32; nSikl++)
for(int k=0; k<n; k++)
nF[k]=nR[k]+KeyRound [Sikl[31–nSikl], k];
if(nF[k]>=N) nF[k]–=N;
int ns=0;
int nVal;
uint nFS=0;
for (int ni=k*8; ni<(k+1)*8; ni++)
nVal=(int) (nF [k]>>ns )&0×0F;
nVal=GOST[ni*16+nVal];
nFS<<=4;
nFS|=(uint)nVal;
ns+=4;
nF[k]=nFS;
_nF[k]=nF [k];
nF[k]<<=11;
nF[k]&=0×FFFFF800;
_nF[k]>>=21;
nF[k]|=_nF [k]& 0×7FF;
nF[k]^=nL [k];
nL[k]=nR [k];
nR[k]=(uint) nF [k];
for(int k1=0; k1<n; k1++)
uint a=nL [k1];
nL[k1]=nR[k1];
nR [k1] = a;
for (int j=0; j<n; j++)
c1=BitConverter. GetBytes (nL [j]);
outstream. Write (c1,0,4);
for (int j=0; j<n; j++)
c1=BitConverter. GetBytes (nR [j]);
outstream. Write (c1, 0, 4);
```

Маълумки **ГОСТ 28147-89** да 8 та S – блоклар ишлатилган. Келтирилган дастурий таъминотда $n=1$ дан $n=0$ гача такомиллаштириш имконияти яратилган. Келтирилган такомиллаштирилган шифрлаш алгоритми учун S – блоклар стандарт шифрлаш алгоритми S – блокларини $n=1$ да чапга 1 циклик суришдан, $n=2$ да эса чапга 2 хонага циклик суришдан ва ҳакозо $n=0$ да чапга 10 хонага суриш билан ҳосил қилинган.

Такомиллашган ва такомиллашмаган **ГОСТ 28147-89** шифрлаш алгоритмлари дастурий таъминотини киёслаш учун такомиллашмаган **ГОСТ 28147-89** шифрлаш алгортми дастурий таъминотини ҳам келтириш лозим топилди.

```
sbyte []GOST = {4, 10, 9, 2, 13, 8, 0, 14, 6, 11, 1, 12, 7, 15, 5, 3,
                14, 11, 4, 12, 6, 13, 15, 10, 2, 3, 8, 1, 0, 7, 5, 9,
                5, 8, 1, 13, 10, 3, 4, 2, 14, 15, 12, 7, 6, 0, 9, 11,
                7, 13, 10, 1, 0, 8, 9, 15, 14, 4, 6, 12, 11, 2, 5, 3,
                6, 12, 7, 1, 5, 15, 13, 8, 4, 10, 9, 14, 0, 3, 11, 2,
                4, 11, 10, 0, 7, 2, 1, 13, 3, 6, 8, 5, 9, 12, 15, 14,
                13, 11, 4, 1, 3, 15, 5, 9, 0, 10, 14, 7, 6, 8, 2, 12,
                1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12};
byte []KeySikle={0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5,
6, 7, 7, 6, 5, 4, 3, 2, 1, 0};
uint [] Round=new uint [8];
public string KeyFileName;
                //==================================
public void GenKey(){
byte []Key_Byte=new byte[32];
FileStream Key_Fin=new
FileStream (KeyFileName, FileMode. Open, FileAccess. Read);
Key_Fin. Read(Key_Byte,0,32);
for(int i=0; i<8; i++)
KeyRound [i]=BitConverter. ToUInt32(Key_Byte, i*4);
                //==================================
public ulong Crypt (ulong nSource)
uint nR = (uint) nSource & 0×00000000FFFFFFFF;
uint nL = (uint)((nSource & 0×FFFFFFFF00000000) >> 32)&
0×00000000FFFFFFFF;
long nF,nF1, N=4294967296;
for (int nSikl=0; nSikl<32; nSikl++)
nF=nR+KeyRound [KeySikle [nSikl]];
if (nF>=N) nF-=N;
int ns=0;
int nVal;
uint nFS=0;
for (int ni=0; ni<8; ni++)
nVal=(int) (nF>>ns )&0×0F;
nVal=GOST [ni*16+nVal];
nFS<<=4;
```

```
nFS|=(uint)nVal;
ns+=4;
nF=nFS;
nF1=nF;
nF<<=11;
nF&=0×FFFFF800;
nF1>>=21;
nF|=nF1& 0×7FF;
nF^=nL;
nL=nR;
nR=(uint)nF;
ulong nLR=0;
nLR=nR;
nLR<<=32;
nLR&=0×FFFFFFFF00000000;
nLR|=nL;
return nLR;
public ulong Decrypt(ulong nSource)
uint nR=(uint) nSource & 0×00000000FFFFFFFF;
uint nL=(uint) ((nSource & 0×FFFFFFFF00000000) >> 32)&
0×00000000FFFFFFFF;
long nF,nF1,N=4294967296;
for(int nSikl=0; nSikl<32; nSikl++)
nF=nR+KeyRound [KeySikle [31-nSikl]];
if(nF>=N) nF-=N;
int ns=0;
int nVal;
uint nFS=0;
for (int ni=0;ni<8;ni++)
nVal=(int) (nF>>ns )&0×0F;
nVal=GOST [ni*16+nVal];
nFS<<=4;
nFS|=(uint) nVal;
ns+=4;
nF=nFS;
nF1=nF;
nF<<=11;
nF&=0×FFFFF800;
nF1>>=21;
nF| nF1& 0×7FF;
nF^=nL;
nL R;
```

```
nR=(uint)nF;
ulong nLR=0;
nLR=nR;
nLR<<=32;
nLR&= ×FFFFFFFF00000000;
nLR|=nL;
return nLR;
```

Юқорида такомиллашган ва такомиллашмаган Фейстел тармоғи функционал схемаси ва дастурий таъминотидан кўриниб турибдики, такомиллашган Фейстел тармоғида такомиллаштириш параметри n га боғлиқ бўлган ҳолда бир нечта ,..., Фейстел функциялари учрайди. Бу эса n га боғлиқ ҳолда бир нечта $F(R_{i-1}^1, K_i^1)$, $F(R_{i-1}^2, K_i^2)$,...,$F(R_{i-1}^n, K_i^n)$ Фейстел тармоғига асосланган алгоритмлар функцияларидан ёки бир нечта S–блоклардан фойдаланиш имконини беради. Шунингдек, n га боғлиқ равишда калит узунликлари ҳам ортиб боради, яъни $n=1$ да калит узунлиги 256 бит бўлса, $n=$ да калит узунлиги 512 ва ҳоказо бўлади. Калит узунлиги ва такомиллаштириш параметри n орасида қуйидагича боғлиқлик ўрнатиш мумкин:

$$l_1 = l \cdot n$$

бу ерда: l – такомиллашмаган алгоритм калит узунлиги, l_1 – такомиллашган алгоритм калит узунлиги.

Фейстел тармоғига асосланган такомиллашган ва такомиллашмаган алгоритмларнинг шифрлаш ва дешифрлаш тезлиги тенг, чунки $n=1$ да алгоритм блок узунлиги 64 га тенг бўлиб, алгоритм тезлиги 20 тактдан иборат бўлса, $n=2$ да такомиллашган алгоритм блок узунлиги 128 бит бўлиб, тезлиги 40 тактдан иборат бўлади.

Демак, такомиллашган Фейстел тармоғи қуйидаги афзалликларга эга:

1) Такомиллаштириш параметри n га боғлиқ бўлган ҳолда шифрлаш алгоритми хоссалари ва бардошлилигини сақлаб қолган ҳолда алгоритм калит узунлигини ошириб бориш имконияти мавжуд. Бу эса ўз навбатида ҳисоблаш техникаси қурилмаларининг такомиллашуви натижасида алгоритм калит узунлиги тўлиқ танлаш усулига бардошсиз бўлиб қолишини олдини олади.

2) Алгорим тезлиги такомиллаштириш параметри n га боғлиқ эмас, яъни Фейстел тармоғига асосланган такомиллашган ва такомиллашмаган алгоритм тезликлари тенг. Бу хосса, ўз навбатида, алгоритм тезлигини сақлаб қолган ҳолда такомиллаштириш имкониятини беради.

§4.4. Blowfish симметрик блокли шифрлаш алгоритми

Blowfish алгоритми Б. Шнайер томонидан ишлаб чиқилган бўлиб, 1993 йилда эълон қилинган. Б. Шнайер алгоритмга қуйидагиларни асос қилиб олинган.

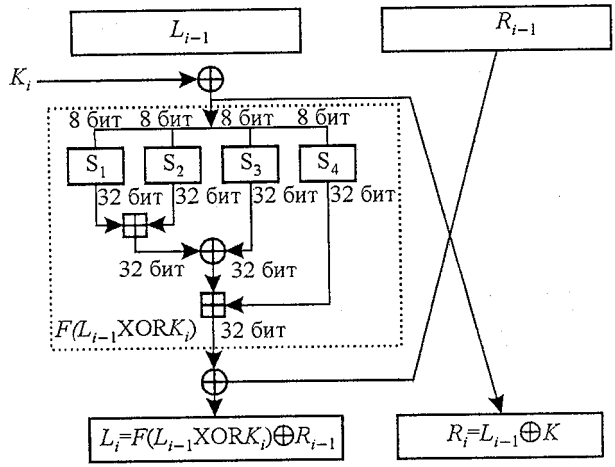
Тезлик. 32 разрядли микропроцессорларда **Blowfish** алгоритми 26 тактда шифрлайди.

Компактлилик. **Blowfish** компьютер хотирасида энг камида 5 килобайт жой эгаллайди.

Оддийлик. **Blowfish** алгоритмида қўшиш, XOR, ва таблицадан ўрин алмаштириш қўлланилади. Бу эса таҳлилни осонлаштиради.

Ўзгарувчан бардошлилик. **Blowfish** алгоритми калити узунлиги ўзгарувчан ва 448 битгача етиш мумкин.

Бу алгоритм Фейстел тармоғига асосланган бўлиб, блоки узунлиги 64 битга, раундлар сони 16 га ва раунд калитлари узунлиги эса 32 битга тенг. **Blowfish** шифрлаш алгоритми i-раунди қуйидаги функционал схемада келтирилган:



4.5-расм. **Blowfish криптоалгоритмининг i–раунди**

Blowfish криптоалгоритми раундилари F функцияси қуйидагича:

1) 32 битли L_{i-1} чап қисм 8 битли (a, b, c) қисмларга бўлинади.

2) (a, b, c) 8 битли қисмлар блоклар ёрдамида ўрин алмаштирилади. Ўрин алмаштириш формуласи қуйидагича:

$$F(L_{i-1}) = (((S_{1,a}+S_{2,b})\text{mod}2^{32})\text{XOR}\ S_{3,c})+S_{4,d})\text{mod}2^{32}.$$

Бу ерда ҳар бир S–блок 256 та 32 битли элементга эга, яъни 8 битли сонни 32 битли сонга акслантириб беради. Масалан, биринчи S–блок қуйидагича:

0×d1310ba6	0×98dfb5ac	0×2ffd72db	0×d01adfb7	0×b8e1afed	0×6a267e96	0×ba7c9045	0×f12c7f99
0×24a19947	0×b3916cf7	0×0801f2e2	0×858efc16	0×636920d8	0×71574e69	0×a458fea3	0×f4933d7e
0×0d95748f	0×728eb658	0×718bcd58	0×82154aee	0×7b54a41d	0×c25a59b5	0×9c30d539	0×2af26013
0×c5d1b023	0×286085f0	0×ca417918	0×b8db38ef	0×8e79dcb0	0×603a180e	0×6c9e0e8b	0×b01e8a3e
0×d71577c1	0×bd314b27	0×78af2fda	0×55605c60	0×e65525f3	0×aa55ab94	0×57489862	0×63e81440
0×55ca396a	0×2aab10b6	0×b4cc5c34	0×1141e8ce	0×a15486af	0×7c72e993	0×b3ee1411	0×636fbc2a
0×2ba9c55d	0×741831f6	0×ce5c3e16	0×9b87931e	0×afd6ba33	0×6c24cf5c	0×7a325381	0×28958677
0×3b8f4898	0×6b4bb9af	0×c4bfe81b	0×66282193	0×61d809cc	0×fb21a991	0×487cac60	0×5dec8032
0×ef845d5d	0×e98575b1	0×dc262302	0×eb651b88	0×23893e81	0×d396acc5	0×0f6d6ff3	0×83f44239
0×2e0b4482	0×a4842004	0×69c8f04a	0×9e1f9b5e	0×21c66842	0×f6e96c9a	0×670c9c61	0×abd388f0
0×6a51a0d2	0×d8542f68	0×960fa728	0×ab5133a3	0×6eef0b6c	0×137a3be4	0×ba3bf050	0×7efb2a98
0×a1f1651d	0×39af0176	0×66ca593e	0×82430e88	0×8cee8619	0×456f9fb4	0×7d84a5c3	0×3b8b5ebe
0×e06f75d8	0×85c12073	0×401a449f	0×56c16aa6	0×4ed3aa62	0×363f7706	0×1bfedf72	0×429b023d
0×37d0d724	0×d00a1248	0×db0fead3	0×49f1c09b	0×075372c9	0×80991b7b	0×25d479d8	0×f6e8def7
0×e3fe501a	0×b6794c3b	0×976ce0bd	0×04c006ba	0×c1a94fb6	0×409f60c4	0×5e5c9ec2	0×196a2463
0×68fb6faf	0×3e6c53b5	0×1339b2eb	0×3b52ec6f	0×6dfc511f	0×9b30952c	0×cc814544	0×af5ebd09
0×bee3d004	0×de334afd	0×660f2807	0×192e4bb3	0×c0cba857	0×45c8740f	0×d20b5f39	0×b9d3fbdb
0×5579c0bd	0×1a60320a	0×d6a100c6	0×402c7279	0×679f25fe	0×fb1fa3cc	0×8ea5e9f8	0×db3222f8
0×3c7516df	0×fd616b15	0×2f501ec8	0×ad0552ab	0×323db5fa	0×fd238760	0×53317b48	0×3e00df82
0×9e5c57bb	0×ca6f8ca0	0×1a87562e	0×df1769db	0×d542a8f6	0×287effc3	0×ac6732c6	0×8c4f5573
0×695b27b0	0×bbca58c8	0×e1ffa35d	0×b8f011a0	0×10fa3d98	0×fd2183b8	0×4afcb56c	0×2dd1d35b
0×9a53e479	0×b6f84565	0×d28e49bc	0×4bfb9790	0×e1ddf2da	0×a4cb7e33	0×62fb1341	0×cee4c6e8
0×ef20cada	0×36774c01	0×d07e9efe	0×2bf11fb4	0×95dbda4d	0×ae909198	0×eaad8e71	0×6b93d5a0
0×d08ed1d0	0×afc725e0	0×8e3c5b2f	0×8e7594b7	0×8ff6e2fb	0×f2122b64	0×8888b812	0×900df01c
0×4fad5ea0	0×688fc31c	0×d1cff191	0×b3a8c1ad	0×2f2f2218	0×be0e1777	0×ea752dfe	0×8b021fa1
0×e5a0cc0f	0×b56f74e8	0×18acf3d6	0×ce89e299	0×b4a84fe0	0×fd13e0b7	0×7cc43b81	0×d2ada8d9
0×165fa266	0×80957705	0×93cc7314	0×211a1477	0×e6ad2065	0×77b5fa86	0×c75442f5	0×fb9d35cf
0×ebcdaf0c	0×7b3e89a0	0×d6411bd3	0×ae1e7e49	0×00250e2d	0×2071b35e	0×226800bb	0×57b8e0af
0×2464369b	0×f009b91e	0×5563911d	0×59dfa6aa	0×78c14389	0×d95a537f	0×207d5ba2	0×02e5b9c5
0×83260376	0×6295cfa9	0×11c81968	0×4e734a41	0×b3472dca	0×7b14a94a	0×1b510052	0×9a532915
0×d60f573f	0×bc9bc6e4	0×2b60a476	0×81e67400	0×08ba6fb5	0×571be91f	0×f296ec6b	0×2a0dd915
0×b6636521	0×e7b9f9b6	0×ff34052e	0×c5855664	0×53b02d5d	0×a99f8fa1	0×08ba4799	0×6e85076a

Бу S–блокка кирувчи қиймат 85 га тенг бўлсин, у ҳолда S–блокдан чиқувчи қиймат сифатида S–блокнинг 85-ўрнида жойлашган $0\times21c66842_{16}=566650946_{10}=00100001110001100110100001000010_2$ олинади.

16 раунддан сўнг ҳосил бўлган чап ва ўнг ярим блоклар K_{18} ва K_{17} раунд калитлари билан XOR амали бўйича қўшилади.

Қуйида **Blowfish** криптоалгоритмида қўлланилган акслантиришлар ва уларнинг бардошлиликка таъсири кўриб ўтилади.

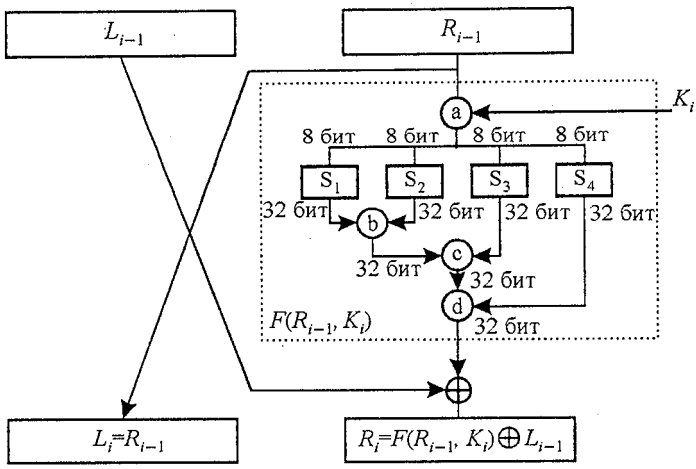
Бу алгоритм асосида ётадиган тамойил қўлланиш конструкциясининг оддийлигидадир. Криптоалгоритмда қўлланилган: **MOV**, **ADD** ва **XOR** амаллари замонавий микропроцессор архитектураларида эффектив қўлланилади. **Blowfish** алгоритмида чап ярим блокнинг ўзгариши ўнг ярим блок ўзгаришига олиб келади. Бундан ташқари калитнинг ўзгариши ҳар бир раунддан сўнг чап ва ўнг ярим блокларга ҳам таъсир этади. Алгоритмда тўртта S–блок қўлланилган бўлиб, бу S–блокларнинг кириши 8 битга ва чиқиши 32 битга тенг, яъни ҳар бир S–блокда 32 разрадли 256 та бир-бирини такрорламайдиган сонлар қатнашган. S–блоклар махфий бўлган ҳолатда битта S–блокни тўлиқ танлаш 256! ни ташкил этади. Бу алгоритм S–блоклар маълум бўлган ҳолатда калитлар генерациясида иштирок этган P–массивни дифференциал криптотаҳлил усулида 2^{8r+1} та танлаб олинган очиқ маълумот ва шифрмаълумот ёрдамида калитни топиш мумкин. $r=16$ да бу қиймат 2^{129} га тенг. Акслантиришларнинг келтирилган хусусиятлари алгоритмнинг самарадорлигини оширишга қаратилган.

§4.5. CAST симметрик блокли шифрлаш алгоритми

CAST криптоалгоритми Карлайл Эдамс ва Стаффорд Таварес томонидан яратилган бўлиб, бу алгоритмнинг криптопараметрлари ўзгарувчандир.

***S*–блоклар конструкцияси. CAST** криптоалгоритмининг S–блоклари ўлчами $m\times n$ бўлиб, m битли сонни n битга акслантиради, яъни S–блокнинг кириши m ва чиқиши n га тенг. **CAST** криптоалгоритмида $m=8$, $n=32$ га тенг бўлиб, 8 битли сонни 32 битга айлантириб беради.

Раунд функцияси. Қуйида **CAST** криптоалгоритмининг i–раундининг функционал схемаси келтирилган.



4.6-расм. CAST криптоалгоритмининг i–раунд.

Бу функционал схемадан кўриниб турибдики, 32 битли блок K_i калит билан *'a'* амал бўйича комбинация қилинади. Натижаси 8 битли тўртта қисмга бўлинади ва S-блокларга жойлаштирилади. S–блокларда ўрин алмаштиришлардан сўнг, S_1 ва S_2 блок натижаси *'b'* амал бўйича комбинация қилинади. S_3 блок натижаси *'b'* амал бўйича комбинация қийматига *'c'* амал бўйича комбинация қилинади. S_4 блок натижаси *'c'* амал бўйича комбинация қийматига *'d'* амал бўйича комбинация қилинади. Раунд функцияси натижаси сифатида охирги қиймат олинади. **CAST** криптоалгоритми раунд функцияси амаллари *'a'*, *'b'*, *'c'*, *'d'* сифатида иккилик саноқ тизимида қўшиш, $2^{32}-1$ ёки $2^{32}+1$ модуль бўйича кўпайтириш амалларини олиш мумкин. Қуйида **CAST-128** криптоалгоритмларини кўриб ўтамиз.

CAST-128 криптоалгоритми

CAST-128 криптоалгоритми **CAST** криптоалгоритми асосида яратилган бўлиб, калит узунлиги 128 бит, блок узунлиги 64 бит ва раундлар сони 16 га тенг. Бу алгоритмда *'a'*, *'b'*, *'c'*, *'d'* амаллар сифатида иккилик саноқ тизимида қўшиш (XOR), модуль 2^{32} бўйича қўшиш ва айириш амаллари ишлатилган. Раунд функциясида бу амалларнинг ҳар хил учта варианти қўлланилади. S–блоклари ўлчами 8×32га тенг. Раунд функциясида иккита калит фойдаланилади. 32 битли K_m калит R_i блокка бирор амал бўйича комбинация қилинса, 5 битли K_r калит натижани суриш учун хизмат қилади. Бу алгоритмда қуйидаги раунд функцияларидан фойдаланилади.

Биринчи тип раунд функцияси:

$$I = ((K_{mi} + R_{i-1}) <<< K_{Ri}),\ i = \overline{1...32}$$

$$F = ((S_1[I_1] \wedge S_2[I_2]) - S_3[I_3]) + S_4[I_4];$$

Иккинчи тип раунд функцияси:

$$I = ((K_{mi} + R_{i-1}) <<< K_{Ri}),\ i = \overline{1...32}$$

$$F = ((S_1[I_1] - S_2[I_2]) + S_3[I_3]) \wedge S_4[I_4];$$

Учинчи тип раунд функцияси:

$$I = ((K_{mi} + R_{i-1}) <<< K_{Ri}),\ i = \overline{1...32}$$

$$F = ((S_1[I_1] + S_2[I_2]) \wedge S_3[I_3]) - S_4[I_4];$$

Бу ерда:

$S_i - i-S-$блок,

$I_j - I$ векторнинг j – байти,

<<< – чапга циклик суриш,

^-XOR бўйича қўшиш;

–, + $-\mathrm{mod}2^{32}$ бўйича айириш ва қўшиш;

Биринчи тип раунд функцияси 1, 4, 7, 10, 13 ва 16 раундда қўлланилади.

Иккинчи тип раунд функцияси 2, 5, 8, 11 ва 14 раундда қўлланилади.

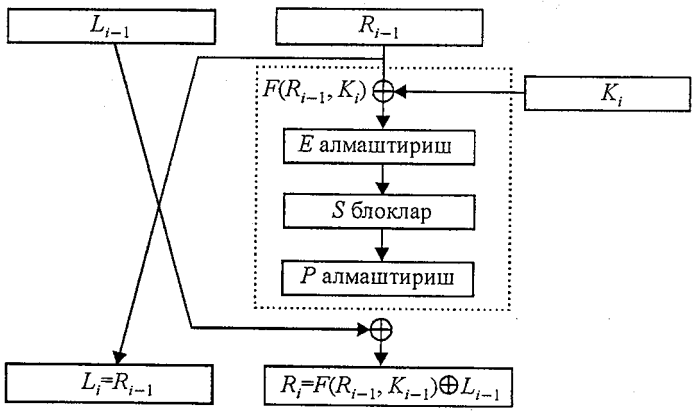
Учинчи тип раунд функцияси 3, 6, 9, 12 ва 15 раундда қўлланилади.

Қуйида **CAST-128** криптоалгоритмида қўлланилган акслантиришлар ва уларнинг бардошлиликка таъсири кўриб ўтилади.

CAST-128 криптоалгоритмида K_{mi} раунд калити ўнг ярим блок R_i билан mod 2^{32} амали бўйича қўшилади, яъни $C = (K_{mi} + R_{i-1}) \mathrm{mod}\, 2_{32}$. Бундан ташқали, бу амал умумлашган тўлдириш хусусиятига эга. Калит ёки чап ўнг блокнинг битта ўзгариши натижанинг камида битта бити ўзгаришига олиб келади. Шунингдек, натижа калитга боғлиқ равишда чапга циклик сурилади, калит эса номаълум. Бу амаллар бардошлиликни оширади. Алгоритмнинг ягона чизиқли бўлмаган акслантириши S – блоклар бўлиб, криптоалгоритмда тўртта S – блок қатнашади, ҳар бир S – блокда 256 та 32 разрядли сон қатнашади. S – блоклар ҳам бардошлиликни оширишга хизмат қилади. Ҳар бир S – блок натижалари бўйича айириш, қўшиш ва XOR амаллари қўлланилади. Бу амаллар битлар аралашувига таъсир этади.

§4.6. LOCI 91 симметрик блокли шифрлаш алгоритми

LOCI 91 криптоалгоритми блок узунлиги 64 га, калит узунлиги 64 га, раундлар сони 16 га тенг бўлиб, Австралияда ишлаб чиқилган. **LOCI 91** алгоритми i – раунди қуйидаги функционал схемада келтирилган:



4.7-расм. LOCI 91 криптоалгоритмининг i–раунди

Бу криптоалгоритмнинг $F(R_{i-1}, K_i)$ функцияси қуйидаги алмаштиришлардан ташкил топган:

1) Блокнинг 32 битли чап блокини 32 битли калитга XOR амали бўйича қўшиш;

2) E алмаштиришда 1) амал натижаси жадвал ёрдамида 32 битдан 48 битга кенгайтирилади;

3) S–блок акслантиришларида алгоритмда келтирилган тўртта S–блоклардан фойдаланилади. Бу S–блокларнинг кириши 12 битга, чиқиши 8 битга тенг. S-блок акслантиришлари қуйидагича:

– чап ва ўнг тарафдаги иккита бит олинади ва r сони топилади;

– ўртада жойлашган 8 та бит олинади ва s сони топилади.

Масалан $a=a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}$ бўлса, $r=a_1a_2a_{11}a_{12}$ ва $s=a_3a_4a_5a_6a_7a_8a_9a_{10}$.

r ва s сонлар топилгандан сўнг қуйидагича алмаштириш бажарилади:

$$S=S(r, s)=(s+((r\cdot 17)\oplus 0\times FF)\,\&\,0\times FF)31 \bmod P_r.$$

P_r қийматлар қуйидаги жадвал бўйича танлаб олинади.

r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P_r	375	279	391	395	397	415	419	425	433	445	451	463	471	477	448	488

4) P алмаштиришда 32 битли блок битлари ўринлари алмаштирилади, бунда блок узунлиги ўзгармайди.

Ўнг ярим блокни раунд калити билан қўшишда қўлланилган XOR амали бардошлиликни оширади, чунки алгоритм калити номаълум. Алгоритмда қўлланилган E ва P алмаштиришлар бардошлиликка тўғридан-тўғри таъсир этмаса ҳам, битлар аралашишига таъсир этади. Алгоритмда қўлланилган S–блокларга кирувчи қийматларни билган ҳолда чиқувчи қийматни топиш мумкин, лекин акси, яъни, чиқувчи қийматларни билган ҳолда кирувчи қийматни бир қийматли топиш мумкин эмас. Шундай бўлсада, Л. Кнудсен бу алгоритмни таҳлил қилиб, криптоалгоритм чизиқли ва дифференциал криптотаҳлил усулларига бардошли эмаслигини кўрсатган [14]. У танлаб олинган очиқ матнлар учун боғлиқ калитлар билан таҳлил қилиш тўлиқ танлаш усулига нисбатан деярли тўрт марта осонлигини кўрсатди. Шунингдек, **LOCI91** криптоалгоритмини танлаб олинган калитлар ва 2^{32} та танлаб олинган очиқ матн ёки 2^{48} та маълум очиқ матн ёрдамида очиш мумкинлиги кўрсатилган. Шунинг учун бу алгоритм бардошли алгоритмлар орасидан ўз ўрнини топа олмаган.

§4.7. FEAL симметрик блокли шифрлаш алгоритми

FEAL-8 криптоалгоритми япон криптографлари Акихиро Шимузи ва Шожи Миягучи томонидан таклиф этилган бўлиб, калит ва блок узунлиги 64 битга тенг тенг.

Бу криптоалгоритмда очиқ матн 64 битли X_i, $i=1, 2, 3, ...$блокларга бўлинади. 64 битли X_i блок 16 битли тўртта раунд калитига иккилик саноқ тизимидаги қўшиш, яъни XOR амали бўйича қўшилади: . Натижа X_i' эса узунлиги 32 битга тенг бўлган ва қисмларга бўлинади: $X_i' = (L_0, R_0)$. L_0 ва R_0 қисмлар XOR амали бўйича қўшилиб, янги қисм ҳосил қилинади, L_0 қисм эса ўзгармайди, яъни:

$$\begin{cases} L_0 = L_0 \\ R_0 = R_0 \oplus L_0 \end{cases}$$

Ҳосил бўлган L_0 ва R_0 узунлиги 32 битга тенг бўлган блокларга саккиз марта раунд функцияси F қўлланилади, яъни саккизта раунддан ўтади. Шуни таъкидлаб ўтиш керакки, ҳар бир раундда 16 битли раунд калити иштирок этади. Саккизта раунд қўлланилгандан сўнг,

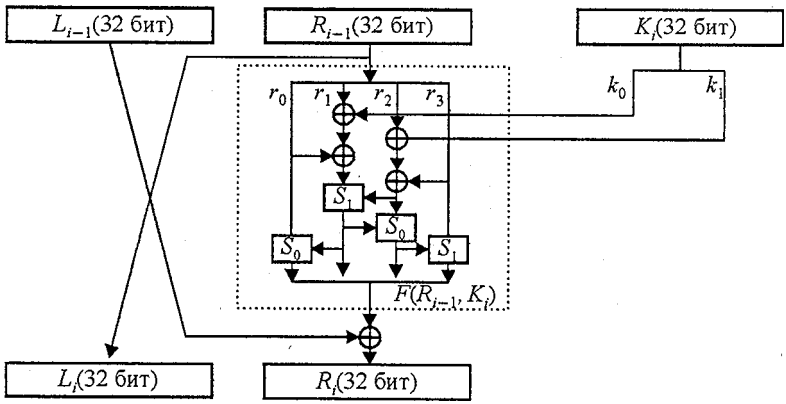
L_8 ва R_8 натижалар ҳосил бўлади. 32 битли ўнг R_8 блок ўзгармайди, чап L_8 блокка ўнг R_8 блок қ ўшилади ва чап тараф ҳосил қилинади:

$$\begin{cases} L_8 = L_8 \\ R_8 = R_8 \oplus L_8 \end{cases}$$

Ҳосил бўлган L_8 ва R_8 қисмлар ўрин алмаштирилиб ва янги $Y_i' = (R_8, L_8)$ 64 битли блок ҳосил қилинади. Ҳосил бўлган Y_i' блокка 16 битли тўртта раунд калити қўшилиб $Y_i = Y_i' \oplus [K_{12} \cup K_{13} \cup K_{14} \cup K_{15}]\ Y_i$ блок ҳосил қилинади.

FEAL криптоалгоритмида дешифрлаш функцияси шифрлаш функциясидагидек бўлиб, фақат раунд калитларининг қўлланилиши ўзгаради, яъни шифрлашда қўлланилган раунд калитлари тескари тартибда қўлланилади. Шунингдек, Фейстел тармоғигача бўлган ва Фейстел тармоғидан кейинги акслантиришлар ҳам ўзгармайди.

Қуйида **FEAL–8** криптоалгоритм ***i***-раундининг функционал схемаси келтирилган.



4.8-расм. FEAL–8 криптоалгоритмининг i–раунди

Бу ерда: $k_0, k_1, r_0, r_1, r_2, r_3$ узунлиги 8 битга тенг бўлган векторлар, S_0 эса $(a+b) \bmod 256$ ни чапга циклик 2 бит суриш натижаси ва S_1 эса $(a+b+1) \bmod 256$ ни чапга циклик 2 бит суриш натижасига тенг, яъни:

$$\begin{cases} S_0(a,b) = (a+b) \bmod 256) < 2; \\ S_1(a,b) = (a+b+1) \bmod 256) < 2. \end{cases}$$

Бу ердан кўриниб турибдики, **FEAL-8** криптоалгоритмида қўлланилган ягона чизиқли бўлмаган акслантиришлардан бири ўнг 32 битли блокни калит билан XOR амали бўйича қўшишдир, чунки калит номаълум. Лекин калит узунлиги 16 битга тенг, демак калитларни мумкин бўлган ҳамма қийматларини танлаш мумкин. $S_0(a, b)$ ва $S_1(a, b)$ акслантиришларида қўлланилган a ва b сонлар 8 битга тенг. $S_0(a, b)$, $S_1(a, b)$ қийматларни $1/2^{16}$ эҳтимоллик билан аниқлаш мумкин. Бу криптоалгоритмда чизиқсиз бўлмаган жадвал кўринишдаги S–блоклар қўлланилмаган.

Бихам ва Шамир **FEAL – 8** алгоритмини 10000 та танлаб олинган очиқ матнлар маълум бўлганда очиш мумкинлигини, дифференциал криптотаҳлил усулида қўллаб 2000та танлаб олинган ёки $2^{37,5}$ та маълум очиқ матнлар маълум бўлганда очиш мумкинлигини кўрсатишган. М.Мациу ва А.Ямагиши эса чизиқли криптотаҳлил усулини қўллаб, **FEAL – 8** криптоалгоритмини 2^{15} очиқ матнлар маълум бўлганда очиш мумкинлигини кўрсатишди. Шунингдек, 1995 йилда К.Аоки ва К.Ота **FEAL – 8** алгоритмини танлаб олинган 12 та очиқ матнлар асосида дешифрлашни кўрсатганлар. Муаллиф **FEAL** алгоритмини n раундли варианти **FEAL – *n*** ($n>8$)ни ҳам ишлаб чиқишган. Лекин бу алгоритм ҳам бардошли криптоалгоритмлар орасидан ўз ўрнини топа олмади. Дифференциал криптотаҳлил усули муаллифлари Бихам ва Шамир **FEAL-16** алгоритмини 2^{28} та танлаб олинган ёки $2^{46,5}$ та очиқ матн ёрдамида очиш мумкинлигини кўрсатишган [14].

Қуйидаги параграфда қўлингиздаги китоб муаллифига тегишли бўлган Фейстел тармоғи асосидаги симметрик блокли шифрлаш алгоритми ҳақида сўз юритилади.

§ 4.8. Асосий акслантиришлари: матрицали кенгайтириш, 256 байтли S –блок ва сиқиш жадвалидан иборат Фейстел тармоқли симметрик блокли шифрлаш алгоритми

Юқорида ўрганилган блокли шифрлаш алгоритмларининг энг асосий фарқлари раундлар итерациясида қўлланилган асосий акслантиришлар тузилишларининг (конструкцияларининг) ҳар-хиллигидадир. Бу акслантиришлар, электрон элементлар базасида қулай амалга оширилиши, криптобардошлилик хусусиятларни таъминлаши ва аппарат-техник қурилмалар модификациялари учун қулай ва самарали бўлиши лозим.

Эътиборингизга ҳавола этилаётган алгоритмда маълумот блоклари мос битларини раунд калитлари мос битларига mod 2 бўйича – XOR амали бўйича қўшиш, ҳамда, ***бу алгоритм муаллифи томонидан таклиф этилган*** *4 – байтли (32-битли) маълумот блокининг характеристикаси 256 бўлган чекли майдонда аниқланган тўғри тўртбурчакли матрица орқали кенгайтириш, байтларни 256 байтли S – блокдан фойдаланиб алмаштириш ва кенгайтирилган блокни сиқиш жадвали асосида дастлабки ўлчамига келтириш акслантиришлари қўлланилган.*

Алгоритм санаб ўтилган акслантиришлар комбинацияси асосида 64-битли маълумотни 256-битли калит орқали саккиз раундли итерация билан шифрлаш жараёнини амалга оширади.

Шифрлаш ва дешифрлаш жарёнларини ёритиш учун қуйидаги белгилашлар киритилади:

– T_0 – 64-битли очиқ маълумот блоки;

– $T_ш$ – 64-битли шифрланган маълумот блоки;

– t_i – 64-битли очиқ маълумот блокининг i – бити, бу ерда $i=1, ..., 64$; $i=0, 1, 2, ..., 8$;

– L_i ва R_i – 64-битли блокнинг мос равишда 32-битли чап ва ўнг қисмлари бўлиб, бу ерда $i=0, 1, 2, ..., 8$;

– $(a_1(i), a_2(i). ..., a_{32}(i))$ – i–раунд акслантиришининг чап қисми, яъни $L_i = (a_1(i), a_2(i). ..., a_{32}(i))$;

– $(b_1(i), b_2(i). ..., b_{32}(i))$ i–раунд акслантиришининг ўнг қисми, яъни $R_i = (b_1(i), b_2(i). ..., b_{32}(i))$;

– $x_{4\times1}=(x_1, x_2, x_3, x_4)$ – матрицали акслантиришга кирувчи 4-байтли (32-битли) вектор, бу ерда x_i – байтлар қийматлари ушбу оралиқдан олинади $0 \le x_i \le 255$, $i=1, 2, 3, 4$;

– $A_{n\times4}$ – тўғри тўртбурчакли матрица (олдиндан аниқланган қоида бўйича калит кетма-кетлигидан генерация қилиниб, махфий ҳисобланади ёки алоҳида генерация қилиниб, очиқ бўлиши ҳам мумкин), бунда $n=2^m$, $m=2, ..., M$, $M<\infty$,матрица элементлари $a_{ij}=(i=1, ..., n; j=1, 2, 3, 4;)$ бир байт билан ифодаланиб, ушбу $0 \le a_{ij} \le 255$ тенгсизликни қаноатлантиради;

– $y_{n\,1}=(y_1, y_2, ..., y_n)$ характеристикаси 256 бўлган чекли майдонда матрицали $A_{n\times4}x_{4\times1}$ акслантириш натижасини ифодаловчи вектор, яъни $y_{n\times4}=A_{n\times4}(\text{mod}256)$, бу ерда: y_i – байтлар, $0 \le a_{ij} \le 255$, $i=1, 2, ..., n$;

– $k=k_1k_2...k_8$ – сакизта k_i, $i=1, ..., 8$; 32-битли қисмкалитлардан иборат бўлган 256-битли калит;

– S–блок (олдиндан аниқланган қоида бўйича калитдан генерация қилинувчи 256-байтли блок, махфий) акслантириш, 256 та S_0, S_1, ..., S_{255} – байтлардан иборат бўлган:

S_0	S_1	S_2	...	S_{255}

жадвал, бу ерда: $0 \le S_1, S_2, ..., S_{256} \le 255$, $S_i \ne S_j$, яъни $0 \le S_i \le 255$ шартни қаноатлантирувчи S_i – сонларнинг тасодифий жойлашувидан иборат;

– блоклар векторларининг мос битларини mod2 бўйича – XOR амали билан қўшиш;

– $=(z_1, z_2, ..., z_n)$ – матрицали кенгайтириш акслантириши натижаси бўлган $y_{n\times1}=(y_1, y_2, ..., y_n)$ – векторни S – блок акслантиришлари натижаси, яъни $z_{n\times1}=S(y_{n\times1})$, бу ерда: z_i – байтлар, $0 \le z_i \le 255$, $i=1, 2, ..., n$;

– $k_i=k_1(i)\ k_2(i)\ ...\ k_{32}(i)$ – 32-битли i-қисмкалит;

– $k_i^1=...$, $k_i^4=k_{25}(i)\ k_{26}(i)\ ...\ k_{32}(i)$32-битли i – қисмкалитнинг тўртта байти;

– $k_{pi}=(k_1(\text{pi})\ k_2(pi)\ ...\ k_{32}(pi)$ – 32-битли i–раунд калити, бу ерда: $pi=1, ..., 8$.

– k_n – 64-битли бошланғич калит;

– k_k – 64-битли охирги калит;

– f – шифрлаш функцияси;

– СЖ –сиқиш жадвали, ўлчови 16×16 (махфий, калит билан биргаликда узатилади ёки олдиндан аниқланган қоида бўйича калитдан генерация қилинади), q_{ij} – элементлари $0 \le q_{ij} \le 15$, i=0, ..., 15, j=0, ..., 15 ва тенг тақсимланган:

q_{00}	q_{01}	...	$q_{0,15}$
q_{10}	q_{11}	...	$q_{1,15}$
...	...	...	...
$q_{15,0}$	$q_{15,1}$	...	$q_{15,15}$

– $w_{n\times1}=(w_1, w_2, w_3, w_4)$ – 32-битли (4-байтли) вектор, СЖ натижаси.

Шифрлашда калитларни сақлаш қурилмасига (массивига) 32-битли бўлган саккизта k_i–қисмкалитлардан ташкил топган 256-битли $k=k_1k_2$

... k_8 – калит блоки киритилади, очиқ маълумот 64-битли блокларга ажратилиб, ҳар бир T_0 – блок 8-раундли акслантиришлар жараёнидан ўтказилади. Ҳар бир i – раунд калити k_{pi} 32-битли $k_i = k_1(i)\ k_2(i)\ ...\ k_{32}(i)$ – қисмкалитни тўртта $k_i = (k_i^1, k_i^2, k_i^3, k_i^4) = (k_1(i)\ k_2(i)\ ...\ k_8(i),\ ...,\ k_{25}(i)\ k_{26}(i)\ ...\ k_{32}(i))$ байтларга ажратилиб, ҳосил бўлган ,..., байтларни ўнлик саноқ тизимидаги $(k_i^1)_{10}$, ..., $(k_i^4)_{10}$ –қийматлари бўйича S – блок ячейкалари тартиб сони (номери) аниқланади, ҳамда, ҳар бир k_i^l – байт S– блокнинг $(k_i^l)_{10}$ – тартиб сонли ячейкасида турган сонига алмаштириш билан аниқланади, яъни

$$k_{pi} = S(k_i^1, ..., k_i^4) = \left(S(k_i^1), ..., S(k_i^4)\right) = \left((S_{k_i^1})_2, ..., (S_{k_i^4})_2\right) = (k_{pi}^1, ..., k_{pi}^4).$$

Дастлабки 256-битли $k = k_1\ k_2\ ...\ k_8$ – калит икки марта СЖ акслантиришидан ўтказилиб, 64-битли бошланғич калит k_n ҳосил қилинади.

Дастлабки 256-битли $k = k_1\ k_2\ ...\ k_8$ – калит S–блок акслантиришларидан ўтказилиб, ҳосил бўлган 256-битли натижа икки марта СЖ акслантиришларидан ўтказилиб, 64-битли охирги калит k_k олинади.

Очиқ маълумот блоки T_0 мос битлари бошланғич калит k_n мос битлари билан XOR амали бўйича қўшилиб, яъни $T_0 \oplus k_n = T_0'$, ҳосил бўлган натижа T_0', яна T_0 ўзгарувчига берилиб $T_0 = T_0'$, иккита 32-битли қисмларга ажратилади:

$$T_0 = (t_1(0), t_2(0), ..., t_{32}(0), t_{33}(0), ..., t_{64}(0)) = (a_1(0), a_2(0), ...,$$
$$a_{32}(0), b_1(0), b_2(0), b_{32}(0)) = (L_0, R_0)$$

Биринчи раундда f – функция қийматини ҳисоблаш қуйидагича амалга оширилади:

1. Блок R_0 мос битлари раунд калити $k_{p1} = k_1(p1)\ k_2(p1)\ ...\ k_{32}(p1)$ мос битлари билан XOR амали бўйича қўшилади, яъни

$$b_1(0)\ b_2(0)\ ...\ b_{32}(0) \oplus k_1(p1)\ k_2(p1)\ ...\ k_{32}(p1) =$$
$$(b_1(0) \oplus k_1(p1))\ (b_2(0) \oplus k_2(p1))\ ...\ (b_3(0) \oplus k_{32}(p1)) =$$
$$= x_1(1)\ x_2(1)\ ...\ x_8(1)\ x_1(2)\ x_2(2)\ ...\ x_8(2)\ x_1(3)\ x_2(3)\ ...\ x_8(3)\ x_1(4)\ x_2(4)\ ...\ x_8(4) =$$
$$= (x_1, x_2, x_3, x_4) = x_{4\times 1};$$

2. Олдинги босқич натижаси $x_{4\times 1}$ характеристикаси 256 бўлган чекли майдонда аниқланган тўғри тўртбурчакли матрица $A_{n\times 4}$ орқали акслантирилади:

$$y_{n\times 1} = (A_{n\times 4}\ x_{4\times 1}) \bmod 256;$$

3. Байтлари сони n та бўлган $y_{n\times 1}$ – векторнинг ҳар бир i – байти y_i, $i=1,...n$, – блок акслантиришларидан ўтказилади, бунда байтнинг $(y_1(i)\ y_2(i)\ ...\ y_8(i))_2 = y_i$ ўнлик саноқ тизимидаги ифодаси $(y_1(i)\ y_2(i)\ ...\ y_8(i))_2 = (y_i)_{10}$ бўйича S – блок ячейкалари тартиб сони аниқланиб, $(y_1(i)\ y_2(i)\ ...\ y_8(i))_2 = y_i$ байт S – блокнинг $(y_i)_{10}$ – тартиб сонли ячейкасида турган $S_{(y_i)}$ сонига алмаштириш билан аниқланади, яъни:

$$z_1 = S(y_1) = S(y_1(i)\ y_2(i)\ ...\ y_8(i)) = (S_{y_i})_2;$$

4. Сиқиш жадвали СЖ бўйича $8\times n$ – битли (n – байтли) вектор $z_{n\times 1}$ 32 – битли (4-байтли) векторга $w_{4\times 1} = (w_1,\ w_2,\ w_3,\ w_4)$ акслантирилади:

$z_{n\times 1}$ – векторнинг ҳар бир z_i байти ярим байтли қисмларга ажратилади, яъни $z_{n\times 1} = (z_1,\ ...,\ z_n) = (z_1',\ ...,\ z_2') = z'_{2n\times 1}$;

– ярим байтли z_1' ва z'_{2n} блокларнинг ўнлик саноқ тизимидаги қийматлари $(z_1')_{10}$ ва $(z'_{2n})_{10}$ бўйича мос равишда СЖ сатр ва устун тартиб сонлари аниқланиб, уларнинг кесишган жойидаги ярим байт $q_{(z_1')_{10}(z'_{2n})_{10}}$ ярим байтли z_1' ва z'_{2n} иборат бўлган байтни $q_{(z_1')_{10}(z'_{2n})_{10}}$ – ярим байтга СЖ акслантириши натижаси ҳисобланади. Сўнгра, бу жараён барча $(z_2',\ z'_{2n-1})$, $(z_3',\ z'_{2n-2})$, ..., $(z_n',\ z'_{n+1})$, яъни $(z_i',\ z'_{2n-(i-1)})$, бу ерда $i=1,\ldots,\ n$; жуфтликлар учун қўлланилади;

–олдинги қадамдаги СЖ акслантириши $(m-2)$ марта қўлланилиб, натижада 32-битли (4-байтли) $w_{4\times 1} = (w_1,\ w_2,\ w_3,\ w_4)$ блок олинади;

5. Тўла сиқиш натижаси бўлган $w_{4\times 1} = (w_1,\ w_2,\ w_3,\ w_4)$ – 32-битли (4-байтли) векторнинг битлари XOR амали бўйича L_0 – блокнинг мос битларига қўшилади:

$$L_0 \oplus w_{4\times 1} = t_1(0)\ t_2(0)\ ...\ t_{32}(0) \oplus w_1(1)\ w_2(1)\ ...\ w_8(1)\ w_1(2)\ w_2(2)\ ...\ w_8(2)$$
$$w_1(3)\ w_2(3)\ ...\ w_8(3)\ w_1(4)\ w_2(4)\ ...\ w_8(4) = L_0 \oplus f(R_0,\ k_{p1}) = R_1$$

бу ерда: функция $f(R_0,\ k_{p1})$ орқали 1–4 –босқичлар акслантиришлари белгиланган;

6. R_0 – блокнинг қиймати ўзгаришсиз L_1 – блокга берилади: $L_1 = R_0$.

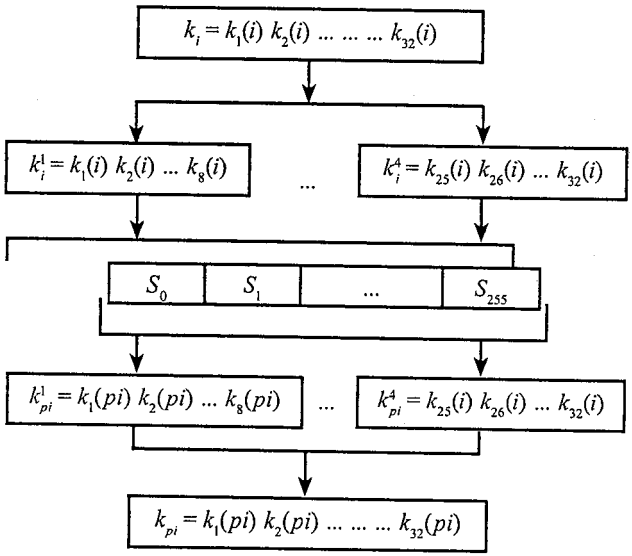
Юқорида келтирилган 1–6 –босқич акслантиришлари эътиборингизга ҳавола этилаётган шифрлаш алгоритмининг 1-раунд акслантиришларини ифодалайди.

Биринчи раунд акслантиришлари натижаларини ифодаловчи L_1 ва R_1 ўзгарувчилар қийматларини мос равишда L_0 ва R_0 ўзгарувчиларга берилиб, яъни $L_0=L_1$, $R_0=R_1$, ҳамда, биринчи раунд калити массивига иккинчи раунд калити массиви қийматини бериб $k_{p1}=k_{p2}$, сўнгра, 1–6 – босқичлар акслантиришларини қўллаб, 2-раунд акслантиришлари амалга оширилади. Шундай қилиб, агарда $(i-1)$ – раунд акслантириш натижалари маълум бўлса, ушбу $L_0=L_{i-1}$, $R_0=R_{i-1}$ ва $k_{p1}=k_{pi-1}$ амаллар бажарилиб, сўнгра 1–6 – босқичлар акслантиришларини қўллаб, i – раунд акслантиришлари амалга оширилади. Ҳавола этилаётган алгоритмнинг раундлари сони 8 та, яъни $i=1, 2, \ldots, 8$.

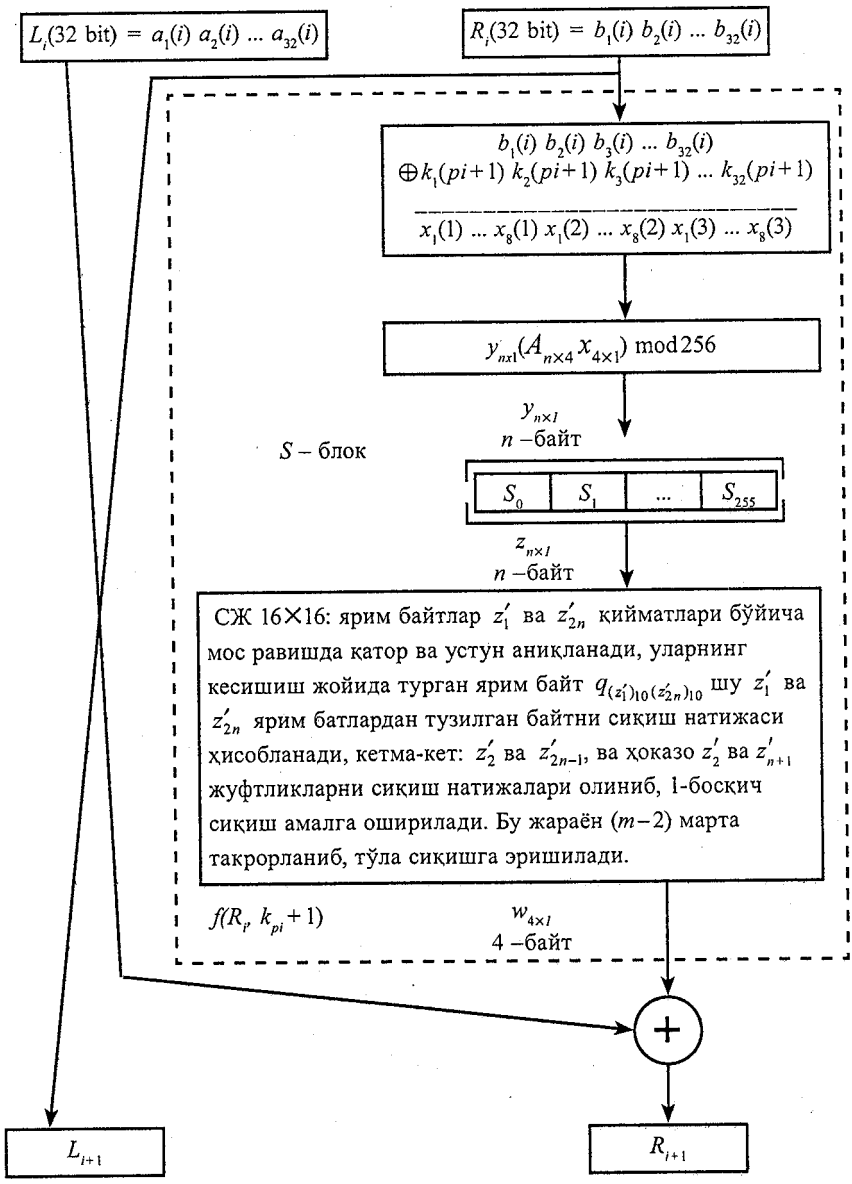
L_8 ва R_8 – блокларнинг бирлашмасидан тузилган $T_k=R_8L_8$ – блокнинг битлари k_k – блокнинг мос битларига XOR амали билан қўшилади, яъни $T_k \oplus k_k = T_{ш}$, очиқ маълумотнинг битта 64-битли блокини шифрлаш жараёни тамомланади.

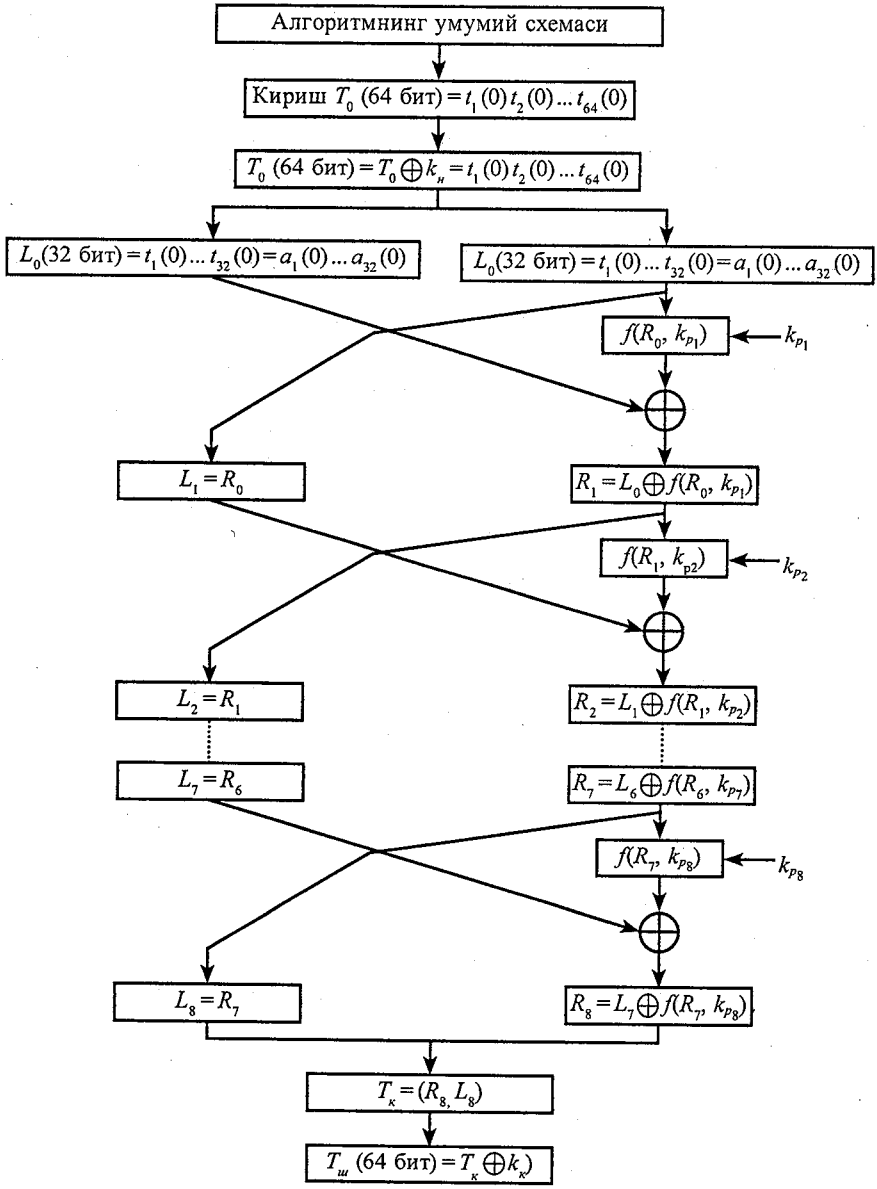
Қуйида, дастлабки калитдан раунд калитларини генерация қилиш, алгоритм шифрлаш жараёнининг i – раунди, ҳамда, алгоритмнинг умумий блок схемаси келтирилган:

i – раунд калити генерациясининг блок схемаси



Алгоритм *i*-раўндининг блок схемаси





Алгоритм шифрлаш жараёнининг келтирилган умумий схемасида $T_k = (R_8, L_8)$ ва 64-битли $T_ш$ – блок ифодаси $T_ш = T_k \oplus k_k = (R_8, L_8) \oplus k_k$ – аниқланган. Бундай аниқланишлар аппарат-дастурий қурилмалардан фойдаланиб шифрлаш ва дешифрлаш жар аёнларини амалга оширишнинг қулайлигини таъминлаш билан боғлиқ. Ҳақиқатан ҳам, кирувчи

блок сифатида $T_ш$ – блок ва бошланғич калит сифатида k_k – олиниб, раунд калитлари тескарисига қўлланилиб: 1-раундда k_{p8}, 2-раундда k_{p7}, …, 8-раундда k_{p1} ҳамда охирги калит сифатида $k_н$ – калит ишлатилиб, дешифрлаш жараёни худди шифрлаш жараёни каби амалга оширилади.

Криптобардошлиликни ошириш мақсадида, ҳар бир блокни шифрлашдан олдин 256-битли k – калитни, битлари сонини сақлаган ҳолда, λ – битга (бу ерда $3 \le \lambda 255$) суриш мумкин. Бундай суриш очиқ маълумот блокларини ҳар-хил калитлар билан шифрлаш имкониятини беради.

Таклиф этилган алгоритм Фейстел тармоғи функциясининг асосий акслантиришлари очиқ маълумот ва унинг акслантиришларининг блоки оралиқ қийматлари битларини раунд калитларининг мос битлари билан mod 2 бўйича қўшиш каби амалларга қўшимча тарзда характеристикаси 256 бўлган чекли майдонда аниқланган матрицали акслантириш, S–блок ҳамда СЖ акслантиришлари амалий жиҳатдан бир томонлама акслантиришлар ҳисобланади. Бундан ташқари S–блок ва СЖ юқори чизиқсизликни таъминловчи акслантиришлардир, ҳамда, XOR амали акслантириши юқори корреляция иммунитетини таъминлайди, матрицали акслантириш юқори бўлмаган чизиқсизлик ва корреляция иммунитетларини таъминлаб, кўпроқ тарқалиш тамоилини таъминлайди [15].

Юқорида таъкидлаганидек, тўғри тўртбурчакли $A_{n\times4}$ – матрица (бу ерда $n = 2^m$, $m = 2, ..., M$, $M<\infty$), S – блок ва СЖ дастлабки калитдан генерация қилиниши таъкидлаган эди. Қуйида уларни генерация қилиш қоидалари келтирилади.

Тўғри тўртбурчакли $A_{n\times4}$ матрица элементлари $a_{ij}(i=1, ..., n; j=1, 2, 3, 4)$ байтлардан иборат бўлиб, уларнинг сони $4n$ та:

1) $m=3$ бўлганда $n=2^3=8$ бўлиб, матрица элементлари сони 32 тадан иборат, уларни 256-битли дастлабки калитни 32 та байтга ажратиб, жуфт-жуфти билан ҳар-хил бўлакларидан, ҳар бир сатрда камида битта тоқ қийматли элемент бўладиган қилиб олинади;

2) $m=4$ бўлганда $n=2^4=16$ бўлиб, матрица элементлари сони 64 тадан иборат, уларни 256-битли дастлабки калитни бирор λ – битга битлари сонини йўқотмасдан (циклик) суриб, сўнгра байтларга ажратиб, жуфт-жуфти билан ҳар-хил бўлганларидан олдинги 32 тагача бўлган элементларини, олдин λ – битга сурилган 256-битли блокни яна λ – битга суриб, байтларга ажратиб, жуфт-жуфти билан ҳар-хил бўлакларидан кейинги 32 тагача бўлган элементларни ҳар бир сатрда камида битта тоқ қийматли элемент ҳамда матрицанинг ҳамма элементлари ҳар-хил бўладиган қилиб олинади ва ҳоказо.

Шу келтирилган қоида бўйича $A_{n\times 4}$ ($n=2^m$, $m=2$, ..., M, $M<\infty$) матрицанинг барча элементлари ҳосил қилинади.

Энди S–блок генерациясига ўтамиз. Дастлабки 256-битли калитни байтларга ажратиб, ҳосил бўлган байтлар қийматларини жуфт-жуфти билан солиштириб, сони 32 тадан ортиқ бўлмаган ҳар-хил байтлар билан S – блок ячейкалари тўлдирилади. Сўнгра, дастлабки калит битлари сонини йўқотмаган ҳолда $\lambda=2$, $\lambda=3$, ..., $\lambda=255$ битга сурилиб, ҳосил бўлган блокни байтларга ажратиб, ҳосил бўлган байтлар қийматларини S – блокнинг тўлдирилган ячейкаларидаги қийматларга солиштириб, улардан фарқли бўлган байтлар билан бўш ячейкаларни кетма-кет тўлдирилади. Бу жараённи дастлабки калит блокининг битлари сонини йўқотмаган ҳолда $\lambda=1$ битларга суриш билан S – блок ячейкаларини ҳаммаси тўлгунича давом эттирилади. Агар шунда ҳам S – блок ячейкалари охиригача тўлмаса, у ҳолда қолган ячейкалар S – блокнинг тўлдирилган ячейкаларида учрамаган байтлар билан охиригача тўлдирилади. Бунинг учун, ушбу $y=x^z \bmod 257$ формула бўйича $\{0\le y\le 256 : y=x^z \bmod 257,\ z=\text{const},\ 0\le x\le 256\}$ – тўплам элементлари кетма-кет ҳисобланади, бу ерда $0\le x\le 256$, $z=\text{const}$ бўлиб, у калит билан биргаликда узатилади. Сўнгра, $\{0\le \gamma\le 255 : \gamma = y \bmod 256,\ 0\le y\le 256\}$ тўпламнинг γ–элементлари S–блокнинг тўлган ячейкаларидаги байтлар қийматлари билан солиштирилиб, улардан фарқли бўлган байтлар билан кетма-кет тўлдирилади.

СЖ генерацияси 256-битли дастлабки калитни 64 та ярим байтли блокларга ажратиб, жуфт-жуфти билан ҳар-хил бўлган 16 та ярим байт билан 1-сатр, кейинги жуфт-жуфти билан ҳар-хил бўлган 16 та ярим байт билан 2-сатр ва ҳоказо 16-сатр тўлдирилади. Бунда, СЖ сатр ва устунларида, ҳамда, бош диагоналлар элментларида бир хилдаги ярим байтларнинг такрорланмаслиги ҳисобга олинади.

Таъкидлаш жоизки, тўғри тўртбурчакли матрица $A_{n\times 4}$ (бу ерда: $n=2^m$, $m=2$, ..., M, $M<\infty$), S–блок ва СЖ дастлабки калитдан олдиндан ҳисобланган ҳолда шу калит билан биргаликда узатилиши мумкин ёки дастлабки калитга боғлиқ бўлмаган ҳолда уларнинг юқорида келтирилган хоссаларини таъминлаш орқали генерация қилиниши мумкин.

Юқоридаги параграфларда Фейстел тармоғига асосланган симметрик блокли шифрлаш алгоритмлари ва уларнинг асосий акслантиришлари кўриб ўтилди. Кейинги параграфда Фейстел тармоғига асосланмаган AES-FIPS 197 ва бошқа алгоритмлар келтирилган.

§4.9. AES-FIPS 197 стандарт симметрик блокли шифрлаш алгоритми

DES блокли шифрлаш алгоритми 1999 йилгача АҚШда стандарт шифрлаш алгоритмлари сифатида ишлатиб келинган.

1974 йилдан Америка қўшма штатларининг стандарт шифрлаш алгоритми сифатида қабул қилинган DES шифрлаш алгоритми қуйидаги:

– калит узунлигининг кичиклиги, яъни 56 бит бўлиб, унинг 128 битдан кичиклиги;

– *S*–блок акслантиришларининг дифференциал криптотаҳлил усулига бардошсизлиги;

– ва бошқа сабабларга кўра эскирган деб саналади [25]. Айниқса 1999 йилда DES шифрлаш алгоритми ёрдамида шифрланган маълумотнинг Интернет тармоғига уланган 300 та паралел компьютер томонидан 24 соат давомида очилиши ҳақидаги маълумотнинг тасдиқланиши бундан кейин мазкур стандарт алгоритми ёрдамида маълумотларни криптографик муҳофаза қилиш масаласини қайтадан кўриб чиқиш ва янги стандарт қабул қилиш заруратини келтириб чиқарди.

Америка қўшма штатларининг «Стандартлар ва технологиялар Миллий Институти (NIST)» томонидан 1997 йилда XXI асрнинг маълумотларни криптографик муҳофазаловчи янги шифрлаш алгоритми стандартини қабул қилиш мақсадида танлов эълон қилинди. 2000 йилда стандарт шифрлаш алгоритми қилиб RIJNDAEL шифрлаш алгоритми асос қилиб олинган AES (Advanced Encryption Standard) (FIPS 197) қабул қилинди. Алгоритмнинг яратувчилари Белгиялик мутахассислар Йон Дэмен (Joan Daemen) ва Винсент Рюмен (Vincent Rijmen)ларнинг фамилияларидан RIJNDAEL номи олинган [25].

AES FIPS 197 блокли шифрлаш алгоритмида 8 ва 32-битли (1-байтли ва 4-байтли) векторлар устида амаллар бажарилади. AES FIPS 197 шифрлаш алгоритми XXI асрнинг энг барқарор шифрлаш алгоритми деб ҳисобланади [25]. Бу алгоритм бошқа мавжуд стандарт симметрик шифрлаш алгоритмларидан фарқли ўлароқ, Фейстел тармоғига асосланмаган блокли шифрлаш алгоритмлари қаторига киради.

4.9.1. AES криптоалгоритмининг математик асоси

AES алгоритмида байтлар устида амаллар бажарилади. Байтлар $GF(2^8)$ чекли майдон элементлари сифатида қаралади. $GF(2^8)$ майдон элементларини даражаси 7 дан катта бўлмаган кўпҳад сифатида тасвирлаш мумкин. Агарда байтлар

$$\{a_7a_6a_5a_4a_3a_2a_1a_0\},\ a_i \in \{0,\ 1\},\ i = \overline{0...7},$$

кўринишда тасвирланган бўлса, у ҳолда майдон элементлари қуйидагича кўпҳад кўринишда ёзилади:

$$a_7 \cdot x^7 + a_6 \cdot x^6 + a_5 \cdot x^5 \cdot a_4 + x^4 + a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0.$$

Мисол учун {11010101} байтга $x_7 + x_6 + x_4 + x_2 + a_0$ кўринишдаги кўпҳад мос келади.

Чекли $GF(2^8)$ майдон элементлари учун аддитивлик ва мультипликативлик хоссаларига эга бўлган қўшиш ва кўпайтириш амаллари аниқланган.

Кўпҳадларни қўшиш

AES алгоритмида кўпҳадларни қўшиш $\oplus$ **(XOR)** (берилган кўпҳадларга мос келувчи иккилик саноқ тизимидаги сонларнинг мос битларини mod 2 бўйича қўшиш) амали орқали бажарилади. Масалан, $x^7+x^6+x^4+x^2+x$ ва $x^7+x^5+x^3+x+1$ кўпҳадлар натижаси қуйидагича ҳисобланади:

$$(x^7+x^6+x^4+x^2+x) \oplus (x^7+x^5+x^3+x+1) = (x^6+x^5+x^4+x^3+x^2+1)$$

Бу амал иккилик ва ўн олтилик саноқ системаларида қуйидагича ифодаланади:

$$\{11010110\}_2 \oplus \{10101011\}_2 = \{01111101\}_2 \text{ ва } D6_{16} \oplus AB_{16} = 7D_{16}.$$

Чекли майдонда исталган нолга тенг бўлмаган a элемент учун унга тескари бўлган $-a$ элемент мавжуд ва $a+(-a)=0$ тенглик ўринли, бу ерда ноль элементи сифатида $\{00\}_{16}$ қаралади. $GF(2^8)$ майдонда $a \oplus a = 0$ тенглик ўринли.

Кўпҳадларни кўпайтириш

AES алгоритмида кўпҳадларни кўпайтириш қуйидагича амалга оширилади:

– иккита кўпҳад ўнлик саноқ тизимида кўпайтирилади;

– еттинчи даражадан катта бўлган ҳар қандай кўпҳадни саккизинчи даражали $\varphi(x)=x^8+x^4+x^3+x+1$ келтирилмайдиган кўпҳадга бўлганда қолдиқда етти ва ундан кичик бўлган даражадаги кўпҳадлар

ҳосил бўлиб, улар натижа сифатида олинади, бунда бўлиш жараёнида бажариладиган айириш амали иккилик саноқ тизимида, юқорида келтирилгани каби, $\oplus$ амали асосида бажарилади.

Шундай қилиб киритилган кўпайтириш амали $\bullet$ билан белгиланади.

Масалан, $(x^6+x^4+x^2+x+1)$ ва (x^7+x+1) кўпҳадлар қуйидагича кўпайтирилади:

– бу кўпҳадлар ўнлик саноқ тизимида кўпайтирилади

$(x^6+x^4+x^2+x+1)\bullet(x^7+x+1)=(x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1)$;

– натижа $\varphi(x)=x^8+x^4+x^3+x+1$ келтирилмайдиган кўпҳадга бўлинади ва қолдиқ олинади.

$$(x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1) \bmod (x^8+x^4+x^3+x+1)=(x^7+x+1).$$

Ҳақиқатан ҳам $(x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1)=(x^5+x^3)\bullet$
$\bullet(x^8+x^4+x^3+x+1)\oplus(x^7+x+1)$.

Ҳар қандай нолга тенг бўлмаган элемент учун $a\bullet 1=$ тенглик ўринли. $GF(2^8)$ майдонда бир элемент сифатида $\{01\}_{16}$ тушунилади.

Киритилган кўпайтириш амали умумий ҳолда қуйидагича бажарилади. Ихтиёрий еттинчи даражали

$$a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x+a_0$$

кўпҳадни x га кўпайтириб, қуйидагига эга бўламиз

$$a_7x^8+a_6x^7+a_5x^6+a_4x^5+a_3x^4+a_2x^3+a_1x^2+a_0\times.$$

Бу кўпҳадни $\varphi(x)=x^8+x^4+x^3+x+1=1\{1b\}$ модуль бўйича ҳисоблаб, чекли $GF(2^8)$ майдонга тегишли элементни ҳосил қиламиз. Бунинг учун $a_7=1$ бўлганда $\varphi(x)=x^8+x^4+x^3+x+1$ кўпҳадни юқорида олинган саккизинчи даражали кўпҳаддан *XOR* амали билан айириш кифоя, яъни :

$$(a_7\oplus 1)x^8+(a_6\oplus 0)x^7+(a_5\oplus 0)x^6+(a_4\oplus 0)x^5+(a_3\oplus 1)x^4+(a_2\oplus 1)x^3+$$
$$+(a_1\oplus 0)x^2+(a_0\oplus 1)x+1=(a_6\oplus 0)x^7+(a_5\oplus 0)x^6+(a_4\oplus 0)x^5+$$
$$+(a_3\oplus 1)x^4+(a_2\oplus 1)x^3+(a_1\oplus 0)x^2+(a_0\oplus 1)x+1,$$

бу ерда: $a_7=1$ бўлгани учун

$$(a_7\oplus 1)x^8=(1\oplus 1)x^8=0.$$

Агарда $a_7=0$ бўлса, у ҳолда натижа: $a_6x^7+\ldots+a_1x^2+a_0x$ кўпҳаднинг ўзи бўлади.

Ушбу x *time*( ) функция юқорида киритилган кўпайтириш амалига нисбатан берилган кўпҳадни x га кўпайтиришни ифодалайди. Шу функцияни n марта қўллаб x^n га кўпайтириш амали аниқланади. Бе-

восита ҳисоблаш билан қуйидагиларнинг ўринли эканлигига ишонч ҳосил қилиш мумкин:

$$\{57\} \bullet \{13\} = \{fe\},$$

чунки

$$\{57\} \bullet \{02\} = x\ time\ (\{57\}) = \{ae\}$$

$$\{57\} \bullet \{04\} = x\ time\ (\{ae\}) = \{47\}$$

$$\{57\} \bullet \{08\} = x\ time\ (\{47\}) = \{8e\}$$

$$\{57\} \bullet \{10\} = x\ time\ (\{8e\}) = \{07\},$$

бундан

$$\{57\} \bullet \{13\} = \{57\} \bullet (\{01\} \oplus \{02\} \oplus \{10\}) = \{57\} \oplus \{ae\} \oplus \{07\} = \{fe\}.$$

Юқорида таъкидланганидек алгоритм акслантиришлари байтлар ва тўрт байтли сўзлар устида бажарилади. Тўрт байтли сўзларни коэффициентлари $GF(2^8)$ чекли майдондан олинган даражаси учдан катта бўлмаган кўпҳадлар кўринишида ифодалаш мумкин:

$$a(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0,$$

бу ерда: $ai = (a_7^i a_6^i a_5^i a_4^i a_3^i a_2^i a_1^i a_0^i)$, $a_j^i \in \{0;1\}$, $i = 0,1,2,3$; $j = 0, 1, ..., 7$.

Бундай иккита кўпҳадларни қўшиш ўхшаш ҳадлар олдидаги коэффициентларни $\oplus$ амали билан қўшиш орқали амалга оширилади, яъни:

$$a(x) + b(x) = (a_3 \oplus b_3)x^3 + (a_2 \oplus b_2)x^2 + (a_1 \oplus b_1)x + (a_0 \oplus b_0).$$

Кўпайтириш амали қуйидагича амалга оширилади. Иккита тўрт байтли сўзлар мос кўпҳадлар билан ифодаланган бўлсин:

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \quad \text{и} \quad b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$$

Кўпайтириш натижаси олтинчи даражадан катта бўлмаган кўпҳад

$$a(x)b(x) = c(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0,$$

бўлиб, бу ерда: $c_0 = a_0 \bullet b_0$, $c_1 = a_1 \bullet b_0 \oplus a_0 \bullet b_1$, $c_2 = a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2$, $c_3 = a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3$, $c_4 = a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3$, $c_5 = a_3 \bullet \bullet b_2 \oplus a_2 \bullet b_3$, $c_6 = a_3 \bullet b_3$.

Кўпайтириш натижаси тўрт байтли сўздан иборат бўлиши учун, учинчи даражадан катта бўлган ҳар қандай кўпҳадни тўртинчи даражали $\varphi = x^4 + 1$ келтирилмайдиган кўпҳадга бўлганда қолдикда учинчи ва ундан кичик бўлган даражадаги кўпҳадлар ҳосил бўлишини

ҳисобга олган ҳолда, улар натижа сифатида олинади, бунда бўлиш жараёнида бажариладиган айириш амали иккилик саноқ тизимида, юқорида келтирилгани каби, амали асосида бажарилади.

Қуйидаги ифода ўринли:

$$x^i \bmod (x^4+1)=x^{i \bmod 4}.$$

Шундай қилиб, $a(x)$ ва $b(x)$ кўпҳадларни $\oplus$ – купайтмасини ифодаловчи

$$a(x)\oplus b(x)=d(x)=d_3x^3+d_2x^2+d_1x+d_0,$$

натижавий $d(x)$ – кўпҳад коэффициентлари қуйидагича аниқланади:

$$d_0=a_0\bullet b_0\oplus a_3\bullet b_1 a_2\bullet b_2\oplus a_1\bullet b_3,\ d_1=a_1\bullet b_0\oplus a_0\bullet b_1\oplus a_3\bullet b_2\oplus a_2\bullet b_3,$$
$$d_2=a_2\bullet b_0\oplus a_1\bullet b_1\oplus a_0\bullet b_2\oplus a_3\bullet b_3,\ d_3=a_3\bullet b_0\oplus a_2\bullet b_1\oplus a_1\bullet b_2\oplus a_0\bullet b_3.$$

Юқорида келтирилган амалларни матрица кўринишида қуйидагича ифодалаш мумкин:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \bullet \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Квадрат архитектурага эга **AES** блокли шифрлаш алгоритми ўзгарувчан узунликдаги калитлар орқали шифрланади. Калит ва блок узунликлари бир-бирига боғлиқ бўлмаган ҳолда 128, 192 ёки 256 бит бўлади. Биз **AES** шифрлаш алгоритмини блоклар узунлиги 128 бит бўлган ҳол учун кўриб чиқамиз.

Блок ўлчами 128 битга тенг кириш бу 16 байтли массив 4 та қатор ва 4 та устундан иборатдир (ҳар бир сатр ва ҳар бир устун бу ҳолда 32 битли сўз деб қаралади).

Шифрлаш учун кираётган маълумот байтлари:

$$s_{00},\ s_{10},\ s_{20},\ s_{30},\ s_{01},\ s_{11},\ s_{21},\ s_{31},\ s_{02},\ s_{12},\ s_{22},\ s_{32},\ s_{03},\ s_{13},\ s_{23},\ s_{33},$$

кўринишида белгиланади.

Кираётган маълумот қуйидаги **4.1-жадвалдаги** квадрат массив кўринишида киритилади. Яъни, байтларни тартиб билан устун бўйича тўлдириб борилади. Биринчи тўртта байт ($s_{00},\ s_{10},\ s_{20},\ s_{30}$) биринчи устунга мос тушади, иккинчи тўртта байт ($s_{01},\ s_{11},\ s_{21},\ s_{31}$) иккинчи устунга мос тушади, учинчи тўртта байт ($s_{02},\ s_{12},\ s_{22},\ s_{32}$) учинчи устунга мос тушади, тўртинчи тўртта байт ($s_{03},\ s_{13},\ s_{23},\ s_{33}$) тўртинчи устунга мос тушади.

s_{00}	s_{01}	s_{02}	s_{03}
s_{10}	s_{11}	s_{12}	S_{13}
s_{20}	s_{21}	s_{22}	s_{23}
s_{30}	s_{31}	s_{32}	s_{33}

4.1-жадвал. **Кираётган маълумотларнинг ҳолат жадвали.**

Худди шундай тартибда шифрлаш калити ҳам квадрат жадвал шаклида киритилади. Улар 128 бит = 16 байт = 4 сўз (тўртта 32 битлик блок) дан иборат:

k_{00}, k_{10}, k_{20}, k_{30}, k_{01}, k_{11}, k_{21}, k_{31}, k_{02}, k_{12}, k_{22}, k_{32}, k_{03}, k_{13}, k_{23}, k_{33}.

K_{00}	k_{01}	k_{02}	k_{03}
K_{10}	k_{11}	k_{12}	k_{13}
K_{20}	k_{21}	k_{22}	k_{23}
K_{30}	k_{31}	k_{32}	k_{33}

4.2-жадвал. **Шифрлаш калити ҳолат жадвали.**

Шунингдек, **AES** шифрлаш алгоритмининг раундлар сони N_r, кириш блоклар ўлчами N_b ва калит узунлиги N_k га боғлиқ ҳолда қуйидаги 4.3-жадвалга мос ҳолда қўлланилади.

N_r	N_b=4 128 бит	N_b=6 192 бит	N_b=8 256 бит
N_k=4 128 бит	10	12	14
N_k=6 192 бит	12	12	14
N_k=8 256 бит	14	14	14

4.3-жадвал.

4.9.2. Раунд акслантиришлари

Ҳар бир раунд шифрлаш жараёнлари қуйида келтирилган тўртта акслантиришлардан фойдаланилган ҳолда амалга оширилади [25]:

1) ***SubBytes*** – алгоритмда қайд этилган 16×16 ўлчамли жадвал асосида байтларни алмаштириш, яъни S – блок акслантиришларини амалга ошириш;

2) ***ShiftRows*** – алгоритмда берилган жадвалга кўра ҳолат байтларини циклик суриш;

3) ***MixColumns*** – устун элементларини аралаштириш, яъни алгоритмда берилган матрица бўйича акслантиришни амалга ошириш;

4) ***AddRoundKey*** – раунд калитларини қўшиш, яъни блоклар мос битларни ***XOR*** амали билан қўшиш.

Қуйида келтирилган акслантиришларнинг математик моделлари ва уларнинг умумий қўлланилиш схемалари кўриб чиқилади.

***SubBytes* (*S*–блок акслантиришлари жадвали)** – акслантириши ҳар бир ҳолат байтларига боғлиқсиз ҳолда байтларни чизиқли бўлмаган амаллар асосида ўрин алмаштиришларни амалга оширади. Бу жараён икки босқичдан иборат бўлиб:

а) ҳар бир s_{ij} ҳолат байтининг mod $(x^8+x^4+x^3+x+1)$ бўйича s_{ij}^{-1} тескариси топилади

$$s_{ij}s_{ij}^{-1} \equiv 1\,(x^8+x^4+x^3+x+1);$$

б) ҳар бир s_{ij} ни тескариси бўлган s_{ij}^{-1} ни $b = s_{ij}^{-1}$ деб белгилаб олиб, бир байтдан иборат бўлган b сонини унинг битлари орқали $b = (b_0, b_1, ..., b_7)$ кўринишда тасвирлаб, унинг устида қуйидаги афин акслатириши бажарилади

$$Cb + c \,(\mathrm{mod}\, x^8+1) = b$$

Бу ерда $C = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix}$ – матрица ва $c = (c_0, c_1, ..., c_7) =$

$= (1, 1, 0, 0, 0, 1, 1, 0)$ – вектор алгоритмда берилган ўзгармас ифодага эга бўлиб, келтирилган афин акслантириши

$$\begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} (\mathrm{mod}\, 257) = \begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix}$$

кўринишда амалга оширилади.

Натижавий $b' = (b'_0, b'_1, ..., b'_7)$ векторнинг координаталари $b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$, i=0, 1,2,...,7; ифода билан рационал ҳисобланади.

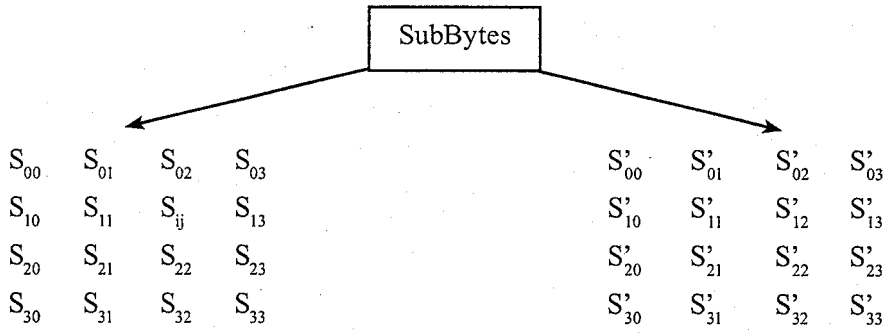
Юқоридаги а) ва б) қисмларда берилган барча мантиқий ва арифметик амалларни бажариш билан амалга ошириладиган ўрнига қўйиш акслантириши 4.4-жадвалдаги S–блок акслантиришларига (алмаштиришлари га) келтирилган. Бу эса алгоритмнинг дастурий таъминоти ва аппарат қурилмасини яратишда қулайлик туғдиради.

S–блок акслантиришларидан фойдаланиб берилган s–байтни 16-лик санок тизимида $s = (s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7) = \{s_0 s_1 s_2 s_3,\ s_4 s_5 s_6 s_7\} = \{xy\}$ каби ифодалаб x – сатр ва y – устунлар кесишмасидаги байтлар алмаштириш натижаси сифатида олинади. Мисол учун {62} – ни {*aa*} га алмаштирилади.

X	Y															
	0	1	2	3	4	5	6	7	8	9	A	b	c	d	e	F
0	63	7c	77	7b	12	6b	6f	C5	30	01	67	2b	fe	d7	ab	76
1	Ca	82	c9	7d	Fa	59	47	F0	ad	d4	a2	af	9c	a4	72	c0
2	b7	Fd	93	26	36	3f	F7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	62	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	Fc	b1	5b	6a	Cb	Be	39	4a	4c	58	cf
6	d0	Ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	Da	21	10	ff	f3	d2
8	Cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	18	73
9	60	81	4f	dc	22	2a	90	88	46	Ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	Ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	D5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	Ba	78	25	2e	1c	A6	b4	c6	e8	Dd	74	1f	4d	bd	8b	8a
d	70	3e	b5	66	48	03	F6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	D9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	Bf	E6	42	68	41	99	2d	0f	b0	54	bb	16

4.4-жадвал. ***S* – блок алмаштириш жадвали.**

***SubBytes* (*S* – блок акслантиришлари жадвали)** байтларни алмаштириш жараёнининг умумий схемасини қуйидагича тасвирлаш мумкин



***ShiftRows* (Ҳолат байтларини циклик суриш)** акслантиришининг қўлланилиши қуйидагича амалга оширилади. Ҳолат байтларини циклик суришда ҳолат жадвали сатрлари қуйидагича белгилаб олинади.

C_0 -сатр	S'_{00}	S'_{01}	S'_{02}	S'_{03}
C_1 -сатр	S'_{10}	S'_{11}	S'_{12}	S'_{13}
C_2 -сатр	S'_{20}	S'_{21}	S'_{22}	S'_{23}
C_3 -сатр	S'_{30}	S'_{31}	S'_{32}	S'_{33}

4.5-жадвал.

***ShiftRows* (Ҳолат байтларини циклик суриш)** акслантиришида жадвалдаги охирги учта сатр ҳар бир байтлари чапга циклик , яъни 1-сатр C_1 байтга, 2-сатр C_2 байтга, 3-сатр C_3 байтга сурилади. C_1, C_2, C_3 сурилиш қиймати N_b блок узунлигига боғлиқ. бўлиб, улар алгоритмда кўрсатилганидек, қуйидаги 4.6-жадвалда аниқланган [5]:

l	N_b	C_0	C_1	C_2	C_3
128	4	0	1	2	3
192	6	0	1	2	3
256	8	0	1	3	4

4.6-жадвал.

Келтирилган жадвалга кўра $l = 128$ *битли* шифрлаш учун $N_b = 4$ га тенг бўлиб, биринчи сатр бўйича ҳолат байтларини циклик суриш

бажарилмайди, иккинчи сатр бўйича 1 байтга, учинчи сатр бўйича 2 байтга, тўртинчи сатр бўйича 3 байтга циклик суриш амалга оширилади.

$l = 192$ *битли* шифрлаш учун $N_b = 6$ га тенг бўлиб, биринчи сатр бўйича ҳолат байтларини циклик суриш бажарилмайди, иккинчи сатр бўйича 1 байтга, учинчи сатр бўйича 2 байтга, тўртинчи сатр бўйича 3 байтга циклик суриш бажарилади.

$l = 256$ *битли* шифрлаш учун $N_b = 8$ га тенг бўлиб биринчи сатр бўйича ҳолат байтларини циклик суриш бажарилмайди, иккинчи қатор бўйича 1 байтга, учинчи сатр бўйича 3 байтга, тўртинчи сатр бўйича 4 байтга циклик суриш амал га оширилади.

4.7-жадвалда эса $l = 128$ *битли* шифрлаш учун $N_b = 4$ га тенг бўлганда, сатрларни циклик суриш бажарилгандан кейинги байтларнинг ўрни қай тарзда ўзгариши келтирилган

S'_{00}	S'_{01}	S'_{02}	S'_{03}			S'_{00}	S'_{01}	S'_{02}	S'_{03}
S'_{10}	S'_{11}	S'_{12}	S'_{13}			S'_{10}	S'_{11}	S'_{12}	S'_{13}
S'_{20}	S'_{21}	S'_{22}	S'_{23}	*Shift Rows*	→	S'_{20}	S'_{21}	S'_{22}	S'_{23}
S'_{30}	S'_{31}	S'_{32}	S'_{33}			S'_{33}	S'_{30}	S'_{31}	S'_{32}

4.7-жадвал.

MixColumns **(Устун элементларини аралаштириш) акслантириш**да ҳолат устунлари элементлари учинчи даражадан катта бўлмаган кўпҳаднинг коэффициентлари сифатида ифодаланиб, ана шу кўпҳад алгоритмда берилган:

$$g(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

кўпҳадга $x^4 + 1$ модуль бўйича кўпайтирилади.

Қуйидагича белгилаш киритилиб:

$$\begin{aligned} s_{00} &= s'_{00},\ s_{10} = s'_{11},\ s_{20} = s'_{22},\ s_{30} = s'_{33}, \\ s_{01} &= s'_{01},\ s_{11} = s'_{12},\ s_{21} = s'_{23},\ s_{31} = s'_{30}, \\ s_{02} &= s'_{02},\ s_{12} = s'_{13},\ s_{22} = s'_{20},\ s_{32} = s'_{31}, \\ s_{03} &= s'_{03},\ s_{13} = s'_{10},\ s_{23} = s'_{21},\ s_{33} = s'_{32}, \end{aligned} \tag{4.1}$$

таъкидланган кўпҳадлар кўпайтмасининг матрица кўринишидаги ифодаси:

$$\begin{bmatrix} s'_{0j} \\ s'_{1j} \\ s'_{2j} \\ s'_{3j} \end{bmatrix} = \begin{bmatrix} \{02\} & \{03\} & \{01\} & \{01\} \\ \{01\} & \{02\} & \{03\} & \{01\} \\ \{01\} & \{01\} & \{02\} & \{03\} \\ \{03\} & \{01\} & \{01\} & \{02\} \end{bmatrix} \bullet \begin{bmatrix} s_{0j} \\ s_{1j} \\ s_{2j} \\ s_{3j} \end{bmatrix}, \quad 0 \le c \le 3,$$

бўлади, бу ерда c – устун номери.

Охирги тенглик

$$\begin{aligned} s'_{0c} &= (\{02\} \bullet s_{0c}) \oplus (\{03\} \bullet s_{1c}) \oplus s_{2c} \oplus s_{3c}, \\ s'_{1c} &= s_{0c} \oplus (\{02\} \bullet s_{1c}) \oplus (\{03\} \bullet s_{2c}) \oplus s_{3c}, \\ s'_{2c} &= s_{0c} \oplus s_{1c} \oplus (\{02\} \bullet s_{2c}) \oplus (\{03\} \bullet s_{3c}), \\ s'_{3c} &= (\{03\} \bullet s_{0c}) \oplus s_{1c} \oplus s_{2c} \oplus (\{02\} \bullet s_{3c}, \end{aligned} \tag{4.2}$$

тенгликларга эквивалент.

***AddRoundKey* (Раунд калитларини қўшиш) акслантиришда** ҳолат блокининг битлари калит блоки мос битлари билан характеристикаси икки бўлган чекли майдонда қўшилади, яъни, массивнинг ҳар бир устуни ва шу устуннинг элементлари калит массивининг мос устун ва элементларига XOR амали билан қўшилади.

4.9.3. Калитлар генерацияси алгоритми (*Key Schedule*)

Раунд калитлари даслабки калитдан, алгоритмда кўзда тутилган ҳамма раундлар учун яратиб олинади. Бу жараён:

- калитни кенгайтириш (**Key Expansion**);
- раунд калитларини танлаш (**Round Key Selection**);
- босқичларидан иборат.

Раунд калитларининг умумий битлари сони кириш маълумотининг битлари сонининг раунд сонига кўпайтмасига ва яна битта кириш маълумоти битлари сонини йиғиндисига тенг (мисол учун 128 битли шифрлаш учун $128 \cdot 10 + 128 = 1408$ бит раунд калити керак бўлади), яъни N_b $(N_r + 1)$ ва $l(N_r + 1) = 128 \cdot 11 = 1408$ бит.

Демак, 128 бит узунликдаги блок ва 10 раунд учун 1408 бит раунд калитлари талаб қилинади.

Дастлабки калитни кенгайтиришда, аввал 128 битли (16 байт, символ) бошланғич кирувчи калит киритиб олинади ва тўртта (w_1, w_2, w_3, w_4) 32 битдан иборат бўлакка бўлинади. Қолган кенгайтирилган калитлар мана шу тўртта (w_1, w_2, w_3, w_4) кенгайтирилган калитлар ёрдамида топилади. Кенгайтирилган калитлар сони

$$N[w(i)] = N_b(N_r + 1);$$

Биз кўраётган ҳолатда $N_b = 4$, $N_r = 10$ га тенг яъни, байт узунлиги 4 га, раундлар сони 10 га тенг. Шуларни билган ҳолда $N[w(i)]$ ни топилади:

$$N[w(i)] = 4 \cdot (10 + 1) = 44$$

Демак, 128 битли кириш блокига ва 10 та раундга эга бўлган шифрлаш учун 44 та кенгайтирилган калитлар керак бўлар экан.

Раунд калитлари кенгайтирилган калитлардан қуйида баён қилинган қоида асосида яратилади. Калитлар генерациясининг формулалари қуйидаги кўринишларга эга:

$$w[i] = w[i-1] \oplus w[i-N_k], \tag{4.3}$$

ва

$$w[i] = SubWord\ (RotWord\ (w[i-1])) \oplus Rcon[i/N_k] \oplus w[i-N_k]. \tag{4.4}$$

Бизнинг ҳолатда $N_k = 4$ бўлганлиги сабабли $i = 4, 8, 12, 16, 20, \ldots$ қийматлар учун (4.2) формуладан фойдаланиб, кенгайтирилган калитлар топилади. Яъни, i нинг 4 га каррали, 4 га қолдиқсиз бўлинадиган қийматларида (4.2) формуладан фойдаланилади. Қолган барча $i = 5,6,7,9,10,11,13,\ldots$ қийматларида (4.3) формуладан фойдаланилади. Бу ерда $w(i)$ – 32 бит – сўзлардан иборат.

Масалан, биз кўраётган ҳолатда раунд калитининг узунлиги 128 бит тенг бўлиб, у тўртта кенгайтирилган калитга тенг бўлади, яъни,

$$128 : 32 = 4 \text{ демак, } w(i) = 1,2,3,4$$

$w_1 = W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8, W_9, W_{10}, W_{11}, W_{12}, W_{13}, W_{14}, W_{15}, W_{16}, W_{17}, W_{18}, W_{19}, W_{20}, W_{21}, W_{22}, W_{23}, W_{24}, W_{25}, W_{26}, W_{27}, W_{28}, W_{29}, W_{30}, W_{31}, W_{32};$

$$w_2 = W_{33}, W_{34}, W_{35}, W_{36}, W_{37}, W_{38}, W_{39}, W_{40}, W_{41}, W_{42}, W_{43}, W_{44}, W_{45}, W_{46}, W_{47}, W_{48},$$
$$W_{49}, W_{50}, W_{51}, W_{52}, W_{53}, W_{54}, W_{55}, W_{56}, W_{57}, W_{58}, W_{59}, W_{60}, W_{61}, W_{62}, W_{63}, W_{64};$$
$$w_3 = W_{65}, W_6, W_{67}, W_{68}, W_{69}, W_7, W_{71}, W_{72}, W_{73}, W_{74}, W_{75}, W_{76}, W_{77}, W_{78}, W_{79}, W_8,$$
$$W_{81}, W_{82}, W_{83}, W_{84}, W_{85}, W_{86}, W_{87}, W_{88}, W_{89}, W_9, W_{91}, W_{92}, W_{93}, W_{94}, W_{95}, W_{96};$$

$$w_4 = W_{97}, W_{98}, W_{99}, W_{100}, W_{101}, W_{102}, W_{103}, W_{104}, W_{105}, W_{106}, W_{107}, W_{108}, W_{109},$$
$$W_{110}, W_{111}, W_{112}, W_{113}, W_{114}, W_{115}, W_{116}, W_{117}, W_{118}, W_{119}, W_{120}, W_{121}, W_{122}, W_{123},$$
$$W_{124}, W_{125}, W_{126}, W_{127}, W_{128};$$

0 – раунд калити кириш калити	w_0, w_1, w_2, w_3.
1 – раунд калити	w_4, w_5, w_6, w_7
2 – раунд калити	w_8, w_9, w_{10}, w_{11}.
3 – раунд калити	$w_{12}, w_{13}, w_{14}, w_{15}$.
4 – раунд калити	$w_{16}, w_{17}, w_{18}, w_{19}$.
5 – раунд калити	$w_{20}, w_{21}, w_{22}, w_{23}$.
6 – раунд калити	$w_{24}, w_{25}, w_{26}, w_{27}$
7 – раунд калити	$w_{28}, w_{29}, w_{30}, w_{31}$.
8 – раунд калити	$w_{32}, w_{33}, w_{34}, w_{35}$.
9 – раунд калити	$w_{36}, w_{37}, w_{38}, w_{39}$.
10 – раунд калити	$w_{40}, w_{41}, w_{42}, w_{43}$.

4.8-жадвал. **Алгоритм барча раунди калитлари**

4.8-жадвалда раунд калитлари келтирилган бўлиб, 0 – раунд калити бошланғич кириш калити ҳисобланади, тўқ қора ранг билан берилган кенгайтирилган калитлар (4.4) формуладан, қолган калитлар эса (4.3) формуладан ҳисоблаб топилади.

(4.4) формуладаги акслантиришлар қуйидаги функциялар асосида амалга оширилади:

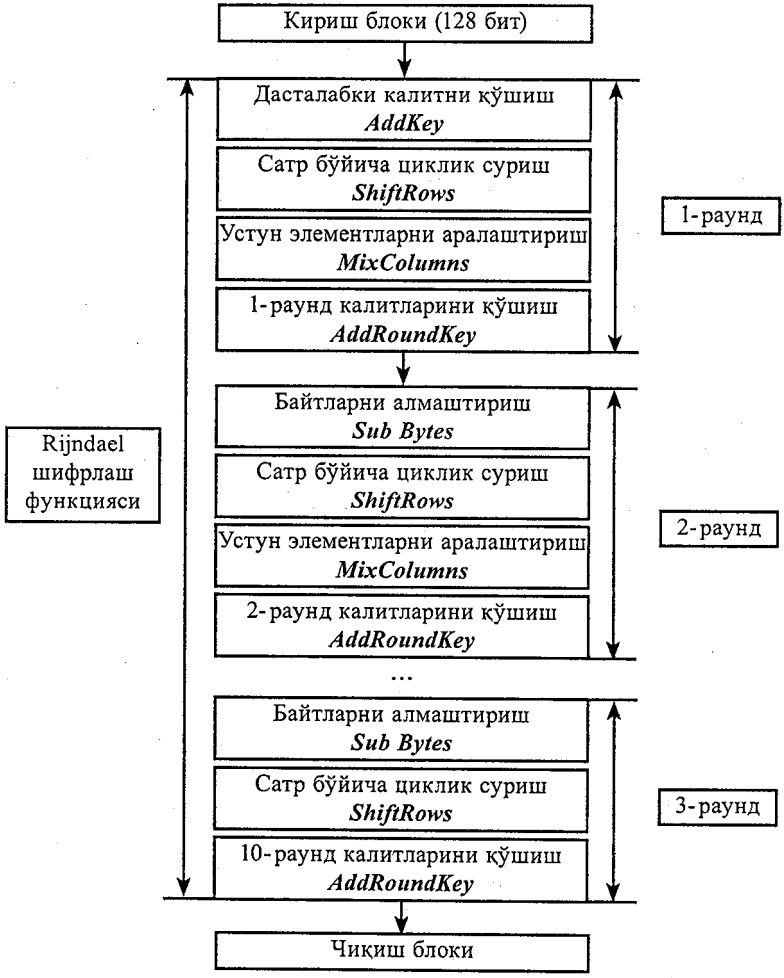
– ***RotWord*** – 32 битли сўзни байт бўйича қуйидаги кўринишда суриш бажарилади $\{a_0\, a_1\, a_2\, a_3\}\{a_1\, a_2\, a_3\, a_0\}$;

– ***SubWord*** – S блокдан ва ***SubBytes***( ) функциясидан фойдаланган ҳолда байт бўйича акслантириш бажарилади.

– ***Rcon*** $[j] = 2^{j-1}$, бу ерда $j = (i/N_k)$, i/N_k – бўлиш натижаси бутун сон чиқади, чунки N_k = const бўлиб, i нинг N_k га каррали қийматлари учун бўлиш амали бажарилади.

4.9.4. AES криптоалгоритми шифрлаш ва дешифрлаш жараёнларининг блок схемаси

Шифрлаш жараёни:



Дешифрлаш жараёни:

Шифрлаш жараёнида фойдаланилган ***Sub Bytes*** (1), ***ShiftRows*** (2), ***MixColumns*** (3) ва ***AddRoundKey*** (4) алмаштиришларига мос равишда тескари:

- ***invSub Bytes*( ),**
- ***invShiftRows*( ),**

– **inv*MixColumns*** ( ),

– ***AddRoundKey*( )**,

алмаштиришлар мавжуд бўлиб, бундай ҳолат қаралаётган симметрик шифрлаш алгритмининг аппарат-техник қурилмасини яратишда муҳим омиллардан ҳисобланади.

Қуйида мазкур тескари алмаштиришларни батафсил кўриб чиқамиз:

***Add Round Key*( )** – алмаштришида ишлатилаётган XOR амалининг хоссасига мувофиқ, ушбу функция ўз-ўзига тескари ҳисобланади.

2. ***inv Sub Bytes*( )** – алмаштириши шифрлаш жараёнида фойдаланиладиган S-блокка (**4.4-жадвал**) тескари амал бажаришга асосланган. Масалан $\{a5\}$ байт учун тескари байт алмаштириши амалининг натижаси $S-$блокда 2-сатр ва 9-устун элементларининг кесишган ерида жойлашгани учун жавоб: ***invSub Bytes*** $(\{a5\})=\{29\}$.

3. **inv *Shift Rows* ( )** – алмаштириши охирги ҳолат матрицасининг 3-та сатри берилган жадвал асосида ўнгга циклик суриш орқали амалга оширилади.

4. **inv *Mix Columns* ( )** – алмаштиришида ҳолат матрицаси устунлари $GF(2^8)$ майдонда учинчи даражали кўпҳад кўринишида қаралиб,

$$g^{-1}(x)=\{0b\}x^3+\{0d\}x^2+\{09\}x+\{0e\}$$

кўпҳадга модуль x^4+1 кўпҳад бўйича кўпайтирилади. Мазкур фикрларнинг математик ифодасини қуйидагича тасвирлаш мумкин:

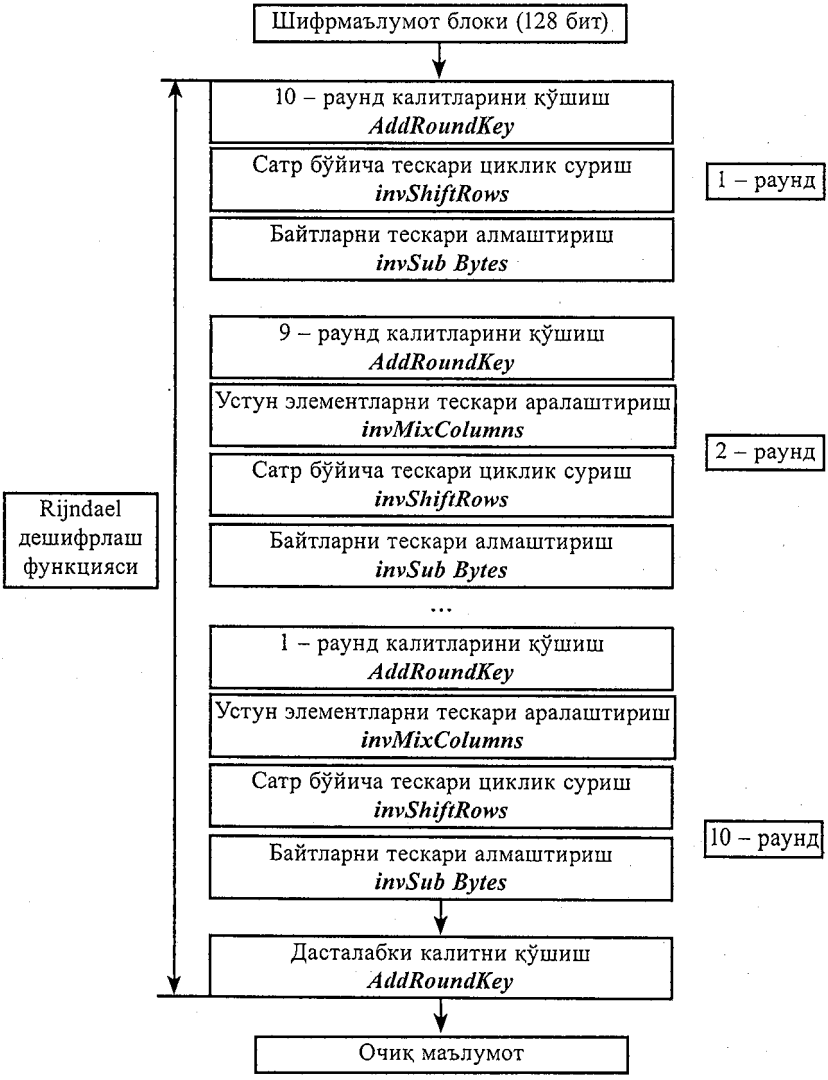
$$\begin{bmatrix} s_{0j} \\ s_{1j} \\ s_{2j} \\ s_{3j} \end{bmatrix} = \begin{bmatrix} \{05\} & \{0b\} & \{0d\} & \{09\} \\ \{09\} & \{0e\} & \{0b\} & \{0d\} \\ \{0d\} & \{09\} & \{0e\} & \{0b\} \\ \{0b\} & \{0d\} & \{09\} & \{0e\} \end{bmatrix} \bullet \begin{bmatrix} s'_{0j} \\ s'_{1j} \\ s'_{2j} \\ s'_{3j} \end{bmatrix} =$$

$$= \begin{bmatrix} (\{0e\}\bullet s'_{0j})\oplus(\{0b\}\bullet s'_{1j})\oplus(\{0d\}\bullet s'_{2j})\oplus(\{09\}\bullet s'_{3j}) \\ (\{09\}\bullet s'_{0j})\oplus(\{0e\}\bullet s'_{1j})\oplus(\{0b\}\bullet s'_{2j})\oplus(\{0d\}\bullet s'_{3j}) \\ (\{0d\}\bullet s'_{0j})\oplus(\{09\}\bullet s'_{1j})\oplus(\{0e\}\bullet s'_{2j})\oplus(\{0b\}\bullet s'_{3j}) \\ (\{0b\}\bullet s'_{0j})\oplus(\{0d\}\bullet s'_{1j})\oplus(\{09\}\bullet s'_{2j})\oplus(\{0e\}\bullet s'_{3j}) \end{bmatrix}$$

Ушбу тескари алмаштиришлардан фойдаланиб, дешифрлаш жараёнида генерация қилинган раунд калитлари охиргидан бошлаб биттадан камайтириб қўшиб борилади, яъни дешифрлаш жараёнининг

1-раундида шифрмаълумот блокига 10-раунд калити, 2-раундида 9-раунд калити ва ҳоказо, 10-раундида 1-раунд калити ва охирида эса дастлабки калит қўшилади. Юқорида таъкидланган жараённи бошқа тескари акслантиришлар билан биргаликда амалга ошириш-нинг умумий блок схемаси қуйида келтирилган.

Дешифрлаш жараёнининг умумий блок схемаси



4.9.5. AES FIPS 197 криптоалгоритми дастурий таъминотининг коди.

```
int const Nb=4;
int const Nk==4;
int const Nr=10;
int const KeyLenght=128;
unsigned int AesKey[Nk];
unsigned int KeyRound [(Nr+1)*Nb];
unsigned char state [4][Nb], W[(Nr+1)*Nb][4];

unsigned int rcon []={
0×00, 0×01, 0×02, 0×04, 0×08, 0×10, 0×20, 0×40, 0×80, 0×1B,
0×36, 0×6C, 0×D8, 0×AB, 0×4D, 0×9A, 0×2F, 0×5E, 0×BC, 0×63,
0×C6, 0×97, 0×35, 0×6A, 0×D4, 0×B3, 0×7D, 0×FA, 0×EF, 0×C5, 0×91};

unsigned char SBox[]={
0×63, 0×7c, 0×77, 0×7b, 0×f2, 0×6b, 0×6f, 0×c5, 0×30, 0×01, 0×67, 0×2b,
0×fe, 0×d7, 0×ab, 0×76,
0×ca, 0×82, 0×c9, 0×7d, 0×fa, 0×59, 0×47, 0×f0, 0×ad, 0×d4, 0×a2, 0×af,
0×9c, 0×a4, 0×72, 0×c0,
0×b7, 0×fd, 0×93, 0×26, 0×36, 0×3f, 0×f7, 0×cc,0×34, 0×a5, 0×e5, 0×f1,
0×71, 0×d8, 0×31, 0×15,
0×04, 0×c7, 0×23, 0×c3, 0×18, 0×96, 0×05, 0×9a, 0×07, 0×12, 0×80, 0×e2,
0×eb, 0×27, 0×b2, 0×75,
0×09, 0×83, 0×2c, 0×1a, 0×1b, 0×6e, 0×5a, 0×a0, 0×52, 0×3b, 0×d6, 0×b3,
0×29, 0×e3, 0×2f, 0×84,
0×53, 0×d1, 0×00, 0×ed, 0×20, 0×fc, 0×b1, 0×5b, 0×6a, 0×cb, 0×be, 0×39,
0×4a, 0×4c, 0×58, 0×cf,
0×d0, 0×ef, 0×aa, 0×fb, 0×43, 0×4d, 0×33, 0×85, 0×45, 0×f9, 0×02, 0×7f,
0×50, 0×3c, 0×9f, 0×a8,
0×51, 0×a3, 0×40, 0×8f, 0×92, 0×9d, 0×38, 0×f5, 0×bc, 0×b6, 0×da, 0×21,
0×10, 0×ff, 0×f3, 0×d2,
0×cd, 0×0c, 0×13, 0×ec, 0×5f, 0×97, 0×44, 0×17, 0×c4, 0×a7, 0×7e, 0×3d,
0×64, 0×5d, 0×19, 0×73,
0×60, 0×81, 0×4f, 0×dc, 0×22, 0×2a, 0×90, 0×88, 0×46, 0×ee, 0×b8, 0×14,
0×de, 0×5e, 0×0b, 0×db,
0×e0, 0×32, 0×3a, 0×0a, 0×49, 0×06, 0×24, 0×5c, 0×c2, 0×d3, 0×ac, 0×62,
0×91, 0×95, 0×e4, 0×79,
0×e7, 0×c8, 0×37, 0×6d, 0×8d, 0×d5, 0×4e, 0×a9, 0×6c, 0×56, 0×f4, 0×ea,
0×65, 0×7a, 0×ae, 0×08,
```

```
0×ba, 0×78, 0×25, 0×2e, 0×1c, 0×a6, 0×b4, 0×c6, 0×e8, 0×dd, 0×74, 0×1f,
0×4b, 0×bd, 0×8b, 0×8a,
0×70, 0×3e, 0×b5, 0×66, 0×48, 0×03, 0×f6, 0×0e, 0×61, 0×35, 0×57, 0×b9,
0×86, 0×c1, 0×1d, 0×9e,
0×e1, 0×f8, 0×98, 0×11, 0×69, 0×d9, 0×8e, 0×94, 0×9b, 0×1e, 0×87, 0×e9,
0×ce, 0×55, 0×28, 0×df,
0×8c, 0×a1, 0×89, 0×0d, 0×bf, 0×e6, 0×42, 0×68, 0×41, 0×99, 0×2d, 0×0f,
0×b0, 0×54, 0×bb, 0×16};

unsigned char InvSBox[] = {
0×52, 0×09, 0×6a, 0×d5, 0×30, 0×36, 0×a5, 0×38, 0×bf, 0×40, 0×a3, 0×9e,
0×81, 0×f3, 0×d7, 0×fb,
0×7c, 0×e3, 0×39, 0×82, 0×9b, 0×2f, 0×ff, 0×87, 0×34, 0×8e, 0×43, 0×44,
0×c4, 0×de, 0×e9, 0×cb,
0×54, 0×7b, 0×94, 0×32, 0×a6, 0×c2, 0×23, 0×3d, 0×ee, 0×4c, 0×95, 0×0b,
0×42, 0×fa, 0×c3, 0×4e,
0×08, 0×2e, 0×a1, 0×66, 0×28, 0×d9, 0×24, 0×b2, 0×76, 0×5b, 0×a2, 0×49,
0×6d, 0×8b, 0×d1, 0×25,
0×72, 0×f8, 0×f6, 0×64, 0×86, 0×68, 0×98, 0×16, 0×d4, 0×a4, 0×5c, 0×cc,
0×5d, 0×65, 0×b6, 0×92,
0×6c, 0×70, 0×48, 0×50, 0×fd, 0×ed, 0×b9, 0×da, 0×5e, 0×15, 0×46, 0×57,
0×a7, 0×8d, 0×9d, 0×84,
0×90, 0×d8, 0×ab, 0×00, 0×8c, 0×bc, 0×d3, 0×0a, 0×f7, 0×e4, 0×58, 0×05,
0×b8, 0×b3, 0×45, 0×06,
0×d0, 0×2c, 0×1e, 0×8f, 0×ca, 0×3f, 0×0f, 0×02, 0×c1, 0×af, 0×bd, 0×03,
0×01, 0×13, 0×8a, 0×6b,
0×3a, 0×91, 0×11, 0×41, 0×4f, 0×67, 0×dc, 0×ea, 0×97, 0×f2, 0×cf, 0×ce,
0×f0, 0×b4, 0×e6, 0×73,
0×96, 0×ac, 0×74, 0×22, 0×e7, 0×ad, 0×35, 0×85, 0×e2, 0×f9, 0×37, 0×e8,
0×1c, 0×75, 0×df, 0×6e,
0×47, 0×f1, 0×1a, 0×71, 0×1d, 0×29, 0×c5, 0×89, 0×6f, 0×b7, 0×62, 0×0e,
0×aa, 0×18, 0×be, 0×1b,
0×fc, 0×56, 0×3e, 0×4b, 0×c6, 0×d2, 0×79, 0×20, 0×9a, 0×db, 0×c0, 0×fe,
0×78, 0×cd, 0×5a, 0×f4,
0×1f, 0×dd, 0×a8, 0×33, 0×88, 0×07, 0×c7, 0×31, 0×b1, 0×12, 0×10, 0×59,
0×27, 0×80, 0×ec, 0×5f,
0×60, 0×51, 0×7f, 0×a9, 0×19, 0×b5, 0×4a, 0×0d, 0×2d, 0×e5, 0×7a, 0×9f,
0×93, 0×c9, 0×9c, 0×ef,
0×a0, 0×e0, 0×3b, 0×4d, 0×ae, 0×2a, 0×f5, 0×b0, 0×c8, 0×eb, 0×bb, 0×3c,
0×83, 0×53, 0×99, 0×61,
```

0×17, 0×2b, 0×04, 0×7e, 0×ba, 0×77, 0×d6, 0×26, 0×e1, 0×69, 0×14, 0×63,
0×55, 0×21, 0×0c, 0×7d};

static unsigned char Logtable [] = {
0 , 0, 25, 1, 50, 2, 26, 198, 75, 199, 27, 104, 51, 238, 223, 3,
100, 4, 224, 14, 52, 141, 129, 239, 76, 113, 8, 200, 248, 105, 28, 193,
125, 194, 29, 181, 249, 185, 39, 106, 77, 228, 166, 114, 154, 201, 9, 120,
101, 47, 138, 5, 33, 15, 225, 36, 18, 240, 130, 69, 53, 147, 218, 142,
150, 143, 219, 189, 54, 208, 206, 148, 19, 92, 210, 241, 64, 70, 131, 56,
102, 221, 253, 48, 191, 6, 139, 98, 179, 37, 226, 152, 34, 136, 145, 16,
126, 110, 72, 195, 163, 182, 30, 66, 58, 107, 40, 84, 250, 133, 61, 186,
43, 121, 10, 21, 155, 159, 94, 202, 78, 212, 172, 229, 243, 115, 167, 87,
175, 88, 168, 80, 244, 234, 214, 116, 79, 174, 233, 213, 231, 230, 173, 232,
44, 215, 117, 122, 235, 22, 11, 245, 89, 203, 95, 176, 156, 169, 81, 160,
127, 12, 246, 111, 23, 196, 73, 236, 216, 67, 31, 45, 164, 118, 123, 183,
204, 187, 62, 90, 251, 96, 177, 134, 59, 82, 161, 108, 170, 85, 41, 157,
151, 178, 135, 144, 97, 190, 220, 252, 188, 149, 207, 205, 55, 63, 91, 209,
83, 57, 132, 60, 65, 162, 109, 71, 20, 42, 158, 93, 86, 242, 211, 171,
68, 17, 146, 217, 35, 32, 46, 137, 180, 124, 184, 38, 119, 153, 227, 165,
103, 74, 237, 222, 197, 49, 254, 24, 13, 99, 140, 128, 192, 247, 112, 7};

static unsigned char ALogtable [] = {
1, 3, 5, 15, 17, 51, 85, 255, 26, 46, 114, 150, 161, 248, 19, 53,
95, 225, 56, 72, 216, 115, 149, 164, 247, 2, 6, 10, 30, 34, 102, 170,
229, 52, 92, 228, 55, 89, 235, 38, 106, 190, 217, 112, 144, 171, 230, 49,
83, 245, 4, 12, 20, 60, 68, 204, 79, 209, 104, 184, 211, 110, 178, 205,
76, 212, 103, 169, 224, 59, 77, 215, 98, 166, 241, 8, 24, 40, 120, 136,
131, 158, 185, 208, 107, 189, 220, 127, 129, 152, 179, 206, 73, 219, 118,
154,
181, 196, 87, 249, 16, 48, 80, 240, 11, 29, 39, 105, 187, 214, 97, 163,
254, 25, 43, 125, 135, 146, 173, 236, 47, 113, 147, 174, 233, 32, 96, 160,
251, 22, 58, 78, 210, 109, 183, 194, 93, 231, 50, 86, 250, 21, 63, 65,
195, 94, 226, 61, 71, 201, 64, 192, 91, 237, 44, 116, 156, 191, 218, 117,
159, 186, 213, 100, 172, 239, 42, 126, 130, 157, 188, 223, 122, 142, 137,
128,
155, 182, 193, 88, 232, 35, 101, 175, 234, 37, 111, 177, 200, 67, 197, 84,
252, 31, 33, 99, 165, 244, 7, 9, 27, 45, 119, 153, 176, 203, 70, 202,
69, 207, 74, 222, 121, 139, 134, 145, 168, 227, 62, 66, 198, 81, 243, 14,
18, 54, 90, 238, 41, 123, 141, 140, 143, 138, 133, 148, 167, 242, 13, 23,
57, 75, 221, 124, 132, 151, 162, 253, 28, 36, 108, 180, 199, 82, 246, 1};

```
int shifts [2] [4]={{0, 1, 2, 3}, {0, 3, 2, 1}};

unsigned int RotWord(unsigned int a)
{
return (((a)<<8)^((a)>>24));
}
unsigned int SubWord (unsigned int X)
{
return ((unsigned int)(SBox[X & 0×ff])) ^
((unsigned int)(SBox[(X>>8)& 0×ff]<<8)) ^
((unsigned int)(SBox[(X>>16)& 0×ff]<<16)) ^
((unsigned int)(SBox[(X>>24)& 0×ff]<<24));
}
void KeyExpansion(int KeyLenght)
{
unsigned int temp;
for(int i=0; i<Nk; i++)
{
KeyRound[i]=AesKey[i];
W[i][0]=(unsigned char) (KeyRound [i] &&0×ff);
W[i][1]=(unsigned char) ((KeyRound [i] >> 8)&&0×ff);
W[i][2]=(unsigned char) ((KeyRound [i] >> 16)&&0×ff);
W[i][3]=(unsigned char) ((KeyRound [i] >> 24)&&0×ff);
}
for(int i==Nk; i<Nb*(Nr+1); i++)
{
temp=Key Round [i–1];
if (i%Nk=0) temp=SubWord(RotWord(temp))^rcon[i/Nk];
else if((Nk>6) && ((i%Nk)==4)) temp=SubWord(temp);
KeyRound [i]=KeyRound[i-Nk]^temp;
W[i] [0]=(unsigned char) (KeyRound [i] &&0×ff);
W[i] [1]=(unsigned char) ((KeyRound [i] >>8) &&0×ff);
W[i] [2]=(unsigned char) ((KeyRound [i] >>16) &&0×ff);
W[i] [3]=(unsigned char) ((KeyRound [i] >>24) &&0×ff);
}
}
unsigned char MUL(unsigned char a, unsigned char b)
{
if ((a!=0) && (b!=0)) return ALogtable[(Logtable[a] +
Logtable[b])%255];
```

```
else return 0;
}
void AddRoundKey(int k)
{
for(int i=0; i<4; i++)
{
for(int j=0; j<Nb; j++)
{
state [j] [i]^=W[i+k] [j];
}
}
}
void SubBytes()
{
for(int i=0; i<4; i++)
{
for(int j=0; j<Nb; j++)
{
state[i][j]=SBox[state[i] [j]];
}
}
}
void InvSubBytes( )
{
for(int i=0; i<4; i++)
{
for(int j=0; j<Nb; j++)
{
state [i] [j]=InvSBox [state [i] [j]];
}
}
}
void ShiftRows()
{
for(int k=1; k<4; k++)
{
for(int i=0; i<shifts[0][k]; i++)
{
unsigned char a1=state[k][0];
for(int j=1; j<4; j++)
```

```
{
state[k][j–1]=state[k][j];
}
state[k][3]=a1;
}
}
}
void InvShiftRows()
{
for(int k=1; k<4; k++)
{
for(int i=0; i<shifts [1] [k]; i++)
{
unsigned char a1=state [k] [0];
for(int j=1;j<4;j++)
{
state [k] [j–1]=state [k] [j];
}
state [k] [3]=a1;
}
}
}
void MixCol(unsigned char *b)
{
unsigned char s[4];
s[0] =(MUL(0×2, b[0])^MUL(0×3, b[1])^b[2]^b[3]);
s[1] = (b[0]^MUL(0×2, b[1])^MUL(0×3, b[2])^b[3]);
s[2] = (b[0] ^ b[1] ^MUL(0×2, b[2]) ^ MUL(0×3, b[3]));
s[3] = (MUL(0×3, b[0]) ^ b[1] ^ b[2] ^MUL(0×2, b[3]));
b[0]=s[0]; b[1]=s[1]; b[2]=s[2]; b[3]=s[3];
}
void InvMixCol(unsigned char *b)
{
unsigned char s[4];
s[0] =(MUL(0×e, b[0]) ^ MUL(0×b, b[1]) ^ MUL(0×d, b[2]) ^ MUL(0×9,
b[3]));
s[1] =(MUL(0×9, b[0]) ^ MUL(0×e, b[1]) ^ MUL(0×b, b[2]) ^ MUL(0×d,
b[3]));
s[2] =(MUL(0×d, b[0]) ^ MUL(0×9, b[1]) ^ MUL(0×e, b[2]) ^ MUL(0×b,
b[3]));
s[3] =(MUL(0×b, b[0]) ^ MUL(0×d, b[1]) ^ MUL(0×9, b[2]) ^ MUL(0×e,
b[3]));
```

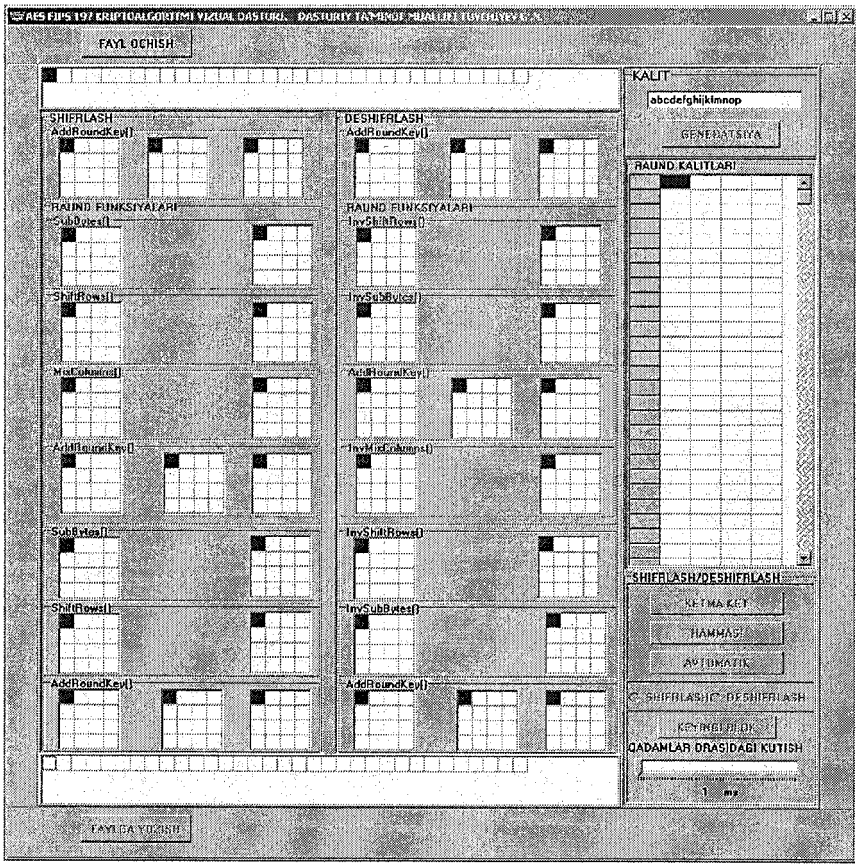
```
b[0]=s[0]; b[1]=s[1]; b[2]=s[2]; b[3]=s[3];
}
void MixColumns ()
{
unsigned char a [4];
for (int j=0; j<4; j++)
{
for (int i=0; i<4; i++)
{
a[i]=state [i] [j];
}
MixCol(a);
for (int i=0; i<4; i++)
{
state[i][j]=a[i];
}
}
}
void InvMixColumns ()
{
unsigned char a[4];
for (int j=0; j<4; j++)
{
for (int i=0; i<4; i++)
{
a[i]=state [i] [j];
}
InvMixCol(a);
for (int i=0; i<4; i++)
{
state[i][j]=a[i];
}
}
}
void Crypt ()
{
AddRoundKey (0);
for (int i=1; i<Nr; i++)
{
SubBytes ();
ShiftRows ();
MixColumns ();
```

```
AddRoundKey (i*4);
}
SubBytes ();
ShiftRows ();
AddRoundKey(Nr);
}
void Decrypt ()
{
AddRoundKey (Nr);
for (int i=Nr-1; i>0; i--)
{
InvShiftRows ();
InvSubBytes ();
AddRoundKey (i*4);
InvMixColumns ();
}
InvShiftRows ();
InvSubBytes ();
AddRoundKey (0);
}
unsigned char ByteInBlock [16];
unsigned char ByteOutBlock [16];
void AESCrypt(bool encrypt)
{
for (int i=0; i<4; i++)
{
for (int j=0; j<4; j++)
{
state [i][j]=ByteInBlock[j+(4*i)];
}
}
if (encrypt)
Crypt ();
else Decrypt ();
for (int i=0; i<4; i++)
{
for (int j=0; j<4; j++)
{
ByteOutBlock [j+(4*i)]=state [i] [j];
    }
    }
    }
```

§ 4.10. AES FIPS 197 стандарт блокли шифрлаш алгоритмининг дастур пакетидан фойдаланиш йўриқномаси

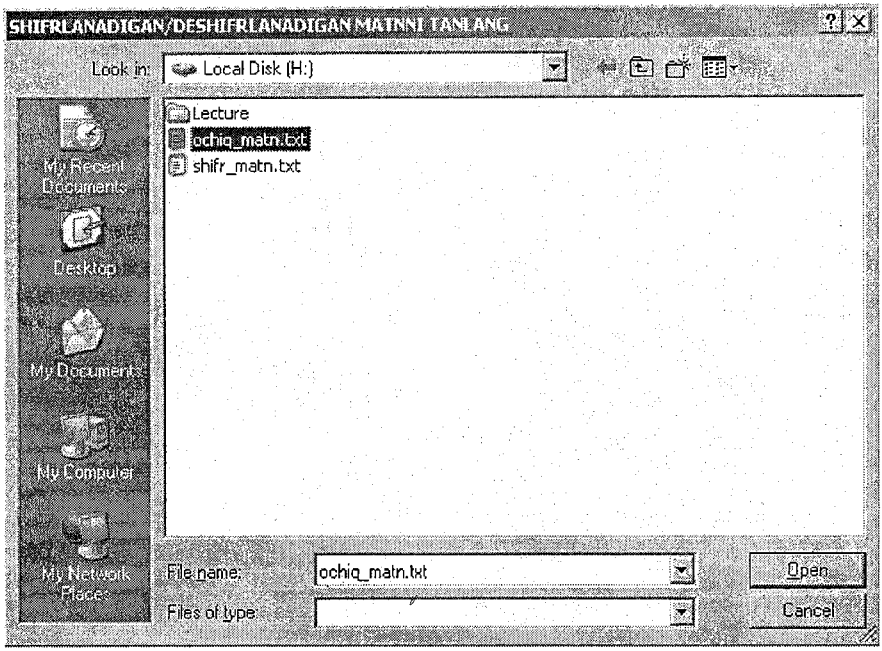
4.10.1. AES FIPS 197 стандарт блокли шифрлаш алгоритмининг визуал дастурий таъминотида матнларни киритиш

AES FIPS 197 стандарт блокли шифрлаш алгоритмининг дастурий таъминоти **Borland C++Bulder 6** дастурлаш тили асосида тузилган бўлиб, дастурнинг ишлаши қуйида батафсил келтирилади. Ушбу дастур ёрдамида алгоритмнинг ҳар бир раунд акслантиришларини таҳлил қилиш имконияти мавжуд. Дастурий таъминотнинг асосий ойнаси 4.9 –расмда келтирилган.



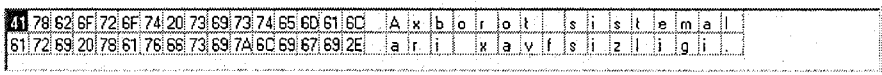
4.9-расм. **Дастурнинг асосий ойнаси.**

Бирор файлни шифрлаш ёки дешифлаш учун **FAYL OCHISH** тугмасини босиб, шифрланиши ёки дешифрланиши керак бўлган матн файлини танлаймиз.



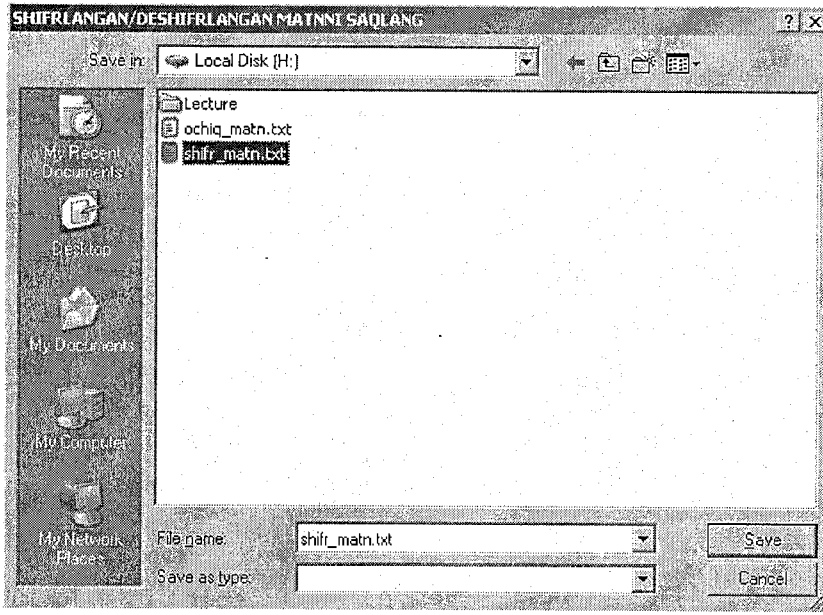
4.10-расм. **Шифрланиши керак бўлган матнни танлаш ойнаси.**

Дастур танлаб олинган файл матнини 16 байтли блокларга бўлиб олади ва 4.11-расмда кўрсатилганидек, чап томонда маълумот биринчи блокининг ўн олтилик саноқ тизимидаги қийматлари, ўнг томонда эса маълумотнинг биринчи блоки келтирилади. Кейинги қаторда маълумотнинг иккинчи блокининг ўн олтилик саноқ тизимидаги қийматлари ва маълумотнинг иккинчи блоки келтирилади. Шу усулда *n* та блокни келтириш мумкин.



4.11-расм. **Танлаб олинган маълумотнинг асосий ойнада келтирилиши.**

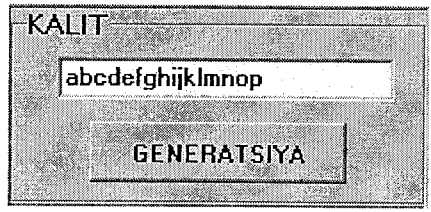
Шифрланган ёки дешифланган файлни сақлаш учун **FAYLGA YOZISH** тугмаси босилиб, натижада ушбу ойна ҳосил бўлади.



4.12-расм.

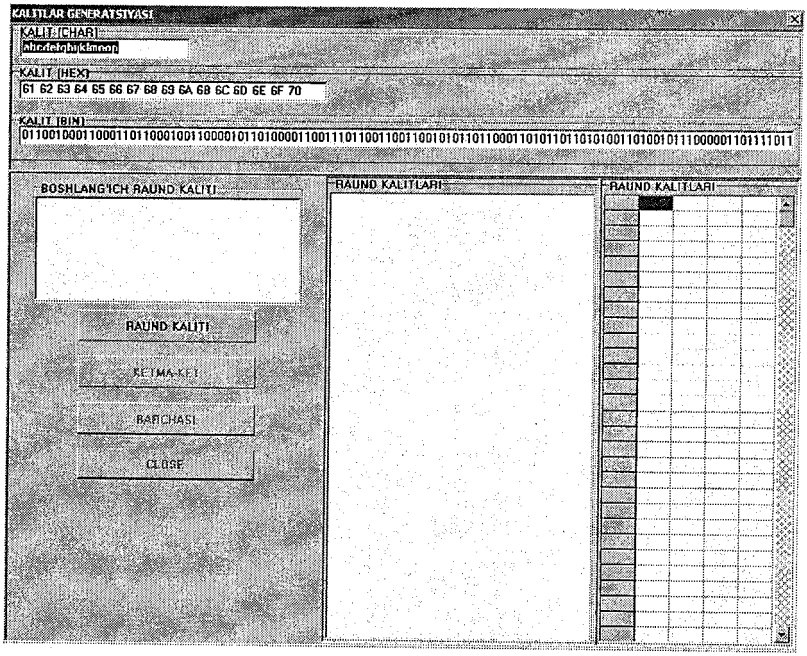
4.10.2. Криптоалгоритм раунд калитлари генерацияси функциясининг визуал дастурий таъминоти ва унинг ишлаш принципи.

Шифрланиши ёки дешифрланиши керак бўлган файл танлангандан сўнг, раунд калитлари генерациясига ўтилади. Алгоритм раунд калитларини 128 битли калитдан генерация қилади. Бунинг учун 16 байтли ихтиёрий символ киритилади.



4.13-расм. **Калитни киритиш объекти.**

Калит киритилгандан сўнг, **GENERATSIYA** тугмаси босилиб, раунд калитларини генерация қилувчи **KALITLAR GENERATSIYASI** ойнаси чиқади.

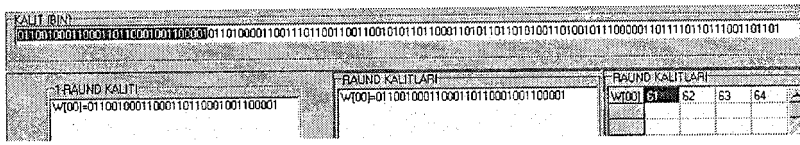


4.14-расм. **Раунд калитларини ишлаб чиқувчи ойна.**

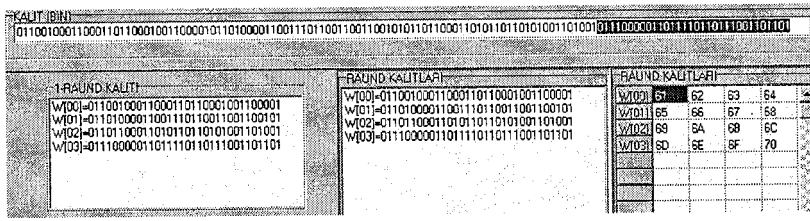
Алгоритмнинг 128 битли калити 32 битли бўлган 4 та **W(0), W(1), W(2), W(3)** бошланғич калитларга бўлиб олинади. **RAUND KALITI** тугмасини босиб **W(0), W(1), W(2), W(3)** бошланғич раунд калитларини олиш мумкин.

4.15-расм. **KALITLAR GENERATSIYASI ойнасининг режимларга ўтиш тугмалари.**

Иккилик кўринишидаги алгоритм калитининг биринчи 32 бити W(0) раунд калитига ўтади ва ҳ.к. W(1), W(2), W(3) бошланғич раунд калитлари олинади.

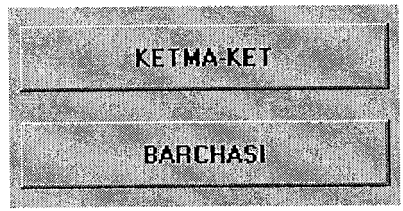


4.16-расм. **Алгоритм калитидан W(0) бошланғич калитни олиш.**



4.17-расм. **Алгоритм калитидан W(0), W(1), W(2), W(3) бошланғич калитларни олиш.**

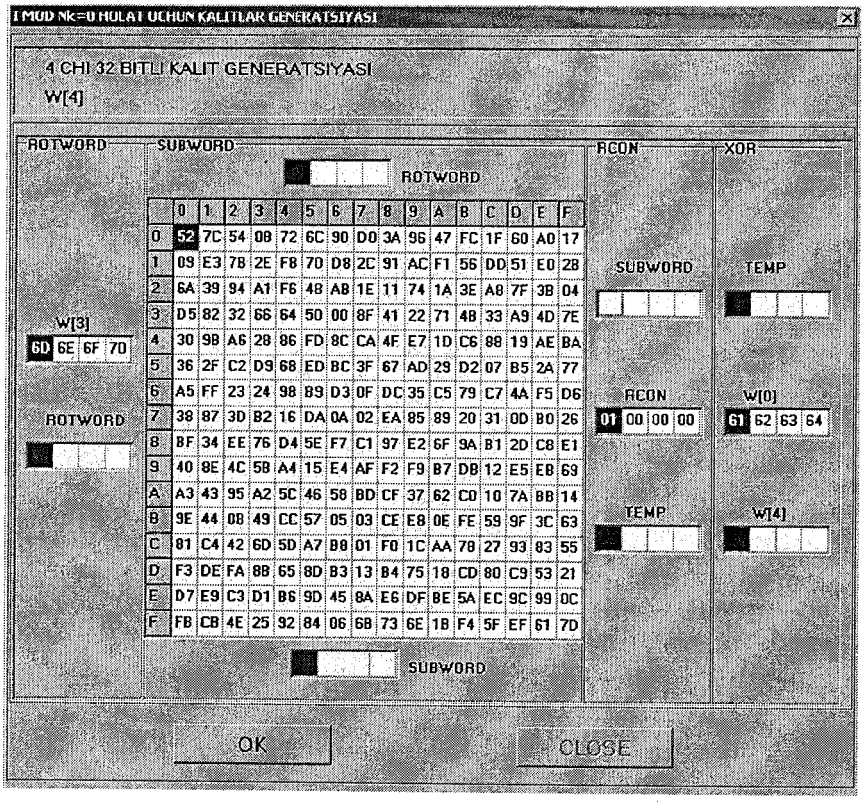
Бошланғич калитлардан яна 40 та калит (W(4), W(5), ..., W(44)) ишлаб чиқиш керак бўлади. Агар **KETMA-KET** тугмаси босилса, барча кенгайтирилган калитларни ишлаб чиқиш босқичма-босқич амалга оширилади, **BARCHASI** тугмаси босилса, ҳамма кенгайтирилган калитлар кетма-кет ишлаб чиқилади.



4.18-расм. **KALITLAR GENERATSIYASI ойнасининг режимларга ўтиш тугмалари.**

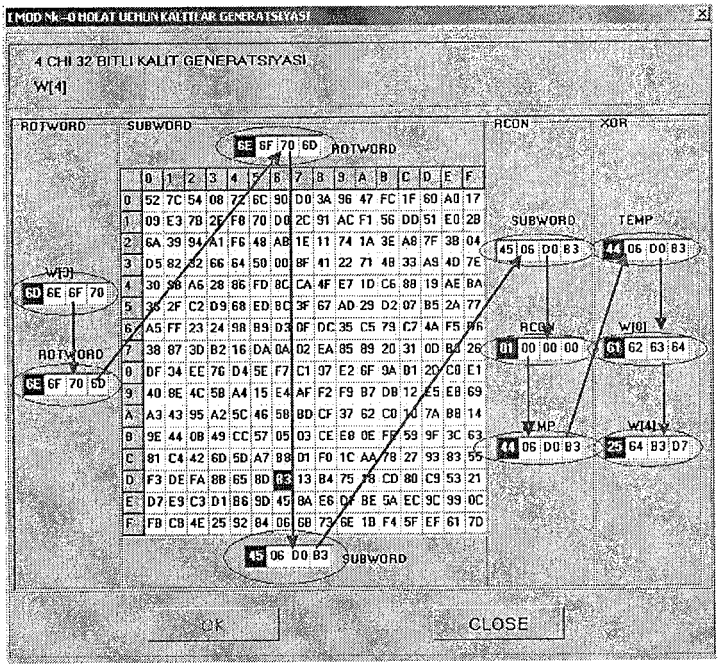
Айтайлик, дастур ишлатилиши давомида **KETMA-KET** тугмаси босилган бўлсин, у ҳолда **I MOD Nk=0 HOLAT UCHUN KALITLAR GENERATSIYASI** ойнаси (**4.19-расм**) чиқади. Бу ойна ёрдамида тўртга каррали бўлган, яъни W(4), W(8), W(12), W(16),

W(20), W(24), W(28), W(32), W(36), W(40), W(44) кенгайтирилган калитлари (4.4) формула ёрдамида ишлаб чиқилади. Қолган барча кенгайтирилган калитлар эса **I MOD Nk<>0 HOLAT UCHUN KALITLAR GENERATSIYASI** ойнасида (4.**19-расм**) амалга оширилади.



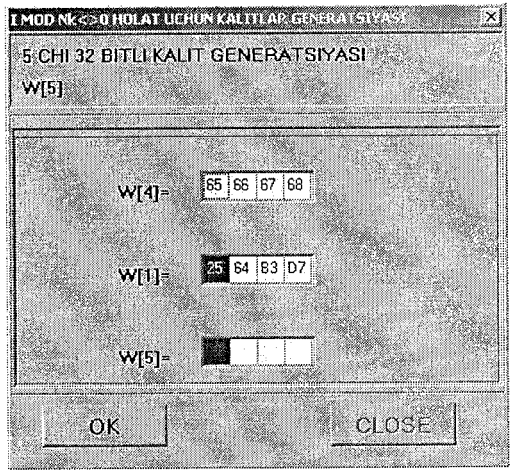
4.19-расм. **Тўртга каррали бўлган раунд калитларини ишлаб чиқиш ойнаси.**

Масалан, тўртга каррали бўлган W(4) кенгайтирилган калитни ишлаб чиқиш учун биринчи **ROTWORD** функцияси бажарилади, сўнгра **ROTWORD** функциясидан чиққан натижадан фойдаланган ҳолда, **SUBWORD** функцияси бажарилади, учинчи **RCON** функцияси бажарилади, тўртинчи **SUBWORD** функциясидан ва **RCON** функциясидан чиққан натижалар ва натижа *W(i–Nк)* кенгайтирилган калитлари устида XOR амали бажарилади.



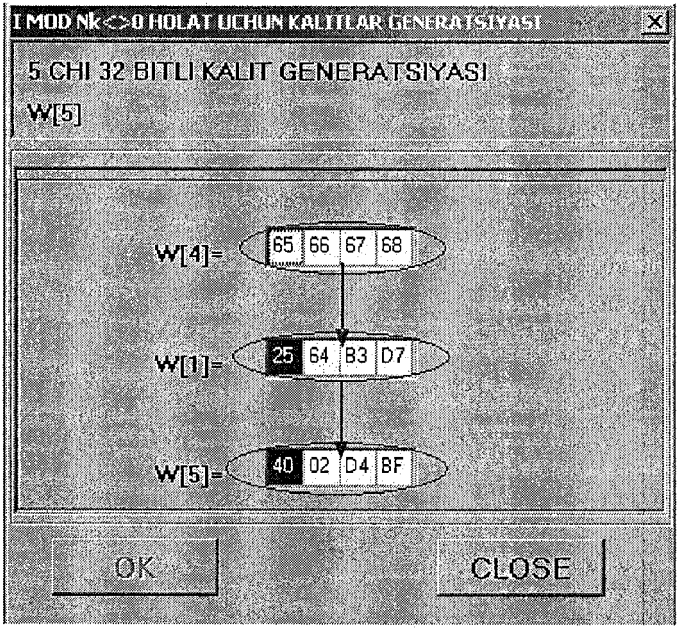
4.20-расм. **Тўртга каррали бўлган раунд калитларини ишлаб чиқиш ойнаси.**

Тўртга каррали бўлмаган калитлар **I MOD Nk<>0 HOLAT UCHUN KALITLAR GENERATSIYASI** ойнасида (4.21-расм) ишлаб чиқилади



4.21-расм. **Тўртга каррали бўлмаган раунд калитларини ишлаб чиқиш ойнаси.**

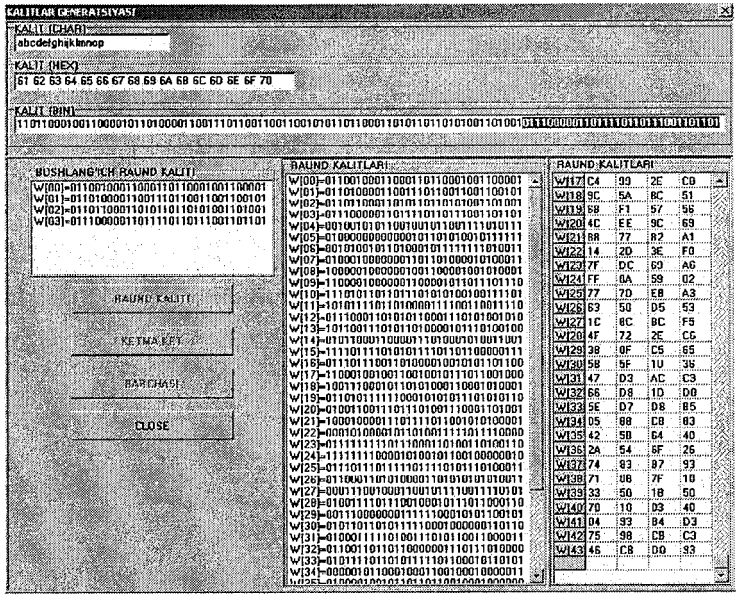
Бу ойна ёрдамида тўртга каррали бўлмаган калитлар яъни, W(5), W(6), W(7), W(9), W(10), W(11), W(13), W(14), W(15), W(17), W(18), W(19), W(21), W(22), W(23), W(25), W(26), W(27), W(29), W(30), W(31), W(33), W(34), W(35), W(37), W(38), W(39), W(41), W(42), W(43) кенгайтирилган калитлари (3.1) формула ёрдамида ишлаб чиқилади. Масалан W(5) кенгайтирилган калитни ишлаб чиқиш учун (4.4) формулага асосан W(4) ва W(1) кенгайтирилган калитлари устида XOR амали бажарилади.



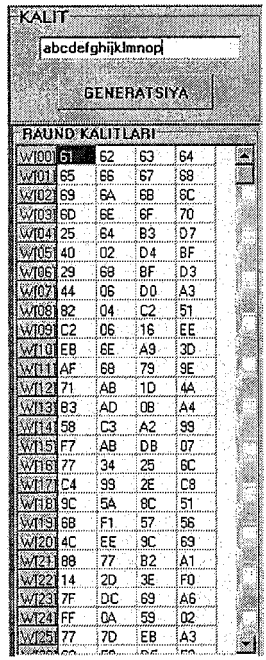
4.22-расм. **Тўртга каррали бўлмаган раунд калитларини ишлаб чиқиш ойнаси.**

Барча кенгайтирилган калитлар ишлаб чиқилгандан сўнг, **CLOSE** тугмаси босилади. Қуйида **KALITLAR GENERATSIYASI** ойнасида ҳосил бўлган барча раунд калитлари келтирилган.

Дастурий таъминот асосий ойнасининг махсус ажратилган жойида раунд калитлари кўрсатилади. Симметрик шифрлаш алгоритми шифрлаш ва дешифрлаш жараёнида фойдаланиладиган барча раунд калитлари **4.23-расмда** ўз аксини топган.



4.23-расм. **Раунд калитларини ишлаб чиқувчи KALITLAR GENERATSIYASI ойнаси.**

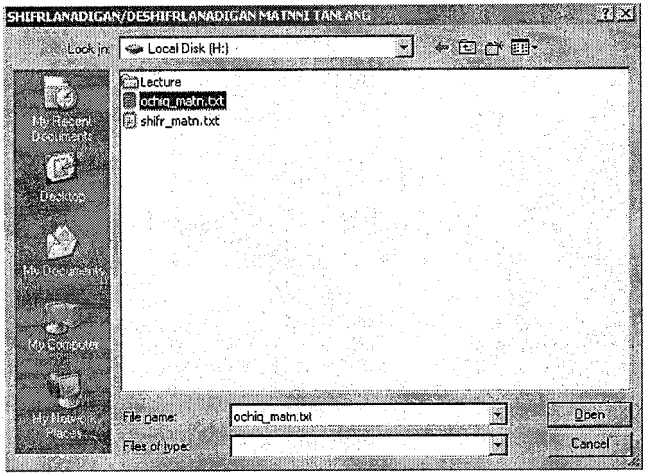


4.24-расм. **Асосий ойнада жойлашган раунд калитлари.**

Шундан сўнг калит генерация қилиш жараёни якунланиб, матнни шифрлаш ёки дешифрлаш жараёни бошланади.

4.10.3. Криптоалгоритм шифрлаш функцияси визуал дастурий таъминоти ва унинг ишлаш принципи

AES FIPS 197 стандарт шифрлаш алгоритмининг визуал дастурий таъминотида маълумотларни шифрлаш учун **FAYL OCHISH** тугмаси босилади ва очиқ маълумот танланади.

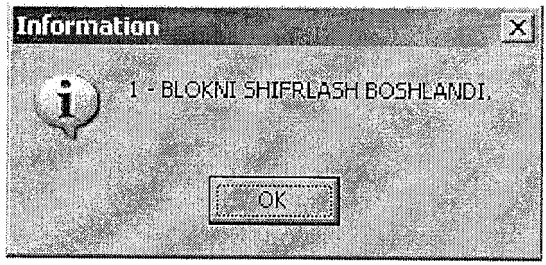


4.25-расм

Очиқ маълумот танлангандан сўнг, раунд калитлари генерация қилингач, **SHIFRLASH** тугмаси босилади ва шифрлаш жараёнига ўтилади.

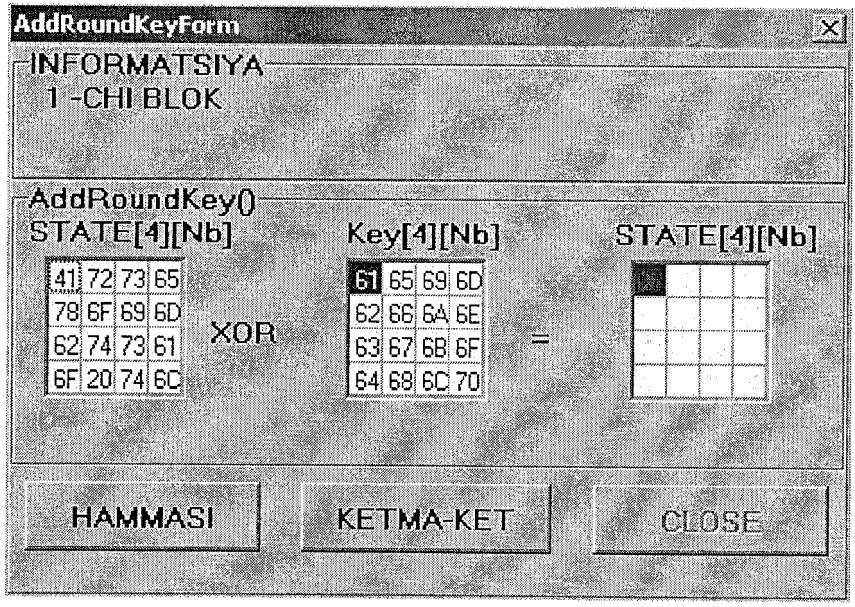
4.26-расм. **Шифрлаш режимлари.**

Агарда **KETMA-KET** тугмаси босилса, шифрлаш жараёнининг барча раундлари босқичма-босқич амалга оширилади. **KETMA-KET** тугмаси босилгандан сўнг **Information** ойнаси ҳосил бўлади. Биринчи блокни шифрлашни бошлаш учун **OK** тугмасини босиш керак.



4.27-расм. **Биринчи блокни шифрлаш ойнаси.**

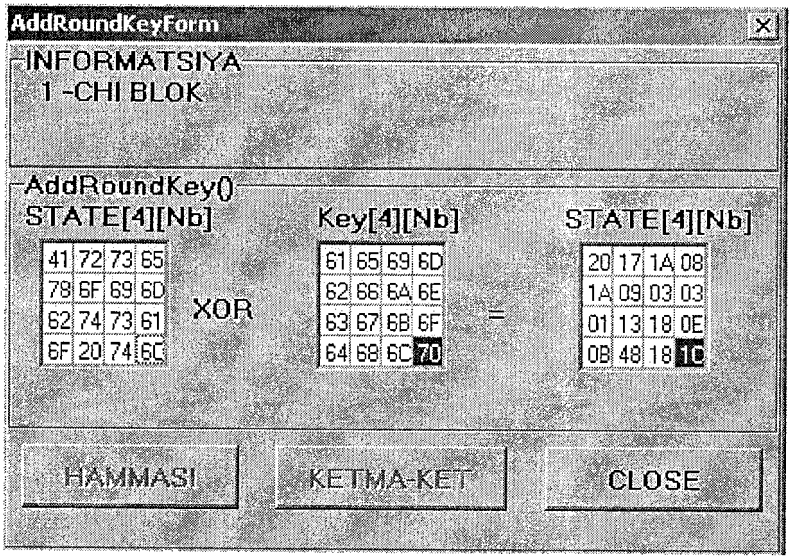
OK тугмаси босилгандан сўнг, **AddRoundKeyForm** ойнаси чиқади.



4.28-расм. **AddRoundKeyForm ойнаси.**

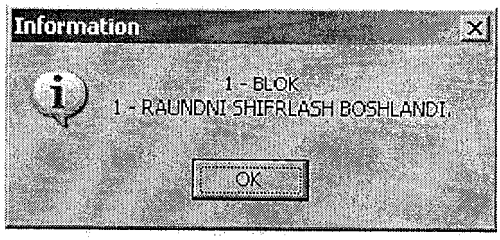
AESFORM асосий ойнасида эса ҳар бир раундда бажариладиган раунд акслантиришларига кириш ва ундан чиқиш жадваллари, раунд калитлари берилиб борилади.

AddRoundKeyForm ойнасида очиқ маълумотнинг биринчи блокига бошланғич раунд калити **XOR** амали билан қўшилади. Натижа учинчи (STATE[4] [Nb] жадвалда келтирилади. Натижани олиш учун **KETMA-KET** ёки **HAMMASI** тугмаларидан бири босилади. Агар **HAMMASI** тугмаси босилса, учинчи жадвалда XOR амалининг тўла натижаси чиқади ва CLOSE тугмаси босилади. Агар **KETMA-KET** тугмаси босилса, учинчи жадвалда XOR амали натижасининг ҳар бир байти алоҳида-алоҳида **KETMA-KET** тугмасини босиш орқали ишлаб чиқилади ва **CLOSE** тугмаси босилади.



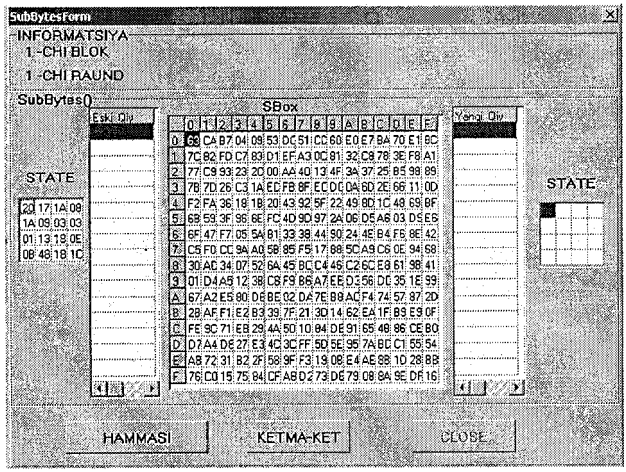
4.29-расм. **Add RoundKeyForm ойнаси.**

Тугма босилгандан сўнг, шифрлашнинг биринчи раундини бошлаш тўғрисидаги **Information** ойнаси чиқади.



4.30-расм. **Биринчи раундни бошлаш ойнаси.**

OK тугмасини босгандан сўнг, **SubBytesForm** ойнаси чиқади.

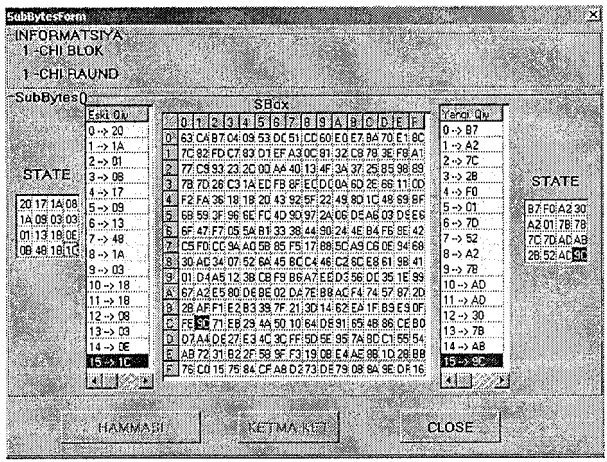


4.31-расм. **SubBytesForm ойнаси.**

SubBytesForm ойнасининг бошланғич қиймати сифатида **AddRound KeyForm** ойнасининг натижавий қиймати олинади.

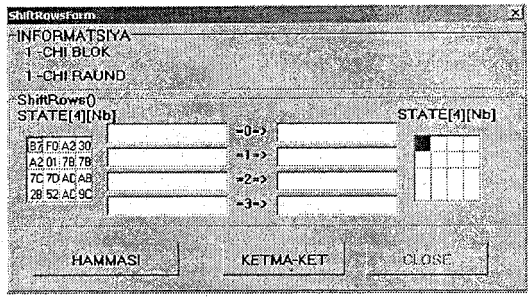
Агар **HAMMASI** тугмаси босилса, ***SubBytes*** раунд акслантиришига кириш жадвалининг барча байтлари автоматик тарзда ***SubBytes*** раунд акслантиришидан ўтказилади ва чиқиш жадвалига ёзилади.

Агар **KETMA-KET** тугмаси босилса, ***SubBytes*** раунд акслантиришига кириш жадвалининг ҳар бир байти алоҳида-алоҳида **KETMA-KET** тугмасини босиш орқали, SubBytes раунд акслантиришидан ўтказилади, натижаси эса чиқиш жадвалига ёзиб борилади. Чиқиш жадвали тўлдирилгандан сўнг, **CLOSE** тугмаси босилади.



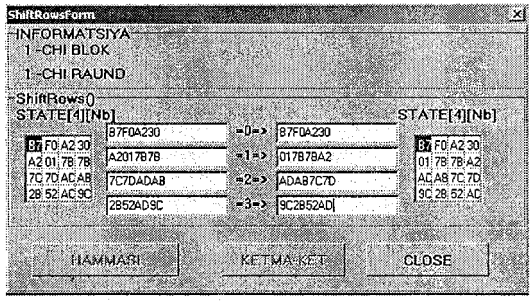
4.32-расм. **SubBytesForm ойнаси.**

CLOSE тугмаси босилгандан сўнг, ShiftRowsForm ойнаси чиқади. **ShiftRowsForm** ойнасидаги кириш жадвали **SubBytesForm** ойнасидаги чиқиш жадвалидан олинади.



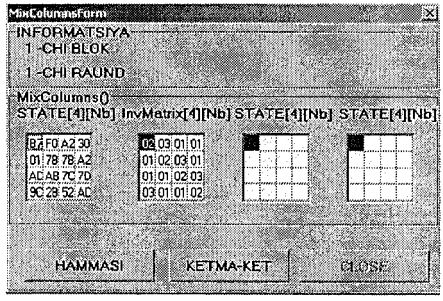
4.33-расм. **ShiftRowsForm ойнаси.**

Бу ерда ҳам **HAMMASI** ёки **KETMA-KET** тугмаларини босиш орқали чиқиш жадвали олинади ва **CLOSE** тугмаси босилади.



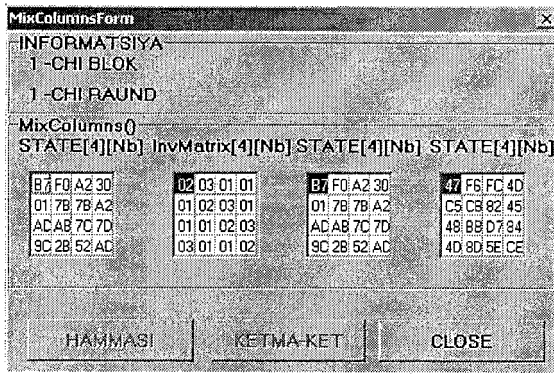
4.34-расм. **ShiftRowsForm ойнаси.**

CLOSE тугмаси босилгандан сўнг, **MixColumnsForm** ойнаси чиқади.



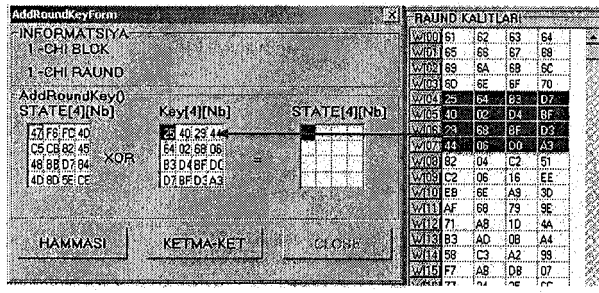
4.35-расм. **MixColumnsForm ойнаси.**

MixColumnsForm ойнасидаги кириш жадвали **ShiftRowsForm** ойнасидаги чиқиш жадвалидан олинади.



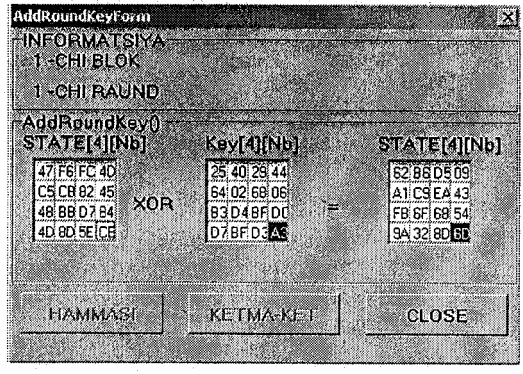
4.36-расм. **MixColumnsForm ойнаси.**

Чиқиш жадвали олингандан сўнг, **CLOSE** тугмаси босилади. Натижада **AddRoundKeyForm** ойнаси чиқади. Бу ерда раунд калити: W(4), W(5), W(6), W(7) калитлари ҳисобланади.



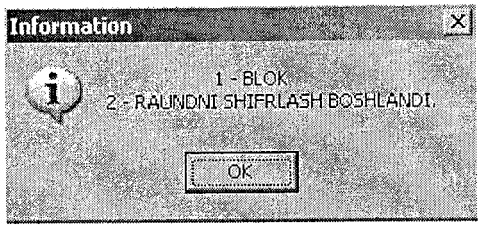
4.37-расм. **AddRoundKeyForm ойнаси.**

Бу ерда ҳам **HAMMASI** ёки **KETMA-KET** тугмаларини босиш орқали чиқиш жадвали олинади.

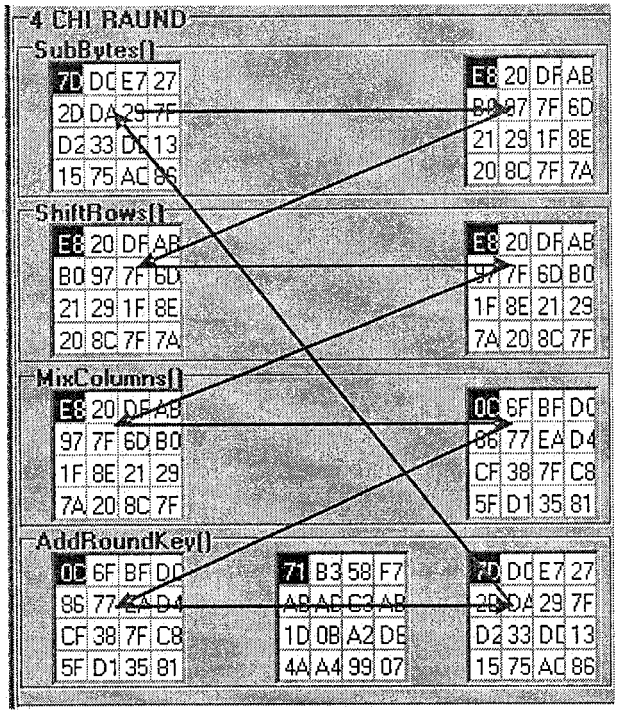


4.38-расм. **AddRoundKeyForm ойнаси.**

Чиқиш жадвали олингандан сўнг, **CLOSE** тугмаси босилади. Натижада шифрлашнинг иккинчи раундини бошлаш тўғрисидаги **Information** ойнаси чиқади.

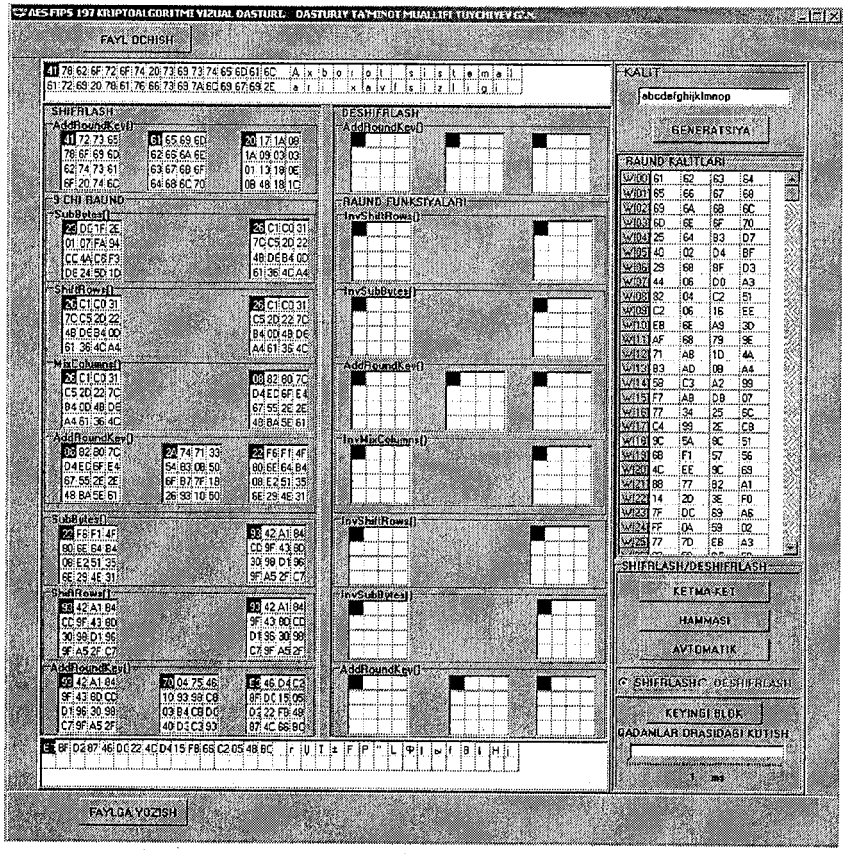


AddRoundKeyForm ойнасининг натижаси **SubBytesForm** ойнасининг кириш қиймати ҳисобланади. Ушбу ҳолат тўққизинчи раундгача юқорида баён қилинганидек давом этади. **SubBytesForm**, **ShiftRowsForm**, **MixColumnsForm** ва **AddRoundKeyForm** ойналарининг кириш ва чиқиш қийматлари ва уларнинг боғлиқлиги:



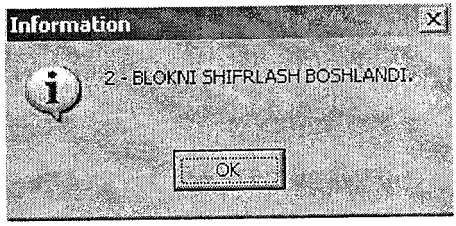
4.39- расм. **SubBytesForm, ShiftRowsForm, MixColumnsForm ва AddRoundKeyForm ойналарининг боғланиши.**

Тўққизинчи раунддан сўнг, **AddRoundKeyForm** ойнасининг чиқиш қиймати **SubBytesForm** ойнасининг кириш қийматига берилади. **SubBytesForm** ойнасининг чиқиш қиймати **ShiftRowsForm** ойнасининг кириш қийматига, бу ойнанинг чиқиш қиймати **AddRoundKeyForm** ойнасинг кириш қийматига берилади. Биринчи блок шифрмаълумоти сифатида **AddRoundKeyForm** ойнасининг кириш қиймати олинади.



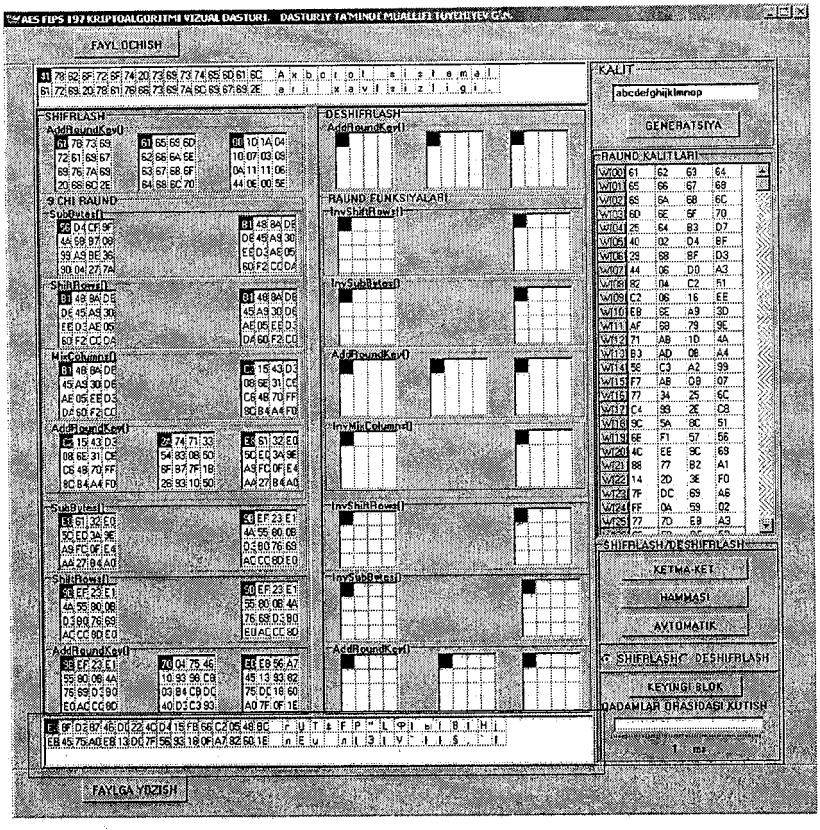
4.40-расм.

Биринчи блок шифрланиб бўлгандан сўнг, иккинчи блокка ўтилади. Бу **KEYINGI BLOK** тугмасини босиш орқали амалга оширилади. **KEYINGI BLOK** тугмаси босилгандан сўнг, **KETMA-KET** ёки **HAMMASI** тугмасини босиш орқали иккинчи блок шифрланади. Иккинчи блок шифрланишдан олдин **Information** ойнаси чиқади.



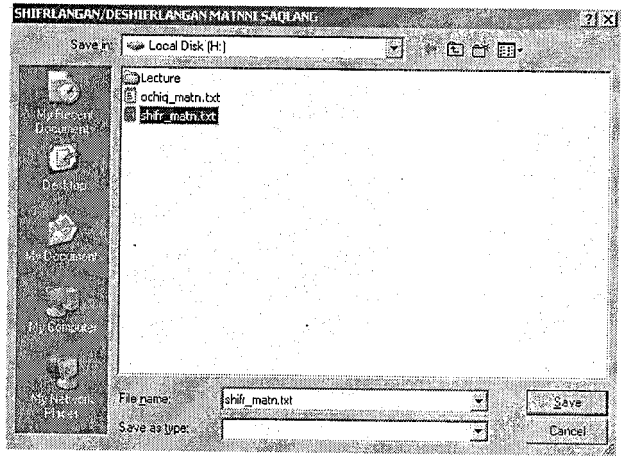
4.41-расм

Иккинчи блокни шифрлаш ҳам худди биринчи блокни шифрлаш каби амалга оширилади ва шифрмаълумот иккинчи блокка ёзилади.



4.42-расм

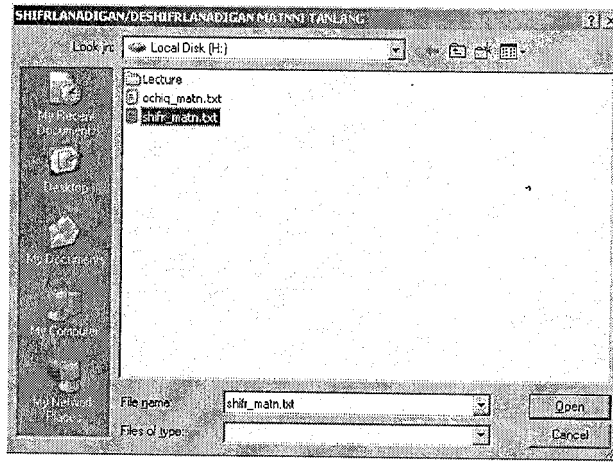
Блоклар шифрланиб бўлгандан сўнг, шифрмаълумот файлга сақланади. Бу эса **FAYLGA YOZISH** тугмасини босиш орқали амалга оширилади.



4.43-расм.

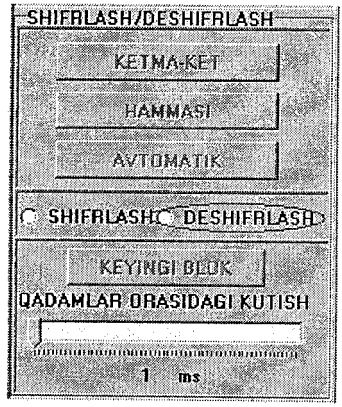
4.10.4. Криптоалгоритм дешифрлаш функцияси визуал дастурий таъминоти ва унинг ишлаш принципи

AES FIPS 197 стандарт шифрлаш алгоритмининг визуал дастурий таъминотида матнларни дешифрлаш учун **FAYL OCHISH** тугмаси босилади ва шифрмаълумот танланади



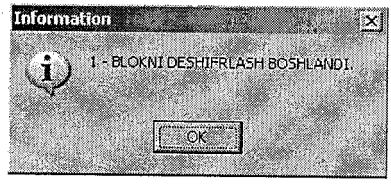
4.44-расм.

Шифрмаълумот танлангандан сўнг, раунд калитлари генерация қилиниб, **DESHIFRLASH** тугмаси босилади ва дешифрлашга ўтилади.



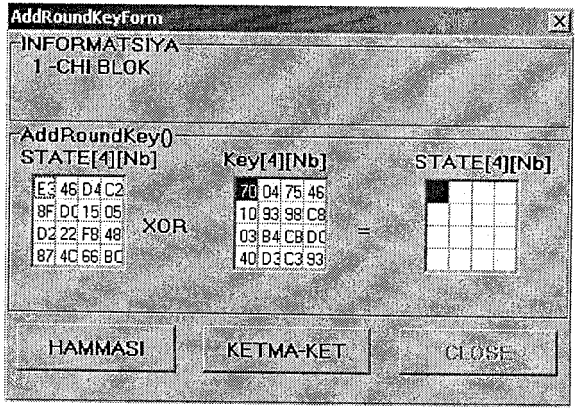
4.45-расм.

Агар **KETMA-KET** тугмаси босилса, дешифрлаш жараёнининг барча раундлари босқичма – босқич амалга оширилади ва **Information** ойнаси ҳосил бўлади. Дастур биринчи блокни дешифрлашни бошлаш учун **OK** тугмасини босиш кераклигини таъкидлайди, яъни



4.46-расм.

OK тугмаси босилгандан сўнг, **AddRoundKeyForm** ойнаси чиқади.

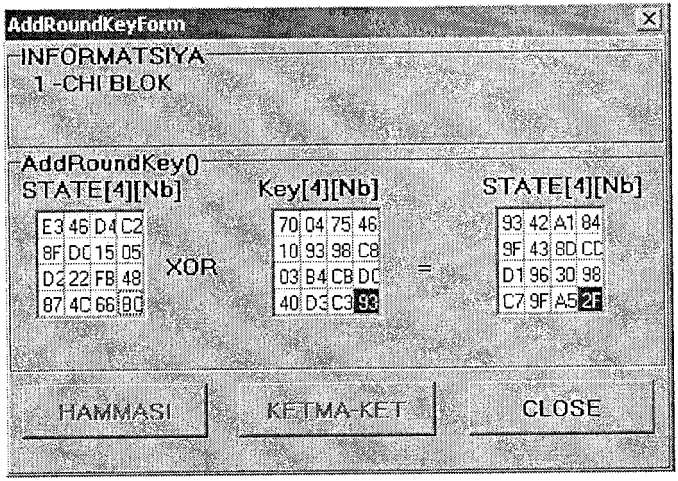


4.47-расм. **AddRoundKeyForm ойнаси.**

AESFORM асосий ойнасида эса ҳар бир раунда бажариладиган акслантиришларининг кириш ва ундан чиқиш жадваллари, раунд калитлари бериб борилади.

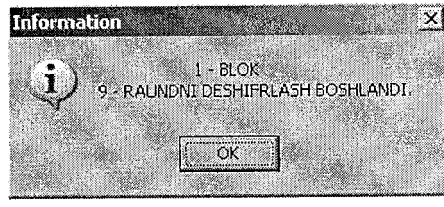
AddRoundKeyForm ойнасида очиқ матннинг биринчи блокига охирги раунд калити XOR амали билан қўшилади.

Натижа учинчи жадвалда келтирилади. Ушбу натижани олиш учун **KETMA-KET** ёки **HAMMASI** тугмаларидан бири босилади. Агар **HAMMASI** тугмаси босилса, учинчи жадвалда XOR нинг натижаси автоматик тарзда чиқади ва **CLOSE** тугмаси босилади. Агар **KETMA-KET** тугмаси босилса, учинчи жадвалда XOR нинг натижасининг ҳар бир байти алоҳида – алоҳида **KETMA-KET** тугмасини босиш орқали ишлаб чиқилади ва **CLOSE** тугмаси босилади.

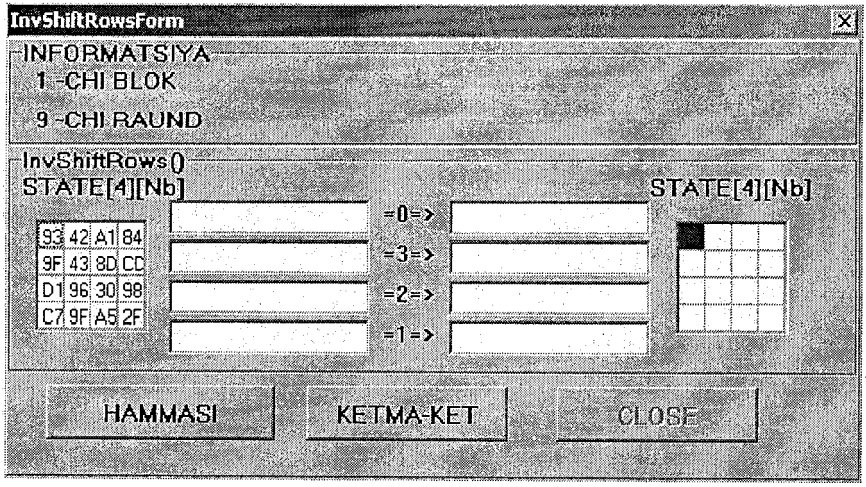


4.48-расм. **AddRoundKeyForm ойнаси.**

Тугма босилгандан сўнг шифрлашни биринчи раундини бошлаш тўғрисидаги **Information** ойнаси чиқади.

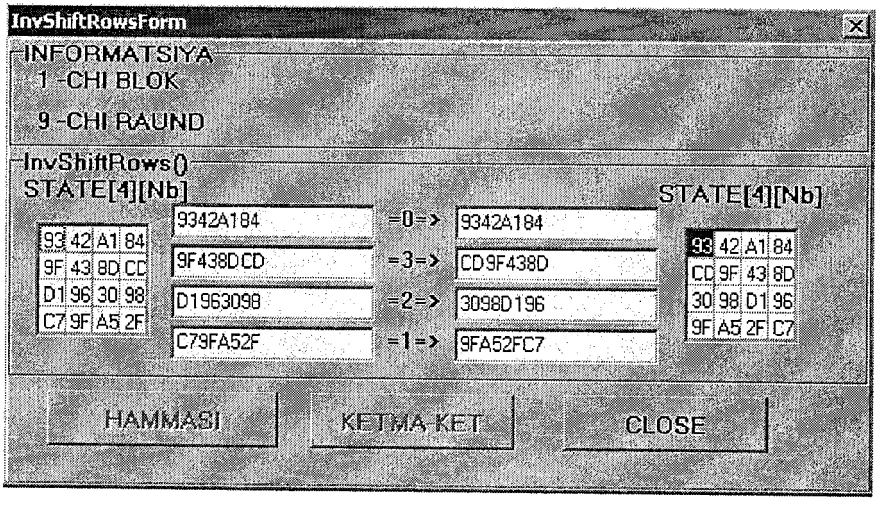


OK тугмасини босилгандан сўнг, **InvShiftRowsForm** ойнаси чиқади.



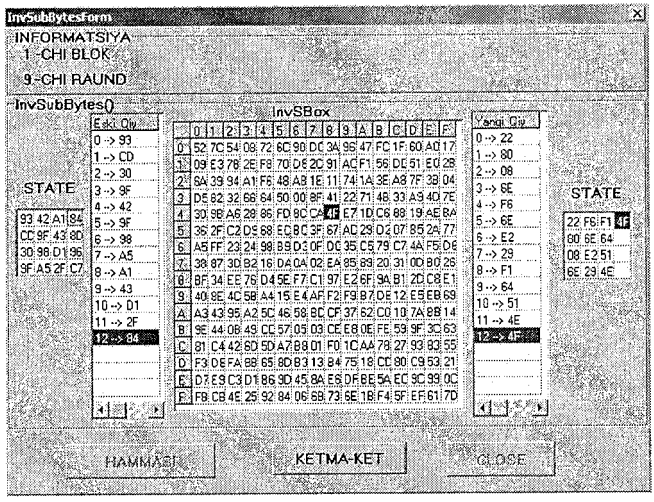
4.49-расм. **InvShiftRowsForm ойнаси.**

InvShiftRowsForm ойнасининг бошланғич қиймати сифатида **AddRoundKeyForm** ойнасининг натижавий қиймати олинади. Бу ерда ҳам **HAMMASI** ёки **KETMA-KET** тугмаларини босиш орқали чиқиш жадвали олинади ва **CLOSE** тугмаси босилади.



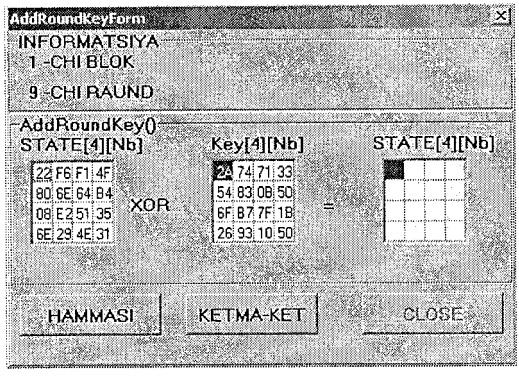
4.50-расм.

CLOSE тугмаси босилгандан сўнг, **InvSubBytesForm** ойнаси чиқади.



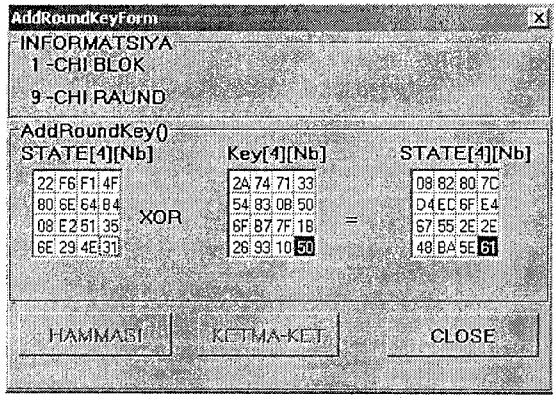
4.51-расм.

Агар **HAMMASI** тугмаси босилса, ***InvSubBytes*** раунд акслантиришига кириш жадвалининг барча байтлари автоматик тарзда ***InvSubBytes*** раунд акслантиришидан ўтказилади ва чиқиш жадвалига ёзилади. Агар **KETMA-KET** тугмаси босилса, ***InvSubBytes*** раунд акслантиришига кириш жадвалининг ҳар бир байти алоҳида-алоҳида **KETMA-KET** тугмасини босиш орқали ***InvSubBytes*** раунд акслантиришидан ўтказилади, натижа эса чиқиш жадвалига ёзиб борилади. Чиқиш жадвали тўлдирилгандан сўнг, **CLOSE** тугмаси босилади ва **AddRoundKeyForm** ойнаси чиқади. **AddRoundKeyForm** ойнасидаги кириш жадвали **InvSubBytesForm** ойнасидаги чиқиш жадвалидан олинади.



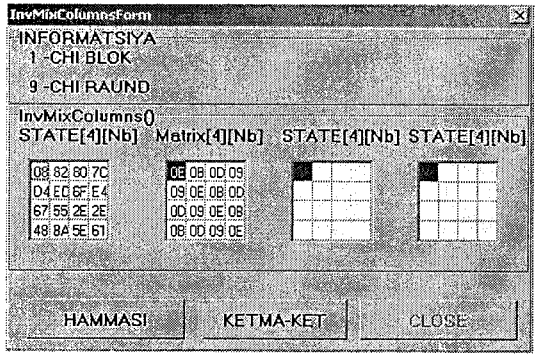
4.52-расм.

Бу ерда ҳам **HAMMASI** ёки **KETMA-KET** тугмаларини босиш орқали чиқиш жадвали олинади ва **CLOSE** тугмаси босилади.



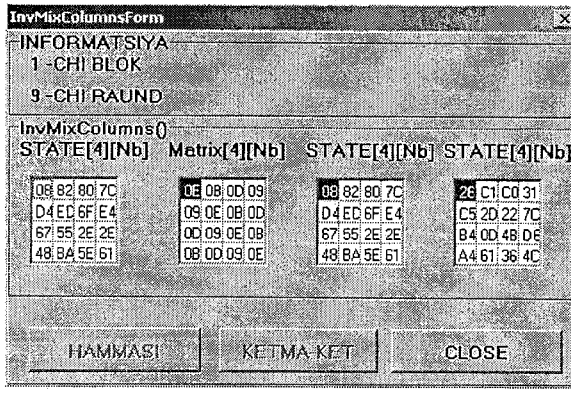
4.53-расм.

CLOSE тугмаси босилгандан сўнг, **InvMixColumnsForm** ойнаси чиқади.



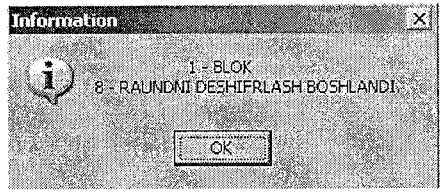
4.54-расм.

InvMixColumnsForm ойнасидаги кириш жадвали **AddRoundKeyForm** ойнасидаги чиқиш жадвалидан олинади.



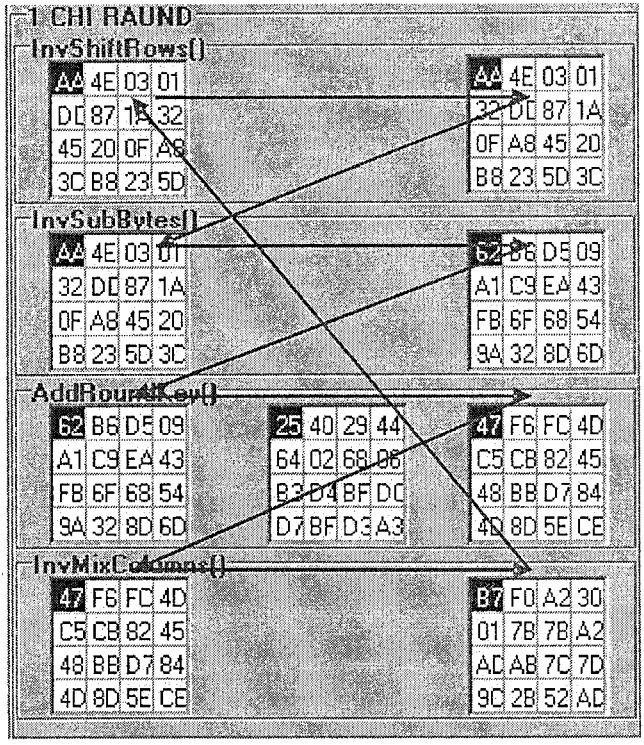
4.55-расм.

Чиқиш жадвали олингандан сўнг, **CLOSE** тугмаси босилади. Тугма босилгандан сўнг, шифрлашнинг иккинчи раундини бошлаш тўғрисидаги **Information** ойнаси чиқади.



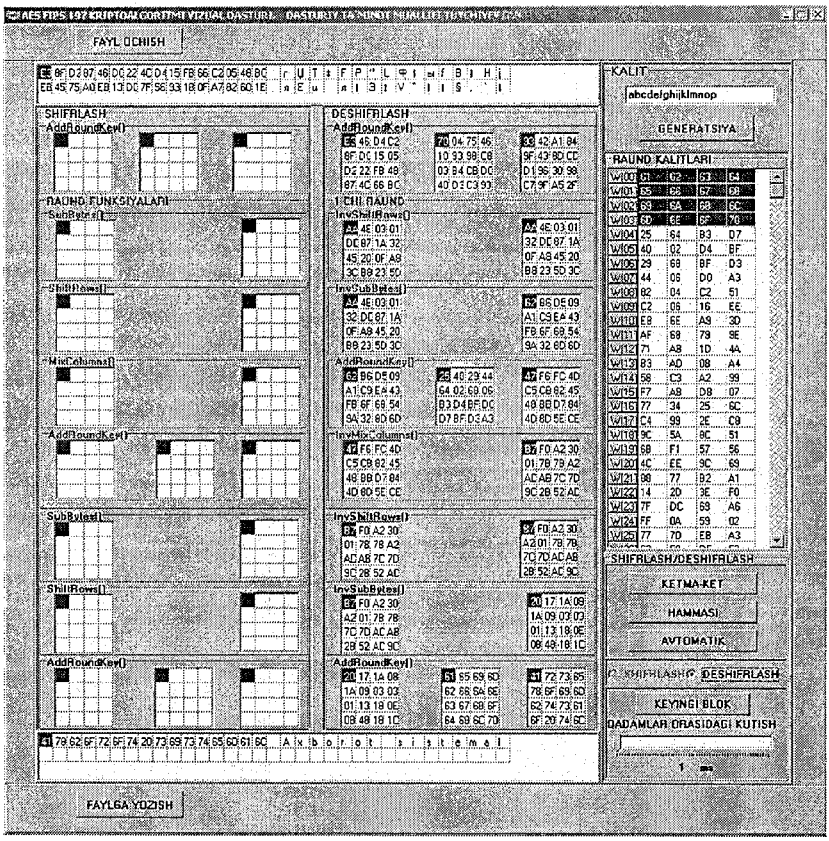
4.56-расм.

InvMixColumnsForm ойнасининг натижаси **InvShiftRowsForm** ойнасининг кириш қиймати ҳисобланади. Бу ҳолат биринчи раундгача давом этади. Қуйидаги расмда **InvShiftRowsForm**, **InvSubBytesForm**, **AddRoundKeyForm** ва **InvMixColumnsForm** ойналарининг кириш ва чиқиш қийматлари ва уларнинг боғлиқлиги кўрсатилган.



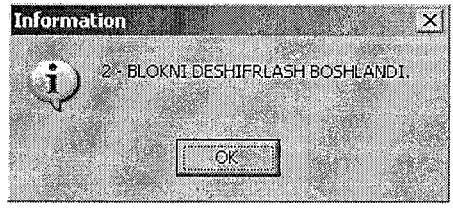
4.57-расм

Биринчи раунддан сўнг, **AddRoundKeyForm** ойнасининг чиқиш қиймати **InvShiftRowsForm** ойнасининг кириш қийматига берилади. **InvShiftRowsForm** ойнасининг чиқиш қиймати эса **InvSubBytesForm** ойнасининг кириш қийматига берилади. Ўз навбатида бу ойнанинг чиқиш қиймати **AddRoundKeyForm** ойнасининг кириш қийматига берилади. Биринчи блок очиқ маълумот сифатида **AddRoundKeyForm** ойнасининг чиқиш қиймати олинади.



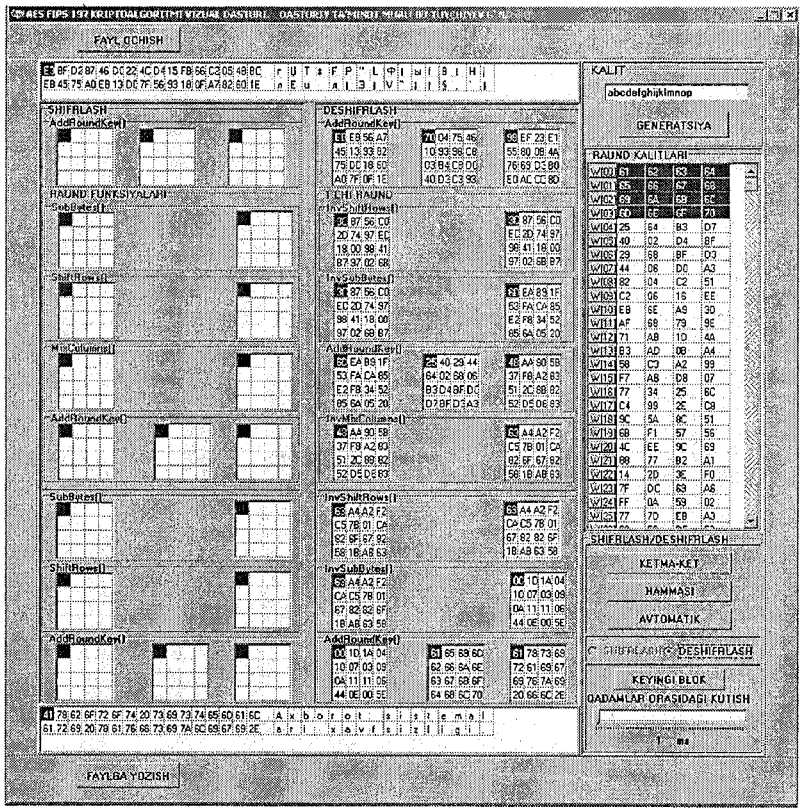
4.58-расм.

Биринчи блок дешифрланиб бўлгандан сўнг, иккинчи блокка ўтилади. Бу **KEYINGI BLOK** тугмасини босиш орқали амалга оширилади. **KEYINGI BLOK** тугмаси босилгандан сўнг, **KETMA-KET** ёки **HAMMASI** тугмасини босиш орқали иккинчи блок шифрланади. Иккинчи блок дешифрланишидан олдин **Information** ойнаси чиқади.



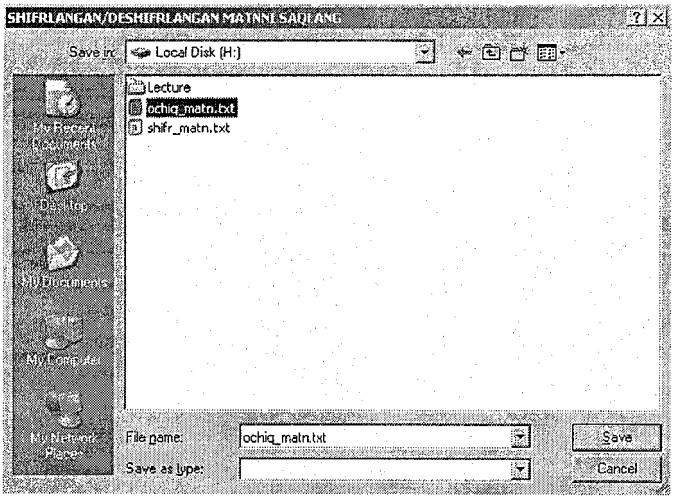
4.59-расм.

Иккинчи блокни шифрлаш ҳам худди биринчи блокни дешифрлаш каби амалга оширилади ва натижавий очиқ маълумот иккинчи блокка ёзилади.



4.60-расм.

Блоклар дешифрланиб бўлгандан сўнг, очиқ маълумот файлга ёзиб қўйилади. Бу эса **FAYLGA YOZISH** тугмасини босиш орқали амалга оширилади.



4.61-расм.

§4.11. Фейстел тармоғига асосланмаган янги симметрик калитли блокли шифрлаш алгоритми ҳақида

2000 йилда АҚШнинг «Стандартлар ва Технологиялар Миллий Институти (NIST)» томонидан давлат аҳамиятига молик муҳим маълумотларни криптографик муҳофазасини таъминлаш мақсадида, 1974 йилдан мавжуд бўлган **DES** –стандарт шифрлаш алгоритми ўрнига, Фейстел тармоғига асосланмаган янги **AES-FIPS 197** – стандарт шифрлаш алгоритми қабул қилинганлигини расмий равишда эълон қилинди. Қуйида эътиборингизга ҳавола қилинадиган янги маълумотларни криптографик муҳофазасини таъминловчи симметрик калитли блокли шифрлаш алгоритми ҳам Фейстел тармоғига асосланмаган бўлиб, қўлингиздаги китоб муаллифининг шу йўналишда олиб борган изланишлари натижасидир.

Ҳавола этилаётган алгоритм маълумотлар блокларининг мос битларини $\oplus$ – XOR (mod 2) амали билан қўшиш, характеристикаси 256 бўлган (mod 256) чекли майдонда аниқланган матрицали акслантириш, S – блок ва жадвалли сиқиш акслантиришлари асосида 128 (такомиллашган вариантида $128+32l$, $l=1,2,\ldots$, d, $d<\infty$) битли очиқ маълумот блокларини 128 (такомиллашган вариантида $128+32l$) битли калитлар билан шифрлашни итератив ҳолда амалга оширилишини таъминлайди.

Алгоритмнинг акслантириш жараёнларини ёритишда қуйидаги белгилашлардан фойдаланилган:

– k – 128 (такомиллашган вариантда $128+32l$) битли калит, яъни $k=k_1k_2...k_{128}$ (такомиллашган вариантда $k=k_1k_2...k_{128}k_{129}...k_{128+32l}$);

– T_0 – 128 (такомиллаштирилганда $128+32l$) –битли (разрядли) очиқ маълумот блоклари, бу ерда $l=1,2,\ldots, d$, $d<\infty$;

– $T_ш$ – 128 (такомиллаштирилганда $128+32l$) – битли (разрядли) шифрланган маълумот блоклари;

– t_i очиқ маълумот битлари кетма-кетлигининг i – бити;

– $A_{n\times4}$ раунд калитларини генерация қилишда фойдаланиладиган тўғри тўртбурчакли матрица, бу матрица элементлари $k=k_1k_2...k_{128}$ -калит кетма-кетлигидан бир томонлама функция асосида генерация қилиниб махфий бўлиши мумкин ёки калитга боғлиқ бўлмаган ҳолда генерация қилиниб, узоқ муддатли калит сифатида махфий ёки очиқ бўлиши мумкин, ҳамда $n=2m$, $m=3, ..., M$, $M<\infty$, матрица элементлари $a_{ij}(i=1, ..., n; j=1, 2, 3, 4;)$ бир байтдан иборат $0\le a_{ij}\le 255$;

– $xd_{4\times1}=(xd_1, xd_2, xd_3, xd_4)$ – 32-битли (тўртта саккиз битли) $A_{n\times4}$ – матрицали акслантиришга кирувчи блок, бу ерда xd_i – байтларнинг қийматлари $0\le xd_i\le 255$, $i=1, 2, 3, 4$, бўлиб, $d=1, 2, ..., q$, q – бирор чекли сон;

– $A_{4\times4}$ квадрат матрица, дастлабки калит кетма-кетлигидан бир томонлама функция асосида генерация қилиниб махфий бўлиши мумкин ёки калитга боғлиқ бўлмаган ҳолда генерация қилиниб, узоқ муддатли калит сифатида махфий ёки очиқ бўлиши мумкин, унинг детерминанти қиймати $\|A_{n\times4}\|\neq 0$ тоқ сон, элементлари $a_{ij}(i=1, 2, 3, 4; j=1, 2, 3, 4;)$ бир байтли $0\le yd_i\le 255$;

– $yd_{n\times1}=(yd_1, yd_2, ..., yd_n)$ – раунд калитларини генерация қилишда ишлатиладиган тўғри тўртбурчакли матрицали аклантириш $A_{n\times4}x_{4\times1}$ $\bmod 256$ натижаси, яъни $yd_{n\times4}=A_{n\times4}xd_{4\times1}$ (mod256) бу ерда yd_i – байтлар, $0\le t_{ij}\le 255$, $i=1, 2, \ldots, n$;

– $t_{4\times4}$(такомиллаштирилганда $t_{4\times l}$, бу ерда $l=1, 2, ..., T$, $T<\infty$) –матрицали $A_{4\times4}$ акслантиришга кирувчи 128 (такомиллаштирилганда $128+32l$) – битли вектор, бу ерда t_{ij} – байтларнинг қийматлари $0\le t_{ij}\le 255$, $i=1, 2, 3, 4; j=1, 2, \ldots, l$;

– S – блок (дастлабки калит блоки кетма-кетлигидан бир томонлама функция асосида генерация қилиниб махфий бўлиши мумкин ёки калитга боғлиқ бўлмаган ҳолда генерация қилиниб, узоқ муддатли калит сифатида махфий ёки очиқ бўлиши мумкин) акслантириш, бир байтли S_0, S_2, ..., S_{255} – тугунлардан ташкил топган, бир байтли кириш ва чиқиш акслантиришларидан иборат бўлган 1×256 ўлчамли жадвал:

S_0	S_1	S_2	...	S_{255}

бу ерда: $0 \le S_1, S_2, ..., S_{256} \le 255$, $S_i \ne i$ ва $S_i \ne S_j$, агар $i \ne j$ бўлса, S_i – сонларнинг қийматлари $0 \le 2S_i \le 255$;

– $\oplus$ – блокларнинг мос битларини mod2 (2 модуль бўйича) қўшиш;

– $zd_{n\times1}=(zd_1, zd_2, ..., zd_n)$ – вектор, тўғри тўртбурчакли матрицали акслантириш натижаси бўлган – $yd_{n\times1}=(yd_1, yd_2, ..., yd_n)$ – вектор координаталарининг блок орқали акслантириш натижаси, яъни – $zd_{n\times1}=S-(yd_{n\times1})$, бу ерда zd_i – байтлар бўлиб, $0 \le zd_i \le 255$, i=1, 2, ..., n;

– СЖ –сиқиш жадвали, ўлчови 16x16 (махфий, калит билан биргаликда узатилади ёки калитдан берилган биртомонлама функция асосида генерация қилинади), ундан раунд калитларини генерация қилишда фойдаланилади, жадвалнинг ҳар бир қатор, устун ва бош диоганалларида тўрт бит билан ифодаланувчи 0 дан 15 гача бўлган q_{ij} – бутун сонлар бир мартадан такрорланган ҳолда жойлашган, яъни $0 \le q^{ij} \le 15$, $i=0, ..., 15$, j=0, ..., 15:

q_{00}	q_{01}	...	$q_{0,15}$
q_{10}	q_{11}	...	$q_{1,15}$
...	...	...	...
$q_{15,0}$	$q_{15,1}$	...	$q_{15,15}$

– $wd_{n\times1}=(wd_1, wd_2, wd_3, ..., wd_4)$ – сиқиш акслантириши натижаси, 32 -битли (4-байтли) блок вектори.

Санаб ўтилган асосий белгилашлардан ташқари ёрдамчи белгилашлардан ҳам фойдаланилади.

Таъкидланганидек, квадрат матрица $A_{4\times1}$, тўғри тўртбурчакли матрица $A_{n\times1}$ (бу ерда $n=2^m$, $m=2, ..., M$, $M \le \infty$), S – блок ва СЖ калит битлари кетма-кетлигидан берилган биртомонлама функция асосида генерация қилинади. Санаб ўтилган акслантиришлар ва раунд калитларини генерация қилиш қоидаларини кўриб ўтамиз.

1. Матрицали акслантириш $A_{4\times4}$ элементлари сони 16 та ҳамда уларнинг ҳар бири $a_{ij}(i=1, 2, 3, 4; j=1, 2, 3, 4;)$ бир байтдан иборат бўлиб, детерминантининг қиймати нолдан фарқли $\|A_{4\times4}\| \ne 0$ ва $\|A_{4\times4}\|$ – тоқ сон. Бундай матрица учбурчакли матрица кўринишида бўлиб, калит $k=k_1k_2...k_{128}$ (такомиллаштирилганда $k=k_1k_2...k_{128}k_{129}...k_{128+321}$) битлари кетма-кетлигидан қуйидагича хосил қилинади:

1) Кетма-кетлик $k=k_1k_2...k_{128}$ (такомиллаштирилганда) байтларга ажратилади:

$k1=k_1...k_8$, $k2=k_9...k_{16}$, ..., $k16=k_{121}...k_{128}$,

(такомиллаштирилганда

$k1=k_1...k_8$, $k2=k_9...k_{16}$, ..., $k16...k_{121}...k_{128}$,

$k17=k_{129}...k_{136}$, $k18=k_{137}...k_{144}$, ..., $k[16+4l]=k_{(15+4l)8+1}...k_{(15+4l)8+8}$);

2) a_{11}, a_{22}, a_{33}, a_{44} – диагонал элементлари ушбу $k1$, $k2$, ..., $k16$ (такомиллаштирилганда $k1$, $k2$, ..., $k16$, $k17$, ..., $k[16+4l]$) кетма-кетликнинг дастлабки тўртта тоқ қийматли элементларидан олинади;

3) Агарда, бу кетма-кетликдан тўртта тоқ қийматли байтнинг ҳаммаси олинмаса, у ҳолда калит k циклик равишда $\lambda=1, 2, \ldots 255$ битларга сурилиб, 1) и 2) бандлардаги жараёнлар такрорланган ҳолда тоқ қийматли диагонал элементлар ҳосил қилинади;

4) Агар, шунда ҳам тўртта тоқ қийматли диогонал элементлари хосил бўлмаса, у ҳолда ушбу $y=x^z \bmod 257$, (бу ерда $z=\text{const}$ калит билан биргаликда узатилувчи ўзгармас) формула орқали 0 дан 256 гача бўлган сонлар кетма-кетлигининг $\{0 \le y \le 256 : y=x^z \bmod 257,\ z=\text{const},\ 0 \le x \le 256\}$ тўплами ҳосил қилинади, чунки -туб сон;

5) Сўнгра $\{0 \le r \le 256 : r=y \bmod 257,\ 0 \le y \le 256\}$ тўпламнинг биринчи ва кейинги тоқ қийматли элементлари билан матрица диагонал элементларининг қолган қийматлари аниқланади;

6) Учбурчакли матрицанинг нолдан фарқли бўлган диагоналдан қуйи ёки юқори бўлган барча элементлари $k1$, $k2$, ..., $k16$ (такомиллаштирилганда $k1$, $k2$, ..., $k16$, $k17$, ..., $k[16+4l]$) кетма-кетликдан ёки уларнинг бирор λ бит суриш билан хосил қилинган кетма-кетликдан олинади.

2. Матрицали акслантириш $A_{n\times4}$ элементлари сони $4n$ та, ҳар-бир элемент $a_{ij}(i=1, ..., n; j=1, 2, 3, 4;)$ бир байтдан иборат.

Агар $m=2$ бўлса, $A_{n\times4}=A_{2^m\times4}$ матрица квадратик, яъни $A_{4\times4}$ бўлиб, у ҳар бири 8 битли 16 та элементга эга. Бундай ҳолда 256- битли калит k циклик равишда чапга λ битга (бу ерда $3 \le \lambda \le 255$ бўлган ихтиёрий тоқ сон) сурилади ва $a_{ij}(i=1, 2, 3, 4; j=1, 2, 3, 4;)$ элементлар циклик суриш натижасида хосил бўлган блокнинг дастлабки 128 битини 16 та байтларга ажратилган қисмблокларидан олинади, мисол учун, λ – байтни a_{11}, 2-байтни a_{12} ва ҳоказо 16-байтни a_{44} деб олинади. Сўнгра, ўзаро солиштириш билан матрицанинг ҳар бир сатрида ақалли битта тоқ қийматли элемент бўлиши ва матрицанинг барча элементлари ҳар-хил қийматли бўлиши ҳамда ихтиёрий 2 та устуни пропорционал бўлиши таъминланади.

Агар $m=3$ бўлса, $A_{n\times4}=A_{2^m\times4}$ матрица тўғри тўртбурчакли, яъни $A_{8\times4}$ бўлиб, у ҳар бири 8 битли 32 та элементга эга. Бунда 256- битли калит циклик равишда чапга битга (бу ерда $3\leq\leq 255$ бўлган ихтиёрий тоқ сон) сурилади ва элементлар циклик суриш натижасида хосил бўлган блок битларини 32 та байтларга ажратилган қисмблокларидан олинади, мисол учун, λ – байтни a_{11}, 2-байтни a_{12} ва ҳоказо 32-байтни $a_{8\times4}$ деб олинади. Сўнгра, ўзаро солиштириш билан матрицанинг ҳар бир сатрида ақалли битта тоқ қийматли элемент бўлиши ва матрицанинг барча элементлари ҳар-хил қийматли бўлиши ҳамда ихтиёрий 2 та устуни пропорционал бўлиши таъминланади.

Агар $m=4$ бўлса, $A_{n\times4}=A_{2\mathrm{m}_{\times4}}$ матрица тўғри тўртбурчакли, яъни $A_{16\times4}$ бўлиб, у ҳар бири 8 битли 64 та элементга эга. Бунда 256- битли калит k циклик равишда чапга λ битга (бу ерда $3\leq\lambda\leq255$бўлган ихтиёрий тоқ сон) сурилади ва $a_{ij}(i=1, 2, 3, 4, ..., 8; j=1, 2, 3, 4;)$ элементлар циклик суриш натижасида ҳосил бўлган блок битларини 32 та байтларга ажратилган қисмблокларидан олинади, мисол учун, 1-байтни a_{11}, 2-байтни a_{12} ва хоказо 32-байтни $a_{8\times4}$ деб олинади. Циклик сурилган калит блоки яна λ битга сурилиб, янги ҳосил бўлган 256 -битли блок байтларга ажратилади ҳамда уларда $a_{ij}(i=9, ..., 16; j=1, 2, 3, 4;)$ элементлар: 1-байтни a_{91}, 2-байтни a_{92} ва хоказо 32-байтни $a_{16,4}$ деб олинади. Сўнгра, ўзаро солиштириш билан матрицанинг ҳар бир сатрида ақалли битта тоқ қийматли элемент бўлиши ва матрицанинг барча элементлари ҳар-хил қийматли бўлиши ҳамда ихтиёрий 2 та устуни пропорционал бўлиши таъминланади.

Юқорида келтирилган жарённи бир неча марта такрорлаш натижасида $n2^m$ $(m=2, ..., M, M<\infty)$ етарли даражада катта бўлганда ҳам $A_{n\times4}$ матрицанинг барча элементларини хосил қилиш мумкин.

3. S – блок генерацияси қуйидагича амалга оширилади. Дастлабки 128 (такомиллаштирилганда $128+32l$) – битли k – калит 16 та (такомиллаштирилганда $16 + 4l$ та) байтга ажратилиб, улар жуфт-жуфти билан солиштирилади, 16 тадан (такомиллаштирилганда $16+4l$ тадан) кўп бўлмаган жуфт-жуфти билан ҳар-хил бир байтли элементлар билан S – блок ячейкалари тўлдирилади. Сўнгра да стлабки k-калит $\lambda=1$ битга сурилиб, ҳосил бўлган кетма-кетлик байтларга ажратилиб, бу байтларни S – блок ячейкалари тўлдирилган байтлар билан биргаликда жуфт-жуфти билан солиштириб, жуфт-жуфти билан ҳар-хил байтлар билан блокнинг қолган ячейкаларини тўлдириш давом эттирилади. S – блок ячейкаларини тўлдириш дастлабки калитни кетма-кет, $\lambda=2, 3, ..., 127$, битларга сурилиб, жуфт-жуфти билан ҳар-хил байтлар орқали амалга оширилади. Агарда $\lambda=127$ битга суриш натижасидан сўнг ҳам S –

блок ячейкалари тўлмаса, у ҳолда ушбу $y=x^z \bmod 257$, (бу ерда $z=\text{const}$. калит билан биргаликда узатилувчи ўзгармас) формула орқали 0 дан 256 гача бўлган сонлар кетма-кетлигининг $\{0 \le y \le 256 : y=x^z \bmod 257, z=\text{const}, 0 \le x \le 256\}$ тўплами хосил қилинади (чунки $n=257$ – туб сон) ҳамда $\{0 \le \gamma \le 255 : \gamma = y \bmod 256, 0 \le y \le 256\}$ – тўпламнинг блок ячейкаларидаги байтлардан фарқли бўлган элементлари билан тўлмай қолган ячейкалар охиригача тўлдирилади.

4. СЖ генерацияси қуйидагича амалга оширилади. Дастлабки 128 (такомиллаштирилганда $128+32l$) – битли k – калит 32 та (такомиллаштирилганда $32+8l$ та) ярим байтга ажратилиб, улар жуфт-жуфти билан солиштирилади, 32 тадан (такомиллаштирилганда $32+8l$ тадан) кўп бўлмаган жуфт-жуфти билан ҳар хил бўлган ярим байтли элементлар билан СЖ ячейкалари ҳар бир сатрда, устунда ва бош диогоиналда 0 дан 15 гача бўлган сонлар бир мартадан қатнашадиган қилиб кетма-кет тўлдирилади. Сўнгра дастлабки k – калит $\lambda=1$ битга сурилиб, ҳосил бўлган кетма-кетлик ярим байтларга ажратилиб, бу ярим байтли блокларни СЖ ячейкалари тўлдирилган ярим байтли элементлар билан биргаликда жуфт-жуфти билан солиштириб, жуфт-жуфти билан ҳар-хил ярим байтли блоклар билан СЖ қолган ячейкаларини тўлдириш давом эттирилади. СЖ ячейкаларини тўлдириш дастлабки калитни кетма-кет $\lambda=2, 3, ..., 127$, битларга сурилиб, жуфт-жуфти билан ҳар-хил ярим байтлар орқали амалга оширилади. Агарда $\lambda=127$ битга суриш натижасидан сўнг ҳам СЖ ячейкалари тўлмаса, у ҳолда ушбу $y=x^z \bmod 17$, (бу ерда $z=\text{const}$. калит билан биргаликда узатилувчи ўзгармас) формула орқали 0 дан 16 гача бўлган сонлар кетма-кетлигининг $\{0 \le y \le 16 : y=x^z \bmod 17, z=\text{const}, 0 \le x \le 16\}$ – тўплами хосил қилинади (чунки $n=17$ туб сон) ҳамда -тўпламнинг элементлари билан СЖ ҳар бир сатри, устуни ва бош диагоналида 0 дан 15 гача бўлган сонлар бир мартадан такрорланадиган қилиб тўлмай қолган ячейкалар охиригача тўлдирилади.

Таъкидлаш жоизки, $A_{n\times 4}$ – учбурчакли квадрат матрицани, $A_{n\times 4}$ – тўғри тўртбурчакли матрицани, S – блок ва СЖ –сиқиш жадвалларини дастлабки калитдан генерация қилинмай, уларни олдиндан генерация қилиниб, калит билан биргаликда узатилиши мумкин.

Шифрлашда очиқ маълумот 128 (такомиллаштирилганда $128+32l$) – битли блокларга ажратилади. Ҳар бир 128 (такомиллаштирилганда $128+32l$) – битли блокни шифрлаш 8 раунддан иборат. Дастлабки 8 та раунд калитларини кетма-кет раунд калитларини хосил қилиш жараёни бир томонлама акслантиришларидан ўтказиб янги 8 раунд калитларини хосил қилиниб, шифрлаш жараёни раундлари сонини 16 та, ёки шу тариқа 24, 32, 40 ва ҳоказо раундга кўпайтириш мумкин.

Ҳар бир i – раунднинг k_{pi}–калити 128 (такомиллаштирилганда $128+32l$) – битли $k=k_1k_2...k_{128}$ (такомиллаштирилганда $k=k_1k_2\ ...\ k_{128}$ $k_{129}\ ...\ k_{128+32l}$) – калитни $i\lambda$ битга чапга циклик суриб, хосил бўлган $k'=k'_1k'_2...k'_{128}$ (такомиллаштирилганда $k'=k'_1k'_2...k'_{128}k'_{129}.....k'_{128+32l}$) – блокни 16 та (такомиллаштирилганда $16+4l$ та) байтга ажратиб

$$k1=k'_1\ ...\ k'_8,\ k2=k'_9\ ...\ k'_{16},\ ...,\ k16=k'_{121}\ ...\ k'_{128}$$

(такомиллаштирилганда

$$k1=k'_1\ ...\ k'_8,\ k2=k'_9...k'_{16},\ ...,\ k16=k'_{121}\ ...\ k'_{128}$$

$$k17=k'_{129}\ ...\ k'_{136},\ k18=k'_{137}\ ...\ k'_{144},\ ...,\ k[16+4l]=k'_{(15+4l)8+1}\ ...\ k'_{(15+4l)8+8}:$$

1) олинган байтлар 32-битли сўзларга бирлаштирилади:

$$x1_{4\times1}=(x1_1,\ x1_2,\ x1_3,\ x1_4)=(k1,\ k2,\ k3,\ k4),\ x2_{4\times1}=(x2_1,\ x2_2,\ x2_3,\ x2_4)=$$
$$=(k5,\ k6,\ k7,\ k8),$$
$$x3_{4\times1}=(x3_1,\ x3_2,\ x3_3,\ x3_4)=(k9,\ k10,\ k11,\ k12),\ x2_{4\times1}=(x4_1,\ x4_2,\ x4_3,\ x4_4)=$$
$$=(k13,\ k14,\ k15,\ k16),$$

2) бу векторлар $xd_{4\times1}$ (бу ерда $d=1,\ 2,\ ..,\ 4+l$) тўғри тўртбурчакли матрица $A_{n\times4}$ билан акслантирилади, яъни $yd_{n\times4}=A_{n\times4}xd_{4\times1}(\text{mod } 256)$;

3) акслантириш натижасида хосил бўлган векторлар $yd_{n\times1}$ (бу ерда $d=1,\ 2,\ ...,\ 4+l$), S – блок орқали акслантирилади: $yd_1,\ ...,\ yd_n$ – байтларнинг ўнлик санoқ тизимидаги $(yd_1)_{10},...,\ (yd^n)_{10}$ – қийматлари бўйича S – блок ячейкасининг тартиб рақамлари аниқланиб, $yd_1,\ ...,\ yd_n$ – байтларнинг акслантириш натижаси сифатида, уларга мос келувчи $S_{yd_1},...,\ S_{yd_n}$ – байтларнинг иккилик санoқ тизимидаги ифодаси олинади $(S_{yd_1}),\ ...,\ (S_{yd_n})_2$:

$zd_{n\times1}=S(yd_1,\ ...,\ yd_n)=(S(yd_1),\ ...,\ S(yd_n))=(S(_{yd_1})_2,\ ...,\ (S(_{yd_n})_2)=(zd_1,\ ...,\ zd_n)$;

4) СЖ бўйича $8\times n$ – битли (n–байтли) векторлар $zd_{n\times1}$, $d=1,\ 2,\ ...,\ 4+1,\ 32$ – битли (4-байтли) векторга $w_{4\times1}=(w_1,\ w_2,\ w_3,\ w_4)$ сиқилади:

– $zd_{n\times1}$ – векторнинг ҳар бир байти zd_i ярим байтларга ажратилади, яъни $zd_{n\times1}=(zd_1,\ ...,\ zd_n)=(z'd_1,\ ...,\ z'd_{2n})=z'd_{2n\times1}$;

– ярим байтларнинг ва ўнлик санoқ тизимидаги $(z'd_1)_{10}$ и $(z'd_{2n})_{10}$ қийматлари бўйича СЖ мос сатр ва устунлари аниқланиб, уларнинг кесишган жойидаги $q(z'd_1)_{10}(z'd_{2n})_{10}$ – ярим байт $z'd_1$ ва $z'd_{2n}$ – ярим байтларнинг сиқиш натижаси ҳисобланади. Бу жараён барча

$(z'd_2, z'd_{2n-1})$, $(z'd_3, z'd_{2n-2})$, ..., $(z'd_n, z'd_{n+1})$ – жуфтликлар учун такрорланади, яъни. Барча $(z'd_i, z'd_{2n-(i-1)})$, – жуфтликлар учун, бу ерда $i=1,\ ...,\ n$;

– олдинги қадамдаги каби СЖ ни $2(m-2)$ марта қўллаб, тўлиқ сиқиш амалга оширилади, яъни $wd_{4\times1}=(wd_1,\ wd_2,\ wd_3,\ wd_4)$ – 32-битли (4-байтли) вектор олинади;

5) шу тариқа олинган кетма-кетлик

$$(w1_{4\times1},\ w2_{4\times1},\ ...,\ w[4+l]_{4\times l})=$$
$$=(w1_1,\ w1_2,\ w1_3,\ w1_4,\ w2_1,\ w2_2,\ w2_3,\ w2_4,\ ...,\ w[4+l]_1,\ w[4+l]_2,\ w[4+l]_3,\ w[4+l]_4,)=$$
$$=k_1(pi)\ k_2(pi)\ ...\ k_{128}(pi)\ ...\ k_{128+32l}(pi))=k_{pi},$$

($l=0$ бўлганда кетма-кетлик $k_1(pi)$ $k_2(pi)$..... $k_{128}(pi)=k_{pi}$) раунд калитини ташкил этади.

Шифрлаш алгоритми жараёни бошида шифрланиши керак бўлган битлар блоки $T_0=(t_1(0), t_2(0), ..., t_{128}(0))$ (такомиллаштирилганда $T_0=(t_1(0), t_2(0), ..., t_{128}(0), ..., t_{128+32l}(0))$) битлари XOR амали билан 128 (такомиллаштирилганда $128+32l$) – битли дастлабки калитнинг (такомиллаштирилганда $k=k_1, k_2 \dots k_{128}\ k_{129} \dots k_{128+32l}$), мос битларига қўшилади, яъни

$$T_0 \oplus k=(t_1(0), t_2(0), ..., t_{128}(0)) \oplus (k_1 k_2 \dots k_{128}),=$$
$$=(t_1(0)\oplus k_1)\ (t_2(0)\oplus k_2) \dots (t_{128}(0)\oplus k_{128})=(t_1(1)\ t_2(1) \dots t_{128}(1)=T_1$$

(такомиллаштирилганда

$$T_0 \oplus k=(t_1(0), t_2(0), ..., t_{32}(0), t_{33}(0), ..., t_{128+32l}(0)) \oplus (k_1 k_2 \dots k_{128} k_{129} \dots k_{128+32l}),=$$
$$=(t_1(0)\oplus k_1)\ (t_2(0)\oplus k_2) \dots (t_{128+32l}(0)\oplus k_{128+32l})=t_1(1)\ t_2(1) \dots t_{128+32l}(1)=T_1).$$

Биринчи раундда қуйидаги акслантиришлар амалга оширилади:

1) битлар кетма-кетлиги $T_1=t_1(1)\ t_2(1) \dots t_{128}(1)$
(такомиллаштирилганда $T_1=t_1(1)\ t_2(1) \dots t_{128}(1)\ t_{129}(1)\ t_2(1) \dots t_{128+32l}(1)$)
ушбу $t1=t_1(1) \dots t_8(1)$, $t2=t_9(1) \dots t_{16}(1)$, ..., $t16=t_{121}(1) \dots t_{128}(1)$
(такомиллаштирилганда

$$t1=t_1(1) \dots t_8(1),\ t2=t_9(1) \dots t_{16}(1), ..., t16=t_{121}(1) \dots t_{128}(1)$$
$$t17=t_{129}(1) \dots t_{136}(1),\ t18=t_{137}(1) \dots t_{144}(1), ..., t[16+4l]=t_{(15+4l)8+1}(1) \dots t_{(15+4l)8+8}(1)),$$

байтларга ажратилади;

2) ҳосил бўлган ti – байтлардан қуйидаги матрица ташкил этилади

$$X_{4\times4}=\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix}=\begin{pmatrix} t1 & t5 & t9 & t13 \\ t2 & t6 & t10 & t14 \\ t3 & t7 & t11 & t15 \\ t4 & t8 & t12 & t16 \end{pmatrix},$$

такомиллаштирилганда

$$X_{4\times4+l}=\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{1,4+l} \\ x_{21} & x_{22} & x_{23} & x_{2,4+l} \\ x_{31} & x_{32} & x_{33} & x_{3,4+l} \\ x_{41} & x_{42} & x_{43} & x_{4,4+l} \end{pmatrix}=\begin{pmatrix} t1 & t5 & t9 & t13 \dots & t[13+4l] \\ t2 & t6 & t10 & t14 \dots & t[14+4l] \\ t3 & t7 & t11 & t15 \dots & t[15+4l] \\ t4 & t8 & t12 & t16 \dots & t[16+4l] \end{pmatrix};$$

3) матрица $X_{4\times4}$ (такомиллаштирилганда матрица $X_{4\times4+l}$) учбурчакли квадрат матрицабилан характеристикаси $Y_{4\times4}=A_{4\times4}X_{4\times4}\ (\mathrm{mod}\,256)$ бўлган чекли майдонда акслантирилади, яъни (такомиллаштирилганда $Y_{4\times4}=A_{4\times4}X_{4\times4}\ (\mathrm{mod}\,256)$);

4) $Y_{4\times4}$(такомиллаштирилганда $Y_{4\times4+l}$) – матрицанинг y_{ij} – элементлари S – блок орқали акслантирилади: $z_{ij}=S(y_{ij})=(S_{y_{ij}})_2$;

5) байтларнинг $z_{11}z_{21}z_{31}z_{41}z_{21}z_{22}z_{23}z_{24}\dots z_{44}$ (такомиллаштирилганда z_{11} $z_{21}z_{31}z_{41}z_{21}z_{22}z_{23}z_{24}\dots z_{44}z_{51}\dots z_{(16+4l)}$) кетма-кетлигидан ташкил топган z_1z_2 $\dots z_{128}$ (такомиллаштирилганда $z_1z_2\dots z_{128}z_{129}\dots z_{128+32l}$) – битлар кетма-кетлиги биринчи раунд калити $k_{p1}=k_1(p1)\ k_2(p1)\dots\ \dots\ \dots\ k_{128}(p1)$ (такомиллаштирилганда $k_{p1}=k_1(p1)\ k_2(p1)\ \dots\ k_{128}(p1)$, $k_{129}(p1)\ \dots\ k_{128+32l}$ $(p1)$) мос битларига XOR амали билан қўшилади, яъни

$z_1z_2\dots z_{128}\oplus k_1(p1)\ k_2(p1)\dots\ \dots\ \dots\ k_{128}(p1)=(z_1\oplus k_1(p1)\,(z_2\oplus k_2(p1)\dots(z_{128}\oplus k_{128}(p1))=t_1(2)\ t_2(2)\ \dots\ t_{128}(2)=T_2.$

(такомиллаштирилганда

$z_1z_2\dots z_{128}z_{129}\dots z_{128+32l}\oplus k_1(p1)\ k_2(p1)\ \dots\ k_{128}(p1)\ k_{129}(p1)\ \dots.\ k_{128+32l}(p1)=$

$=(z_1\oplus k_1(p1))\ (z_2\oplus k_2(p1)\ \dots\ (z_{128}\oplus k_{128}(p1))\ (z_{129+32l}\oplus k_{128+32l}(p1))=$

$=t_1(2)\ t_2(2)\ \dots\ t_{128}(2)\ t_{129}(2)\ \dots\ t_{128+32l}(2)=T_2$

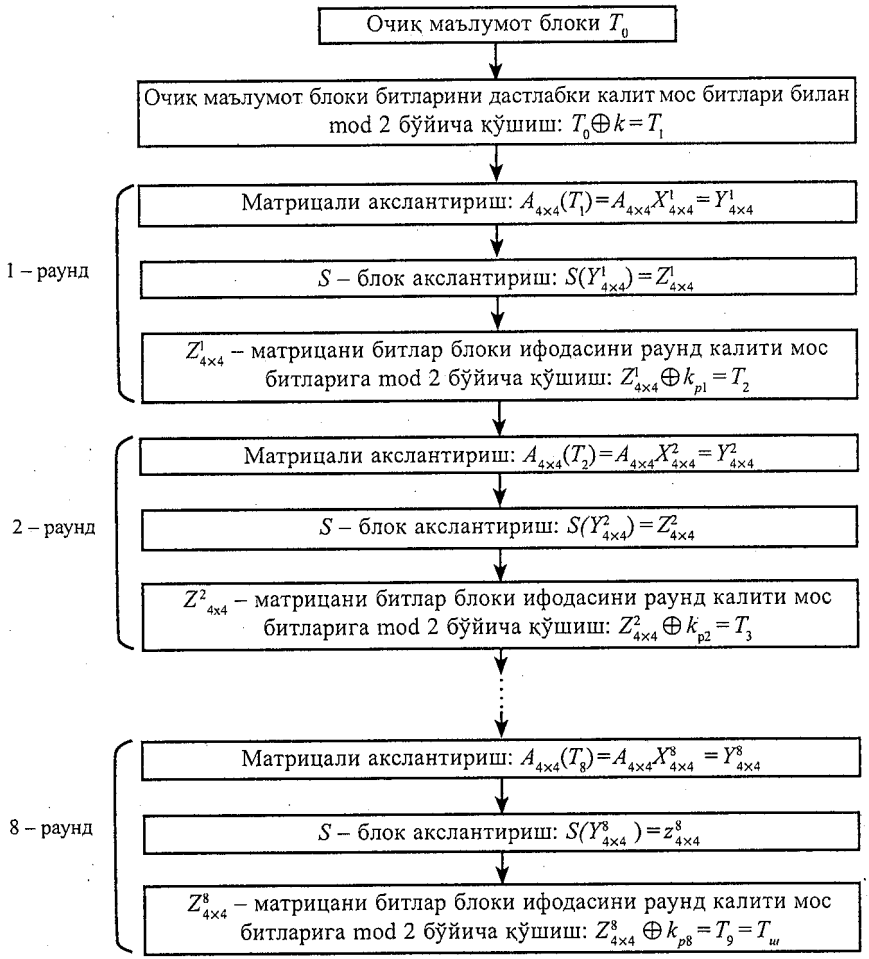
Юқорида келтирилган 1) – 5) – бандлар таклиф этилаётган шифрлаш алгоритмининг 1-раунд акслантиришларини ташкил этади.

Ушбу $T_1=T_2$ ва $k_{p1}=k_{p2}$ амалларни бажариб, сўнгра, юқоридаги 1) – 5) пунктлар акслантиришларини амалга ошириб, 2-раунд бажарилади. Шундай қилиб, агар $(i-1)$ – раунд натижаси олинган бўлса, у ҳолда $T_1=T_{i-1}$, ва $kp_1=k_{pi-1}$ амалларни бажариб, сўнгра, 1) – 5) – бандлар акслантиришларини амалга ошириш натижасида алгоритмнинг i – раунд бажарилади.

Қуйида i – раунд калити генерацияси ва шифрлаш алгоритмнинг умумий блок схемаси келтирилган.

i – раунд калити генерацияси:

Шифрлаш алгоритмининг блок схемаси



Таъкидлаш жоизки, криптобардошлиликни ошириш мақсадида, кейинги T_0 – очиқ маълумот блокини шифрлаш учун дастлабки k – калит λ битга циклик сурилади, бу ерда λ – қиймати 255 дан катта бўлмаган ихтиёрий тоқ сон. Умумий ҳолда, агарда калит узунлиги N бўлса, у ҳолда циклик суриш битлари сонининг қиймати λ, калит узунлигини ифодаловчи N сони билан ўзаро туб бўлиши, дастлабки калитдан максимал сондаги ҳар хил калит блоклари олиш имкониятини таъминлаш, максимал сондаги очиқ маълумот блокларининг ҳар бирини ҳар-хил калитлар билан шифрлашнишини амалга ошириш имкониятини беради.

Алгоритмда қўлланилган: очиқ маълумот блоки битларини дастлабки калит блоки мос битлари билан ва оралиқ натижалар блоклари мос битларини раунд калитининг мос битлари билан mod 2 бўйича қўшиш, mod 256 ўйича матрицали акслантириш, S – блок акслантириш ва сиқиш жадвали акслантиришлари амалий жиҳатдан бир томонлама бўлган функциялар ҳисобланади. Бундан ташқари, сиқиш жадвали акслантириши жараёни чизиқсизлик хусусиятига эга, яъни сиқиш функциясига тескари бўлган кенгайтириш функцияси кўп қийматлилик хусусиятига эга бўлиб, жадвални ташкил этувчи ҳар бир элемент тенг эҳтимоллик билан (16 мартадан) такрорланган. S – блок акслантириш бир қийматли бўлиб, бу жараён функцияси чизиқли, шунинг учун ҳам у дастлабки калитдан бир томонлама функция орқали генерация қилинади ёки калитга боғлиқ бўлмаган ҳолда олдиндан генерация қилиниб, махфий ҳисобланиши мумкин. Очиқ маълумот блоки битларини дастлабки калит блоки мос битлари билан ва оралиқ натижалар блоклари мос битларини раунд калитининг мос битлари билан mod 2 бўйича қўшиш жарёнлари чизиқлилик хусусиятларига эга. Аммо, очиқ маълумот ва калитларнинг номаълумлиги бу акслантиришларнинг амалий жиҳатдан бир томонламалигини таъминлайди. Характеристикаси 256 бўлган чекли майдонда тўғри тўртбурчакли матрицали (mod 256 бўйича) акслантириш чизиқли, аммо жуфт сонлар характеристикаси 256 бўлган чекли майдонда тескарисига эга эмас, бундан ташқари матрицанинг ихтиёрий иккита устуни пропорционал бўлиб, бу акслантиришга тескари акслантириш қуриш имкони йўқ. Алгоритмда иштирок этган акслантиришларнинг санаб ўтилган барча хоссалари унинг бардошлилигини таъминлайди.

Симметрик блокли шифрлаш алгоритмлари учун бардошли калитларни генерация қилиш масаласининг ечими адабиётлар рўйхатида [26] келтирилган ўқув қўлланмасида баён этилиб, криптографик алгоритмларнинг турларига кўра уларнинг ахборот-коммуникация тармоқларида қўлланиши ёритилган.

Шифрланган маълумотни калит маълум бўлганда дешифрлаш алгоритми қуйидагича амалга оширилади:

1. Дастлабки k – калитдан юқорида баён этилган генерация қоидаларига кўра: $A_{4\times4}$ ва $A_{n\times4}$ – матрицалар S – блок, ҳамда раунд калитлари k_{pi} ($i=1, 2, ..., 8$) яратилади.

2. Тескари матрица ҳисобланади:

$$A_{4\times4}^{-1} = \| A \|^{-1} \begin{vmatrix} A_{11} & A_{21} & A_{31} & A_{41} \\ A_{12} & A_{22} & A_{32} & A_{42} \\ A_{13} & A_{23} & A_{33} & A_{43} \\ A_{14} & A_{24} & A_{34} & A_{44} \end{vmatrix},$$

бу ерда: A_{ij} – мос $a_{ij}(i, j=1, 2, \ldots, 4)$ элементларнинг алгебраик тўлдирувчилари, $A_{4\times4}$– матрица генерациясига кўра $A_{4\times4}^{-1}$ – тескари матрица мавжуд ва у учбурчакли.

3. Шифрмаълумот блоки $T_ш = T_9 = t_1(9)t_2(9) \ldots t_{128}(9)$ (такомиллаштирилганда

$T_ш = T_9 = t_1(9)t_2(9) \ldots t_{128}(9)\ t_{128+32l}(9))$ битларига k_{p8} – саккизинчи раунд калитининг мос битлари mod2 бўйича қўшилади: $T_9 \oplus k_{p8} = Z_{4\times4}^8$.

4. $Z_{4\times4}^8$ – матрицанинг $z_{11}z_{21}z_{31}z_{41}z_{21}z_{22}z_{23}z_{24} \ldots z_{44}$ (такомиллаштирилганда $z_{11}z_{21}z_{31}z_{41}z_{21}z_{22}z_{23}z_{24} \ldots z_{44}z_{51} \ldots z_{44}z_{51} \ldots z_{(16+4l)}$ – элементлари S – блок орқали тескари акслантирилади, яъни S – блок ячейкаларидан z_{ij} – элемент топилади, сўнгра элемент жойлашган ячейканинг тартиби бўйича $Y_{4\times4}^8$ – матрицанинг y_{ij} – элементи аниқланади.

5. $Y_{4\times4}^8$ – матрицани $A_{4\times4}^{-1}$ – тескари матрица билан акслантириш натижасида $X_{4\times4}^8$ – матрица олинади, унинг – элементларининг иккилик саноқ тизимидаги (битлар) ифодаси T_8 – блокдан иборат бўлади:

$$A_{4\times4}^{-1}\ Y_{4\times4}^8 = A_{4\times4}^{-1} A_{4\times4} X_{4\times4}^8 = X_{4\times4}^8 = T_8.$$

Таъкидлаш жоизки, 3–5 – бандлар акслантиришлари $T_ш = T_9$ – шифрмаълумот блокини дешифрлаш алгоритмининг 1-раундини ифодалайди.

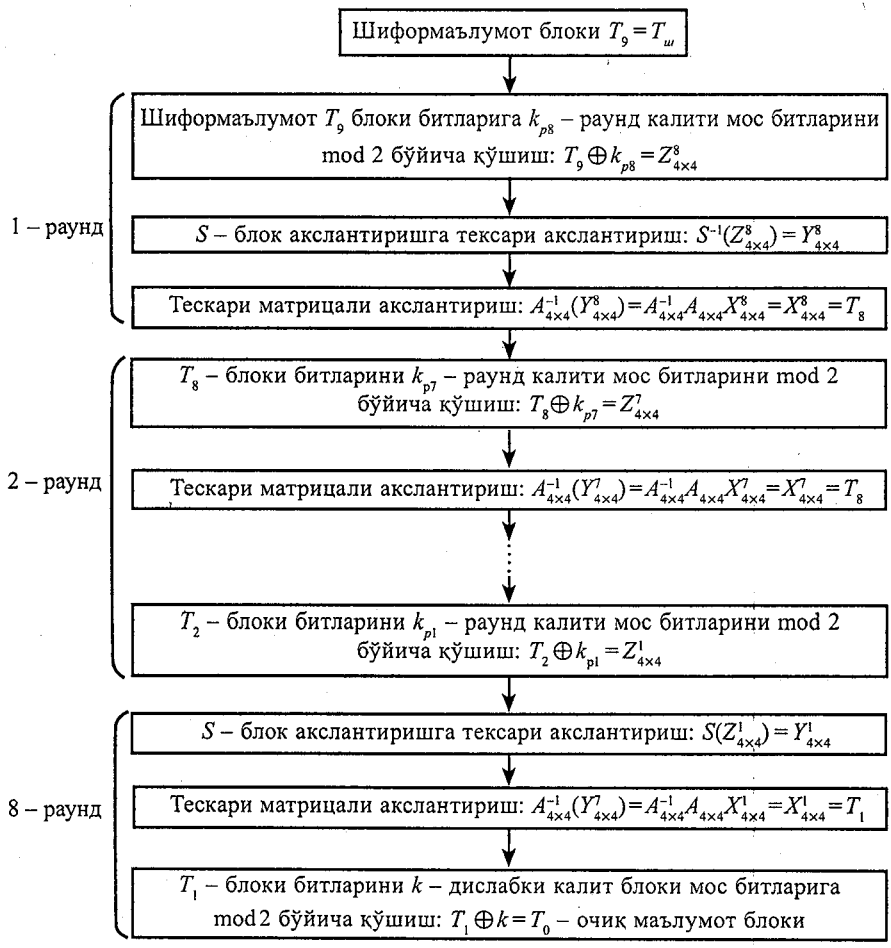
Ушбу $T_9 = T_8$ ва $k_{p8} = k_{p7}$ амалларни бажариб, 3–5 – бандлар акслантиришларини амалга ошириб, дешифрлаш алгоритмининг 2-раунди бажарилади. Шундай қилиб, агар дешифрлаш алгоритмининг i-раунди натижаси T_{9-i} олинган бўлса, у ҳолда $T_9 = T_{9-i}$ ва $k_{p8} = k_{8-pi}$ амалларни бажариб, сўнгра, 3–5 – пунктлар акслантиришларини амалга ошириш натижасида дешифрлаш алгоритмининг $(i+1)$ – раунди бажарилади, бу ерда $i=0, 1, 2, \ldots, 7$. Шифрмаълумот блоки $T_ш = T_9$ учун 3–5 – бандлар акслантиришларини, $T_9 = T_{9-i}$ ва $k_{p8} = k_{8-pi}$ амалларни $i=0, 1, 2, \ldots, 7$, қийматларда бажарган ҳолда такрорлаб, T_1 – блок олинади.

4. Олинган T_1 – блок битларига дастлабки k – калит блоки мос битларини mod2 бўйича қўшиб, очиқ маълумот блоки T_0 олинади, яъни: $T_1 \oplus k = T_0$.

Алгоритмнинг дастурий таминоти C++ тилида тузилиб, унинг тез ва самарали эканлиги аниқланган.

Келтирилган 1–6 – пунктлар дешифрлаш алгоритмини ифодалайди.

Шифрмаълумот блокини дешифрлаш алгоритмининг блок схемаси



Эътиборингизга ҳавола этилган ушбу бобда мавжуд симметрик блокли шифрлаш алгоритмларининг хусусиятлари таҳлил қилиниб, янги алгоритмлар таклиф этилди.

4-боб бўйича хулосалар

Ушбу бобда:

1. Симметрик блокли шифрлаш алгоритмларининг яратилиш заруриятлари, хоссалари ва уларнинг ахборотни муҳофаза қилинишини таъминлашда қўлланилишининг аҳамиятлари тўғрисида илмий асосланган фикр ва мулоҳазалар билдирилди.

2. Аппарат-техник ва аппарат-дастурий криптографик воситаларининг асосини ташкил этувчи, амалий жиҳатдан кенг тарқалган, бугунги кунда ахборот-коммуникация тармоқлари тизимида ахборот мухофазасини таъминлашда самарали натижалар бериб келаётган, Фестел тармоғи деб аталувчи таркибий тузилмага эга бўлган мавжуд шифрлаш алгоритмлари батафсил таҳлил қилинди.

3. Фейстел тармоғига асосланган мавжуд шифрлаш алгоритмларини таҳлили натижасида, тармоқнинг асосий акслантиришлари мавжуд шифрлаш алгоритмлари акслантиришларидан фарқли бўлган, криптографик бардошлилиги юқори, аппарат-техник ва аппарат-дастурий қўлланилиши қулай ҳамда самарали, таккомиллаштириш асосий акслантиришларни ўзгартирмай фақат калит узунлигини оширишга боғлиқ бўлган янги шифрлаш алгоритми таклиф этилди.

4. Фейстел тармоғига асосланмаган 2000 йили АҚШда маълумотларни шифрлаш алгоритми стандарти сифатида қабул қилинган AES-FIPS-197 алгоритми атрофлича таҳлил қилинди.

5. AES-FIPS-197 алгоритми каби, Фейстел тармоғига асосланмаган, криптобардошлилиги юқори, асосий акслантиришлари аппарат-техник ва аппарат-дастурий қўлланилиши жиҳатидан мавжуд шифрлаш алгоритмларидан самарали бўлган, таккомиллаштириш асосий акслантиришларни ўзгартирмай фақат калит узунлигини оширишга боғлиқ бўлган янги шифрлаш алгоритми таклиф этилди

Кейинги бобда ахборот-коммуникация тармоқларида маълумотларнинг муҳофазасини таъминловчи аппарат-техник ва аппарат-дастурий воситаларда кенг ва самарали қўлланиладиган узлуксиз шифрлаш алгоритмлари таҳлил қилинади.

V БОБ
УЗЛУКСИЗ ШИФРЛАШ АЛГОРИТМЛАРИНИНГ ХОССАЛАРИ

Симметрик блокли шифрлаш алгоритмлари каби, узлуксиз (оқимли) шифрлаш алгоритмларининг яратилиши ҳам табиий зарурият асосида вужудга келган. Нисбатан кичик узунликка эга бўлган, яъни кафолатланган криптобардошлиликни таъминловчи узунликка эга бўлган – бугунги кунда 128 битдан кам бўлмаган калит билан бир томонлама криптографик акслантиришлар асосида, етарли даражада катта узунликдаги псевдотасодифий кетма-кетлик (ПТКК) гаммасини ишлаб чиқарувчи генераторлар негизида узлуксиз шифрлаш алгоритмлари яратилади. Узунлиги 128 битдан кам бўлмаган калитларнинг мумкин бўлган барча вариантлари сони 2^{128} тадан кам бўлмай, уларнинг ҳаммасини танлаб чиқиш жараёнини амалга ошириш, бугунги кун ҳисоблаш техника ва технологияларининг мавжуд илғор имкониятларидан фойдаланилганда ҳар доим ҳам самарали натижалар беравермайди. Ана шундай генераторлар ишлаб чиқарган гамма кетма-кетликни ташкил этувчи алифбо белгиларини очиқ маълумот мос алифбо белгилари билан бирор амал бажариш орқали шифр маълумот алифбоси белгиларига алмаштириш – гаммалаштириш амалга оширилади. Бундай шифрлаш жараёни кўп алифболи ўрнига қўйиш шифрлашни амалга оширишни самарали усулини ифодалайди – кафолатли криптобардошлиликни таъминловчи кичик узунликдаги калит билан, очиқ маълумотнинг частотавий хусусиятларини шифрмаълумотга кўчирмайдиган етарли криптобардошлиликни таъминловчи шифрлашни амалга оширади.

Узлуксиз шифрлаш алгоритмлари асосини ПТКК ишлаб чиқарувчи генераторлар ташкил этади. Бундай генераторларнинг асосий криптобардошлилик характеристикаси ушбу генераторлар ҳосил қилган кетма-кетликнинг тасодифийлигидадир. Ҳосил қилинган кетма-кетликлар блокларининг тасодифийлик даражаси маълум бир критерийлар (мезонлар) орқали баҳоланади. Тасодифийлик даражаси юқори бўлган псевдотасодифий кетма-кетликни ишлаб чиқарувчи генераторлар замонавий криптотизимларнинг ажралмас қисми ҳисобланади. Тасодифий кетма-кетликлар криптографияда қуйидаги мақсадларда қўлланилади:

– симметрик криптотизимлар учун тасодифийлик даражаси юқори бўлган сеанс калитлари ва бошқа калитларнинг генерациясида;

– асимметрик криптотизимларда қўлланиладиган катта қийматлар қабул қилувчи параметрларининг тасодифий бошланғич қийматлари генерациясида;

– блокли шифрлаш алгоритмларининг бошланғич тасодифий қиймат талаб қилувчи CBC, OFB ва бошқа қўлланиш тартибқоидалари учун тасодифийлик даражаси юқори бўлган бошланғич векторлар ҳосил қилишда;

– электрон рақамли имзо тизимларида катта қийматга эга параметрлар учун дастлабки тасодифий қийматларини генерациясида;

– битта протокол орқали бир хил маълумотларни ҳар-хил калитлар қўллаш билан шифрлаб турли кўринишда узатиш учун талаб қилинадиган ҳолатларда калит учун етарли узунликдаги тасодифий кетма-кетлик ҳосил қилишда, масалан SSL ва SET протоколларида.

Ташкил этувчилари тенг эҳтимоллик билан тақсимланган тасодифий кетма-кетлик ҳосил қилиш муаммосини ечиш кетма-кетликни ташкил этувчиларнинг текис тақсимланган генерацияси муаммосини ечиш билан боғлиқ. Бирор кетма-кетликни ташкил этувчи элементлар, шу кетма-кетликда деярли тенг миқдорда қатнашган бўлса, бу кетма-кетлик текис тақсимотга эга дейилади. Агар A – кетма-кетликни ташкил этувчи $x_t \in A$ – элементлари сони N та бўлса, у ҳолда ихтиёрий $t \in N$ учун, A – кетма-кетликни ташкил этувчи $x_t \in A$ – элементнинг, шу кетма-кетликдаги частотаси, бошқа элементларнинг частотаси билан деярли бир хил бўлади, яъни ҳар бир $x_t \in A$ – элемент шу кетма-кетликда деярли бир хил эҳтимоллик билан қатнашади.

Тасодифий кетма-кетликлар ҳақиқий тасодифий ва псевдотасодифий кетма-кетликларга бўлинади.

Тасодифий кетма-кетлик физик генераторлар ва дастурий генераторлардан фойдаланиб ҳосил қилиниши мумкин.

Физик ҳодисаларнинг ўзгариш мажмуига асосланган генераторлар орқали ишлаб чиқилган кетма-кетлик **ҳақиқий тасодифий** бўлиб, бу кетма-кетлик бир мартагина ишлаб чиқилиб, уни кейинчалик бирор бир усул ёки восита билан худди шундай тарзда такрорланишини бошқариш мураккаб ҳисобланади. Шу сабабли маълумотларни шифрлаш жараёнида бевосита физик генераторлар билан ишлаб чиқилган кетма-кетликни калитлар гаммаси сифатида қўллаш мақсадга мувофиқ

эмас. Чунки, дешифрлаш жараёнида қўлланиладиган физик генераторнинг айнан шифрлаш жараёнида қўлланилган кетма-кетликни ишлаб чиқиши кафолатланмайди.

Бирор номаълум параметрга (калитга) боғлиқ бўлган математик модель асосида псевдотасодифий кетма-кетлик ишлаб чиқувчи дастурий генераторлар ҳосил қилган **псевдотасодифий** кетма-кетликни, номаълум параметр қийматини билган ҳолда, худди шу математик модель ва унинг дастурий таъминоти асосида кетма-кетликнинг қайта такрорланишини бошқариш мумкин. Бундай ҳолат, маълумотларни шифрлаш жараёнида бевосита дастурий генераторлар билан ишлаб чиқилган псевдотасодифий кетма-кетликни калитлар гаммаси сифатида қўллаш мақсадга мувофиқлигини англатади ва дешифрлаш жараёнида қўлланиладиган дастурий генераторнинг айнан шифрлаш жараёнида қўлланилган псевдотасодифий кетма-кетликни ишлаб чиқиши кафолатланади.

Юқорида келтирилган амалий масалаларни ечишда хақиқий тасодифий кетма-кетликлар ишлаб чиқувчи тасодифий физик ҳодисаларга асосланган генераторлар олдиндан калитлар блоклари мажмуини яратишда, генераторларнинг бошланғич параметрлари қийматларини ўрнатишда ва бошқа шу каби масалаларни ечишда самарали натижалар беради.

Етарли катта давр узунлигига эга ва тасодифийлик даражаси юқори бўлган кетма-кетликлар ҳосил қилувчи дастурий ПТКК генераторини амалда қўланилиши самарали ва қулай бўлиб, криптографик воситаларда кенг қўлланилади.

Узлуксиз шифрлаш тизимларида шифрлаш ва дешифрлаш жараёнларининг тез амалга оширилиши учун ташкил этувчилари текис тақсимланган, тасодифийлик даражаси юқори бўлган псевдотасодифий кетма-кетлик ишлаб чиқарувчи дастурий генераторлардан фойдаланилади.

Мавжуд дастурий генераторлар ва улар асосидаги узлуксиз шифрлаш тизимлари маълум бир ёндашувлар асосида яратилган.

Узлуксиз шифрлаш алгоритмларига қўйиладиган асосий талаблардан бири уларнинг криптографик бардошлилигини таъминловчи бирор ечилиши мураккаб бўлган математик муаммолар асосида яратилишидир. Алгоритмлар криптобардошлилигининг етарли даражада таъминланганлигини кафолатлаш ёки исботлаш асослари нуқтаи-назаридан мавжуд узлуксиз шифрлаш алгоритмларини асосан учта йўналишга ажратиш мумкин:

Тизимли-назарий ёндашув йўналишидаги ПТКК генераторлари асосида яратилган алгоритмлар;

Мураккабликка асосланган назарий ёндашув йўналишидаги ПТКК генераторлари асосида яратилган алгоритмлар;

Комбинациялаш йўналишидаги ПТКК генераторлари асосида яратилган алгоритмлар.

§ 5.1. Тизимли-назарий ёндашув асосида қурилган ПТКК генераторлари

Ушбу асосда узлуксиз шифрлаш алгоритмларини яратиш кўп жиҳатдан блокли шифрлаш алгоритмларини яратишга ўхшаш бўлиб, узлуксиз шифрлаш алгоритмининг криптобардошлилиги фундаментал математик критерийлар ва қонуниятлар асосида шу пайтгача мураккаб ва самарали ечиш усули мавжуд эмас деб ҳисобланган муаммонинг қийинчилигига тенглаштирилади. Бундай ҳолатларда кўпроқ назарий ва амалий жиҳатдан криптографик самара берувчи математик акслантиришлар қўлланилган ҳолда криптографик тузилма (схема) таклиф қилинади ва бу тузилмани (схемани) криптографик бардошлилиги тадқиқ қилинади. Математиканинг назарий ютуқларига асосланган ҳолда етарли катта давр узунлигига, битлар ва байт блокларининг текис тақсимотига, акслантиришларининг аналитик ва мантиқий (чинлик жадвали асосидаги Буль функцияси) математик моделларини ифодаловчи функциялар чизиқсизлик даражаси юқори бўлиши каби бошқа хусусиятларга эга бўлган кетма-кетлик ишлаб чиқувчи алгоритмлар яратилади. Яратилган алгоритмлар акслантиришларининг турли хил криптотаҳлил усулларига бардошлилиги асосланади. Агар яратилган алгоритмлар шу пайтгача мавжуд бўлган криптотаҳлил усулларига бардошли бўлса, ҳамда ҳосил қилинган кетма-кетлик тасодифийлик тестлари талабларига жавоб берса, бу алгоритмни амалиётда қўллаш мумкинлиги тўғрисида хулоса қилинади.

Мавжуд узлуксиз шифрлаш алгоритмлари асосан тизимли-назарий ёндашув натижасида яратилган алгоритмлар синфига (туркимига) киради.

Қуйида тизимли-назарий ёндашув асосидаги узлуксиз шифрлаш алгоритмларига қўйиладиган асосий талаблар келтириб ўтилади [14, 15]:

– алгоритм асосидаги ПТКК генератори етарли узун даврга эга бўлган кетма-кетлик ишлаб чиқишни таъминлаши керак;

– генератор акслантиришларининг аналитик ва мантиқий (чинлик жадвали асосидаги Буль функцияси) математик моделларини ифодаловчи функциялар чизиқсизлик даражаси юқори бўлиши керак;

– ишлаб чиқилган ПТКК блоклари текис статистик тақсимот кўрсаткичига эга бўлиши керак;

– псевдотасодифий кетма-кетликнинг гамма элементлари (бит, байт, қисм блоклари) барча бошқа элементларининг ҳиссаси орқали ҳосил қилиниши — аралашиш самарали бўлиши керак;

– ПТКК гамма элементларининг кескин ўзгариши — тарқалиши самарали бўлиши керак;

– алгоритм акслантиришлари Буль функцияларининг чизиқсизлик шарти бажарилиши ҳамда жадал самара («лавинный эффект») бериши таъминланиши керак.

Тизимли-назарий ёндашув асосида яратилган узлуксиз шифрлаш алгоритмларининг бардошлилиги, бу алгоритмларда қўлланилган акслантиришларнинг назарий ва амалий бир томонламалик хусусиятларининг қай даражада ишончлилигини баҳолаш билан исботланади.

Тизимли-назарий ёндашув асосида яратилган узлуксиз шифрлаш алгоритмлари таркибидаги генераторларни яратилиш асосларига кўра *элементар реккурент ҳисоблашларга, силжитиш регистрларига, бир томонлама функцияларга, байтлар ва битлар блокларининг ўрнини боғлиқсиз алмаштиришга асосланган генераторларга* ажратиш мумкин.

Элементар реккурент ҳисоблашларга асосланган псевдотасодифий кетма-кетлик генераторлари уларда қўлланилган акслантиришларга кўра *чизиқли, мультипликатив, чизиқсиз* туркумларга бўлинади [14, 20].

Чизиқли ва мультипликатив конгруэнт генераторлар

Чизиқли конгруэнт генераторлар умумий ҳолатда $x_{i+1}=(ax_i+c) \bmod N$ формула билан аниқланувчи реккурент ҳисоблашга асосланган. Дастлабки берилган кириш параметрлари асосида кетма-кетликлар ҳосил қилинади.

Кириш параметрлари:

N – чекли майдон характеристикасини ифодаловчи сон, a ва c – ўзгармас мусбат бутун сонлар, x_0 – бошланғич бутун қийматли сон;

Кетма-кетликни ташкил этувчи чиқиш қийматлари:

$$x_{i+1}=(ax_i+c) \bmod N,\ i=0,\ 1,\ 2,\ 3,\ \ldots;$$

Чизиқли конгруэнт генераторнинг кириш параметри $c=0$ бўлса, яъни

$$x_{i+1}=(ax_i) \bmod N,\ i=0,\ 1,\ 2,\ 3,\ \ldots;$$

бўлса, бу генератор чизиқли мультипликатив генератор дейилади.

5.1-тасдиқ. Ушбу $x_{i+1}=(ax_i+c)\bmod N$, $i=0, 1, 2, 3, \ldots$; рекурент формула билан аниқланган псевдотасодифий кетма-кетлик максимал N даврга эга бўлиши учун қуйидаги:

1) c ва N – ўзаро туб сонлар, яъни ЭКУБ $(c, N)=1$;

2) p сони N сонининг бўлувчиси ва $a-1$ сони p сонига каррали;

3) N сони 4 каррали бўлса, $a-1$ сони ҳам 4 га каррали шартларнинг бажарилиши зарур ва етарли.

Қуйида эса чизиқли ва мультипликатив генераторларнинг максимал даврга эга бўлиши билан боғлиқ айрим хоссалар келтириб ўтилади [1, 12, 14].

Бевосита ҳисоблаш билан ишонч ҳосил қилиш мумкинки, ушбу $x_{i+1}=(ax_i+c)\bmod N$, $i=0,1,2,3, \ldots$; тенглик билан аниқланувчи кетма-кетлик умумий ҳади учун

$$x_t=\left(a^t x_0+\frac{at-1}{a-1}c\right)\bmod N,\ t\geq 1$$

формула ўринли.

Чизиқли ва мультипликатив конгруэнт генераторларнинг камчилиги шундаки, ПТКК бирор биграммасини $(z_1; z_2): z_1=xt$, $z_2=x_{t+1}$, билган ҳолда, унинг бошқа ташкил этувчиларини топиш имконияти мавжуд [14, 20, 21]. Ҳақиқатан ҳам, кетма-кетликни барча ташкил этувчи қийматлари $z_2=az_1+c-kN$, $k=0, 1, \ldots$ чизиқлар оиласида ётади.

Чизиқсиз конгруэнт генераторлар

Кириш параметрлари:

N – чекли майдон характеристикасини ифодаловчи сон;

d, a ва c – ўзгармас мусбат бутун сонлар, x_0 – бошланғич қиймат;

Кетма-кетликни ташкил этувчи чиқиш қийматлари:

$$x_{i+1}=(dx_i^2+ax_i+c)\bmod N,$$ бу ерда: $i=0, 1, 2, \ldots$.

Бу генератор квадратик генератор деб ҳам аталади.

5.2-тасдиқ. Квадратик генераторлар ҳосил қилган ПТКК ўзининг $T_{\max}=N$ максимал даврига эга бўлиши учун қуйидаги шартларнинг:

1) c ва N – ўзаро туб сонлар;

2) d, $a-1$ – сонлари бирор p – туб сонга каррали бўлиб, бу p – сони N нинг бўлувчиси;

3) d – жуфт сон бўлиб,

$$d=\begin{cases}(a-1)\bmod 4, & \textit{агар } N \textit{ сони 4 га каррали бўлса;}\\ (a-1)\bmod 2, & \textit{агар } N \textit{ сони 2 га каррали бўлса;}\end{cases}$$

4) агарда N сони 9 га каррали бўлса, у ҳолда $d \bmod 9=0$ ёки $d \bmod 9=1$ ва $cd \bmod 9=6$ бажарилиши зарур ва етарли.

Шунингдек $N=2^q$ бўлса, максимал даврни таъминлаш учун d – тоқ бўлиши ва $a=(d+1)\bmod 4$ бўлиши етарлидир.

Чизиқли ва мультипликатив конгруэнт генераторлар каби чизиқсиз генераторлар ҳам киптотаҳлил усулига бардошсиз [2, 14, 15, 20, 21, 28].

Силжитиш регистрларига асосланган генераторлар

Хозирги пайтгача таклиф этилган ва муваффақиятли равишда ишлатилиб келинаётган узлуксиз шифрлаш алгоритмларининг асосини силжитиш регистрлари ёки аниқ қилиб айтганда чизиқли тескари боғланишли силжитиш регистрлари ташкил қилади. Бундай регистрлар Фиббоначи регистрлари ёки Галуа регистрлари ҳам деб аталади. Бу хилдаги узлуксиз шифрлаш алгоритмларининг оммавий қўлланилишига икки ҳил сабабни кўрсатиш мумкин [14, 15, 20, 28]:

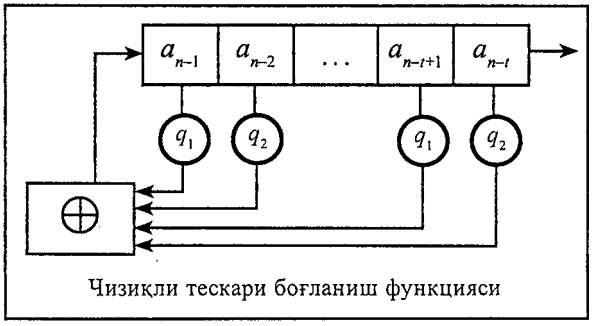
1. Тескари боғланишли силжитиш регистрларига асосланган генераторлар ҳосил қилган кетма-кетликлар яхши тасодифийлик статистик характеристикаларини беради;

2. Силжитиш регистрларига асосланган генераторларнинг хусусиятларини таҳлил қилиш осон.

Радиоэлектрон элементлар кенг қўлланила бошлагандан кейинги давр учун силжитиш регистрларига асосланган узлуксиз шифрлаш алгоритмлари ҳарбий соҳадаги криптографик воситалар тизимларининг асоси бўлиб хизмат қилиб келмоқда.

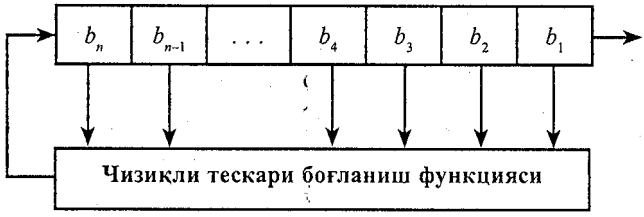
Тескари боғланишли силжитиш регистрлари, ўз навбатида, чизиқли тескари боғланишли ва чизиқсиз тескари боғланишли силжитиш регистрларига бўлинади.

Қуйида силжитиш регистрлари турларининг функционал схемалари келтирилган.



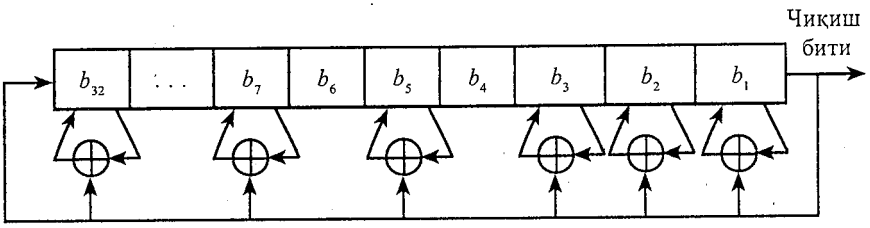
5.1-расм. **Тескари боғланишли силжитиш регистрининг умумий кўриниши.**

Силжитиш регистрларига асосланган генераторлар икки қисмдан ташкил топган: биринчи қисм бу силжитиш регистри бўлса, иккинчи қисми тескари боғланиш функциясидир. Силжитиш регистрларига асосланган алгоритмларнинг дастурий ёки аппарат-дастурий жиҳатдан қўлланилиши қулай ва самаралидир. Амалда аппарат-техник қурилмаларни яратишда қулай ва тез ишлашни таъминлаш учун микропроцессорнинг регистрлари (ячейкалари) сони билан силтиш регистрлари (ячейкалари) сони тенг қилиб олинади. IBM компанияси томонидан ишлаб чиқилаётган Intel процессорлари 64 разрядли регистрларда ишлаганлиги сабабли дастурий таъминотда силжитиш регистрларига асосланган алгоритмда регистр узунлигини 64 га тенг ёки унга каррали қилиб олиш мақсадга мувофиқдир. Силжитиш регистрларининг дастлабки ҳолати ва регистрлардан тескари боғланиш функциясига чиқишлар тўғри танланганда ҳосил қилинган кетма-кетлик даври максимал бўлади. Силжитиш регистрларининг иккинчи қисми бу тескари боғланиш функциясидир. Тескари боғланиш функцияси ҳар бир тактда регистрнинг кўпҳад билан ифодаланувчи ўринларидаги битлар қийматини XOR амали билан қўшиб, ҳосил бўлган қийматни регистрнинг энг катта разряди ўрнига силжитиш орқали киритади. Энг кичик разряд қиймати эса гамма кетма-кетлик элементи сифатида узатилади.



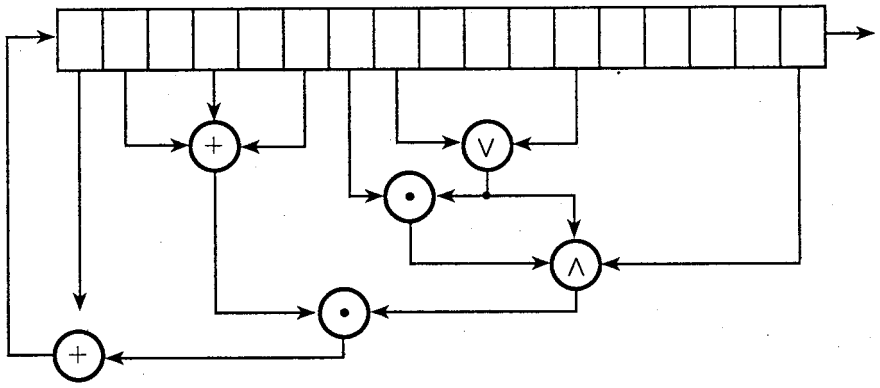
5.2-расм. *Чизиқли тескари боғланишли силжитиш регистри.*

Чизиқли тескари боғланишли силжитиш регистрларидан бири бу Галуа конфигурациясидир. Галуа конфигурациясида гамма кетма-кетлик элементлари сифатида узатиладиган бит қиймати тескари боғланиш функциясида иштирок этади. Чиқиш бити регистрнинг ҳамма битига XOR амали орқали қўшилади ва регистрнинг катта бити ўрнига силжитиш орқали қўйилади. Энг кичик бит қиймати эса гамма кетма-кетлик элементи сифатида чиқади ва тескари боғланиш функциясида иштирок этади. Ушбу регистрдан чиқувчи кетма-кетликнинг даври максимал бўлиши учун тескари боғланиш функцияси аргументлари регистрнинг келтирилмайдиган кўпҳад ҳосил қилувчи ҳадларидан олиниши лозим.



5.3-расм. *Галуа конфигурациясига асосланган силжитиш регистри.*

Чизиқсиз тескари боғланишли силжитиш регистрларида тескари боғланиш функцияси бир неча хил чизиқсиз акслантиришларни қўллаш орқали амалга оширилади.



5.4-расм. *Чизиқсиз тескари боғланишли силжитиш регистри.*

Юқорида келтирилган схемада тескари боғланиш функцияси XOR, AND, OR амаллари орқали амалга оширилган. Ҳозиргача чизиқсиз силжитиш регистрларига асосланган генераторлар ҳосил қилган кетма-кетликларни етарлича таҳлил қилувчи математик усуллар ишлаб чиқилмаган. Шу сабабли, чизиқсиз тескари боғланишли регистрлар орқали амалга оширилган генераторлар билан боғлиқ қуйидаги муаммоларни келтириш мумкин:

– ҳосил қилинган псевдотасодифий кетма-кетликда текис тақсимотдан четланиш бўлиши мумкин, яъни «0» ва «1» белгиларнинг ишлаб чиқилган гамма кетма-кетлик блокларидаги миқдори деярли тенг бўлмаслиги мумкин;

– кетма-кетликнинг даври кутилганидан қисқа бўлиши мумкин;

– кетма-кетлик даври ҳар-хил бошланғич қийматлар учун ҳар-хил бўлиши мумкин, яъни маълум бир талабга жавоб берувчи пара-

метрлар танланганда ҳар қандай ихтиёрий бошланғич қиймат учун генератор ҳосил қилган кетма-кетлик даври максимал бўлади деб бўлмайди;

– ҳосил қилинган гамма кетма-кетлик текшириш ҳисоб-китобларисиз тасодифийга ўхшаб кўриниши мумкин, лекин регистрнинг маълум бир ҳолатидан сўнг, чизиқсизлик амалининг маҳсули сифатида, кейинги ҳосил бўлган гамма кетма-кетлик элементлари фақат «0» ёки «1» лардан иборат бўлиб қолиши мумкин.

Чизиқсиз силжитиш регистрларининг криптографик самарали тарафи бундай регистрларга асосланган узлуксиз шифрларнинг криптографик таҳлили усуллари камлигидир.

Бир томонлама функцияларга асосланган алгоритмлар

ПТКК генераторлари яратишда бир томонлама функциялар кенг қўлланилади. Бир томонлама функцияларнинг ўзига хос хусусиятларидан бири шундан иборатки, бу функциянинг қийматини аргументнинг берилган қиймати бўйича ҳисоблаш полиниомиал-мураккабликка эга бўлиб, функциянинг берилган қиймати бўйича бу қийматга мос бўлган аргумент қийматини ҳисоблаш экспоненциал мураккабликка эга ёки ҳисоблашнинг рационал алгоритми мавжуд эмас (ёки номаълум).

FIPS-186 генератори

Бу алгоритм АҚШ миллий стандарти сифатида қабул қилинган бўлиб, DSA электрон рақамли имзо алгоритми учун махфий параметр ва калит ишлаб чиқишга мўлжалланган.

Бу алгоритмда бир томонлама функциялар сифатида DES ва SHA-I алгоритмларидан фойдаланилган.

Кириш параметрлари: m – бутун сон, q – 160 битли туб сон, $b=160$,

$t=67452301$ EFCDAB89 98BADCFE 10325476 C3D2E1F0$_{16}$ – 160 битли сон;

$y_i=0$;

s – бошланғич 160 битли тасодифий ва махфий сон.

Чиқиш параметрлари:

$x_1, x_2, x_3, \ldots\ldots, x_m$

Алгоритм қадамлари:

1. $z_i=(s+y_i)\mathrm{mod}2^b$;

2. $x_i = G(t, z_i)$;

3. $s = (1 + s + x_i) \bmod 2^b$.

G: – бир томонлама функция сифатида ишлатилган DES-шифрлаш алгоритми

кириш: $t = t_0 || t_1 || t_2 || t_3 || t_4$; $z_i = z_0 || z_1 || z_2 || z_3 || z_4$;

чиқиш: $w = w_0 || w_1 || w_2 || w_3 || w_4$;

Функция:

1. $u_i = t_i \oplus z_i$, бу ерда: $i = 0$ дан 4 гача ўзгаради;

2. $b_1 = z_{(i+4) \bmod 5}$, $b_2 = z_{(i+3) \bmod 5}$, бу ерда: $i = 0$ дан 4 гача ўзгаради;
   $a_1 = u_2$, $a_2 = u_{(i+1) \bmod 5} \oplus u_{(i+4) \bmod 5}$;
   $A = a_1 || a_2$, $B = b_1 || b_2$;
   $y_i = E_B(A)$;
   $y_i = L_i || R_i$;

3. $w_i = w_i + L_i \oplus R_{(i+2) \bmod 5} \oplus L_{(i+3) \bmod 5}$, бу ерда: $i = 0$ дан 4 гача ўзгаради.

4. Натижа: $G(t, z_i) = w_0 || w_1 || w_2 || w_3 || w_4$.

§5.2. Мураккабликка асосланган назарий ёндашув асосида қурилган ПТКК генераторлари

Мураккабликка асосланган назарий ёндашув негизида қурилган узлуксиз шифрлаш алгоритмлари ПТКК ишлаб чиқарувчи генераторларининг криптобардошлилиги: етарли даражада катта сонни туб кўпайтувчиларга ажратиш, характеристикаси етарли катта бўлган чекли майдонларда дискрет логарифмлаш, чекли майдонларда етарли даражада юқори тартибли чизиқли тенгламалар тизимларини ечиш, эллиптик эгри чизиқ нуқталари устида амаллар бажариш билан боғлиқ бўлган масалаларни ечиш мураккабликлари билан аниқланади.

Катта сонларни туб кўпайтувчиларга ажратиш мураккаблигига асосланган RSA генератори

Кириш параметрлари: p ва q – етарли даражада катта бўлган махфий туб сонлар;

$N = pq$ – очиқ , e – сони очиқ ва ушбу ЭКУБ $(e, (p-1)(q-1)) = 1$ шартни қаноатлантиради, x_0 – тасодифий бошланғич сон;

Чиқиш параметрлари:

$x_{i+1} = x_i^e \bmod N$, $i = 0, 1, 2, \ldots$; бўлиб, гамма кетма-кетликнинг элементи сифатида, ҳосил бўлган x_{i+1} натижадан иккилик саноқ тизимидаги ифодасининг охирги бити олинади.

Бу алгоритм катта сонлар устида амаллар бажариш билан боғлиқ бўлиб, нисбатан секин ишлайди. Шу сабабли алоқа тармоғида овозли

ва тасвирий маълумотлар алмашинувининг махфийлигини шу жараён реал кечаётганда тўғридан-тўғри таъминлашда қўлланилувчи узлуксиз шифрлаш алгоритмлари асосидаги ПТКК ишлаб чиқарувчи генераторлар сифатида ишлатиш мақсадга мувофиқ эмас. Бундай генераторларнинг калит блокларини ишлаб чиқувчи генератор сифатида ишлатилиши мақсадга мувофиқ бўлади.

Квадратик чегирма усулига асосланган BBS генератори

Блюм-Блюм-Шуб (Blum-Blum-Shub) оддий, лекин эффектив генератор бўлиб квадратик чегирма усулига асосланган.

Кириш параметрлари:

– p ва q – етарли даражада катта бўлган махфий туб сонлар;

– $N = pq$ – очиқ ;

– тасодифий x сони N сони билан ўзаро туб, яъни ЭКУБ(x, N)=1;

– $x_0 = x^2 \bmod N$ – бошланғич қиймат;

– e–сони очиқ ва ушбу ЭКУБ $(e,(p-1)(q-1))=1$ шартни қаноатлантиради, x_0 – тасодифий бошланғич сон;

Чиқиш параметлари:

– $x_{i+1} = x_i^2 \bmod N$, $i=0, 1, 2, ...$; бўлиб, гамма кетма-кетликнинг элементи сифатида, ҳосил бўлган x_{i+1} натижадан иккили саноқ тизимидаги ифодасининг охирги бити олинади.

Дискрет логарифмлаш масаласининг мураккаблигига асосланган Блюм – Микали генератори

Кириш параметрлари:

g – туб сон, p – туб сон, x_0 – махфий калит.

Чиқиш параметрлари:

$x_{i-1} = g^{x_i} \bmod p$, $i=0, 1, 2, ...$; бўлиб, гамма кетма-кетликнинг элементи сифатида, ҳосил бўлган $x_{i+1} < (p-1)/2$ натижа шартни қаноатлантирса генераторнинг чиқиш киймати «1» бўлади, акс ҳолда «0» бўлади. p – сонининг етарли даражада катта қийматларида дискрет логарифмни ҳисоблаш мураккаблашиб, генераторнинг етарли даражада криптобардошли бўлишини таъминлайди.

§ 5.3. Комбинациялаш асосида қурилган псевдотасодифий кетма-кетлик генераторлари

Юқорида ПТКК ишлаб чиқарувчи генераторлар тизимли-назарий ёндашув асосида ва мураккабликка асосланган назарий ёндашув йўналишлари таҳлил қилинди. Шу йўналишларда яратилган базавий

(таянч)генераторлардан фойдаланиб уларни комбинациялаш асосида янги генераторлар яратиш усуллари, комбинациялаш асосидаги псевдотасодифий кетма-кетлик генераторларини яратиш йўналиши деб аталади.

Бу ёндашувда мавжуд ПТКК ишлаб чиқувчи генераторлар асосидаги акслантиришларнинг (алгоритмларнинг) бирлаштирилиши (комбинацияси) асосида янги генератор яратилади. Бу генераторнинг криптобардошлилиги унинг таркибидаги ҳар бир акслантиришнинг ва алгоритмларнинг мураккаблиги билан экспоненциал боғлиқдир.

Комбинациялаш асосида қурилган ПТКК генераторларини яратиш: полиномиал мураккабликка эга акслантиришларни, тасодифий параметрли алгоритмларни комбинациялаш, Макларен-Марсальи ва бошқа шу каби усуллар орқали амалга оширилади.

Хозирги пайтгача қўлланилиб келинаётган силжитиш регистрларига асосланган ПТКК ишлаб чиқувчи генераторларнинг криптобардошлилиги полиномиал мураккабликка асосланган силжитиш регистрларини комбинациялаш орқали ишлаб чиқилган.

Макларен-Марсальи усули

Бу усул ёрдамида комбинациялаш учун иккита ПТКК ишлаб чиқувчи генераторлар (G_1, G_2) олинади. Бу генераторлар мос равишда $\{x_i\}$ ва $\{y_j\}$ кетма-кетликни ҳосил қилсин. Бирор k (яъни $1\times k$) етарли катта ўлчамга эга, T жадвал мос равишда G_1 генератор натижаси бўлган $x_i (i=1, 2, ..., k)$ қийматлар билан тўлдириб чиқилади. Шундан сўнг, тўлдириб чиқилган жадвалдаги қийматлар бўйича G_2 генератор натижаси бўлган $\{y_j\}$ индексда турувчи $x_{y_j} (1 \le y_j \le k)$ элементни ПТКК гаммаси элементи сифатида узатиш мумкин. Демак бу усулда «тасодифий» $\{x_i\}$ сонлар билан тўлдирилган T жадвалдан «тасодифий» $\{y_j\}$ индекс бўйича қийматлар танланади. Бу усулда таклиф қилинган комбинациялашган генераторлардан бири Chameleon генераторидир. Бу генератор асосида яратилган узлуксиз шифрлаш алгоритмининг хусусиятли томони шундан иборатки, бардошли криптотизимни ишлатган ҳолда калитнинг кам қисми ўзгариши очиқ маълумотнинг ҳам оз ўзгаришига олиб келишидадир. Криптотизимга бундай талаб қўйилиши хозирги пайтда тижорат телевидениеси ва тижорат радиотармоқларининг кўпайиб бориши билан келиб чиқди. Тижорат телевидениеси сотиб олинган ва интеллектуал мулк ҳисобланган қўшиқ ва кинофильмларни шу канал орқали шифрлаб узатади. Тижорат канали абонентлари қабул қилиб

олинган қўшиқ ва кинофильмларни эшитишлари ва кўришлари мумкин, дискларга ёзиб олиб кўпайтириш ва сотиш имкониятларини чегаралашга имкон беради. Баъзи абонентлар қонунга хилоф равишда дешифрланган қўшиқ ва кинофильмларни дискларга ёзиб олиб сотувга чиқаришлари мумкин. Тижорат телевидениеси сотувга чиқарилган диск ўзлари берган калитдаги битлар ўзгаришига қараб дарҳол қайси абонент томонидан дешифрланганини аниқлаб олишга муваффақ бўлади.

Криптотизим қуйидагича ишлайди.

Ихтиерий узлуксиз шифрлаш алгоритми ҳамда 2^{16} та 64 битли элементи бўлган умумий ўлчами 512 килобайтли B массив олинади. Танлаб олинган узлуксиз шифрлаш алгоритмидан чиққан 64 битли кетма-кетлик 4 қисмга бўлиниб, ҳар бир 16 битли қисмга мос келувчи индексларда турувчи бошланғич B массив элементлари ўзаро XOR амали билан қўшилиб, 16 битли гамма ҳосил қилинади. Бу гамма очиқ маълумотнинг 16 битли блоки билан XOR амали билан қўшилиб, натижада шифрмаълумот ҳосил қилинади ва эфирга узатилади. Бу ерда B массив калит сифатида ҳар бир абонентга алоҳида-алоҳида берилади. Лекин ҳар бир берилган 512 килобайтли B массивнинг фақат ихтиёрий битта бити ўзгартирилиб берилади. Натижада ҳар бир абонентга бир хил шифрматн келгани билан 512 килобайтли B массивдаги бир бит ўзгариши ҳисобига дешифрланган ҳар бир 512 килобайтли очиқ маълумотларда 4 та бит ўзгарган ҳолатда дешифрланади. Мана шу ўзгаришлар орқали қайси абонент дешифрлагани аниқланади.

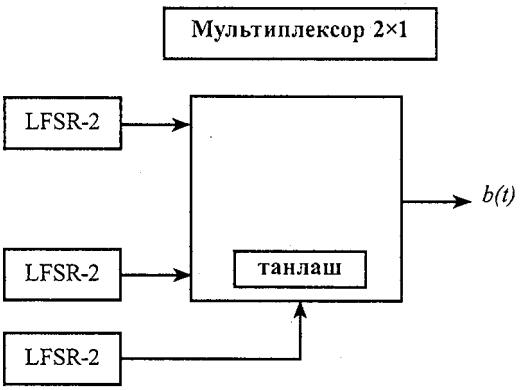
Тасодифий параметрли комбинациялашган генераторлар

Бу усулда генераторларни комбинациялаш учун иккита G_1, G_2 генератор олинади.

G_1 генераторни гамма ҳосил қилувчи ва G_2 ни параметр ўзгартирувчи деб олинади. Агар бу G_1 генераторнинг бошлангич параметрлари a_1, b_1, c_1 деб олинадиган бўлса, бошланғич қиймат x_0 орқали $G_1^1(x_0)=x_1$ ҳисобланади, x_2 ни ҳисоблаш учун эса G_1 генераторнинг бошланғич параметрларини иккинчи G_2 генераторнинг генерация қилган қийматларига қараб $a1=e_i$, $b1=f_i$, $c1=g_i$ ўзгартирилади, бу ерда $\{e_i, f_i, g_i\}=G_2(x)$. Натижада, параметрлари тасодифий ўзгарувчи комбинациялашган генератор G_1 га эга бўлинади.

Полиномиал комбинациялаш

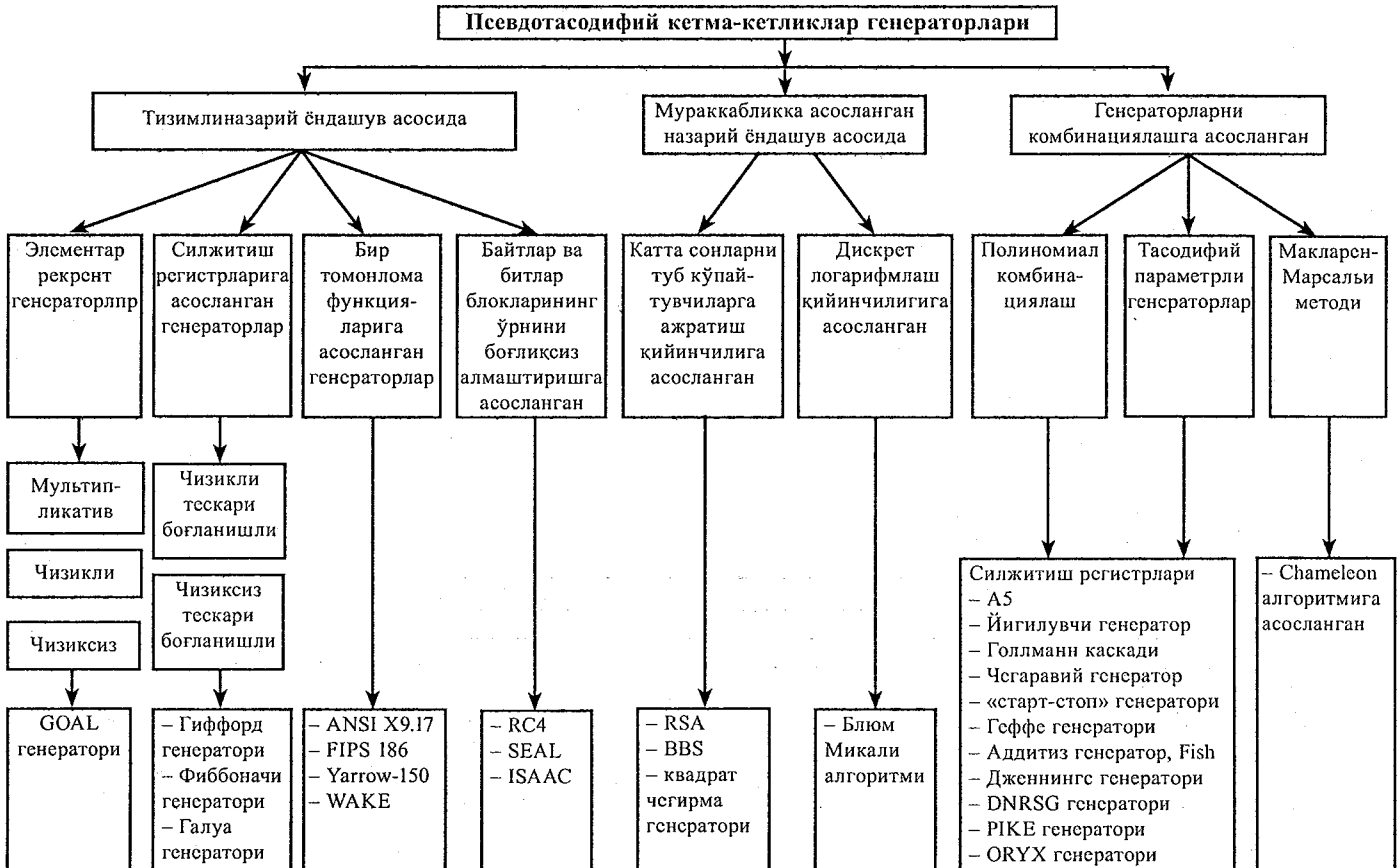
Бу турда кўпроқ силжитиш регистрларини комбинациялаш қўлланилган. Бир нечта силжитиш регистрлари олинади ва шу регистрлардан бири бошқарувчи регистр сифатида чиқиш бити қайси регистрдан олинишини аниқлаб беради. Геффе генераторида учта регистр ишлатилган бўлиб биринчи регистр қолган регистрларни бошқаради.



5.5-расм. *Мультиплексорли комбинацияланган регистр.*

Геффе генератори назарий жиҳатдан бардошли бўлганлиги билан корреляцион ҳужум турига бардошли эмас. Амалий кузатувлар шуни кўрсатадики, 75% гамма битлар битта силжитиш регистрининг чиқишига тенгдир. Агар тескари боғланиш примитив кўпҳади маълум бўлса бу генераторнинг бошланғич қийматини аниқлаш мумкин бўлади. Агар тескари боғланиш примитив кўпҳади учҳад бўлиб, силжитиш регистларининг узунлиги n бўлса, $37n$ битли чиқиш гаммасига эга бўлган ҳолда, учта силжитиш регистрларининг ҳолатини аниқлаш мумкин бўлади. Полиномиал комбинациялаш усулида корреляцион ҳужум турига бардошлилигини ошириш учун, бошқарувчи мультиплексор ёки генератор ҳосил қилган ПТКК тасодифийлик даражаси юқори ва статистик кўрсаткичлари текис тақсимланган бўлишини таъминлаш зарур.

Юқорида баъзи ПТКК генераторлари кўриб ўтилди. Амалда қўлланилиб келинаётган мавжуд генераторларни қуйидагича туркумлаш мумкин [20]:


Псевдотасодифий кетма-кетликлар генераторлари
Тизимлиназарий ёндашув асосида
Мураккабликка асосланган назарий ёндашув асосида
Генераторларни комбинациялашга асосланган
Элементар рекрент генераторлпр
Силжитиш регистрларига асосланган генераторлар
Бир томонлома функция-ларига асосланган генераторлар
Байтлар ва битлар блокларининг ўрнини боғлиқсиз алмаштиришга асосланган
Катта сонларни туб кўпай-тувчиларга ажратиш қийинчилига асосланган
Дискрет логарифмлаш қийинчилигига асосланган
Полиномиал комбина-циялаш
Тасодифий параметрли генераторлар
Макларен-Марсальи методи
Мультип-ликатив
Чизикли
Чизиксиз
Чизикли тескари боғланишли
Чизиксиз тескари боғланишли
GOAL генератори
– Гиффорд генератори
– Фиббоначи генератори
– Галуа генератори
– ANSI X9.17
– FIPS 186
– Yarrow-150
– WAKE
– RC4
– SEAL
– ISAAC
– RSA
– BBS
– квадрат чегирма генератори
– Блюм Микали алгоритми
Силжитиш регистрлари
– A5
– Йигилувчи генератор
– Голлманн каскади
– Чегаравий генератор
– «старт-стоп» генератори
– Геффе генератори
– Аддитиз генератор, Fish
– Дженнингс генератори
– DNRSG генератори
– PIKE генератори
– ORYX генератори
– Chameleon алгоритмига асосланган

Юқорида узлуксиз шифрлаш алгоритмларини яратилишининг криптографик зарурияти асослари ёритилди.

Узлуксиз шифрлаш алгоритмлари асосини ташкил этувчи ПТКК ишлаб чиқарувчи генераторларнинг асосий криптобардошлилик характеристикалари келтирилди.

Тасодифийлик даражаси юқори бўлган псевдотасодифий кетмакетликларни криптографик масалаларни ечишда қўллаш соҳалари кўрсатилди.

Узлуксиз шифрлаш алгоритмларига қўйиладиган асосий талаблар келтирилиб, алгоритмларнинг криптобардошлилиги етарли даражада таъминланганлигини кафолатлаш ёки исботлаш асослари нуқтаи-назаридан мавжуд узлуксиз шифрлаш алгоритмлари асосан уч хил йўналишга ажратилди ва туркумланди (таснифланди).

Бундай туркумланиш (классификациялаш) мавжуд алгоритмлар акслантиришларининг қандай турдаги ечилиши мураккаб бўлган масалаларга асосланганлигини кўрсатиб, янги яратилиши мумкин бўлган узлуксиз шифрлаш алгоритмларини криптобардошлилигини баҳолаш усуллари йўналишларини аниқлайди.

Қуйида, муаллиф томонидан Тошкент ахборот технологиялри университети магистрларига илмий раҳбарлик вақтида олиб борилган тадқиқотлар натижасида аппарат-техник ва аппарат-дастурий криптографик воситаларда кенг қўллаш имконини берувчи, асосий акслантириш функциялари очиқ эълон қилинган мавжуд криптографик акслантиришлардан фарқли бўлган ПТКК ишлаб чиқарувчи янги генераторлар негизида янги яратилган узлуксиз шифрлаш алгоритмлари келтирилади.

§5.4. Байтлар ва битлар ўрнини боғлиқсиз алмаштиришга асосланган узлуксиз шифрлаш алгоритми

Ушбу алгоритм тизимли-назарий ёндашув асосида яратилган бўлиб алгоритм асосини: 256 байтли S – блок ва ўлчови 16×16 бўлиб, элементлари ярим байтдан иборат бўлган (0 дан 15 гача сонларнинг текис тақсимотидан иборат) сиқиш жадвали (СЖ) ташкил этади.

Алгоритм асосида қуйидаги акслантиришлар ётади:

– $(j+S_i+K_j) \bmod 256$ – йиғинди;

– $a<<b$ – бир байтли «a» сонини « b» сонининг охирги 3 битига тенг бўлган қийматга циклик суриш;

– СЖ(S_i)–бир байтли гамма блокни сиқиш жадвалидан ўтказиш;

– $S_j<=>S_i$ – S – блокнинг S_j ва S_i элементлари ўрнини алмаштириш;

– $L||R$ – иккита 4-битли блоклар конкатенациясини ифодаловчи байт.

Босқичлари:

1. Бошланғич ҳолатда S – блок 0 дан 255 гача бўлган сонлар билан кетма-кет тўлдириб чиқилади;

2. Сўнг K (128–2048 битли) –калит ёрдамида аралаштирилади:

$$i=(i+1) \bmod 256,$$
$$j=(j+S_i+K_i)\bmod 256,$$
$$S_j<<(S_i+K_i),$$
S_i ва S_j ўрни алмаштирилади;

3. Аралаштирилгандан сўнг ҳосил бўлган S – блок ёрдамида 2 байтли оралиқ натижа блоки ҳосил қилинади:

$$i=(i+v+1) \bmod 256,$$
$$j=(j+S_i) \bmod 256,$$

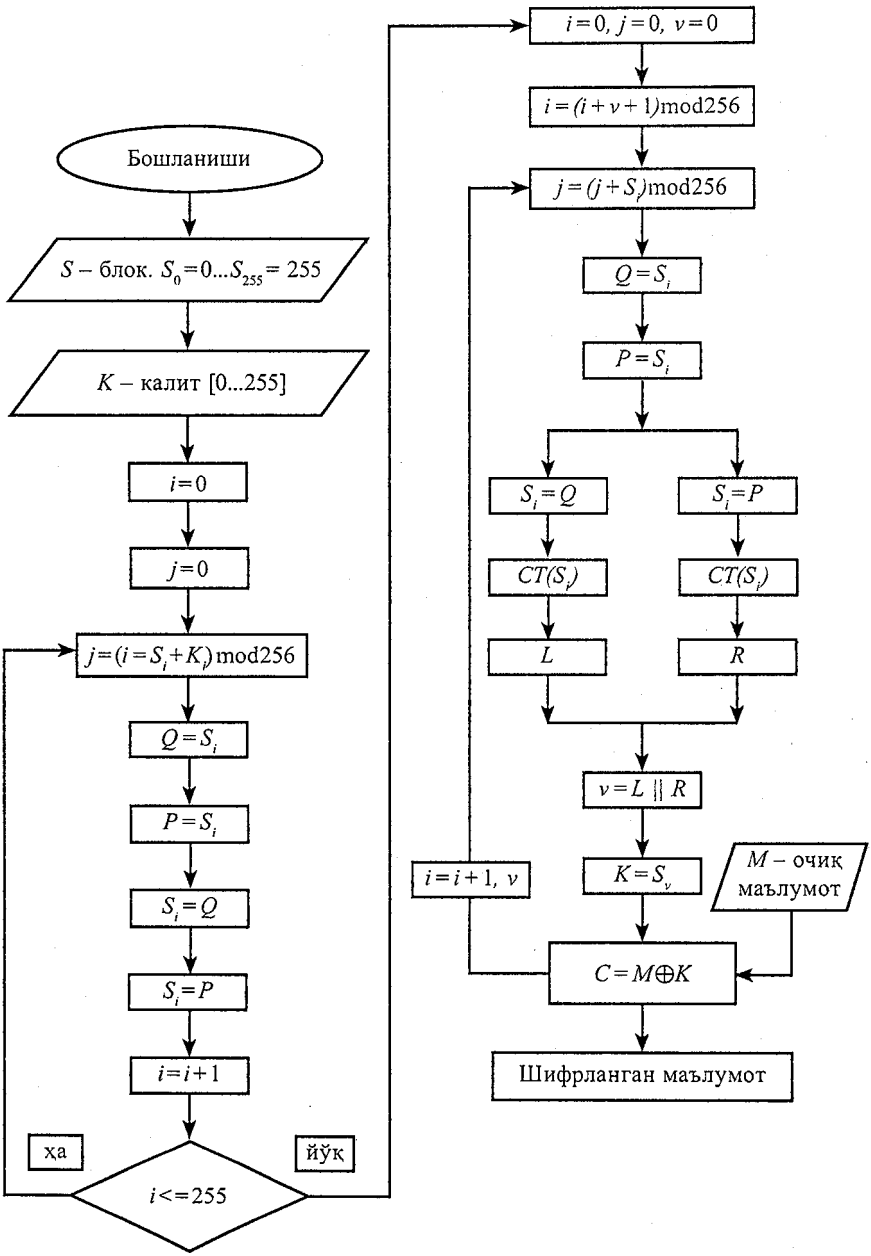
S_i ва S_j ўрни алмаштирилиб, яъни $Q=S_i$ ва $P=S_j$, сўнгра $S_i=P$ ва $S_j=Q$ амаллар бажарилиб, улар конкатенациясидан $S_i||S_j$ – оралиқ натижа олинади;

4. Ҳосил қилинган 2 байтли блок сиқиш жадвалидан (СЖ) ўтказилади ва олинган натижа M – очиқ маълумотнинг 1 байтига XOR амали билан қўшилади, яъни:

$$L=\text{СЖ}(S_i) \text{ ва } R=\text{СЖ}(S_j),$$
$$v=L||R \text{ ва } K=S_v,$$
$$C=M\oplus K.$$

5. S – блок алгоритмда келтирилган қоида орқали ўзгартирилиб яна, 3–4 босқичлар такрорланади.

Алгоритм блок схемаси



```
#include «stdafx.h»
#include «Potochnoe_01.h»
#include «Potochnoe_01Dlg.h»
#include «.\potochnoe_01dlg.h»
#include «TS_8ga 4.h»

#ifdef _DEBUG
#define new DEBUG_NEW
#endif
// CAboutDlg dialog used for App About
class CAboutDlg : public CDialog
{
public:
        CAboutDlg();

// Dialog Data
        enum {IDD = IDD_ABOUTBOX};
        protected:
        virtual void DoDataExchange (CDataExchange* pDX);
// DDX/DDV support
// Implementation protected:
        DECLARE_MESSAGE_MAP()
};

CAboutDlg: :CAboutDlg() : CDialog (CAboutDlg::IDD)
{
}
void CAboutDlg: :DoDataExchange (CDataExchange* pDX)
{
        CDialog: :DoDataExchange (pDX);
}

BEGIN_MESSAGE_MAP (CAboutDlg, CDialog)
END_MESSAGE_MAP()

// CPotochnoe_01Dlg dialog

CPotochnoe_01Dlg: :CPotochnoe_01Dlg(CWnd* pParent /*=NULL*/)
```

```
        : CDialog (CPotochnoe_01Dlg: :IDD, pParent)
        , ochiq1 (_T(«»))
        , shifr1 (_T(«»))
        , kalit1 (_T(«»))
        , len_k (0)
        , chek1 (FALSE)
        , chek2 (FALSE)
        , len_1 (0)
{
        m_hIcon = AfxGetApp()->LoadIcon (IDR_MAINFRAME);
}

void CPotochnoe_01Dlg: :DoDataExchange (CDataExchange* pDX)
{
        CDia log: :DoDataExchange (pDX);
        DDX_Text (pDX, IDC_EDIT1, ochiq1);
        DDX_Text (pDX, IDC_EDIT2, shifr1);
        DDX_Text (pDX, IDC_EDIT3, kalit1);
        DDX_Text (pDX, IDC_EDIT4, len_k);
        DDX_Check (pDX, IDC_CHECK1, chek1);
        DDX_Check (pDX, IDC_CHECK2, chek2);
        DDX_Text (pDX, IDC_EDIT5, len_1);
}

BEGIN_MESSAGE_MAP (CPotochnoe_01Dlg, CDialog)
        ON_WM_SYSCOMMAND()
        ON_WM_PAINT()
        ON_WM_QUERYDRAGICON()
        //}}AFX_MSG_MAP
        ON_BN_CLICKED (IDC_BUTTON1, OnBnClickedButton1)
        ON_BN_CLICKED (IDC_BUTTON2, OnBnClickedButton2)
        ON_BN_CLICKED (IDC_BUTTON3, OnBnClickedButton3)
        ON_BN_CLICKED (IDOK, OnBnClickedOk)
END_MESSAGE_MAP()

// CPotochnoe_01Dlg message handlers

BOOL CPotochnoe_01Dlg::OnInitDialog ()
{
        CDialog: :OnInitDialog ();
```

```
        // Add «About ...» menu item to system menu.

        // IDM_ABOUTBOX must be in the system command range.
        ASSERT ((IDM_ABOUTBOX & 0×FFF0)==IDM_ABOUTBOX);
        ASSERT (IDM_ABOUTBOX < 0×F000);

        CMenu* pSysMenu=GetSystemMenu (FALSE);
        if (pSysMenu !=NULL)
                {
                CString strAboutMenu;
                strAboutMenu. LoadString (IDS_ABOUTBOX);
                if (!strAboutMenu. IsEmpty())
                {
                        pSysMenu –> AppendMenu (MF_SEPARATOR);
                        pSysMenu –> AppendMenu (MF_STRING, IDM_
ABOUTBOX, strAboutMenu);
                }
                }

// Set the icon for this dialog. The framework does this automatically
// when the application's main window is not a dialog
SetIcon  (m_hIcon, TRUE);              // Set big icon
SetIcon (m_hIcon, FALSE);              // Set small icon

// TODO:Add extra initialization here

return TRUE; // return TRUE unless you set the focus to a control
}

void CPotochnoe_01Dlg: :OnSysCommand (UINT nID, LPARAM lParam)
{
        if ((nID & 0×FFF0)==IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout. DoModal();
        }
        else
        {
                CDialog::OnSysCommand nID, lParam);
        }
```

```
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CPotochnoe_01Dlg: :OnPaint()
{
        if (IsIconic())
        {
        CPaintDC dc(this); // device context for painting

        SendMessage (WM_ICONERASEBKGND,
reinterpret_cast<WPARAM> (dc.GetSafeHdc ()), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics (SM_CXICON);
                int cyIcon = GetSystemMetrics (SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width () – cxIcon + 1) / 2;
                int y = (rect.Height () – cyIcon + 1) / 2;

                // Draw the icon
                dc.DrawIcon (x, y, m_hIcon);
        }
        else
        {
                CDialog::OnPaint();
        }
}

// The system calls this function to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CPotochnoe_01Dlg::OnQueryDragIcon()
{
        return static_cast<HCURSOR> (m_hIcon);
```

```
}

void CPotochnoe_01Dlg::OnBnClickedButton1()
{

        CFileDialog dlg(1);
        if (dlg.DoModal()==IDOK)
        {
                ochiq1=dlg.GetPathName();
                UpdateData(0);
        }
        else
        {
                MessageBox («Ochiq text ochilmadi yoki fayl bilan ish-
lashda xato!!!»);
        }
        FILE *f1=fopen (ochiq1,»rb»);
        fseek (f1,0, SEEK_END); //Файл узунлигини аниклаш учун
        len_1=ftell (f1);                    // len га ёзиб қўямиз
                rewind (f1);
        UpdateData (0);

        // TODO: Add your control notification handler code here
}

void CPotochnoe_01Dlg::OnBnClickedButton2 ()
{
CFileDialog dlg (1);
        if(dlg.DoModal ()==IDOK)
        {
                shifr1=dlg. GetPathName ();
                UpdateData (0);
        }
        else
        {
        MessageBox («shifr txt ochilmadi yoki fayl bilan ishlashda xato!!!»);
        }}

void CPotochnoe_01Dlg::OnBnClickedButton3 ()
{
```

```
CFileDialog dlg (1);
        if(dlg.DoModal ()= =IDOK)
        {
                kalit1 =dlg.GetPathName ();
                UpdateData (0);
        }
        else
        {
        MessageBox («Kalit txt ochilmadi yoki fayl bilan ishlashda xato!!!»);
        }}

void CPotochnoe_01Dlg::OnBnClickedOk()
{UpdateData(1);// oynadan bor malumotlarni uqib olish

        FILE *f1 = fopen (ochiq1,»rb»);
                if(f1 = =NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }

        FILE *f2 = fopen (shifr1,»wb»);
                if (f2= =NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }

        FILE *f3=fopen (kalit1,»rb»);
                if (f2= =NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }
        //---------- kiritiladi----------
        byte buff_s[256];
        byte *buff_k;
        buff_k=new byte[len_k];
        byte buff_och[1];
        byte buff_sht[1];


        UINT s1, i, j=0, len_2, len_3=0;
```

```
byte c1, c2, v1;
//-----------------------------------------------------

 TS_8ga4 tabs;// klassdan tabs obekti hosil qilindi
 fread (buff_k, 1, len_k, f3); // kalit fayldan baytlar buferga uqib olindi
for (i=0; i<256; i++)
{
buff_s[i]=i;
}

for (i=0; i<256; i++)
{
j=(j+buff_s[i]+buff_k[i%len_k])%256;
s1=buff_s[i];
buff_s[i]=buff_s[j];
buff_s[j]=s1;

}
j=0;
i=0;
v1=0;
for (;;)
{
i=(i+v1+1)%256;

j=(j+buff_s[i])%256;
c1=buff_s[i];
c2=buff_s[j];
buff_s[i]=c2;
buff_s[j]=c1;
v1=tabs.func0 (c1, c2);

len_2=fread (buff_och, 1, 1,f1);
if (len_2==0)break;
buff_sht[0]=buff_och[0] ^ buff_s[v1];

fwrite (buff_sht, 1, 1, f2);
        i++;
        len_3++;
}
```

```
    fcloseall ();

    if (chek1)
    {
    MessageBox (« Shifrlash jarayoni tugadi «);
    }
    else
    {
    MessageBox («Deshifrlash jarayoni tugadi «);
    }

    // TODO: Add your control notification handler code here
    //OnOK ();
}
```

Алгоритм ҳосил қиладиган тасодифий кетма-кетлик даври узунлиги $(256! \cdot 2^{24}) = 2^{1711}$ байтни ташкил қилади.

Алгоритмга нисбатан қуйидаги криптоҳужум турларини кўриб ўтиш мумкин.

Чиқиш маълумоти блоки бўйича ҳужум.

Чиқиш маълумоти блокининг ярим байтига нисбатан унга мос келувчи СЖ га кириш 1-байтини аниқлаш СЖ очиқ бўлганда 2^4 та ҳолатдан бирини танлашни талаб этади. Аниқланган 2^4 та байтнинг ҳар бирига мос келувчи S – блокнинг байтини аниқлаш масаласи 2^8 та ҳолатдан бирини танлашни талаб этади. Демак, чиқиш бўйича ҳужум турида S – блокнинг 1 байтини аниқлаш учун 2^{12} ҳолатни кўриб чиқиш лозим. 256 байтли S – блокни аниқ тиклаш учун $(2^{12})^{256} = 2^{3072}$ та ҳолатни таҳлил қилиб чиқиш лозим бўлади.

Шифрмаълумот бўйича ҳужум.

Фақат шифрмаълумотга қараб S – блокни аниқлаш масаласини ҳам ҳисоблаб кўрсак $(2^8 \cdot 2^{12})^{256} = 2^{5120}$ та ҳолатдан бирини танлашга олиб келади. Шу сабабли, чиқиш блоки гамма кетма-кетликларининг бирор қисми маълум бўлганда ҳам кетма-кетликнинг кейинги байтини калитни билмаган ҳолда олдиндан аниқлаш эҳтимоллиги 1/256 га тенгдир.

Алгоритм асосини ташкил этувчи 256 байтли S – блок ва ўлчами 16x16 бўлиб, элементлари ярим байтдан иборат бўлган СЖ акслантиришлари уларга кирувчи ва чиқувчи барча блокларнинг тенг тақсимотларини таъминлаганлиги учун бу акслантиришлар чизиқли ва дифференциал криптотаҳлил усулларига бадошли бўлади.

§5.5. Чекли майдонда матрицали кенгайтириш ва жадвалли сиқиш акслантиришларига асосланган узлуксиз шифрлаш алгоритми

Бу алгоритмнинг асосий ўлчови $2^t_{2\times4}$ бўлган тўғри тўртбурчакли $A^t_{2\times4}$ матрица ва ўлчови 16x16 бўлиб элементлари ярим байтдан иборат бўлган (0 дан 15 гача сонлар) СЖ дан ташкил топган. Тўғри тўртбурчакли $A^t_{2\times4}$ матрицанинг иккита устуни пропорционал қилиб танланиб, бу матрица билан аниқланган акслантиришга тескари акслантиришни қуриш имкони мавжуд эмас. Бундан ташқари матрица элементларининг жуфт-жуфти билан ҳар-хил бўлиши, бу акслантиришга кирувчи ва чиқувчи блокларнинг текис тақсимланишини таъминлайди.

Босқичлари:

1. K – 256 битли калит 4 байтли 8 та қисмга ажратилади;

2. Калитнинг 4 байтли қисмлари $A^t_{2\times4}$ матрица орқали акслантирилиб, 2^t та байтга кенгайтирилади (мисол учун t =3 да 2^3=8 ва $A^3_{2\times4}=A_{8\times4}$);

3. Ҳар бир 2^t байтли блок СЖ орқали t марта қайта сиқишлар натижасида 1 байтга келтирилиб, бу байт битлари очиқ маълумотнинг 1 байти мос битларига XOR амали билан қўшилади;

4. Алгоритмда кўрсатилган қоида бўйича K калит байтлари аралаштирилиб, 1–3 босқичлар такрорланади.

Алгоритм асосида қуйидаги акслантиришлар ётади:

$(b_0+S)\text{mod}32$ – mod32 бўйича йиғинди;

$(b_Q+S)\text{mod}256$ – mod256 бўйича йиғинди;

$K<<S$ – S нинг қиймати бўйича K – калит блоки битларини циклик суриш;

w=СЖ(СЖ(СЖ(z)))– бир байтли z сонини сиқиш жадвалидан ўтказиб 1 битли қиймат «w» ҳосил қилиш.

Алгоритм акслантиришлари кетма-кетлиги

1) Тасодифий бўлган 256 битли калит $K=k_1, k_2, \ldots k_{255}, k_{256}$ киритилади ва бу калит 32 битли ёки 4 байтли 8 та блокка ажратилади:

$$K=k_1, k_2, \ldots k_{128},\ldots, k_{128+32L}(\text{бит})=x_1, x_2,\ldots, x_{16},\ldots, x_{16+4L}(\text{байт})=$$
$$=y_1, y_2,\ldots, y_{4+L}(32\text{ битли}) - \text{блоклар, бу ерда: } L=0, 1, 2, \ldots .$$

2) Ҳар бир $x_i=k_{1+8(i-1)}\ k_{2+8(i-1)p}\ k_{3+8(i-1)}\ldots k_{8+8(i-1)}$, i=1,2,...16+4L, бўлиб, агар L=4 бўлса, у ҳолда $K=k_1, k_2, \ldots k_{255}, k_{256} = x_1, x_2, \ldots, x_{31}, x_{32}= y_1, y_2, \ldots y_7, y_8$.

3) Киритилган калит массиви 4 байтли (32 битли) блокларга ажратилади:

$$y_{j4\times1}=(x_{1+4(j-1)}, x_{2+4(j-1)}, x_{3+4(j-1)}, x_{4+4(j-1)}).$$

4) Шундан сўнг $z^t_{j2\times1}=A^t_{2\times4}\ xy_{j4\times1}$ матрицали акслантириш амалга оширилади. Ҳосил бўлган блокнинг ҳар бир байтини сиқиш жадвалидан ўтказиб 1 битли гаммалар ҳосил қиламиз:

$$СЖ(СЖ(СЖ(z_{jv})))=w_{jv}(1\ бит),$$

яъни, $t=3$ бўлганда, битта блокнинг 8 та байтини сиқиш натижасида 8 битли қийматлар конкатенацияси

$$W_i=w_{j1}||w_{j2}||w_{j3}||w_{j4}||w_{j5}||w_{j6}||w_{j7}||w_{j8}$$

натижасида бир байтли W_i гамма ҳосил қилинади.

5) Очиқ маълумотнинг бир байти битларига ҳосил қилинган гаммanинг мос битларини XOR амали билан қўшиб, шифрмаълумот ҳосил қилинади:

$$c=W_i\oplus M.$$

Дастлабки киритилган бошланғич 256 битли калитни матрицали кенгайтириш ва жадвалли сиқиш асосида 64 битли гамма блок ҳосил қилинади. Гамма блок ҳосил қилингандан сўнг бошланғич калит тескари боғланиш орқали ўзгартирилиб борилади. Бу янги калитдан очиқ маълумотнинг кейинги 64 битли қисмини шифрлаш учун гамма блок ҳосил қилинади.

Узунлиги 256 бит бўлган 32 та 8 битли блоклардан иборат $K=b_0, b_1, \ldots b_{32}$ – калитдан тескари боғланиш орқали янги калит ҳосил қилиш қуйидагича амалга оширилади:

1) Ҳосил бўлган

$$W_i=w_{j1}||w_{j2}||w_{j3}||w_{j4}||w_{j5}||w_{j6}||w_{j7}||w_{j8}$$

битларнинг конкатенацияси орқали ифодаланувчи қийматлар орқали

$$S=(S+Wj)\bmod 256$$

ҳисобланиб, ҳосил бўлган S қиймат ва калит массивининг биринчи байти йиғиндиси ушбу

$$Q=(b_0+S)\bmod 32$$

ифода орқали топилади;

2) Натижа Q индексидаги калит байти қуйидагича акслантирилади

$$b_Q=(b_Q+S)\bmod 256;$$

3) Ушбу $K=K<<S$ циклик суриш натижасида янги калит массиви ҳосил қилинади.

Алгоритм ҳосил қиладиган тасодифий кетма-кетлик даври узунлиги $2^{256}\cdot 256^8=2^{320}$ битни ташкил қилади.

Кириш параметрлари:

K [256] – 256 битли калит;

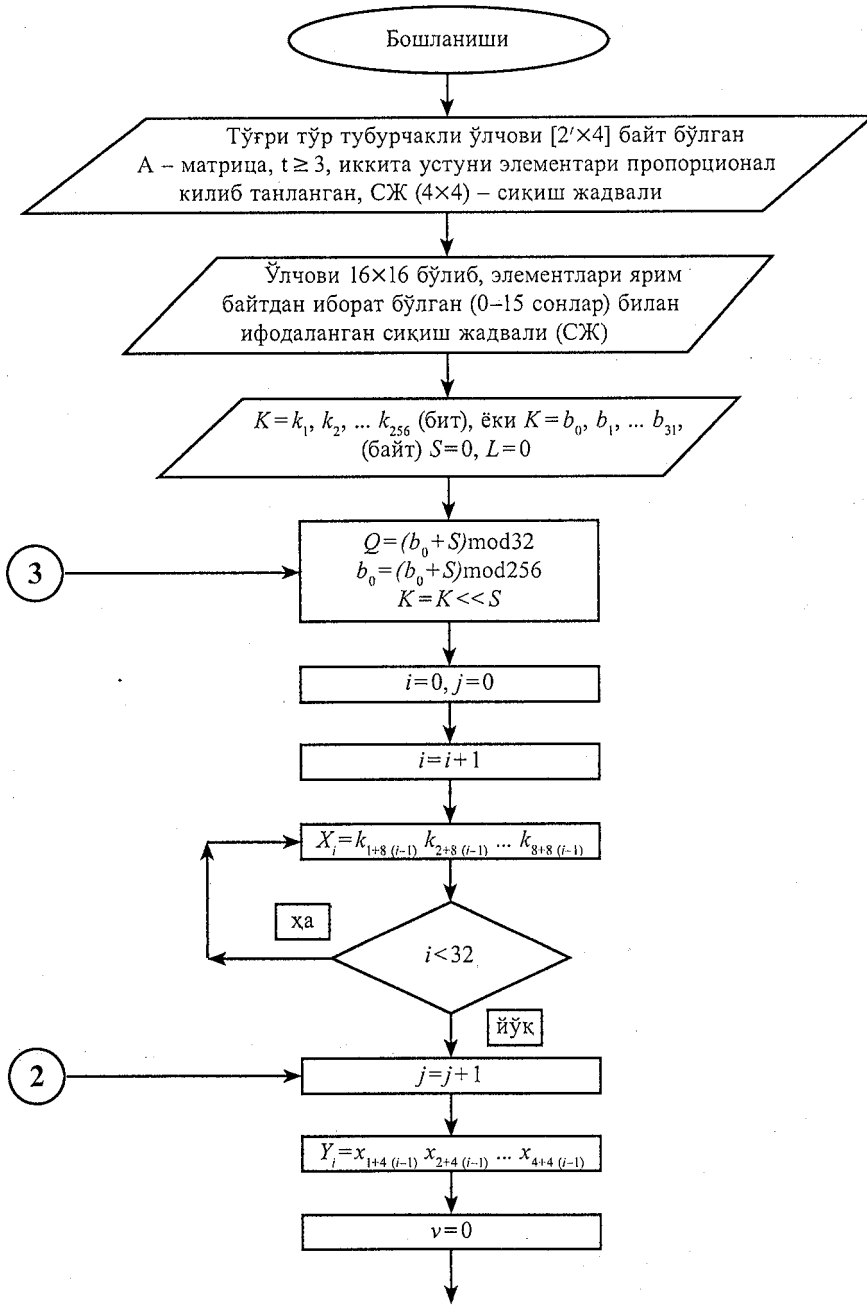
$M[d]$ – очиқ маълумот блоклари;

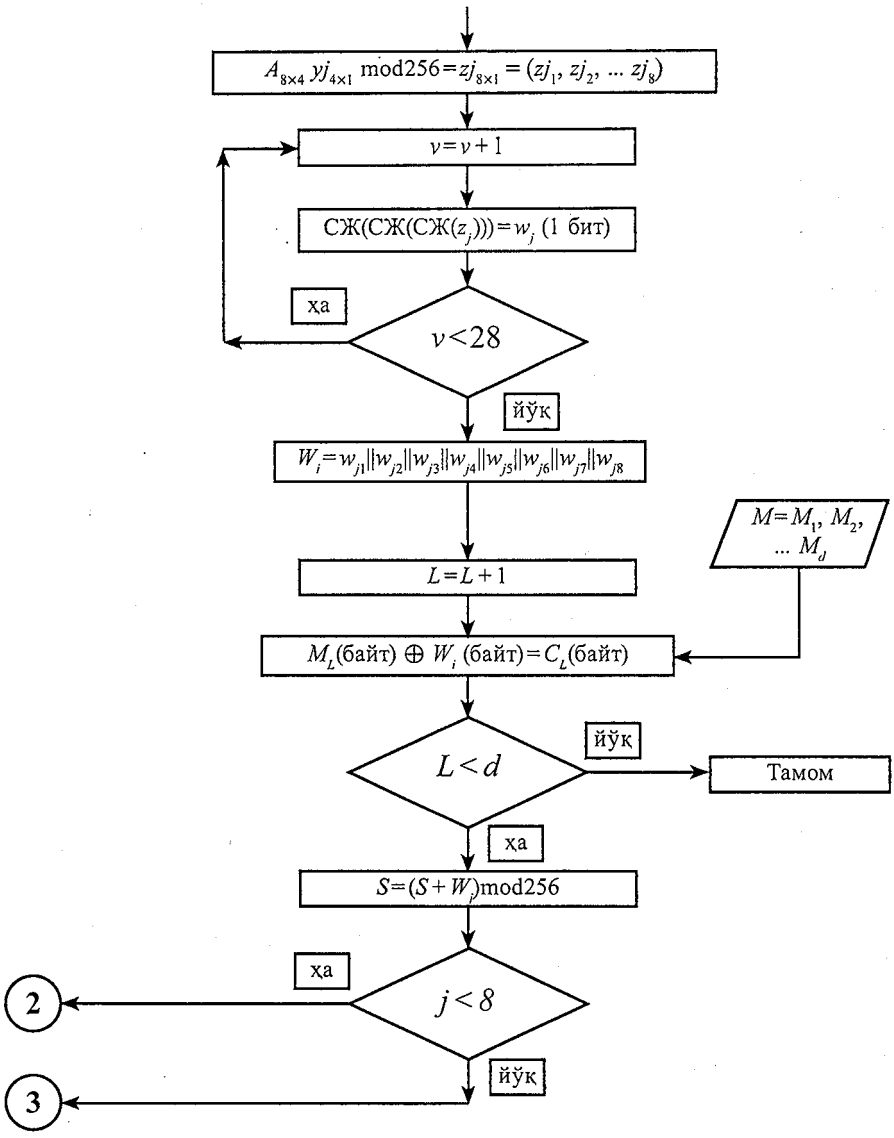
$A^t_{2\times 4}$ – матрица, $t=3$ бўлганда матрица ўлчами 8×4 бўлади;

СЖ[16×16] – сиқиш жадвали.

Натижа: $C[d]$ – шифрланган маълумотдан иборат.

Алгоритм блок схемаси




$A_{8\times4}\, yj_{4\times1}$ mod256$=zj_{8\times1}=(zj_1, zj_2, \ldots zj_8)$
$v=v+1$
СЖ(СЖ(СЖ(z_j)))$=w_j$ (1 бит)
ха
$v<28$
йўк
$W_i=w_{j1}||w_{j2}||w_{j3}||w_{j4}||w_{j5}||w_{j6}||w_{j7}||w_{j8}$
$M=M_1, M_2, \ldots M_d$
$L=L+1$
M_L(байт) ⊕ W_i (байт)$=C_L$(байт)
$L<d$
йўк
Тамом
ха
$S=(S+W_j)$mod256
ха
$j<8$
2
йўк
3

```
// Potochnoe_02Dlg.cpp : implementation file
//

#include «stdafx.h»
#include «Potochnoe_02.h»
#include «Potochnoe_02Dlg.h»
#include «.\potochnoe_02dlg.h»
#include «matr_TS.h»


#ifdef _DEBUG
#define new DEBUG_NEW
#endif


// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
        CAboutDlg();

// Dialog Data
        enum {IDD = IDD_ABOUTBOX};

        protected:
        virtual void DoDataExchange (CDataExchange* pDX);
// DDX/DDV support

// Implementation
protected:
        DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog (CAboutDlg::IDD)
{
}
void CAboutDlg :: DoDataExchange (CDataExchange* pDX)
```

```
{
        CDialog :: DoDataExchange (pDX);
}

BEGIN_MESSAGE_MAP (CAboutDlg, CDialog)
END_MESSAGE_MAP ()


// CPotochnoe_02Dlg dialog


CPotochnoe_02Dlg :: CPotochnoe_02Dlg (CWnd* pParent /*=NULL*/)
        : CDialog (CPotochnoe_02Dlg :: IDD, pParent)
        , ochiq1 (_T(«»))
        , shifr1 (_T(«»))
        , kalit1 (_T(«»))
        , len_1 (0)
{
        m_hIcon=AfxGetApp() – >LoadIcon (IDR_MAINFRAME);
}
void CPotochnoe_02Dlg :: DoDataExchange (CDataExchange* pDX)
{
        CDialog::DoDataExchange (pDX);
        DDX_Text (pDX, IDC_EDIT1, ochiq1);
        DDX_Text (pDX, IDC_EDIT2, shifr1);
        DDX_Text (pDX, IDC_EDIT3, kalit1);
        DDX_Text (pDX, IDC_EDIT4, len_1);
}

BEGIN_MESSAGE_MAP (CPotochnoe_02Dlg, CDialog)
        ON_WM_SYSCOMMAND()
        ON_WM_PAINT()
        ON_WM_QUERYDRAGICON()
        //}}AFX_MSG_MAP
        ON_BN_CLICKED (IDC_BUTTON1, OnBnClickedButton1)
        ON_BN_CLICKED (IDC_BUTTON2, OnBnClickedButton2)
        ON_BN_CLICKED (IDC_BUTTON3, OnBnClickedButton3)
        ON_BN_CLICKED (IDOK, OnBnClickedOk)
END_MESSAGE_MAP()
```

```
// CPotochnoe_02Dlg message handlers

BOOL CPotochnoe_02Dlg::OnInitDialog()
{
        CDialog :: OnInitDialog();

        // Add «About...» menu item to system menu.

        // IDM_ABOUTBOX must be in the system command range.
        ASSERT ((IDM_ABOUTBOX & 0×FFF0)==IDM_ABOUTBOX);
        ASSERT (IDM_ABOUTBOX < 0×F000);

        CMenu* pSysMenu = GetSystemMenu (FALSE);
        if (pSysMenu != NULL)
        {
                CString strAboutMenu;
                strAboutMenu.LoadString (IDS_ABOUTBOX);
                if (!strAboutMenu.IsEmpty())
                {
                        pSysMenu->AppendMenu(MF_SEPARATOR);
                        pSysMenu->AppendMenu(MF_STRING,
                        IDM_ABOUTBOX, strAboutMenu);
                }
        }

        // Set the icon for this dialog. The framework does this automatically
        // when the application's main window is not a dialog
        SetIcon(m_hIcon, TRUE);                 // Set big icon
        SetIcon(m_hIcon, FALSE);                // Set small icon

        // TODO: Add extra initialization here

        return TRUE; // return TRUE unless you set the focus to a control
}

void CPotochnoe_02Dlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if ((nID & 0×FFF0) == IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
```

```
        }
        else
        {
                CDialog :: OnSysCommand (nID, lParam);
        }
}

// If you add a minimize button to your dialog, you will need the code
below
// to draw the icon. For MFC applications using the document/view
model,
// this is automatically done for you by the framework.

void CPotochnoe_02Dlg::OnPaint()
{
        if (IsIconic())
        {
                CPaintDC dc(this); // device context for painting

                SendMessage(WM_ICONERASEBKGND,    reinterpret_
cast<WPARAM>(dc.GetSafeHdc()), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics(SM_CXICON);
                int cyIcon = GetSystemMetrics(SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width() - cxIcon + 1) / 2;
                int y = (rect.Height() - cyIcon + 1) / 2;

                // Draw the icon
                dc.DrawIcon(x, y, m_hIcon);
        }
        else
        {
                CDialog::OnPaint();
        }
}

// The system calls this function to obtain the cursor to display while the
user drags
```

```
// the minimized window.
HCURSOR CPotochnoe_02Dlg::OnQueryDragIcon()
{
        return static_cast<HCURSOR> (m_hIcon);
}

void CPotochnoe_02Dlg :: OnBnClickedButton1 ()
{
CFileDialog dlg(1);
        if(dlg.DoModal ()==IDOK)
        {
                ochiq1=dlg. GetPathName ();
                UpdateData (0);
        }
        else
        {
                MessageBox («Ochiq txt ochilmadi yoki fayl bilan ishlash-
da xato!!!»);
        }

        FILE *f1 = fopen(ochiq1,»rb»);
        fseek(f1,0, SEEK_END); //Файл узунлигини аниклаш учун
        len_1=ftell(f1);          // len га езиб куямиз
        rewind(f1);

        UpdateData(0);
}

void CPotochnoe_02Dlg :: OnBnClickedButton2 ()
{
CFileDialog dlg (1);
        if(dlg.DoModal ()==IDOK)
        {
                shifr1=dlg.GetPathName ();
                UpdateData(0);
        }
        else
        {
                MessageBox(«shifr txt ochilmadi yoki fayl bilan ishlashda
xato!!!»);
```

```
        }
        }

void CPotochnoe_02Dlg :: OnBnClickedButton3 ()
{
CFileDialog dlg (1);
        if(dlg.DoModal ()==IDOK)
        {
                kalit1=dlg.GetPathName ();
                UpdateData (0);
        }
        else
        {
                MessageBox(«Kalit txt ochilmadi yoki fayl bilan ishlashda
xato!!!»);
        }
        }

void CPotochnoe_02Dlg :: OnBnClickedOk()
{
UpdateData (1);

        FILE *f1=fopen(ochiq1,»rb»);
                if(f1==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return ;
                }

        FILE *f2 = fopen (shifr1, «wb»);
                if(f2==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }
        FILE *f3=fopen (kalit1,»rb»);
                if(f2==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }

        //-------------------kiritiladi---------------------------------------
                byte buff_gamma 1 [4];
```

```
byte buff_gamma 2 [8];
byte buff_kalit [32];
byte buff_kalit 2 [32];
byte buff_och [8];
byte buff_sht [8];
__int64 buff_kalit 64 [4];
byte v1, v2, len_3, len4, s3, s4, sd1, sd2;
UINT len_k, i, j, s1=0, s2;
//-----------------------------------------------------------------------


len_k=fread (buff_kalit, 1,32, f3);
if (len_k<32)
{
MessageBox(«Etarli kalit uzunligi yuq»);
return;
}

matr_TS ad1;

for(;;)
{
len_3=fread (buff_och, 1, 8, f1);
if (len_3==0)break;


                for(i=0; i<len_3; i++)
                {
                buff_gamma1 [0]=buff_kalit [i*4];
                buff_gamma1 [1]=buff_kalit [i*4+1];
                buff_gamma1 [2]=buff_kalit [i*4+2];
                buff_gamma1 [3]=buff_kalit [i*4+3];

                v1=ad1.func0 (buff_gamma1);
                buff_sht [i]=buff_och [i]^v1;
                        s1=(s1+v1)%256;
                }
                fwrite (buff_sht, 1, len_3, f2);

        s2=(buff_kalit [0]+s1)%32;
```

```
            buff_kalit [s2]=(buff_kalit [s2]+s1)%256;
            s3=s1/8;
            s4=s1%8;

            for (i=0; i<32; i++)
            {
            buff_kalit 2 [i]=buff_kalit [(i+s3)%32];
            }
            sd1=buff_kalit 2 [0]>>(8-s4);
for (i=0; i<31; i++)
            {
            buff_kalit 2 [i]=(buff_kalit 2 [i]<<s4)^
            ^(buff_kalit 2 [i+1]>>(8-s4));
            buff_kalit [i]=buff_kalit 2 [i];
            }
            buff_kalit 2 [31]=(buff_kalit 2 [31]<<s4)^sd1;
            buff_kalit [31]=buff_kalit 2 [31];

}

fcloseall ();

}
```

Таклиф этилган алгоритмнинг асосини ташкил этувчи матрицали кенгайтириш ва сиқиш жадвали хоссалалари уларнинг амалий бир томонлама акслантиришлар бўлишини, чизиқли ҳамда дифференциал криптотаҳлил усулларига бардошлилигини таъминлайди.

§5.6. Бир томонлама мантиқий функцияларга асосланган генератор

Бу генератор алгоритмининг асосини тўртта 4 аргументли мантиқий функция:

1) $F_i = X_i\, Y_i\, (Z_i \oplus W_i)\, W_i$ – байтлар устида F мантиқий акслантириш;

2) $G_i = W_i\, (X_i Z_i \oplus Y_i) \oplus Z_i W_i$ – байтлар устида G мантиқий акслантириш;

3) $R_i = X_i Y_i Z_i \oplus Z_i W_i \oplus X_i W_i \oplus Y_i W_i$ – байтлар устида R мантиқий акслантириш;

4) $V_i = Y_i W_i Z_i \oplus Z_i W_i \oplus X_i$ – байтлар устида V мантиқий акслантириш;

5) Ўлчами 16×16 бўлиб, элементлари ярим байтдан иборат бўлган (0 дан 15 гача сонларнинг текис тақсимотидан иборат) СЖ:

y / x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	5	13	6	11	1	10	15	8	0	4	7	9	2	12	3	14
1	8	7	2	14	15	3	11	6	1	12	13	10	5	4	9	0
2	14	2	13	4	12	7	1	11	6	9	0	5	3	10	8	15
3	0	14	9	12	3	13	7	4	15	6	5	1	11	2	10	8
4	3	10	7	2	4	12	9	1	14	13	15	8	0	5	11	6
5	2	3	1	8	0	14	5	9	12	11	6	7	10	15	13	4
6	10	4	14	15	9	5	8	2	11	0	1	3	12	6	7	13
7	11	9	10	1	6	4	13	15	3	5	14	0	8	7	2	12
8	1	0	3	7	13	11	10	12	9	14	4	6	15	8	5	2
9	4	8	11	9	14	6	2	5	10	3	12	15	7	13	0	1
10	9	12	15	0	2	1	14	10	5	8	11	13	4	3	6	7
11	6	11	8	13	7	9	0	3	4	15	10	2	14	1	12	5
12	15	1	0	5	10	8	3	7	13	2	9	12	6	14	4	11
13	12	5	4	10	11	2	6	13	8	7	3	14	1	0	15	9
14	7	15	12	6	5	0	4	14	2	10	8	11	13	9	1	3
15	13	6	5	3	8	15	12	0	7	1	2	4	9	11	14	10

ташкил этади. Бир байт иккита ва ярим байтларга ажратилади, сўнгра бу ярим байтларнинг қийматлари бўйича мос равишда сатр ҳамда устун тартиб сонлари (номерлари) топилиб, шу сатр ва устунлар кесишган жойдаги соннинг иккилик саноқ тизимидаги ярим байтли ифодасини бир байтли ифоданинг сиқиш жадвали орқали акслантириш натижаси сифатида қабул қилинади.

Босқичлари:

1. K – 256 битли калит 4 байтли 8 та қисмга ажратилади;

2. Ҳар бир 4 байтли калитдан мантиқий акслантиришлар орқали 4 байтли блок ҳосил қилинади;

3. Ҳосил қилинган 4 байтли блок 2 марта СЖ орқали акслантирилиб, 1 байтли блок ҳосил қилинади;

4. Генератор алгоритмида кўрсатилган қоида бўйича K – калитнинг байтлари аралаштирилиб, 1–4 босқичлар такрорланади.

Алгоритм акслантиришлари кетма-кетлиги

Тасодифий бўлган 256 битли калит $K=k_1, k_2, \ldots k_{255}, k_{256}$ киритилади ва бу калит 32 битли ёки 4 байтли 8 та блокка ажратилади:

1) $K=k_1, k_2, \ldots k_{128},\ldots, k_{128+32L}$(бит)$=x_1, x_2,\ldots, x_{16},\ldots, x_{16+4L}$(байт)$=$ $=y_1, y_2,\ldots, y_{4+L}$(32 битли) – блоклар, бу ерда $L=0, 1, 2, \ldots$.

2) Ҳар бир $x_i=k_{1+8(i-1)}\, k_{2+8(i-1)p}\, k_{3+8(i-1)} \ldots k_{8+8(i-1)}$, $i=1,2, \ldots 16+4L$, бўлиб, агар $L=4$ бўлса, у ҳолда

$$K=k_1, k_2, \ldots k_{255}, k_{256}=x_1, x_2, \ldots, x_{31}, x_{32}=y_1, y_2,\ldots y_7, y_8.$$

3) Киритилган калит массиви 4 байтли (32 битли) блокларга ажратилади:

$$y_{j4\times1}=(x_{1+4(j-1)}, x_{2+4(j-1)}, x_{3+4(j-1)}, x_{4+4(j-1)}).$$

Биринчи 4 байтли блокнинг (X_i, Y_i, Z_i, W_i) ҳар бир байтлари устида мантиқий акслантиришларни бажариб янги 4 байтли $(F_i\|G_i\|R_i\|V_i)$ блокка эга бўламиз. Бу блок байтларини сиқиш жадвалидан ўтказилади ва икки баробар сиқилган 2 байтли блокка $(A\|B)$ эга бўлинади. $(A\|B)$ – блок ҳам сиқиш жадвалидан ўтказилиб, 1 байтли D – блок ҳосил қилинади. D – блок гамма кетма-кетлик элементи сифатида қабул қилинади. Бошланғич берилган 256 битли калитдан шу тариқа 8 байт (64 бит) гамма кетма-кетлик элементлри олинади. 256 битли калитни ўзгартириш тескари боғланиш акслантириши орқали амалга оширилади.

Калитни ўзгартирувчи тескари боғланиш акслантириши қуйидагича бажарилади:

1) Гамма кетма-кетлик ҳосил қилиш жараёнида ҳосил қилинган оралиқ A ва B – блокларни mod 256 бўйича S ўзгарувчига йиғиб борилади:

$$S=(S+A+B)\text{mod}256.$$

2) Сўнгра, 64 бит гамма кетма-кетлик олиниб бошланғич калитнинг барча байтлари ишлатилгандан сўнг K – калитнинг биринчи b_0– байти орқали

$$Q=(b_0+S)\text{mod}32$$

ҳисобланади.

3) «Q» индексда турувчи b_Q қийматини ўзгартирилади:

$$b_Q=(b_Q+S)\text{mod}256.$$

4) Шундан сўнг ўзгартирилган 256 битли K – калит S қийматга циклик сурилади:

$$K=K<<S$$

ва натижада ўзгартирилган янги 256 битли калитга эга бўлинади.

Бу калит кейинги 8 байтли гаммаларни ҳосил қилишда ишлатилади.

K – калит 2^{256} та ҳар-хил қийматга эга бўлиши мумкин. Ҳар бир қийматдан 2^{64} та гамма кетма-кетлик битлари ҳосил қилинади ва гамма кетма-кетлик даври узунлиги $2^{256}\cdot 2^{64}=2^{320}$ битни ташкил қилади.

Алгоритм асосида қуйидаги акслантиришлар ётади:

$(b_0+S)\ \text{mod}\,32-\text{mod}\,32$ бўйича йиғинди;

$(b_Q+S)\ \text{mod}\,256-\text{mod}\,256$ бўйича йиғинди;

$K<<S$ – S нинг қиймати бўйича K – калитни циклик суриш;

$F_i=X_iY_i(Z_i\oplus W_i)\oplus W_i$ – байтлар устида мантиқий акслантиришни бажариш;

$G_i=W_i(X_iZ_i\oplus Y_i)\oplus Z_iW_i$ – байтлар устида мантиқий акслантиришни бажариш;

$R_i=X_iY_iZ_i\oplus Z_iW_i\oplus X_iW_i\oplus Y_iW_i$ – байтлар устида мантиқий акслантиришни бажариш;

$V_i=Y_iW_iZ_i\oplus Z_iW_i\oplus X_i$ – байтлар устида мантиқий акслантиришни бажариш;

СЖ(F_i), СЖ(G_i), СЖ(R_i), СЖ(V_i) – 1 байтли блокларни ярим байтларга сиқиш;

$A=$СЖ(F_i)||СЖ(G_i) ва $B=$СЖ(R_i)||СЖ(V_i) – ярим байтли блоклар конкатенацисидан бир байтли блоклар олиш;

$D=$СЖ(A)||СЖ(B) – ярим байтли сиқиш натижаларининг конкатенацияси (1 байт);

$C_L=M_L\oplus D$ – XOR амали орқали гамма блок битларини очиқ маълумот блоки мос битларига қўшиш.

Кириш параметрлари:

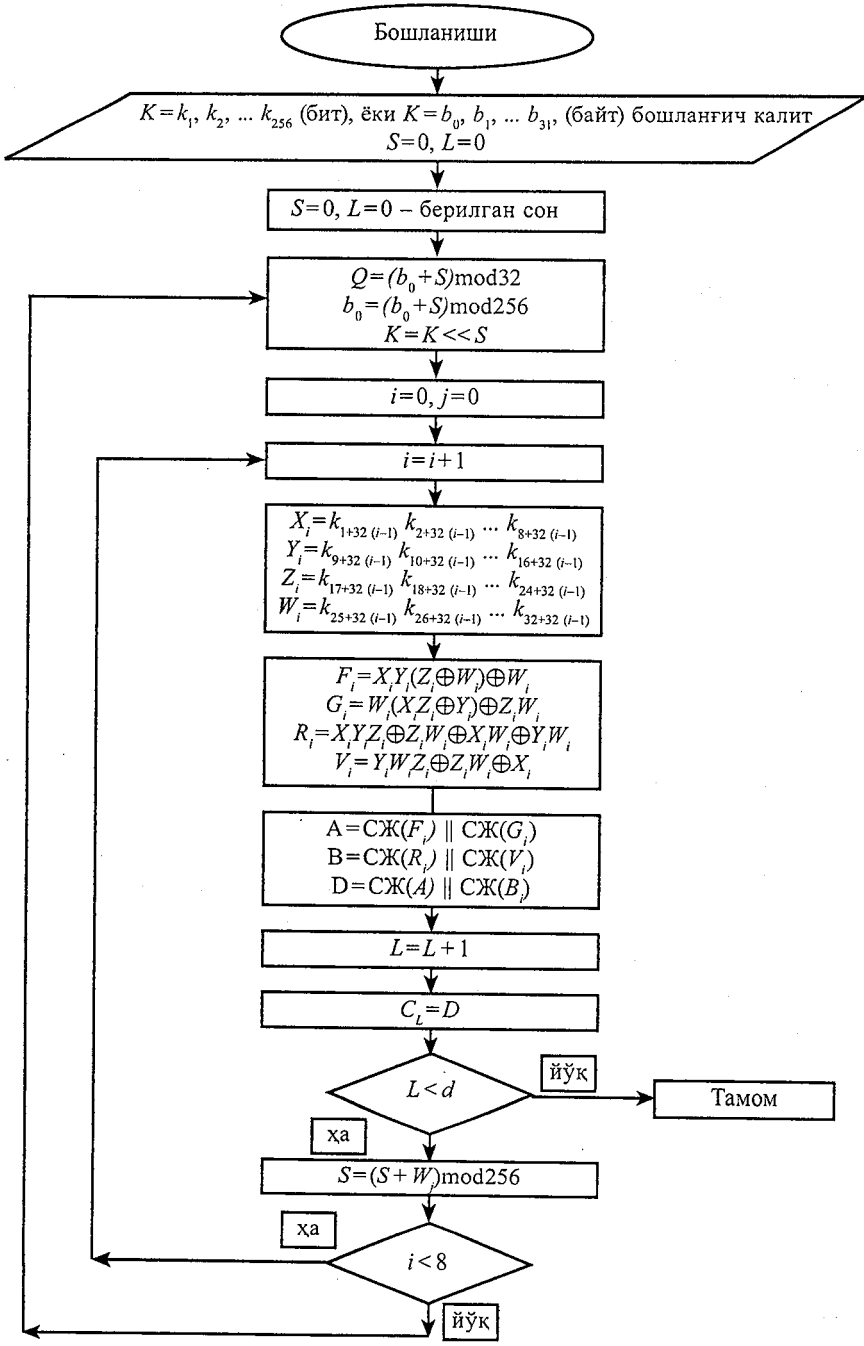
K [256] – 256 битли калит;

$M[d]$ – очиқ маълумот блоклари;

СЖ[16×16] – сиқиш жадвали.

Натижа: C_L – шифрланган маълумотдан иборат.

Алгоритмнинг блок схемаси



```
// Potochnoe_03Dlg.cpp : implementation file
//

#include «stdafx.h»
#include «Potochnoe_03.h»
#include «Potochnoe_03Dlg.h»
#include «.\potochnoe_03dlg.h»
#include «logik_TS.h»

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
        CAboutDlg ();

// Dialog Data
        enum { IDD = IDD_ABOUTBOX };

        protected:
        virtual void DoDataExchange (CDataExchange* pDX); // DDX/
DDV support

// Implementation
protected:
        DECLARE_MESSAGE_MAP()
};

CAboutDlg :: CAboutDlg () : CDialog (CAboutDlg :: IDD)
{
}

void CAboutDlg::DoDataExchange (CDataExchange* pDX)
{
```

```
        CDialog::DoDataExchange (pDX);
}

BEGIN_MESSAGE_MAP (CAboutDlg, CDialog)
END_MESSAGE_MAP ()

// CPotochnoe_03Dlg dialog

CPotochnoe_03Dlg :: CPotochnoe_03Dlg (CWnd* pParent /*=NULL*/)
        : CDialog (CPotochnoe_03Dlg::IDD, pParent)
        , ochiq1 (_T(«»))
        , shifr1 (_T(«»))
        , kalit1 (_T(«»))
        , len_1 (0)
        , st11 (0)
{
        m_hIcon = AfxGetApp ()– >LoadIcon (IDR_MAINFRAME);
}

void CPotochnoe_03Dlg : :DoDataExchange (CDataExchange* pDX)
{
        CDialog::DoDataExchange (pDX);
        DDX_Text (pDX, IDC_EDIT1, ochiq1);
        DDX_Text (pDX, IDC_EDIT2, shifr1);
        DDX_Text (pDX, IDC_EDIT3, kalit1);
        DDX_Text (pDX, IDC_EDIT4, len_1);
        //DDX_Text (pDX, IDC_EDIT5, st11);
        DDX_Control (pDX, IDC_PROGRESS1, pr1);
}

BEGIN_MESSAGE_MAP (CPotochnoe_03Dlg, CDialog)
        ON_WM_SYSCOMMAND ()
        ON_WM_PAINT ()
        ON_WM_QUERYDRAGICON ()
        //}}AFX_MSG_MAP
        ON_BN_CLICKED (IDC_BUTTON1, OnBnClickedButton1)
        ON_BN_CLICKED (IDC_BUTTON2, OnBnClickedButton2)
        ON_BN_CLICKED (IDC_BUTTON3, OnBnClickedButton3)
        ON_BN_CLICKED (IDOK, OnBnClickedOk)
END_MESSAGE_MAP ()
```

```
// CPotochnoe_03Dlg message handlers

BOOL CPotochnoe_03Dlg :: OnInitDialog ()
{
        CDialog::OnInitDialog ();

        // Add «About...» menu item to system menu.
        // IDM_ABOUTBOX must be in the system command range.
        ASSERT ((IDM_ABOUTBOX & 0×FFF0) == IDM_ABOUT-
BOX);
        ASSERT (IDM_ABOUTBOX < 0×F000);

        CMenu* pSysMenu=GetSystemMenu (FALSE);
        if (pSysMenu !=NULL)
        {
                CString strAboutMenu;
                strAboutMenu.LoadString (IDS_ABOUTBOX);
                if (!strAboutMenu.IsEmpty ())
                {
                        pSysMenu – >AppendMenu (MF_SEPARATOR);
                        pSysMenu – >AppendMenu (MF_STRING, IDM_
ABOUTBOX, strAboutMenu);
                }
        }
        // Set the icon for this dialog. The framework does this automati-
cally
        // when the application's main window is not a dialog
        SetIcon (m_hIcon, TRUE);                    // Set big icon
        SetIcon (m_hIcon, FALSE);              // Set small icon
        // TODO: Add extra initialization here

        return TRUE; // return TRUE unless you set the focus to a
control
}
void CPotochnoe_03Dlg :: OnSysCommand (UINT nID, LPARAM lParam)
{
        if ((nID & 0×FFF0)==IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
```

```
        }
        else
        {
                CDialog :: OnSysCommand (nID, lParam);
        }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CPotochnoe_03Dlg::OnPaint()
{
        if (IsIconic ())
        {
                CPaintDC dc(this); // device context for painting

                SendMessage(WM_ICONERASEBKGND,
reinterpret_cast < WPARAM > (dc.GetSafeHdc ()), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics (SM_CXICON);
                int cyIcon = GetSystemMetrics (SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width () – cxIcon+1)/2;
                int y = (rect.Height () – cyIcon+1)/2;

                // Draw the icon
                dc.DrawIcon (x, y, m_hIcon);
        }
        else
        {
                CDialog :: OnPaint ();
        }
}
// The system calls this function to obtain the cursor to display while the
user drags
// the minimized window.
HCURSOR CPotochnoe_03Dlg :: OnQueryDragIcon ()
```

```
{
        return static_cast < HCURSOR > (m_hIcon);
}
void CPotochnoe_03Dlg :: OnBnClickedButton1 ()
{
CFileDialog dlg(1);
        if (dlg.DoModal ()==IDOK)
        {
                ochiq1=dlg. GetPathName ();
                UpdateData (0);
        }
        else
        {
        MessageBox («Ochiq txt ochilmadi yoki fayl bilan ishlashda xato!!!»);
        }

        FILE *f1 = fopen(ochiq1, «rb»);
        fseek (f1,0, SEEK_END); // Файл узунлигини аниклаш учун
        len_1=ftell (f1);                    // len га езиб куямиз
        rewind(f1);

        UpdateData (0);
}

void CPotochnoe_03Dlg :: OnBnClickedButton2 ()
{
CFileDialog dlg (1);
        if(dlg.DoModal ()==IDOK)
        {
                shifr1=dlg.GetPathName ();
                UpdateData (0);
        }
        else
        {
                MessageBox («shifr txt ochilmadi yoki fayl bilan ishlashda
xato!!!»);
        }
        }
void CPotochnoe_03Dlg::OnBnClickedButton3 ()
{
```

```
CFileDialog dlg (1);
        if(dlg.DoModal ()==IDOK)
        {
                kalit1=dlg.GetPathName ();
                UpdateData (0);
        }
        else
        {
                MessageBox («Kalit txt ochilmadi yoki fayl bilan ishlashda
xato!!!»);
        }
        }
void CPotochnoe_03Dlg :: OnBnClickedOk ()
{
UpdateData(1);
        FILE *f1 = fopen (ochiq1, «rb»);
                if (f1==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }
        fseek(f1,0, SEEK_END); //Файл узунлигини аниклаш учун int
len_1p=ftell(f1);                    // len га езиб куямиз
        rewind(f1);
        FILE *f2 = fopen (shifr1,»wb»);
                if (f2==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }
        FILE *f3 = fopen (kalit1,»rb»);
                if (f2==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }
                //------------------kiritiladi-----------------------------
                byte buff_gamma1[4];
                byte buff_kalit [32];
                byte buff_kalit 2 [32];
                byte buff_och [8];
                byte buff_sht [8];
                byte v1, len_3, s3, s4, sd1;
```

```
            UINT len_k, i, s1=0, s2;
            int s_pr=0, len_1i;
        //-------------------------------------------------------------
        len_k=fread (buff_kalit, 1,32, f3);
            if (len_k<32)
            {
            MessageBox («Etarli kalit uzunligi yuq»);
            return;
            }
pr1.SetRange32 (0,len_1p);
            logik_TS ad1;
for(;;)
            {
            len_3=fread (buff_och, 1, 8, f1);
            if (len_3==0)break;
                for (i=0; i<len_3; i++)
                {
                buff_gamma1 [0]=buff_kalit [i*4];
                buff_gamma1 [1]=buff_kalit [i*4+1];
                buff_gamma1 [2]=buff_kalit [i*4+2];
                buff_gamma1 [3]=buff_kalit [i*4+3];
                s_pr++;
                pr1.SetPos (s_pr);
                v1=ad1.func0 (buff_gamma1);
                buff_sht [i]=buff_och [i]^v1;
                s1=(s1+buff_gamma1 [0]+buff_gamma1 [1])%256;
                }
                fwrite (buff_sht, 1, len_3, f2);
                s2=(buff_kalit [0]+s1)%32;
                buff_kalit [s2]=(buff_kalit [s2]+s1)%256;
                s3=s1/8;
                s4=s1%8;
                for (i=0; i<32; i++)
                {
                buff_kalit 2 [i]=buff_kalit [(i+s3)%32];
                }
                sd1=buff_kalit 2 [0] >> (8–s4);
            for(i=0; i<31; i++)
                {
                buff_kalit 2 [i]=(buff_kalit 2 [i]<<s4)^(buff_kalit2
[i+1] >> (8–s4));
```

```
                    buff_kalit [i]=buff_kalit 2 [i];
                    }
                    buff_kalit 2 [31]=(buff_kalit 2 [31]<<s 4)^sd1;
                    buff_kalit [31]=buff_kalit 2 [31];
            }
fcloseall();
UpdateData(0);
}
```

Бу таклиф этилган алгоритмнинг асосини ташкил этувчи мантиқий акслантиришлар, уларга кирувчи ва чиқувчи барча блокларнинг текис тақсимотини таъминлайди. Мантиқий акслантиришлар ва сиқиш жадвали хоссалари уларнинг амалий бир томонлама акслантиришлар бўлишини, чизиқли ҳамда дифференциал криптотаҳлил усулларига бардошлиликни кафолатлайди.

§5.7. Криптобардошли алгоритмларни комбинациялашга асосланган узлуксиз шифрлаш алгоритми

Алгоритмнинг кириш параметрлари қуйидагилардан иборат:

K-калит, ўзгарувчан узунликка эга бўлиб, 8 га каррали ва 128 битдан катта;

w – калит узунлиги (байтларда);

S – блок узунлиги 256 байт, бошланғич ҳолати 0 дан 255 гача сонлар билан кетма-кет тўлдирилади.

Алгоритм асосида қуйидаги акслантиришлар ётади:

S – блокнинг байтлардан иборат бўлган S_i – элементларини *K*-калитнинг байтлардан иборат бўлган K_i – қисмблокларини характеристикаси 256 бўлган чекли майдонда ушбу $(j+S_i+K_j)$ mod256 формула билан қўшиш;

Циклик суриш $a<<b$, яъни бир байтли «*a*» сонини «*b*» сонининг охирги 3 битига тенг бўлган қийматга суриш;

S – блокнинг S_j ва S_i – элементлари ўринларини алмаштирш.

Бу алгоритм байтлар ва битлар ўрнини боғлиқсиз алмаштиришга асосланган 5 генераторни комбинациялашга асосланган. Ҳар бир генератор алоҳида калитларда гамма кетма-кетлик ҳосил қилишга тайёргарлик жараёнини амалга оширади. Шундан сўнг биринчи генератор ҳосил қилган гамма блок охирги 2 бити қийматига қараб қолган 4 генераторнинг қайси биридан чиқиш гаммаси блоки олиниши аниқланади. Аниқланган тартибли генератор акслантиришлари амалга оширилади (тактланади) ва ҳосил қилган гамма блок битлари очиқ маълумот блоки мос битларига XOR амали билан қўшишга узатилади.

Бу алгоритмдаги генераторлар сонини 256 тагача етказиш имкони мавжуддир. Ҳозирги ҳисоблаш қурилмалари техника ва технологиялари ривожланган шароитда, комбинациялашган 256 та генераторли алгоритм учун 256 та 256 байтли массив ажратиш (умумий миқдори 65 Кбайт бўлган) муаммо эмас. Шу сабабли замонавий аппарат-техник ва аппарат-дастурий микропроцессорлар тез ишловчи хотирасининг кенгайиб бориши кейинчалик гамма кетма-кетлик даври узунлиги 2^{512000} байт бўлган комбинациялашган алгоритмнинг етарли даражада тез ишлашини таъминловчи аппарат-техник ёки аппарат-дастурий криптографик воситаларини яратиш имконини беради.

Алгоритм таркибидаги битта генераторнинг гамма кетма-кетлик ҳосил қилишга тайёрланиш жараёни:

S – блок дастлаб 0 дан 255 гача бўлган сонлар билан кетма-кет тўлдирилиб чиқилади. Тасодифий танлаб олинган, узунлиги 32 байтдан 256 байтгача бўлган, К – калит ёрдамида S – блок 3 марта аралаштирилиб чиқилади.

S – блокни калит билан бир марта аралаштириш қуйидагича амалга оширилади:

$i=(i+1) \bmod 256$,

$j=(j+S_i+K_i) \bmod 256$,

$S_j<<(S_i+K_i)$,

S_i ва S_j ўрни алмаштирилади.

Уч марта аралаштирилган *S* – блокдан гамма блок ҳосил қилиш жараёни бошланади. Бу жараён қуйидагича амалга оширилади:

$i=(i+1) \bmod 256$,

$j=(j+S_i) \bmod 256$,

S_i ва S_j ўрни алмаштирилади,

$v = (S_i + S_j) \bmod 256$

$V = S_v$.

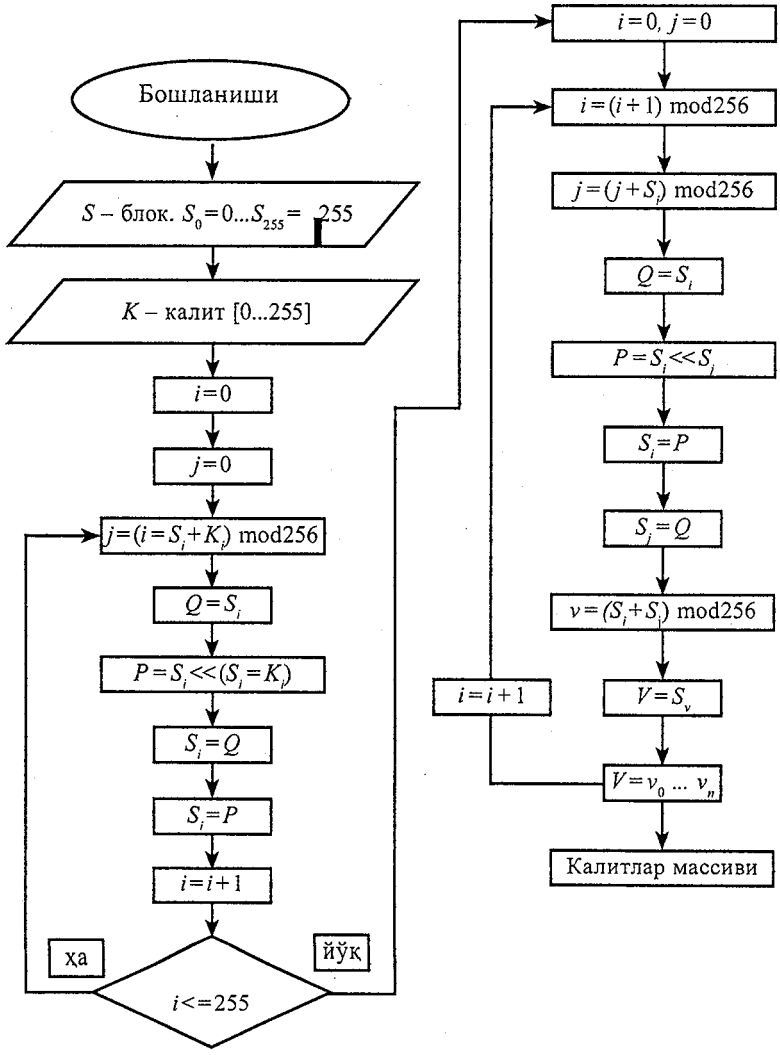
Ҳосил қилинган *V* калит блоки гаммаси сифатида узатилади ва очиқ маълумот блокига XOR амали билан қўшишга узатилади.

Комбинациялашган алгоритм таркибидаги 5 та генераторнинг ҳар бири учун алоҳида калит генерация қилинади. Калит генерацияси битта генератор орқали амалга оширилади. Алгоритм учун танланган *K*(128–2048 битгача) – калитдан 5 та 256 битли (1280 бит) гамма блок ҳосил қилинади. Ҳар бир алоҳида 256 битли K_1, K_2, K_3, K_4, K_5 – калитлар алгоритм таркибидаги генераторлар учун алоҳида калит ҳисобланади. Ҳар бир генератор алоҳида олинган калит билан гамма блок ҳосил қилишга тайёрланиш жараёнини, яъни S-блок элементларини калит элементлари билан аралаштириш босқичини амалга оширгандан сўнг генератор гамма блок ҳосил қилиш жараёнига тайёр ҳисобланади. Ҳар бир тактда бош генератор ҳосил қилган гамма блокнинг охирги 2 бити қиймати бўйича

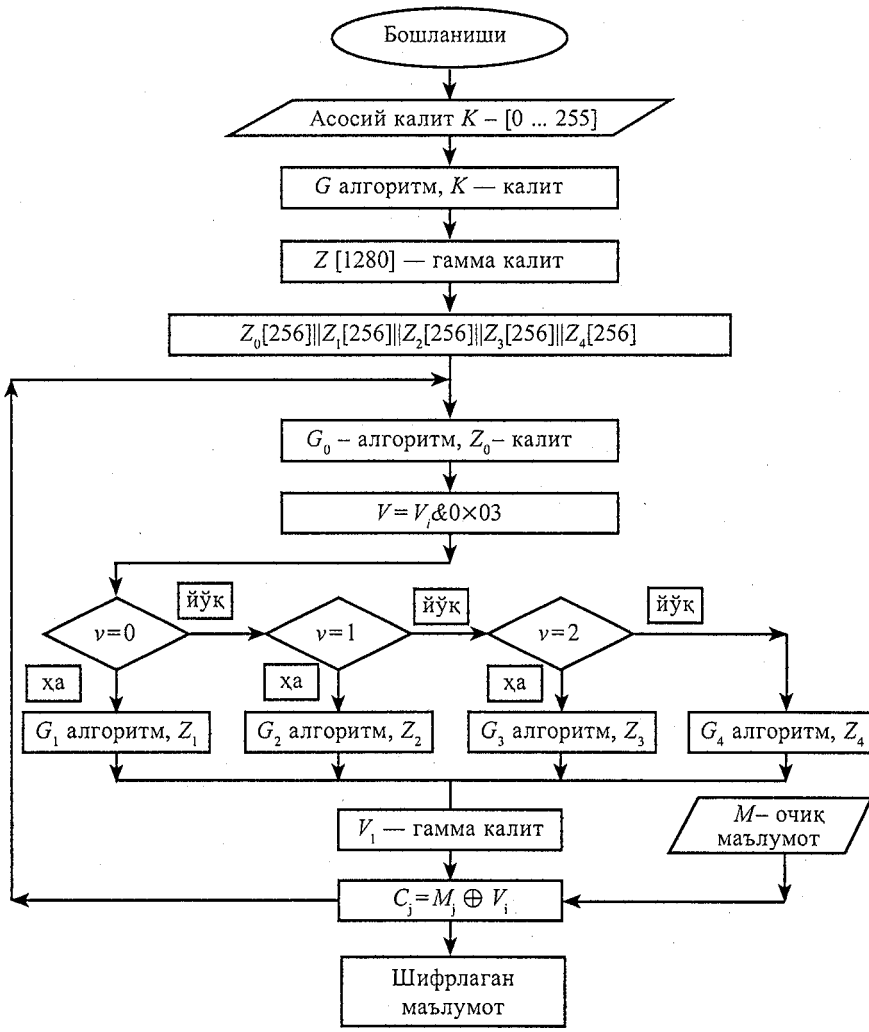
танланадиган генератор тартиби аниқланади. Мос тартибдаги генератор орқали ҳосил қилинган гамма блок – байт очиқ маълумот блоки – байтига XOR амали билан қўшилиб, шифрмаълумот блоки ҳосил қилинади.

Бевосита ҳисоблашлар билан алгоритмдаги битта генератор ишлаб чиққан гамма кетма-кетлик даври узунлиги 2^{2020} байтга тенглигига ишонч ҳосил қилиш мумкин. Алгоритм таркибида 5 генератор иштирок этганлиги сабабли умумий ҳосил бўлган гамма кетма-кетлик даври узунлиги $2^{2020 \cdot 5} = 2^{10100}$ байтни ташкил қилади.

Алгоритм таркибидаги битта генераторнинг блок-схемаси



Комбинациялашган алгоритмнинг умумий блок-схемаси



```
// Potochnoe_04Dlg.cpp : implementation file
//

#include «stdafx.h»
#include «Potochnoe_04.h»
#include «Potochnoe_04Dlg.h»
#include «.\potochnoe_04dlg.h»
#include «generator.h»

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
        CAboutDlg();

// Dialog Data
        enum {IDD = IDD_ABOUTBOX};

        protected:
        virtual void DoDataExchange (CDataExchange* pDX); // DDX/
DDV support

// Implementation
protected:
        DECLARE_MESSAGE_MAP()
};

CAboutDlg :: CAboutDlg() : CDialog (CAboutDlg :: IDD)
{
}

void CAboutDlg :: DoDataExchange (CDataExchange * pDX)
{
        CDialog::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP (CAboutDlg, CDialog)
```

```
END_MESSAGE_MAP ()

// CPotochnoe_04Dlg dialog

CPotochnoe_04Dlg::CPotochnoe_04Dlg (CWnd* pParent /*=NULL*/)
        : CDialog (CPotochnoe_04Dlg :: IDD, pParent)

        , ochiq1 (_T(«»))
        , shifr1 (_T(«»))
        , kalit1 (_T(«»))
        , len_1 (0)
        , len_k1 (0)
{
        m_hIcon = AfxGetApp ()->LoadIcon (IDR_MAINFRAME);
}

void CPotochnoe_04Dlg :: DoDataExchange (CDataExchange* pDX)
{
        CDialog::DoDataExchange (pDX);
        DDX_Text (pDX, IDC_EDIT4, len_1);
        DDX_Text (pDX, IDC_EDIT1, ochiq1);
        DDX_Text (pDX, IDC_EDIT2, shifr1);
        DDX_Text (pDX, IDC_EDIT3, kalit1);
        DDX_Text (pDX, IDC_EDIT4, len_1);
        DDX_Text (pDX, IDC_EDIT5, len_k1);
}

BEGIN_MESSAGE_MAP(CPotochnoe_04Dlg, CDialog)
        ON_WM_SYSCOMMAND ()
        ON_WM_PAINT ()
        ON_WM_QUERYDRAGICON ()
        //}}AFX_MSG_MAP
        ON_BN_CLICKED(IDC_BUTTON1, OnBnClickedButton1)
        ON_BN_CLICKED(IDC_BUTTON2, OnBnClickedButton2)
        ON_BN_CLICKED(IDC_BUTTON3, OnBnClickedButton3)
        ON_BN_CLICKED(IDOK, OnBnClickedOk)
END_MESSAGE_MAP ()

// CPotochnoe_04Dlg message handlers

BOOL CPotochnoe_04Dlg :: OnInitDialog ()
{
```

```
        CDialog :: OnInitDialog ();

        // Add «About...» menu item to system menu.

        // IDM_ABOUTBOX must be in the system command range.
        ASSERT ((IDM_ABOUTBOX & 0×FFF0)==IDM_ABOUTBOX);
        ASSERT (IDM_ABOUTBOX < 0×F000);

        CMenu* pSysMenu=GetSystemMenu (FALSE);
        if (pSysMenu !=NULL)
        {
                CString strAboutMenu;
                strAboutMenu.LoadString (IDS_ABOUTBOX);
                if (!strAboutMenu.IsEmpty ())
                {
                        pSysMenu–>AppendMenu (MF_SEPARATOR);
                        pSysMenu–>AppendMenu (MF_STRING, IDM_
ABOUTBOX, strAboutMenu);
                }
        }

        // Set the icon for this dialog. The framework does this automatically
        // when the application's main window is not a dialog
        SetIcon (m_hIcon, TRUE);            // Set big icon
        SetIcon (m_hIcon, FALSE);           // Set small icon

        // TODO : Add extra initialization here

        return TRUE; // return TRUE unless you set the focus to a control
}

void CPotochnoe_04Dlg :: OnSysCommand (UINT nID, LPARAM lParam)
{
        if ((nID & 0×FFF0) == IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
        }
        else
        {
```

```
                CDialog :: OnSysCommand (nID, lParam);
        }
}

// If you add a minimize button to your dialog, you will need the code
below
// to draw the icon. For MFC applications using the document/view
model,
// this is automatically done for you by the framework.

void CPotochnoe_04Dlg :: OnPaint ()
{
        if (IsIconic ())
        {
                CPaintDC dc (this); // device context for painting

                SendMessage (WM_ICONERASEBKGND, reinterpret_
cast <WPARAM> (dc.GetSafeHdc()), 0);

                // Center icon in client rectangle
                int cxIcon=GetSystemMetrics (SM_CXICON);
                int cyIcon=GetSystemMetrics (SM_CYICON);
                CRect rect;
                GetClientRect(&rect);
                int x = (rect.Width () – cxIcon+1)/2;
                int y = (rect.Height () – cyIcon+1)/2;

                // Draw the icon
                dc.DrawIcon (x, y, m_hIcon);
        }
        else
        {
                CDialog::OnPaint();
        }
}

// The system calls this function to obtain the cursor to display while the
user drags
// the minimized window.
HCURSOR CPotochnoe_04Dlg :: OnQueryDragIcon ()
```

```
{
        return static_cast < HCURSOR> (m_hIcon);
}

void CPotochnoe_04Dlg :: OnBnClickedButton1 ()
{UpdateData (1);
CFileDialog dlg (1);
        if(dlg.DoModal ()= =IDOK)
        {
                ochiq1 =dlg.GetPathName ();
                UpdateData (0);
        }
        else
        {
                MessageBox («Ochiq text ochilmadi yoki fayl bilan ishlash-
da xato!!!»);
        }

        FILE *f1 = fopen(ochiq1,»rb»);
        fseek(f1,0, SEEK_END);
        //Файл узунлигини аниқлаш учун len_1 =ftell(f1);
        // len га ёзиб қўямиз rewind(f1);
        UpdateData (0);
}

void CPotochnoe_04Dlg :: OnBnClickedButton2 ()
{
CFileDialog dlg (1);
        if(dlg.DoModal ()= =IDOK)
        {
                shifr1 =dlg. GetPathName ();
                UpdateData (0);
        }
        else
        {
        MessageBox(«shifr txt ochilmadi yoki fayl bilan ishlashda xato!!!»);
        }
}

void CPotochnoe_04Dlg :: OnBnClickedButton 3 ()
```

```
{UpdateData (1);

if (len_k1<16)
{
MessageBox(»Kalitni kiriting eki kalit uzunligini 16 baitdan katta oling»);
return;


}
CFileDialog dlg (1);
        if(dlg.DoModal ()==IDOK)
        {
                kalit1=dlg.GetPathName ();
                UpdateData (0);
        }
        else
        {
        MessageBox(«Kalit txt ochilmadi yoki fayl bilan ishlashda xato!!!»);
        }
}

void CPotochnoe_04Dlg::OnBnClickedOk ()
{
UpdateData(1);

        FILE *f1=fopen (ochiq1,»rb»);
                if(f1==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return ;
                }
        fseek (f1,0, SEEK_END); //Файл узунлигини аниклаш учун
int len_1p=ftell (f1);                          // len га езиб куямиз
        rewind (f1);
                FILE *f2=fopen (shifr1,»wb»);
                if (f2==NULL)
                {MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
                return;
                }

        FILE *f3=fopen (kalit1,»rb»);
```

```
if (f2==NULL)
{MessageBox («ochiq faylga ezish kerak, fayl ochilmadi»);
return ;
}

//---------------------------- ------------------------------
byte buff_gen [256], buff_gen 0 [256], buff_gen 1 [256],
buff_gen 2 [256], buff_gen 3 [256], buff_gen 4 [256];

byte buff_och [256], buff_kalit [256], buff_kalit 3 [256];
byte buff_kalit 2 [1280];
byte buff_sht [1];
byte buff_k 2 [1];
byte vl;
UINT len_g; // kerakli gamma uzunligi
UINT i, j, len_3, len_k ;
UINT j0, j1, j2, j3, j4;
UINT i0, i1, i2, i3, i4;
//-----------------------------------------------------------------

len_k=fread (buff_kalit, 1,256, f3);
if (len_k<256)
{
MessageBox («Etarli kalit uzunligi yuq»);
return;
}
for (i=0; i<256; i++)
{
buff_gen [i]=buff_gen 0 [i]=buff_gen1 [i ]=buff_gen
2 [i]=buff_gen 3 [i]=buff_gen 4 [i]=i;

}
generator ad1; // generator klassdan obekt hosil qilamiz
len_g=1280; //5 ta generator uchun 1280 bait kalit kerak
ad1.func 0 (buff_kalit, buff_gen, len_k1);
i=j=0;
vl=ad1. func1 (buff_gen, buff_kalit2, j, i, len_g);
for(i=0; i<5; i++)// 5 ta generator 1280 bait kalit bilan
aralashtiriladi
{
```

```
for(j=0; j<256; j++)
{
buff_kalit3 [j]=buff_kalit 2 [i * 256+j];
// 1280 5 ta kalitga ajratiladi
}
switch (i)
{case 0:
{
adl.func 0 (buff_kalit 3, buff_gen 0,256);
break;
}
case 1:
{
adl.func 0 (buff_kalit 3, buff_g enl, 256);
break;
}

case 2:
{
adl.func 0 (buff_kalit3, buff_gen 2, 256);
break;
}

case 3:
{
adl.func 0 (buff_kalit3, buff_gen 3, 256);
break;
}
case 4:

dl.func 0 (buff_kalit3, buff_gen 4, 256);
break;
 }
 }
 j0=j1=j2=j3=j4=0;
 i0=i1=i2=i3=i4=0;
for (;;)
{
len_3=fread (buff_och, 1, 1, f1);
if (len_3==0) break;
```

```
            vl=adl.funcl (buff_gen0, buff_k2, j0, i0, 1);
            j0=vl;
            i0=(i0+1) %256;
            switch (buff_k2 [0] &0×03)// asosiy generator bergan gam-
maning ohirgi 2 biti ga qarab qaysi generator ishlashini aniqlaymiz
            {
            case (0):
            {
            vl=adl.funcl(buff_genl,buff_k2, jl, il, 1);
            jl=vl;
            il=(il+1) %256;
            break;
            }
            case (1):
            {
            vl=adl.funcl (buff_gen2, buff_k2, j2, i2, 1);
            j2=vl;
            i2=(i2+1)%256;
            break;
            }
            case (2):
            {vl=adl. funcl (buff_gen3, buff_k2, j3, i3, 1);
            j3=vl;
            i3=(i3+1)%256;
            break;
            }
            case (3):
            {vl=adl.funcl (buff_gen4, buff_k2, j4, i4,1);
            j4=vl;
            i4=(i4+1) %256;
            break;
            }
            }
            buff_sht [0]=buff_och[0]^buff_k2[0];
            fwrite (buff_sht, 1, 1, f2);
            }
      fcloseall();
            }
```

Қуйидаги жадвалда янги яратилган ва бошқа мавжуд алгоритмларнинг баъзи криптографик хоссаларининг ўзаро солиштириш таҳлили келтирилган.

Алгоритм номи	Калит узунлиги (бит)	Даври (бит)	Тезлиги (Кбайт/сек)	Чиқиш гамма бўйича ҳужумга бардошлилиги
Силжитиш регистри	256	2^{256}	5	*Бардошли эмас*
Конгруент генератор	256	2^{256}	5	*Бардошли эмас*
RC4	128–2048	2^{1700}	164	*Бардошли*
SEAL	160	2^{160}	381	*Бардошли*
Байтлар ва битлар ўрнини боғлиқсиз алмаштириш	128-2048	2^{1711}	≈164	*Бардошли*
Матрицали кенгайтириш ва жадвалли сиқиш	256	2^{267}	<164	*Бардошли*
Бир томонлама мантиқий функциялар	256	2^{267}	<164	*Бардошли*
Комбинациялашган алгоритм	128–2048	$(2^{2020})^5 = 2^{10100}$	>164	*Бардошли*

5-боб бўйича хулосалар

Ушбу бобда:

1. Мавжуд узлуксиз шифрлаш алгоритмларининг асослари тадқиқ қилиниб, улар яратилиш йўналишларининг туркумлари (синфлари) изоҳлаб берилди.

2. Муаллиф томонидан, ахборот-коммуникация тизимларидаги маълумотларнинг муҳофазасини таъминловчи аппарат-техник воситаларда самарали қўлланувчи, янги криптобардошли акслантиришлардан фойдаланиб янги яратилган узлуксиз шифрлаш алгоритмлари мутахассислар ва шу соҳага қизиқувчи барча ўқувчиларнинг эътиборига ҳавола этилди.

3. Яратилган янги узлуксиз шифрлаш алгоритмларининг криптографик хусусиятларини ўрганиш ва таҳлил қилиш учун уларнинг дастурий таъминотлари берилди.

4. Олинган натижаларни криптологик нуқтаи назардан таҳлили ва янги яратилган узлуксиз шифрлаш алгоритмларининг криптобардошлилик даражаси баҳоланди. Навбатдаги бобда асимметрик шифрлаш алгоритмларининг асослари таҳлил қилиниб, уларнинг криптография масалаларини ечишда қўлланилиши ёритилади.

VI БОБ

АСИММЕТРИК ШИФРЛАШ АЛГОРИТМЛАРИ ВА УЛАРНИНГ КРИПТОГРАФИЯ МАСАЛАЛАРИНИ ЕЧИШДА ҚЎЛЛАНИЛИШИ

§ 6.1. Очиқ калитли криптотизимлар ҳақида

Ахборот-коммуникация тармоқларида маълумотлар алмашинувининг муҳофазасини таъминлаш масалаларини ечишда симметрик калитли крпитоалгоритмлар асосида яратилган криптотизим қанчалик ишончли бўлмасин, бари-бир ундан амалда фойдаланиш жараёнида баъзи ечилиши керак бўлган муҳим ҳавфсизликни таъминлаш масалалари келиб чиқиши мумкин. Масалан, калитларни тизим фойдаланувчиларига тарқатиш масаласи. Бу масалани ечиш учун, ишлаб чиқилган бардошли калитларни тизим фойдаланувчиларига етказиш хавфсизлиги кафолатли таъминланган бўлиши талаб этилади. Бунинг учун эса яна бирор криптотизимдан фойдаланишга тўғри келади. Бу масаланинг ечими классик ва замонавий алгебрада олинган илмий натижалар асосида яратилган *очиқ калитли криптотизим*ларнинг вужудга келиши билан ҳал этилди.

Очиқ калитли криптотизим моҳияти ҳар бир фойдаланувчи учун бирини билган ҳолда иккинчисини топиш, ечилиши мураккаб бўлган масала билан боғлиқ калитлар жуфтлигини яратишдан иборат. Бу жуфтликни ташкил этувчи калитлардан бири очиқ, иккинчиси махфий деб эълон қилинади. Очиқ калит ошкора эълон қилинади, махфий калит фақат унинг эгасигагина маълум бўлади. Бирор фойдаланувчининг очиқ калитини билган ҳолда унинг махфий калитини топишнинг амалий жиҳатдан мумкин эмаслиги, ечилиши мураккаб бўлган масаланинг ҳал этилишини талаб қилиши билан кафолатланади. Очиқ маълумот, шу маълумотни олиши керак бўлган фойдаланувчининг очиқ калити билан шифрланиб унга узатилади. Шифрланган маълумотни олган фойдаланувчи фақат ўзигагина маълум бўлган махфий калит билан уни дешифрлаб, очиқ маъумотга эга бўлади.

Таъкидлаш лозимки, очиқ калитли криптотизимлар алгоритмларидан қуйидаги мақсадларда фойдаланилади:

1. Сақланадиган ва узатиладиган маълумотларнинг махфийлиги муҳофазасини таъминловчи мустақил восита сифатида.

2. Калитлар тақсимотининг муҳофазасини таъминловчи восита сифатида. Очиқ калитли криптотизимлар алгоритмлари анъанавий

криптотизимлар алгоритмларига нисбатан мураккаб ҳисоблаш жараёнларини талаб этиши натижасида паст тезликка эга бўлиб, ундан кўпроқ калитларни тақсимлашда фойдаланилади. Сўнгра, катта ҳажмдаги маълумотларни узатишда соддароқ ҳисоблашларга асосланган юқори тезликка эга бўлган тизимлардан фойдаланилади.

3. Аутентификация, яъни маълумотлар ва уларнинг муаллифлари ҳақиқийлигини аниқлаш услублари воситаси сифатида. Бу ҳақида «Электрон рақамли имзо» бўлимида батафсил тўхталади.

Очиқ калитли криптотизимлар *бир томонлама* деб аталувчи акслантиришларга (функцияларга) асосланади.

§6.2. Бир томонлама функциялар

К.Э. Шеннонниng 1949 йилдаги мақоласи [5] криптология соҳасидаги очиқ илмий изланишларнинг кўпайишига олиб келмади. Биринчидан, бу мақолада келтирилган «махфий алоқа тизимларининг назарий бардошлилик назарияси» моҳияти жиҳатидан атрофлича ва тўлиқ бўлиб, бундай назарияга кўра, махфий алоқа канали бўйлаб узатиладиган калитнинг ҳажми, узатиладиган маълумот ҳажмининг катталашиб бориши билан, катталашиб боради. Иккинчидан, амалий бардошлилик масалаларининг ечимлари ҳақидаги илмий натижалар, янги криптотизимлар яратиш йўналишларининг вужудга келишидан кўра, кўпроқ мавжуд криптотизимларнинг такомиллашувига олиб келди. Шундай бўлсада, Шеннониниг бу назариясидаги «етарли даражадаги (амалий) бардошли криптотизимлар яратиш масаласи, моҳияти жиҳатидан, маълум шартларни қаноатлантирувчи ва ечими сарф-ҳаражатларни қопламайдиган мураккаб масалага асосланиши керак», деб ифодаланган изоҳи, станфордлик олимлар У. Диффи ва М.Е. Хеллман илмий изланишларининг самарали натижаларида ўз аксини топди. Улар томонидан 1976 йилда «Криптологияда янги йўналиш» [6], деб номланган илмий мақоланинг чоп этилиши, махфий алоқа тизимларида махфий калитни махфий алоқа канали бўйлаб узатишга ҳожат йўқ бўлган амалий бардошли криптотизимлар яратишнинг асосини очиб берди. У. Диффи ва М.Е. Хеллманнинг илгари сурган ғоялари «бир томонлама функция»нинг Р.М. Нидхэмнинг ҳисоблаш тизимларига киришнинг муҳофазаси ҳақидаги ишларидан олинган таърифини криптотизимлар учун мослаштирилган ва такомиллаштирилган ифодасидир.

Бир томонлама функция – бу, таъриф бўйича, шундай $y=f(x)$ функцияки, унинг аниқланиш соҳасидан бўлган ихтиёрий x учун $f(x)=y$ қиймат осон ҳисобланиб, қийматлар соҳасининг барча y қийматларига мос келувчи x қийматларни ҳисоблаб топишни амалий жиҳатдан имконияти йўқ. Кўриниб трубдики, бир томонлама функциянинг бундай таърифи «осон ҳисобланадиган», «барча қийматлар учун», «амалий жиҳатдан», «ҳисоблашнинг имконияти йўқ» иборалар асосида берилиб, математика нуқтаи назаридан аниқ эмас. Шундай бўлсада, бу таъриф амалий криптотизим масалалари нуқтаи назаридан етарли даражада аниқ бўлиб, алоҳида олинган криптотизимлар учун такомиллаштирилиб, мутлақо аниқ ифодланиши мумкин. Шундай функциялардан криптографияда қандай фойдаланилиши ҳақида қисқача тўхталамиз. Яширин ёки махфий услубли бир томонлама функция, таъриф бўйича бирор $z \in Z$ параметрларга боғлиқ бўлиб, тескарисига эга бўлган шундай f_z функциялар синфики, берилган z параметрда аниқланиш соҳасидаги барча $x \in X$ аргументлар учун $f_z(x)=y$ қийматларни осон ҳисоблаш алгоритми E_z мавжуд бўлиб, қийматлар соҳасидаги барча $y \in Y$ қийматлар учун $f_z^{-1}(x)=y$ қийматлар маълум бўлган алгоритм билан ҳисоблашнинг имконияти йўқ (ёки бошқача айтганда $f_z^{-1}(y)=x$ қийматларни ҳисоблаш сарф-ҳаражатлари ва вақти мақсадга мувофиқ эмас). Бундай таъриф математика нуқтаи назаридан аниқ бўлмасада, амалий криптология масалаларида самарали қўлланилиши мумкинлигига шак-шубҳа йўқ.

§ 6.3. Очиқ калитли шифрлаш алгоритмларининг асослари ва уларга қўйиладиган талаблар

Очиқ калитли криптотизимлар алгоритмлари уларнинг асосини ташкил этувчи бир томонлама функциялар билан фарқланади. Аммо ҳар қандай бир томонлама функция ҳам очиқ калитли криптотизимлар яратиш учун ва улардан амалдаги аҳборотлар тизимида махфий алоқа хизматини ўрнатиш алгоритмини қуриш учун қулайлик туғдирмайди.

Бир томонлама функцияларнинг аниқланиш таърифида назарий жиҳатдан тескариси мавжуд бўлмаган функциялар эмас балки, берилган функцияга тескари бўлган функциянинг қийматларини ҳисоблаш амалий жиҳатдан мақсадга мувофиқ бўлмаган функциялар тушунилиши таъкидланган эди. Шунинг учун, маълумотнинг ишончли муҳофазасини таъминловчи очиқ калитли криптотизимларга қуйидаги муҳим талаблар қўйилади:

1. Дастлабки очиқ маълумотни шифрмаълумот кўринишига ўтказиш бир томонлама жараён ва шифрлаш калити билан шифрмаълумотни очиш – дешифрлаш мумкин эмас, яъни шифрлаш калитини билиш шифрмаълумотни дешифрлаш учун етарли эмас.

2. Очиқ калитнинг маълумлигига асосланиб, махфий калитни замонавий фан ва техника ютуқлари ёрдамида аниқлаш учун бўладиган сарф-ҳаражатлар ҳамда вақт мақсадга мувофиқ эмас. Бунда, шифрни очиш учун бажарилиши керак бўладиган энг кам миқдордаги амаллар сонини аниқлаш муҳимдир.

Очиқ калитли шифрлаш алгоритмларидан ахборот тизимида маълумотларнинг махфийлигини таъминлашда замонавий илғор услуб сифатида фойдаланиб келинмоқда. Очиқ калитли криптотизимларни яратишнинг RSA алгоритми жаҳон стандарти сифатида қабул қилинган. Умуман олганда, замонавий очиқ калитли криптотизимлар қуйидаги турдаги масалаларни ечишнинг кўп вақт талаб қилиши ва ҳисоб-китоблар учун ҳисоблаш қурилмаларида катта ҳажмдаги хотира талаб этилиши билан боғлиқ бўлган мураккабликларга таянади:

1. Етарли катта сонларни туб кўпайтувчиларга ёйиш.

2. Характеристикаси етарли катта бўлган чекли майдонларда дискрет логарифмларни ҳисоблаш.

3. Етарли катта тартибдаги алгебраик тенгламалар тизимининг илдизларини чекли майдонларда ҳисоблаш.

4. Эллиптик эгри чизиқларда рационал координатали нуқталарни топиш, уларни қўшиш ҳамда тартибини аниқлаш каби.

Ўзбекистон алоқа ва ахборотлаштириш агентлигининг Фан-техника ва маркетинг тадқиқотлари марказида криптология йўналишида олиб борилаётган илмий изланишлар параметрли алгебра асосидаги криптотизимлар яратиш усуллари ва алгоритмлари билан боғлиқ бўлиб, бу соҳада олинган натижалар диққатга сазовордир [42]. Бу илмий изланишлар натижаларида номаълум параметр бўйича киритилган амал асосида мавжуд асимметрик алгоритмларнинг туб моҳиятини сақлаган ҳолда уларни такомиллаштириш имкониятлари ёритилган. ЭРИ, ХФ ва шифрлаш алгоритмлари акслантиришлари асосларида криптографик жиҳатдан самара берувчи восита сифатида қўлланилиш хусусиятлари ҳамда ҳоссалари тасдиқлар ва мисоллар билан исботланган. Симметрик криптоалгоритмларда номаълум параметр бўйича киритилган амал асосида калит билан боғлиқ ҳолда самарали такомиллаштириш имкониятларини беради.

Қуйида нисбатан оммавийлашган очиқ калитли криптотизимлар қисқача кўриб ўтилади.

§ 6.4. Очиқ калитли RSA криптоалгоритми

У. Диффи ва М.Е. Хеллман махфий услубли бир томонлама функциянинг аниқланишига асосланиб, махфий алоқа тизими фойдаланувчилари учун, очиқ калитли криптотизимлар тузилишини (структурасини) таклиф этдилар. Ҳар бир i – фойдаланувчи бирор Z_i бутун сонни (даража кўрсаткичини) танлайди ва уни махфий сақлайди. Сўнгра, бу Z_i асосида E_{Z_i} алгоритм тузиб очиқ маълумотлар китобига бу алгоритмни жойлаштиради. Бундан ташқари Z_i асосида махфий сақланадиган D_{Z_i} алгоритмни ҳам тузади ва уни сир тутади. Агарда j – фойдаланувчи i – фойдаланувчига X махфий маълумотни узатмоқчи бўлса, у ҳолда j – фойдаланувчи очиқ маълумотлар китобидан E_{Z_i} алгоритмни олиб, $Y=f_{Z_i}(x)$, $x \in X$ услуб билан шифрмаълумотни тузиб (ҳосил қилиб), i – фойдаланувчига жўнатади. Махфий маълумотни шифрмаълумот кўринишида қабул қилиб олган i – фойдаланувчи ўзининг махфий D_{Z_i} алгоритмидан фойдаланиб $f_{Z_i}^{-1}(Y)=X$ услуб билан очиқ маълумотни ҳосил қилади. Агарда f_z, ҳақиқатан ҳам махфий услубли бир томонлама функция бўлса, у ҳолда бу функция асосида қурилган алгоритм амалий бардошлиликни таъминлайди. У. Диффи ва М.Е. Хеллман, агарда бир томонлама f_z функциянинг аниқланиш соҳасидаги даража кўрсаткичининг барча $z \in Z$ қийматлари тўплами билан, айнан шу f_z функциянинг қийматлари тўплами устма-уст тушса, яъни функциянинг аниқланиш соҳаси билан қийматлар соҳаси бир хил тўпламни ташкил этса, бундай бир томонлама функция асосида рақамли имзо олиш мумкинлигини таъкидлаганлар. Агарда i – фойдаланувчи алоқа тизими бўйича махфий бўлмаган X маълумотни барча фойдаланувчиларга етказиб, бу махфий бўлмаган маълумотни жўнатувчини маълумотни қабул қилиб олувчилар томонидан беҳато аниқланиши учун, ўз махфий калити билан алгоритм $Y=f_{Z_i}^{-1}(X)$ асосида рақамли имзо қўяди. Ҳар бир фойдаланувчи очиқ калит билан алгоритм E_{Z_i} ни билган ҳолда $f_{Z_i}(Y)=X$ ни олади, лекин фойдаланувчидан бошқа i – фойдаланувчи X маълумотни $Y=f_{Z_i}^{-1}(X)$ криптограмма кўринишидаги рақамли имзо ифодасига ўтказа олмайди, чунки фақат i – фойдаланувчининг ўзигина очиқ алгоритм асосланган f_{Z_i} функцияга тескари бўлиб, махфий алгоритм асосини ташкил этувчи $f_{Z_i}^{-1}$ ни ҳисоблай олади. Ўз-ўзидан тушинарлики, i – фойдаланувчи j – фойдаланувчига махфий маълумотни ҳам рақамли имзо билан жўнатиши мумкин. Бунинг учун, i – фойдаланувчи j – фойдаланувчининг f_{Z_j} функцияга асосланган очиқ алгоритми (очиқ шифрлаш калити) E_{Z_j} дан фойдаланиб, жўнатилиши керак бўлган маълумотни шифрлайди. Бу шифрланган маълумотни қабул қилиб ол-

ган j – фойдаланувчи ўзининг $f_{Z_i}^{-1}$ функцияга асосланган махфий D_{Z_i} дешифрлаш алгоритми билан очади.

1976 йилда У. Диффи ва М.Е. Хеллман ўзларининг «Криптологияда янги йўналиш» [6] деб номланган илмий ишларида бир томонлама функция сифатида $y=g^a \bmod n$ ифода билан аниқланган дискрет даражага кўтариш функциясини таклиф қилиб, $a=\log_g y \bmod n$ ифодадаги дискрет логарифмни ҳисоблашнинг амалий жиҳатдан мураккаблигига асосланган эдилар. 1978 йилда эса, Массачусетс технология институтининг олимлари: Р.Л. Ривест, А. Шамир, Л. Адлман, ўзларининг илмий мақолаларида биринчи бўлиб махфий услубли (йўлли) ва ҳақиқатан ҳам бир томонлама бўлган функцияни таклиф этдилар. Бу мақола «Рақамли имзоларни қуриш услублари ва очиқ калитли криптотизимлар» деб аталиб, кўпроқ аутентификация масалаларига эътибор қаратилган. Ҳозирги кунда, юқорида номлари келтирилган олимлар таклиф этган функцияни, уларнинг шарафига RSA бир томонлама функцияси дейилади. Бу функция мураккаб бўлмай, унинг аниқланиши учун, элементар сонлар назариясидан баъзи маълумотлар керак бўлади.

Мусбат бутун бўлган i ва n сонларининг энг катта умумий бўлувчисини ЭКУБ (i, n) деб белгилаймиз. Мисол учун: ЭКУБ (12, 18)=6, ЭКУБ (9, 27)=9. Ҳар қандай мусбат бутун сон n учун Эйлер функцияси $\varphi(n)$ n дан катта бўлмаган ЭКУБ (i, n)=1 шартни қаноатлантирувчи барча i сонларининг саноғини билдиради. Мисол учун:

$\varphi(2)=1$, $\varphi(3)=2$, $\varphi(4)=2$, $\varphi(5)=4$, $\varphi(6)=2$ ва ҳоказо. Ихтиёрий туб сон p учун $\varphi(p)=p-1$, ҳамда $\varphi(1)=1$ деб қабул қилинган. Бундан ташқари, ихтиёрий p ва q туб сонлари учун ушбу $\varphi(pq)=(p-1)(q-1)$ ифода ўринли бўлади. Мисол учун: $\varphi(6)=\varphi(2x3)=1\cdot 2=2$.

Буюк математик олим Эйлер (1707–1783) теоремасига кўра ихтиёрий мусбат бутун x ва n ($0<x<n$) сонлари учун ЭКУБ (x, n) = 1 шартини қаноатлантирувчи $x\varphi(n)=1(\bmod n)$ тенглик бажарилади. Мисол учун:

$$\text{ЭКУБ } (5,\ 6)=1 \text{ ва } 5^2=1(\bmod n).$$

Сонлар назарияси курсидан маълумки, агарда, e ва m бутун сонлар $0<e<m$ ва ЭКУБ (m, e)=1 шартларни қаноатлантирса, у ҳолда

$$de=1 \bmod n$$

$0<d<m$ тенгсизликни ва тенгликни қаноатлантирувчи ягона d бутун сон мавжуд бўлиб, ЭКУБ (m, e) ни топишнинг «кенгайтирилган» Евклид алгоритмидан фойдаланиб d ни топиш мумкин.

Юқорида келтирилган маълумотлардан фойдаланиб махфий услубли RSA бир томонлама функциянинг аниқланишини кўриб чиқамиз. Бу функция бирор n сони модули бўйича дискрет даражага кўтариш функцияси, яъни

$$fz(x)=x^e \bmod n$$

кўринишида аниқланади. Бу ерда: x – мусбат бутун сон бўлиб, $n=pq$ сондан катта эмас; $n=pq$, яъни p ва q туб сонлари учун бутун e сони $\varphi(n)$ дан кичик ва ЭКУБ $(e, \varphi(n))=1$. Шифрлашнинг E_z очиқ алгоритми асосини ташкил этувчи $y=f_z(x)=x^e \pmod n$ функция қийматларини ҳисоблашни осонгина квадратга кўтариш ва кўпайтириш амалларига келтириш мумкин. E_z алгоритмни очиқ калитлар базасига (китобига) киритиш, n ва e сонларини фойдаланувчилар учун очиқ эълон қилиш демакдир ва бунда n сонининг кўпайтувчилари бўлган p ва q туб сонлари махфий тутилади. Тескари функция қуйидаги

$$f_z^{-1}(y)=y^d \bmod n$$

кўринишда бўлиб, бу ерда d сони n сонидан кичик ва ушбу

$$de=1 \bmod \varphi(n)$$

тенгликни қаноатлантиради.

p, q, d –сонларидан иборат $\{p, q, d\}=z$ параметрлар тўплами $f_z(x)=x^e \bmod n$ тенглик билан аниқланган бир томонлама функциянинг криптографик махфийлик услуби хоссасининг асосини ташкил этади. Махфий D_z дешифрлаш алгоритмининг асосини ташкил этувчи тескари f_z^{-1} функциянинг қийматларини ҳисоблаш ҳам квадратга кўтариш ва кўпайтириш амаллари орқали амалга оширилади ва бунда даража кўрсаткичи бўлган d сони ЭКУБ $(e, \varphi(n))$ни ҳисоблашнинг Евклид алгоритми бўйича аниқланади. Юқорида $f_z^{-1}(y)=y^d \bmod n$ ифода билан аниқланган функциянинг $f_z(x)=x^e \bmod n$ ифода билан аниқланган функцияга ҳақиқатан ҳам тескари функция эканлиги қуйидагича кўрсатилади. Бутун сонлар арифметикасидан маълумки, бирор бутун Q сонида

$$de=1 \bmod n=\varphi(n)\cdot Q+1$$

тенглик ўринли бўлади. Юқоридаги тенгликларга ва Эйлер теоремасига кўра

$$f_z^{-1}(y)=y^d \bmod n=(x^e)d \bmod n=$$
$$=x^{\varphi(n)Q+1} \bmod n=(x^{\varphi(n)Q+1}) \bmod n=x \bmod n$$

тенгликка эга бўлинади. Демак, $de=1 \bmod n=\varphi(n)\cdot Q+1$ тенгликни қаноатлантирувчи d ва e сонлари учун: бирор $x<n$ сонларнинг n модуль бўйича d даражага кўтариш амали, шу x сонларни худди шу n модуль бўйича e даражага кўтариш амалига тескари экан. Энди нима учун Р.В. Ривест, А.Шамир ва Л. Адлман юқорида келтирилган ифода билан аниқланган $f_z(x)$ функцияни n ва e сонларини билган

ҳолда, унга тескари $f_z^{-1}(y)$ функцияни ҳисоблаш мумкин эмаслиги таъкидлаганлигини кўриб чиқамиз. Бундан ташқари p ва q туб сонлари қандай қилиб танланганда, рақиб томоннинг бу сонларни била олмаслигини ҳам кўриб чиқамиз.

Рақиб томонга n ва e сонлари маълум бўлсин. Агарда рақиб томон n сонини p ва q туб сонларининг кўпайтмасидан иборат, яъни $n=pq$ кўринишида ифодалай олса, у ҳолда махфийлик параметри $z=\{p, q, d\}$ ни тўла аниқлаган ҳолда, маълумотлар криптограммасини, маълумотни ҳақиқатан ҳам олиши керак бўлган фойдаланувчи каби, қийинчиликсиз дешифрлаш имкониятига эга бўлади. Шунинг учун RSA криптотизимининг бардошлилик даражаси n сонини p ва q туб сонларининг кўпайтмасига ёйишнинг қийинлик даражасига эквивалентдир, яъни тенг кучлидир. Агарда p ва q сонларининг узунлиги 300 дан ортиқ ўнли рақамдан иборат бўлса, ҳозирги замонавий ҳисоблаш техникаларидан фойдаланилганда, n сонини туб кўпайтувчиларга ажратиш учун сарфланадиган вақт етарли даражада кўп бўлиб, бундай туб кўпайтувчиларга ажратиш билан шуғулланишининг амалий жиҳатдан мақсадга мувофиқ эмаслиги келиб чиқади.

Юқоридаги мулоҳазалардан табиий равишда, «етарли даражада катта p ва q туб сонларини қандай аниқлаш мумкин?» –деган савол туғилади. Бундай саволга жавоб топиш учун Чебешев теоремасига мурожаат қиламиз: бирор бутун m сонидан кичик бўлган барча бутун сонлар тўпламидан танлаб олинган бирор сонни, туб сон бўлиш эҳтимоллиги $(\ln m)^{-1}$ қийматга яқин.

Мисол учун 10^{300} дан кичик бўлган барча мусбат бутун сонлар тўпламидан танлаб олинган бирор сонни туб сонга бўлиш эҳтимоллиги $(\ln 10^{300})^{-1} = -\frac{1}{300\ln 10}$ қийматга эга. Агарда бу танлаб олиш фақат 10^{300} дан кичик бўлган барча бутун мусбат тоқ сонлар тўпламида амалга оширилаётган бўлса, бу эҳтимоллик қиймати икки баробар кўпаяди. Тоқ сонлардан туб сонларни фарқлаш Ферма теоремасига асосланади: бирор p туб сонидан катта бўлмаган бутун мусбат сон учун

$$b^{p-1} = 1 \bmod p$$

тенглик ўринлидир.

Мисол учун, $2^4 = 1 \bmod 5$ ёки $3^4 = 1 \bmod 5$. Агарда r сонининг туб ёки туб эмаслигини текширмоқчи бўлсак, r сонидан кичик бўлган бутун мусбат b сонини олиб

$$b^{r-1} = 1 \bmod r$$

тенглик бажарилишини текшириш кифоя:

– тенглик бажарилса r туб сон бўлиши мумкин, чунки бу муносабат r туб бўлишини зарурий шарти;

– тенглик бажарилмаса r туб сон эмас.

Шундай қилиб, агарда $b^{r-1}=1 \bmod r$ муносабат ўринли бўлмаса қатъий ҳолда r сони туб эмас, деб айта оламиз. Аммо, $b^{r-1}=1 \bmod r$ муносабат ўринли бўлса, фақат, r сони туб бўлиши мумкин, лекин қатъий ҳолда r туб сон, деб тасдиқлай олмаймиз.

Шунинг учун, r сони етарли даражада катта бўлиб, тасодифий олинган мумкин қадар кўп бутун мусбат $b(1 \le b < r)$сонлари учун $b^{r-1}=1 \bmod r$ муносабат бажарилса r сонининг туб эканлигига шунчалик кўп даражада ишонч ҳосил қилиш мумкин. Агарда b нинг уч юзта қийматида бу муносабат ўринли бўлса, у ҳолда r сонининг туб бўлмаслиги ҳодисасининг эҳтимоли қиймати $\frac{1}{2^{300}}=2^{-300}$ га тенг бўлади.

Юқорида келтирилган алгоритмдан бугунги кунда ҳам бирор r сонининг тублигини аниқлашда фойдаланиб келинмокда.

Ҳар қандай очиқ калитли криптотизимнинг бардошлилиги очиқ маълумотга ёки унинг бирор қисмига мос келувчи шифрмаълумот маълум бўлганда, ҳамда шифрлаш алгоритми E_z маълум бўлганда, тўла шифрмаълумот дешифрлаш имконияти қанчалик мураккаблиги билан баҳоланади.

Шундай қилиб, очиқ калитли RSA алгоритми тизимидан фойдаланувчиларга калитлар генерация қилиш қуйидаги теоремаларга асосланган.

6.1-теорема. Агар $n=pq$, $p \neq q$ – туб сонлар, ва $(x, p)=1$, $(x, q)=1$ бўлса, у ҳолда

$$x^{\varphi(n)}=1 \bmod n.$$

Исботи. Агар $(x, p)=1$, $(x, q)=1$ муносабатлар ўринли бўлса, у ҳолда

$$x^{p-1}=1 \bmod p$$
$$x^{q-1}=1 \bmod q,$$

бўлиб, $y=x^{\varphi(n)}=x^{(p-1)(q-1)}$ модуль p бўйича ҳам модуль q бўйича ҳам 1 га тенг бўлади. Ҳақиқатан ҳам:

$$y=x^{\varphi(n)} \bmod p=x^{(p-1)(q-1)} \bmod p=[x^{(p-1)} \bmod n]^{(q-1)} \bmod n=1^{(q-1)} \bmod n=1$$

ёки

$$y=x^{\varphi(n)} \bmod p=x^{(p-1)(q-1)} \bmod p=[x^{(q-1)} \bmod n]^{(p-1)} \bmod n=1^{(p-1)} \bmod n=1$$

Бундан эса, $(y-1)$ нинг p ва q сонларига қолдиқсиз бўлиниши келиб чиқади, хамда $y=1 \bmod pq$ тенглик ўринли бўлади.

6.2-теорема. Агар $n=pq$, $p \neq q$ – туб сонлар, ва $(e, \varphi(n))=1$ бўлса, у ҳолда ушбу

$$E_{e,n}: x \to x^e \bmod n$$

акслантириш $Z_n = \{0; 1; 2; ...; n-1\}$ – чекли майдонда ўзаро бир қийматли акслантириш бўлади.

Исботи. Агар $(e, \varphi(n)) = 1$ бўлса, у ҳолда шудай d – ҳақиқий сони мавжуд бўладики, унинг учун

$$ed = 1 \bmod \varphi(n),$$

муносабат ўринли бўлади. Бундан эса ушбу муносабат

$$(x^e)^d = x^{ed} = x^{1+K\varphi(n)} = x(\bmod n)$$

$(x, n) = 1$ ифодани қаноатлантирувчи барча x лар учун бажарилади.

Агар $x = py$ бўлса, бу ерда $(y, q) = 1$, у ҳолда

$$p | x^{1+K\varphi(n)} - x.$$

Бу ерда: x сони q га қолдиқсиз бўлинмаганлигидан

$$x^{1+K\varphi(n)} - x = x[(x^{q-1})^{K(p-1)} - 1]$$

келиб чиқади.

Ферманинг кичик теоремасига кўра $x^{q-1} = 1 \bmod q$ ва натижада, квадрат қавс ичидаги ифода модуль p бўйича ҳам ва модуль q бўйича ҳам 0 га тенг бўлиб, бундан ушбу

$$x^{1+K\varphi(n)} - x = 0 \bmod n$$

тенгликнинг ўринлилиги келиб чиқади.

Худди шу каби, агар $x = qy$ бўлса, бу ерда: $(y, p) = 1$, у ҳолда

$$q | x^{1+K\varphi(n)} - x.$$

Бу ерда: x сони q га қолдиқсиз бўлинмаганлигидан

$$x^{1+K\varphi(n)} - x = x[(x^{p-1})^{K(q-1)} - 1]$$

келиб чиқади.

Ферманинг кичик теоремасига кўра $x^{p-1} = 1 \bmod p$ ва натижада, квадрат қавс ичидаги ифода модул p бўйича ҳам ва модул q бўйича ҳам 0 га тенг бўлиб, бундан ушбу

$$x^{1+K\varphi(n)} - x = 0 \bmod n$$

тенгликнинг ўринлилиги келиб чиқади.

Келтирилган теоремалардан фойдаланиб, тизимнинг ҳар бир i – фойдаланувчиси учун (e_i, d_i) – калитлар жуфтлиги яратилади (генерация қилинади). Етарли катта бўлган p ва q – туб сонлари олиниб (бу сонлар махфий тутилади), $n = pq$ – сони ва Эйлер функциясининг

қиймати $\varphi(n)=(p-1)(q-1)$ ҳисобланади (бу сон ҳам махфий тутилади). Сўнгра, $(e_i, \varphi(n))=1$ шартни қаноатлантирувчи, яъни $\varphi(n)$ – сони билан ўзаро туб бўлган e_i – сон бўйича d_i – сони ушбу $e_i d_i = 1 \bmod \varphi(n)$ формула орқали ҳисобланади. Бу $(e_i; d_i)$ жуфтликда e_i – очиқ калит ва d_i – махфий калит деб эълон қилинади.

Шундан сўнг i – фойдаланувчидан j – фойдаланувчига шифрланган маълумотни жўнатиш қуйидагича амалга оширилади:

1. **Шифрлаш қоидаси**: ушбу ифода $M^{e_j} \bmod n = C$ ҳисобланади, бу ерда M – очиқ маълумот, C – шифрланган маълумот;

2. **Дешифрлаш қоидаси**: ушбу ифода $C^{d_j} \bmod n = M^{e_j d_j} \bmod n = M$ ҳисобланиб, очиқ маълумот M хосил қилинади.

Кўриб ўтилганидек, RSA очиқ калитли шифрлаш алгоритми берилган етарли катта тоқ сонни туб кўпайтувчиларга ажратишнинг рационал усули мавжуд эмаслигига асосланган. Кейинги параграфда дискрет логарифмлаш масаласини характеристикаси етарли катта бўлган чекли майдонда ҳисоблашнинг мураккаблигига асосланган Эл-Гамал алгоритми ўрганилади.

§ 6.5. Эл – Гамал криптоалгоритми

Эл–Гамал алгоритми RSA алгоритмига муқобил (алтернатив) бўлиб, бу криптоалгоритмларнинг калитларини ўлчов узунликлари тенг бўлганда бир хил криптобардошлиликка эга бўладилар.

Эл–Гамал криптоалгоритми Диффи-Хеллман алгоритмига ўхшаш бўлиб, дискрет логарифмларни ҳисоблаш масаласи ечимининг мураккаблигига асосланган. Бу криптоалгоритм асосини туб бўлган p ва бутун бўлган a сонлари ташкил этади. Қуйида ушбу алгоритмнинг моҳиятини очиб берувчи мисолни келтирамиз.

Бирор фойдаланувчи (А) махфий калит x сонини танлаб олади ва $y=a^x \bmod p$ бўлган очиқ калитни ҳисоблайди. Агарда мана шу фойдаланувчи (А) билан бирор бошқа фойдаланувчи (Б) махфий маълумот алмашинувини амалга оширмоқчи бўлса, у ҳолда (Б) p сонидан кичик бўлган бирор криптотизим сони k ни танлаб олиб

$$y_1 = a^k \bmod p \text{ ва } y_2 = (m/y^k) \bmod p,$$

сонларини ҳисоблайди. Сўнгра (Б) $(y_1; y_2)$ маълумотларини (А)га жўнатади. Ўз навбатида (А) бу шифрланган маълумотни қабул қилиб, қуйидагича

$$(y_1^x \cdot y_2) \bmod p = m$$

ҳисоблаш билан очиқ маълумотни тиклайди.

Эл-Гамал криптоалгоритмига асосланган криптотизимнинг ҳар бир i – фойдаланувчиси учун (y_i, x_i) – калитлар жуфтлиги қуйидагича яратилади: бирор p_i – туб сони ва $g_i<p_i$ – тенгсизликни қаноатлантирувчи g_i (фойдаланувчилар гуруҳи учун умумий p ва $g<p$ тенгсизликни қаноатлантирувчи g) сонлари танланади. Ушбу $x_i<p_i$ тенгсизликни қаноатлантирувчи махфий бўлган x_i – сони бўйича очиқ деб эълон қилинадиган y_i – сони ушбу формула $y_i=g_i^{x_i} \bmod p_i$ (фойдаланучилар гуруҳи учун $x_i<p$ ҳамда $y_i=g^{x_i} \bmod p$) орқали ҳисобланади. Шундай қилиб, (p_i, g_i, y_i) – учлик (фойдаланувчилар гуруҳи учун p ва g умумий бўлиб, (p, g, y_i)) – учлик) очиқ калит, x_i – эса махфий калит деб олинади.

Шундан сўнг i – фойдаланувчидан j – фойдаланувчига шифрланган маълумотни жўнатиш қуйидагича амалга оширилади:

1. **Шифрлаш қоидаси**: ушбу ифода, $a_j=g_j^k \bmod p_j$, $b_j=y_j^k M \bmod p_j$ (фойдаланувчилар гуруҳи учун p ва g умумий бўлганда: $a=g^k \bmod p$, $b=y_j^k M \bmod p$) ҳисобланади, бу ерда M – очиқ маълумот, k – маълумотни шифрлаб жўнатувчи томонидан танланган тасодифий сон бўлиб, $y(p_j-1)$ – сони билан ўзаро туб, $(a_j, b_j)=C$ (p ва g умумий бўлганда $(a, b)=C$ – шифрланган маълумот);

2. **Дешифрлаш қоидаси**: $b_j \Big/ a_j^{x_j} \bmod p_j = M$ (p ва g умумий бўлганда: $b \Big/ a^{x_j} \bmod p = M$); ҳақиқатан ҳам, $b_j \Big/ a_j^{x_j} \bmod p_j \equiv g_j^{x_i k} M \Big/ g_j^{k x_i} \bmod p_j \equiv M$ (p ва g умумий бўлганда: $b \Big/ a^{x_j} \bmod p \equiv y_j^k M \Big/ a^{x_j} \bmod p \equiv g^{x_j k} M \Big/ g^{k x_j} \bmod p = M \bmod p = M$, чунки $M<p$).

Юқорида кўриб ўтилган У. Диффи ва М.Е. Хеллиманнинг бир томонлама функцияси ҳамда RSA бир томонлама функцияси очиқ калитли криптотизимларнинг хоссаларини етарли даражада очиб беради. Бу бир томонлама функциялардан ташқари ҳам кўплаб бир томонлама функциялар криптология соҳасидаги илмий нашрётларда эълон қилинган. Уларнинг баъзилари криптотизимларга қўйилган талабларга етарли даражада жавоб бермаган. Шуни таъкидлаш жоизки, назарий жиҳатдан бир томонлама бўлган функция сифатида ихтиёрий иккита сатри ёки устуни пропорционал бўлган $A_{n\times n}$ – матрицали $A_{n\times n}$ $x_{n\times 1} \bmod 256 = y_{n\times 1}$акслантиришни мисол сифатида келтириш мумкин, бу ерда $x_{n\times 1}$ ва $y_{n\times 1}$ вектор элементлари байтлардан иборат. Бундай хоссага эга бўлган матрицанинг деатамаанти нолга тенг бўлиб, унинг тескариси мавжуд эмас. Бу матрицани бирор бошқа матрицага кўпайтмасидан

ҳосил бўлган матрицанинг деатамаанти ҳам яна нолга тенг бўлиб, унга тескари матрица топиш имконияти йўқ. Матрицали акслантиришлар кўплаб шифрлаш алгоритмларида самарали қўлланилган [2, 14, 19].

Қуйида очиқ калитли криптоалгоритмлар асосини ташкил этувчи етарли катта сонларни туб кўпайтувчиларга ёйиш, характеристикаси етарли катта бўлган чекли майдонларда дискрет логарифмларни ҳисоблаш, эллиптик эгри чизиқларда рационал координатали нуқталарни топиш, уларни қўшиш ҳамда тартибини аниқлаш масалаларини ечиш мураккабликлари билан боғлиқ ҳолда параметрли алгебра амалларидан фойдаланиб яратилган янги асимметрик алгоритмлар мисол сифатида келтирилади.

§ 6.6. Параметрли алгебра амалларидан фойдаланиб яратилган янги асимметрик алгоритмлар

Ушбу: $a \circledR b = a + b + aRb \pmod{p}$ параметрли кўпайтириш амали киритилади [42].

6.1-таъриф. b – сони a – сонига тескари дейилади, агарда $a \to b = 0$ бўлса, ҳамда a – сонига тескари бўлган сон $a^{\backslash -1}$ деб белгиланади.

Берилган сонга тескари сонни қандай топишни кўриб чиқамиз.

Агар b – сони a – сонига тескари бўлса, $c = a \circledR b = a + b + aRb \pmod{p} = 0$ муносабат бажарилиши керак. Бу муносабатдан ушбу

$c - a \equiv b + aRb \pmod{p}$ ёки $c - a \equiv b(1 + aR) \pmod{p}$ ёки $b \equiv (c - a)(1 + aR)^{-1} \pmod{p}$ таққосламага эга бўламиз. Бу ерда: $c = 0$ бўлганда b – сони a – сонига тескари бўлишини ҳисобга олсак $b \equiv (0 - a)(1 + aR)^{-1} \pmod{p} = -a(1 + aR)^{-1} \pmod{p} = a^{\backslash -1}$ бўлиши келиб чиқади.

I. Киритилган амалдан фойдаланиб, характеристикаси етарли катта бўлган чекли майдонларда дискрет логарифимлаш масаласининг мураккаблигига асосланган асимметрик шифрлаш алгоритми яратиш масаласини ечишни кўриб чиқамиз.

Ушбу сон $R_i = g^{x_i} \bmod p$ ҳисобланади, бу ерда x_i – номаълум. Сўнгра $(a_i; R_i)$ – жуфтликни очиқ калит, x_i – номаълумни эса махфий калит деб эълон қиламиз.

Криптотизимнинг j – фойдаланувчиси i – фойдаланувчига M – очиқ маълумотни шифрлаб жўнатишни қуйдагича амалга оширади:

1. Фақат j – фойдаланувчининг ўзигагина маълум бўлган бирор k – сонини тасодифий ҳолда танлаб, $R = (R_i)^k \bmod p = g^{kx_i} \bmod p$ – кийматни ҳисоблайди.

2. Шифрлашни $a_i \to M = a_i + M + a_i RM \pmod{p} = a^i + M + a_i (g^{kx_i} \bmod p)$ $M \pmod{p} = w$ кўринишда амалга ошириб, шифрмаълумот сифатида $C = (w; d = g^k \bmod p)$ – жуфтлик жўнатилади.

Шифр маълумот $C=(w;\ d=g^k \bmod p)$ни қабул қилиб олган i – фойдаланувчи дешифрлашни қуйидагича амалга оширади:

1. Фақат i – фойдаланувчининг ўзига маълум бўлган x_i – махфий калитдан фойдаланиб $d^{x_i} \bmod p = g^{kx_i} \bmod p = D$ – қиймат ҳисобланади.

2. Очиқ a_i – калитга тескари бўлган элемент $(a_i)^{-1} = -a_i(1+a_iD)^{-1} \bmod p$ ҳисобланади.

3. Ушбу $R=D$ қийматни алмаштириш амалини бажариб, дешифрлаш амалга оширилади:

$$(a_i)^{-1} \circledR w = [-a_i(1+a_iD)^{-1} \bmod p] \circledR [a_i + M + a_iRM(\bmod p)] =$$

$$= [-a_i(1+a_iR)^{-1} \bmod p] \circledR [a_i + M + a_iRM(\bmod p)] =$$

$$\equiv [-a_i(1+a_iR)^{-1} + [a_i + M(1+a_iR)] + [-a_i(1+a_iR)^{-1}]R[a_i + M(1+a_iR)](\bmod p) \equiv$$

$$\equiv -a_i + a_i + M + a_iRM - a_iRM\ (\bmod p) = M.$$

II. Киритилган амалдан фойдаланиб, танланган эллиптик эгри чизиқнинг рационал нуқталари устида амаллар бажариш масаласининг мураккаблигига асосланган асимметрик шифрлаш алгоритми яратиш масаласини ечишни кўриб чиқамиз.

Ушбу нуқта $R_i=[x_i]G$ координаталари, танлаб олинган эллиптик эгри чизиққа тегишли бўлган G – рационал координатали етарли катта тартибга эга бўлган маълум нуқта орқали ҳисобланади, бу ерда x_i – номаълум. Сўнгра $(a_i;\ R_i)$ жуфтликни очиқ калит, x_i – номаълумни эса махфий калит деб эълон қиламиз.

Криптотизимнинг j – фойдаланувчиси i – фойдаланувчига M – очиқ маълумотни шифрлаб жўнатишни қуйидагича амалга оширади:

1. Фақат j – фойдаланувчининг ўзигагина маълум бўлган бирор k – сонини тасодифий ҳолда танлаб, эллиптик эгри чизиқда $R=[k]R_i=[k][x_i]G=[kx_i]G=(x_G,\ y_G)$ – нуқта топилади ва бу нуқтанинг Ox ўқидаги x_G – координатаси (ёки Oy ўқидаги y_G – координатаси) $R=x_G$ (ёки $R=y_G$ ёки $R=f(x_G,\ y_G)$) деб қабул қилинади. Шифрлашни $a_i \circledR M = a_i + M + a_iRM(\bmod p) = a_i + M + a_ix_GM\ (\bmod p) = w$ кўринишда амалга ошириб, шифрмаълумот сифатида $C=(w;\ d=[k]G)$– жуфтлик жўнатилади. Шифр маълумот $C=(w;\ d=[k]G)$ни қабул қилиб олган i – фойдаланувчи дешифрлашни қуйидагича амалга оширади:

1. Фақат i – фойдаланувчининг ўзига маълум бўлган x_i – махфий калитдан фойдаланиб $[x_i]d=[x_i][k]G=[x_ik]G=D$ – нуқта топилади.

2. Очиқ a_i – калитга тескари бўлган элемент $(a_i)^{\backslash -1} = -a_i(1+a_iD)^{-1} \bmod p$ ҳисобланади.

3. Ушбу $R = x_D$ қийматни алмаштириш амалини бажариб, дешифрлаш амалга оширилади:

$$(a_i)^{\backslash -1} \circledR w = [-a_i(1+a_iD)^{-1} \bmod p] \circledR [a_i + M + a_iRM(\bmod p)] =$$
$$= [-a_i(1+a_iR)^{-1} \bmod p] \circledR [a_i + M + a_iRM(\bmod p)] =$$
$$\equiv [-a_i(1+a_iR)^{-1} + [a_i + M(1+a_iR)] + [-a_i(1+a_iR)^{-1}]R[a_i + M(1+a_iR)](\bmod p) \equiv$$
$$\equiv [-a_i(1+a_iR)^{-1}](1+a_iR) + [a_i + M(1+a_iR)] - a_iRM(\bmod p) \equiv$$
$$\equiv -a_i + a_i + M + a_iRM - a_iRM \ (\bmod p) = M.$$

III. Киритилган амалдан фойдаланиб, етарли катта сонни туб кўпайтувчиларга ажратиш масаласининг мураккаблигига асосланган асимметрик шифрлаш алгоритми яратиш масаласини ечишни кўриб чиқамиз.

Етарли катта ва махфий тутилиши керак бўлган p ва q – туб сонлари танлаб олиниб, $n = pq$ ҳисобланади. Ушбу $e_id_i \equiv 1 \bmod \varphi(n)$ таққосламадан (бу ерда $\varphi(n) = (p-1)(q-1)$ – махфий) e_i – параметрга бирор қиймат бериб $e_id_i \equiv 1 \bmod (p-1)(q-1)$ муносабатни қаноатлантирувчи d_i – сонини топиш мумкин. $(e_i; n)$ – жуфтликни очиқ, $(d_i; \varphi(n))$ – жуфтликни махфий деб, шифрлаш ва дешифрлаш жараёнлари қуйидагича аниқланади.

Криптотизимнинг j – фойдаланувчиси i – фойдаланувчига M – очиқ маълумотни шифрлаб жўнатишни қўйидагича амалга оширади:

1. Фақат j – фойдаланувчининг ўзигагина маълум бўлган бирор k ва r – сонларини тасодифий ҳолда танлаб, $R = r^k \bmod n$ – қийматни ҳисоблайди (бу ерда $r \neq p$ ва $r \neq q$).

2. Шифрлашни $a_i \circledR M = a_i + M + a_iRM(\bmod n) = a_i + M + a_i(r^k \bmod n)$ $M(\bmod n) = w$ кўринишда амалга ошириб, шифрмаълумот сифатида – жуфтлик жўнатилади.

Шифр маълумот $C = (w; d = R^{e_i} \bmod n)$ ни қабул қилиб олган i – фойдаланувчи дешифрлашни қуйидагича амалга оширади:

1. Фақат i – фойдаланувчининг ўзига маълум бўлган d_i – махфий калитдан фойдаланиб $d^{d_i} \bmod n = r^{ke_id_i} \bmod n = r^k \bmod n = R$ – қиймат ҳисобланади.

2. Очиқ a_i – калитга тескари бўлган элемент $(a_i)^{\backslash -1} = -a_i(1+a_iR)^{-1} \bmod n$ ҳисобланади.

Дешифрлаш амалга оширилади:

$$(a_i)^{\backslash -1} \circledR w=[-a_i(1+a_iR)^{-1} \bmod n] \circledR [a_i+M+a_iRM(\bmod n)]=$$

$$=[-a_i(1+a_iR)^{-1}\bmod n] \circledR [a_i+M+a_iRM(\bmod n)]=$$

$$\equiv[-a_i(1+a_iR)^{-1}+[a_i+M(1+a_iR)]+[-a_i(1+a_iR)^{-1}]R[a_i+M(1+a_iR)](\bmod n)\equiv$$

$$\equiv[-a_i(1+a_iR)^{-1}](1+a_iR)+[a_i+M(1+a_iR)]-a_iRM(\bmod n)\equiv$$

$$\equiv -a_i+a_i+M+a_iRM-a_iRM\ (\bmod n)=M.$$

Параметрли алгебра амаллари хусусиятлари мавжуд мураккабликларни композициялари негизида такомиллашган янги асимметрик алгоритмлар яратиш имкониятларини беради.

6-боб бўйича хулосалар

Ушбу бобда:

1. Криптографик калитларни тизим фойдаланувчиларига очиқ ахборот коммуникация тармоғида кафолатли мухофазаланган ҳолда узатиш масаласининг ечиш йўналишида олиб борилган илмий тадқиқотлар натижаси очиқ калитли шифрлаш алгоритмларининг яратилишига туртки бўлганлиги ёритилди.

2. Очиқ калитли криптографик алгоритмларнинг асосини ташкил этувчи бир томонлама функция таърифининг мантиқий моҳияти изоҳланди.

3. Асимметрик тизимлар яратишда самарали қўлланилиб келинаётган биртомонлама функция мураккабликлари туркумининг моҳиятлари баён этилди.

4. Очиқ калитли криптотизимларга қўйиладиган талаблар ва бундай алгоритмларнинг ахборот мухофазаси масалаларини ечишдаги қулайликлари таҳлил қилинди.

5. Нисбатан оммавийлашган ҳамда очиқ калитли шифрлаш алгоритмларининг хоссаларини етарли даражада очиб берувчи RSA ва Эл-Гамал алгоритмлари асослари баён қилинди.

6. Очиқ калитли криптоалгоритмлар асосини ташкил этувчи: етарли катта сонларни туб кўпайтувчиларга ёйиш, характеристикаси етарли катта бўлган чекли майдонларда дискрет логарифмларни ҳисоблаш, эллиптик эгри чизиқларда рационал координатали нуқталарни топиш, уларни қўшиш ҳамда тартибини аниқлаш масалаларини ечиш мураккабликлари билан боғлиқ ҳолда параметрли алгебра амалларидан фойдаланиб яратилган янги асимметрик алгоритмлар таклиф этилди.

VII БОБ
ХЭШ-ФУНКЦИЯ ВА УНИНГ АХБОРОТНИ МУҲОФАЗА ҚИЛИШ МАСАЛАЛАРИНИ ЕЧИШДА ҚЎЛЛАНИЛИШИ

Хэш-функция деб ихтиёрий узунликдаги (бит ёки байт бирликларида) маълумотни бирор фиксирланган (қайд қилинган) узунликдаги (бит ёки байт бирликларида) қийматга ўтказувчи функцияга айтилади. Хэш-функциялар статистик тажрибаларни ўтказишда, мантиқий қурилмаларни текширишда, тез қидириб топиш алгоритмларини тузишда ва маълумотлар базасидаги маълумотларнинг тўлалигини текширишда қўлланилади. Масалан, ҳар хил узунликдаги маълумотларнинг катта рўйхатидан керакли маълумотни тез қидириб топишда бу маълумотларни бир-бири билан таққослашдан кўра, уларнинг назорат йиғиндиси вазифасини бажарувчи хэш қийматларини солиштириш қулайроқдир [2, 29].

Криптографияда хэш-функциялар қуйидаги масалаларни ҳал қилиш учун ишлатилади:

– маълумотни узатишда ёки сақлашда унинг тўлалигини назорат қилиш учун;

– маълумот манбаини аутентификация қилиш учун.

Маълумотни узатишда ёки сақлашда унинг тўлалигини назорат қилиш учун ҳар бир маълумотнинг хэш қиймати (бу хэш қиймат маълумотни аутентификация қилиш коди ёки «имитовставка» – маълумот блоклари билан боғлиқ бўлган қўшимча киритилган белги дейилади) ҳисобланилади ва бу қиймат маълумот билан бирга сақланилади ёки узатилади. Маълумотни қабул қилган фойдаланувчи маълумотнинг хэш қийматини ҳисоблайди ва унинг назорат қиймати билан солиштиради. Агар таққослашда бу қийматлар мос келмаса, маълумот ўзгарганлигини билдиради.

«Имитовставка»лар ҳосил қилиш учун ишлатиладиган хэш-функциялар назорат йиғиндисидан фарқли равишда маълумотни сақлаш ва узатишда рўй берадиган тасодифий хатоларни топибгина қолмасдан, рақиб томонидан қилинган актив ҳужумлар тўғрисида ҳам огоҳлантиради. Рақиб хэш қийматни мустақил ўзи ҳисоблаб топа олмаслиги ва муваффақиятли ҳолда имитация қилиши ёки маълумотни ўзгартира олмаслиги учун хэш-функция рақибга маълум бўлмаган махфий калитга эга бўлиши керак. Бу калит фақатгина маълумотни

узатувчи ва қабул қилувчи томонларга маълум бўлиши керак. Бундай хэш-функцияларга *калитли хэш-функциялар* дейилади.

Калитли хэш-функциялар ёрдамида ҳосил қилинадиган «имитовставка»лар имитация (impersonation) туридаги ҳужумларда қалбаки маълумотларни ҳосил қилишга (fabrication) ва «ўзгартириш» (substitution) туридаги ҳужумларда узатиладиган маълумотни модификация (modification) қилишга йўл қўймаслик учун ишлатилади.

Маълумот манбаини аутентификация қилиш масаласи ахборот-коммуникация тизимининг бир-бирига ишонмайдиган фойдаланувчилари ўртасида маълумот алмашинувида юзага келади. Бу масалани ҳал қилишда иккала томон ҳам биладиган махфий калитни қўллаб бўлмайди. Бу ҳолатда маълумот манбаини аутентификация қилишга имкон берадиган рақамли имзо схемаси қўлланилади. Бунда одатда фойдаланувчининг махфий калитига асосланган имзо қўйишдан олдин хатолик кодини аниқловчи хэш-функция ёрдамида маълумот сиқилади. Бу ҳолда хэш-функция махфий калитга эга бўлмайди ҳамда у фиксирланган бўлиши ва ҳаммага маълум бўлиши мумкин. Унга қўйилган асосий талаб имзоланган ҳужжатни ўзгартириш ҳамда бир хил хэш қийматга эга бўлган иккита ҳар хил маълумотни танлаш имконияти йўқлигининг кафолатидир. Агар бир хил хэш қийматга эга бўлган иккита ҳар хил маълумот мавжуд бўлса, бу маълумотлар жуфти *коллизия* ҳосил қилади дейилади.

Юқорида келтирилганларни формаллаштириб, қуйидаги таъриф киритилади. X орқали элементлари маълумотлардан иборат бўлган тўпламни белгилаймиз. Одатда маълумотлар бирор алифбонинг, кўпинча иккилик саноқ тизими алифбоси символлари кетма-кетлигидан тузилган бўлади. Y фиксирланган узунликдаги иккилик саноқ ситстемасида аниқланган векторлар тўплами бўлсин.

Хэш-функция $h: X \to Y$ деб, ихтиёрий узунликдаги M маълумотни фиксирланган узунликдаги $h(M)=H$ қийматга акслантирувчи, осон ҳисобланадиган бир томонлама функцияга айтилади.

Хэш қиймат бошқа номлар билан –«хэш код», «свертка», «дайджест», «бармоқ излари» деб ҳам аталади [14].

Хэш-функцияга қуйидаги талаблар қўйилади [2]:

1. Ихтиёрий узунликдаги матнга қўллаб бўлади.
2. Чиқишда тайинланган узунликдаги қийматни беради.
3. Ихтиёрий берилган x бўйича $h(x)$ осон ҳисобланади.
4. Ихтиёрий берилган H бўйича $h(x)=H$ тенгликдан x ни ҳисоблаб топиб бўлмайди. (Бир томонламалик хоссаси)

5. Олинган x ва $y \neq x$ матнлар учун $h(x) \neq h(y)$ бўлади. (Коллизияга бардошлилик хоссаси)

Одатда мумкин бўлган маълумотларнинг сони мумкин бўлган хэш қийматлар сонидан кўп бўлади, шунинг учун ҳар бир хэш қийматга бир нечта матнлар тўплами, яъни бир хил хэш қийматли маълумотлар тўплами мос келади.

Кўпинча хэш-функциялар *бир қадамли сиқувчи функциялар* деб аталадиган, икки ўзгарувчили $y=f(x_1, x_2)$ функциялар асосида қурилади. Бу ерда x_i ва y векторлар узунликлари мос равишда m ва n бўлган векторлар бўлиб, n хэш қиймат узунлигидир. M маълумотнинг $h(M)$ қийматини олиш учун, бу маълумот узунлиги m бўлган блокларга ажратилади. Агар, маълумотнинг узунлиги m га каррали бўлмаса, охирги блок бирон-бир махсус кўринишда m гача тўлдириб олинади. Ҳосил қилинган M_1, M_2, ..., M_N блокларга хэш қийматни ҳисоблаш функцияси кетма-кет қўлланилади:

$$H_0=v,$$
$$H_i=f(M_i, H_{i-1}), \quad i=1, ..., N, \tag{7.1}$$
$$h(M)=H_N.$$

Бу ерда n фиксирланган бирор бошланғич вектор. Агар f функция калитга боғлиқ бўлса, бу векторни нол векторга тенг деб олиш мумкин. Агарда, f функция калитга боғлиқ бўлмаса, кичик маълумотларни тўла танлаш имкониятини йўқотиш учун, бу векторни вақт, маълумот рақами ва каби маълумотларни аниқловчи белгилардан тузиш мумкин. Бундай ҳолда хэш-функциянинг хоссалари бир қадамли f сиқиш функциясининг хоссалари билан тўлиқ аниқланади.

Хэш-функциялар иккита муҳим турга, *калитли* ва *калитсиз* хэш-функцияларга ажратилади. Калитли хэш-функциялар симметрик калитли тизимларда ишлатилади. Уларга маълумотни аутентификация қилиш кодлари (message authentication code (MAC)) ҳам дейилади. Улар бир-бирига ишонувчи фойдаланувчилар тизимида қўшимча воситаларсиз манбанинг ҳақиқийлигини, маълумотнинг тўлалигини кафолатлайди.

§ 7.1. Калитли хэш-функциялар ва уларнинг хоссалари

Калитли хэш-функцияларни қўллашда уларга қуйидаги асосий талаблар қўйилади:

– фабрикация имкониятининг мавжуд эмаслиги;

– модификациянинг имконияти йўқлиги.

Биринчи талаб хэш қиймат берилганда унга мос бўлган маълумотни танлашнинг мураккаб бўлишини билдиради. Иккинчи талаб маълумот ва унинг хэш қиймати берилганда, хэш қиймати шунга тенг бўладиган бошқа маълумотни танлаш мураккаб бўлишини билдиради.

Баъзан, бу иккита хоссани битта кучлироқ хоссага – *ҳисоблаш бардошлилиги* хоссасига бирлаштирилади. Бу талаб хэш қийматлари маълум бўлган берилган $\{x_1, x_2, ..., x_l\}$ маълумотлар учун хэш қийматлари шулардан бирига тенг бўладиган бошқа x, $x \neq x_i$, $i=1,l$ маълумотни танлашнинг мураккаблигини билдиради.

Мураккаб деганда, масалани реал вақт давомида замонавий ҳисоблаш курилмаларидан фойдаланиб ҳал қилиш имконияти бўлмайдиган ҳисоблаш мураккаблиги тушунилади.

Калитли хэш-функциялар бир-бирига ишонувчи томонлар ўртасида ишлатилади ва улар умумий махфий калитга эга бўладилар. Одатда бу шароитда иккинчи томон маълумотни қабул қилиб олганлигини тан олмаслик ёки уни ўзгартириш ҳолатидан ахборот-коммуникация тизимини ҳимоя қилиш талаб қилинмайди. Шунинг учун калитли хэш-функциялардан коллизияларга бардошлилик талаб қилинмайди.

Калитли хэш-функцияларга «имитация» қилиш, яъни бўш каналда қалбаки маълумотни узатиш ҳамда узатилаётган маълумотни қалбаки маълумотга алмаштириш каби ҳужумлар бўлиши мумкин.

Ҳисоблаш бардошлилиги хоссасидан хэш-функцияда қўлланилаётган калитни аниқлаш имконияти йўқлиги келиб чиқади, калитни билиш эса ихтиёрий маълумотнинг хэш қийматини ҳисоблаш имкониятини беради. Тескари тасдиқ эса ўринли эмас, чунки баъзи бир ҳолларда калитни олдиндан билмасдан туриб, хэш қийматни танлаш мумкин.

Мисол учун, кенг тарқалган, бир қадамли сиқиш функцияси ёрдамида қурилган қуйидаги кўринишдаги хэш-функцияни кўриш мумкин:

$$f_k(x, H)=E_k(x \oplus H),$$

бу ерда: E_k – блоклаб шифрлаш алгоритми.

M – маълумотнинг $h(M)$ қийматини ҳисоблаш учун маълумот кетма-кет келган m битли $M_1, M_2, ..., M_N$ – блоклар кўринишида ифодаланади. Агар маълумот узунлиги блокнинг узунлигига каррали бўлмаса, охирги блок бирор махсус шаклда тўлиқ блоккача тўлдирилади. Хэш қийматни ҳисоблаш алгоритми қуйидаги кўринишда бўлади:

$$H_0=v,$$

$$H_i = E_k(M_i \oplus H_{i-1}), \quad i = 1, \ldots, N, \qquad (7.2)$$
$$h(M) = H_N.$$

Калитли хэш-функцияларни қуришнинг яна бир усули калитсиз хэш-функциялардан фойдаланишдир. Бунда хэш қийматни ҳисоблаш учун калит берилган маълумотга қўшиб ёзиб қўйилади.

Агар калит берилган маълумотнинг бошига ёки охирига тўғридан-тўғри қўшиб қўйилса, баъзи ҳолларда маълумотни модификация қилишга имкон бериши мумкин.

Масалан, k калит маълумотнинг бошига $hk(x) = h(k, x)$ формулага асосан қўшиб қўйилган бўлсин. Агар h функция (7.1) формулага асосан бир қадамли сиқувчи функциялар ёрдамида қурилган бўлса, у ҳолда M ва $H = h(k, M)$ ларнинг маълум қийматлари бўйича бирор M' қўшиб ёзилган (M, M') кўринишдаги ихтиёрий маълумот учун бу функциянинг қийматларини ҳисоблаш мумкин. Бу хэш-функцияни ҳисоблашнинг итеративлиги билан изоҳланади, чунки $H' = h(k, M, M')$ қийматни топиш учун k калитнинг қийматини билиш шарт эмас, H қийматнинг ҳисобланган оралиқ қийматларидан фойдаланиш етарли. Шунинг учун бундай функция модификацияга бардошли эмас.

Агар калит маълумотнинг охирига $H = h_k(M) = h(M, k)$ формулага асосан қўшилган бўлса, h функция учун коллизияни, яъни $h(x_1) = h(x_2)$ бўладиган x_1, x_2, $x_1 \neq x_2$ жуфтликни билиш ихтиёрий k калит учун $h(x_1, k) = h(x_2, k)$ қийматни ҳисоблаш имконини беради. Шунинг учун $M = x_1$ маълумотни модификация қилиш мураккаблиги $O(2^n)$ катталик билан эмас, балки коллизияларни қидириш мураккаблиги билан таққосланади ва $O(2^{n/2})$ билан баҳоланади, чунки бу ҳолда *«туғилган кун» парадокси*га асосланган ҳужум ўринли бўлади.

Шуларни эътиборга олиб, калитни маълумотга бир марта эмас, бир неча марта қўядиган усуллар ишлатилади. Бунга қуйидаги иккита усулни мисол қилиб келтириш мумкин:

$$H = h(k, y, M, k),$$
$$H = h(k, y_1, h(k, y_2, M)),$$

бу ерда: y, y_1 ва лар k калитнинг n узунликдаги блокнинг каррали сигача ўлчовга тўлдирилганидир. Калитсиз хэш-функциялар учун бундай усул эффектив ҳисобланадиган ва ҳужумларга бардошли калитли хэш-функцияларни куриш имконини беради. Бундай усулнинг камчилик томони шундаки, хэш қийматнинг n узунлиги жуда катта бўлади. Одатда, тўлаликни текшириш учун хэш қиймат узунлиги 32

дан 64 битгача бўлиши, $2^{32} \le n \le 2^{64}$ бажарилиши керак, аутентификация учун эса $n \le 2^{138}$ шартнинг бажарилиши зарур.

Юқорида айтиб ўтилган блоклаб шифрлаш алгоритмига асосланган ёки калитсиз хэш-функцияни ҳисоблашга асосланган алгоритмлардан ташқари замонавий ЭҲМларда қўллаш самарадорлигини ҳисобга олиб тузилган алгоритмлар ҳам мавжуд. Бунга MAA (Message Authenticator Algorithm) калитли хэш-функция алгоритмини мисол қилиб келтиришимиз мумкин [2, 14, 20, 28].

Ўзбекистон Республикасининг хэш-функция давлат стандарти O'z DSt 1106:2006 да калитли хэш-функция келтирилган бўлиб, калит узунлиги 128 бит ёки 256 бит бўлиши мумкин. Европа Ҳамжамиятининг RACE дастури доирасида ишлаб чиқилган RIPE–MAC1 ва RIPE–MAC3 хэш-функция алгоритмлари, Nippon Telephone and Telegraph компанияси томонидан ишлаб чиқилган N–хэш хэш-функция алгоритми, шунингдек CBC–MAC ва CRC–MAC хэш-функцияларини калитли хэш-функция алгоритмларига мисол қилиб келтиришимиз мумкин [14, 31–33].

§ 7.2. Калитсиз хэш-функциялар ва уларнинг хоссалари

Калитсиз хэш-функциялар *хатоларни аниқлаш кодлари* (modification detection code (MDC) ёки manipulation detection code, message integrity code (MIC)) деб ҳам юритилади. Калитсиз хэш-функция – қўшимча воситалар (шифрлаш ёки рақамли имзо) ёрдамида маълумотнинг тўлалигини кафолатлайди. Бу хэш-функциялар бир-бирига ишонувчи ҳамда бир-бирига ишонмайдиган фойдаланувчилар тизимларида ишлатилади [2, 32, 33].

Одатда калитсиз хэш-функциялардан қуйидаги хоссаларни қаноатлантириши талаб қилинади:

1) бир томонламалик;

2) коллизияга бардошлилик;

3) хэш қийматлари тенг бўлган иккита маълумотни топишга бардошлилик.

Биринчи шарт берилган хэш қийматга эга бўлган маълумотни, иккинчи шарт бир хил хэш қийматга эга бўлган маълумотлар жуфтини, учинчи шарт хэш қиймати маълум бўлган берилган маълумот учун хэш қиймати шунга тенг бўлган иккинчи маълумотни топишнинг мураккаб эканлигини билдиради.

Масалан, назорат йиғиндини топувчи CRC хэш-функцияси чизиқли акслантириш бўлади ва шунинг учун ҳам бу учта шартдан бирортасини ҳам қаноатлантирмайди.

Калитсиз хэш-функция сифатида юқорида қаралган «имитовставка»ни ишлаб чиқиш режимидаги блоклаб шифрлаш алгоритми асосида қурилган (7.2) кўринишдаги хэш-функциядан фойдаланиш ҳам мақсадга мувофиқ эмас. Чунки, блоклаб шифрлаш алгоритмининг тескариланувчанлиги ихтиёрий хэш қиймат учун фиксирланган ва ҳаммага маълум бўлган калитда кирувчи маълумотни танлаш имконини беради.

Биринчи шартни қаноатлантирувчи хэш-функцияга мисол қуриш учун

$$g_k(x)=E_k(x)\oplus x$$

формула билан берилган функцияни қарайлик. Бу ерда E_k – блоклаб шифрлаш алгоритми, яъни криптографик функцияси. Бундай функциялар иккала аргументи бўйича ҳам бир томонлама бўлади. Шунинг учун, (7.1) қоидага асосан бир қадамли сиқувчи функцияни

$$H=f(x, H)=E_H(x)\oplus x \tag{7.3}$$

ёки

$$H=f(x, H)=E_x(H)\oplus H \tag{7.4}$$

функциялардан бири деб олиниб, унинг асосида хэш-функцияни қуриш мумкин.

Россиянинг хэш-функция стандарти ГОСТ Р 34.11-94 асосида (7.4) формула, АҚШнинг хэш-функция стандарти SHA асосида (7.3) формула ётади.

Қуйидаги тасдиқ ўринли:

7.1-тасдиқ. Агар h хэш-функция (7.1) қоидага кўра бир қадамли сиқувчи f функцияга асосан қурилган бўлса, у ҳолда f функциянинг коллизияга бардошлилигидан h функциянинг ҳам коллизияга бардошлилиги келиб чиқади.

Ҳақиқатан ҳам, агарда h функция коллизияга эга бўлса, у ҳолда бирор i – қадамда f функция ҳам коллизияга эга бўлиши керак. Бу ерда коллизияни аниқлашда $f(x_1, x_2)$ функция x_1 ва x_2 ўзгарувчиларни битта кириш векторига конкатенация қилишдан ҳосил қилинган бир ўзгарувчили функция деб қаралиши керак.

Қуйида 1) ва 2) хоссалар орасида ўзаро боғлиқлик мавжудлигини кўрсатилади:

7.2-тасдиқ. Агар хэш-функция коллизияга бардошли бўлса, у ҳолда у ўзининг хэш қийматлари тенг бўлган иккита маълумотни топишга ҳам бардошли бўлади.

Ҳақиқатан ҳам, агар берилган маълумотнинг хэш қиймати бўйича шу хэш қийматга эга бўлган бошқа маълумотни танлаш мумкин бўлса, у ҳолда ҳосил қилинган маълумотлар жуфти коллизияни ташкил қилади.

7.3-тасдиқ. Коллизияга бардошли хэш-функция бир томонлама бўлиши шарт эмас.

Бу тасдиққа мисол сифатида сиқувчи бўлмаган $f(x)=x$ функцияни келтириш мумкин. Равшанки бу функция коллизияга бардошли, лекин бир томонлама функция эмас.

Сиқувчи хэш-функцияга мисол сифатида қуйидаги шартлар билан аниқланган h функция кўрилиши мумкин:

– $h(x)=(1, x)$, *агар x нинг узунлиги n битга тенг бўлса;*

– $h(x)=(0, g(x))$, *агар x нинг узунлиги n битдан катта бўлса.*

Бу ерда $g(x)$ коллизияга бардошли бўлган, сиқувчи n битлик функция. h функция коллизияларга ҳамда хэш қийматлари тенг бўлган иккита маълумотни топишга бардошли функция, лекин у бир томонлама функция эмас.

7.4-тасдиқ. $h:X\to Y$ хэш-функция берилган бўлиб, $|X|>2|Y|$ бўлсин. Агарда h функциянинг тескарисини топишнинг самарали алгоритми мавжуд бўлса, у ҳолда h функциянинг коллизияларини муваффақиятли топишнинг эҳтимоли $\frac{1}{2}$ дан катта бўлган эҳтимолий алгоритми мавжуд бўлади.

Бир томонлама функция учун хэш қийматлари тенг бўлган иккита маълумотни танлаш ёки хэш қийматлари тенг бўлган иккита маълумотни қидириш мураккаблик даражаси $O(2^n)$ билан баҳоланади. Шу билан бирга коллизияни қидириш мураккаблик даражаси $O(2^{n/2})$ билан баҳоланади, чунки бу ҳолатда «туғилган кун» парадоксига асосланган ҳужумни қўллаш мумкин.

Қуйида блоклаб шифрлаш алгоритмлари асосида қурилган хэш-функцияларга мисоллар кўриб ўтилади.

E_k – блоклаб шифрлаш алгоритми, n – блокнинг узунлиги, l – калит узунлиги ва G узунлиги n бўлган векторга l узунликдаги векторни мос қўювчи бирор акслантириш бўлсин. E_k – блоклаб шифрлаш алгоритми асосида қурилган қуйидаги бир қадамли сиқувчи функциялар кўрилади:

а) $f(x, H)=E_x(H)\oplus H$ (Дэвис-Мейер);

б) $f(x, H)=E_{G(x)}(x)\oplus x$ (Матиас-Мейер-Осеас);

в) $f(x, H)=E_{G(x)}\oplus x\oplus H$ (Миагучи-Принель).

Бу келтирилган бир қадамли сиқувчи функциялардан фойдаланиб қурилган ихтиёрий хэш-функция қийматининг узунлиги ўлчами n бўлган

блок узунлигига тенг вектор бўлади. Агар бу узунлик етарли бўлмаса, у ҳолда бир қадамли f функцияни узунлигининг ўлчами ундан икки марта катта бўлган f' функция билан алмаштириш мумкин. Буни масалан, f функцияни икки марта қўллаш ва ундан кейин ярим блокларни аралаштириш билан қуйидаги формула асосида амалга ошириш мумкин:

$$f'(x, H_1, H_2)=\pi(f(x, H_1), f(x, H_2)),$$

бу ердаги: π функция ихтиёрий a, b, c, d – ярим блокларни $\pi((a, b), (c, d))=(a, d, c, b)$ қоида бўйича алмаштиради. Бундай усул Матиас-Мейер-Осеас схемасидан фойдаланиб, MDC-2 бир қадамли функциясини куришда қўлланилган.

Умуман олганда блоклаб шифрлаш алгоритмларидан фойдаланилиб қуриладиган калитсиз хэш-функцияларда блок узунлиги хэш қиймат узунлигига тенг бўладиган схемалар мавжуд. Қуйида ушбу типдаги алгоритмларнинг умумий схемаси келтирилган:

$$H_0=I_H,$$
$$H_i=E_A(B)\oplus C.$$

Бу ерда: I_H – бошланғич тасодифий қиймат, A, B ва C лар M_i, H_{i-1}, $(M_i\oplus H_{i-1})$ га тенг бўлиши мумкин, M_i – кирувчи блок, H_i – итерациянинг i – қадами. Ушбу алгоритмларнинг кўриниши қуйидагича [14, 30]:

1. $H_i=E_{Hi-1}(M_i\oplus H_{i-1})\oplus M_i\oplus H_{i-1}$
2. $H_i=E_{Hi-1}(M_i)\oplus H_{i-1}\oplus M_i$
3. $H_i=E_{Hi-1}(M_i\oplus E_{Hi-1})\oplus M_i$
4. $H_i=E_{Mi}(H_{i-1})\oplus H_{i-1}$
5. $H_i=E_{Mi}(M_i\oplus H_{i-1})\ M_i\oplus H_{i-1}$
6. $H_i=E_{Mi}(H_{i-1})\oplus M_i\oplus H_{i-1}$
7. $H_i=E_{Mi}(M_i\oplus H_{i-1})\oplus H_{i-1}$
8. $H_i=E_{Mi\oplus Hi-1}(M_i)\oplus M$
9. $H_i=E_{Mi\oplus Hi-1}(H_{i-1})\oplus H_{i-1}$
10. $H_i=E_{Mi\oplus Hi-1}(M_i)\oplus H_{i-1}$
11. $H_i=E_{Mi\oplus Hi-1}(H_{i-1})\oplus M_i$

Бошқа калитсиз хэш-функцияларга MD4, MD5 ва SHA хэш-функциялари мисол бўла олади. Бу алгоритмлар 32 разрядли ЭҲМларда самарали қўлланилишга мўлжалланиб, махсус лойиҳалаштирилган алгоритмлардир.

Бу алгоритмлардан фойдаланилганда берилган M маълумот узунлиги $m=512$ бит бўлган блокларга ажратилади. Охирги блок маълумот охирига блокнинг узунлиги 448 бит бўлгунча 1000....000 комбинация-

ни қўшиш билан ҳосил қилинади, ундан кейин маълумот узунлигини ифодаловчи 64 битли комбинация қўшилади. Кейин $f(x-H)=E_H(H)\oplus H$ формула билан берилган бир қадамли сиқувчи функциядан фойдаланиб, (7.1) тартиботга (процедурага) асосан хэш қиймат ҳисобланади. Бу ерда x – узунлиги $m=512$ бит бўлган маълумот блоки, $H-n$ битлик блок, E_x – блоклар тўпламидаги бирор акслантириш. Бошланғич векторнинг қиймати E_x акслантиришни аниқлашда берилади.

ГОСТ Р 34.11-94 хэш-функция стандартида $n=m=512$ қийматлар қабул қилинган. $H_i=f(x_i, H_{i-1})$ қийматларни кетма-кет ҳисоблашда фойдаланиладиган бир қадамли $f(x, H)$ сиқувчи функция ҳар бири 256 бит калитга эга бўлган ва 64 бит узунликдаги блоклар билан амаллар бажарувчи тўртта параллел ишловчи блоклаб шифрлаш схемаси (ГОСТ 28147-89) негизида қурилган. Ҳар бир калит мос равишда кирувчи x_i маълумот блоки ва H_{i-1} қийматнинг бирор чизиқли функцияси кўринишида ҳисобланади. H_i қиймат кирувчи x_i маълумот блоки ва H_{i-1} қиймат шифрланиши натижасининг чизиқли функцияси бўлади. H_N қийматни M_1, M_2, ..., M_N блоклар кетма-кетлиги учун ҳисоблагандан кейин

$$H=h(M)=f(Z\oplus M_N, f(L, H_N))$$

формулага асосан яна икки қадам ҳисоблаш бажарилади. Бу ерда Z – маълумот барча блокларининг модуль икки бўйича йиғиндиси, L – маълумот узунлиги.

Ҳозирги кунда кўплаб давлат стандартлари хэш – функцияларининг алгоритмлари калитсиз хэш – функция алгоритмларидир. Бунга мисол қилиб Россиянинг ГОСТ Р 34.11-94 хэш-функция давлат стандартини, АҚШнинг федерал стандарти FIPS PUB 180 да келтирилган SHA-0, FIPS PUB 180-1 да келтирилган SHA-1, FIPS PUB 180-2 да келтирилган SHA-256, SHA-384, SHA-512 хэш – функцияларини, Беларусь Республикасининг хэш – функция давлат стандарти СТБ 1176.1–99 ни, АҚШнинг федерал стандарти SHA туридаги хэш – функцияларни яратишга асос бўлган MD туридаги хэш – функциялар ва уларнинг модификациялари MD2, MD4 ва MD5 хэш – функцияларини (АҚШнинг федерал стандарти айнан MD5 хэш – функцияси асосида ишлаб чиқилган), Европа Ҳамжамиятининг RACE дастури доирасида MD4 асосида ишлаб чиқилган RIPE-MD ва унинг модификациялари RIPEMD-160, RIPEMD-256 ва RIPEMD-320 хэш – функцияларини, MD5 асосида ишлаб чиқилган HAVAL хэш-функциясини ва юқоридаги хэш – функциялар алгоритмларидан фарқ қилувчи алгоритмга эга бўлган TIGER хэш – функциясини келтириш мумкин [14, 29, 30, 31, 34–37].

Аутентификация атамаси ахборот-коммуникация тармоқларида маълумотлар алмашинуви субъектларининг ҳақиқийликни аниқлашини

билдиради. Бу маълумот алмашишдаги барча аспектларга таъллуқли бўлиб, алоқа сеансининг, томонларнинг, маълумотнинг ҳақиқийлигини билдиради. Бу алоқа тармоғи орқали узатилган маълумот манбаи ва мазмуни жиҳатидан, маълумотнинг яратилган вақти ҳамда жўнатилган вақти жиҳатидан текширганда ҳақиқий бўлишини англатади.

Маълумот тўлалиги – маълумот яратилгандан кейин уни сақлашда ва узатишда унинг бегоналар томонидан ўзгартирилмаганлигига ишонч ҳосил қилишни билдиради. Маълумотни ўзгартириш деганда одатда унга қўшимчалар қўшиш, тушириб қолдириш, ўзгартириш ва маълумот қисмларининг ўрнини алмаштириш тушунилади.

Маълумотнинг манбаини аутентификация қилиш – қабул қилинган электрон ҳужжат ҳақийқий манба томонидан яратилганининг тасдиғини олишдир. Бунда ҳужжат яратилган вақт ва электрон ҳужжатнинг ягоналигини текшириш талаб қилинмайди. Ҳужжат ягоналигининг бузилиши деганда, уни қайтадан узатиш ёки ундан қайтадан фойдаланиш тушунилади.

Маълумотнинг ҳақиқийлиги ва маълумот манбаини аутентификация қилиш тушунчалари бир-бири билан чамбарчас боғлиқдир. Ҳақиқатан ҳам, агар маълумот модификация қилинган бўлса, унинг манбаи ҳам ўзгаради. Агар манба аниқланмаган бўлса, тўлалик масаласини ҳал қилиб бўлмайди.

Энди хэш-функцияни ахборот-коммуникация тизимларида қўллаш схемалари қараб чиқилади [29]:

а) $A \rightarrow B$: $E_K[M \| H(M)]$ (7.1 *а*)-расм)

– махфийликни таъминлайди (K калит фақат A ва B томонларга маълум).

– тўлаликни таъминлайди ($H(M)$ криптографик ҳимояланган).

б) $A \rightarrow B$: $M \| E_K[H(M)]$ (7.1 *б*)-расм)

– тўлаликни таъминлайди (H(M) криптографик ҳимояланган).

в) $A \rightarrow B$: $M \| E_Y[H(M)]$ (7.1 *в*)-расм)

– тўлалик ва рақамли имзони таъминлайди ($H(M)$ криптографик ҳимояланган ҳамда $E_Y[H(M)]$ ни фақат A томон ҳосил қилиши мумкин).

г) $A \rightarrow B$: $E_K[M \| E_Y[H(M)]]$ (7.1 *г*)-расм)

– тўлалик ва рақамли имзони таъминлайди.

– махфийликни таъминлайди (K калит фақат A ва B томонларга маълум).

д) $A \rightarrow B$: $M \| H(M \| S)$ (7.1 *д*)-расм)

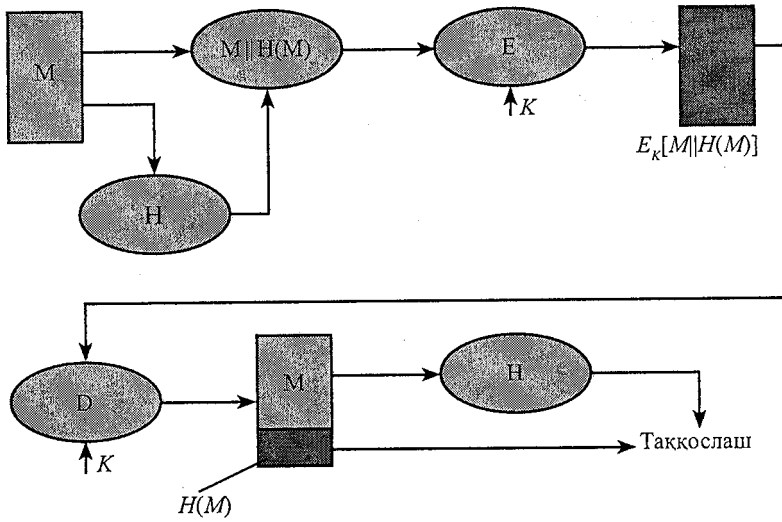
– тўлаликни таъминлайди (S фақат A ва B томонларга маълум).

е) $A \rightarrow B$: $E_K[M \| H(M \| S)]$ (7.1 *е*)-расм)

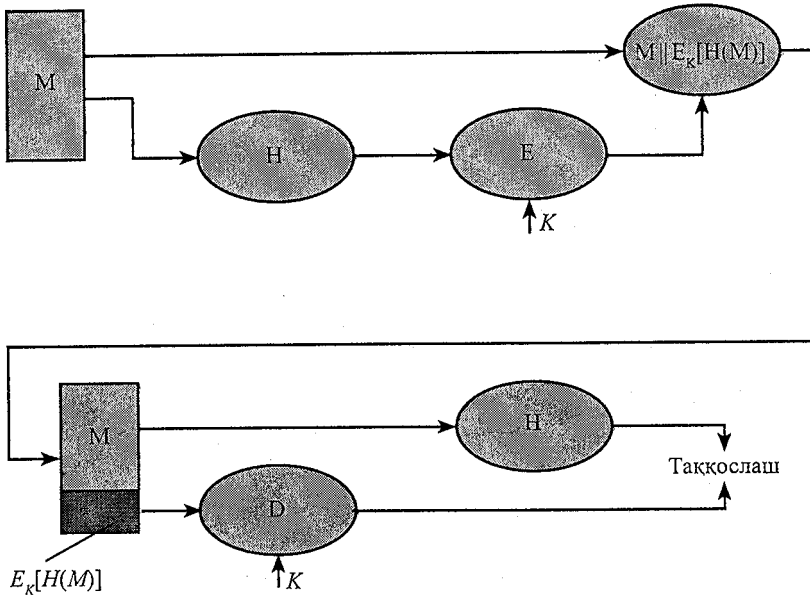
– махфийликни таъминлайди (K калит фақат A ва B томонларга маълум).

– тўлаликни таъминлайди (S фақат A ва B томонларга маълум).

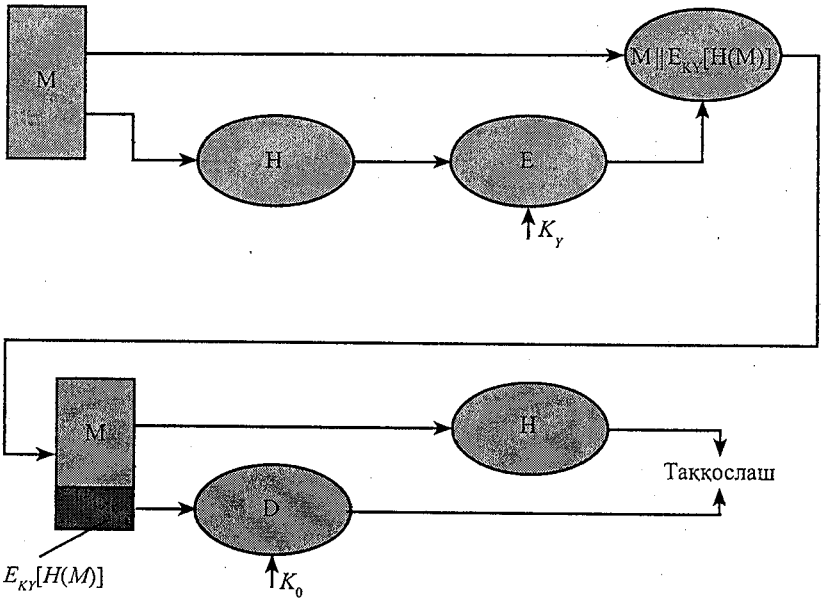
Ахборот-коммуникация тизимларида хэш-функцияларни қўллаш схемалари:



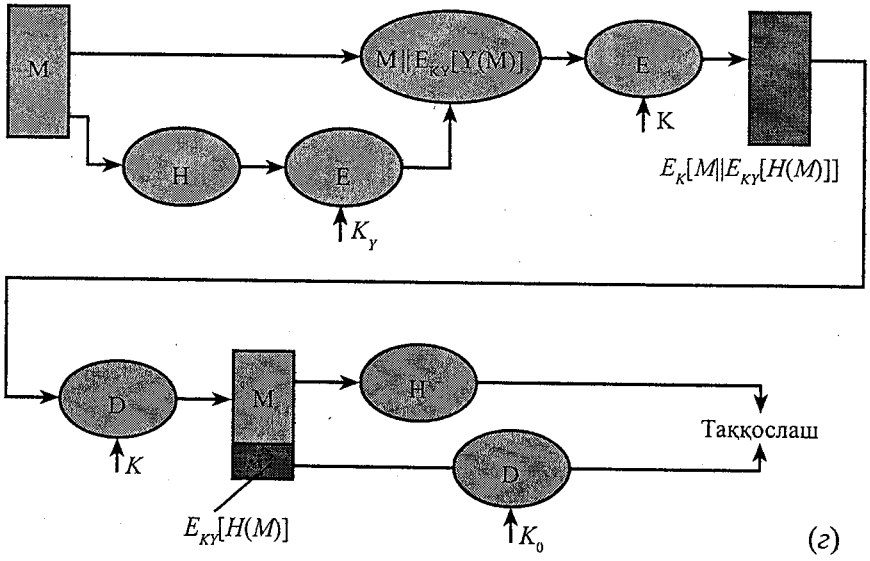
7.1-а-расм.



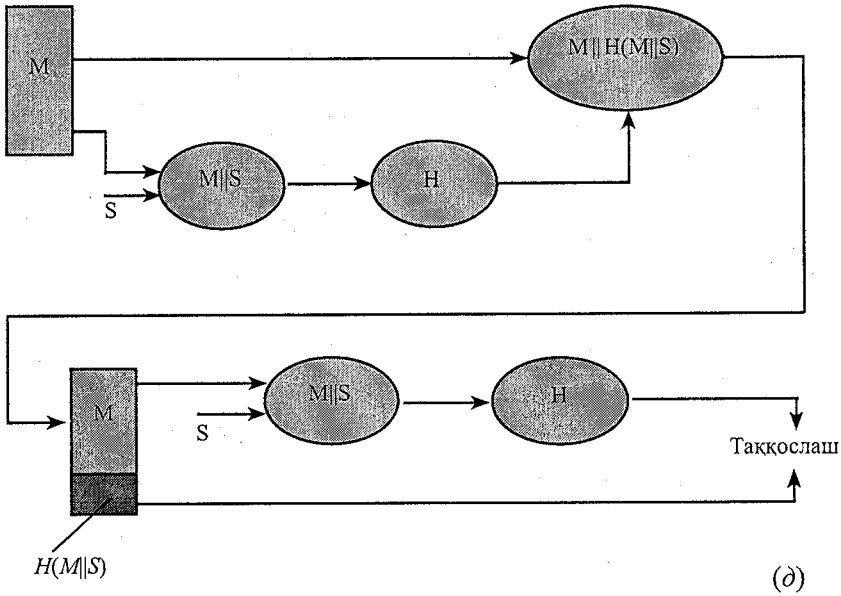
7.1-б-расм.



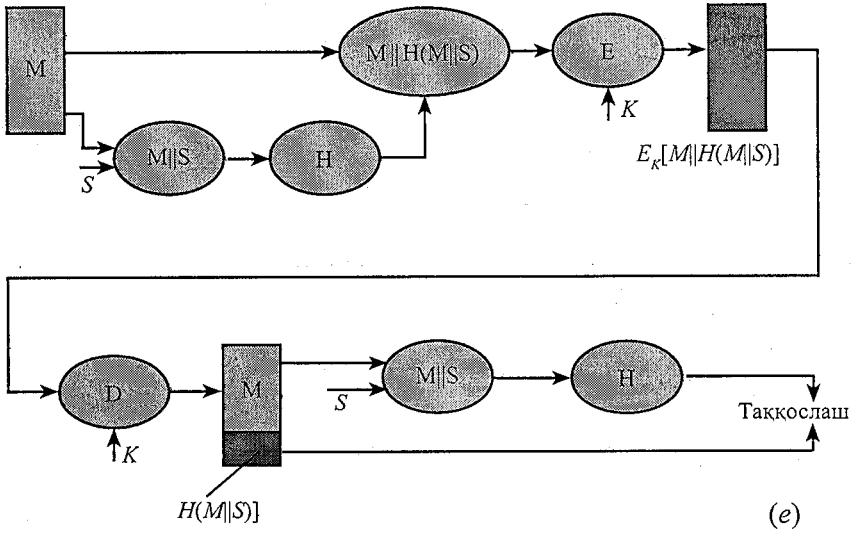
7.1-в-расм.



7.1-г-расм



7.1-д-расм



7.1-е-расм

Қуйида мавжуд стандарт хэш-функцияларининг криптографик хоссалари жадвали келтирилади:

7.1-жадвал

	Хэшланадиган матн узунлиги	Кириш блокининг узунлиги	Хэш қиймат узунлиги	Ҳар бир блокни хэшлаш қадамлари сони
ГОСТ Р 34.11-94	Ихтиёрий	256	256	19
MD 2	Ихтиёрий	512	128	1598
MD 4	Ихтиёрий	512	128	72
MD 5	Ихтиёрий	512	128	88
SHA-1	$<2^{64}$	512	160	80
SHA-256	$<2^{64}$	512	256	64
SHA-384	$<2^{128}$	1024	384	80
SHA-512	$<2^{128}$	1024	512	80
СТБ 1176.1–99	Ихтиёрий	256	$142 \le L \le 256$	77
O'z DSt 1106 : 2006	Ихтиёрий	128, 256	128, 256	$16b+74$, $16b+46$, Бу ерда: b–блоклар сони

§ 7.3. ГОСТ Р 34.11-94 хэш-функцияси алгоритми

Россиянинг ГОСТ Р 34.11-94 хэш-функция стандарти ахборотни криптографик усулда муҳофаза қилиш учун, хусусан ГОСТ Р 34.10-94 ва ГОСТ Р 34.10-2001 электрон рақамли имзо алгоритмларида ишлатиш учун мўлжалланган [35]. Хэш-функциянинг қийматини ҳисоблаш жараёнида ГОСТ 28147-89 шифрлаш стандартидан фойдаланилади.

ГОСТ Р 34.11-94 хэш-функция стандартида чиқиш узунлиги белгиланган қадамли хэшлаш функциясидан фойдаланувчи кетма-кет хэшлаш усулидан фойдаланилади. Хэш-функция аргументининг узунлиги 256 бит бўлган функция бўлиб, хэш қиймат узунлиги 256 бит бўлади. Хэшланадиган маълумот узунлиги ихтиёрий бўлиб, маълумот узунлиги 256 бит бўлган блокларга ажратилади. Охирги блок узунлиги 256 битдан кичик бўлса, 256 битгача ноль билан тўлдирилади. Ундан ташқари, бу блокларнинг охирига маълумот узунлигининг кодини билдирувчи ва назорат йиғиндисини билдирувчи яна иккита 256 битлик блоклар қўшилади. Маълумот узунлигининг кодини блок хэшланадиган маълумотнинг бит узунлиги mod 2^{256} бўйича ҳисобланиб (бу процедура MD кучайтириш дейилади) ҳосил қилинади. Назорат йиғиндисининг кодини билдирувчи блок эса, охирги тўлиқмас блок ноль билан тўлдирилгандан кейин барча блокларнинг йиғиндиси mod 2^{256} бўйича ҳисобланиб ҳосил қилинади.

ГОСТ Р 34.11-94 хэш – функциясини ҳисоблашда қуйидаги белгилашлардан фойдаланилади:

M – хэшланиши керак бўлган маълумот,

h – M маълумотни $h(M) \in V_{256}(2)$ га акслантирувчи хэш – функция, бу ерда $V_{256}(2)$ – узунлиги 256 бит бўлган барча иккилик сўзлар тўплами,

$E_K(A)$ – A ни ГОСТ 28147-89 шифрлаш алгоритмидан фойдаланиб K калитда шифрлаш натижаси,

$H \in V_{256}(2)$ – берилган бошланғич вектор.

ГОСТ Р 34.11-94 хэш – функциясини ҳисоблаш учун қуйидагилар зарур:

- қадамли хэшлаш функцияси $\chi: V_{256}(2) \times V_{256}(2) \to V_{256}(2)$ ни ҳисоблаш алгоритми;
- хэш қийматни итератив ҳисоблаш жараёни.

Қадамли хэшлаш функцияси уч босқичда ҳисобланади. Биринчи босқичда узунликлари 256 бит бўлган тўртта K_1, K_2, K_3, K_4 калит генерация қилинади. Иккинчи босқичда бошланғич H вектор ҳар бирининг узунлиги 64 бит бўлган тўртта блокка ажратилади ва бу блоклар мос K_1, K_2, K_3, K_4 калитлар билан ГОСТ 28147-89 алгоритми ёрдамида шифрланади. Учинчи босқичда шифрлаш натижасини аралаштирувчи акслантириш бажарилади.

Қуйида хэш-функция стандартидаги акслантириш жараёнларининг ҳар бир қадами кўриб чиқилади:

1. Калитлар генерацияси.

$X=(b_{256}, b_{255}, \ldots, b_1) \in V_{256}(2)$ берилган бўлсин.

$X=x_4\|x_3\|x_2\|x_1=\eta_{16}\|\eta_{15}\|\ldots\|\eta_1=\xi_{32}\|\xi_{31}\|\ldots\|\xi_1$ деб оламиз, бу ерда $x_i \in V_{64}(2)$ $i=1, 2, 3, 4$, $\eta_j \in V_{16}(2)$, $j=1,\ldots,16$, $\xi_k \in V_8(2)$, $k=1,\ldots,32$ бўлади.

$A(X)=(x_1 \oplus x_2)\|x_4\|x_3\|x_2$, деб белгиланади.

P: $V_{256}(2) \to V_{256}(2)$ акслантириш $\xi_{32}\|\xi_{31}\|\ldots\|\xi_1$ ни $\xi_{\varphi(32)}\|\xi_{\varphi(31)}\|\ldots\|\xi_{\varphi(1)}$ га акслантирсин, бу ерда $\varphi(i+1+4(k-1))=8i+k$, $i=0, 1, 2, 3$; $k=1, 2, \ldots, 8$.

K_1, K_2, K_3, K_4 калитларни генерация қилиш учун H, $M \in V_{256}(2)$ берилганлар ҳамда $C_2=C_4=0^{256}$ ва $C_3=1^80^81^{16}0^{24}1^{16}0^8(0^81^8)^21^80^8(0^81^8)^4(1^80^8)^4$ – ўзгармаслардан фойдаланилади.

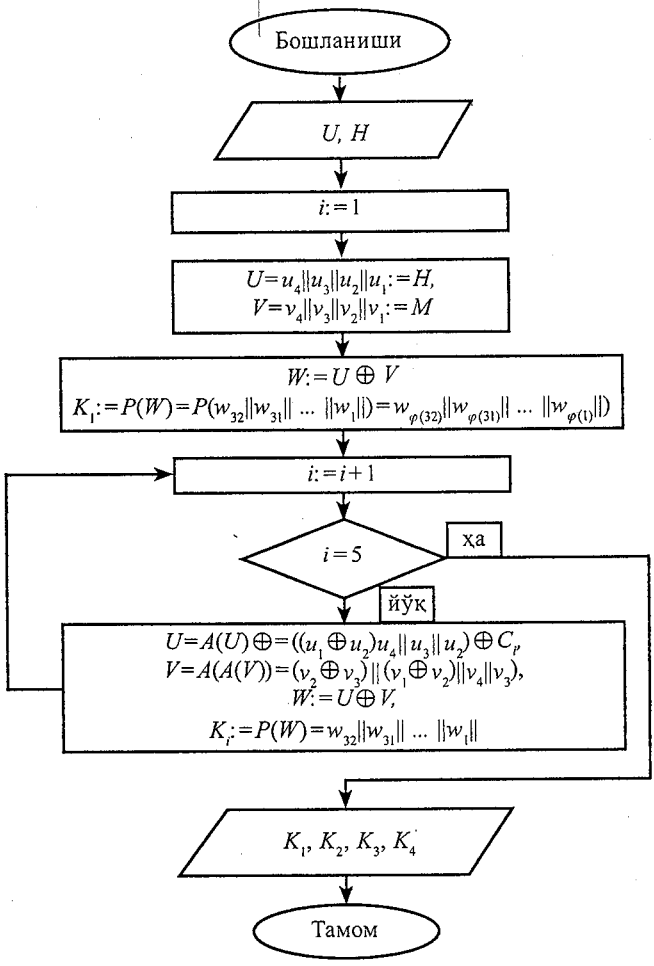
Махфий калитларни генерация қилиш алгоритми қуйидаги қадамлар асосида амалга оширилади:

1. $i=1$, $U=H$, $V=M$.
2. $W=U \oplus V$, $K_1=P(W)$.
3. $i=i+1$.
4. $i=5$ шартни текширамиз, агар бу шарт бажарилса 7-қадамга ўтилади, акс ҳолда 5-қадамга ўтилади.
5. $U=A(U) \oplus C_i$, $V=A(A(V))$, $W=U \oplus V$, $K_i=P(W)$.

6. 3-қадамга ўтилади.

7. Алгоритм ишини тугатади.

Қуйида ушбу алгоритмнинг блок-схемаси келтирилган:



2. Шифрловчи акслантириш.

Бу босқичда H ни тўртта 64 битлик қисмларга ажратамиз ва уларни K_1, K_2, K_3, K_4 – калитлар ёрдамида шифрлаймиз.

Шифрловчи акслантиришда $H=h_4||h_3||h_2||h_1$, $h_i \in V_{64}(2)$ $i=1, 2, 3, 4$ берилганлар ва K_1, K_2, K_3, K_4 – калитлардан фойдаланилади.

Шифрлагандан кейин $s_i=E_{Ki}(h_i)$, $i=1, 2, 3, 4$ ни ҳосил қиламиз. Натижада $S=s_4||s_3||s_2||s_1$ вектор ҳосил бўлади.

3. Аралаштирувчи акслантириш.

Бу босқичда берилган 256 битлик кетма-кетлик 16 битлик сўзларга ажратилиб, уларни аралаштирувчи акслантириш бажарилади. Бунинг учун бизга H, M, $S \in V_{256}(2)$лар берилган бўлади.

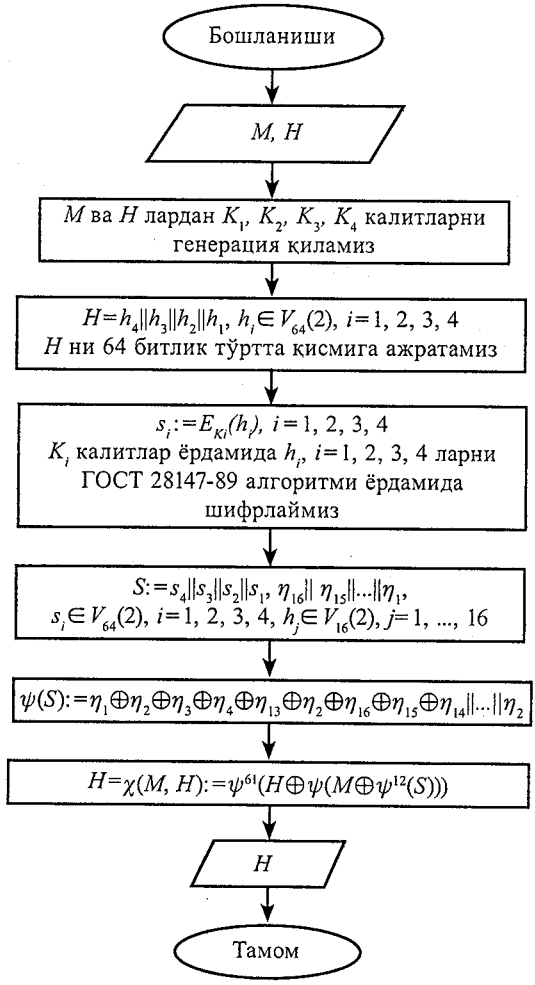
$\psi : V_{256}(2) \to V_{256}(2)$ акслантириш $\eta_{16}\|\eta_{15}\|\ldots\|\eta_1$, η_j $V_{16}(2)$, $j = 1, \ldots, 16$ сўзни $\eta_1 \oplus \eta_2 \oplus \eta_3 \oplus \eta_4 \oplus \eta_{13} \oplus \eta_{16}\|\eta_{16}\|\eta_{15}\| \ldots \|\eta_2$ сўзга акслантирсин.

У ҳолда қадамли хэшлаш (итерация) функцияси қуйидагича аниқланади:

$$\chi(M, H) = \psi^{61}(H \oplus \psi\ (M \oplus \psi^{12}(S))).$$

Бу ерда: ψ^i – ψ акслантиришнинг i – даражаси.

Қуйида қадамли хэшлаш функциясини ҳисоблаш алгоритмининг блок-схемаси келтирилган:



Хэш – функцияни ҳисоблаш тартиботи (процедураси).

M – хэшланиши керак бўлган маълумот берилган бўлсин. h хэш қийматни ҳисоблаш учун параметр сифатида $H \in V_{256}(2)$ бошланғич вектор берилган бўлсин.

h хэш қийматни ҳисоблаш жараёнининг ҳар бир итерациясида қуйидаги миқдорлардан фойдаланилади:

M – берилган хэшланиши керак бўлган маълумотнинг олдинги итерацияларда хэшлаш жараёнидан ўтмаган қисми;

$H \in V_{256}(2)$ – хэш-функциянинг жорий қиймати;

$Sum \in V_{256}(2)$ – назорат йиғиндисининг жорий қиймати;

$L \in V_{256}(2)$ – берилган маълумотнинг олдинги итерациялардан ўтган қисми узунлигининг жорий қиймати.

Хэш қийматни ҳисоблаш жараёни қуйидаги учта босқичдан иборат:

1-босқич.

Жорий миқдорларга бошланғич қийматлар берилади:

1.1 $M := M$;

1.2. $H := H$;

1.3. $Sum := 0^{256}$;

1.4. $L := 0^{256}$.

2-босқич.

2.1. $|M|>256$ шарт текширилади;

Агар бу шарт бажарилса 3-босқичга ўтилади, акс ҳолда қуйидаги ҳисоблашлар кетма-кетлиги бажарилади:

2.2. $L=(L+|M|) \bmod 2^{256}$;

2.3. $T=0^{256-|M|}\|M$;

2.4. $\mathrm{Sum}=(\mathrm{Sum}+T) \bmod 2^{256}$;

2.5. $H=\chi(T, H)$;

2.6. $H=\chi(L, H)$;

2.7. $H=\chi(Sum, H)$;

2.8. Алгоритм ўз ишини тугаллайди.

3-босқич.

3.1. Берилган M маълумотнинг $M_S \in V_{256}(2)$ қисми ажратиб олинади ($M=M_P\|M_S$); кейин қуйидаги ҳисоблашлар кетма-кетлиги бажарилади:

3.2. $\mathrm{H}=\chi(M_S, H)$;

3.3. $L=(L+256) \bmod 2^{256}$;

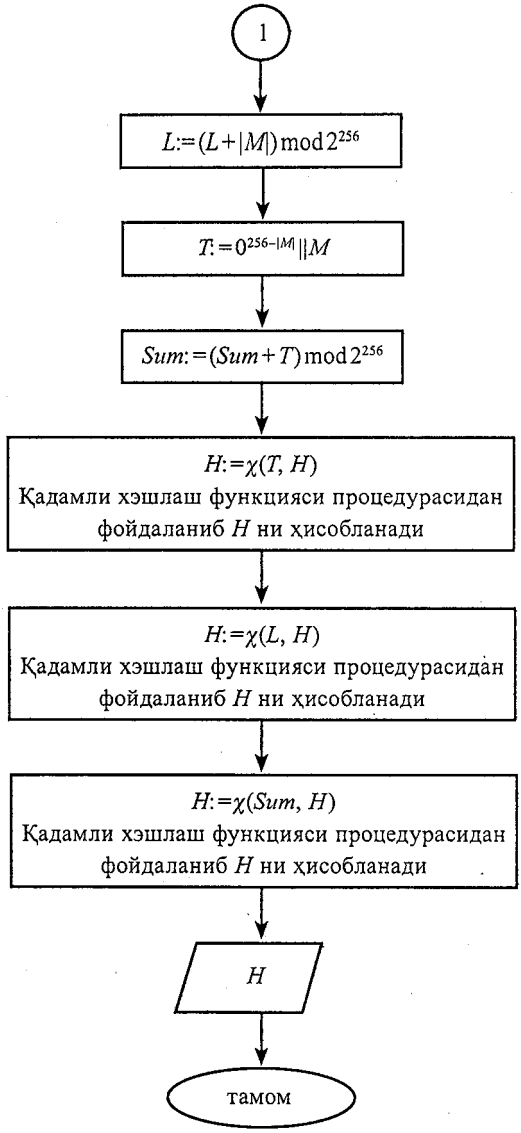
3.4. $Sum=(Sum+M_s) \bmod 2^{256}$;

3.5. $M=M_p$;

3.6. Иккинчи босқичга ўтилади.

2.7-қадамда олинган H нинг қиймати 3.2-қадамда M маълумотнинг хэш қиймати бўлади.

Қуйида ГОСТ 34.11-94 хэш – функция алгоритмининг блок-схемаси келтирилган:



§ 7.4. MD 5 хэш – функцияси акслантиришларининг мураккаблик даражаларини баҳолаш

MD 5 хэш-функцияси алгоритми Массачусетс технология институти профессори Рональд Ривест томонидан 1992 йилда ишлаб чиқилган. Бу алгоритмда кирувчи маълумот узунлиги ихтиёрий бўлиб, хэш қиймат узунлиги 128 бит бўлади. MD 5 хэш – функция-

си алгоритмида кирувчи маълумот 512 битлик блокларга ажратилиб, улар 16 та 32 битлик қисм блокларга ажратилади ва булар устида амаллар бажарилади [38].

Фараз қилайлик, бизга узунлиги b бит бўлган, бу ерда b – ихтиёрий манфий бўлмаган бутун сон, маълумот берилган бўлсин ва бу маълумотнинг битлари $m_0 m_1 \ldots m_{(b-1)}$ тартибда ёзилган бўлсин.

Хэш қийматни ҳисоблаш учун қуйидаги бешта босқич бажарилади:

1-босқич. Тўлдириш битларини қўшиш.

Берилган маълумот узунлиги 512 модуль бўйича 448 билан таққосланадиган (маълумот узунлиги≡448 mod512) қилиб тўлдирилади, яъни, кенгайтирилган маълумотнинг узунлиги унга энг яқин бўлган 512 га каррали сондан 64 битга кичик бўлиши керак. Тўлдириш ҳамма вақт, ҳаттоки, маълумот узунлиги 512 модуль бўйича 448 билан таққосланадиган бўлса ҳам бажарилади.

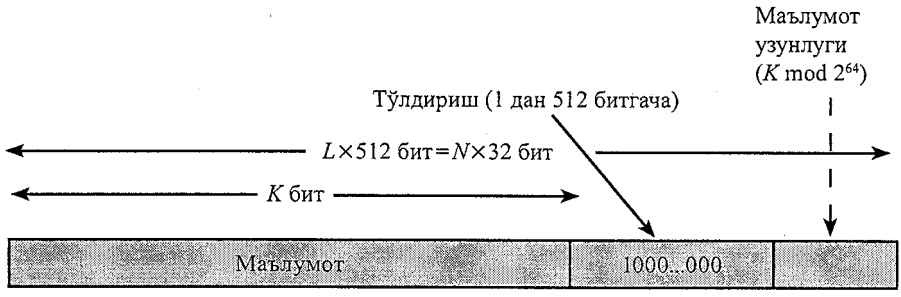
Тўлдириш қуйидаги тартибда амалга оширилади: маълумотга 1 га тенг бўлган битта бит қўшилади, қолган битлар эса ноль билан тўлдирилади. Шунинг учун қўшилган битлар сони 1 дан 512 тагача бўлади.

2- босқич. Маълумотнинг узунлигини қўшиш.

1-босқич натижасига берилган маълумот узунлигининг 64 битлик қиймати қўшилади. Агар маълумот узунлиги 2^{64} битдан катта бўлса, бу узунлик mod 2^{64} бўйича олиниб қўшилади.

Шундай қилиб, биринчи иккита босқич бажарилгандан кейин узунлиги 512 битга каррали бўлган маълумот олинади, яъни кенгайтирилган маълумот узунлиги 16 та 32 битлик сўздан иборат блок узунлигига каррали бўлади. Натижада ҳосил қилинган маълумотнинг сўзларини $M[0, \ldots, N-1]$ орқали белгилаймиз, у ҳолда N сони 16 га каррали бўлади. Шундай қилиб, $N=L\times16$ бўлади.

Ушбу иккита босқични қуйидагича тасвирлаш мумкин:



3-босқич. Хэш қиймат учун буфер инициализация қилиш.

Хэш – функциянинг оралиқ ва охирги натижаларини сақлаш учун 128 битлик буфердан фойдаланилади. Бу буферни тўртта 32 битлик A, B, C, D регистрлар кўринишида тасвирлаш мумкин. Бу регистрларга 16 лик саноқ тизимида қуйидаги бошланғич қийматлар берилади:

A=0×01234567

B=0×89ABCDEF

C=0×FEDCBA98

D=0×76543210.

4-босқич. Маълумотни 512 битлик блокларга ажратиб қайта ишлаш.

Аргументи ва қиймати 32 битлик сўз бўладиган тўртта ёрдамчи функцияни аниқлаймиз:

$F(X, Y, Z)=(X \wedge Y) \vee (\neg X \wedge Z)$

$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$

$H(X, Y, Z)=X \oplus Y \oplus Z$

$I(X, Y, Z)=Y \oplus (X \vee \neg Z)$

Бу ерда битлар бўйича мантиқий AND, OR, NOT, XOR амаллари мос равишда $\wedge$, $\vee$, $\neg$, $\oplus$ белгилари билан ифодаланган.

Бу босқичда синус функцияси асосида 64 та сўздан қурилган $T[1,\ldots,64]$ жадвалдан фойдаланилади. $T[i]=[4294967296 \times \text{abs}(\sin(i))]$ бўлиб, жадвалнинг i – элементини ифодалайди. Бу ерда $[q]$ ифода q соннинг бутун қисмини билдиради, i эса радианларда ифодаланган.

Ушбу босқичда қуйидаги амаллар бажарилади:

/* Ҳар бир 16 сўзлик блок қайта ишланади. */

for I=0 to N/16 – 1 do

/* i – блок X га ёзиб олинади. */

For j = 0 to 15 do

X[j] = M[i*16+j].

/* A нинг қиймати AA га, B нинг қиймати BB га, C нинг қиймати CC га, D нинг қиймати DD га ёзиб олинади. */

$AA=A$

$BB=B$

$CC=C$

$DD=D$

/* 1-қадам. */

/* [*abcd k s i*] ифода қуйидаги амални билдиради:

$a=b+((a+\mathrm{F}(b,c,d)+\mathrm{X}[k]+\mathrm{T}[i])$ <<< $s)$. */

/* Қуйидаги 16 та амал бажарилади. */

[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

/* 2-қадам. */

/* [*abcd k s i*] ифода қуйидаги амални билдиради:

$a=b+((a+\mathrm{G}(b,c,d)+\mathrm{X}[k]+\mathrm{T}[i])$ <<< $s)$. */

/* Қуйидаги 16 та амал бажарилади.*/

[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

/* 3-қадам. */

/* [*abcd k s i*] ифода қуйидаги амални билдиради:

$a=b+((a+\mathrm{H}(b,c,d)+\mathrm{X}[k]+\mathrm{T}[i])$ <<< $s)$. */

/* Қуйидаги 16 та амал бажарилади.*/

[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]

/* 4-қадам. */

/* [*abcd k s i*] ифода қуйидаги амални билдиради:

$a=b+((a+\mathrm{I}(b,c,d)+\mathrm{X}[k]+\mathrm{T}[i])$ <<< $s)$. */

/* Қуйидаги 16 та амал бажарилади. */

[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

/* Қуйидаги қўшиш амали бажарилади. */

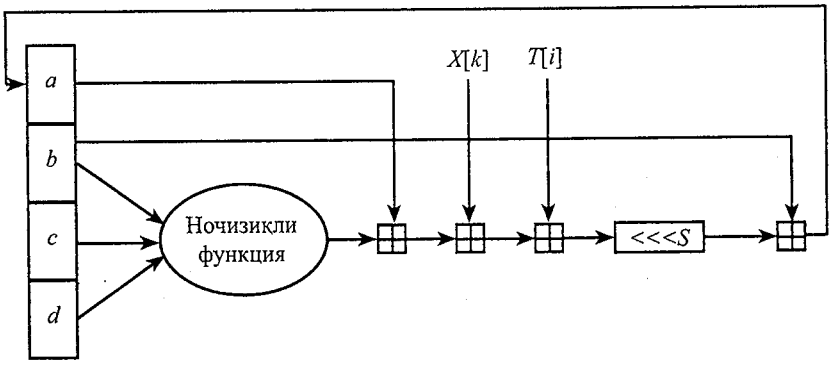
$A=A+AA$

$B=B+BB$

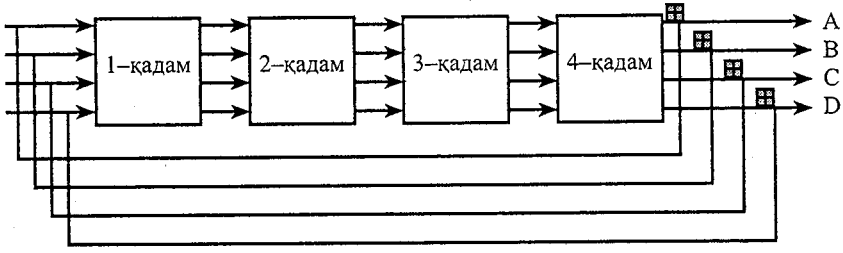
$C=C+CC$

$D=D+DD$

end /* I бўйича цикл */



MD 5 алгоритмида битта регистр қийматини ҳисоблаш.

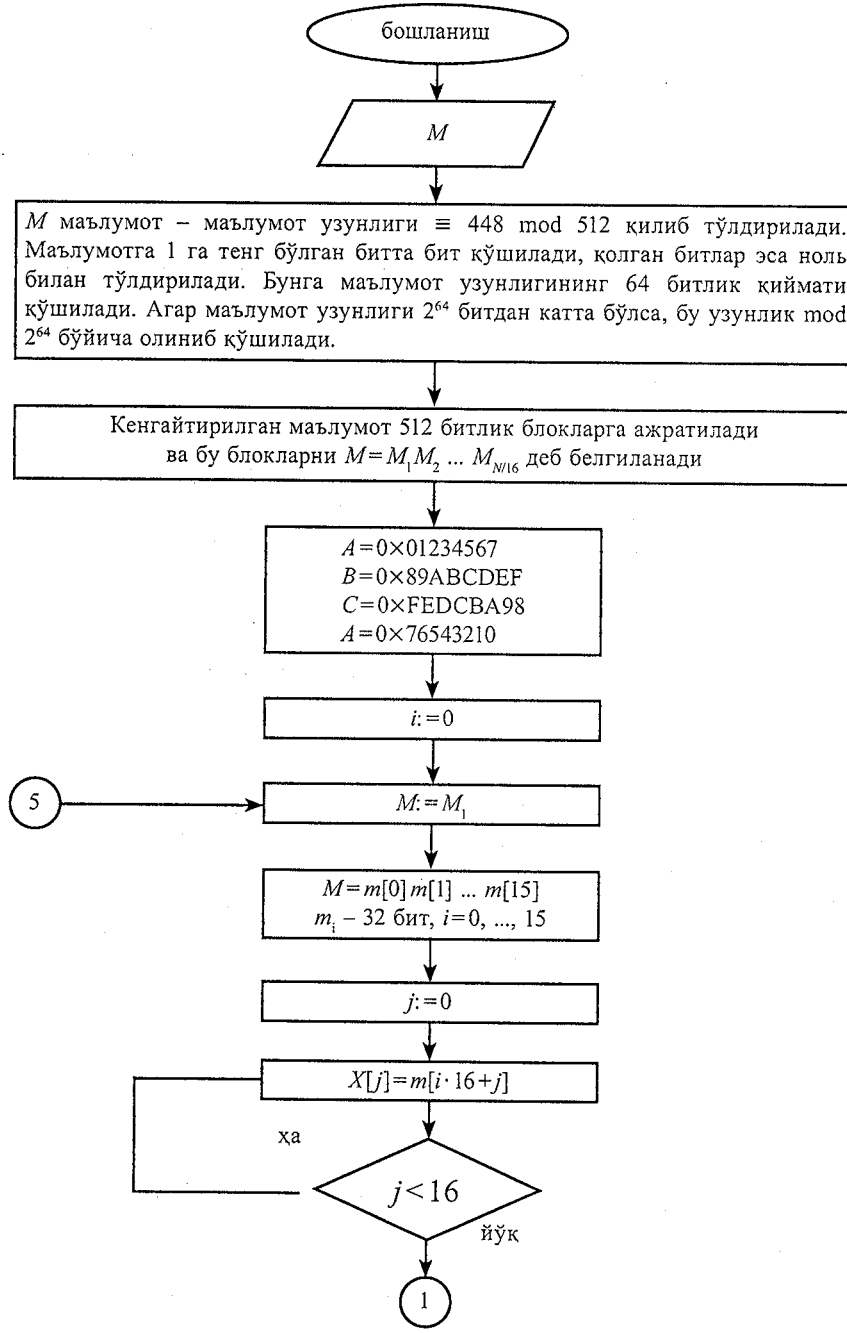


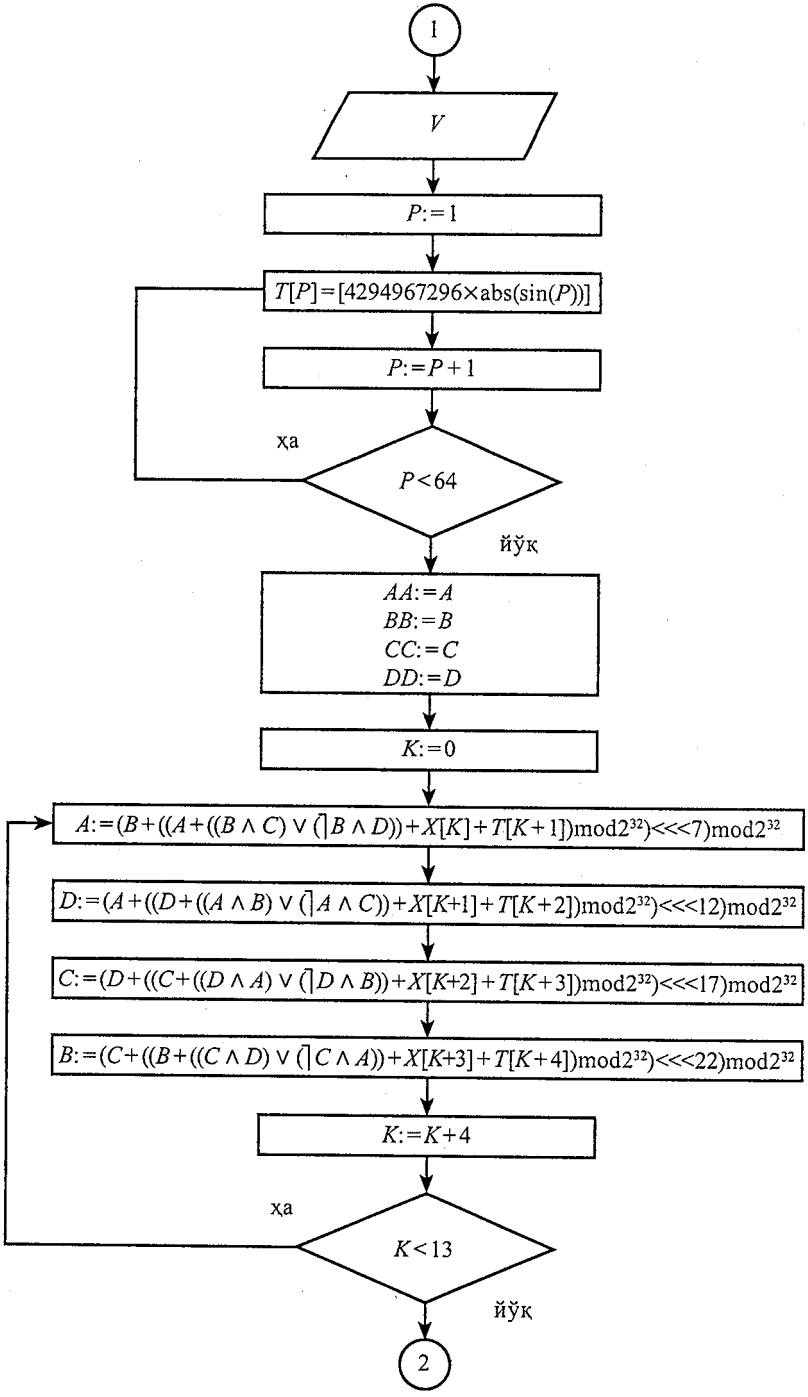
MD 5 алгоритмидаги асосий цикл.

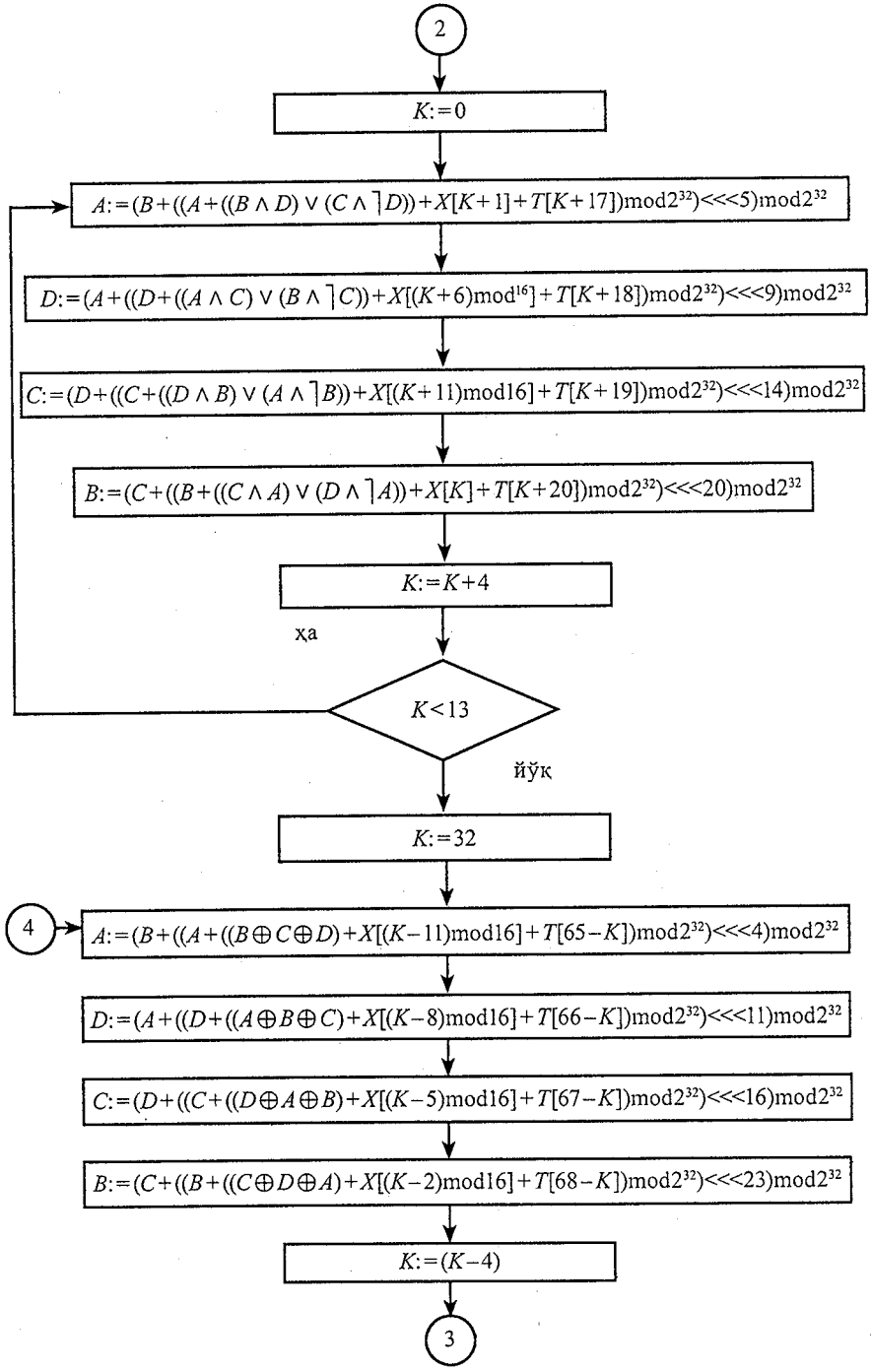
5- босқич. Натижа.

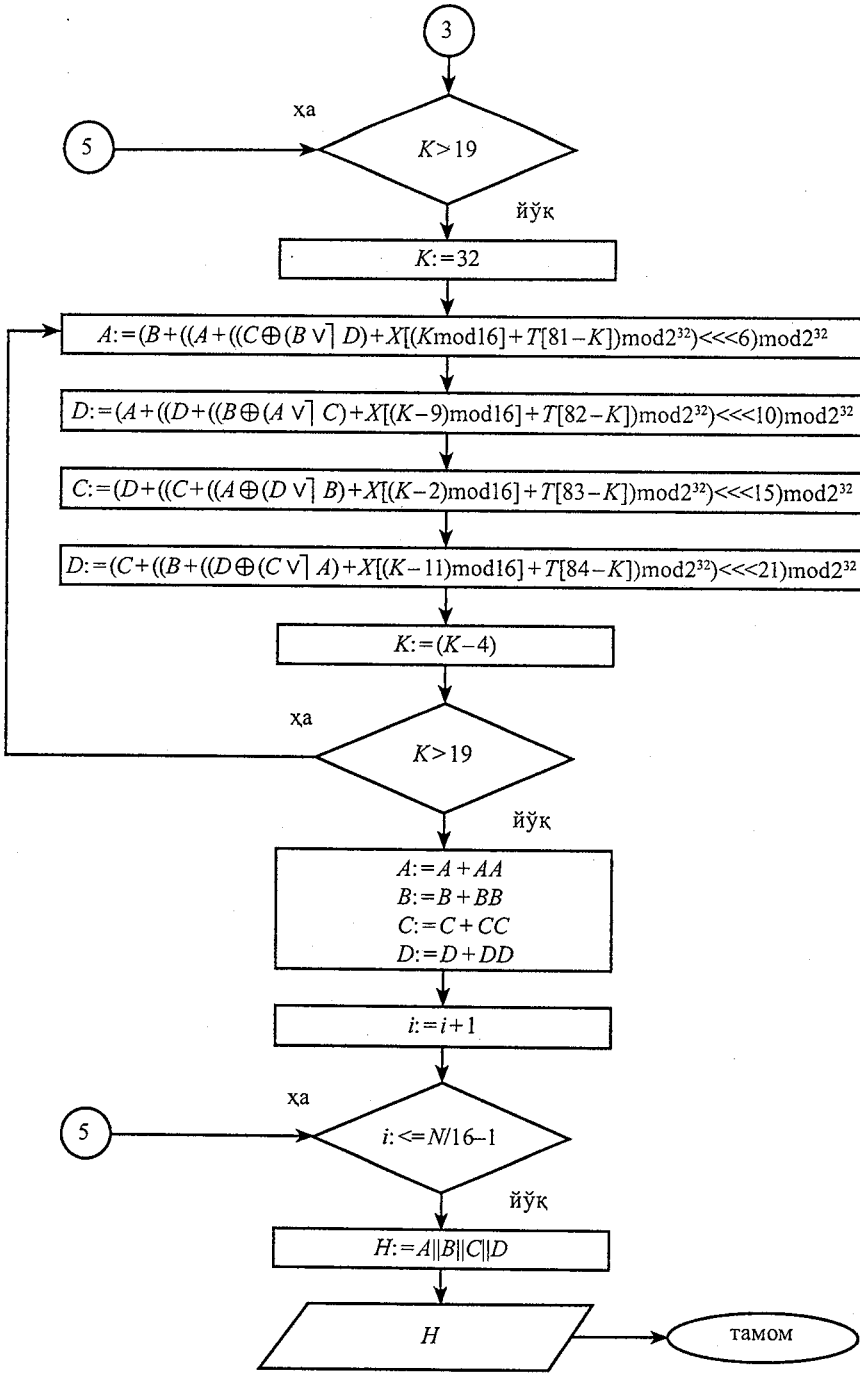
Маълумотнинг хэш қиймати A, B, C, D регистрлардаги қийматларни бирлаштириш натижасида ҳосил қилинади.

Қуйида MD 5 хэш-функцияси алгоритмининг блок схемасини келтирилади:


бошланиш
M
M маълумот – маълумот узунлиги ≡ 448 mod 512 қилиб тўлдирилади. Маълумотга 1 га тенг бўлган битта бит қўшилади, қолган битлар эса ноль билан тўлдирилади. Бунга маълумот узунлигининг 64 битлик қиймати қўшилади. Агар маълумот узунлиги 2^64 битдан катта бўлса, бу узунлик mod 2^64 бўйича олиниб қўшилади.
Кенгайтирилган маълумот 512 битлик блокларга ажратилади ва бу блокларни M=M1M2 ... MN/16 деб белгиланади
A=0×01234567
B=0×89ABCDEF
C=0×FEDCBA98
A=0×76543210
i:=0
5
M:=M1
M=m[0] m[1] ... m[15]
mi – 32 бит, i=0, ..., 15
j:=0
X[j]=m[i·16+j]
ҳа
j<16
йўқ
1


1
V
P:=1
T[P]=[4294967296×abs(sin(P))]
P:=P+1
ҳа
P<64
йўқ
AA:=A
BB:=B
CC:=C
DD:=D
K:=0
A:=(B+((A+((B ∧ C) ∨ (¬B ∧ D))+X[K]+T[K+1])mod2^32)<<<7)mod2^32
D:=(A+((D+((A ∧ B) ∨ (¬A ∧ C))+X[K+1]+T[K+2])mod2^32)<<<12)mod2^32
C:=(D+((C+((D ∧ A) ∨ (¬D ∧ B))+X[K+2]+T[K+3])mod2^32)<<<17)mod2^32
B:=(C+((B+((C ∧ D) ∨ (¬C ∧ A))+X[K+3]+T[K+4])mod2^32)<<<22)mod2^32
K:=K+4
ҳа
K<13
йўқ
2


2
K:=0
A:=(B+((A+((B ∧ D) ∨ (C ∧ ¬D))+X[K+1]+T[K+17])mod2^32)<<<5)mod2^32
D:=(A+((D+((A ∧ C) ∨ (B ∧ ¬C))+X[(K+6)mod16]+T[K+18])mod2^32)<<<9)mod2^32
C:=(D+((C+((D ∧ B) ∨ (A ∧ ¬B))+X[(K+11)mod16]+T[K+19])mod2^32)<<<14)mod2^32
B:=(C+((B+((C ∧ A) ∨ (D ∧ ¬A))+X[K]+T[K+20])mod2^32)<<<20)mod2^32
K:=K+4
ҳа
K<13
йўқ
K:=32
4
A:=(B+((A+((B⊕C⊕D)+X[(K−11)mod16]+T[65−K])mod2^32)<<<4)mod2^32
D:=(A+((D+((A⊕B⊕C)+X[(K−8)mod16]+T[66−K])mod2^32)<<<11)mod2^32
C:=(D+((C+((D⊕A⊕B)+X[(K−5)mod16]+T[67−K])mod2^32)<<<16)mod2^32
B:=(C+((B+((C⊕D⊕A)+X[(K−2)mod16]+T[68−K])mod2^32)<<<23)mod2^32
K:=(K−4)
3


3
ха
5
K>19
йўқ
K:=32
A:=(B+((A+((C⊕(B ∨¬ D)+X[(Kmod16]+T[81−K])mod2^32)<<<6)mod2^32
D:=(A+((D+((B⊕(A ∨¬ C)+X[(K−9)mod16]+T[82−K])mod2^32)<<<10)mod2^32
C:=(D+((C+((A⊕(D ∨¬ B)+X[(K−2)mod16]+T[83−K])mod2^32)<<<15)mod2^32
D:=(C+((B+((D⊕(C ∨¬ A)+X[(K−11)mod16]+T[84−K])mod2^32)<<<21)mod2^32
K:=(K−4)
ҳа
K>19
йўқ
A:=A+AA
B:=B+BB
C:=C+CC
D:=D+DD
i:=i+1
ха
5
i:<=N/16−1
йўқ
H:=A||B||C||D
H
тамом

§ 7.5. SHA-1 хэш – функцияси алгоритми

Кафолатланган бардошлиликка эга бўлган хэшлаш алгоритми SHA (Secure Hash Algorithm) АҚШнинг стандартлар ва технологиялар Миллий институти (NIST) томонидан ишлаб чиқилган бўлиб, 1992 йилда ахборотни қайта ишлаш федерал стандарти (PUB FIPS 180) кўринишида нашр қилинди [36]. 1995 йилда бу стандарт қайтадан кўриб чиқилди ва SHA-1 деб номланди (PUB FIPS 180-1). SHA алгоритми MD4 алгоритмига асосланади ва унинг тузилиши MD4 алгоритмининг тузилишига жуда яқин. Бу алгоритм DSS стандарти асосидаги электрон рақамли имзо алгоритмларида ишлатиш учун мўлжалланган.

Бу алгоритмда кирувчи маълумот узунлиги 2^{64} битдан кичик, хэш қиймат узунлиги 160 бит бўлади. Киритилаётган маълумот 512 битлик блокларга ажратилиб қайта ишланади.

Хэш қийматни ҳисоблаш жараёни қуйидаги босқичлардан иборат:

1-босқич. Тўлдириш битларини қўшиш.

Берилган маълумот узунлиги 512 модуль бўйича 448 билан таққосланадиган (маълумот узунлиги≡448 mod 512) қилиб тўлдирилади. Тўлдириш ҳамма вақт, ҳаттоки маълумот узунлиги 512 модуль бўйича 448 билан таққосланадиган бўлса ҳам бажарилади.

Тўлдириш қуйидаги тартибда амалга оширилади: маълумотга 1 га тенг бўлган битта бит қўшилади, қолган битлар эса ноль билан тўлдирилади. Шунинг учун қўшилган битлар сони 1 дан 512 тагача бўлади.

2- босқич. Маълумотнинг узунлигини қўшиш.

1-босқич натижасига берилган маълумот узунлигининг 64 битлик қиймати қўшилади.

3- босқич. Хэш қиймат учун буфер инициализация қилиш.

Хэш – функциянинг оралиқ ва охирги натижаларини сақлаш учун 160 битлик буфердан фойдаланилади. Бу буферни бешта 32 битлик A, B, C, D, E регистрлар кўринишида тасвирлаш мумкин. Бу регистрларга 16 лик саноқ тизимида қуйидаги бошланғич қийматлар берилади:

A=0×67452301,
B=0×EFCDAB89,
C=0×98BADCFE,
D=0×10325476,
E=0×C3D2E1F0.

Кейинчалик бу ўзгарувчилар мос равишда янги a, b, c, d ва e ўзгарувчиларга ёзиб олинади.

4-босқич. Маълумотни 512 битлик блокларга ажратиб қайта ишлаш.

Бу хэш – функциянинг асосий цикли қуйидагича бўлади:

for (t=0; t < 80; t++){
$temp=(a <<< 5)+f_t(b, c, d)+e+W_t+K_t$;
$e=d$; $d=c$; $c=b <<< 30$; $b = a$; $a=temp$;
},

Бу ерда: <<< – чапга циклик суриш амали. K_t лар 16 лик саноқ тизимида ёзилган қуйидаги сонлардан иборат:

$$K_t = \begin{cases} 5A8227999, & t = 0, ..., 19, \\ 6ED9EBA1, & t = 20, ..., 39, \\ 8F1BBCDC, & t = 40, ..., 59, \\ CA62C1D6, & t = 60, ..., 79. \end{cases}$$

$f_t(x, y, z)$ функциялар эса қуйидаги ифодалар орқали аниқланади:

$$f_t(x, y, z) = \begin{cases} X \wedge Y \wedge \neg X \wedge Z, & t = 0, ..., 19, \\ X \oplus Y \oplus Z, & t = 20, ..., 39,\ 60, ..., 79, \\ X \wedge Y \vee X \wedge Z \vee Y \wedge Z, & t = 40, ..., 59, \end{cases}$$

W_t лар кенгайтирилган маълумотнинг 512 битлик блокининг 32 битлик қисм блокларидан қуйидаги қоида бўйича ҳосил қилинади:

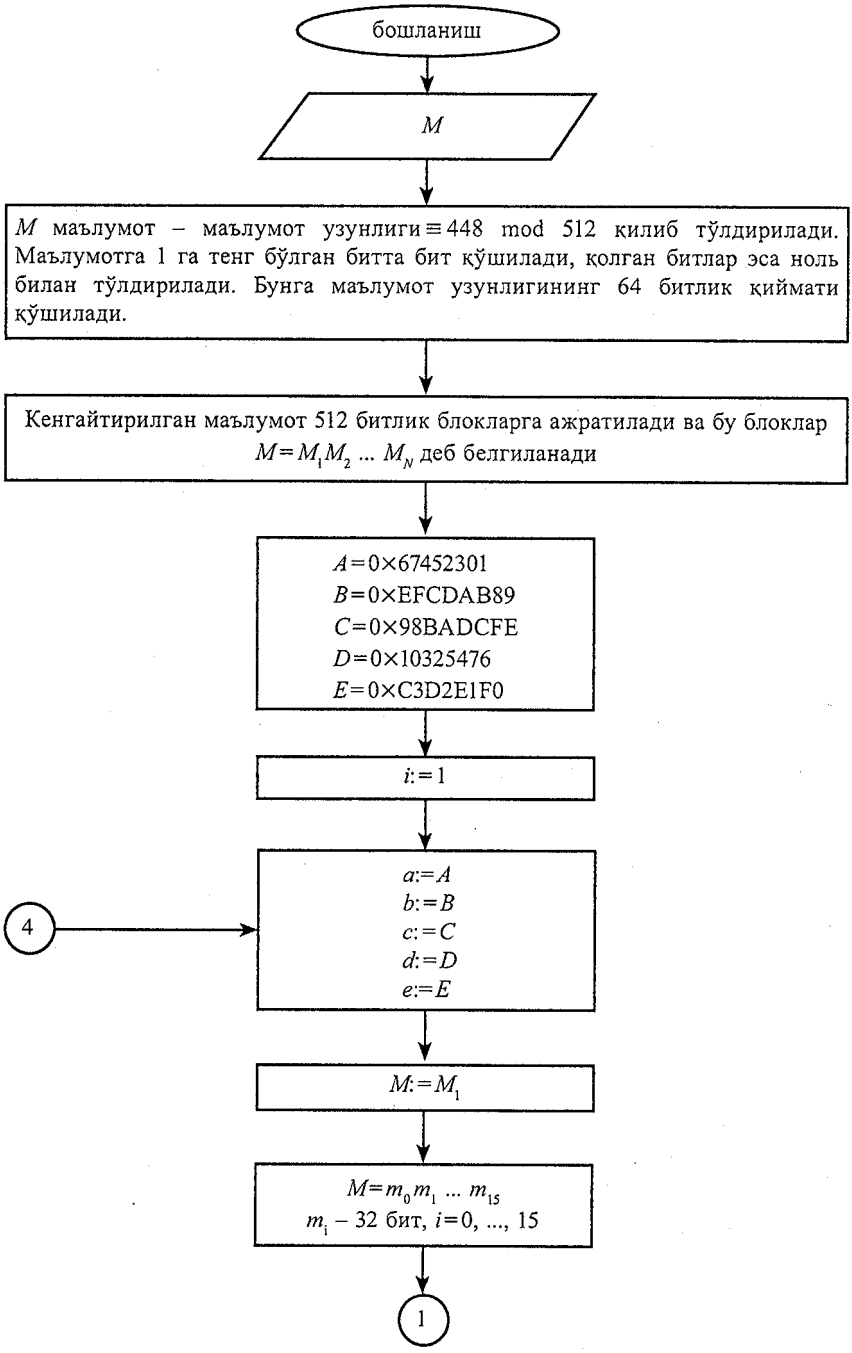
$$W_t = \begin{cases} M_t, & t = 0, ..., 15, \\ (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1, & t = 16, ..., 79. \end{cases}$$

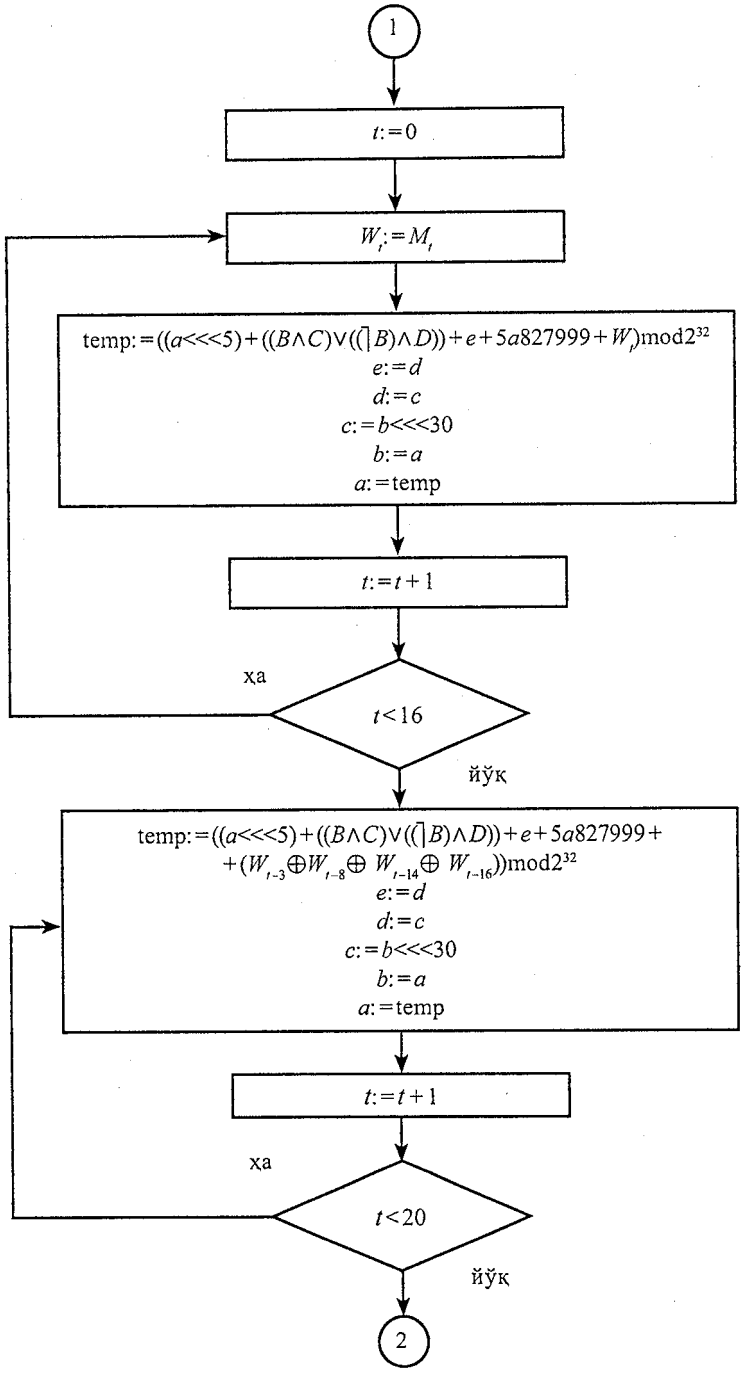
Асосий цикл тугагандан кейин, a, b, c, d ва e ларнинг қийматлари мос равишда A, B, C, D ва E регистрлардаги қийматларга қўшилади ҳамда шу регистрларга ёзиб қўйилади ва кенгайтирилган маълумот кейинги 512 битлик блокини қайта ишлашга ўтилади.

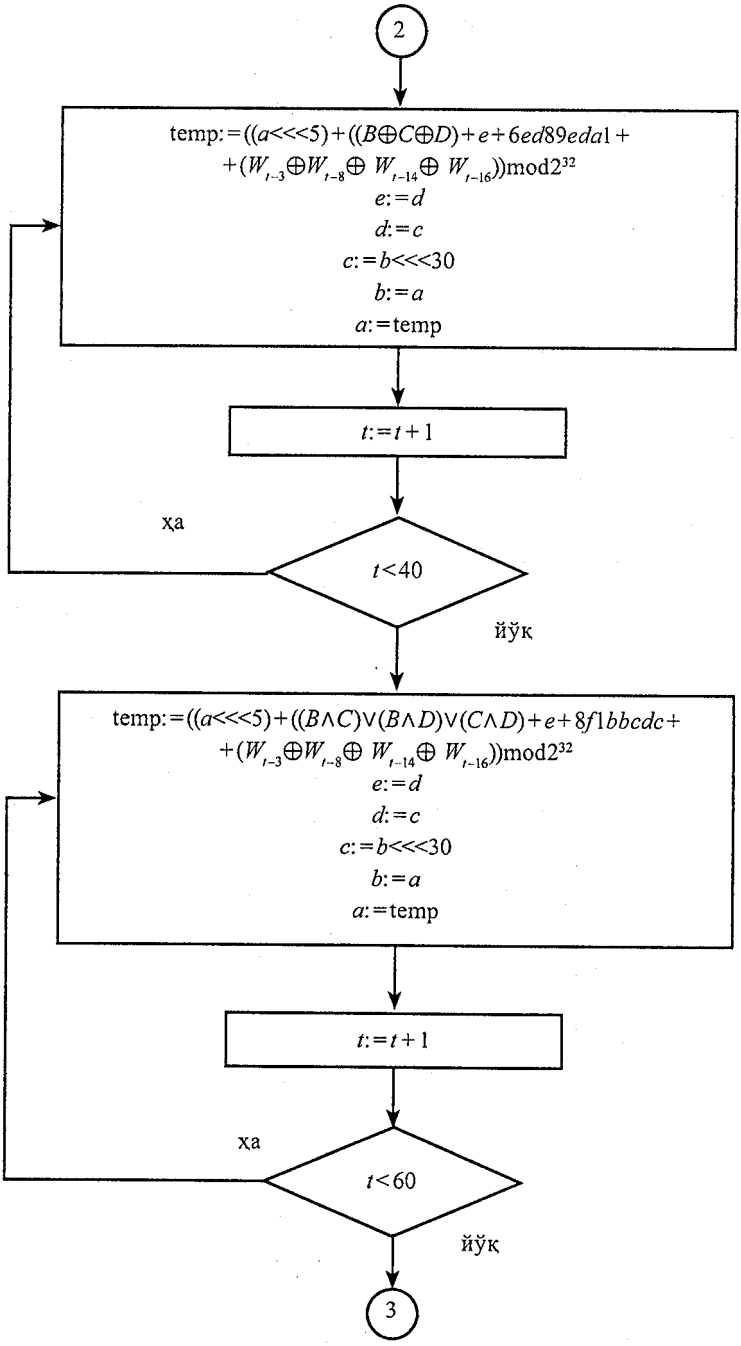
5-босқич. Натижа.

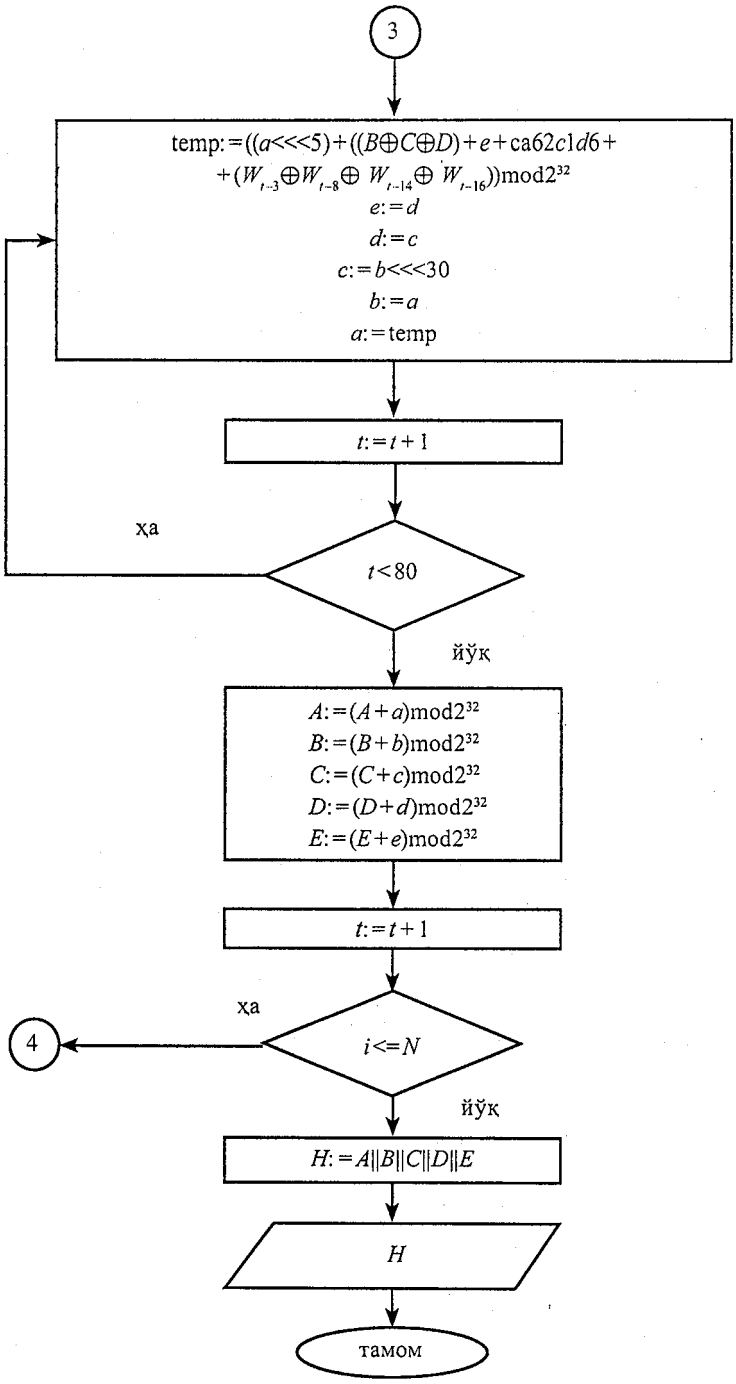
Маълумотнинг хэш қиймати A, B, C, D ва E регистрлардаги қийматларни бирлаштириш натижасида ҳосил қилинади.

Қуйида SHA-1 хэш – функцияси алгоритмининг блок схемаси келтирилган:


бошланиш
M
M маълумот – маълумот узунлиги ≡ 448 mod 512 қилиб тўлдирилади. Маълумотга 1 га тенг бўлган битта бит қўшилади, қолган битлар эса ноль билан тўлдирилади. Бунга маълумот узунлигининг 64 битлик қиймати қўшилади.
Кенгайтирилган маълумот 512 битлик блокларга ажратилади ва бу блоклар M=M1M2 ... MN деб белгиланади
A=0×67452301
B=0×EFCDAB89
C=0×98BADCFE
D=0×10325476
E=0×C3D2E1F0
i:=1
4
a:=A
b:=B
c:=C
d:=D
e:=E
M:=M1
M=m0m1 ... m15
mi – 32 бит, i=0, ..., 15
1


1
t:=0
W_t:=M_t
temp:=((a<<<5)+((B∧C)∨((¬B)∧D))+e+5a827999+W_t)mod2^32
e:=d
d:=c
c:=b<<<30
b:=a
a:=temp
t:=t+1
ҳа
t<16
йўқ
temp:=((a<<<5)+((B∧C)∨((¬B)∧D))+e+5a827999+
+(W_t-3⊕W_t-8⊕ W_t-14⊕ W_t-16))mod2^32
e:=d
d:=c
c:=b<<<30
b:=a
a:=temp
t:=t+1
ҳа
t<20
йўқ
2


2
temp:=((a<<<5)+((B⊕C⊕D)+e+6ed89eda1+
+(W_{t-3}⊕W_{t-8}⊕ W_{t-14}⊕ W_{t-16}))mod2^32
e:=d
d:=c
c:=b<<<30
b:=a
a:=temp
t:=t+1
ҳа
t<40
йўк
temp:=((a<<<5)+((B∧C)∨(B∧D)∨(C∧D)+e+8f1bbcdc+
+(W_{t-3}⊕W_{t-8}⊕ W_{t-14}⊕ W_{t-16}))mod2^32
e:=d
d:=c
c:=b<<<30
b:=a
a:=temp
t:=t+1
ҳа
t<60
йўк
3


3
temp:=((a<<<5)+((B⊕C⊕D)+e+ca62c1d6+
+(W_{t-3}⊕W_{t-8}⊕ W_{t-14}⊕ W_{t-16}))mod2^32
e:=d
d:=c
c:=b<<<30
b:=a
a:=temp
t:=t+1
ҳа
t<80
йўқ
A:=(A+a)mod2^32
B:=(B+b)mod2^32
C:=(C+c)mod2^32
D:=(D+d)mod2^32
E:=(E+e)mod2^32
t:=t+1
ҳа
4
i<=N
йўқ
H:=A||B||C||D||E
H
тамом

Хэшлаш функциясининг матнлар тўқнашувини топишга нисбатан бардошлилиги $2^{n/2}$ га тенг. АҚШда калит узунлиги 128, 192 ва 256 бит бўлган янги шифрлаш стандарти ишлаб чиқилганлиги муносабати билан шу даражадаги бардошлиликка эга бўлган янги хэш – функциялар алгоритмларини яратишга эҳтиёж пайдо бўлди. Шу сабабли 2002 йилда АҚШнинг янги хэш – функция стандарти PUB FIPS 180-2 қабул қилинди. Бу стандартда тўртта хэш – функция – SHA-1, SHA-256, SHA-384 ва SHA-512 алгоритмлари келтирилган.

Қуйида SHA-256 хэш – функцияси алгоритми кўриб ўтилади. Бу алгоритмда кирувчи маълумот узунлиги 2^{64} битдан кичик, хэш қиймат узунлиги 256 бит бўлади. Ушбу алгоритмни икки қисмга – сиқиш функцияси ва маълумотни қайта ишлаш алгоритмига бўлиш мумкин. Сиқиш функцияси узунлиги 256 бит бўладиган оралиқ хэш қийматни матннинг навбатдаги блокини калит сифатида олиб шифрлаш алгоритмидан иборат. Сиқиш функциясида олдинги белгилашлардан ташқари қуйидаги белгилашлар ҳам ишлатилади: R^n – сўзни n бит ўнгга суриш, S^n – сўзни n бит ўнгга циклик суриш. Сўзнинг ўлчами 32 битга тенг деб, қўшиш эса mod 2^{32} бўйича олинади. Бошланғич хэшлаш вектори $H^{(0)}$ 8 та 32 разрядлик сўзлардан иборат бўлиб, у қуйидаги туб сонлардан олинган квадрат илдизларнинг каср қисмларига тенг қилиб олинади:

$H^{(0)}$ = {6a09e667, bb67ae85, 3c6ef372, a54ff53a, 510e527f, 9b05688c, 1f83d9ab, 5be0cd19}.

Кейинги ҳисоблашлар қуйидаги схема бўйича олиб борилади:

1. Бошланғич қайта ишлаш. Хэшланувчи маълумот SHA-1 га ўхшаб узунлиги 512 га каррали бўлгунча тўлдирилади. Тўлдиришда маълумотдан кейин 1 ёзилади ва қолган битлар ноль билан тўлдирилади. Бунда маълумот узунлиги 512 модуль бўйича 448 билан таққосланадиган қилиб тўлдирилади. Кейин берилган маълумотнинг 64 битлик узунлиги ёзилади.

2. Маълумотни 512 битлик блокларга ажратиш. Кенгайтирилган маълумот 512 битлик $M(1)$, $M(2)$, ..., $M(N)$ блокларга ажратилади.

3. Асосий цикл. Бу циклни ёзиш учун аргументи ва қийматлари 32 бит бўлган олтита мантиқий функциядан фойдаланилади:

$Ch\ (x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$,

$Maj\ (x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$,

$\Sigma_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x)$,

$\Sigma_1(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x)$,

$\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x)$,

$\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x)$.

$M^{(i)}$ блокни $M^{(i)}=M_0^{(i)}M_1^{(i)}...M_{15}^{(i)}$ 16 та 32 битлик сўзларга ажратилади ва W_0, ..., W_{63} лар қуйидагича аниқланади:

$W_j=M_j^{(i)}, j=0, ..., 15,$

for $j=16$ to 63 {

$W_j=\sigma_1(W_{j-2})+W_{j-7}+\sigma_0(W_{j-15})+W_{j-16}$

}

K_0, ..., K_{63} ўзгармаслар сифатида эса қуйидаги 64 та 16 лик кўринишда тасвирланган туб сонлардан чиқарилган куб илдизлар каср қисмларининг биринчи 32 бити олинади:

428a2f98 71374491 b5c0fbcf e9b5dba5 3956c25b 59f111f1 923f82a4 ab1c5ed5

d807aa98 12835b01 243185be 550c7dc3 72be5d74 80deb1fe 9bdc06a7 c19bf174

e49b69c1 efbe4786 0fc19dc6 240ca1cc 2de92c6f 4a7484aa 5cb0a9dc 76f988da

983e5152 a831c66d b00327c8 bf597fc7 c6e00bf3 d5a79147 06ca6351 14292967

27b70a85 2e1b2138 4d2c6dfc 53380d13 650a7354 766a0abb 81c2c92e 92722c85

a2bfe8a1 a81a664b c24b8b70 c76c51a3 d192e819 d6990624 f40e3585 106aa070

19a4c116 1e376c08 2748774c 34b0bcb5 391c0cb3 4ed8aa4a 5b9cca4f 682e6ff3

748f82ee 78a5636f 84c87814 8cc70208 90befffa a4506ceb bef9a3f7 c67178f2

Асосий цикл қуйидагича бўлади:

for $i=0$ to N {//N – кенгайтирилган маълумотнинг блоклари сони.

//a, b, c, d, e, f, g, h регистрларни хэш-функциянинг (i–1) оралиқ қиймати билан // инициализация қилиш.

$a=H_1^{(i-1)}$; $b=H_2^{(i-1)}$; $c=H_3^{(i-1)}$; $d=H_4^{(i-1)}$; $e=H_5^{(i-1)}$; $f=H_6^{(i-1)}$; $g=H_7^{(i-1)}$; $h=H_8^{(i-1)}$;

//a, b, c, d, e, f, g, h регистрларга сиқиш функциясини қўллаймиз.

for $i=0$ to 63 {//Ch (e, f, g), Maj (a, b, c), $\Sigma_0(a)$, $\Sigma_1(e)$ ва W_j ларни ҳисоблаймиз.

$T_1=h+\Sigma_1(e)+Ch(e, f, g)+K_j+W_j$

$T_2=\Sigma_0(a)+Maj(a, b, c)$

$h=g$; $g=f$; $f=e$; $e=d+T_1$; $d=c$; $c=b$; $b=a$; $a=T_1+T_2$

}

// i – оралиқ хэш қиймат $H^{(i)}$ ни ҳисоблаш.

$H_1^{(i)}=a+H_1^{(i-1)}$; $H_2^{(i)}=b+H_2^{(i-1)}$; $H_3^{(i)}=c=H_3^{(i-1)}$; $H_4^{(i)}=d+H_4^{(i-1)}$;

$H_5^{(i)}=e+H_5^{(i-1)}$; $H_6^{(i)}=f+H_6^{(i-1)}$; $H_7^{(i)}=g+H_7^{(i-1)}$; $H_8^{(i)}=h+H_8^{(i-1)}$

}

// i – бўйича цикл.

Натижада $H^{(N)}=H_1^{(N)}\| H_2^{(N)}\| H_3^{(N)}\| H_4^{(N)}\| H_5^{(N)}\| H_6^{(N)}\| H_7^{(N)}\| H_8^{(N)}\|$ ифода M маълумотнинг хэш қийматини беради.

SHA-512 хэш-функцияси ўзининг тузилишига кўра, SHA-256 хэш–функциясига ўхшайди, лекин унда узунлиги 64 бит бўлган сўзлар устида амаллар бажарилади. Бу алгоритмда кирувчи маълумотнинг узунлиги 2^{128} битдан кичик, хэш қиймат узунлиги 512 бит бўлади. Маълумотнинг узунлиги 1024 га каррали қилиб тўлдирилади. Тўлдиришда маълумот охирига 1 ёзилиб, қолган қисми ноль билан шундай тўлдириладики, маълумот узунлиги 1024 га каррали сондан 128 бит кам бўлиши керак. Охирига берилган маълумотнинг 128 бит узунлиги қўшилади. Шундай қилиб, кенгайтирилган маълумот узунлиги 1024 га каррали бўлади. Бошланғич вектор $H^{(0)}$ 8 та 64 разрядли сўзлардан иборат бўлиб, у қуйидаги туб сонлар квадрат илдизларининг каср қисмларига тенг қилиб олинади:

$H^{(0)}$={6a09e667f3bcc908, bb67ae8584caa73b, 3c6ef372fe94f82b, a54ff53a5f1d36f1, 510e527fade682d1, 9b05688c2b3e6c1f, 1f83d9abfb41bd6b, 5be0cd19137e2179}.

Маълумот 1024 битлик $M^{(1)}$, $M^{(2)}$, ..., $M^{(N)}$ блокларга ажратилади ва улар кетма-кет қайта ишланади.

Асосий цикл худди SHA-256 алгоритмидагидек бўлиб, фақат SHA-512 алгоритмидаги функциялар ва бажариладиган амаллар 64 битлик сўзларда аниқланган ҳамда қўшиш mod 2^{64} бўйича олинади. Сиқиш функцияси эса фақат циклдаги итерациялар сони билан фарқ қилади:

for i=0 to 79 {//Ch (e, f, g), Maj (a, b, c), $\Sigma_0(a)$, $\Sigma_1(e)$ ва W_j ларни ҳисобланади.

$T_1=h+\Sigma_1(e)+Ch(e, f, g)+K_j+W_j$

$T_2=\Sigma_0(a)+Maj(a, b, c)$

$h=g$; $g=f$; $f=e$; $e=d+T_1$; $d=c$; $c=b$; $b=a$; $a=T_1+T_2$

}

Натижада $H^{(N)}=H_1^{(N)}\| H_2^{(N)}\| H_3^{(N)}\| H_4^{(N)}\| H_5^{(N)}\| H_6^{(N)}\| H_7^{(N)}\| H_8^{(N)}\|$ ифода M маълумотнинг хэш қийматини беради.

Мантиқий функциялар эса SHA-256 алгоритмидаги мантиқий функциялардан қуйидагича фарқ қилади:

$\Sigma_0(x)=S^{28}(x)\oplus S^{34}(x)\oplus S^{39}(x)$,

$\Sigma_1(x)=S^{14}(x)\oplus S^{18}(x)\oplus S^{41}(x)$,

$\sigma_0(x)=S^{1}(x)\oplus S^{8}(x)\oplus R^{7}(x)$,

$\sigma_1(x)=S^{19}(x)\oplus S^{61}(x)\oplus R^{6}(x)$.

$M^{(i)}$ блокни $M^{(i)} = M_0^{(i)} M_1^{(i)} ... M_{15}^{(i)}$ 16 та 64 битлик сўзларга ажратилади ва W_0, ..., W_{79} лар қуйидагича аниқланади:

$W_j = M_j^{(i)}$, $j = 0, ..., 15$,

for $j = 16$ to 79 {

$W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$

}

K_0, ..., K_{63} ўзгармаслар сифатида эса қуйидаги 80 та 16 лик кўринишда тасвирланган туб сонлардан чиқарилган куб илдизлар каср қисмларининг биринчи 64 бити олинади:

428a2f98d728ae22	7137449123ef65cd	b5c0fbcfec4d3b2f	e9b5dba58189dbbc
3956c25bf348b538	59f111f1b605d019	923f82a4af194f9b	ab1c5ed5da6d8118
d807aa98a3030242	12835b0145706fbe	243185be4ee4b28c	550c7dc3d5ffb4e2
72be5d74f27b896f	80deb1fe3b1696b1	9bdc06a725c71235	c19bf174cf692694
e49b69c19ef14ad2	efbe4786384f25e3	0fc19dc68b8cd5b5	240ca1cc77ac9c65
2de92c6f592b0275	4a7484aa6ea6e483	5cb0a9dcbd41fbd4	76f988da831153b5
983e5152ee66dfab	a831c66d2db43210	b00327c898fb213f	bf597fc7beef0ee4
c6e00bf33da88fc2	d5a79147930aa725	06ca6351e003826f	142929670a0e6e70
27b70a8546d22ffc	2e1b21385c26c926	4d2c6dfc5ac42aed	53380d139d95b3df
650a73548baf63de	766a0abb3c77b2a8	81c2c92e47edaee6	92722c851482353b
a2bfe8a14cf10364	a81a664bbc423001	c24b8b70d0f89791	c76c51a30654be30
d192e819d6ef5218	d69906245565a910	f40e35855771202a	106aa07032bbd1b8
19a4c116b8d2d0c8	1e376c085141ab53	2748774cdf8eeb99	34b0bcb5e19b48a8
391c0cb3c5c95a63	4ed8aa4ae3418acb	5b9cca4f7763e373	682e6ff3d6b2b8a3
748f82ee5defb2fc	78a5636f43172f60	84c87814a1f0ab72	8cc702081a6439ec
90befffa23631e28	a4506cebde82bde9	bef9a3f7b2c67915	c67178f2e372532b
ca273eceea26619c	d186b8c721c0c207	eada7dd6cde0eb1e	f57d4f7fee6ed178
06f067aa72176fba	0a637dc5a2c898a6	113f9804bef90dae	1b710b35131c471b
28db77f523047d84	32caab7b40c72493	3c9ebe0a15c9bebc	431d67c49c100d4c
4cc5d4becb3e42b6	597f299cfc657e2a	5fcb6fab3ad6faec	6c44198c4a475817.

SHA-384 хэш–функцияси алгоритми SHA-512 алгоритмидан фақат бошланғич вектори:

$H^{(0)}$ = {cbbb9d5dc1059ed8, 629a292a367cd507, 9159015a3070dd17, 152fecd8f70e5939, 67332667ffc00b31, 8eb44a8768581511, db0c2e0d64f98fa7, 47b5481dbefa4fa4} билан фарқ қилади. Бу алгоритмда кирувчи маълумотнинг узунлиги 2^{128} битдан кичик бўлиб, хэш

қиймат узунлиги 384 бит бўлади. Бошқа ҳамма ҳисоблашлар SHA-512 алгоритми билан бир хил бўлади. Натижада, чиқувчи хэш қиймат сифатида:

$H^{(N)}=H_1^{(N)}\| H_2^{(N)}\| H_3^{(N)}\| H_4^{(N)}\| H_5^{(N)}\| H_6^{(N)}\| H_7^{(N)}\| H_8^{(N)}$ нинг чап томондан 384 бити, яъни $H^{(N)}=H_1^{(N)}\| H_2^{(N)}\| H_3^{(N)}\| H_4^{(N)}\| H_5^{(N)}\| H_6^{(N)}$ олинади.

§7.6. СТБ 1176.1 – 99 хэш – функцияси алгоритми

Беларусь Республикасининг «Ахборот технологияси. Ахборотни муҳофаза қилиш. Хэш – функция» Давлат стандарти 1999 йил 30 сентябрда қабул қилинган [20]. Стандартда аниқланган h хэшлаш функцияси байтлар кетма-кетлигига таъсир қилади ва хэш қиймат узунлиги L бўлган ҳамда $142\le L\le 256$ диапазонда ётадиган иккилик сўз бўлади. 256 битлик бошланғич хэш қийматдан фойдаланилиб, у барча фойдаланувчилар гуруҳи учун ягона қилиб ихтиёрий равишда танланади. Қуйида h ни ҳисоблаш алгоритми келтирилади.

1. Белгилашлар.

Стандартдаги алмаштиришлар $Z(n)=\{0, 1, ..., 2^n-1\}$тўпламдаги сонлар устида бажарилади. $a\in Z(n)$ сонга a нинг иккилик кўринишида ёзилган V_n сон мос қўйилади ва $Z(n)$ га $\oplus$ – XOR, $\boxplus$ – mod 2^{32} бўйича қўшиш, $<<<$ – циклик суриш амаллари бажарилади. a соннинг 2^m асос бўйича ёйилмаси $a=\sum_{i=0}^{k-1} a_i(2^m)^i$ $a_i\in Z(m)$ ни $a=a_{k-1}\|...\|a_1\|a_0$ кўринишида ёзилади.

2. Алмаштиришлар .

h нинг қийматини ҳисоблашда қуйидаги ёрдамчи алмаштиришлардан фойдаланилади:

1. $\rho_0, \rho_1, \rho_1, \rho_3$: $Z(512)\rho\to Z(512)$ алмаштиришлар $X=x_{15}\|x_{14}\|$ ... $\|x_1\|x_0$, $x_i\in Z(32)$ сонга қуйидаги қоида бўйича таъсир қилади:

$\rho_0(x)=((x_{15}\oplus x_{13}\oplus x_3\oplus x_0)\boxplus C_0)\|x_{15}\|...\|x_1$;

$\rho_1(x)=((x_{15}\oplus x_2\oplus x_0)\boxplus C_1)\|x_{15}\|...\|x_1$;

$\rho_2(x)=((x_9\oplus x_4\oplus x_0)\boxplus C_2)\|x_{15}\|...\|x_1$;

$\rho_3(x)=((x_{13}\oplus x_8\oplus x_0)\boxplus C_3)\|x_{15}\|...\|x_1$;

Бу ерда: $C_0=0\times 2BDA732E$, $C_1=0\times 3920FE85$, $C_2=0\times BC1641F9$, $C_3=0\times 75FE243B$.

2. Стандартда S блоклардан фойдаланилади $s:Z(8)\to Z(8), j\to s_j$, бу ерда у $s_{255}\|...\|s_1\|s_0\in Z(2048)$ сонлар билан берилади. $X\in Z(128)$ соннинг ω алмаштириши Y_1, ..., Y_8 сонлар билан 8 та S блокни аниқлайди. $\omega(X; Y_1, ..., Y_8)$ алмаштириш Фейстел ўрнига қўйишларининг 32 карра таъсири натижаси бўлади: $X\to X_0\|(X_1\oplus f(X_0))$, $X=X_1\|X_0$, $X_i\in Z$ (64). Бу ерда ишлатилаётган $f: Z(64)\to Z(64)$ такт функцияси қуйидаги қоида

бўйича таъсир қилади: $x_8 \| ... \| x_1 \mapsto (s_8(x_8) \| ... \| s_1(x_1)) <<< 3$, $x_i \in Z(8)$. Бу ердаги $s_i : Z(8) \to Z(8)$ Y_i сон билан аниқланадиган S блок.

3. $\xi : Z(256) \to Z(256)$ алмаштириш $X = x_7 \| ... \| x_1 \| x_0$, $x_i \in Z(32)$ сонга $\xi(X) = x_6 \| ... \| x_0 \| (x_0 \oplus x_2 \oplus x_4 \oplus x_7)$ сонни мос қўяди.

4. $\varphi : Z(128) \times Z(256)$ акслантириш $X = x_3 \| ... \| x_0$ ва $Y = y_7 \| ... \| y_0$, x_i ва $y_i \in Z(32)$, сонларга $\varphi(X, Y) = y_3 \| ... \| y_0 \| ((y_3 \boxplus x_3) \oplus y_7) \| ... \| ((y_0 \boxplus x_0) \oplus y_4)$ (сонни мос қўяди.

3. Берилганлар.

Хэшланиши керак бўлган M маълумот $m_1, ..., m_l \in Z(8)$ байтлар кетма-кетлигига ажратилади. Маълумот $n = (1+k)/32$ бутун сон бўладиган қилиб минимал сондаги m_{l+1}, ... m_{l+k} нол байтлар билан тўлдирилади. Кейин куйидаги сонлар аниқланади: $M_i = m_{32i} \| ... \| m_{32(i-1)+2} \| m_{32(i-1)+1}$, $i = \overline{1, n}$ $M_{n+1} = l$.

4. Ўзгарувчилар.

Хэш қийматни ҳисоблашда қуйидаги ўзгарувчилардан фойдаланилади:

1. $H \in Z(256)$, $H = H_1 \| H_0 = h_7 \| ... \| h_0$ ўзгарувчи. Бу ерда $H_i \in Z(128)$, $h_i \in Z(32)$. Ҳисоблаш бошланишида у бошланғич хэш қийматдан, алгоритмнинг бажарилиши жараёнида у жорий хэш қийматдан иборат.

2. $T_0, T_1, ..., T_7 \in Z(2048)$ ўзгарув чилар. T_1, T_3, T_5, T_7 ўзгарувчиларнинг қийматлари алгоритмнинг бажарилиши жараёнида ўзгармайди ва 7.2 жадвалда $t_{ij} \in Z(32)$ сонлар берилган бўлиб, $T_i = t_{i,63} \| ... \| t_{i0}$ кўринишида аниқланади. T_0, T_2, T_4, T_6 ўзгарувчилар $r_{ij} \in Z(512)$ сонлар билан $T_i = r_{i3} \| r_{i2} \| r_{i1} \| r_{i0}$ кўринишида аниқланади.

7.2 -жадвал

$t_{i,j}$, $i = 1,3,5,7$ нинг қийматлари (ўн олтилик саноқ тизимида)

j	$i=1$	$i=3$	$i=5$	$i=7$
0	AA2AA82E	4DCDCF4F	5557455D	B2B03212
1	8A0A088E	69E9EB6B	4715547C	921A13BF
2	A222A026	65E5E767	5644465C	BAB83A18
3	82020086	41C1C343	1416177D	9A101BB7
4	AE2CAC28	49C9CB4B	51534159	B6B43616
5	8C0E0C88	6DEDEF6F	43115048	961E17BB

6	A624A420	61E1E363	7072607A	BEBC1C3E
7	84061E9A	45C5C747	F8FAF9D1	9E143F11
8	AB2BA92F	5DDDCC4C	5F4D4F05	B3B13303
9	8B0B098F	79F9E868	1D1F5EF4	931902AE
10	A323A127	F17140C0	585A4852	BDAF3715
11	B32133A7	D55564E4	4A185BF1	9F0706AA
12	8D0F9F1B	51D1C242	494B1913	B5A73505
13	AF2D3FBB	75F5EC6C	1B091AF0	970E04B9
14	07870583	FD7D4ECE	7E6C6E74	ABA90131
15	17859D19	D95960EA	F6E4F50D	9B0A3057
16	BA3AB83E	5CDCDE5E	7577657F	A2A02200
17	98381ABE	6AF8FA78	6735764C	900853AD
18	B231B025	66E6E062	64626668	A8FA380C
19	806312A5	48C88C0C	3430710F	98520BA3
20	AD3DBF29	7AA8AA28	7332616A	A6A43455
21	89B94B6F	8E0E4ACA	63313840	943C1F82
22	BD3CEF6A	53D3C444	420710D8	ACFE5623
23	CFBC1532	FF7F6EEE	DADCD5FB	9C263D24
24	EB3BE96D	1C9CDF5F	FDFFEDF7	A1217391
25	9B396BCD	2AB8FB7B	EFBDDE36	839D502A
26	B130E336	E2700181	E8FCEAE6	FC2E7680
27	E11003B6	9E0D7CAD	B8FEDD33	8ED2473B
28	9E1C9C18	72F48357	E9BAEBA8	954599F6
29	EACA4A6E	46D0A92D	D9DBF212	A58A2C74
30	4EDF4C99	8D5B0FDB	4E0B2408	885A288F
31	5E7FB481	F3747E91	B4E0C83C	F8FF092F
32	E868FA6C	54C6D456	D7C5C726	F2F02042
33	C8485ACE	20F0F222	9597D03A	D05851FD
34	E262E034	2CFC1EFE	D4D6C41C	EAE87A4A
35	92424096	58D838BA	C694D379	DA405BF7

320

36	EE7EEC7A	50D28052	C1C3917B	F4F5668B
37	DD0D1FCB	30A0A232	9381963E	861D890F
38	E666E460	3A825ADA	E2B0F369	7E2DD8FB
39	971D4F93	08889092	282A2C84	5E87F984
40	FB697BED	04D6D705	DFCDCFE1	F3F17143
41	C949DB4D	73A1ACAE	CCD2B99D	8159D3EE
42	F36137F1	BC0AAB1D	AA9FB28D	EFE7E58D
43	0173B771	8F2F842E	1	EC7CDFDB
44	5B5C0459	188A8909	C9CB9901	DDE62754
45	CC5F5D16	29B9F63D	CA86BF9B	850D4625
46	47945291	A406F721	CE9C87A9	3972776E
47	35FEF813	853F8677	6D03ABC0	78E2416C
48	F272787C	07879515	E7B5B76F	E06A6444
49	907058FC	23BBA32B	A5A7A00A	C248D9EB
50	147946B5	A5AF1A98	B6A29E2E	685FCB2B
51	DAD9C267	009A25BD	A4A6E38C	CAC95D29
52	E765E511	31B19303	B18EB389	E44FC870
53	C0F0437D	0B19A7B5	A1A3EC04	7BEDE975
54	F764C344	3C3E14BF	C2EE8B98	4D7F7DC0
55	C6FFF9C4	B33927B7	9A880020	8C67DED7
56	F5775654	1F117624	372527AD	E36365D5
57	9545FD75	26948B35	8A6B223D	D1C76FE1
58	C1D8D376	991B9B97	02061E90	C379625C
59	C77441DE	9D34963B	BC0C0E23	4ED6606D
60	D25351D0	3337B0BE	BE928FAC	CFDC6B4B
61	C5D7D5F6	02101716	80AF3F83	69CDD4CE
62	50D6D157	12A69FB4	39AE823B	C649614C
63	D455DCF4	B61336B2	2D292F2B	C5CCC1C4

3. $V \in Z(512)$, $V = v_{15} \| ... \| v_1 \| v_0$, $v_i \in Z(32)$ ўзгарувчи. v_i нинг бошланғич қийматлари 7.3-жадвалда келтирилган.

v_i нинг бошланғич қийматлари (ўн олтилик саноқ тизимида)

i	v_i	7	0D817489	15	0B1294AC
0	D1845AC6	8	87D45A6F		
1	AC3D25C6	9	3D5721C6		
2	F467247D	10	573714C8		
3	079294AB	11	078274DB		
4	F19A24CD	12	2A8A1A76		
5	B47D25C6	13	DC6715C6		
6	D4522491	14	B4F1257D		

4. $K \in Z(256)$, $K = K_1 \| K_0 = k_7 \| \ldots \| k_0$ ўзгарувчи. Бу ерда $K \in Z(128)$, $k_i \in Z(32)$.

5. $W \in Z(256)$, $W = W_1 \| W_0$ ўзгарувчи. Бу ерда $W_i \in Z(128)$.

5. Алгоритм.

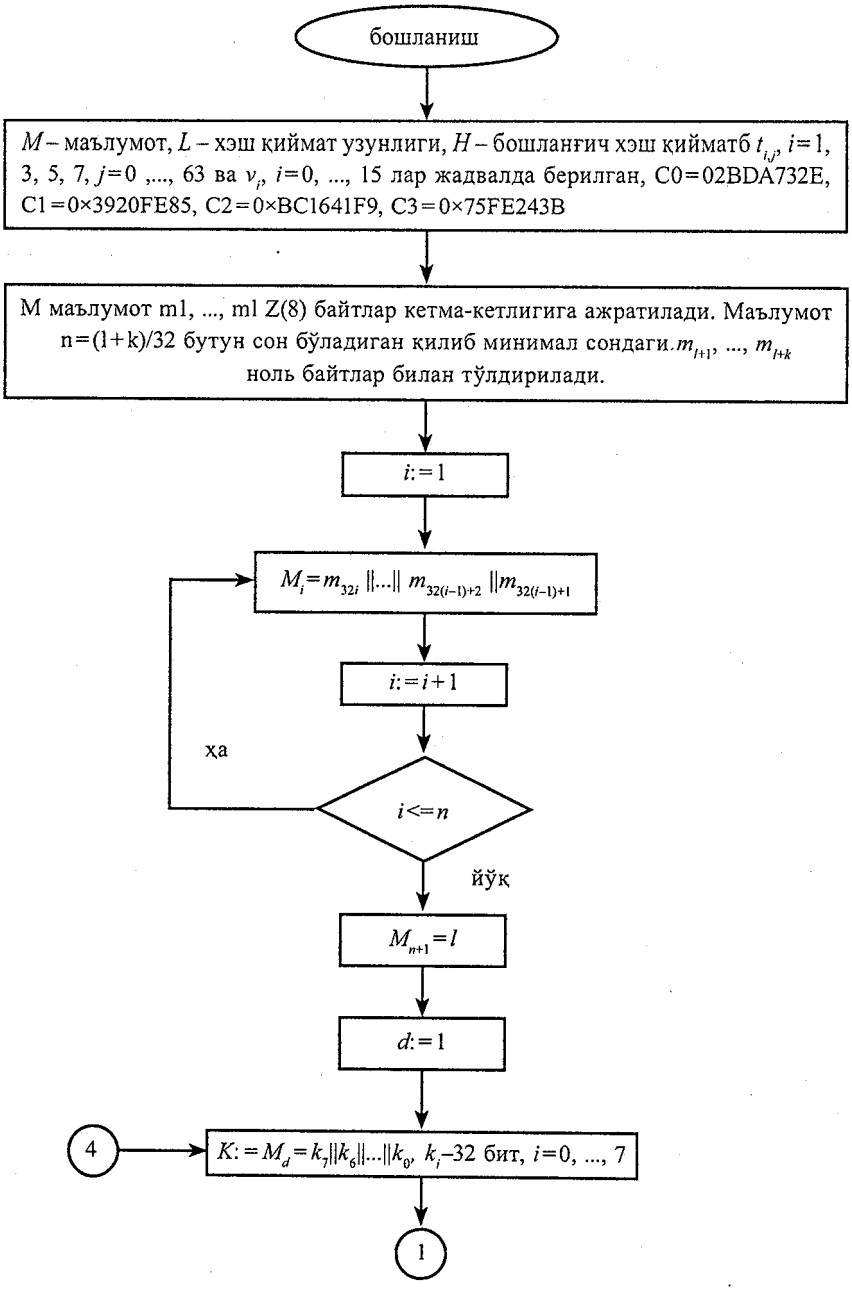
$h(M)$ хэш қийматни ҳисоблаш алгоритми бир неча қадамдан иборат:

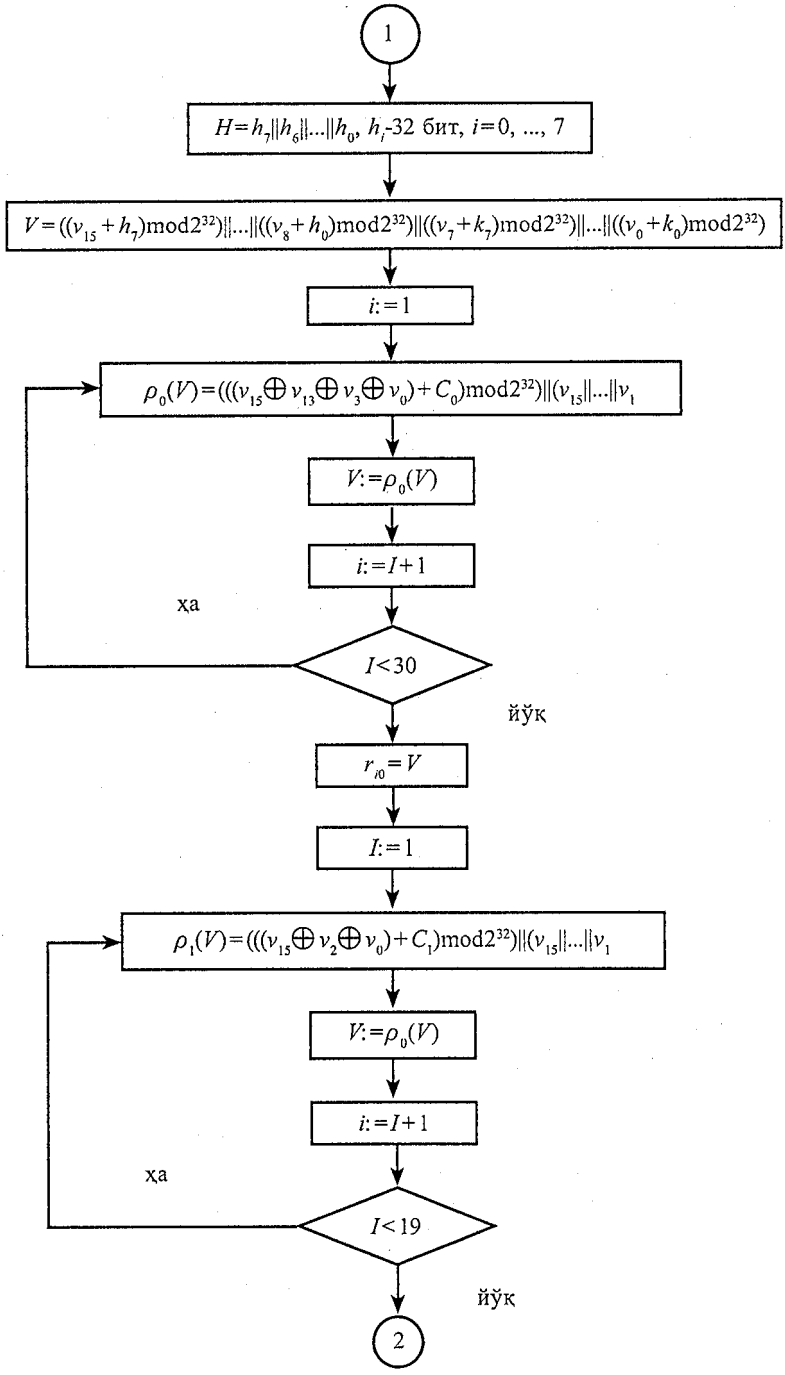
1. $d \leftarrow 1$
2. $K \leftarrow M_d$.
3. $V \leftarrow (v_{15} \boxplus h_7) \| \ldots \| (v_8 \boxplus h_0) \| (v_7 \boxplus k_7) \| \ldots \| (v_0 \boxplus k_0)$.
4. $i = 0, 2, 4, 6$ учун қуйидагиларни бажарамиз:

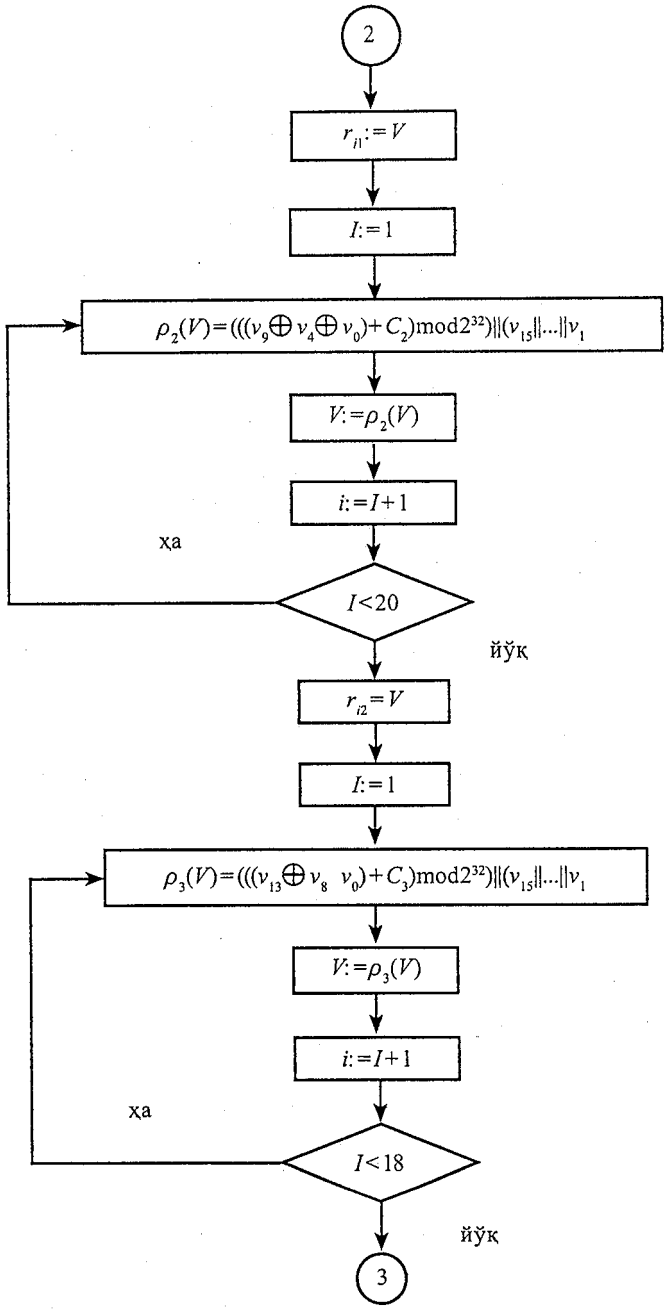
а) $V \leftarrow \rho_0^{29}(V)$, $r_{i0} \leftarrow V$;
б) $V \leftarrow \rho_1^{18}(V)$, $r_{i1} \leftarrow V$;
в) $V \leftarrow \rho_2^{19}(V)$, $r_{i2} \leftarrow V$;
г) $V \leftarrow \rho_3^{17}(V)$, $r_{i3} \leftarrow V$;

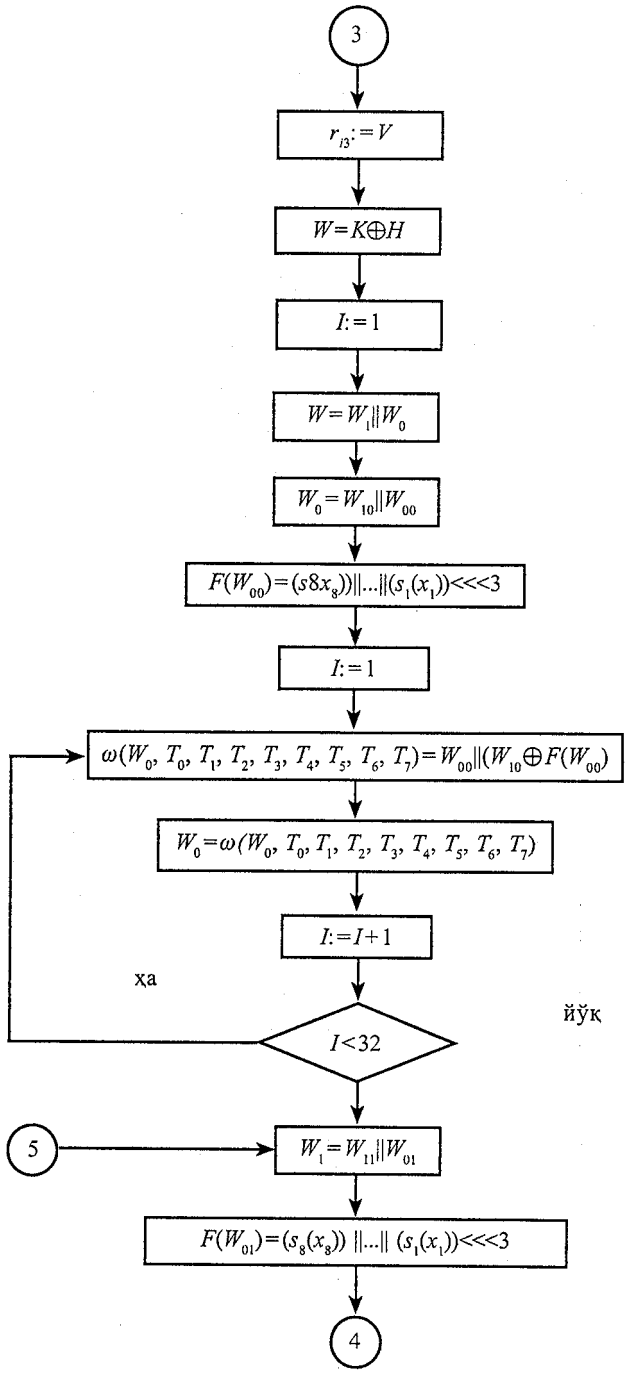
5. $W = K \oplus H$.
6. $W_0 \leftarrow (W_0, T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7)$.
7. $W_1 \leftarrow (W_1, T_4, T_1, T_0, T_3, T_6, T_5, T_2, T_7)$.
8. $W \leftarrow \xi^{31}(W)$.
9. $W \leftarrow \varphi(H_0, \varphi(H_0, W))$.
10. $W \leftarrow \varphi(K_0, \varphi(K_0, W))$.
11. $W \leftarrow \varphi(H_1, \varphi(H_1, W))$.
12. $W \leftarrow \varphi(K_1, \varphi(K_1, W))$.
13. $H \leftarrow W$.
14. $d \leftarrow d + 1$.
15. Агар $d < n + 2$ бўлса, у ҳолда 2-қадамга қайтамиз. Акс ҳолда кейинги қадамга ўтилади.
16. $H \bmod 2^L$ ни ҳисобланади.

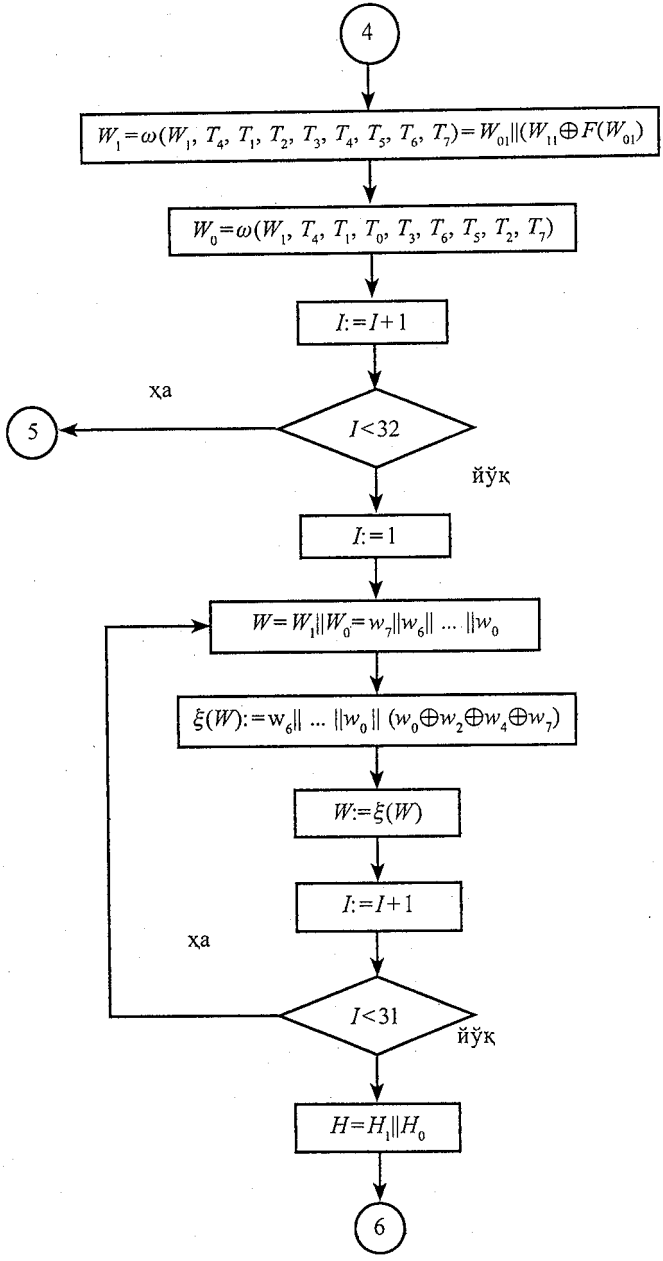
Қуйида СТБ 1176.1–99 хэш-функцияси алгоритмининг блок схемаси келтирилган:

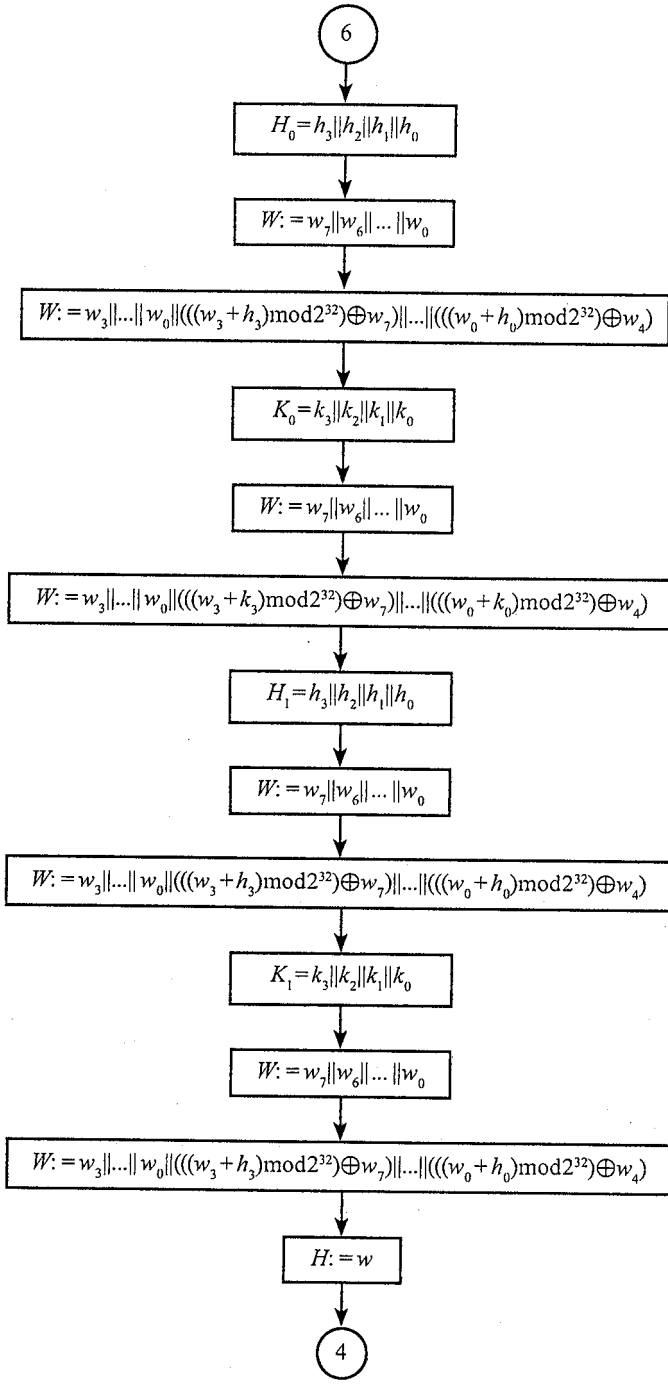

бошланиш
M – маълумот, L – хэш қиймат узунлиги, H – бошланғич хэш қийматб $t_{i,j}$, i=1, 3, 5, 7, j=0 ,..., 63 ва v_i, i=0, ..., 15 лар жадвалда берилган, C0=02BDA732E, C1=0×3920FE85, C2=0×BC1641F9, C3=0×75FE243B
M маълумот m1, ..., ml Z(8) байтлар кетма-кетлигига ажратилади. Маълумот n=(l+k)/32 бутун сон бўладиган қилиб минимал сондаги m_{l+1}, ..., m_{l+k} ноль байтлар билан тўлдирилади.
i:=1
$M_i=m_{32i}\|...\|m_{32(i-1)+2}\|m_{32(i-1)+1}$
i:=i+1
ҳа
i<=n
йўқ
$M_{n+1}=l$
d:=1
4
$K:=M_d=k_7\|k_6\|...\|k_0$, k_i–32 бит, i=0, ..., 7
1

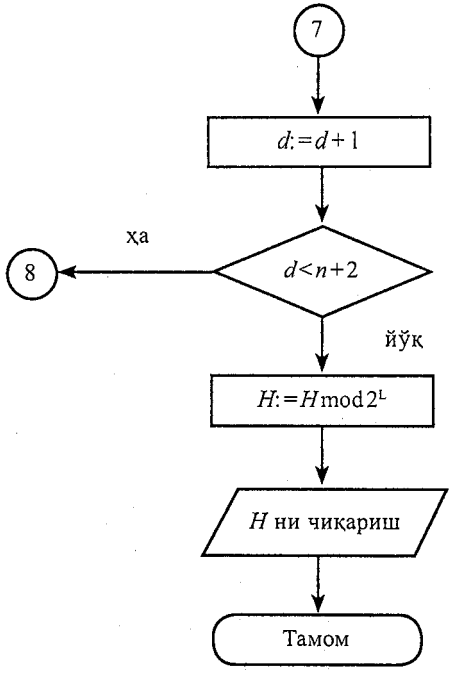

1
$H=h_7\|h_6\|...\|h_0$, h_i-32 бит, $i=0, ..., 7$
$V=((v_{15}+h_7)\bmod 2^{32})\|...\|((v_8+h_0)\bmod 2^{32})\|((v_7+k_7)\bmod 2^{32})\|...\|((v_0+k_0)\bmod 2^{32})$
$i:=1$
$\rho_0(V)=(((v_{15}\oplus v_{13}\oplus v_3\oplus v_0)+C_0)\bmod 2^{32})\|(v_{15}\|...\|v_1$
$V:=\rho_0(V)$
$i:=I+1$
ҳа
$I<30$
йўқ
$r_{i0}=V$
$I:=1$
$\rho_1(V)=(((v_{15}\oplus v_2\oplus v_0)+C_1)\bmod 2^{32})\|(v_{15}\|...\|v_1$
$V:=\rho_0(V)$
$i:=I+1$
ҳа
$I<19$
йўқ
2


2
$r_{i1} := V$
$I := 1$
$\rho_2(V) = (((v_9 \oplus v_4 \oplus v_0) + C_2) \bmod 2^{32}) \| (v_{15} \| ... \| v_1$
$V := \rho_2(V)$
$i := I + 1$
ҳа
$I < 20$
йўқ
$r_{i2} = V$
$I := 1$
$\rho_3(V) = (((v_{13} \oplus v_8 \ v_0) + C_3) \bmod 2^{32}) \| (v_{15} \| ... \| v_1$
$V := \rho_3(V)$
$i := I + 1$
ҳа
$I < 18$
йўқ
3


3
$r_{i3} := V$
$W = K \oplus H$
$I := 1$
$W = W_1 \| W_0$
$W_0 = W_{10} \| W_{00}$
$F(W_{00}) = (s8x_8)) \| ... \| (s_1(x_1)) <<< 3$
$I := 1$
$\omega(W_0, T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7) = W_{00} \| (W_{10} \oplus F(W_{00})$
$W_0 = \omega(W_0, T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7)$
$I := I + 1$
ха
йўк
$I < 32$
5
$W_1 = W_{11} \| W_{01}$
$F(W_{01}) = (s_8(x_8)) \| ... \| (s_1(x_1)) <<< 3$
4


4
$W_1=\omega(W_1, T_4, T_1, T_2, T_3, T_4, T_5, T_6, T_7)=W_{01}\|(W_{11}\oplus F(W_{01})$
$W_0=\omega(W_1, T_4, T_1, T_0, T_3, T_6, T_5, T_2, T_7)$
$I:=I+1$
ҳа
5
$I<32$
йўқ
$I:=1$
$W=W_1\|W_0=w_7\|w_6\| \dots \|w_0$
$\xi(W):=w_6\| \dots \|w_0\| (w_0\oplus w_2\oplus w_4\oplus w_7)$
$W:=\xi(W)$
$I:=I+1$
ҳа
$I<31$
йўқ
$H=H_1\|H_0$
6


6
$H_0 = h_3 || h_2 || h_1 || h_0$
$W := w_7 || w_6 || \ldots || w_0$
$W := w_3 || \ldots || w_0 || (((w_3 + h_3) \bmod 2^{32}) \oplus w_7) || \ldots || (((w_0 + h_0) \bmod 2^{32}) \oplus w_4)$
$K_0 = k_3 || k_2 || k_1 || k_0$
$W := w_7 || w_6 || \ldots || w_0$
$W := w_3 || \ldots || w_0 || (((w_3 + k_3) \bmod 2^{32}) \oplus w_7) || \ldots || (((w_0 + k_0) \bmod 2^{32}) \oplus w_4)$
$H_1 = h_3 || h_2 || h_1 || h_0$
$W := w_7 || w_6 || \ldots || w_0$
$W := w_3 || \ldots || w_0 || (((w_3 + h_3) \bmod 2^{32}) \oplus w_7) || \ldots || (((w_0 + h_0) \bmod 2^{32}) \oplus w_4)$
$K_1 = k_3 || k_2 || k_1 || k_0$
$W := w_7 || w_6 || \ldots || w_0$
$W := w_3 || \ldots || w_0 || (((w_3 + h_3) \bmod 2^{32}) \oplus w_7) || \ldots || (((w_0 + h_0) \bmod 2^{32}) \oplus w_4)$
$H := w$
4



Таъкидлаш жоизки, Интернет тармоғи ва илмий манбаларда АҚШ стандарт хэш-функциясининг коллизияга бардошсизлиги ҳақида Хитойлик ҳамда бошқа криптотаҳлилчиларнинг кўплаб асосли маълумотлари берилган.

§7.7. O‘z DSt 1106 : 2006 хэш-функцияси акслантиришларининг мураккаблик даражаларини баҳолаш

Ўзбекистон Республикасининг O‘z DSt 1106 : 2006 хэш-функцияси 2006 йилда бир йил синов муддати билан қабул қилинган калитли хэш-функция бўлиб, унда калит узунлиги 128 бит ёки 256 бит бўлиши назарда тутилган. Чиқувчи хэш қиймат узунлиги ҳам мос равишда 128 бит ёки 256 бит бўлади [25].

Ушбу стандарт ихтиёрий узунликдаги матн учун хэш-функцияни ҳисоблаш алгоритми ва кетма-кетлигини аниқлаб, ахборотларни криптографик усуллар асосида қайта ишлаш ва ҳимоялашда, шу билан бирга ахборот-коммуникация тизимларида маълумотларни узатиш, қайта ишлаш ва сақлашда, ЭРИ жараёнини таъминлашда қўллашга мўлжалланган.

O‘z DSt 1106 : 2006 хэш-функция стандарти параметрлар алгебраси асосида қурилган бўлиб, параметрлар алгебрасининг кўпайтириш, даражага кўтариш, тескарилаш амалларидан фойдаланилади:

1) a ва b сонларни R – коэффициент асосида модуль p – бўйича кўпайтириш формуласи:

$$a \oplus b = a + (1 + R \cdot a) \pmod{p}.$$

2) a сонини R – коэффицент бўйича бирор x даражага кўтариш формуласи (R ва n лар ўзаро туб):

$$a^{\backslash x} = ((1 + R \cdot a)^{\backslash x} - 1) \cdot R^{-1} \pmod{n}.$$

3) a сонини R – коэффицент асосида модуль n – бўйича тескарисини топиш формуласи:

$$a^{\backslash -1} = a \cdot ((1 + R \cdot a)^{-1} \pmod{n}.$$

Шунингдек, мазкур стандартда циклик силжитишлардан ҳам кенг фойдаланилган.

Хэш-функция алгоритмида 128 бит узунликдаги блоклар устида амал бажарилганда базавий бирлик сифатида ярим байт («полубайт») – 4 битлик кетма-кетликдан фойдаланилган, 256 бит узунликдаги блоклар устида амал бажарилганда байтлар устида амаллар бажарилади.

Алгоритмда 128 битлик блоклар учун босқичлар сони $b+10$, 256 битлик блоклар учун эса босқичлар сони $b+6$ қилиб белгиланган, бу ерда, b – блоклар сони.

Хэш-функция алгоритми кетма-кет бажарилувчи 3 та қисмдан иборат бўлиб, биринчи қисмда, фақат кирувчи блоклар устида амаллар бажарилади, иккинчи қисмда биринчи қисмнинг охирги блок натижаси устида 10(6) босқич давомида акслантиришлар амалга оширилади, учинчи қисм эса, иккита акслантиришдан иборат.

Алгоритм иккита режимда ишлашга мўлжалланган бўлиб, 0-режимда биринчи қисмнинг ҳар бир босқичи **Aralash ( ), Daraja (holat,R), SurKalit ( ), SurHolat ( ), Teskari (k_e, R), Qo’shBosqichKalit( ), TuzilmaKalit (k_e, R)** акслантиришлар кетма-кетлигидан, 1-режимда эса, биринчи қисмнинг ҳар бир босқичи **Aralash ( ), Daraja (holat, R), Daraja (k_e,R), SurKalit ( ),**

SurHolat( ), Teskari (holat, R), Teskari (k_e, R), Qo’shBosqichKalit( ), TuzilmaKalit(k_e, R) акслантиришлар кетма-кетлигидан иборат [31].

§ 7.8. Калитсиз хэш-функция алгоритмини яратишга мисол

Ушбу хэш-функция ахборотнинг тўлалигини текширишни криптографик усулда амалга ошириш учун; электрон рақамли имзо генерацияси алгоритмига махфий калит билан биргаликда бошланғич кирувчи қиймат ҳисобланади. Таклиф қилинаётган хэш-функция алгоритмига кирувчи блок узунлиги 256 бит бўлган функция бўлиб, чиқувчи хэш қиймат блоки узунлиги 256 бит бўлади.

Хэш қийматни ҳисоблаш жараёни қуйидаги босқичлардан иборат:

1-босқич. Тўлдириш битларини қўшиш.

Хэшланадиган маълумот узунлиги ихтиёрий бўлиб, маълумот узунлиги 256 бит бўлган блокларга ажратилади. Агар охирги блок узунлиги 256 битдан кичик бўлса, 256 битгача ноль билан тўлдирилади.

2-босқич. Маълумот узунлигини қўшиш.

1-босқич натижасига берилган маълумот узунлигининг 256 битлик қиймати бириктирилади. Битлар сони билан аниқланадиган маълумот узунлигини билдирувчи блок хэшланадиган маълумотнинг узунлиги mod 2^{256} бўйича ҳисобланиб ҳосил қилинади.

3-босқич. Назорат йиғиндисини қўшиш.

2-босқич натижасига берилган маълумотнинг назорат йиғиндисини билдирувчи 256 битлик блок бириктирилади. Назорат йиғиндисини билдирувчи блок, охирги тўлиқ бўлмаган блок ноль билан тўлдирилгандан кейин, барча блоклар қийматларининг ўнлик саноқ тизимидаги йиғиндиси mod 2^{256} бўйича ҳисобланиб ҳосил қилинади.

4-босқич. М аълумотни 256 битлик блокларга ажратиб қайта ишлаш.

Ушбу 3 та босқич бажарилгандан кейин кенгайтирилган маълумот 256 битли блокларга ажратилади. Бу блоклар сони N га тенг бўлсин. Ушбу блоклар M_1, M_2, ..., M_N деб белгиланади. Маълумотни 256 битли блокларга ажратиб қайта ишлаш қуйидагича амалга оширилади:

1. $S=S_0$, ..., S_{255} маълум сонлар блоки киритилади, бу ерда $0 \le S_l \le 255$, $l=\overline{0,255}$ (бу S блок фойдаланувчилар гуруҳи учун умумий бўлади). $H=0$ ва $n=1$ деб олинади.

2. M_n – хэшланувчи маълумот блоки киритилади:

$$T(0)=M_n=t_0(0)\ t_1(1)\ \ldots\ t_{255}(0)\ (256\ \text{бит})$$

3. $T(0)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_0 = t_0(0)\ t_1(0)\ \ldots\ t_7(0)\ k_1 = t_8(0)\ t_9(0)\ \ldots\ t_{15}(0),\ \ldots,$$
$$k_{31} = t_{248}(0)\ t_{249}(0)\ \ldots\ t_{255}(0).$$

$T(0)$ блок чапга 29 бит циклик сурилади ва у орқали белгиланади:

$$T(1) = T(0) <<< 29 = t_{29}(0)\ t_{30}(0)\ \ldots\ t_{255}(0)\ t_0(0)\ \ldots\ t_{28}(0) =$$
$$= t_0(1)\ t_1(1)\ \ldots\ t_{255}(1)$$

$T(1)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_{0+32\cdot 1} = t_0(1)\ t_1(1)\ \ldots\ t_7(1)\ k_{1+32\cdot 1} = t_8(1)\ t_9(1),\ \ldots,\ t_{31+32\cdot 1} = t_{248}(1)\ t_{249}(1)\ \ldots\ t_{255}(1).$$

$T(1)$ блок чапга 29 бит циклик сурилади ва у $T(2)$ орқали белгиланади:

$$T(2) = T(1) <<< 29 = t_{29}(1)\ t_{30}(1)\ \ldots\ t_{255}(1)\ t_0(1)\ \ldots\ t_{28}(1) =$$
$$= t_0(2)\ t_1(2)\ \ldots\ t_{255}(2)$$

$T(2)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_{0+32\cdot 2} = t_0(2)\ t_1(2)\ \ldots\ t_7(2)\ k_{1+32\cdot 1} = t_8(2)\ t_9(2)\ \ldots\ t_{15}(2),\ \ldots,\ k_{31+32\cdot 2} =$$
$$= t_{248}(2) t_{249}(2)\ \ldots\ t_{255}(2).$$

Юқоридагилар каби $T(3)$, $T(4)$, $T(5)$, $T(6)$ ларнинг ифодасидан қуйидаги калитлар ҳосил қилинади:

$$k_{0+32\cdot 3},\ \ldots,\ k_{31+32\cdot 3},\ k_{0+32\cdot 4},\ \ldots,\ k_{31+32\cdot 4},\ k_{0+32\cdot 5},\ \ldots,\ k_{31+32\cdot 5},\ k_{0+32\cdot 6},\ \ldots,\ k_{31+32\cdot 6}.$$

$T(6)$ блок чапга 29 бит циклик сурилади ва у $T(7)$ орқали белгиланади:

$$T(7) = T(6) <<< 29 = t_{29}(6)\ t_{30}(6)\ \ldots\ t_{255}(6)\ t_0(6) \ldots\ t_{28}(6) = t_0(7)\ t_1(7) \ldots\ t_{255}(7).$$

$T(7)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_{0+32\cdot 7} = t_0(7)\ t_1(7)\ \ldots\ t_7(7)\ k_{1+32\cdot 1} = t_8(7)\ t_9(7)\ \ldots\ t_{15}(7),\ \ldots,\ k_{31+32\cdot 2} =$$
$$= t_{248}(7) t_{249}(7)\ \ldots\ t_{255}(7).$$

4. Бу ҳосил қилиб олинган бошланғич калитлар ва S блок ёрдамида қуйида келтирилган мураккаб бўлмаган акслантиришлар асосида қисм калитлар учун (32 байтдан иборат бўлган 8 та) 256 байтдан иборат калитлар ҳосил қилинади:

4.1. $i = 0; j = 0;$

4.2. $j = (j + S_i + k_i) \bmod 256;$

4.3. $P = S_i;\ Q = S_j;$

4.4. $S_i=Q$; $S_j=P$;

4.5. $i<255$ шарт текширилади. Агар бу шарт бажарилса, $i=i+1$ деб олиниб, 4.2. қадамга ўтилади, акс ҳолда кейинги қадамга ўтилади;

4.6. $i=0$; $j=0$; $l=0$;

4.7. $i=(i+1) \bmod 256$;

4.8. $j=(j+S_i) \bmod 256$;

4.9. $P=S_i$; $Q=S_j$;

4.10. $S_i=Q$; $S_j=P$;

4.11. $t=(S_i+S_j) \bmod 256$;

4.12. $k_l=S_t$;

4.13. $l=l+1$;

4.14. $l<256$ шарт текширилади. Агар бу шарт бажарилса, 4.7-қадамга ўтилади, акс ҳолда $K=k_0$, k_1, ..., k_l, ..., k_{255} калитлар ҳосил қилиниши тугатилади.

5. $i=0$.

6. $K=k_0$, k_1, ..., k_{31} (256 бит = 32 байт) қисм калит билан $T(0)=t_0(0)$ ... $t_7(0)t_8(0)$ ... $t_{15}(0)$ ... $t_{248}(0)$ ... $t_{255}(0)$ блок устида $\oplus$ амали бажарилади ва натижа $H(0)=h_0(0)$ $h_1(0)$ ... $h_{255}(0)$ деб белгиланади, яъни $K \oplus T(0)=H(0)$.

7. Ҳосил бўлган 256 битли блок тўртта 64 битли блокларга ажратилади $X=h_0(0)$ ... $h_{63}(0)$, $Y=h_{64}(0)$ ... $h_{127}(0)$, $Z=h_{128}(0)$ ... $h_{191}(0)$, $W=h_{192}(0)$ ... $h_{255}(0)$ ва қуйидаги мантиқий функцияларнинг қийматлари ҳисобланади:

$$F(X; Y; Z; W)=(X \wedge Y) \vee (Z \wedge W);$$
$$G(X; Y; Z; W)=(X \wedge Z) \vee (Y \wedge W);$$
$$R(X; Y; Z; W)=X \oplus Y \oplus Z \oplus W;$$
$$V(X; Y; Z; W)=(X \vee Y) \oplus (\overline{Z} \vee \overline{W}).$$

Бу ерда битлар бўйича мантиқий AND, OR, NOT, XOR амаллари мос равишда $\wedge$, $\vee$, $\overline{\ }$, $\oplus$ белгилари билан ифодаланган. Бу мантиқий функцияларнинг қийматлари 64 битли блоклар бўлади.

8. Юқорида берилган тўртта 64 битли блоклар конкатенация қилинади ва ҳосил қилинган 256 битли блок $L(0)$ орқали белгиланади:

$$L(0)=F(X; Y; Z; W) \| G(X; Y; Z; W) \| R(X; Y; Z; W) \| V(X; Y; Z; W)=$$
$$=l_0(0)l_1(0)...l_{255}(0).$$

9. $H(0)$ ва $L(0)$ конкатенация қилинади ва ҳосил қилинган 512 битли блок $A(0)$ орқали белгиланади:

$$A(0)=H(0) \| L(0)=h_0(0)\ h_1(0) \ldots h_{255}(0)\ l_0(0)\ l_1(0) \ldots l_{255}(0)=a_0(0)=$$
$$=a_1(0) \ldots a_{511}(0)$$

10. $A(0)$ блокни 4 битдан блокларга қуйидагича ажратиб олинади:

$a_0(0)\ a_1(0)\ a_2(0)\ a_3(0)=x_0$, $a_4(0)\ a_5(0)\ a_6(0)\ a_7(0)=y_0$,

$a_8(0)\ a_9(0)\ a_{10}(0)\ a_{11}(0)=x_1$, $a_{12}(0)\ a_{13}(0)\ a_{14}(0)\ a_{15}(0)=y_1$, ...,

$a_{0+8i}(0)\ a_{1+8i}(0)\ a_{2+8i}(0)\ a_{3+8i}(0)=x_i$, $a_{4+8i}(0)\ a_{5+8i}(0)\ a_{6+8i}(0)\ a_{7+8i}(0)=y_i$, ...,

$a_{0+8\cdot63}(0)a_{1+8\cdot63}(0)a_{2+8\cdot63}(0)a_{3+8\cdot63}(0)=x_{63}$, $a_{4+8\cdot63}(0)a_{5+\cdot63}(0)a_{6+8\cdot63}(0)a_{7+8\cdot63}(0)=y_{63}$.

$x_i \| y_i = b_i$ деб белгиланади.

11. Сиқиш жадвали берилган бўлиб, унда ҳар бир сатрда 0 дан 15 гача бўлган сонлар маълум тартибда жойлаштирилган. Бу жадвалдан фойдаланишда бир байтлик блокни ярим байтли блокка сиқиш натижаси сифатида x_i сатр ва y_i устунлар кесишган катакдаги сон олинади. Сиқиш жадвали қуйидаги кўринишда бўлади:

7.4-жадвал

	0	1	2	...	y_i	...	15
0	$d_0(0)$	$d_1(0)$	$d_2(0)$	...	$d_{yi}(0)$	...	$d_{15}(0)$
1	$d_0(1)$	$d_1(1)$	$d_2(1)$	...	$d_{yi}(1)$	...	$d_{15}(1)$
2	$d_0(2)$	$d_1(2)$	$d_2(2)$	...	$d_{yi}(2)$	...	$d_{15}(2)$
...	...	...	...	...	...	...	...
x_i	$d_0(x_i)$	$d_1(x_i)$	$d_2(x_i)$	...	$d_{yi}(x_i)$	...	$d_{15}(x_i)$
...	...	...	...	...	...	...	...
15	$d_0(15)$	$d_1(15)$	$d_2(15)$	...	$d_{yi}(15)$	...	$d_{15}(15)$

Сиқиш жадвали акслантириши СЖ деб белгиланса, бу акслантириш натижасида ҳосил қилинган 256 битли блокни $T(1)$ орқали белгиланади:

$СЖ(x_0y_0x_1y_1 \dots x_{63}y_{63})$ (512 бит)$=d_{y0}(x_0)\ d_{yi}(x_1) \dots d_{y63}(x_{63})$ (256 бит)$=t_0(1)\ t_1(1) \dots t_{255}(1)=T(1)$.

12. $i=i+1$.

13. $i<8$ шарт текширилади. Агар бу шарт бажарилса,

$T(0)=T(1)$,

$k_0=k_{0+32\cdot i}$, ..., $k_{31}=k_{31+32\cdot i}$

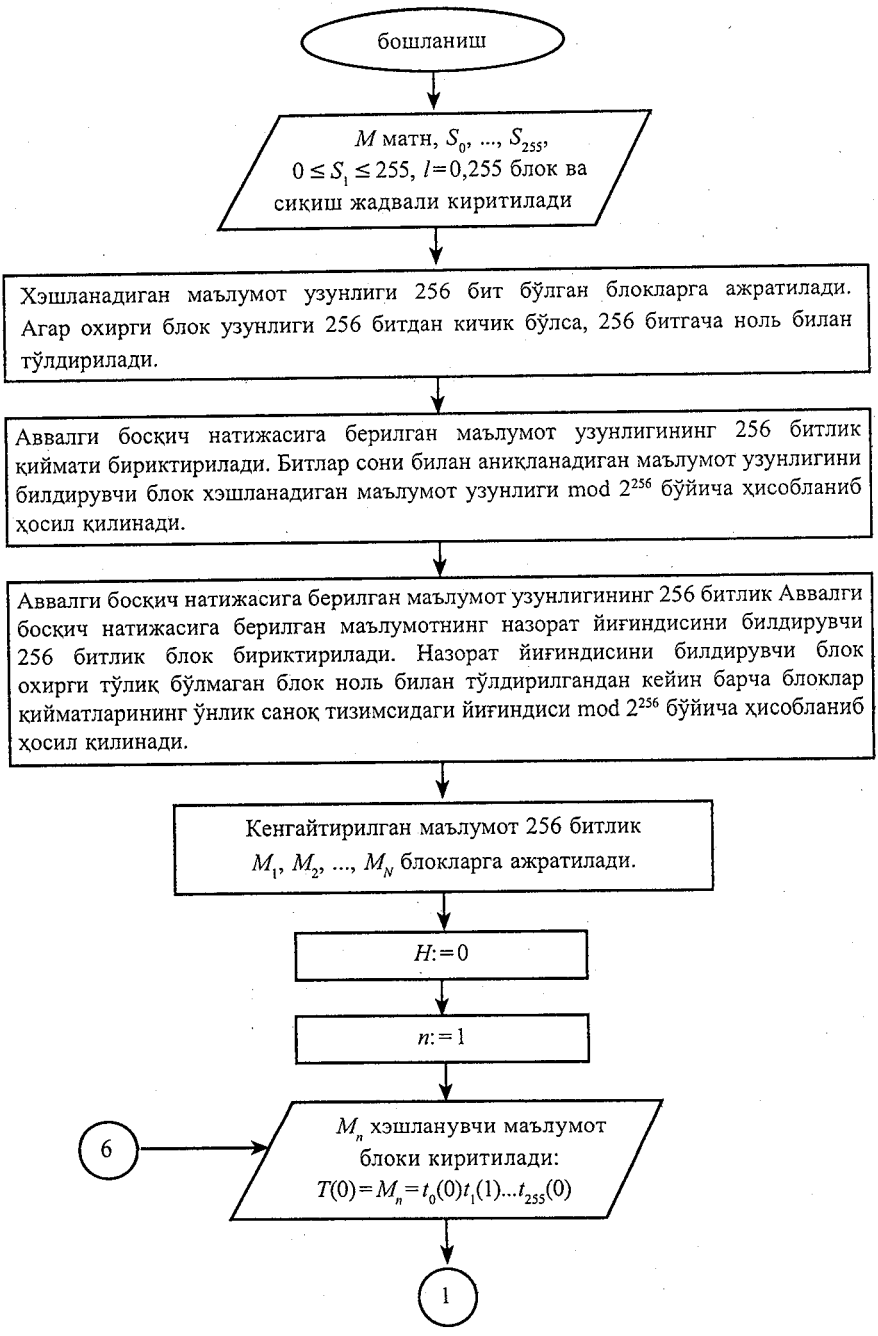
деб олинади ва 6-қадамга ўтилади, акс ҳолда кейинги қадамга ўтилади.

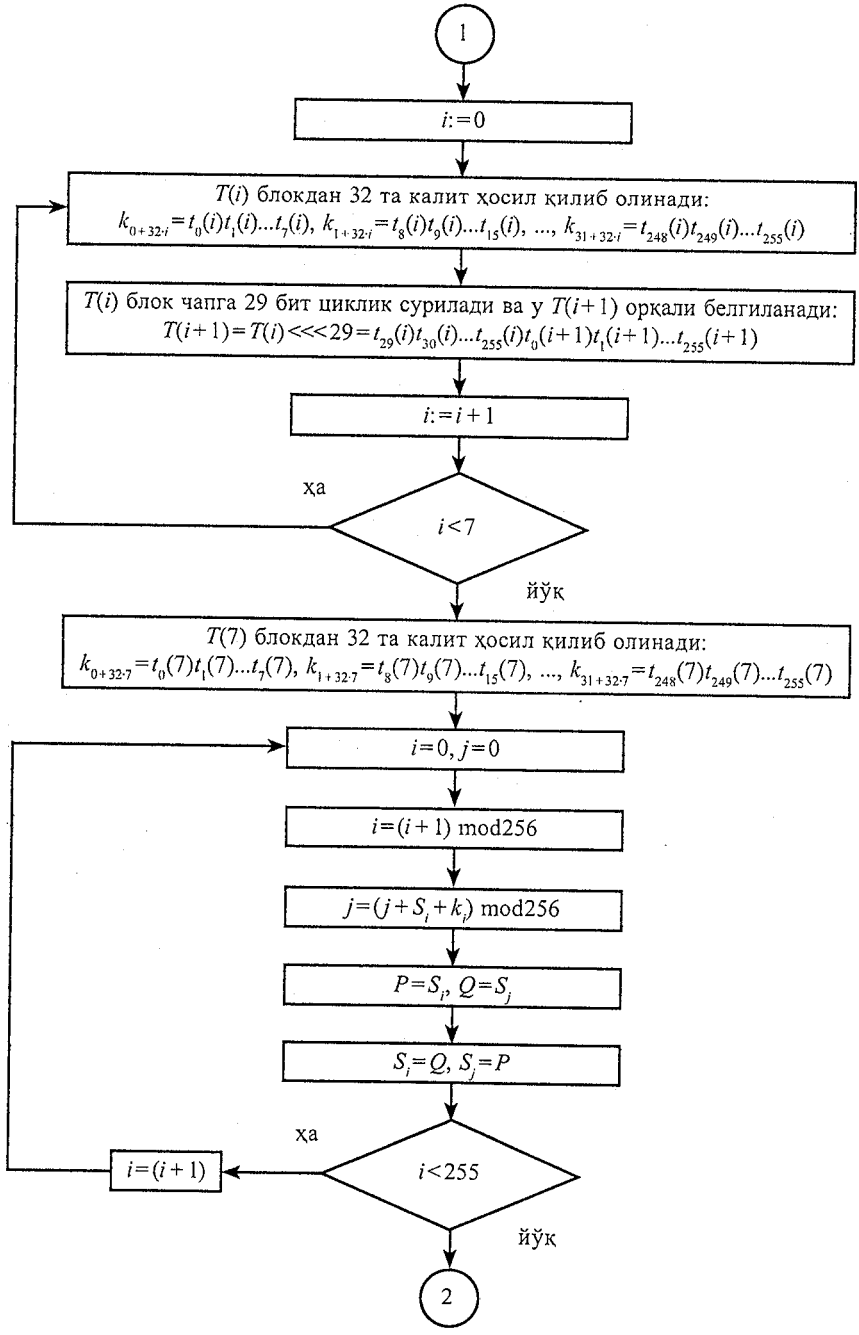
14. $H=H\oplus T(1)$.

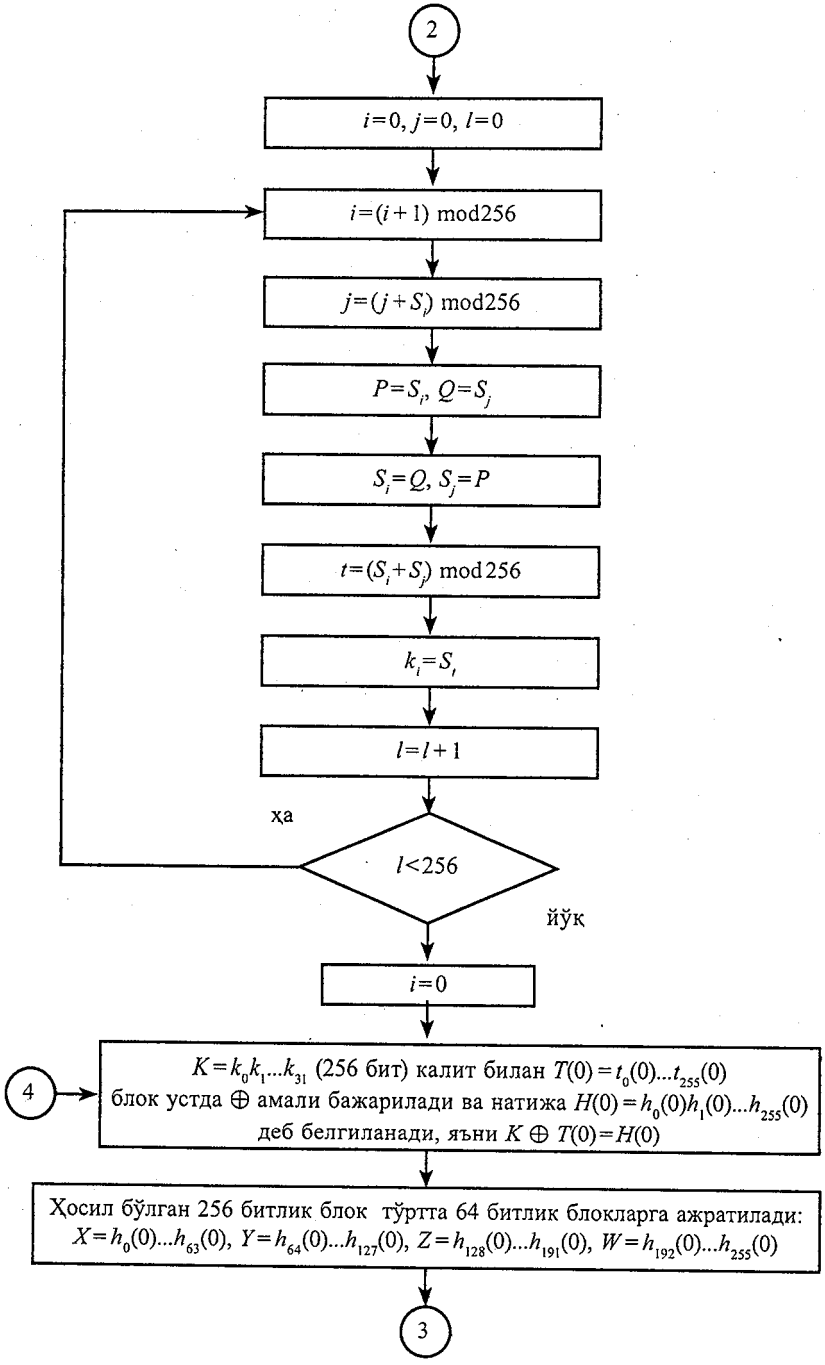
15. $n=n+1$.

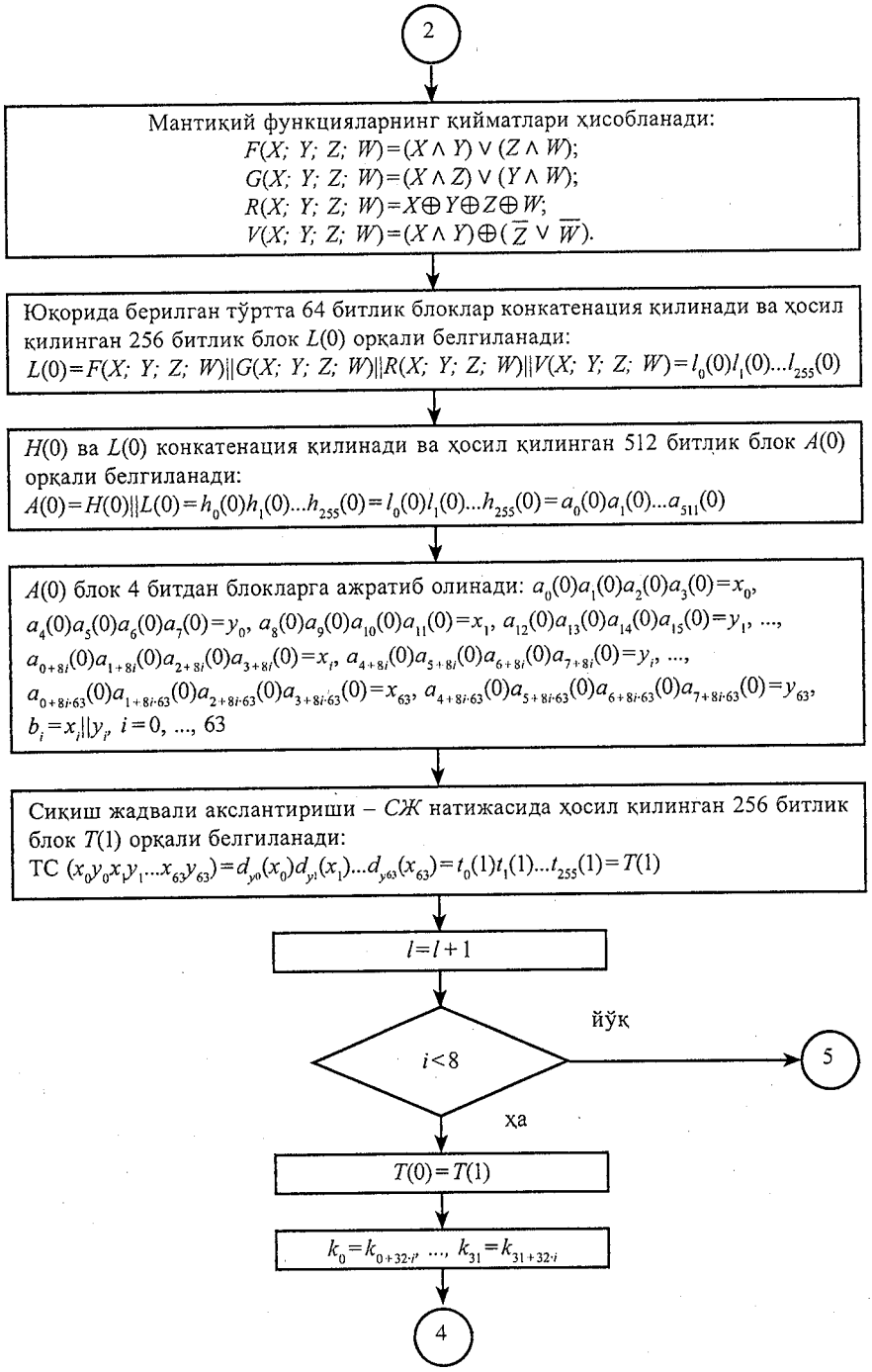
16. $n \le N$ шарт текширилади. Агар бу шарт бажарилса, 2-қадамга ўтилади, акс ҳолда H нинг қиймати берилган M матннинг хэш қиймати бўлади.

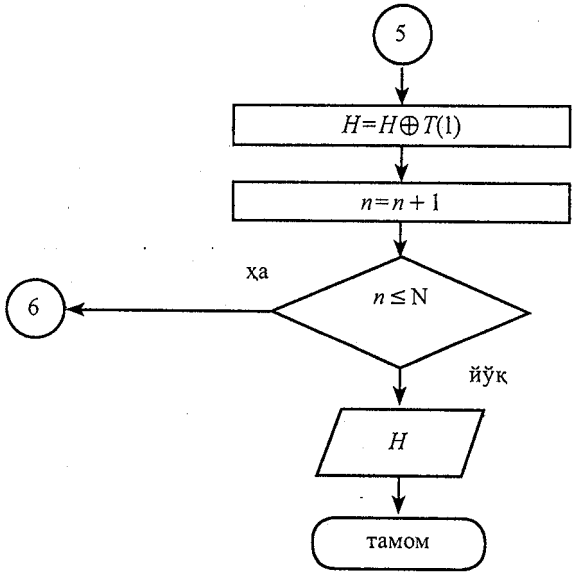
Калитсиз хэш – функция алгоритмининг блок-схемаси




1
i:=0
T(i) блокдан 32 та калит ҳосил қилиб олинади:
k0+32·i=t0(i)t1(i)...t7(i), k1+32·i=t8(i)t9(i)...t15(i), ..., k31+32·i=t248(i)t249(i)...t255(i)
T(i) блок чапга 29 бит циклик сурилади ва у T(i+1) орқали белгиланади:
T(i+1)=T(i)<<<29=t29(i)t30(i)...t255(i)t0(i+1)t1(i+1)...t255(i+1)
i:=i+1
ҳа
i<7
йўқ
T(7) блокдан 32 та калит ҳосил қилиб олинади:
k0+32·7=t0(7)t1(7)...t7(7), k1+32·7=t8(7)t9(7)...t15(7), ..., k31+32·7=t248(7)t249(7)...t255(7)
i=0, j=0
i=(i+1) mod256
j=(j+Si+ki) mod256
P=Si, Q=Sj
Si=Q, Sj=P
ҳа
i<255
i=(i+1)
йўқ
2


2
$i=0, j=0, l=0$
$i=(i+1) \bmod 256$
$j=(j+S_i) \bmod 256$
$P=S_i, Q=S_j$
$S_i=Q, S_j=P$
$t=(S_i+S_j) \bmod 256$
$k_i=S_t$
$l=l+1$
ха
$l<256$
йўқ
$i=0$
4
$K=k_0k_1...k_{31}$ (256 бит) калит билан $T(0)=t_0(0)...t_{255}(0)$
блок устда ⊕ амали бажарилади ва натижа $H(0)=h_0(0)h_1(0)...h_{255}(0)$
деб белгиланади, яъни $K \oplus T(0)=H(0)$
Ҳосил бўлган 256 битлик блок тўртта 64 битлик блокларга ажратилади:
$X=h_0(0)...h_{63}(0)$, $Y=h_{64}(0)...h_{127}(0)$, $Z=h_{128}(0)...h_{191}(0)$, $W=h_{192}(0)...h_{255}(0)$
3


2
Мантиқий функцияларнинг қийматлари ҳисобланади:
$F(X;\ Y;\ Z;\ W)=(X\wedge Y)\vee(Z\wedge W)$;
$G(X;\ Y;\ Z;\ W)=(X\wedge Z)\vee(Y\wedge W)$;
$R(X;\ Y;\ Z;\ W)=X\oplus Y\oplus Z\oplus W$;
$V(X;\ Y;\ Z;\ W)=(X\wedge Y)\oplus(\overline{Z}\vee\overline{W})$.
Юқорида берилган тўртта 64 битлик блоклар конкатенация қилинади ва ҳосил қилинган 256 битлик блок $L(0)$ орқали белгиланади:
$L(0)=F(X;\ Y;\ Z;\ W)||G(X;\ Y;\ Z;\ W)||R(X;\ Y;\ Z;\ W)||V(X;\ Y;\ Z;\ W)=l_0(0)l_1(0)...l_{255}(0)$
$H(0)$ ва $L(0)$ конкатенация қилинади ва ҳосил қилинган 512 битлик блок $A(0)$ орқали белгиланади:
$A(0)=H(0)||L(0)=h_0(0)h_1(0)...h_{255}(0)=l_0(0)l_1(0)...h_{255}(0)=a_0(0)a_1(0)...a_{511}(0)$
$A(0)$ блок 4 битдан блокларга ажратиб олинади: $a_0(0)a_1(0)a_2(0)a_3(0)=x_0$, $a_4(0)a_5(0)a_6(0)a_7(0)=y_0$, $a_8(0)a_9(0)a_{10}(0)a_{11}(0)=x_1$, $a_{12}(0)a_{13}(0)a_{14}(0)a_{15}(0)=y_1$, ..., $a_{0+8i}(0)a_{1+8i}(0)a_{2+8i}(0)a_{3+8i}(0)=x_i$, $a_{4+8i}(0)a_{5+8i}(0)a_{6+8i}(0)a_{7+8i}(0)=y_i$, ..., $a_{0+8i\cdot63}(0)a_{1+8i\cdot63}(0)a_{2+8i\cdot63}(0)a_{3+8i\cdot63}(0)=x_{63}$, $a_{4+8i\cdot63}(0)a_{5+8i\cdot63}(0)a_{6+8i\cdot63}(0)a_{7+8i\cdot63}(0)=y_{63}$, $b_i=x_i||y_i$, $i=0$, ..., 63
Сиқиш жадвали акслантириши – СЖ натижасида ҳосил қилинган 256 битлик блок $T(1)$ орқали белгиланади:
ТС $(x_0y_0x_1y_1...x_{63}y_{63})=d_{y0}(x_0)d_{y1}(x_1)...d_{y63}(x_{63})=t_0(1)t_1(1)...t_{255}(1)=T(1)$
$l=l+1$
$i<8$
йўқ
5
ҳа
$T(0)=T(1)$
$k_0=k_{0+32\cdot i}$, ..., $k_{31}=k_{31+32\cdot i}$
4



Калитсиз хэш – функция алгоритми учун келтирилган мисолнинг дастурий таъминоти

Аввалги бобда келтирилган янги калитсиз хэш-функциянинг C++ дастурлаштириш тилида тузилган дастури қуйидагидан иборат:

```
// HeshDlg.cpp : файл реализации
//
#include «stdafx.h»
#include «Hesh.h»
#include «HeshDlg.h»
#include «.\heshdlg.h»
CString str,str1;
byte M[32];
byte T[32], T1[32], sl, sr, s;
byte k[256];
byte S_b[256], Q;
        for (int i=0; i<=255; i++)
                S_b[i]=i;

Update Data(1);
str = fayl_s;
        FILE *f1;
```

```
        f1 = fopen(str,»rb»);
byte r[1];
byte uz_f[32];
        for (i=0; i<=31; i++)
        uz_f[i] = 0;
uz_f[31] = 1;
int j;
        while((fread(r, 1, 1, f1)) !=0)
        {
        for ( j=31; j>0; j--)
        {
        if (uz_f[j]==0)
        uz_f[j-1]=uz_f[j-1]+1;
        else break;
        }
        uz_f[31]++;
        }
        j=31;
        while(uz_f[j]==0)
        j--;
        uz_f[j]--;
byte uj;
        for ( j=0; j < 31; j++)
        {
uz_f[j] = uz_f[j] << 3;
        uj=uz_f[j-1] >> 5;
        uz_f[j]=uz_f[j] | uj;
        }
        uz_f[31]=uz_f[31] << 3;

fseek( f1, 0, SEEK_SET);
UpdateData(false);

byte H[32], Sum[32];
        for (i=0; i<=31; i++)
                Sum[i]=0;
for (i=0; i<=31; i++)
                H[i]=0;
bool mantiq1=false;
bool mantiq2=false;
```

```
int nechta=0;
while(true)
{
for (i=0; i<=31; i++)
        M[i]=0;
        int nechta=fread(M, 1, 32, f1);
        if (nechta==0)
        {
        if (mantiq1)
        {
        for (i=31; i >=0; i--)
        T[i]=Sum[i];
        mantiq2=true;
        }
        else
        {
        mantiq1=true;
        for (i=31; i >=0; i--)
        T[i]=uz_f[i];
        }
        }
        if (!mantiq1)
        {
        int per=0;
        //========== Controlniy summa ==========
        for (i=31; i>=0; i--)
        {
                int si=Sum[i];
                int mi=M[i];
                si = si+mi+per;
                Sum[i]=si;
                per=si >> 8;
        }
        // ----------------------------------

        for (i=0; i<=31; i++)
        T[i]=M[i];
        }

        for (i=0; i<=7; i++)
```

```
        {
        for (j=0; j<=31; j++)
        k[j+32*i]=T[j];

        // ============ T <<< 29 ============
        for (j=0; j<=31; j++)
        T1[j] = T[(j+3)%32];
        byte t10=T1[0];
        for (j=0; j<31; j++)
        {
        sl=T1[j]<<5;
        sr=T1[j+1]>>3;
        s=sl | sr;
        T[j]=s;
        }
        sl=T1[31]<<5;
        t10=t10 >>3;
        T1[31]=sl | t10;
        // ----------------------------------
        }
        i=0;
        j=0;
        //============S_b aralashtirish ============
        for (i=0; i<=255; i++)
        {
                j=(j+S_b[i])%256;
                Q=S_b[i];
                S_b[i]=S_b[j];
                S_b[j]=Q;
        }
        // ----------------------------------

        // ========== S_b aralashtirib yangi kalitlarga berish
i=0;
j=0;
int t;
        for (int l=0; l<=255; l++)
        {
                i=(i+1)%256;
                j=(j+S_b[i]+k[i])%256;
```

```
        Q=S_b[i];
S_b[i]=S_b[j];
S_b[j]=Q;
t = (S_b[i]+S_b[j]) % 256;
k[l]=S_b[t];
}
// ----------------------------------
// ============= Katta sikl boshi
for (i=0; i<=7; i++)
{
for (j=0; j<=31; j++)
H[j] = T[j]^k[32*i+j];          // H=T XOR k;
byte X[8], Y[8], Z[8], W[8];
for (l=0; l<=7; l++)
{
X[l]=H[l];
Y[l]=H[l+8];
Z[l]=H[l+16];
W[l]=H[l+24];
}

//============ Funksiyalar bilan ishlash ==============
byte F[8], G[8], R[8], V[8], L0[32];

for (l=0; l<=7; l++)
{
        F[l]=(X[l] & Y[l]) | (Z[l] & W[l]);
        G[l =(X[l] & Z[l]) | (Y[l] & W[l]);
        R[l]=X[l] ^ Y[l] ^ Z[l] ^ W[l];
        V[l]=(X[l] | Y[l]) ^ ((~Z[l]) | (~W[l]));
}

for (l=0; l<=7; l++)
{
        L0[l]=F[l];
        L0[l+8]=G[l];
        L0[l+16]=R[l];
        L0[l+24]=V[l];
        }
```

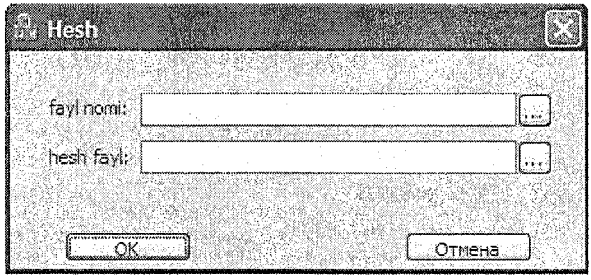
```
                byte A0[64];
        for (l=0; l<=31; l++)
        {
        A0[l]=H[l];
        A0[l+32]=L0[l];
        }

//=================================================
s_j siqish jadvali ==================
        byte s_j [16] [16];
        for (int is=0; is<=15; is++)
        for (int js=0; js<=15; js++)
        {
        s_j [is] [js]=(js+is)%16;
        }
        byte xi, yi, kn=0;
        for (int in=0; in<=63; in=in+2)
        {
        xi=A0 [in] >> 4;
        yi=(A0 [in] << 4);
        yi=yi>>4;
        byte al = s_j [xi] [yi] << 4;

        xi=A0 [in+1]>>4;
        yi=(A0 [in+1]<<4);
        yi=yi >> 4;
        byte ar=s_j [xi][yi];

        T[kn]=al | ar;
        kn++;
        }
// ------------------------------------------------------------------------
        for (int js=0; js<=31; js++)
        {
        H[js] = H[js]^T[js];
        }
        // ----------------------------------
        }
        if (mantiq2) break;
}
```

```
fclose(f1);
str1 = «»;
str = «»;
int in;
        // ============== hesh qiymatni 16 likka o'tkazish
for (i=0; i<32; i++)
{
        char ch=H[i];
        if (ch < 0)
        in=256+ch;
        else
        in = ch;
        str.Format(«%x», in);
        if (strlen(str)==1)
        str1=«0»+str;
        str1= str1+str+«,»;
        UpdateData(0);
}
        //---------------- hesh qiymatni 16 likka o'tkazish tugadi -----
        //============== hesh qiymatni 2 likka o'tkazish
        CString str3=«»;
for (i=0; i<32; i++)
{
        char ch=H[i];
        if (ch < 0)
        in =256+ch;
        else
        in=ch;
        CString str=«»;
        CString str2=«»;
        for (int i1=1; i1<=8; i1++)
        {
        int qol=in % 2;
        in =in/2;
        str.Format(«%d»,qol);
        str2=str+str2;
        }
        str3=str3+str2+«,»;
}
        //---------------- hesh qiymatni 2 likka o'tkazish tugadi ------
```

```
MessageBox («Jarayon tugadi», «Natija»);
CStdio File fayl;
str = hesh_s;
        fayl. Open(str, CStdio File :: modeCreate | CStdioFile :: modeWrite);
        fayl. WriteString (str1);
        fayl. WriteString («\n»);
        fayl. WriteString (str3);
fayl.Close();
```

Ушбу дастур қуйидагича ишлайди:

Heshdlg.cpp файлини компиляция қилиш натижасида Hesh.exe файлини оламиз. Мазкур файлни ишлатиш натижасида



кўринишдаги мулоқот ойнаси ҳосил бўлади. fayl nomi дарчасида биз хэшланадиган файл номини ва hesh fayl дарчасида эса хэш қиймат ёзиладиган файл номини кўрсатамиз. OK тугмачасини босиш натижасида хэш қиймат ёзилган файл ҳосил бўлади.

Қуйида хэш қийматни ҳисоблашга доир мисоллар натижалари келтирилган:

1) M=a бўлса, у ҳолда бу матннинг хэш қиймати:

H=3c305025bba6fdacdb07b6f5fa2e99d0163670b1fbd99fe9bfd1fd7dee43bf3f.

2) M=b бўлса, у ҳолда бу матннинг хэш қиймати:

H=978746d8fefc3c590d68b99aa3b565cd73248f4de25a80b3a1ecda9343487de2.

3) M=123456789abcdefghijklmnopqrstuvwxyz бўлса, у ҳолда бу матннинг хэш қиймати:

H=0989f924c10c57f60e6b04d148f123e50f2a48bc2f8a7c06ac0a82bf02c2099d.

§ 7.9. Калитсиз хэш – функция алгоритмига мисол сифатида келтирилган алгоритмнинг бардошлилигини баҳолаш

Мисол сифатида таклиф этилган янги хэш – функция алгоритмида қуйидаги акслантиришлардан фойдаланилган:

1) Матндан калитларни генерация қилиш акслантиришлари;

2) $K \oplus T(0) = H(0)$;

3) Мантиқий функциялар акслантиришлари:

$$F(X;\ Y;\ Z;\ W) = (X \wedge Y) \vee (Z \wedge W);$$

$$G(X;\ Y;\ Z;\ W) = (X \wedge Z) \vee (Y \wedge W);$$

$$R(X;\ Y;\ Z;\ W) = X \oplus Y \oplus Z \oplus W;$$

$$V(X;\ Y;\ Z;\ W) = (X \vee Y) \oplus (\overline{Z} \vee \overline{W}).$$

4) Конкатенация қилиш натижасида олинадиган 256 битлик блок

$L(0) = F(X;\ Y;\ Z;\ W) \| G(X;\ Y;\ Z;\ W) \| R(X;\ Y;\ Z;\ W) \| V(X;\ Y;\ Z;\ W)$;

5)Конкатенация қилиш натижасида олинадиган 512 битлик блок

$A(0) = H(0) \| L(0) = h_0(0)\ h_1(0) \ldots h_{255}(0)\ l_0(0)\ l_1(0) \ldots l_{255}(0) = a_0(0)\ a_1(0) \ldots a_{511}(0)$;

6) Сиқиш жадвали

$TC(x_0 y_0 x_1 y_1 \ldots x_{63} y_{63})$ (512 бит) $= d_{y_0}(x_0) d_{y_1}(x_1) \ldots d_{y_{63}}(x_{63})$ (256 бит);

7) $H = H \oplus T(1)$.

Хэш-функцияда қатнашган акслантиришларнинг мураккаблик даражалари жадвали

7.5.-жадвал

Акслантиришлар	Акслантириш асосидаги амал	Бит узунлиги	Тескариси бир қийматлилиги	Тескариси кўп қийматлилиги	Мураккаблик даражаси
Матндан калитларни генерация қилиш акслантиришлари	$\wedge$, $\vee$, $\neg$	256 байт	–	+	2^{1700}
$K \oplus T(0) = H(0)$	$\oplus$	256	–	+	2^{256}
$F(X;\ Y;\ Z;\ W) = (X \wedge Y) \vee (Z \wedge W)$	$\wedge$, $\vee$,	64	–	+	2^{192}
$G(X;\ Y;\ Z;\ W) = (X \wedge Z) \vee (Y \wedge W)$	$\wedge$, $\vee$,	64	–	+	2^{192}
$R(X;\ Y;\ Z;\ W) = X \oplus Y \oplus Z \oplus W$	$\oplus$	64	–	+	2^{192}
$V(X;\ Y;\ Z;\ W) = (X \vee Y) \oplus (\overline{Z} \vee \overline{W})$	$\oplus$, $\vee$, $\neg$	64	–	+	2^{192}
$L(0) = F(X;\ Y;\ Z;\ W) \| G(X;\ Y;\ Z;\ W) \| \| R(X;\ Y;\ Z;\ W) \| V(X;\ Y;\ Z;\ W)$	$\|$	256	+	–	1
$A(0) = H(0) \| L(0)$	$\|$	512	+	–	1
$TC(x_0 y_0 x_1 y_1 \ldots x_{63} y_{63})$ (512 бит) $= d_{y_0}(x_0) d_{y_1}(x_1) \ldots d_{y_{63}}(x_{63})$ (256 бит)	жадвал	256	–	+	2^{256}
$H = H \oplus T(1)$	$\oplus$	256	–	+	2^{256}

Қуйида янги калитсиз хэш – функция мураккаблик даражасининг жадвали келтирилади:

7.6-жадвал

Акслантиришлар	Акслантириш асосидаги амал	Нечта битдан иборат	Тескариси бир қийматли	Тескариси кўп қийматли	Мураккаблик даражаси
$H=H\oplus T$	$\oplus$	256	–	+	2^{256}
$TC(x_0y_0x_1y_1...x_{63}y_{63})$ (512 бит)= $= d_{y_0}(x_0)d_{y_1}(x_1)...d_{y_{63}}(x_{63})$(256 бит)	жадвал	256	–	+	2^{512}
$A(0)=H(0)\,\|\|\,L(0)$	$\|\|$	512	+	–	2^{512}
$L(0)=F(X;\,Y;\,Z;\,W)\|\|G(X;\,Y;\,Z;\,W)\|\|$ $\|\|\,R(X;\,Y;\,Z;\,W)\|\|\,V(X;\,Y;\,Z;\,W)$	$\|\|$	256	+	–	2^{512}
$V(X;\,Y;\,Z;\,W)=(X\vee Y)\oplus(\overline{Z}\vee\overline{W})$	$\oplus$, $\vee$, $\neg$	64	–	+	2^{704}
$R(X;\,Y;\,Z;\,W)=X\oplus Y\oplus Z\oplus W$	$\oplus$	64	–	+	2^{704}
$G(X;\,Y;\,Z;\,W)=(X\wedge Z)\vee(Y\wedge W)$	$\wedge$, $\vee$	64	–	+	2^{704}
$F(X;\,Y;\,Z;\,W)=(X\wedge Z)\vee(Y\wedge W)$	$\wedge$, $\vee$	64	–	+	2^{704}
$K\oplus T(0)=H(0)$	$\oplus$	256	–	+	2^{960}
8 раундда					$2^{960\cdot7+704}$
М матн					2^{7424}

Демак, бир блок узунликдаги матнни тиклашнинг мураккаблик даражаси 2^{7424} га тенг.

§ 7.10. Мисол сифатида таклиф этилган калитсиз хэш – функция алгоритмининг криптотаҳлили

Хэш – функцияларга қуйидаги криптоҳужум турлари мавжуд [14, 29, 39]:

1. Хэш-функция алгоритмида қатнашган акслантиришлар бўйича криптоҳужум;
2. Алгоритм акслантиришларининг дифференциал криптотаҳлил усулига бардошлилиги;
3. «Туғилган кун парадокси»га асосланган криптоҳужум;
4. Матнларнинг тўқнашуви усули асосидаги криптоҳужум;
5. Барча мумкин бўлган вариантларни танлаш асосидаги ҳужум.

Мисол сифатида таклиф этилган янги калитсиз хэш – функция алгоритмида қатнашган асосий акслантиришлар бир томонлама

бўлиб, уларнинг тескариси кўп қийматлидир. Шунинг учун бу хэш-функцияда акслантиришлар шундай танланганки, алгоритмдаги акслантиришларда ва раундлар орасида коллизияни топиб бўлмайди, бу эса алгоритмда коллизияни топиш имкониятини йўқотади.

Келтирилган алгоритмда кириш ва чиқиш битлари сони тенг, шунинг учун бу алгоритмга дифференциал криптотаҳлил усули асосидаги ҳужумни амалга ошириб бўлмайди. Фақат сиқиш жадвали акслантиришида кириш ва чиқиш битлари сони тенг бўлмаганлиги учун бу акслантиришга нисбатан дифференциал криптотаҳлил усули асосидаги ҳужумни амалга ошириш имконияти борга ўхшаб кўринади, лекин бу акслантиришда кириш битлари сони 512 ва чиқиш битлари сони 256 бўлгани учун, кирувчи блокнинг мумкин бўлган вариантлари сони 2^{512} ва чиқувчи блокнинг мумкин бўлган вариантлари сони 2^{256} бўлади. Бу эса замонавий ҳисоблаш қурилмалари имкониятларидан самарали фойдаланилганда ҳам дифференциал айирмада ҳар бир берилган сонга мос келувчи блокларни тиклаш имкониятини бермайди.

Ювалнинг «Туғилган кун парадокси»га асосланган криптоҳужуми хэш – функцияларда коллизияларни топиш учун ишлатиладиган асосий криптоҳужумлардан биридир. Бу криптоҳужумга асосан хэш қиймат берилганда унга мос бўлган маълумотни танлашнинг мураккаблиги $O(2^n)$ катталик билан, маълумот ва унинг хэш қиймати берилганда, хэш қиймати шунга тенг бўладиган бошқа маълумотни танлашнинг мураккаблиги $O(2^{n/2})$ катталик билан баҳоланади. Биз тадқиқ қилаётган мазкур хэш – функцияда хэш қиймат берилганда унга мос бўлган маълумотни танлашнинг мураккаблиги $O(2^{256})$ катталик билан, маълумот ва унинг хэш қиймати берилганда, хэш қиймати шунга тенг бўладиган бошқа маълумотни танлашнинг мураккаблиги $O(2^{128})$ катталик билан баҳоланади. Бу эса замонавий ҳисоблаш қурилмалари имкониятларидан самарали фойдаланилганда ҳам коллизиялар ҳосил қилиш имкониятини бермайди.

Хэш-функцияларни қуришдаги итератив усул матнлар тўқнашуви усули асосида коллизиялар ҳосил қилиш имконини беради. Унинг олдини олиш учун хэшланувчи матн охирига маълумот узунлиги ва назорат йиғинди қўшиб қўйилади. Яратилган янги калитсиз хэш – функцияда ҳам ушбу ҳолат эътиборга олинган.

Мазкур хэш-функцияда барча мумкин бўлган вариантларни танлаш асосидаги криптоҳужум ҳам самара бермайди. Чунки берилган хэш қиймат бўйича барча мумкин бўлган матнларни тикласак, уларнинг сони 2^{7424} га тенг бўлади. Бунинг эса амалий жиҳатдан имконияти йўқ, чунки замонавий ҳисоблаш технологиялари имкониятлари доирасида буни амалга ошириб бўлмайди.

7-боб бўйича хулосалар

Ушбу бобда:

1. Электрон ҳужжатли маълумотларни очиқ турдаги алоқа тармоғи орқали узатишда уларнинг тўлалигини таъминлаш (ўзгармаган ҳолда бир фойдаланувчидан иккинчи фойдаланувчига етказиш) ва маълумот манбаини аутентификация қилиш (қабул қилинган электрон ҳужжат айнан кўрсатилган манба томонидан яратилганининг тасдиғини олиш) муҳим масалалардан бири бўлиб, бу масалани фақатгина криптография усулларини қўллаган ҳолда ҳал қилиш қулай ва самарали эканлиги, яъни бу масалани ҳал қилишда махсус криптографик алгоритм – хэш – функция алгоритмидан фойдаланилиши ёритилди.

2. Мавжуд хэш – функциялар калитли ва калитсиз функцияларга ажратиб ўрганилди, уларнинг хоссалари таҳлил қилинди, уларнинг ахборот-коммуникация тизимларида маълумот алмашинуви муҳофазасини таъминлаш масалаларини ечишда қўлланилиш схемалари берилди.

3. Мавжуд стандарт хэш – функцияларнинг хоссалари таҳлил қилинди., Жумладан, Россия Федерациясининг давлат стандарти ҳисобланган ГОСТ Р 34.11-94 хэш – функцияси, АҚШнинг федерал стандарти SHA туркумидаги хэш – функцияларни яратишга асос бўлган MD5 хэш – функцияси, АҚШ федерал стандарти ҳисобланган SHA-1 хэш – функцияси, Беларусь Республикасининг хэш – функция давлат стандарти ҳисобланган СТБ 1176.1–99 хэш – функцияси, Ўзбекистон Республикасининг давлат стандарти ҳисобланган O‘z DSt 1106:2006 хэш – функциялари алгоритмлари таҳлил қилинди.

4. Тадқиқ қилинган стандарт хэш-функцияларда қатнашган бардошли акслантиришлардан ва бу хэш – функцияларнинг ижобий томонларидан фойдаланиб, ҳамда ҳоссалари мавжуд криптографик акслантиришлардан фарқли бўлган янги акслантиришлар асосида мисол сифатида янги калитсиз хэш – функция алгоритми яратилди, унинг блок-схемаси ва C++дастурлаш тилида дастурий таъминоти берилди.

5. Олинган натижалар криптологик нуқтаи назардан таҳлил қилинди, мисол сифатида келтирилган янги калитсиз хэш – функциянинг криптоҳужумларга бардошлилиги исботланиб, унинг самарадорлиги баҳоланди. Ушбу янги калитсиз хэш – функция архитектураси жиҳатидан дастурий таминотини яратишга қулай бўлиб, у ахборот-коммуникация тизимларида қўлланилганда, фақат бу хэш – функциянинг ўзидан фойдаланиш ёки электрон рақамли имзо тизимида фойдаланиш мумкин.

VIII БОБ
ЭЛЕКТРОН РАҚАМЛИ ИМЗО АЛГОРИТМЛАРИ

§8.1. Электрон имзо

Қабул қилиб олинган маълумотларнинг ҳақиқий ёки ҳақиқий эмаслигини аниқлаш масаласини, яъни маълумотлар аутентификацияси масаласининг моҳияти ҳақида тўҳталамиз.

Ҳар қандай ёзма хат ёки ҳужжатнинг охирида шу ҳужжатни тузувчиси ёки тузиш учун жавобгар бўлган шахснинг имзоси бўлиши табиий ҳолдир. Бундай ҳолат одатда қуйидаги иккита мақсаддан келиб чиқади. Биринчидан, маълумотни олган томон ўзида мавжуд имзо наъмунасига олинган маълумотдаги имзони солиштирган ҳолда шу маълумотнинг ҳақиқийлигига ишонч ҳосил қилади. Иккинчидан, шахсий имзо маълумот ҳужжатига юридик жиҳатдан муаллифликни кафолатлайди. Бундай кафолат эса савдо–сотиқ, ишончнома, мажбурият ва шу каби битимларда алоҳида муҳимдир.

Ҳужжатлардаги қўйилган шахсий имзоларни сохталаштириш нисбатан мураккаб бўлиб, шахсий имзоларнинг муаллифларини ҳозирги замонавий илғор криминалистика услубларидан фойдаланиш орқали аниқлаш мумкин. Аммо электрон рақамли имзо хусусиятлари бундан фарқли бўлиб, иккилик саноқ тизими хусусиятлари билан белгиланадиган хотира регистрлари битларига боғлиқ. Хотира битларининг маълум бир кетмакетлигидан иборат бўлган электрон имзони кўчириб бирор жойга қўйиш ёки ўзгартириш компьютерлар асосидаги алоқа тизимларида мураккаблик туғдирмайди.

Бугунги юқори даражада ривожланган бутун дунё цивилизациясида ҳужжатлар, жумладан махфий ҳужжатларнинг ҳам, электрон кўринишда ишлатилиши ва алоқа тизимларида узатилиши кенг қўлланилиб борилаётганлиги электрон ҳужжатлар ва электрон имзоларнинг ҳақиқийлигини аниқлаш масалаларининг муҳимлигини келтириб чиқармоқда.

Очиқ калитли криптографик тизимлар қанчалик қулай ва криптобардошли бўлмасин, аутентификация масаласининг тўла ечилишига жавоб бера олмайди. Шунинг учун аутентификация услуби ва воситалари криптографик алгоритмлар билан биргаликда комплекс ҳолда қўлланилиши талаб этилади.

Қуйида иккита (А) ва (Б) фойдаланувчиларнинг алоқа муносабатларида аутентификация тизими рақиб томоннинг ўз мақсади йўлидаги

қандай хатти-ҳаракатларидан ва криптотизим фойдаланувчиларининг фойдаланиш протоколини ўзаро бузилишлардан сақлаши кераклигини кўрсатувчи ҳолатлар кўриб чиқилади.

Рад этиш (ренегатство)

Фойдаланувчи (*А*) фойдаланувчи (*Б*) га ҳақиқатан ҳам маълумот жўнатган бўлиб, узатилган маълумотни рад этиши мумкин.

Бундай қоида бузилишининг (тартибсизликнинг) олдини олиш мақсадида электрон (рақамли) имзодан фойдаланилади.

Модификациялаш (ўзгартириш)

Фойдаланувчи (*Б*) қабул қилиб олинган маълумотни ўзгартириб, шу ўзгартирилган маълумотни фойдаланувчи (*А*) юборди, деб таъкидлайди (даъво қилади).

Сохталаштириш

Фойдаланувчи (Б) нинг ўзи маълумот тайёрлаб, бу сохта маълумотни фойдаланувчи (А) юборди деб даъво қилади.

Фаол модификациялаш (ўзгартириш)

(*А*) ва (*Б*) фойдаланувчиларнинг ўзаро алоқа тармоғига учинчи бир (*В*) фойдаланувчи ноқонуний тарзда боғланиб, уларнинг ўзаро узатаётган маълумотларини ўзгартирган ҳолда деярли узлуксиз узатиб туради.

Ниқоблаш (имитациялаш)

Учинчи фойдаланувчи (*В*) фойдаланувчи (*Б*) га фойдаланувчи (*А*) номидан маълумот жўнатади.

Юқорида санаб ўтилган: модификациялаш, сохталаштириш, фаол модификациялаш, ниқоблаш каби алоқа тизими қоидаларининг бузилишини олдини олиш мақсадида рақамли сигнатурадан – рақамли имзо ва узатиладиган маълумотнинг бирор қисмини тўла ўз ичига олувчи рақамли шифрматндан иборат бўлган маълумотдан фойдаланилади.

Такрорлаш

Фойдаланувчи (*В*) фойдаланувчи (*А*) томонидан фойдаланувчи (Б) га жўнатилган маълумотни такроран (*Б*) га жўнатади. Бундай ноқонуний ҳатти–ҳаракат алоқа усулидан банклар тармоқларида

электрон ҳисоб–китоб тизимидан фойдаланишда ноқонунийлик билан ўзгалар пулларини талон-тарож қилишда фойдаланилади. Ана шундай ноқонуний усуллардан муҳофазаланиш учун қуйидаги чора – тадбирлар кўрилади:

– имитациялашга бардошлилик – имитабардошлилик;

– криптотизимга кираётган маълумотларни муҳофаза мақсадларидан келиб чиқиб тартиблаш.

Электрон рақамли имзо алоқа тизимларида бир неча тур қоида бузилишларидан муҳофаза қилинишни таъминлайди, яъни:

– махфий калит фақат фойдаланувчи (А)нинг ўзигагина маълум бўлса, у ҳолда фойдаланувчи (Б) томонидан қабул қилиб олинган маълумотни фақат (А) томонидан жўнатилганлигини рад этиб бўлмайди;

– қонун бузар (рақиб томон) махфий калитни билмаган ҳолда модификациялаш, сохталаштириш, фаол модификациялаш, ниқоблаш ва бошқа шу каби алоқа тизими қоидаларининг бузилишига имконият туғдирмайди;

– алоқа тизимидан фойдаланувчиларнинг ўзаро боғлиқ ҳолда иш юритиши муносабатидаги кўплаб келишмовчиликларни бартараф этади ва бундай келишмовчиликлар келиб чиққанда воситачисиз аниқлик киритиш имконияти туғилади.

Кўп ҳолларда узатилаётган маълумотларни шифрлашга ҳожат бўлмай, уни электрон рақамли имзо билан тасдиқлаш керак бўлади. Бундай ҳолатларда очиқ матн жўнатувчининг ёпиқ калити билан шифрланиб, олинган шифрматн очиқ матн билан бирга жўнатилади. Маълумотни қабул қилиб олган томон жўнатувчининг очиқ калити ёрдамида шифрматнни дешифрлаб, очиқ матн билан солиштириши мумкин.

1991 йилда АҚШ даги Стандартлар ва Технологиялар Миллий Институти DSA (Digital Signature Algorithm) рақамли имзо алгоритмининг стандартини DSS (Digital Signature Standard) биз юқорида келтирган Эл-Гамал ва RSA алгоритмлари асосида яратиб, фойдаланувчиларга таклиф этган.

Дастлаб таъкидланганидек, имзо ҳужжатнинг юридик мақомини кафолатлайди. Ҳозирги ривожланган жамиятда ахборот коммуникация тармоқларида электрон маълумот алмашинувининг кенгайиб бориши маълумотларнинг махфийлигини, ҳақиқийлигини ва муаллифликни ўрнатиш масалаларини ечишни талаб этади. Масалан, алмашилган электрон маълумотлар асосида у ёки бу ҳолатнинг ўзгариши, бу маълумотлар муаллифи манфаатларига зид келиб, у электрон маъ-

лумот муаллифлигидан бош тортиши мумкин. Шундай ҳолатларнинг олдини олиш механизми маълумот муаллифини ўзигагина маълум бўлган бирор сонли параметр (махфий калит) билан боғлиқ ҳолда ҳосил қилинадиган сонлар кетма-кетлигидан иборат бўлган электрон рақамли имзо (ЭРИ) ҳисобланади.

ЭРИ ахборот коммуникация тармоғида электрон ҳужжат алмашинуви жараёнида қуйидаги учта масалани ечиш имконини беради:

– электрон ҳужжат манбаининг ҳақиқийлигини аниқлаш;

– электрон ҳужжат яхлитлигини (ўзгармаганлигини) текшириш;

– электрон ҳужжатга рақамли имзо қўйган субъектни муаллифликдан бош тортмаслигини таъминлайди.

Ҳар қандай ЭРИ алгоритми иккита қисмдан иборат:

– имзо қўйиш;

– имзони текшириш.

Имзо қўйиш муаллиф томонидан, фақат унга маълум бўлган махфий калит билан амалга оширилади. Имзонинг ҳақиқийлигини текшириш эса исталган шахс томонидан, имзо муаллифининг очиқ калити билан амалга оширилиши мумкин.

Электрон коммуникациялар ва электрон ҳужжат алмашинуви ҳозирги кунда иш юзасидан бўладиган муносабатларнинг ажралмас қисми ҳисобланиб, ҳар қандай замонавий ташкилотни электрон ҳужжатлар алмашинуви ва Интернетсиз тасаввур қилиш қийин.

Интернет тармоғидан электрон ҳужжатлар алмашинуви асосида молиявий фаолият олиб боришда маълумотлар алмашинувини ҳимоя қилиш ва электрон ҳужжатнинг юридик мақомини таъминлаш биринчи даражали аҳамият касб этади.

Электрон ҳужжатли маълумот алмашинуви жараёнида ЭРИни қўллаш ҳар хил турдаги тўлов тизимлари (пластик карточкалар), банк тизимлари ва савдо соҳаларининг молиявий фаолиятини бошқаришда электрон ҳужжат алмашинуви тизимларининг ривожланиб бориши билан кенг тарқала бошлади.

Ҳозирда ЭРИ тизимини яратишнинг бир нечта йўналишлари мавжуд. Бу йўналишларни учта гуруҳга бўлиш мумкин:

1) очиқ калитли шифрлаш алгоритмларига асосланган;

2) симметрик шифрлаш алгоритмларига асосланган;

3) имзони ҳисоблаш ва уни текширишнинг махсус алгоритмларига асосланган рақамли имзо тизимларидир.

Очиқ калитли шифрлаш алгоритмларига асосланган ЭРИ тизимлари қуйидагича ташкил қилинади. Агар ахборот – коммуникация

тармоғининг i – фойдаланувчиси j – фойдаланувчисига имзоланган электрон ҳужжат жўнатмоқчи бўлса, i – фойдаланувчи ўзининг махфий калити $k_i^м$ билан имзоланиши керак бўлган ҳужжатни ўзини шифрлаб ёки унинг хэш қийматини шифрлаб, шу ҳужжат билан биргаликда жўнатади. Бу электрон ҳужжатни қабул қилиб олган j – фойдаланувчи, шифрланган маълумотни i – фойдаланувчининг очиқ калити k_i^0 билан дешифрлаб, хосил бўлган матнни ҳужжат матнига ёки унинг хэш қийматига солиштиради. Агар матнлар билан хэш қийматлар бир хил бўлса имзо ҳақиқий, акс ҳолда ҳақиқий эмас деб қабул қилинади.

Симметрик шифрлаш алгоритмларига асосланган ЭРИ тизимлари қуйидагича ташкил этилади. i – фойдаланувчи бир вақтнинг ўзида i – фойдаланувчига ҳам j – фойдаланувчига ҳам маълум бўлиб, бошқа фойдаланувчиларга маълум бўлмаган $k_{ij}^м$ – калит билан имзоланиши керак бўлган электрон ҳужжатни ёки унинг хэш қийматини шифрлаб, шу ҳужжат билан биргаликда жўнатади. Электрон ҳужжатни қабул қилиб олган j – фойдаланувчи, шифрланган маълумотни $k_{ij}^м$ – калит билан дешифрлаб, ҳосил бўлган матнни ҳужжат матнига ёки унинг хэш қийматига солиштиради. Агар матнлар билан хэш қийматлар бир хил бўлса имзо ҳақиқий, акс ҳолда ҳақиқий эмас деб қабул қилинади. Бундай ЭРИ тизими бир марталик ҳисобланади, чунки $k_{ij}^м$ – калитдан иккинчи марта фойдаланиш имконияти электрон ҳужжатларни қалбакилаштириш имкониятини яратади. Бундай ҳолатга чек қўйиш учун электрон ҳужжат алмашинуви ишончли учинчи томон орқали амалга оширилиши мумкин: i – фойдаланувчи ўзига ва фақат ишончли учинчи томонга маълум бўлган калит $k_{i3}^м$ билан рақамли имзони амалга ошириб, имзоланган электрон ҳужжатни учинчи ишончли томонга жўнатади, учинчи томон имзони хақиқийлигини $k_{i3}^м$ – калит билан текшириб, агар ҳақиқий бўлса, j – фойдаланувчининг ўзига ва фақат ишончли учинчи томонга маълум бўлган калит $k_{i3}^м$ билан рақамли имзони амалга ошириб, имзоланган электрон ҳужжатни j – фойдаланувчига жўнатади. Бундай ЭРИ тизими фойдаланувчилар учун ноқулай бўлиб, кўплаб келишмовчиликларни келтириб чиқаради.

Амалда, учинчи турдаги имзони ҳисоблаш ва уни текширишнинг махсус алгоритмларига асосланган рақамли имзо тизимларидан кенг фойдаланилади.

Махсус ЭРИ алгоритмлари рақамли имзони ҳисоблаш ва имзони текшириш қисмларидан иборат. Рақамли имзони ҳисоблаш қисми имзо қўйувчининг махфий калити ва имзоланиши керак бўлган ҳужжатнинг хэш қийматига боғлиқ бўлади. Имзони текшириш қисми

имзо эгасининг очиқ калитига ва қабул қилиб олинган ҳужжатнинг хэш қийматига боғлиқ ҳолда амалга оширилади.

Махсус ЭРИ стандартлари туркумига [2]:

1. Россия ЭРИ стандарти: ГОСТ Р 34.10-94 ва унинг эллиптик эгри чизиқда такомиллаштирилган варианти **ГОСТ Р 34.10-2001;**

2. Америка ЭРИ стандарти: DSA ва унинг эллиптик эгри чизиқда такомиллаштирилган варианти **ECDSA -2000;**

3. Ўзбекистон Республикаси стандарти: O'zDSt 1092:2005; алгоритмлари мисол бўла олади

Рақамли имзо битлар кетма-кетлигида ифодаланган бирор сондан иборат. Шунинг учун уни бошқа электрон ҳужжатларга кўчириш ёки ўзгартириш киритиш катта қийинчилик туғдирмайди. Шунинг учун электрон ҳужжат алмашинуви тизимида ЭРИ ни қалбакилаштиришнинг олдини олиш чора – тадбирлари – ЭРИ алгоритмининг электрон ҳужжатларни қалбакилаштиришга бардошлилик масаласини ечиш талаб этилади.

ЭРИ алгоритмининг бардошлилиги қуйдаги учта масаланинг мураккаблиги билан аниқланади [2, 14, 20]:

– *имзони қалбакилаштириш*, берилган ҳужжатга, махфий калитга эга бўлмаган ҳолда тўғри имзо ҳисоблаш;

– *имзоланган маълумотни ташкил этиш*, махфий калитга эга бўлмаган ҳолда тўғри имзоланган маълумотни топиш;

– *маълумотни алмаштириш*, бир хил имзога эга бўлган иккита ҳар хил маълумотни топиш.

Келтирилган ЭРИ алгоритмлари стандартлари бардошлиликлари дискрет логарифимлаш ва эллиптик эгри чизиқ рационал нуқталари устида амаллар бажариш масалаларининг мураккаблигига асосланган.

Қуйида ахборт-коммуникация тармоғининг махфий электрон ҳужжат алмашиш тизими асимметрик шифрлаш алгоритмидан иборат бўлганда ЭРИни очиқ калитли шифрлаш алгоритми асосида амалга ошириш мисол тариқасида кўриб ўтилади.

Шудай қилиб, i – фойдаланувчи M – махфий маълумотни j – фойдаланувчига имзо қўйган ҳолда жўнатмоқчи бўлса, у ҳолда i – фойдаланувчи қуйдагиларни амалга ошириши керак:

1. Маълумо M тизим фойдаланувчиларининг барчасига маълум бўлган хэш-функция $h: X \rightarrow Y$ (бу ерда X – очиқ матнлар тўплами, Y – хэшлаш натижасида хосил бўлган қиймат) билан қайд қилинган бит узунлигидаги ифодага сиқилади;

2. Маълумотни хэш қиймати $h(M)=H$ фақат i – фойдаланувчининг ўзига маълум бўлган махфий калитга k_i^u боғлиқ бўлган бир томонлама функция E, орқали шифрланади, яъни $E_{k_i^m}(h(M))=S$.

3. Сўнгра, j – фойдаланувчининг очиқ калити k_j^0 билан маълумот M ва S бирлаштириланган кенгайтирилган маълумот шифрланади, яъни $E_{k_j^0}(M \cup S)=E_{k_j^0}(M)\cup E_{k_j^0}(S)=E_{k_j^0}(M)\cup E_{k_j^0}(E_{k_j^m}(h(M)))=C_1\cup C_2=C$

4. Шифрланган маълумот C очиқ алоқа тармоғи орқали j – фойдаланувчига жўнатилади.

Шифрланган маълумотни олган j – фойдаланувчи, фақат унинг ўзига маълум бўлган махфий калит k_j^u билан дешифрлашни амалга оширади, яъни

$$D_{k_j^3}(C)=D_{k_j^3}(C_1\cup C_2)=D_{k_j^3}(C_1)\cup D_{k_j^3}(C_2)=D_{k_j^3}(E_{k_j^0}(M))\cup$$

$$\cup D_{k_j^3}(E_{k_j^0}(E_{k_j^3}(h(M))))=M\cup E_{k_j^3}(h(M))$$

бу ерда: ЭРИ ифодаси $E_{k_i^m}(h(M))$ хали дешифрланмаган.

5. Маълумот эгасини ва маълумотнинг ўзини ҳақиқийлигига ишоч хосил қилиш учун j – фойдаланувчи i – фойдаланувчининг очиқ калити k_i^0 билан ЭРИ қисмини $E_{k_i^m}(h(M))$ дешифрлаб $h(M)$ – ифодани олади, яъни

$$D_{k_i^0}(E_{k_i^m}(h(M)))=h(M)$$

6. Сўнгра, j – фойдаланувчи дешифрлаш натижасида олган $D_{k_j^m}(C_1)$ очиқ маълумотни калитсиз хэш – фунция билан хэшлайди $h(D_{k_j^3}(C_1))$ ва ушбу $D_{k_i^0}(E_{k_i^m}(h(M)))=h(M)$ таққослаш билан имзонинг тўғрилигига ишонч ҳосил қилиши мумкин, агарда $h(D_{k_j^3}(C_1))$ $= D_{k_i^0}(E_{k_i^m}(h(M)))=h(M)$ бўлса, акс ҳолда имзо нотўғри, ҳамда, электрон ҳужжат ҳақиқий бўлмайди.

ЭРИ имзонинг тўғрилиги маълумотни ўзини, унинг авторини ва манбасининг ҳақиқийлигини кафолатлайди.

Таъкидлаш жоизки, 1 – 6 – бандлар, асимметрик криптотизимларда маълумот алмашинувчи томонларнинг ЭРИ протоколини ифодалайди. Криптографик протокол деб, икки ва ундан ортиқ томонлар қатнашган ҳолда махфий маълумот алмашинуви жарёнида томонларнинг ўз вазифаларини бажариши кетма-кетлиги тушунилади.

Қуйида очиқ калитли шифрлаш алгоритмларига асосланган ЭРИ алгоритмлари кўриб ўтилади.

§8.2. RSA очиқ калитли шифрлаш алгоритми асосидаги ЭРИ

Тизимнинг ҳар бир i – фойдаланувчиси (e_i, d_i) – калитлар жуфтлигини яратади. Бунинг учун етарли катта бўлган p ва q – туб сонлари олиниб (бу сонлар махфий тутилади), $n=pq$ – сони ва Эйлер функциясининг қиймати $\varphi(n)=(p-1)(q-1)$ ҳисобланади (бу сон ҳам махфий тутилади). Сўнгра, $(e_i, \varphi(n))=1$ шартни қаноатлантирувчи, яъни $\varphi(n)$ – сони билан ўзаро туб бўлган e_i – сон бўйича d_i – сони ушбу $e_i d_i = 1 \bmod \varphi(n)$ формула орқали ҳисобланади. Бу $(e_i; d_i)$ жуфтликда e_i – очиқ калит ва d_i – махфий калит деб эълон қилинади.

Шундан сўнг i – фойдаланувчидан j – фойдаланувчига шифрланган маълумотни имзолаган ҳолда жўнатиши қуйидагича амалга оширилади:

1. Шифрлаш қоидаси: $M^{e_j} \bmod n = C$, бу ерда M – очиқ маълумот, C – шифрланган маълумот;

2. Дешифрлаш қоидаси: $C^{d_j} \bmod n = M^{e_j d_j} \bmod n = M$;

3. ЭРИ ни ҳисоблаш: $H(M)^{d_j} \bmod n = P_i$,

бу ерда: i – фойдаланувчининг P_i – имзоси M – маълумотнинг H(M) – хэш – функция қиймати бўйича ҳисобланган;

4. ЭРИ ни текшириш:

$(P_i)^{e_i} \bmod n = H(M)^{e_i d_i} \bmod n = H(M)$, агар $H(M)=H(M_1)$ бўлса (бу ерда M_1 – дешифрланган маълумот), у ҳолда электрон ҳужжат ҳақиқий, акс ҳолда ҳақиқий эмас, чунки хэш – функция хоссасига кўра $M=M_1$ бўлса уларнинг хэш қийматлари ҳам тенг бўлади.

5. Маълумотни махфий узатиш протоколи:

$$[M \cup H(M)^{d_i}]^{e_j} \bmod n = [M \cup P_i]^{e_j} \bmod n = C$$

6. Махфий узатилган маълумотни қабул қилиш протоколи:

$C^{d_j} \bmod n = [M \cup P_i]^{e_j d_j} \bmod n = M \cup P_i$ умуман қараганда дастлабки маълумот ўзгартирилган бўлиши мумкин, шунинг учун $C^{d_j} \bmod n = M_i \cup P_i$ бўлиб, натижада, хэш қиймат имзо бўйича ушбу ифода $(P_i)^{e_i} \bmod n = H(M)^{e_i d_i} \bmod n = H(M)$ билан ҳисобланади ва қабул қилиб олинган маълумотнинг хэш қиймати $H(M_1)$ бўлса, у ҳолда $H(M)=H(M_1)$ бўлганда электрон ҳужжат ҳақиқий, аксинча бўлса қалбаки ҳисобланади.

§8.3. Эл-Гамал очиқ калитли шифрлаш алгоритми асосидаги ЭРИ

Эл-Гамал очиқ калитли шифрлаш алгоритмига асосланган криптотизимнинг ҳар бир i – фойдаланувчиси учун очиқ ва махфий калитлар генерацияси қуйидагича амалга оширилади, очиқ эълон қилинадиган p_i – туб сон (ёки фойдаланувчилар гуруҳи учун умумий бўлган p – туб сон) танланади, ушбу $g_i<p_i$ (ёки фойдаланувчилар гуруҳи учун $g<p$) шартни қаноатлантирувчи g_i (ёки фойдаланувчилар гуруҳ учун g) сони танланади, ушбу $y_i=g^{x_i} \bmod p_i$ (p – умумий бўлганда $y_i=g^{x_i} \bmod p$, $x_i<p$) формула билан x_i – махфий калит бўйича y_i сони ҳисобланади. Шундай қилиб, (p_i, g_i, y_i) – параметрлар бирикмаси (умумий p ва g учун (p, g, y_i) – параметрлар бирикмаси очиқ калитни ташкил этади, махфий калит x_i ҳисобланади.

Тизимда i – фойдаланувчидан j – фойдаланувчига шифрланган маълумотнинг имзоланган ҳолда жўнатилиши қуйидагича амалга оширилади:

1. Шифрлаш қоидаси: $a_j=g_j^k \bmod p_j$, $b_j=y_j^k M \bmod p_j$ (умумий p и g лар учун $a=g^k \bmod p$, $b_j=y_j^k M \bmod p$), бу ерда k – тасодифий сон бўлиб маълумотни имзоловчи томонидан танланади, бу сон (p_j-1) сони билан ўзаро туб ЭКУБ $(k, p_j-1)=1$ (p ва g умумий бўлганда ЭКУБ $(k, p_j-1)=1$), M – очиқ маълумот, шифрланган маълумот $(a_j, b_j)=C$ (p ва g умумий бўлганда, $(a, b_j)=C$).

2. Дешифрлаш қоидаси: $b_j \Big/ a_j^{x_j} \bmod p_j = M$ (p ва g умумий бўлганда $b \Big/ a^{x_j} \bmod p = M$), ҳақиқатан ҳам $b_j \Big/ a_j^{x_j} \bmod p_j \equiv g_j^{x_i k} M \Big/ g_j^{kx_i} \bmod p_j \equiv M$ (p ва g умумий бўлганда $b \Big/ a^{x_j} \bmod p \equiv y_j^k M \Big/ a^{x_i} \bmod p \equiv g^{x_j k} M \Big/ g^{kx_i} \bmod p$ так как $M<p$);

3. ЭРИ ни ҳисоблаш қоидаси: $a_i=g_i^k \bmod p_i$, b_i сони эса $M=(x_i a_i+kb_i) \bmod (p_i-1)$ ёки $H(M)=(x_i a_i+kb_i) \bmod (p_i-1)$ тенгламадан топилади, яъни $b_i=(M-a_i x_i)k^{-1} \bmod (p_i-1)$ ёки $b_i=(M-a_i x_i)k^{-1} \bmod (p_i-1)$ (p ва g умумий бўлганда $a=g^k \bmod p$, b сони эса $M=(x_i a+kb) \bmod (p-1)$ ёки $H(M)=(xa+kb) \bmod (p_i-1)$ тенгламадан топилади, яъни $b=(M-ax_i)k^{-1} \bmod (p-1)$ ёки $b=(H(M)-ax_i)k^{-1} \bmod (p-1)$ ёки ЭКУБ $(k, p-1)=1$) $H(M)$ – маълумотнинг хэш қиймати, x_i – махфий калит, имзо сифатида a_i ва b_i жуфтлик, яъни $(a_i, b_i)=P_i$, (p ва g умумий бўлганда (a, b)) имзо деб қабул қилинади.

4. Имзони текшириш қоидаси:

Агар $y_i^{a_i} a_i^{b_i} \ mod p_i = g_i^M \bmod p_i$ ёки $y_i^{a_i} \ a_i^{b_i} \ mod p_i = g_i^{H(M)} \bmod p_i$ бўлса, у ҳолда электрон ҳужжат ҳақиқий, акс ҳолда қалбаки ҳисобланади. Чунки,

$$y_i = g_i^{x_i} \ mod p_i \text{ ва } a_i = g_i^k \ mod p_i$$

тенгликлар ўринли бўлиб, Ферма теоремасига кўра ушбу айният ўринли:

$$y_i^{a_i} a_i^{b_i} \ mod p_i = (g_i^{x_i})^{a_i} (g_i^k)^{b_i} \bmod p_i = g_i^{a_i x_i + k b_i} \bmod p_i = g_i^{d(p_i-1)+M} \bmod p_i =$$
$$= g_i^{d(p_i-1)} g_i^M \bmod p_i = (g_i^{(p_i-1)})^d \bmod p_i \cdot \ g_i^M \bmod p_i \ (\bmod p_i) =$$
$$= 1^d \bmod p_i \cdot \ g_i^M \bmod p_i \ (\bmod p_i) = g_i^M \bmod p_i;$$

5. **Маълумотни махфий узатиш протоколи**:

$a_j \cdot g_j^k \bmod p_j$, $b_j \cdot y_j^k M \bmod p_j = y_j^k [M \cup P_i] \bmod p_j$, $(a_j, b_j) = C$ шифрмаълумот;

6. **Махфий узатилган маълумотни қабул қилиш протоколи:**

$$b_j \Big/ a_j^{x_j} \bmod p_j = M' = M \cup P_i$$

умуман қараганда, дастлабки маълумот ўзгартирилган бўлиши мумкин, шунинг учун

$$b_j \Big/ a_j^{x_j} \bmod p_j = M' = M_1 \cup P_i$$

бўлиб, $H(M_1)$ – хэш қиймат ҳисобланади. Агар $y_i^{ai} \ a_i^{bi} \bmod p_i = g_i^{M_1} \bmod p_i$ ёки $y_i^{ai} \ a_i^{bi} \bmod p_i = g_i^{H(M)_1} \bmod p_i$ бўлса, у ҳолда электрон ҳужжат ҳақиқий, акс ҳолда қалбаки ҳисобланади.

Энди имзони ҳисоблаш ва уни текширишга асосланган ЭРИ алгоритмлари DSA ва ГОСТ Р 34.10-94 стандарлари билан танишилади. Бу алгоритмларнинг асосини Эл-Гамал шифрлаш алгоритми ташкил этади.

§8.4. DSA ЭРИ стандарти

1991 йилда NIST (National Institute of Standard and Technology) томонидан DSA (Digital Signature Algorithm) алгоритмига асосланган DSS (Digital Signature Standard) ЭРИ стандартининг лойиҳаси муҳокамага қўйилди. Ушбу алгоритм бардошлилиги етарли катта туб характеристикага эга бўлган чекли майдонда дискрет логарифмлаш масаласининг мураккаблигига асосланган [2, 14, 20]. Қуйида алгоритм қадамлари кетма-кетлиги келтирилган.

Имзони шакллантириш

1. Маълумот жўнатувчи M – маълумотни ва қуйидаги параметрларни кенг доирадаги тизим фойдаланувчиларига очиқ эълон қилади:

p – туб сон, $2^{512} < p < 2^{1024}$, бит узунлиги 64 га каррали;

q – туб сон, $2^{159} < q < 2^{160}$, $p-1$ нинг бўлувчиси;

$g = h^{(p-1)/q} \bmod p$, бу ерда h ушбу $0 < h < p$ ва $h^{(p-1)/q} \bmod p > 1$ шартларни қаноатлантирувчи бутун сон;

y – очиқ калит бўлиб, $y = q^x \bmod p$ формула орқали аниқланади. Бу ерда x – махфий калит бўлиб, $0 < x < q$ оралиқдан олинган ва фақат имзоловчининг ўзигагина маълум;

$H(M)$ – M маълумотдан [1; q] оралиқдаги бутун сонни генерация қилувчи хэш-функция.

2. Маълумот жўнатувчи $0 < k < q$ оралиқдан тасодифий k сонни танлайди, уни махфий тутади ва имзо генерациясидан кейин дарҳол йўқотади.

3. Маълумот жўнатувчи r ва s қийматларни қуйидаги қонуният орқали ҳисоблайди:

$$r = g^k \bmod p \bmod q,$$
$$s = k^{-1}(xr + H(M)) \bmod q.$$

M – маълумотга қўйилган имзо (r, s) сонлар жуфтлигидан иборат.

Имзони текшириш. Қабул қилувчи M' маълумотни ва (r', s') имзони қабул қилиб олади. У M ва M' маълумотларнинг мос келишини текшириши лозим. Бунинг учун у қуйидаги қадамлар кетмакетлигини бажаради:

1. $0 < s' < q$ ёки $0 < r' < q$ шартлардан бирортаси бажарилмаса, имзо қалбаки деб ҳисобланади ва имзони текшириш тугатилади.

2. $v = (s')^{-1} (\bmod q)$ топилади.

3. $z_1 = H(M')\ v \bmod q$, $z_2 = r'\ v \bmod q$ ҳисобланади.

4. Кейин $u = g^{z_1} y^{z_2} \bmod p \bmod q$ ҳисобланади.

Агар $r' = u$ тенглик ўринли бўлса, у ҳолда имзо ҳақиқий ва $M = M'$ тенглик тўғри.

Алгоритмнинг тўғрилиги. $M = M'$, $s' = s$ ва $r' = r$ бўлсин. У ҳолда $r = u$ тенглик ўринли бўлиши кўрсатилади.

Демак, $v = (s')^{-1} \bmod q$, $z_1 = H(M')\ v \bmod q$, $z_2 = r'\ v \bmod q$ эканлигидан, қуйидагини ёзиш мумкин:

$$u = g^{z1} y^{z1} \bmod p \bmod q = g^{H(M)s-1}\ g^{xrs-1} \bmod p \bmod q =$$
$$= g^{k(xr + H(M)) - 1(xr + H(M))} \bmod p \bmod q = g^k \bmod p \bmod q = r.$$

Бундан кўриш мумкинки, $r=u$ тенглик ўринли. Шундай қилиб, алгоритм тўғрилиги исботланди.

§8.5. ГОСТ Р 34.10-94 электрон рақамли имзоси

Ушбу параграфда 2000 йилгача Россия стандарти ҳисобланган ГОСТ Р 34.10–94 ЭРИ алгоритми қараб чиқилади. Бу алгоритм DSA алгоритмига ўхшаш ва қуйидаги бошланғич очиқ параметрлардан фойдаланади:

1) Узунлиги L бўлган катта p туб сон танланади, бу ерда L сон 509 битдан 512 битгача ёки 1020 битдан 1024 битгача ораликдан танланади, яъни $2^{509}<p<2^{512}$ ёки $2^{1020}<p<2^{1024}$.

2) Узунлиги L_1 бўлган катта q туб сон танланади, бу ерда L_1 сон 254 битдан 256 битгача ораликдан танланади, яъни $2^{254}<p<2^{256}$.

3) $g^q \bmod p=1$ шартни қаноатлантирувчи $0<g<p-1$ ораликдаги g сон танланади.

4) $y=g^x \bmod p$ дан y – очиқ калит ҳисобланади, бу ерда $0<x<q$ ораликдан олинган x – махфий калит.

5) $H(M)$ – хэш-функция берилган M – маълумот бўйича ҳисобланган бутун сон бўлиб, 1 дан q гача ораликдаги қийматларни қабул қилади, яъни $1<H(M)<q$.

Имзони генерация қилиш алгоритми. Бошланғич маълумотлар: M – маълумот, берилган параметрлар ва махфий калит. Натижа: имзо – (r, s).

1) $1 \le k \le q$ интервалдан тасодифий k сони олинади, у махфий сақланади ва имзо қўйилгандан кейин дарҳол йўқотилади.

2) $r=(g^k \bmod p) \pmod q$ ҳисобланади.

3) Жўнатилаётган M – маълумотнинг $e:=H(M)$ – хэш қиймати ҳисобланади.

4) Агар $r=0$ ёки $H(M) \bmod q=0$ бўлса, у ҳолда 1- қадамга ўтилиб, бошқа k танланади.

5) $s=(xr+kH(M)) \bmod q$ ҳисобланади, бу ерда махфий калит x фақат имзо қўювчининг ўзигагина маълум.

6) Агар $s=0$ бўлса, у ҳолда 1-қадамга борилади.

7) M маълумот имзоси – (r, s) жуфтлигидан иборат.

Имзони текшириш алгоритми. Бошланғич маълумотлар: M маълумот, берилган параметрлар, имзони текшириш калити ва M маълумот имзоси. Натижа: имзо ҳақиқийлиги ёки қалбакилиги ҳақидаги тасдиқ.

1) Агар $1 \le r,\ s \le n-1$ шарт бажарилмаса, у ҳолда имзо қалбаки ва имзони текшириш алгоритми тугатилади. Бу шартлар бажарилса кейинги қадамга ўтилади.

2) $e := h(m)$ ҳисобланади.

3) $w := H(M)^{(q-2)} \bmod q$ ҳисобланади.

4) $u_1 := sw \bmod q$ ҳисобланади.

5) $u_2 := (q-r)w \bmod q$ ҳисобланади.

6) $u := (g^{u_1} y^{u_2} \bmod p)$ ҳисобланади.

Агар $u = r$ шарт бажарилса, у ҳолда имзо ҳақиқий, акс ҳолда имзо қалбаки ва имзони текшириш алгоритми тугатилади.

ГОСТ Р 34.10-94 имзо алгоритмининг тўғрилиги. ГОСТ Р 34.10-94 электрон рақамли имзо генерацияси алгоритмидан олинган r параметрнинг қийматини имзони текшириш алгоритмидаги u параметр қиймати билан тенглигини кўрсатишимиз керак.

Ҳақиқатан, $u = (g^{u_1} y^{u_2} \bmod p) \pmod q = g^{sw \bmod q} \cdot g^{x(q-r)w \bmod q} \bmod p \bmod q =$

$= g^{(s+xq-xr)w \bmod q} \bmod p \bmod q = g^{(xr+kH(M)+xq-xr)w \bmod q} \bmod p \bmod q =$

$= g^{(kH(M)+xq)w \bmod q} \bmod p \bmod q =$

$= g^{kH(M)w \bmod q} \bmod p \cdot (g^q \bmod p)^{xw \bmod q} \pmod q = /(g^q \bmod p) = 1$

шартга кўра, $(g^q \bmod p)^{xw \bmod q} \pmod q = 1$ тенглик ўринли/ =

$= g^{kH(M)w \bmod q} \bmod p \pmod q = /w = H(M)^{(q-2)} \bmod q$, $0 < H(M) < q$ (q – туб) шартга ва Эйлера – Ферма теоремасига кўра $H(M)^{(q-2)} \bmod q = H(M)^{-1}$ эканлиги келиб чиқади, шунга кўра g нинг даражасини $kH(M)w = kH(M)$ $H(M)^{q-2} \bmod q = kH(M)H(M)^{-1} = k$ каби ифодалаш мумкин/ $= g^k \bmod p \pmod q = r$. Шундай қилиб талаб қилинган шарт кўрсатилди.

§8.6. Эллиптик эгри чизиқларга асосланган электрон рақамли имзо алгоритмлари

Эллиптик эгри чизиқларга асосланган криптотизимлар криптографияга 1985 йилда В. Миллер ва Н. Коблиц [19, 21] томонидан тадбиқ қилинган. Асимметрик криптографик алгоритмларнинг кўпчилиги чекли майдонда дискрет логарифмлаш масаласининг мураккаблигига асосланган бўлиб, бу алгоритмларни эллиптик эгри чизиқларга ўтказиш масаласи алоҳида изланиш талаб этади. Қуйида халқаро стандарт сифатида қабул қилинган EC DSA ва ГОСТ Р 34.10-2001 эллиптик эгри чизиқларга асосланган электрон рақамли имзо алгоритмлари кўриб ўтилади.

Электрон рақамли имзо алгоритмининг бошланғич параметрлари

Қаралаётган алгоритмнинг асосий параметрлари характеристикаси p – туб сондан иборат бўлган чекли майдонда аниқланган E – эллиптик эгри чизиқ ва шу чизиқда олинган катта туб тартибга эга бўлган $G \in E(F_p)$ – базавий нуқта ҳисобланади.

Бу чизиқ қуйидаги тенглама билан берилади [20, 22, 40]:

$$y^2 = x^3 + ax + b \pmod{p}.$$

Турли (a, b) параметрлар жуфтлиги изоморф эллиптик эгри чизиқларни аниқлайди. Тенгламанинг муҳим параметрлари эса мос равишда дискриминант $d = -16(4a^3 + 27b^2) \neq 0$ва инвариант $j = 1728(4a)^3/d$ кўринишда бўлади. Тенгламанинг коэффициентлари a ва b маълум j инвариант бўйича қуйидагича аниқланади [20]:

бу ерда $k \equiv \frac{j}{1728-j} \pmod{p}$, $j \neq 0$, $j \neq 1728$

$$\begin{cases} a \equiv 3k \pmod{p} \\ b \equiv 2k \pmod{p} \end{cases}$$

Эллиптик эгри чизиқнинг G нуқтаси F_p: $G = (x_G, y_G)$ майдондан олинган (x_G, y_G) элементлар жуфтлиги билан аниқланади. Бу нуқтани ҳисоблашнинг ягона аниқ усули йўқ. Шунинг учун танлаш усули билан – бирор x – қиймат олинади ва F_p майдонда $x^3 + ax + b$ ифоданинг қиймати ҳисобланиб, бу қиймат бирор сонни квадрат илдизи бўлиши ёки бўлмаслиги текширилади. Агарда квадрат илдиз мавжуд бўлса, бу илдиз y деб олинади. Квадрат илдиз мавжудлиги Лежандр $\left(\frac{x^3+ax+b}{p}\right)$ символи ёрдамида текширилади [20].

Криптографик бардошли рақамли имзо тизимини олиш учун қуйидаги шартлар бажарилиши керак [20, 29]:

1) Эллиптик эгри чизиқ суперсингуляр бўлмаслиги керак, яъни $\#E(F_p) \neq p + 1$;

2) $p^k \neq 1 \pmod{n}$ барча $k \in \{1, ..., C\}$ лар учун, бу ерда C етарли катта сон, қайсики $F_p c$ майдонда дискрет логарифмлашни ҳисоблаш вақт нуқтаи назаридан мумкин бўлмасин (одатда $C = 20$ қилиб танланади);

3) Эллиптик эгри чизиқ аномал бўлмаслиги керак, яъни $\#E(F_p) \neq p$.

4) Эллиптик эгри чизиқнинг рационал координатали нуқталари сони m қуйидаги шартни қаноатлантирсин:

$\#E(F_p) = m$, $m = tq$, бу ерда q – катта туб сон, $t \in \mathrm{N}$ ва $t \geq 1$

Ҳозирда махсус чизиқлар синфларига мавжуд бўлган ҳужумлардан ҳимояланишнинг мавжуд усули – E – Эллиптик эгри чизиқни юқоридаги шартларни қаноатлантирувчи қилиб олишдан иборатдир.

Ушбу параметрлар фойдаланувчилар гуруҳи учун умумий бўлиши мумкин. Имзони генерация қилиш ва текшириш учун ишлатиладиган индивидуал параметрлар – махфий ва очиқ калитлар деб номланади.

Имзо қўйиш калити (махфий калит) – бу $[0; q]$ d интервалдаги ихтиёрий d сони.

Имзони текшириш калити (очиқ калит) – бу эллиптик эгри чизиқдаги $Q=[d]G$ нуқта.

Бундан ташқари рақамли имзо алгоритмида h – хэш-функциядан ҳам фойдаланилади.

§8.7. EC DSA рақамли имзо алгоритми

Имзони генерация қилиш алгоритми.

Бошланғич маълумотлар: M – имзоланиши керак бўлган маълумот, берилган параметрлар ва имзо калити. Натижа: имзо (r, s).

1) $1 \le k \le n-1$ интервалдан тасодифий k сони танланланади, бу ерда G нуқта тартиби $n > \max\{2^{160}, 4\sqrt{p}\}$ шартни қаноатлантирувчи туб сон бўлиши керак.

2) $(x_1, y_1) := [k]G$ ҳисобланади.

3) $r := x_1 \bmod n$ ҳисобланади.

4) Агар $r=0$ бўлса, у ҳолда 1-қадамга борилади, акс ҳолда кейинги қадамга ўтилади.

5) $z := k^{-1} \bmod n$ ҳисобланади.

6) $e := h(M)$ ҳисобланади.

7) $s := z(e+dr) \bmod n$ ҳисобланади.

8) Агар $s=0$ бўлса, у ҳолда 1-қадамга борилади.

9) M – маълумот имзоси – (r, s) жуфтлигидан иборат.

Имзони текшириш алгоритми. Бошланғич маълумотлар: M – маълумот, берилган параметрлар, имзони текшириш калити ва M – маълумот имзоси. Натижа: имзо ҳақиқийлиги ёки қалбакилиги ҳақидаги тасдиқ.

1) Агар $1 \le r, s \le n-1$ шарт бажарилмаса, у ҳолда имзо қалбаки ва имзони текшириш алгоритми тугатилади.

2) $e := h(M)$ ҳисобланади.

3) $w := s^{-1} \bmod n$ ҳисобланади.

4) $u_1 := ew \bmod n$ ҳисобланади.

5) $u_2 := rw \bmod n$ ҳисобланади.

6) $X := [u_1]G + [u_2]Q = (x_1, y_1)$ ҳисобланади.

7) Агар $r = x_1 \bmod n$ шарт бажарилса, у ҳолда имзо ҳақиқий, акс ҳолда имзо қалбаки ва имзони текшириш алгоритми тугатилади.

Имзо алгоритмининг тўғрилиги. Исбот қилиниши керакки, имзо генерация қилиш алгоритми орқали қўйилган ихтиёрий имзо имзони текшириш алгоритми орқали текширилганда тўғри натижага эришилади.

Имзони генерация қилиш алгоритмидан олинган r ва s параметрлар n бутун сонга бўлингандаги қолдиқ сифатида олинганлиги учун $n-1$ кийматдан катта бўлмайди. Иккинчи томондан эса тўртинчи ва саккизинчи қадамларига кўра, бу параметрлар 0 дан фарқли кийматга эга бўлади. У ҳолда, имзони текшириш алгоритмининг биринчи қадамида берилган шарт r ва s параметрлар учун ҳар доим ўринли бўлиши шарт.

Имзони генерация қилиш алгоритмининг бешинчи ва еттинчи қадамларидан фойдаланиб, $ks \equiv e + dr (\mathrm{mod}\, n)$ муносабатга эга бўламиз. У ҳолда $w := s^{-1} \bmod n$ имзони текшириш алгоритмининг учинчи қадами) эканлигидан $ks \equiv we + wdr (\mathrm{mod}\, n)$ келиб чиқади. Алгоритм шартига кўра, G нуқта n тартибга эга, натижада:

$$[k]G = [we + wdr]G = [we]G + [wr][d]G = [we]G + [wr]Q = [u_1]G + [u_2]Q = X.$$

Шундай қилиб, имзони текшириш алгоритмининг 6-қадамидан олинадиган X нуқта $[k]G$ нуқтага тенг бўлади. X нуқтанинг биринчи координатаси x_1 ва унинг $\mathrm{mod}\, n$ бўйича қолдиғи r га тенг (имзони генерация қилиш алгоритмининг 3-қадамига кўра). Яъни, алгоритм тўғрилиги исботланди.

Мисол: F_p майдонда $y^2 = x^3 + 13x + 17$ тенгламага нисбатан параметрлар танланади. Майдон характеристикаси $p = 1000003$ га тенг бўлсин. Бошланғич нуқта сифатида $P(439; 1900)$ нуқта олинади. У ҳолда танланган эллиптик эгри чизиқнинг шу нуқтадаги тартиби алгоритм талабига кўра, туб *ва* $n = 500933$ га тенг. Энди алгоритмнинг кейинги қадамларига мувофиқ очиқ ва махфий калитлар танланади.

Имзо қўйиш калити (махфий калит) сифатида – $[0; n]$ интервалдаги ихтиёрий $d = 63289$ сонини оламиз.

Имзони текшириш калити (очиқ калит) сифатида эллиптик эгри чизиқдаги $q = [d]G$, яъни $Q = [63289]\ G = (168708; 576959)$ нуқта ҳисобланади.

Бундан ташқари рақамли имзо алгоритмида h – хэш-функциядан ҳам фойдаланилади. Ушбу мисолда хэш қиймат сифатида $h = 86347$ кийматдан фойдаланилади.

Имзони генерация қилиш қадамлари:

1) $1 \le k \le n-1$ интервалдан тасодифий $k = 100000$ сони танланади.

2) $(x_1, y_1) := [100000]\ G = (246423; 150077)$ ҳисобланади.

3) $r:=x_1 \bmod n = 246423 \bmod 500933 = 246423$ ҳисобланади.

4) $r=0$ шарт бажарилмаганлиги учун кейинги қадамга ўтилади. Бизнинг мисолда r нолдан фарқли. Ушбу қиймат ЭРИ нинг биринчи параметри сифатида қабул қилинади.

5) $z:=k^{-1} \bmod n = 100000^{-1} \bmod 500933 = 142280$ ҳисобланади.

6) $e:=h(m)$ ҳисобланади. Унинг қиймати $e:=h(m)=86347$, деб олинган.

7) Юқоридаги параметрлар асосида $s:=z(e+dr) \bmod n$ ҳисобланади. Натижада $s=278097$ олинади.

8) Агар $s=0$ бўлса, у ҳолда 1–қадамга борилади. Берилган мисолда бу қиймат нолдан фарқли ва у ЭРИнинг иккинчи параметри ҳисобланади.

9) M – маълумот имзоси – $(r, s)=(246423; 278097)$ жуфтлигидан иборат.

Энди ушбу имзони текшириш қадамлари билан танишиб чиқамиз. Очиқ калит сифатида $Q(168708; 576959)$ – нуқта олинган эди. Бундан ташқари имзоланган матнни ўзгармаган деб фараз қилиб, имзони текшириш қадамлари кўриб ўтилади.

Имзони текшириш қадамлари:

1) Аввало $1 \le r,\ s \le n-1$ шарт текширилади. Бу мисолда ушбу шарт бажарилгани учун кейинги қадамларга ўтилади.

2) $e=h(M)$ ҳисобланади ва матн жўнатилиш жараёнида ўзгартирилмаганлиги учун $e:=86347$ га тенг бўлади.

3) $w:=s^{-1} \bmod n = 278097^{-1} \bmod 500933 = 32706$ ҳисоблаб топилади.

4) $u_1:=ew \bmod n = 86347 \cdot 32706 \bmod 500933 = 305661$ ҳисобланади.

5) $u_2:=rw \bmod n = 246423 \cdot 32706 \bmod 500933 = 500534$ ҳисобланади.

6) Юқоридаги параметрлар асосида $X:=[u_1]G+[u_2]Q=(x_1, y_1)$ ҳисобланади, яъни $X:=[305661]\ G+[500534]Q$ ҳисобланади. Бунда $[305661]\ G=(977993; 407823)$, $[500534]\ Q=(641345; 818591)$, ва $X:=(246423; 150077)$.

7) $r=x_1 \bmod n = 246423$ шарт текширилади. Яъни $246423 \bmod 500933 = 246423$ имзонинг биринчи параметри билин таққосланади. Келтирилган мисолда бу иккала қиймат ҳам $r=x_1 \bmod n = 246423$ га тенг ва «**имзо ҳақиқий**» деган хулосага келинади. Агар электрон ҳужжат узатилиш жараёнида ўзгартирилган бўлса, у ҳолда имзони текширишдаги хэш қиймат имзо қўйишдаги хэш қийматдан фарқланганлиги ҳисобига, 7-шарт бажарилмайди ва «**имзо қалбаки**» деган хулосага келинади.

Қуйида EC DSA каби, ГОСТ Р 34. 10–94 ЭРИ алгоритмининг эллиптик эгри чизиққа ўтказилган модификацияси ҳисобланган Россия стандарти ГОСТ Р 34.10-2001 алгоритми ҳақида тўхталинади.

§8.8. ГОСТ Р 34.10-2001 электрон рақамли имзо алгоритми

Имзони генерация қилиш алгоритми. Бошланғич маълумотлар: M маълумот, берилган (эллиптик чизиққа алоқадор) параметрлар ва имзо махфий калити. Ушбу алгоритмда Эллиптик эгри чизиқ тенгламаси $p>2^{255}$ шартни қаноатлантирувчи туб характеристикали F_p майдонда деб қаралди. Натижа, имзо (r, s)

Имзони генерация қилиш алгоритми қадамлари:

1. $1 \le k \le n-1$ интервалдан ихтиёрий k сони танлансин, бу ерда G нуқта тартиби $2^{254}<n<2^{256}$ шартни қаноатлантирувчи сон.

2. $(x_1, y_1)= [k]G$ ҳисоблансин, яъни танланган эгри чизиққа тегишли G нуқтани k марта қўшилсин.

3. $r=x_1 \bmod n$ ҳисоблансин. Агар $r=0$ бўлса, 1-қадамга қайтилсин ва бошқа k сони танлансин.

4. M маълумотнинг хэш-функцияси ҳисоблансин, яъни $e=H(M)$. Агар $H(M) \bmod n=0$ бўлса, у ҳолда $H(M) \bmod n=1$ деб олинсин.

5. $0<d<n$ интервалдан олинган d махфий калит асосида $s=(dr+ke) \bmod n$ ҳисоблаб топилсин.

6. Агар $s=0$ бўлса, 1-қадамга қайтилсин ва бошқа k сони танлансин.

7. Ҳосил бўлган (r, s) сонлар жуфтлиги M маълумотга қўйилган имзо ҳисобланади.

Имзони текшириш алгоритми. Бошланғич маълумотлар M маълумот, берилган (эллиптик чизиққа алоқадор) параметрлар, имзони текшириш калити ва M маълумот имзоси – (r, s). Натижа: имзо ҳақиқийлиги ёки қалбакилиги ҳақидаги тасдиқ .

Имзони текшириш алгоритми қадамлари:

1. Агар $1 \le r, s \le n-1$ бажарилмаса, у ҳолда имзо қалбаки ва текширишни шу ерда тўхтатиш мумкин.

2. $e= H(M)$ ҳисоблансин.

3. $w= H(M)^{(n-2)} \bmod n$ ҳисоблансин.

4. $u_1 =sw \bmod q$ ҳисоблансин.

5. $u_2 =(n-r)w \bmod n$ ҳисоблансин.

6. $X=[u_1]G+[u_2]Q=(x_1, y_1)$ ҳисоблансин.

7. Агар $x_1 \bmod n=r$ бўлса, имзо ҳақиқий, акс ҳолда имзо қалбаки ва алгоритм тўхтатилади.

ГОСТ Р 34.10-2001 имзо алгоритмининг тўғрилиги. Исбот қилишимиз керакки, келтирилган алгоритм асосида қўйилган ихтиёрий имзо шу алгоритм билан текширилганда ҳар доим ҳақиқий.

Имзони шакллантириш жараёнига мувофиқ r ва s параметрлар n бутун сонга бўлишдаги қолдиқ сифатида олинганлиги сабабли $n-1$ қийматдан ошиб кетмайди. Имзони генерация қилиш алгоритмидаги 3 ва 6-қадамларга кўра $r, s \neq 0$ шарт ҳам ҳар доим бажарилади. Шунга кўра, имзони генерация қилиш алгоритмидан олинган r ва s параметрлар имзони текшириш алгоритмининг 1-қадамидаги шартни қаноатлантиради.

Имзони генерация қилишнинг 5-қадамига кўра, $s = ke + dr \pmod n$ тенглик бажарилади. Бундан $(dr + ke - s)/n = t$ ёки $k = (nt + s - dr)e^{-1}$ эканлиги келиб чиқади. Охирги келтирилган учта тенглик ихтиёрий бутун манфий бўлмаган t лар учун эквивалент ва шу билан бирга $t = d$ да ҳам бажарилади. Агар $t = d$ бўлса, охирги тенглик $k = s\ e^{-1} + (n-r)de^{-1}$ кўринишга келади. Хэш-функция қиймати $0 < e < n$ ораликда ётади ва n нинг тублигидан ЭКУБ$(e, n) = 1$ экани келиб чиқади. У ҳолда Эйлера – Ферма теоремасига кўра, $e^{-1} \equiv e^{\varphi(n)-1} \bmod n = e^{n-2} \bmod n$ бўлади. Ушбу таққосламадан фойдаланиб, k параметрни қуйидагича ифодалаш мумкин $k = s\ e^{-1} + (n-r)de^{-1} = s\ e^{n-2} \bmod n + (n-r)e^{n-2} \bmod n \cdot d$. Бу ердан $e^{n-2} \bmod n = w$ шартга кўра $k = sw + (n-r)wd$ ни олиш мумкин.

G нуқта n тартибга эга, яъни $[n]G = E$ ва ихтиёрий $k < n$ ларда $[k]G \neq E$ бўлади. Иккинчи томондан эса:

$$[k]G = [sw + (n-r)wd]G = [sw]G + [(n-r)w][d]G = [sw]G + [(n-r)w]Q = [u_1]G + [u_2]Q = X.$$

Демак, имзони генерация қилишдаги 6-қадамдан олинган X нуқта имзони генерация қилиш алгоритмидан олинган $[k]\ G$ нуқтага мос келади. Имзони генерация қилиш алгоритмининг 3 – қадамига кўра, X нуқтанинг биринчи координатаси x_1 бўлиб, унинг $\bmod n$ бўйича қолдиғи r га тенг, яъни $x_1 \bmod n = r$. Натижада алгоритм корректлиги келиб чиқади.

Мисол: F_p майдонда $y^2 = x^3 + 324x + 611$ тенгламага нисбатан параметрлар танлансин. $p = 8443$ га тенг бўлсин. Бошланғич нуқта сифатида $P(141; 65)$ нуқтани оламиз. Шуларга асосан ушбу эллиптик эгри чизиқнинг шу нуқтадаги тартиби $n = 8297$ тенг ва у тубдир. Умуман олганда ГОСТ Р 34.10-2001 рақамли имзо алгоритмида EC DSA-2000 рақамли имзо алгоритмидан фарқли ўлароқ, эллиптик эгри чизиқда олинган базавий G – нуқтанинг тартиби туб бўлиши шарт эмас. Фақат танланган p соннинг туб бўлишининг ўзи етарли. Кўрилаётган мисолда эллиптик эгри чизиққа тегишли бўлган базавий нуқтанинг n -тартибини туб бўлганлигини эса тасодифий ҳолат деб қараш мумкин. Қуйида имзо алгоритми шартига кўра очиқ ва махфий калитлар танланади.

Имзо қўйиш калити (махфий калит) сифатида – $[0; n]$ интервалдаги бирор $d = 725$ сони олинади.

Имзони текшириш калити (очиқ калит) – эллиптик эгри чизиқдаги $Q = [d]G$ нуқта, яъни $Q = [725]G = (914; 3289)$ нуқта ҳисобланади.

Бундан ташқари рақамли имзо алгоритмида h – хэш-функциядан ҳам фойдаланилади. Ушбу мисолда хэш қиймат сифатида $h = 459$ қиймат танланган.

Имзони генерация қилиш қадамлари:

1. $1 \le k \le n-1$ интервалдан ихтиёрий $k = 1625$ сони танланади.

2. $(x_1, y_1) = [1625]G$ ҳисобланади, яъни танланган эгри чизиққа тегишли G нуқтани 1625 марта қўшиб, (5814; 5316) нуқтани ҳосил қилинади.

3. $r = x_1 \bmod n$ ҳисобланади. Агар $r = 0$ бўлса, 1-қадамга қайтилади ва бошқа k сони танланади. Кўрилаётган мисолда $r = 5814 \bmod 8297 = 5814$ га тенг.

4. M – маълумотнинг хэш-функцияси ҳисобланади, яъни $e = H(M)$. Агар $H(M) \bmod n = 0$ бўлса, у ҳолда $H(M) \bmod n = 1$ деб олинади. Қараётган мисолда бу қиймат $h = 459$ га тенг.

5. $0 < d < n$ интервалдан олинган d махфий калит асосида юқорида топилган параметрлардан фойдаланиб, $s = (dr + ke) \bmod n$ ҳисоблаб, топилади, яъни $s = (725 \cdot 5814 + 1625 \cdot 459) \bmod 8297 = 7716$.

6. Агар $s = 0$ бўлса, 1-қадамга қайтилади ва бошқа k сони танланади.

7. Ҳосил бўлган $(r, s) = (5814; 7716)$ – сонлар жуфтлиги M – маълумотнинг имзоси сифатида маълумотга қўшиб узатилади.

Имзони текшириш қадамлари. Қабул қилинган имзо $(r, s) = (5814; 7716)$ сонлар жуфтлигидан иборат бўлсин.

1. Агар олинган имзо $1 \le r, s \le n-1$ шартни қаноатлантирмаса, у ҳолда имзо қалбаки ва текширишни шу ерда тўхтатиш мумкин, лекин юқоридаги имзо ушбу шартни қаноатлантиради, шунинг учун навбатдаги қадамларни бажариш давом эттирилади.

2. $e = H(M)$ ҳисобланади, электрон ҳужжат узатилиш жараёнида ўзгартирилмаган деб қабул қилиниб, ҳужжатнинг хэш қиймати имзо қўйишдаги хэш қиймат билан айнан бир хил, яъни $e = 459$.

3. $w = H(M)^{(n-2)} \bmod n$ ҳисобланади, яъни $w = 459^{(8297-2)} \bmod 8297 = 2133$.

4. $u_1 = s\ w \bmod n = 7716 \cdot 2133 \bmod 8297 = 5277$ ҳисоблаб топилади.

5. $u_2 = (n-r)w \bmod n = (8297-5814) \cdot 2133 \bmod 8297 = 2753$ ҳисоблаб топилади.

6. Юқоридаги параметрлар асосида $X:=[u_1]G+[u_2]$ $Q=(x_1, y_1)$ ҳисобланади, яъни $X:=[5277]$ $G+[2753]Q$. Бунда [5277] $G=(6738; 6040)$ ва [2753] $Q=(3705; 2317)$ га тенг. Шунга кўра, $X:=(5814; 5316)$.

$r=x_1 \bmod n=5814$ шарт текширилади, яъни $5814 \bmod 8297=5814$ имзонинг биринчи параметри билан таққосланади. Бизнинг мисолда бу иккала қиймат ҳам $r=x_1 \bmod q=5814$ тенг ва «**имзо ҳақиқий**» хулосасига келинади. Агар электрон ҳужжат узатилиш жараёнида ўзгартирилган бўлса, у ҳолда имзони текширишдаги хэш қиймат имзо қўйишдаги хэш қийматдан фарқланганлиги ҳисобига 7-шарт бажарилмайди ва «**имзо қалбаки**» деган хулосага келинади.

§8.9. Мавжуд ҳисоблаш мураккаблик масалаларига асосланган ЭРИ алгоритми

Эътиборингизга ушбу китоб муаллифи томонидан мавжуд: характеристикаси катта сон бўлган чекли майдонда дискрет логарифмлашни, етарли катта сонни туб кўпайтувчиларга ажратиш ва эллиптик эгри чизиқ нуқталари устида амаллар бажаришга оид ҳисоблаш мураккаблик масалаларига асосланган ҳолда яратилган ЭРИ алгоритми ҳавола этилади.

Бирор M – маълумотни имзолаш учун, қуйидагилар амалга оширилади:

1) e – очиқ ва d – махфий калитлар $de \equiv 1 \bmod \varphi(n)$ таққосламадан ҳисоблаб олинади, бу ерда $n=p_1q_1$, p_1, q_1 – махфий тутилувчи етарли катта туб сонлар, $\bmod \varphi(n)$ – Эйлер функцияси, аниқлик учун $p_1>q_1$ бўлсин;

2) тасодифий k ва x сонлари танланади, улар ушбу $1<k$, $t<q$, q – туб сон ва $p_1<q_1$, $1<x<n$, ҳамда ЭКУБ $(x, n)=1$ шартларни қаноатлантиради;

3) g – параметр танланади, бу ерда $g<n$ ушбу ЭКУБ $(g, n)=1$ ва $g^q \bmod n \neq 1$ шартларни қаноатлантиради, ҳамда q-сони $\varphi(n)$ – Эйлер функцияси қийматини бўлувчиси эмас.

Очиқ калитлар сифатида ушбу:

1) $y=g^{axd} \bmod n$, бу ерда a –сонининг қиймати $ed-a\varphi(n)=1$ тенгликдан олинади;

2) $Q_1=[t]G$ ва $Q_2=[x]G$, бу ерда G – базавий нуқта танланган эллиптик эгри чизикда q – тартибга эга параметрлар қабул қилинади.

Имзо қуйидагича ҳисобланади:

1. Танланган тасодифий k – сони бўйича (бу сон махфий тутилади ва имзо ҳисоланиб бўлингандан сўнг дарҳол йўқотилади) $[k]G=(x_1, y_1)$ ҳисобланади.

2. $r=g^{x_1 d} \bmod n \bmod q$ ҳисобланади.

3. $\rho=g^{d} \bmod n$ ҳисобланади.

4. $s=[k^{-1}\ (H(M)\rho t+r\rho x)]\bmod q$ ҳисобланади.

5. $\gamma=(g^{-ax}\rho)\bmod n$ ҳисобланади.

Имзо сифатида: (r, s, γ) –учлик қабул қилинади.

Имзони текшириш қуйдагича амалга оширилади:

1. Агар $1 \le r,\ s<q$ ва $1 \le \gamma<n$ шартлар бузилса, уҳолда «имзо қалбаки» ва жараён тўхтатилади, аксинча бўлса кейинги босқичга ўтилади.

2. M – маълумотнинг $H(M)$ – хэш қиймати ҳисобланади, яъни $h=H(M)$.

3. $w=y^{e}\bmod n$ ҳисобланади.

4. $\beta=w\gamma \bmod n$, бу ерда $\beta=w\gamma \bmod n=\rho \bmod n=\rho$ чунки $\rho<n$

5. $u_1=[s^{-1}H(M)\beta]\bmod q$ ҳисобланади, бу ерда

$$u_1=[s^{-1}H(M)\beta]\bmod q=s^{-1}H(M)\rho-a_1 q.$$

6. $u_2=(s^{-1}r\beta]\bmod q$ ҳисобланади, бу ерда $u_2=(s^{-1}r\beta)\bmod q=s^{-1}r\rho-a_2 q$.

7. $[u_1]Q_1+[u_2]Q_2=(x_2, y_2)$ ҳисобланади.

8. Агар $u=\beta^{x_2} \bmod n \bmod q=r$ бўлса, у ҳолда имзо ҳақиқий, акс ҳолда қалбаки бўлади.

ЭРИ алгоритми тўғрилигининг исботи. Бунинг учун $[u_1]Q_1 + +[u_2]Q_2 = (x_2, y_2)=(x_1, y_1)=[k]G$ тенгликнинг ўринли эканлиги кўрсатилади. Ҳақиқатан ҳам, ушбу

$$s=(H(M)\rho t+r\rho x)k^{-1} \bmod q \text{ ифодадан}$$

$$k=[s^{-1}(H(M)\rho t+r\rho x)] \bmod q=s^{-1}H(M)\rho t+s^{-1}r\rho x] \bmod q=$$

$$=s^{-1}H(M)\rho t+s^{-1}r\rho x-a_3 q \text{ топилади.}$$

У ҳолда:

$$[k]G=[s^{-1}H(M)\rho t+s^{-1}r\rho x-a_3 q]G=[s^{-1}H(M)\rho][t]G+[s^{-1}r\rho][x]G-$$

$$-[a_3][q]G=[u_1]Q_1+[u_2]Q_2 \text{ тенглик ўринли.}$$

Иккинчи томондан эса:

$$[u_1]Q_1+[u_2]Q_2=[s^{-1}H(M)\rho-a_1 q][t]G+[s^{-1}r\rho-a_2 q]G=$$

$$=[s^{-1}H(M)\rho t]G+[s^{-1}r\rho x]G-[a_1 t+a_2 x][q]G=[s^{-1}H(M)\rho t+s^{-1}r\rho x]G=$$

$$=[s^{-1}(H(M)\rho t+r\rho x)]G=[k]G.$$

Шундай қилиб талаб қилинган исбот кўрсатилди.

Алгоритмнинг тузилишига кўра, ҳар бир ҳисоблаш мураккаблик турлари бўйича очиқ ва махфий калитлар генерация қилиниб, махфий калитлар билан имзони ҳисоблаш амалга оширилади, очиқ калитлар

билан эса имзо текширилади. Бундай тузилиш алгоритмнинг ўзига хослиги бўлиб, бардошлиликни тубдан оширади.

8-боб бўйича хулосалар

Ушбу бобда:

1. Электрон рақамли имзо моҳиятан электрон ҳужжатнинг юридик мақомини таъминловчи криптографик восита эканлиги ёритилди.

2. ЭРИ ахборот коммуникация тармоғида электрон ҳужжат алмашинуви жараёнида қуйидаги учта масала;

– электрон ҳужжат манбаини ҳақиқийлигини аниқлаш;

– электрон ҳужжат тўлалигини (ўзгармаганлигини) текшириш;

– электрон ҳужжатга рақамли имзо қўйган субъектни муаллифликдан бош тортмаслиги таъминланиши ҳал этиш имконини бериши изоҳланди.

3. ЭРИ тизимини яратиш йўналишлари:

1) очиқ калитли шифрлаш алгоритмларига асосланган;

2) симметрик шифрлаш алгоритмларига асосланган;

3) имзони ҳисоблаш ва уни текширишнинг махсус алгоритмларига асосланган рақамли имзо тизимларининг ўзига хос томонлари кўрсатилди.

4. Баъзи мавжуд ЭРИ алгоритмлари стандартлари: DSA, ГОСТ Р 34.10-94, ЕС DSA-2000 ва ГОСТ Р 34.10-2001 батафсил ёритилди.

5. Мавжуд ҳисоблаш мураккаблик масалалари композициясига асосланган янги ЭРИ алгоритми таклиф этилди.

IX БОБ
БАРДОШЛИ КАЛИТЛАР ИШЛАБ ЧИҚИШ (ГЕНЕРАЦИЯЛАШ)

§9.1. Бардошли калитлар ишлаб чиқиш асослари ва алгоритмлари

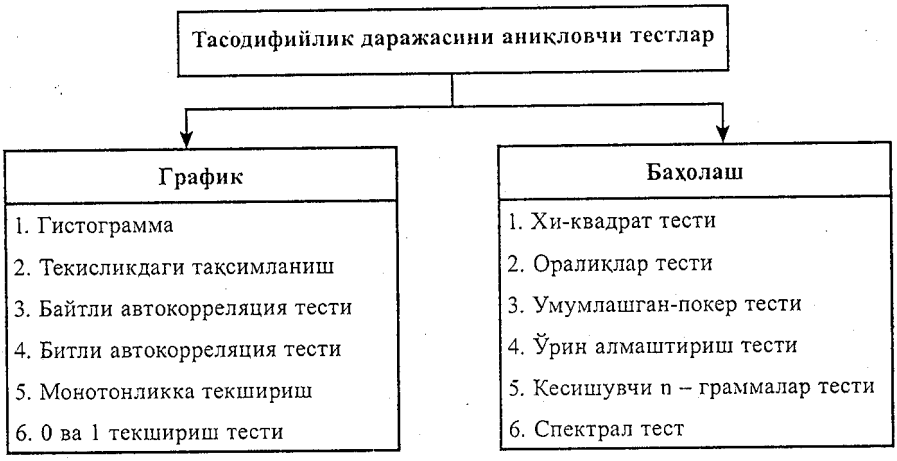
Мавжуд блокли симметрик шифрлаш алгоритмлари DES, AES, ГОСТ 28147-89 мос равишда 56 бит, 128 бит, ёки 256 бит, ёки 512 бит ва 256 бит узунликдаги олдиндан белгилаб қўйилган қоида бўйича генерация қилинган калитлардан фойдаланади. Бироқ стандарт алгоритмларда белгилаб қўйилган қоида бўйича генерация қилинган барча калитлар ҳар доим ҳам шифрматнни очиш мақсадида очиқ алоқа тармоғини назорат қилувчи криптоаналитик томонидан уюштириладиган турли криптоҳужумларга бардошли бўлмаслиги мумкин. Масалан, калитни ташкил этувчи битлар кетма-кетлиги фақат ноллардан ёки бирлардан ёки бўлмаса, нол ва бирларнинг комбинацияси фиксирланган давр билан такрорланиши ёрдамида тузилган бўлса, бу тоифа калитлар бардошсиз ҳисобланади. Чунки ушбу тур битлар кетма-кетлигида, шу кетма-кетликни ташкил этувчи нол ва бир элементлари даврий такрорланишининг математик қонуниятини олдиндан айтиш имконияти мавжуд. У ҳолда бу каби генерация қилинган битлар кетма-кетлигидан симметрик шифрлаш алгоритмлари учун махфий калит сифатида фойдаланиш мақсадга мувофиқ эмас. Демак, юқоридаги фикр-мулоҳазалардан келиб чиқиб, «криптоалгоритмлар махфий калит блоклари учун тасодифий битлар кетма-кетлиги қандай қурилади?» деган саволнинг туғилиши табиий, яъни агар бирор қоида бўйича калит блокининг $k=k_1k_2...k_m$, кетма-кетлиги олинган бўлса, бу ерда $k_i \in \{0,1\}$, ва $m=56$, 128, 192, 256 бўлиши мумкин. У ҳолда $k=k_1k_2...k_m$, калит блокида k_i – битларнинг тақсимоти тасодифий ёки тасодифий эмаслиги қандай аниқланади? Ушбу саволга жавоб олиш учун калит блокида k_i – битларнинг тақсимотини амалиётда кенг тарқалган ва бошқа мавжуд тасодифийлик тестларининг асосларини ташкил этувчи «Хи-квадрат» тақсимотидан фойдаланиб аниқлаш керак бўлади.

Тасодифийликка текширувчи тестлар 2 хил бўлади:

График тестлар – График тестлар фойдаланувчига текширилаётган кетма-кетликнинг маълум бир график боғлиқлиги ҳақидаги

маълумотни бериб, у бўйича текширилаётган кетма-кетлик хоссалари тўғрисида хулоса чиқариш имкониятини беради.

Баҳолаш тестлари – Баҳолаш тестлари текширилаётган кетма-кетлик статистик хоссаларини таҳлил қилиб, унинг чин тасодифийлик даражаси ҳақида хулоса чиқариш имкониятини беради [14, 20, 28]:



Калит блокини ташкил этувчи белгилар тақсимотини тасодифийликка текширишда, аввало, бу калит блокини бирор қоида бўйича ҳосил қилиб олиш зарур. Бу каби ишлар, одатда, псевдотасодифий кетма-кетликлар генераторлари орқали амалга оширилади. Псевдотасодифий кетма-кетлик ишлаб чикарувчи генераторлар ҳақида, уларнинг тузилиш асосларига кўра туркумлари, хусусиятлари, хоссалари, криптографик масаларни ечишдаги қўлланишлари V бобда батафсил таҳлил қилинган. Хусусан:

1) Чизиқли конгруэнт;

2) Квадратик конгруэнт;

3) Бир томонлама уникацияларга, шифрлаш ва хэшлаш алгоритмларига асосланган;

4) Сонлар назарияси муаммоларига асосланган генераторлар таҳлил қилинган.

Бундан ташқари, V бобда тасодифийлик даражаси етарли юқори ва акслантиришлари криптоҳужум турларига бардошли ва самарали:

1) Дифференциал ва чизиқли криптотаҳлил усулларига бардошли бўлган 256 байтли S – блок ва 16×16 ўлчамли сиқиш жадвали (СЖ) акслантиришлари асосида;

2) Иккита устуни пропорционал ва барча элементлари ҳар-хил бўлган ўлчами $2^r \times 4$ бўлган тўғри тўртбурчакли $A_{2^r \times 4}$ – матрица, ҳамда, ўлчами 16×6, бўлиб, элементлари ярим байтдан иборат бўлган (0 дан 15 гача сонларни текис тақсимотидан иборат) сиқиш жадвали (СЖ) акслантиришлари асосида;

3) Тўртта 4 аргументли мантиқий функция ва ўлчами 16×16, элементлари ярим байтдан иборат бўлган (0 дан 15 гача сонларни текис тақсимотидан иборат) сиқиш жадвали (СЖ) акслантиришлари асосида;

4) Байтлар ва битлар ўрнини боғлиқсиз алмаштиришга асосланган псевдотасодифий кетма-кетлик ишлаб чиқарувчи генераторларни 5 марта комбинациялашга асосланган (бошқа синфга тегишли бўлган генератор ҳам олиш мумкин) янги генераторлар ишлаб чиқилган ҳамда уларнинг криптобардошли узулуксиз шифрлаш алгоритмлари сифатида қўлланилиши мумкинлиги илмий асосланган.

Қуйида мисол сифатида бир томонлама функцияларга асосланган псевдотасодифий кетма-кетлик ишлаб чиқарувчи генераторлар келтириб ўтилади [2, 14, 20]:

1) **ANSI X9.17 генератори.** Бу алгоритм АҚШ да псевдатасодифий кетма-кетлик ишлаб чиқувчи Миллий стандарт ҳисобланиб, FIPS (USA Federal Information Processing Standart) таркибига киради. Алгоритмда бир томонлама функция сифатида учлик DES иккита $K1$, $K2 \in V64$ калит ишлатилади: DESK1DESK2 DESK1(64 бит) .

2) **FIPS-186 генератори.** Бу алгоритм ҳам АҚШ Миллий стандарти сифатида қабул қилинган бўлиб, DSA электрон рақамли имзо алгоритми махфий параметрларини ва калитларини генерация қилиш учун мўжалланган. Алгоритм бир томонлама функция сифатида DES шифрлаш алгоритми ва SHA-1 хэшлаш алгоритмини ишлатади.

3) **Yarrow-160 генератори.** Yarrow-160 псевдатасодифий кетма-кетлик ишлаб чиқарувчи генератори Келси, Шнайер ва Фергюсон томонидан таклиф қилинган. Бу ерда учлик DES ва SHA-1 хэшлаш алгоритми ишлатилган.

Сонлар назарияси муаммоларига асосланган генераторлар сифатида [2, 14, 20]:

1) RSA алгоритми асосидаги;

2) Микали-Шнорр RSA алгоритми асосидаги;

3) BBS (Blum-Blum-Shub) – алгоритми асосидаги генераторларни келтириш мумкин.

Агар чизиқли ва мультипликатив конгруэнт генераторлар билан аниқланган сонлар кетма-кетлиги учун z_n, z_{n+1} – битлари маълум бўлса, у ҳолда ҳосил қилинган кетма-кетликнинг қолган ҳадларини топиш имконияти мавжуд [14, 20].

Сонлар назариясининг муаммоларига (туб кўпайтувчиларга ажратиш ва дискрет логарифмлаш) асосланган генераторлардан симметрик шифрлаш алгоритмлари бардошли калитларининг генерация қилинишида фойдаланиш мақсадга мувофиқ, чунки бу генераторлардан фойдаланиб, ҳосил қилинган кетма-кетликнинг ҳадларини бирор қисмини билган ҳолда ундан олдинги ёки кейинги қисмларини аниқлаш имконияти мураккаб масала ҳисобланади.

Биз бундан кейинги фикр-мулоҳазаларимизда, бирор танланган псевдотасодифий кетма-кетликлар генератори орқали керакли узунликдаги калит блоки генерация қилиб олинган деб ҳисоблаймиз.

§9.2. Тақсимотни тасодифийликка текширишнинг «Хи-квадрат» критерийси

Бирор ўтказилаётган тажриба натижаларининг барча мумкин бўлган ҳолатлари $y_1, y_2, ..., y_k$ дан иборат ва уларнинг сони k га тенг бўлиб, бу тажриба бир-бирига боғлиқсиз ҳолда n марта ўтказилсин. Шунда, $y_1, y_2, ..., y_k$ – ҳолатларни, уларнинг n марта ўтказилган тажрибада, бир хил сонда такрорланишидан (текис тақсимотдан ёки бир хил частотага эга бўлишдан) қанчалик четланганлигини баҳолаш масаласини ечилишини кўриб ўтилади. Бунинг учун қуйидагича белгилашлар киритилади:

P_s – эксперимент натижаси y_s бўлишининг эҳтимоллик қиймати;

Y_s – эксперимент натижаларининг y_s ҳолатга тегишлилари (тенглари) сони.

У ҳолда, бу белгилашларга нисбатан «Хи-квадарат» деб аталувчи тақсимот критерийси ушбу

$$V = \sum_{s=1}^{k} \frac{(Y_S - np_S)^2}{np_S},$$

формула орқали аниқланади.

Агар тажриба n мартадан бир неча марта ўтказилганда, ҳар доим $y_1, y_2, ..., y_k$ – ҳолатлар тенг Y_i мартадан такрорланса (текис тақсимланган ёки бир хил частотали бўлса), яъни $Y_1, Y_2, ..., Y_k$ бўлса, у ҳолда $p_1 = p_2 = ... = p_k = \frac{1}{k}$ деб ҳулоса қилинади ва

$$V = \sum_{s=1}^{k} \frac{\left(Y_S - \frac{n}{k}\right)^2}{\frac{n}{k}} = \sum_{s=1}^{k} \frac{\left(\frac{n}{k} - \frac{n}{k}\right)^2}{\frac{n}{k}} = 0$$

тенглик ўринли бўлади. Бундай жараённинг илмий тадқиқот учун қизиғи йўқ. Аммо, амалдаги аксарият жараёнларда бундай ҳолат кузатилмайди, яъни бирор тажриба бир-бирига боғлиқсиз равишда n марта ўтказилганда: $Y_1,=Y_2,=...,=Y_k=\frac{n}{k}$ ҳолат кўзатилмайди. Шунинг учун y_1, y_2, ..., y_k ҳолатларни рўй бериш эҳтимолликлари бир хил $p_1,=p_2,=...,=p_k=\frac{1}{k}$ бўлиб, тажриба бир-бирига боғлиқ бўлмаган равишда n марта ўтказилганда, бу ҳолатларнинг рўй бериши сони мос равишда Y_1, Y_2, ..., Y_k бўлса, у ҳолда ушбу

$$V=\sum_{s=1}^{k}\frac{\left(Y_S-\frac{n}{k}\right)^2}{\frac{n}{k}}=\frac{k}{n}\sum_{s=1}^{k}\left(Y_s-\frac{n}{k}\right)^2$$

формула $Y_1,=Y_2,=...,=Y_k=\frac{n}{k}$ бўлган тенг тақсимотдан Y_1, Y_2, ..., Y_k – тенг бўлмаган тақсимотни ўртача квадратик четланишини ифодалайди. Бу охирги формуладаги $\left(Y_s-\frac{n}{k}\right)$ – ифода бирор ўзгармас сон билан чегараланган, яъни $\left|Y_s-\frac{n}{k}\right|\le C=\text{const}$.

Шунинг учун

$$V=\sum_{s=1}^{k}\frac{\left(Y_S-\frac{n}{k}\right)^2}{\frac{n}{k}}=\frac{k}{n}\sum_{s=1}^{k}\left(Y_s-\frac{n}{k}\right)^2\le\frac{k}{n}\sum_{s=1}^{k}C^2=\frac{(kC)^2}{n}\to 0,\quad \text{агар}\quad n\to\infty$$

бўлса.

Бу охирги формуладан, бирор генератор орқали ҳосил қилинган псевдотасодифий кетма-кетликнинг даври етарли узун бўлиб, барча мумкин бўлган битлар, байтлар ва қисм блокларининг тақсимоти деярли текис (тенг тақсимланган) бўлса, у ҳолда «Хи-квадарат» тақсимот критерийсининг бу кетма-кетликка нисбатан қиймати нолга яқин бўлиб, унинг тасодифийлик даражаси юқори ҳисобланади.

Қуйида эса стандарт DES, ГОСТ 28147-89 и AES-FIPS-197 ва бошқа симметрик шифрлаш алгоритмлари учун махфий калитни тасодифий қилиб генерация қилишнинг Хи-квадрат тақсимоти орқали қандай амалга оширилишини кўриб ўтамиз.

Берилган калит блоки бўйича қуйидаги жадвални тузиб оламиз:

Қиймат (s): 0 1;

Эҳтимоллик (p_s): $\frac{1}{2}$ $\frac{1}{2}$;

Кузатилаётган сон (Y_s): N_0 N_1,

бу ерда: N_0 ва N_1 мос равишда калит блокида иштирок этувчи ноллар ва бирлар, $N_0+N_1=n$, орқали калит узунлигини белгилайди, масалан $n=256$;

Кутилаётган сон (np_s): $\frac{n}{2}\ \frac{n}{2}$;

Хи-квадрат тақсимоти формуласи бўйича [10]:

$$V=\sum_{s=0}^{k-1}\frac{(Y_s-np_s)^2}{np_s}$$ ҳисобланади

Ушбу қаралаётган ҳолатда:

$k=2$; $s=0,1$; $p_0=p_1=\frac{1}{2}$; $Y_0=N_0$; $Y_1=N_1$; $n=256$; у ҳолда, қуйидагича катталикка эга бўламиз:

$$V=\frac{(N_0-128)^2+(N_1-128)^2}{128}$$

Бу катталикни ҳисоблаш учун бизга Хи-квадрат тақсимотининг критик нуқталари жадвали деб аталувчи жадвал керак бўлади.

	p=1%	p=5%	p=25%	p=50%	p=75%	p=95%	p=99%
$N=1$	0.00016	0.00393	0.1015	0.4549	1.323	3.841	6.635
$N=2$	0.02010	0.1026	0.5754	1.386	2.773	5.991	9.210
$N=3$	0.1148	0.3518	1.213	2.366	4.108	7.815	11.34
$N=4$	0.2971	0.7107	1.923	3.357	5.385	9.488	13.28
$N=5$	0.5543	1.1455	2.675	4.351	6.626	11.07	15.09
$N=6$	0.8721	1.635	3.455	5.348	7.841	12.59	16.81
$N=7$	1.239	2.167	4.255	6.346	9.037	14.07	18.48
$N=8$	1.646	2.733	5.071	7.344	10.22	15.21	20.09
$N=9$	2.088	3.325	5.899	8.343	11.39	16.92	21.67
$N=10$	2.558	3.940	6.737	9.342	12.55	18.31	23.21
$N=11$	3.053	4.575	7.584	10.34	13.70	19.68	24.72
$N=12$	3.571	5.226	8.438	11.34	14.85	21.03	26.22
$N=15$	5.229	7.261	11.04	14.34	18.25	25.00	30.58
$N=20$	8.260	10.585	15.45	19.34	23.83	31.41	37.57
$N=30$	14.95	18.49	24.48	29.34	34.80	43.77	50.89
$N=50$	29.71	34.76	42.94	49.33	56.33	67.50	76.15
$N>30$	$v+\sqrt{2v}x_p+\frac{2}{3}x_p^2-\frac{2}{3}+O\left(\frac{1}{\sqrt{v}}\right)$						
$x_n=8$	–2.33	–1.36	–0.674	0.00	0.674	1.64	2.33

«Хи-квадрат» критерийси жадвали $v=k-1=2-1=1$, сатридан V қиймат жойлашиш оралиғини топамиз. Агар V қиймат жадвал устунининг $p=25\%$ дан $p=25\%$, оралиғида бўлса, у ҳолда псевдотасодифий генератор ёрдамида ҳосил қилинган калит блок битлари кетма-кетлиги тасодифий деб олинади.

Гарчанд псевдотасодифий генератор ёрдамида ҳосил қилинган калит блок битлари кетма-кетлиги тасодифийликка «Хи-квадрат» критерийси бўйича текширилганда ижобий жавоб олинган бўлса ҳам, ундан кўра ишончли ва мукаммал бўлган жавоб олиш учун қаралаётган битлар кетма-кетлигини бошқа мавжуд тасодифийлик тестларига ҳам текшириб кўриш лозим. Бу критерийларга текширув натижаларида қанчалик кўп ижобий жавоблар олинса, критерий шунчалик яхши натижа деб қаралади. Бундан ташқари қуйидаги жараён ҳам тасодифийликка текширишда чиқариладиган хулосанинг ижобийлигига сезиларли даражада таъсир кўрсатади, яъни псевдотасодифий генератор ёрдамида ишлаб чиқилган калитларнинг амалиётда ўрнатилган бардошсиз калитлардан ўртача квадрат четланишининг ўртача қийматини ифодаловчи жараён.

Айтайлик, псевотасодифий генератор ёрдамида ҳосил қилинган калит блоки:

$$k=k_1 k_2 \ldots k_n=k_1 k_2 \ldots k_{256},\ \text{бу ерда}\ k_i \in \{0;\ 1\},\ i=1,\ 2,\ \ldots,\ n=256$$

юқорида келтирилган критерий бўйича тасодифийликка текширилган ва қониқарли жавоб олинган. Амалиёт жараёнида шифртизимлар билан ишлашда аниқланган бардошсиз калитларни $k=k_{н1} k_{н2} \ldots k_{нт}$, каби белгилаймиз.

Псевотасодифий генератор ёрдамида ҳосил қилинган калит блоки: $k=k_1 k_2 \ldots k_n=k_1 k_2 \ldots k_{256}$ ва амалиёт жараёнида бардошсиз деб топилган $k=k_{н1} k_{н2} \ldots k_{нт}$, калитларнинг фарқи кўриб ўтилади:

$r_1=k_{н1} \oplus k=r_1(1)\ r_2(1)\ \ldots\ r_{256}(1)$, бу фарқ бўйича мос равишда 0 ва 1 битлар сони $N_0(1)$, $N_1(1)$;

$r_2=k_{н2} \oplus k=r_1(2)\ r_2(2)\ \ldots\ r_{256}(2)$, бу айирма бўйича мос равишда 0 ва 1 битлар сони $N_0(2)$, $N_1(2)$;

$r_m=k_{нт} \oplus k=r_1(m)\ r_2(m)\ \ldots\ r_{256}(m)$, бу айирма бўйича мос равишда 0 ва 1 битлар сони $N_0(m)$, $N_1(m)$; бу катталиклардан фойдаланган ҳолда, қуйидагиларни ҳисоблаймиз:

$$V_1=\frac{(N0(1)-128)^2+(N_1(1)-128)^2}{128};$$

$$V_2 = \frac{(N0(2) - 128)^2 + (N_1(2) - 128)^2}{128};$$

$$V_m = \frac{(N0(m) - 128)^2 + (N_1(m) - 128)^2}{128};$$

$$V = \frac{V_1 + V_2 + ... + V_m}{m}.$$

«Хи-квадрат» критерийси жадвали $v=k-1=2-1=1$, сатридан V – қиймат жойлашиш оралиғини топамиз. Агар V қиймат жадвал устунининг $p=25\%$ дан $p=25\%$, оралиғида бўлса, у ҳолда псевдотасодифий генератор ёрдамида ҳосил қилинган калит блок битлари кетма-кетлиги тасодифий деб олинади.

9-боб бўйича хулосалар

Ушбу бобда:

1. Бардошли калитлар ишлаб чиқиш асослари узликсиз шифрлаш алгоритмлари асослари каби псевдотасодифий кетма-кетлик ишлаб чиқарувчи генераторлар масалаларининг ечимлари билан боғлиқлилиги асосланди.

2. Псевдотасодифий кетма-кетлик блоклари белгиларини тасодифийликка текширувчи тестлар ҳақида фикр ва мулоҳазалар билдирилиб, уларнинг график ва баҳолаш тестларидан иборатлиги изоҳланди.

3. Бир томонлама функцияларга асосланган АҚШ да псевдатасодифий кетма-кетлик ишлаб чиқувчи Миллий стандарт ҳисобланувчи **ANSI X9.17, FIPS-186, Yarrow-160** генераторлари мисол сифатида келтириб ўтилди.

4. Бугунги кунда самарали ва криптобардошли деб ҳисобланувчи сонлар назарияси муаммоларига асосланган генераторлар сифатида RSA, Микали-Шнорр RSA, BBS (Blum-Blum- Shub) алгоритми асосидаги генераторлар мисол сифатида келтириб ўтилди.

5. Псевдотасодифий кетма-кетлик блокларининг белгилари ва белгилар бирикмаларини тақсимотини тасодифийликка текширишнинг баҳолаш тестлари асосини ташкил этувчи «Хи-квадрат» критерийсининг амалий қўлланилиш усули кўрсатилди.

X БОБ
КАЛИТЛАРНИ ТАҚСИМЛАШНИ БОШҚАРИШ АЛГОРИТМЛАРИ (ПРОТОКОЛЛАРИ)

§ 10.1. Калитларни бошқариш

Ахборот-коммуникация тизимида маълумотлар амашинувига мос келувчи криптографик тизимни яратиш билан бир қаторда шу тизимда калитлар бошқариш масаласини оптимал (қулай ва ишончли) ҳал этиш муҳим ўрин тутади. Чунки танланган криптотизим қанчалик мураккаб ва ишончли бўлмасин, барибир ундан амалда фойдаланиш жараёнлари калитларни бошқариш масаласи билан боғликдир. Агарда маълумотларнинг махфий алмашинуви оз сонли фойдаланувчилар доирасида бўлса, калитлар алмашинуви жараёнида ноқулайликлар туғилмайди. Аммо ахборот-коммуникация тизимида маълумотларнинг махфий алмашинуви юзлаб, минглаб ва хатто миллионлаб фойдаланувчилар доирасида бўлса (мисол учун модем ва Интернет алоқа тизимлари орқали банк, савдо-сотиқ, давлат аҳамиятига боғлиқ ҳамда бошқа муҳим соҳалардаги алоқа жараёни фойдаланувчилари доирасида) калитларни бошқаришнинг ўзига хос алоҳида муҳим масалалари келиб чиқади.

Калитлар ҳақидаги маълумот деганда ахборот-коммуникация криптотизимида мавжуд бўлган барча калитлар тўплами ва уларнинг муҳофазаси билан боғлиқ маълумотлар тушунилади. Агарда калитлар ҳақидаги маълумотларни етарли даражадаги ишончли муҳофазали бошқаруви таъминланмаса, табиийки, рақиб томонга ахборот-коммуникация тизимидаги деярли ихтиёрий маълумотни олиш учун тўла имконият туғилади.

Калитларни бошқариш жараёни қуйидаги учта муҳим бўлган:

– барча калитларнинг ўзаро боғлиқ ҳолда, яъни бир бутун ҳолда ишлаш жараёнини таъминлаш (калитлар генерацияси);

– калитлар тўпламининг мақсадли кенгайиб боришини таъминлаш (калитларларнинг тўпланиши);

– калитларларни фойдаланувчилар доирасида тақсимлаш (калитларларнинг тақсимланиши) жараёнларига аҳамият беришни талаб этади.

§10.2. Калитларнинг очиқ тақсимланиш алгоритми ҳақида

Яна У. Диффи ва М. Е. Хеллман бир томонлама функция сифатида таклиф этган ушбу

$$f(x)=\alpha^x(\mathrm{mod}p) \qquad (10.1)$$

p модуль бўйича дискрет даражага кўтариш функциясига тўхталамиз. Илгари таъкидланганидек, бу ерда: x – бутун сон бўлиб, 1 дан $(p-1)$ гача бўлган қийматларни қабул қилиши мумкин; p – етарли катта бўлган туб сон; α – бутун сон бўлиб, 1 p дан гача бўлган қийматларни қабул қилади ва унинг даражалари α, α^2, ..., α^{p-1} қандайдир тартибда 1, ..., $p-1$ қийматларни қабул қилади. Мисол учун, $p=7$, $\alpha=3$ бўлса, $\alpha=3$, $\alpha^2=2$, $\alpha^3=6$, $\alpha^4=4$, $\alpha^5=5$, $\alpha^6=1$ ифодаларга эга бўламиз.

Алгебрада мана шундай α сонини чекли $GP(p)$ майдоннинг содда элементи дейилади ва маълумки, бундай ҳар доим мавжуд бўлади.

Агарда $y=f(x)=\alpha^x$ бўлса, у ҳолда табиийки, бу фунцияга тескари функция

$$x=f^{-1}(y)=\log_a y) \qquad (10.2)$$

бўлиб, берилган y лар бўйича x қийматларни топиш дискрет логарифмларни топиш масаласи дейилади. Хаттоки, p нинг етарли катта бўлган қийматларида ҳам, мисол учун $p=2^{1000}$ бўлганда ҳам, 1000 дан кўп бўлмаган квадратга кўтариш ва кўпайтириш амалларини бажариб, $f(x)$ функцияни осон ҳисоблаш мумкин.

Агарда дискрет даражага кўтариш функцияси ҳақиқатан ҳам бир томонлама бўлса, у ҳолда $\log_a$ ифодани y нинг барча, яъни ушбу $1 \le y \le p$ тенгсизликни қаноатлантирувчи, барча қийматларида ҳисоблашни амалий жиҳатдан имконияти йуқ бўлиши керак. М.Е. Хеллман ва унинг шогирди Полиг, фақатгина p сони катта туб сон бўлгандагина эмас, балки $(p-2)$ сони катта туб купайтувчи q га эга (ёки шу q туб сон 2 га купайтирилган) бўлганда, (10.1) ифода билан аниқланган функциянинг y қийматларига кўра $\log_a y$ ифодани ҳисоблаш амалий жиҳатдан мураккаб эканлигини кўрсатдилар. У.Диффи ва М.Е. Хеллман махфий алоқа тизимлари фойдаланувчилари учун, дискрет логорифмлардан фойдаланиб, махфий калитларни ўзаро алмашувини алоҳида махфий каналсиз амалга ошириш алгоритмини яратдилар. Бу алгоритм бўйича:

1. α ва p сонлари ҳамма фойдаланувчиларга маълум.

2. Ҳар бир фойдаланувчи, масалан i – фойдаланувчи 1 билан $(p-1)$ сонлари оралиғидаги бирор бутун X_i сонини танлаб олади бу сонни махфий тутади.

3. i – фойдаланувчи $Y_i=\alpha^{X_i} \pmod p$ қийматни ҳисоблаб, бу Y_i қийматни махфий тутмай ҳамма фойдаланувчилар томонидан тасдиқланган ва улар ҳар доим фойдалана оладиган очиқ маълумотлар китобига киритади.

4. Агарда, махфий алоқа тизимининг i – фойдаланувчиси j – фойдаланувчи билан махфий алоқа ўрнатмоқчи бўлса, i – фойдаланувчи очиқ маълумотлар китобидан Y_i ни олиб, ўзининг махфий калити X_i ёрдамида

$$Z_{ij}=(Y_j)^{X_i}=(\alpha^{X_j})^{X_i}=\alpha^{X_iX_j} \pmod p$$

қийматни ҳисоблайди.

5. Худди шу каби j – фойдаланувчи ҳам Z_{ji} ни ҳисоблайди. Бунда $Z_{ij}=Z_{ji}$ бўлиб, i ва j фойдаланувчилар ўз махфий алоқаларини таъминловчи симметрик калитли криптотизимда Z_{ij} қийматни махфий калит сифатида ишлатишлари мумкин. Агар рақиб томон дискрет логарифмларни ҳисоблаш масаласини еча олса, очиқ маълумотлар китобидан Y_i ва Y_j ларни олиб $X_i=\log_\alpha Y_i$, ва $X_j=\log_\alpha Y_j$ қийматларни ҳисоблаб, Z_{ji} махфий калитга эга бўлган бўлар эди (i ва j – фойдаланувчилар каби).

Шу ерда таъкидлаб ўтиш жоизки, очиқ маълумотлар китоби ахборотларнинг махфий алоқа тизими фойдаланувчиларигагина очиқ.

Юқорида келтирилган алгоритмдан кўриниб турибдики, ҳали бу нарса назарий жиҳатдан тўла исботланган бўлмасада, рақиб томон Z_{ij} қийматни бошқа бирор услуб билан ҳисоблай олмайди. Келтирилган алгоритм У. Диффи ва М. Е. Хеллманнинг калитларни очиқ тақсимлаш тизими дейилади. Бу махфий алоқа тизимида махфий калитларни махфий канал билан узатишнинг ҳожати йўқлигини таъминловчи биринчи тизим бўлиб, бугунги кунда ҳам бардошли ва қулай очиқ калитли бошқа криптотизимларнинг асосини ташкил этади.

У. Диффи ва М.Е. Хеллманнинг калитларни очиқ тақсимлаш тизими очиқ калитли бошқа криптотизимлар каби махфий калитни махфий канал орқали узатилишининг ҳожати йўқлигини таъминлайди, аммо аутентификация масаласини ечмайди.

Махфий алоқа тизимида очиқ маълумотлар китобини сақловчи, махфий бўлмаган Y_i ни, очиқ маълумотлар китобига i – фойдаланувчининг фақат ўзи томонидангина киритилганига ишонч ҳосил қилиши керак, i – фойдаланувчи эса, ўз навбатида, Y_j ни

фақат очиқ маълумотлар китобини сақловчи томонидан берилганига ишонч ҳосил қилиши керак. Яъни очиқ калитлар тўплами ҳам муҳофаза қилиниши керак. Чунки, бирор субъект томонидан ноқонуний (руҳсатсиз) равишда очиқ калитлар тўпламига ўзининг очиқ калитини жойлаштириши унинг учун шу тизимга ноқонуний (руҳсатсиз) фойдаланиш имкониятига эга бўлганлигини таъминлайди. Шунинг учун ҳам сертификатланган калитлар тўплами умумфойдаланиш ахборот-коммуникация тизимида сақланмайди, у алоҳида фаолият кўрсатувчи компьютер ёки нисбатан кичик сондаги компьютерлар тизимида сақланади. Тизимнинг бирор i – фойдаланувчиси бирор j – фойдаланувчи билан муҳофазаланган алоқа ўрнатиш учун j – фойдаланувчининг очиқ калитига эга бўлиши керак. Бунинг учун:

1) Умумфойдаланиш тизимидаги барча фойдаланувчилар компьютерларига ва улар бевосита боғланган бош компьютерга ахборот муҳофазасининг криптографик усулларининг асосий воситалари бўлган шифрлаш, хэш-функция ва ЭРИ алгоритмларининг дастурий таъминотлари ўрнатилган бўлиб, бош компьютер администраторининг очиқ калити ҳамма фойдаланувчиларга маълум бўлади.

2) i – фойдаланувчи бош компьютер маъмурига j – фойдаланувчи билан алоқа ўрнатмоқчи эканлигини M – очиқ матнни бош компьютер маъмурининг k_A^0 – очиқ калити билан шифрлаган ҳолда $E_{k_A^0}(M)$ ҳамда маъмур бу маълумотни ва унинг авторини ҳақиқийлигига ишонч ҳосил қилиши учун, M – маълумот хэш-қийматини $h(M)$ ушбу $E_{k_A^0}(M) \cup h(M)$ кўринишда бирлаштириб ва ҳосил бўлган кенгайтирилган $M' = E_{k_A^0}(M) \cup h(M)$ маълумотни ўзи k_i^m – махфий калити билан шифрлаб $E_{k_i^m}(M') = C$ (ёки $M' = [M \cup P(k_i^m, h(M))]$ – кенгайтирилган маълумотни маъмурнинг очиқ калити k_A^0 билан шифрлаб $E_{k_A^0}[M \cup P(k_i^m, h(M))] = C)$ юборади.

3) Маъмур $C = E_{k_i^m}(M')$ шифрланган маълумотни k_i^0 – калит билан очади: $D_{k_i^0}(C) = D_{k_i^0}(E_{k_i^m}(M')) = M' = E_{k_A^0}(M) \cup h(M)$ Сўнгра маъмур ўзининг k_A^M – махфий калити билан $D_{k_A^m} = (E_{k_A^0}(M)) = M_1$ – очиқ маълумотга эга бўлади.

4) Бу олинган очиқ маълумот хэшланади $h(M_1)$, ҳамда $h(M_1) = h(M)$ тенглик текширилади. Агар тенглик ўринли бўлса, маълумот ва унинг муаллифи ҳақиқий, агар тенглик ўринли бўлмаса маълумот ва унинг муаллифи ҳақиқий эмас деган хулоса чиқарилади.

5) Агар маъмурга $C = E_{k_A^0}[M \cup P(k_i^m,\ h(M))]$ – шифрмаълумот юборилган бўлса, у ўзининг k_A^M – махфий калити билан бу маълумотни дешифрлайди: $D_{k^m}(C) = D_{k^m}\{E_{k_A^0}[M \cup P(k_i^m,\ h(M))]\} = M \cup P(k_i^m,\ h(M))$. Сўнгра $P(k_i^м,\ h(M))$ – ЭРИ тўғрилигини текширади, агар тўғри бўлса маълумот ва унинг муаллифи ҳақиқий, аксинча бўлса маълумот ва унинг муаллифи ҳақиқий эмас деб хулоса чиқарилади.

6) Юборилган маълумот ва унинг муаллифини (i – фойдаланувчини) ҳақиқийлиги ўрнатилгандан сўнг, маъмур i – фойдаланувчининг k_j^0 – очиқ калитини ва у билан боғлиқ бўлган (масалан амал қилиш вақти ва шу каби) бошқа M_j – маълумотларни алоҳида фаолият кўрсатувчи компъютердан олиб, бош компъютер орқали i – фойдаланувчининг k_i^0 – очиқ калити билан шифрлаб $E_{k_i^0}(M_j) = C_j$ ҳамда i – фойдаланувчи бу маълумотни ва унинг муаллифини ҳақиқийлигига ишонч ҳосил қилиши учун, M_j – маълумотнинг хэш – қийматини $h(M_j)$ ушбу $E_{k_i^0}(M_j) \cup h(M_j)$ кўринишда бирлаштириб ва хосил бўлган кенгайтирилган $M_j' = E_{k_i^0}(M_j) \cup h(M_j)$ – маълумотни ўзининг $k_A^м$ – махфий калити билан шифрлаб $E_{k_A^m}(M_j') = C_j'$ (ёки $M_j' = [M_j \cup P(k_A^m, h(M_j))]$ кенгайтирилган маълумотни i – фойдаланувчининг очиқ калити k_i^0 билан шифрлаб $E_{k_i^0}[M_j \cup P(k_A^m, h(M_j))] = C_j')$ очиқ алоқа канали орқали юборади.

7) i – фойдаланувчи $C_j' = E_{k_i^0}(M_j')$ шифрланган маълумотни k_A^0 – калит билан очади $D_{k_A^0}(C_j') = D_{k_A^0}(E_{k_A^m}(M_j')) = M_j' = E_{k_i^0}(M_j) \cup h(M_j)$. Сўнгра i – фойдаланувчи ўзининг $k_i^м$ – махфий калити билан $D_{k_i^m}(E_{k_i^0}(M_j')) = M1_j$ – очиқ маълумотга эга бўлади.

8) Бу олинган очиқ маълумот хэшланади $h(M1_j)$, ҳамда $h(M1_j) = h(M_j)$ тенглик текширилади. Агар тенглик ўринли бўлса, маълумот ва унинг муаллифи ҳақиқий, агар тенглик ўринли бўлмаса маълумот ва унинг муаллифи ҳақиқий эмас деган хулоса чиқарилади.

9) Агар i – фойдаланувчига $C_j = E_{k_м^0}[M_j \cup P(k_A^м, h(M_j))]$ шифрмаълумот юборилган бўлса, у ўзининг $k_i^м$ – махфий калити билан бу маълумотни дешифрлайди: $D_{k_i^м}(C_j) = D_{k_i^м}\{E_{k_i^0}\ [M_j \cup P(k_A^м, h(M_j))]\}$. Сўнгра $P(k_A^м, h(M_j))$ – ЭРИ тўғрилигини текширади, агар тўғри бўлса маълумот ва унинг автори ҳақиқий, аксинча бўлса маълумот ва унинг автори ҳақиқий эмас деб хулоса чиқарилади.

Шундай қилиб, i – фойдаланувчи j – фойдаланувчи билан очиқ алоқа тармоғида муҳофазаланган ахборот алмашинувини ўрнатиши учун j – фойдаланувчининг k_i^0 – сертификатланган очиқ калитига эга бўлди. Очиқ калитлар тўпламининг алоҳида компъютерда сақланиши ва очиқ калитларни 1) – 9) босқич жараёнларида тарқатилиши самарали криптографик муҳофазани ташкил этиш услубини ёки протоколини аниқлайди. Ҳақиқатан ҳам бундай ташкилий жарён фақат шифрлаш, хэшлаш ва ЭРИ алгоритмларидан фойдаланган ҳолда кафолатли муҳофазани таъминлашини тушуниш қийин эмас.

§10.3. Криптотизим фойдаланувчилари учун калитларни тақсимлашнинг тартиб ва қоидалари (протоколи)

Махфий услубли бир томонлама функцияга асосланган очиқ калитли криптотизимлар ўз моҳиятига кўра ундан фойдаланишнинг алоҳида тартиб ва қоидаларини (протоколини) талаб этади. Бу алоҳида тартиб ва қоидаларга кўра тизимнинг фойдаланувчилари ва тизим фойдаланувчиларигагина очиқ бўлган очиқ маълумотлар тўпламини (китобини) маъмури (сақловчиси) биргаликда шу тизимда узатиладиган маълумотларнинг махфийлигини таъминлайдилар.

Очиқ калитли криптотизимларнинг бардошлилигига тўла ишонч билдирмай ишончсизлик ва иккиланиш билан қарайдиган баъзи криптолог мутахассислар, фойдаланувчиларга муҳофазаланган услубда очиқ калитларни тақсимлаш ва махфий калитларни узатиш масалаларини, яъни калитлар билан боғлиқ жараёнларни мақсадли бошқаришни криптографиянинг бош амалий масаласи, деб биладилар. Мисол учун, агарда криптотизим фойдаланувчиларининг сони S та бўлса ва ҳар бир мумкин бўлган алоқа жуфтлари учун алоҳида махфий калит талаб этилса, уларнинг сони $c_s^2 = s(s-1)/2$ бўлиб, фойдаланувчилар сони кўп бўлган тизимлар учун бундай ҳолат баъзида мақсадга мувофиқ бўлмаслиги мумкин. Бирор фойдаланувчининг бошқа барча фойдаланувчиларга махфий бўлган маълумотни юбориши махфий алоқа моҳиятига зид жараён. Бундан ташқари махфий алоқа тизимида қайси фойдаланувчининг бошқа қайси бир фойдаланувчи билан махфий алоқа қилишни ҳоҳлаши олдиндан маълум эмас. Мана шундай ҳолатлар фойдаланувчиларга калитларни тақсимлаш тартиб ва қоидалари масалаларини келтириб чикаради. Бундай масалаларнинг ечилиши эса, ахборот-коммуникация тизимида маълумотларнинг махфийлиги муҳофазасини таъминловчи криптотизимда калитларни рўйхатга олиш маркази (КРОМ) ташкил этишни тақазо

этади. Калитларни тақсимлаш тартиб ва қоидалари (протоколи) қуйидагича:

1. КРОМ муҳофазаланган алоқа тармоғи орқали барча $i=1, 2, ..., S$ фойдаланувчиларга махфий Z_i калитларни тақдим этади.

2. Фойдаланувчи i фойдаланувчи j билан махфий алоқа ўрнатмоқчи бўлса, у умумий алоқа тармоғи орқали (очиқ матн билан бўлиши мумкин) КРОМга мурожаат қилиб, фойдаланувчи j билан махфий алоқа қилиш калитини сўрайди.

3. КРОМ махфий алоқа учун очиқ матннинг бирор қисмини ташкил этувчи Z_{ij} махфий калитни танлаб олади. Қолган қисмини i ва j фойдаланувчилар кўрсатилган «бош қисм» («заголовка») ёки «номланиш қисми» деб аталувчи бўлак ташкил этади. КРОМ бу очиқ матнни криптотизимда қабул қилинган шифрлаш алгоритмига кўра Z_i ва Z_j калитлар билан шифрлаб, умумий алоқа тармоғи орқали Z_i калит билан шифрланган криптограммани i фойдаланувчига ва Z_j калит билан шифрланган криптограммани j фойдаланувчига жўнатади.

4. Олинган криптограммаларни i ва j фойдаланувчилар дешифрлаб, кейинги олинган маълумотларни дешифрлашнинг махфий калитига эга бўладилар.

Калитларни тақсимлашнинг бундай тартиб ва қоидалари (протоколи) оддий бўлиб, унинг бардошлилиги шифрлаш алгоритмининг бардошлилиги билан белгиланади. Ҳақиқатдан ҳам 3-бандда (қадамда) келтирилганидек, криптоаналитикка ҳар-хил калитлар билан шифрланган бир хил очиқ матннинг криптограммаси маълум бўлиб, бундай ҳолат унга криптотаҳлил қилишда қўл келади. Шундай қилиб, очиқ матнни шифрлаш алгоритми криптотаҳлилга бардошли бўлса, калитларни тақсимлаш протоколи ҳам бардошли бўлади. Бу ерда шуни ҳам унутмаслик керакки, калитларни тақсимлашда шифрлаш алгоритмидан фойдаланиш шу тақсимлаш тартиб ва қоидаларининг бузилишига, криптобардошсизликка ва шу каби номутоносибликларга олиб келмаслиги керак.

§ 10.4. Симметрик шифрлаш алгоритми орқали калит алмашув протоколлари ва уларнинг криптоҳужумга заиф томонларини аниқлаш

Ушбу параграфда симметрик шифрлаш алгоритми ёрдамида генерация қилинган калитни алмашиш протоколлари кўриб чиқилади. Бу протоколларда ахборот алмашинуви субъектлари бўлган A ва B фойдаланувчилар умумий k_{AB} калитга эга деб қабул қилинади. Бу про-

токоллар, учинчи ишончли томоннинг иштирок этиши ёки этмаслигига боғлиқ равишда икки турга бўлинади. Аввало учинчи ишончли томон иштирок этмаган протоколларни кўриб ўтилади. Бунинг учун қуйидаги белгилашлар киритилади:

E – шифрлаш алгоритми;

t_A – вақт белгиси;

r_A – A – фойдаланувчининг тасодифий сони;

n_A – A – фойдаланувчининг генерация қилиш тартиб рақами;

B – B – фойдаланувчининг идентификацион рақами;

k_{AB} – иккала томонга ҳам маълум бўлган калит.

Симметрик калитли криптотизимда фойдаланувчилардан ташқари калитларни тарқатувчи томон, яъни калитларни тарқатиш маркази ҳам иштрок этади. Симметрик криптотизим ёрдамида калитларни алмашиш протоколи қуйидагича амалга оширилади:

1. A – фойдаланувчи B – фойдаланувчи билан алоқа ўрнатиш учун калит тарқатувчига мурожаат қилади ва сеанс калитини сўрайди.

2. Калит тарқатувчи сеанс калитни генерация қилади ва бу калитини икки нусхада шифрлаб, A – фойдаланувчига узатади.

3. A – фойдаланувчи ўзига тегишли шифрланган сеанс калитини дешифрлайди.

4. A – фойдаланувчи шифрланган сеанс калитининг иккинчи нусхасини B – фойдаланувчига узатади.

5. B – фойдаланувчи ўзининг шифрланган калитини дешифрлайди.

6. A ва B – фойдаланувчилар махфий алоқа учун юқорида ҳосил қилинган сеанс калитидан фойдаланадилар.

Бу протоколда сеанс калитлар тарқатувчини ишончли томон деб қабул қиладилар. Агар криптоаналитик актив ҳужум ёрдамида ёки бошқа қандайдир усул билан сеанс калитларини қўлга киритса, у ҳолда криптоаналитик алоқа тармоғига уланиб, тармоқдаги барча алмашинувчи махфий маълумотларни кузатиш ёки эшитиш имкониятига эга бўлади.

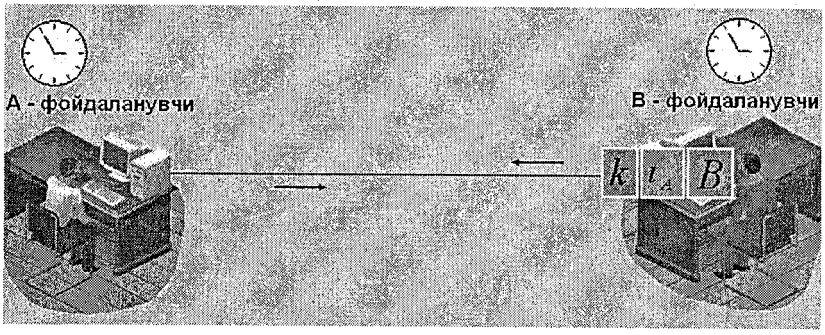
Юқорида баён қилинган тизимнинг яна бир камчилиги шундаки, ҳар бир калит алмашишда қатнашувчи учинчи томон, яъни калитларни тарқатиш маркази, мазкур тизимнинг нозик нуқтаси ҳисобланади. Агар унда бирор камчилик кузатилса, бутун тизимга таъсир этади. Қуйида шу каби бир нечта протоколлар ҳақида тўхталиб ўтилади.

1-протокол

Симметрик шифрлаш алгоритми ёрдамида генерация қилинган калитни узатиш протоколининг содда кўриниши – сеанс калитини бир раундда узатиш. Бутун протокол ягона маълумотдан ташкил топган:

$$A \rightarrow B: E_{k_{AB}}(r_A, t_A, B)$$

B – фойдаланувчи умумий калит ёрдамида бу маълумотни дешифрлайди. Бу ҳолда r_A – сеанс калит вазифасини бажаради [2]



Хулоса

Агар кўриб чиқилган протоколда:

1. Вақт белгиси узатилмаса, криптоаналитик айнан шу маълумотни қайта узатиши мумкин.

2. B – фойдаланувчининг идентификацион рақами кўрсатилмаса, криптоаналитик бу маълумотни A – фойдаланувчининг ўзига узатиши мумкин ва натижада A – фойдаланувчи маълумот B – фойдаланувчидан келган ёки келмаганлигини аниқлай олмайди.

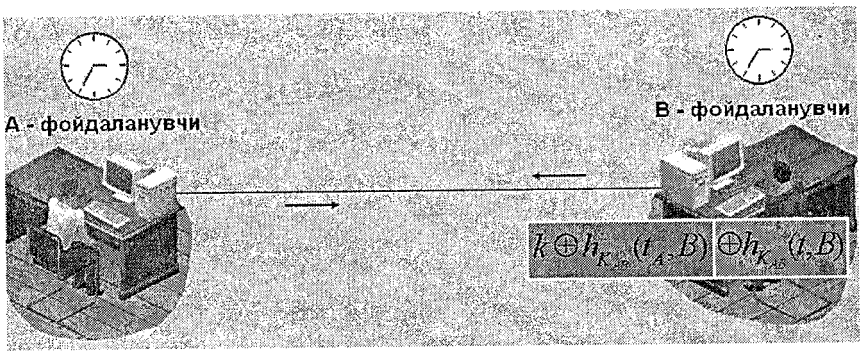
3. Сеанс калит $f(r_A, r_B)$ функция ёрдамида ҳисоблаб топилиши мумкин. Агар f функция сифатида бир томонлама функциядан фойдаланилса, томонларнинг ҳеч бири натижавий калитни назорат қила олмайди.

Юқорида келтирилган протоколда шифрлаш алгоритми ўрнига калит ёрдамидаги хэш-функциядан фойдаланиш мумкин:

$$A \rightarrow B: <t_A, r_A \oplus h_{k_{AB}}(t_A, B)>.$$

B – фойдаланувчи маълумотни қабул қилади. У ҳам калит орқали хэшлаш функциясини билади. Қабул қилган маълумотидан вақт белгисини ажратиб олади. Унинг кейинги вазифаси вақт белгиси ва ўзининг идентификация рақамини бирлаштириб калитли хэш-функция ёрдамида хэшлашни амалга ошириш. Чиққан $h_{k_{AB}}(t_A, B)$ нати-

жани қолган $r_A \oplus h_{k_{AB}}(t_A, B)$ маълумотга XOR амали бўйича қўшилади. Натижада r_A – сеанс калит ҳосил бўлади.



Агар тизим умумий синхрон вақтга эга бўлмаса, лекин калитнинг янгилигига ишонч ҳосил қилиш талаб қилинса, у ҳолда вақт белгисини тартиб рақам билан алмаштириш мумкин. У ҳолда протокол қуйидаги кўринишга келади:

2 – протокол

B – фойдаланувчи ўзининг n_B тасодифий сонини ҳосил қилиб уни А – фойдаланувчига узатади:

$$B \rightarrow A: n_B$$

A – фойдаланувчи бу тасодифий сонни қабул қилиб, унга ўзи ҳосил қилган сеанс калитини ва *B* – фойдаланувчининг идентификация рақамини бирлаштириб иккала фойдаланувчи учун умумий бўлган калит ёрдамида шифрлайди ҳамда *B* – фойдаланувчига узатади:

$$A \rightarrow B: E_{k_{AB}}(r_A, n_B, B).$$

В – фойдаланувчи n_B ва *B* ни текшириб, r_A – сеанс калитининг тўғри эканлигига ишонч ҳосил қилади. Хэш-функциядан фойдаланилса протоколнинг кўриниши қуйидагича бўлади:

$$B \rightarrow A: n_B$$
$$A \rightarrow B: r_A \oplus h_{k_{AB}}(n_A, B).$$

Ушбу протоколни шундай ўзгартириш мумкинки, натижада k= r_A – сеанс калитини бир томон эмас, балки иккала томон биргаликда генерация қиладилар [2].

A ва *B* – фойдаланувчилар r_A ва r_B – сонларидан бошқа тасодифий n_A ва n_B – сонларни генерация қиладилар. Бу ерда r_A ва r_B – сонлари калит материаллари сифатида фойдаланилади, n_A ва n_B – сонлари эса

калитнинг янги калит эканлигини таъминлайди. У ҳолда протокол қуйидагича амалга оширилади:

1) юқорида келтирилган протокол каби B – фойдаланувчи ўзининг n_B – тасодифий сонини A – фойдаланувчига узатади:

$$B \rightarrow A: n_B;$$

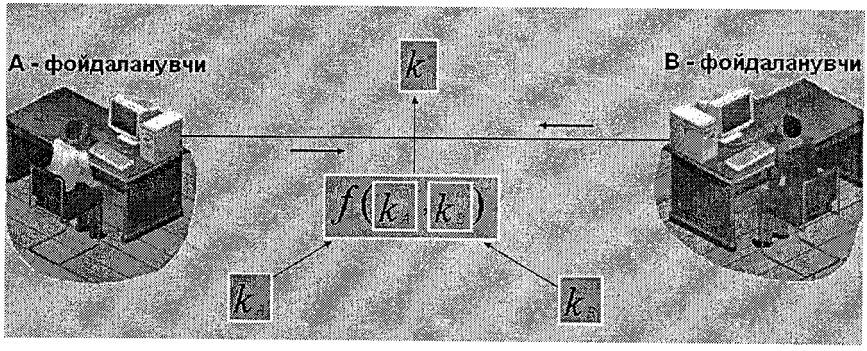
2) A – фойдаланувчи бу тасодифий сонни қабул қилади. Ўзаро аутентификацияни таъминлаш ҳамда сеанс калитни биргаликда ҳосил қилиш учун қуйидаги маълумотни B – фойдаланувчига узатади:

$$A \rightarrow B: E_{k_{AB}}(r_A, n_A, n_B, B);$$

3) B – фойдаланувчи маълумотни дешифрлаб, n_B – тасодифий сонни текширади. Натижа тўғри бўлса, A – фойдаланувчига r_B, n_B, n_A, A – ни умумий калит билан шифрлаб узатади:

$$B \rightarrow A: E_{k_{AB}}(r_B, n_B, n_A, B);$$

4) Натижада ҳар бир томон умумий калитни олдиндан келишиб олинган бирор функция ёрдамида $k=f(r_A, r_B)$ қонуният билан ҳисоблаб топиши мумкин.



Қуйида эса **Шамир протоколи** деб аталувчи (калитсиз) умумий махфий маълумотдан фойдаланмаган ҳолда калитни узатиш протоколини кўриб чиқилади. Бу протокол қадамларига мувофиқ калитнинг махфийлик масаласи таъминланади.

Шундай шифрлаш ва дешифрлаш ўзгартиришлари мавжудки [2, 14] барча x – маълумотлар, k_1 ва k_2 – калитлар учун қуйидаги шарт бажарилади:

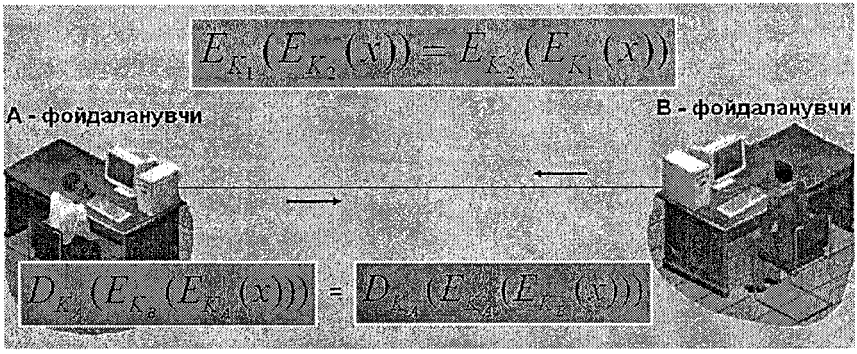
$$E_{k_1}(E_{k_2}(x)) = E_{k_2}(E_{k_1}(x)).$$

У ҳолда A ва B – фойдаланувчилар k – сеанс калитини узатувчи қуйидаги 3 – босқичли протколдан фойдаланишлари мумкин:

(1) $A \rightarrow B$: $E_{k_A}(k)$.

(2) $B \rightarrow A$: $E_{k_B}(E_{k_A}(k))$.

(3) $A \rightarrow B$: $D_{kA}(E_{kB}(E_{k_A}(k)))$.



Хусусан, Шамир протоколида модуль бўйича даражага кўтариш амалидан фойдаланиш таклиф этилган, яъни $E_{k_A}(k) = k^{k_A} \bmod p$. Шундай қилиб, бу протоколнинг криптобардошлиги дискрет логарифмлаш масаласининг мураккаблигига асосланган [2, 20]. Шамир протоколининг камчилиги шундаки, бу протоколда аутентификация масаласи ҳал этилмаган.

Нидхем-Шрёдер протоколи

Рожер Нидхем ва Михаэл Шрёдерлар томонидан яратилган бу протоколда арбитр ва симметрик криптотизимдан фойдаланилади:

1. A – фойдаланувчи ишончли томонга (W) ўзининг исмини, B – фойдаланувчининг исмини ва ўзининг тасодифий сонини узатади.

$$A \rightarrow W: A, B, R_A.$$

2. 3-ишончли томон сеанс калитни генерация қилади. Бу сеанс калитни ва A – фойдаланувчининг исмини B – фойдаланувчи билан умумий бўлган калит орқали шифрлайди. Сўнгра A – фойдаланувчи ва ўзи учун умумий бўлган калит ёрдамида A – фойдаланувчининг тасодифий сони, B – фойдаланувчининг исми, калит ва шифрматнни шифрлайди. Ниҳоят у шифрланган маълумотни A – фойдаланувчига узатади:

$$W \rightarrow B: E_A(R_A, B, k, E_B, (k, A)).$$

3. A – фойдаланувчи маълумотни дешифрлаб, k – калитни олади. У R_A ва 1-босқичда узатилган R_A ни солиштиради. Сўнгра A – фойдаланувчи ишончли томон шифрлаган маълумотни B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_B(k, A).$$

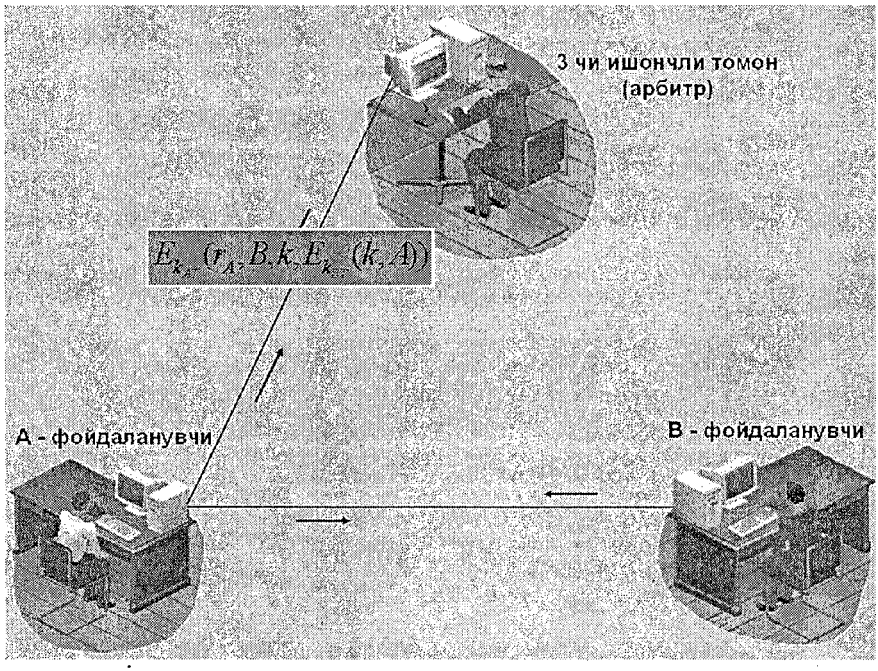
4. В – фойдаланувчи бу маълумотни дешифрлайди ва k – калитни олади. Сўнгра у тасодифий R_B – сонини генерация қилади. Бу тасодифий сонни k – калит ёрдамида шифрлайди ва A – фойдаланувчига узатади:

$$B \rightarrow A\colon E_k(R_B).$$

5. А – фойдаланувчи k – калит ёрдамида маълумотни дешифрлайди. A – фойдаланувчи тасодифий $R_B - 1$ – сонини генерация қилади. Бу сонни k – калит ёрдамида шифрлаб қайта B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_k(R_B - 1).$$

6. B – фойдаланувчи маълумотни дешифрлаб, $R_B - 1$ – сонини текширади ва ҳақиқатдан A – фойдаланувчи билан алоқа ўрнатаётганига ишонч ҳосил қилади.



Бу протоколда R_A, R_B, ва R_B-1 – сонларидан такроран фойдаланилади. Агар криптоаналитик аввал фойдаланилган k – калитни қўлга киритса, 3-босқичда A – фойдаланувчи номидан B – фойдаланувчига маълумот узатиши мумкин.

Криптотаҳлилчи ҳужуми кетма-кетлиги

1. Криптотаҳлилчи B – фойдаланувчига қуйидаги маълумотни узатади:

$$C \rightarrow B\colon E_B(k, A).$$

2. В – фойдаланувчи k – калитни олади, тасодифиий R_B – сонини генерация қилади ва A – фойдаланувчига қуйидаги маълумотни узатади:

$$B \rightarrow A\colon E_k(R_B).$$

3. Криптотаҳлилчи маълумотни қўлга киритиб, k – калит ёрдамида очади. У B – фойдаланувчига қуйидаги маълумотни узатади:

$$C \rightarrow B\colon E_k(R_B-1).$$

4. B – фойдаланувчи маълумотни дешифрлаб R_B-1 ни олиб текширади. Сўнгра A – фойдаланувчи билан алоқа ўрнатаётганига ишонч ҳосил қилади.

Криптотаҳлилчи B – фойдаланувчини шу тартибда ишонтириши мумкин.

Хулоса. Бу камчиликни бартараф этиш учун вақт белгисидан фойдаланиш мақсадга мувофиқ бўлади. Чунки 2) – босқичда ишончли томон маълумотига вақт белгиси қўшилади. Вақт белгиси тизимда аниқ ва ишончли вақтни талаб қилади.

Агар криптотаҳлилчи A – фойдаланувчининг умумий калитини қўлга киритса, k – сеанс калитга ҳам эга бўлиши ва B – фойдаланувчи билан алоқа боғлаши мумкин. Бу ҳолат A – фойдаланувчи ўзининг умумий калитини ўзгартирган тақдирда ҳам давом этиши мумкин.

Wide-Mouth Frog протоколи

Ушбу протоколни ишончли сервер учун фойдаланиладиган калитларни алмашувчи симметрик протокол дейиш мумкин. A ва B – фойдаланувчилар арбитр билан биргаликда умумий калитлардан фойдаланадилар. Wide-Mouth Frog протоколида A – фойдаланувчи B – фойдаланувчига сеанс калитни қуйидагича узатади:

1. A – фойдаланувчи вақт белгисини, B – фойдаланувчининг исмини ва сеанс калитни бирлаштириб умумий калит билан шифрлайди. Ўзининг исмини ва шифрматнни арбитр (W) га узатади:

$$A \rightarrow W\text{: } A, E_A(t, A, k).$$

2. Арбитр A – фойдаланувчининг малумотини дешифрлайди. Сўнгра янги вақт белгисини, A – фойдаланувчининг исмини ва сеанс калитни бирлаштириб ўзи ва B – фойдаланувчи учун умумий бўлган калит билан шифрлайди. Натижани B – фойдаланувчига узатади:

$$W \rightarrow B\text{: } E_B(t, A, k).$$

3. B – фойдаланувчи бу маълумотни қабул қилиб умумий калит билан дешифрлайди ва вақт белгисини олиб қабул қилган вақти билан солиштиради. Агар бу вақтлар орасидаги фарқ белгиланган интервалдан ошмаса k калитни ҳақиқий деб қабул қилади.

Yahalom протоколи

Бу пртоколга мувофиқ A ва B – фойдаланувчилар арбитр билан умумий калитдан фойдаланадилар. Протокол қадамлари кетмакетлиги қуйидагидан иборат:

1. A – фойдаланувчи ўзи исми ва тасодифий сонини бирлаштириб, B – фойдаланувчига узатади:

$$A \rightarrow B\text{: } A, R_A.$$

2. B – фойдаланувчи A – фойдаланувчининг исмини, унинг тасодифий сонини ва ўзининг тасодифий сонини бирлаштириб умумий калит билан шифрлайди. Ўзининг исмини ва натижани бирлаштириб арбитрга узатади:

$$B \rightarrow W\text{: } B, E_B(A, R_A, R_B).$$

3. Арбитр иккита маълумотни ҳосил қилади. Биринчи маълумот B – фойдаланувчининг исми, сеанс калит, A ва B – фойдаланувчиларнинг тасодифий сонларидан ташкил топган. Бу маълумотни ўзининг ва A – фойдаланувчининг умумий калити билан шифрлайди. Иккинчи маълумот A – фойдаланувчининг исми ва сеанс калитидан ташкил топган. Арбитр бу маълумотни ўзи ва B – фойдаланувчи учун умумий бўлган калит билан шифрлайди. Сўнгра бу маълумотларни A – фойдаланувчига узатади:

$$W \rightarrow A\text{: } E_A(B, k, R_A, R_B), E_B(A, k).$$

4. A – фойдаланувчи биринчи маълумотни дешифрлайди ва k – калитни олади. У R_A ни 1) – босқичда узатилган қиймати билан солиштиради ва тўғри эканлигига ишонч ҳосил қилади. Сўнгра A – фойдаланувчи B – фойдаланувчига иккита маълумот узатади, биринчи – арбитрнинг маълумоти, иккинчиси – сеанс калит билан шифрланган R_B тасодифий сон:

$$A \rightarrow B\colon E_B(A, k), E_k(R_B).$$

5. В – фойдаланувчи биринчи маълумотни дешифрлаб, k – калитни олади. Бу калит ёрдамида иккинчи маълумотни очиб, R_B нинг қиймати 2) – босқичда юборилгани билан мос келишига ишонч ҳосил қилади.

Натижада А ва В – фойдаланувчилар айнан бир-бирлари билан алоқа боғлаганларига ишонч ҳосил қиладилар.

Отвей-Риис протоколи

Бу протоколда ҳам симметрик шифрлаш алгоритмидан фойдаланилади. Протокол қадамлари кетма-кетлиги қуйидагича:

1. A – фойдаланувчитартиб рақами, ўзининг исми, B – фойдаланувчининг исми ва тасодифий R_A – сонидан ташкил топган маълумотни ҳосил қилади ва уни шифрлайди. Сўнгра у шифрматнни, тартиб рақамини, ўзининг ва B – фойдаланувчининг исмини B – фойдаланувчига узатади:

$$A \rightarrow B\colon I, A, B, E_A(R_A, I, A, B).$$

2. B – фойдаланувчи тасодифий R_B – сони, тартиб рақами, A – фойдаланувчи ва ўзининг исмидан ташкил топган маълумотни ҳосил қилади. Бу маълумот умумий калит билан шифрланади. Сўнгра B – фойдаланувчи бу маълумотни, A – фойдаланувчи юборган маълумотни, тартиб рақами, ўзи ва A – фойдаланувчининг исмини арбитрга узатади:

$$B \rightarrow W\colon I, A, B, E_A(R_A, I, A, B), E_B(R_B, I, A, B).$$

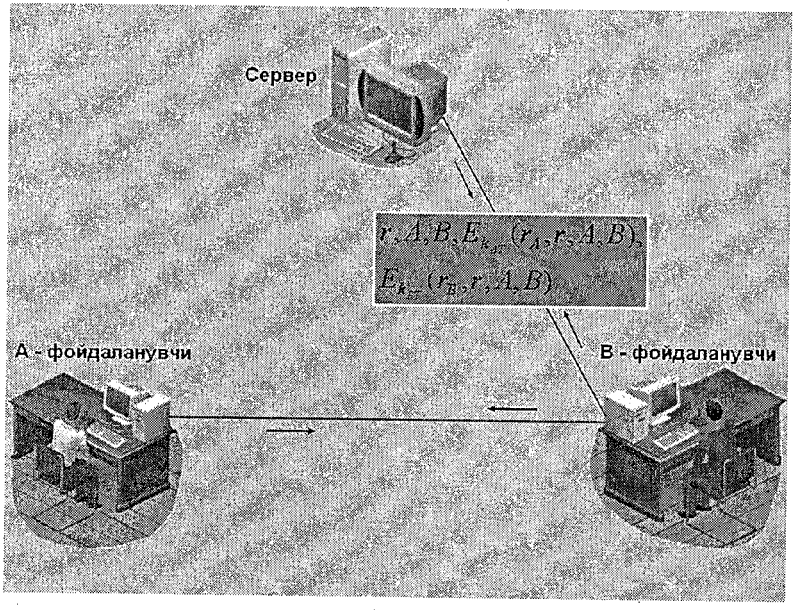
3. Арбитр тасодифий сеанс калитини хосил қилади. Сўнгра иккита маълумотни ҳосил қилади, биринчиси – A – фойдаланувчининг умумий калити билан шифрланган A – фойдаланувчининг тасодифий R_A – сони, иккинчиси – B – фойдаланувчининг умумий калити билан шифрланган A – фойдаланувчининг тасодифий R_A – сони. Арбитр тартиб рақамини ва иккала маълумотни бирлаштириб B – фойдаланувчига узатади:

$$W \rightarrow B: I, E_A(R_A, k), E_B(R_B, k).$$

4. B – фойдаланувчи A – фойдаланувчининг калити билан шифрлаган тасодифий сон ва k – сеанс калитни A – фойдаланувчига узатади:

$$B \rightarrow A: I, E_A(R_A, k).$$

5. A – фойдаланувчи маълумотни дешифрлаб ўзининг тасодифий сони ва k – сеанс калитига эга бўлади. A – фойдаланувчи протокол бажарилиши натижасида улар ўзгармасдан қолганига ишонч ҳосил қилади.



Агар протокол бажарилиши натижасида барча тасодифий сонлар тўғри ва тартиб рақами ўзгармаган бўлса, у ҳолда A ва B – фойдаланувчилар бир-бирларининг хақ эканликларига ишонч ҳосил қиладилар ва ўзаро маълумот алмашиш учун махфий калитни қабул қиладилар.

Kerberos протоколи

Kerberos протоколи **Нидхем-Шрёдер** протоколининг модификацион варианти ҳисобланади. A – фойдаланувчи B – фойдаланувчи билан маълумот алмашиши учун уларга сеанс калити қуйидагича амалга оширилади:

1. A – фойдаланувчи арбитрга ўзининг исми ва В – фойдаланувчининг исмидан ташкил топган маълумотни узатади:

$$A \rightarrow W\colon A, B.$$

2. Арбитр иккита маълумотни ҳосил қилади, биринчиси вақт белгиси, ҳаётий вақт L, тасодифий сеанс калит ва A – фойдаланувчининг исмидан ташкил топган. Арбитр бу маълумотни ўзи ва B – фойдаланувчи учун умумий бўлган калит билан шифрлайди, иккинчиси вақт белгиси, ҳаётий вақт, тасодифий сеанс калит ва B – фойдаланувчининг исмидан ташкил топган. Арбитр бу маълумотни ўзи ва A – фойдаланувчи учун умумий бўлган калит билан шифрлайди. У иккала шифрматнни A – фойдаланувчига узатади:

$$W \rightarrow A\colon E_A(t, L, k, B), E_B(t, L, k, A).$$

3. A – фойдаланувчи ўзининг калити билан биринчи шифрматнни дешифрлайди. У ўзининг исми ва вақт меткасини бирлаштириб, k – сеанс калит билан шифрлайди. Бу шифрматнни ва арбитрдан қабул қилган иккинчи шифрматнни B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_k(A, t), E_B(t, L, k, A).$$

4. B – фойдаланувчи ўзининг калити ёрдамида иккинчи шифрматнни дешифрлайди ва сеанс калитига эга бўлади. Бу сеанс калит ёрдамида биринчи шифрматнни дешифрлайди. Натижада ҳосил бўлган A – фойдаланувчининг исми ва вақт белгиси аввалгиси билан мос бўлса, B – фойдаланувчи A – фойдаланувчини идентификация қилади. Энди A – фойдаланувчи уни идентификация қилиши учун вақт белгисига 1 рақамини қўшиб сеанс калит билан шифрлайди. Ҳосил бўлган шифрматнни A – фойдаланувчига узатади:

$$B \rightarrow A\colon E_k(t+1).$$

Агар ҳар бир фойдаланувчининг соатлари арбитрнинг соати билан синхрон равишда ишласа, бу протокол яхши натижа беради.

Neuman-Stubblebine **протоколи**

Тизимдаги бирор камчилик ёки соатлар синхронизациясининг носозлиги туфайли ишлатилмоқчи бўлаётган протокол амалга ошмаслиги мумкин. Агар узатувчининг соати қабул қилувчининг соатидан илгарилаб кетса, криптоаналитик узатувчининг маълумотини қабул қилиб, маълум вақтдан сўнг бу маълумотни такроран узатиши мумкин. «Яширин такрорий узатиш» деб номланган бу ҳужум усули кўнгилсиз оқибатларга олиб келиши мумкин.

Бу протокол яширин такрорий узатиш ҳужумига бардошли ҳисобланади. У Yahalom протоколининг яхшиланган варианти бўлиб, алгоритм қадамлари кетма-кетлиги қуйидагилардан иборат:

1) A – фойдаланувчи ўзининг исмини ва тасодифий сонини В – фойдаланувчига узатади:

$$A \rightarrow B\text{: } A, R_A.$$

2) B – фойдаланувчи бу маълумотга вақт белгисини бирлаштириб, уни ўзи ва арбитр учун маълум бўлган калит билан шифрлайди. B – фойдаланувчи ўзи исми, тасодифий сони ва шифрматнни арбитрга узатади:

$$B \rightarrow W\text{: } B, R_B, E_B(A, R_A, t).$$

3) Арбитр бу маълумотни олиб шифрматнни дешифрлайди. У сеанс калитни генерация қилади. Сўнгра у иккита маълумотни ҳосил қилади, биринчиси B – фойдаланувчининг исми, A – фойдаланувчининг тасодифий сони, сеанс калит ва вақт белгисидан иборат. Арбитр бу маълумотни ўзи ва A – фойдаланувчи учун умумий бўлган калит билан шифрлайди, иккинчи маълумот A – фойдаланувчининг исми, сеанс калит ва вақт белгисидан иборат. Арбитр бу маълумотни ўзи ва B – фойдаланувчи учун умумий бўлган калит билан шифрлайди. У иккала шифрматнни ва B – фойдаланувчининг тасодифий сонини A – фойдаланувчига узатади:

$$W \rightarrow A\text{: } E_A(B, R_A, k, t), E_B(A, k, t), R_B.$$

4) A – фойдаланувчи биринчи шифрматнни дешифрлаб k – сеанс калитни олади ва R_A – тасодифий сонини 1) – босқичда юборилган қиймати билан тенг эканига ишонч ҳосил қилади. A – фойдаланувчи сеанс калит ёрдамида R_B – тасодифий сонни шифрлайди. У бу шифрматнни ва арбитрнинг иккинчи шифрматнини B – фойдаланувчига узатади:

$$A \rightarrow B\text{: } E_B(A, k), E_k(R_B).$$

5) B – фойдаланувчи ўзининг калити ёрдамида биринчи шифрматнни очади ва k – сеанс калитга эга бўлади. Сеанс калит ёрдамида иккинчи шифрматнни очади ва тасодифий сонни ҳосил қилиб, иккинчи босқичда узатилган қиймати билан тенг эканлигига ишонч ҳосил қилади.

Агар иккала тасодифий сонлар ва вақт белгилари мос бўлса, A ва B – фойдаланувчилар бир-бирларининг ҳақ эканлигига ишонч ҳосил

қиладилар ва махфий калитни қабул қиладилар. Соатларнинг синхронизацияси талаб қилинмайди, вақт белгиси фақат B – фойдаланувчининг соати ёрдамида аниқланади ва фақат B – фойдаланувчигина вақт меткасини текшира олади.

Бу протоколнинг яна бир афзаллик томони шундан иборатки, A – фойдаланувчи арбитрдан қабул қилган малумотдан кейинчалик, B – фойдаланувчининг ҳақлигини текшириш учун бирор вақт оралиғида такроран фойдаланиши мумкин. Фараз қилайлик, A ва B – фойдаланувчилар юқорида келтирилган протоколни бажаришди ва сеанс алоқаси якунланди. Энди A ва B – фойдаланувчилар арбитрга мурожаат қилмасдан туриб бир-бирларининг ҳақ эканликларини текширишлари мумкин. Яъни:

1) A – фойдаланувчи учинчи босқичда арбитр томонидан юборилган шифрматнни ва янги тасодифий сонни B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_B(A, k, t), R'_A.$$

2) B – фойдаланувчи A – фойдаланувчининг тасодифий сонини сеанс калит билан шифрлайди. Бундан ташқари ўзи ҳам бошқа тасодифий сон ҳосил қилиб, шифрматн билан бирга A – фойдаланувчига узатади:

$$B \rightarrow A\colon R'_B, E_k(R'_A).$$

3) A – фойдаланувчи B – фойдаланувчининг тасодифий сонини сеанс калит билан шифрлаб B – фойдаланувчига узатади.

$A \rightarrow B\colon E_k(R'_B)$. Янги тасодифий сонларни қайта узатиш натижасида пртоколнинг очилишини химоя қилиш мумкин экан.

Маълумот алмашувчи томонлар аутентификациясини симметрик калитли алгоритмлар ёрдамида ташкил қилиш протоколлари ва уларнинг криптоҳужумга бардошлилиги

Криптографияда идентификация протоколлари «савол-жавоб» шаклидаги ғояга асосланган бўлиб, унга мувофиқ узатувчи ўз маълумотини қабул қилувчига ҳақиқатдан ҳам ўзи узатганини ва унинг бирор қиймати билан исботлашга қаратилган бўлади.

Сўров – одатда ҳар бир протоколда текширувчи томонидан ишлаб чиқилади. Агар алоқа канали криптоаналитик томонидан назо-

рат остига олинган бўлса, идентификация протоколидаги ихтиёрий қиймати криптоаналитик учун кейинги босқичда ёлғон маълумот узатиш имконини бермаслиги лозим. Бундай вазиятларнинг олдини олиш мақсадида, криптографик протоколларда одатда тасодифий сонлар ёки вақт белгиларидан фойдаланилади.

Ҳар бир тасдиқловчи ва текширувчи томонлар учун алоҳида такрорланмайдиган кетма-кетликдаги сонлар ишлатилади. Бундан ташқари A дан B га ва B дан A га маълумот узатиш учун турли-хил кетма-кетликдаги сонлардан ҳам фойдаланиш мумкин.

Томонлар бундай кетма-кетликларни яратишнинг аввалдан бошлаб қўйилган қоидасига қаътий риоя қиладилар. Кетма-кетлик олдин фойдаланилмаган ва маълум қоидани қаноатлантирган тақдирдагина қабул қилинади. Бу усулдан фойдаланишнинг камчилиги шундаки, ҳар бир тасдиқловчи ва текширувчига тегишли маълумотлар маълум жойда сақланиши лозим.

Вақт белгисидан фойдаланиш узатилган маълумотнинг ягоналигини ва ўз вақтида етиб боришини кафолатлайди. Вақт белгисидан фойдаланувчи протоколлар қуйидагича амалга оширилади. Маълумотни узатувчи томон ўзининг тизим вақтини маълумот давомига қўшади. Иккинчи томон бундай маълумотни қабул қилиб, ўзининг тизим вақти билан солиштиради. Агар бу иккала томон тизим вақтларининг фарқи қайд қилинган интервал оралиғида бўлса, маълумот қабул қилинади. Вақт белгисига асосланган усулнинг ишончлилиги томонлар тизим вақтлари синхронизациясининг аниқлигига асосланган [2]

«Савол-жавоб» алгоритмидан фойдаланувчи идентификация протоколи тасдиқловчи текширувчи томонлар учун умумий бўлган махфий калитни талаб қилади. Алгоритмни тасвирлаш учун қуйидаги белгилашлар киритирилади.

z_A – A – фойдаланувчининг тасодифий сони;

t_A – А – фойдаланувчининг вақт белгиси;

E_A – k – калит ёрдамида шифрлаш алгоритми;

$id(B)$ – B – фойдаланувчининг идентификация рақами.

1. Вақт белгисидан фойдаланиб бир томонлама идентификацияни таъминлаш протоколи

A – фойдаланувчи ўзининг вақт белгиси ва B – фойдаланувчининг идентификация рақамини бирлаштириб, шифрлайди. Ҳосил бўлган маълумотни B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_k(t_A,\ id(B)).$$

Текширувчи B – фойдаланувчи бу маълумотни дешифрлаб, вақт меткасини ва идентификация рақамини узатишнинг сабаби шундаки, криптоаналитик бу маълумотни дарҳол A – фойдаланувчининг ўзига қайта узатиши мумкин.

2. Тасодифий сондан фойдаланиб бир томонлама идентификацияни таъминлаш протоколи

Вақт белги тасодифий сон билан алмаштирилиши мумкин.

У ҳолда протокол қуйидаги кўринишга келади. B – фойдаланувчи ўзининг тасодифий сонини ҳосил қилиб, A – фойдаланувчига очиқ ҳолда узатади:

$$B \rightarrow A\colon z_B.$$

A – фойдаланувчи бу тасодифий сонни ҳамда B – фойдаланувчининг идентификация рақамини бирлаштириб, аввалдан маълум бўлган умумий калит билан шифрлайди ва B фойдаланувчига узатади:

$$A \rightarrow B\colon E_k(z_B,\ id(B)).$$

Бу протоколда текширувчи B – томон, маълумотни дешифрлаб, қабул қилинган тасодифий сон билан мос келишини текширади. Шундан сўнг қабул қилинган ва ўзининг идентификация рақамларини солиштиради.

3. Тасодифий сондан фойдаланиб биргаликда идентификацияни таъминлаш протоколи

Бу протокол қуйидагича амалга оширилади:

$$B \rightarrow A\colon z_B.$$
$$A \rightarrow B\colon E_k(z_A,\ z_B,\ id(B)).$$
$$B \rightarrow A\colon E_k(z_A,\ z_B).$$

B – фойдаланувчи иккинчи босқичда маълумотни дешифрлаб юқорида айтилган текширишларни амалга оширади. Шундан сўнг, 3) – босқичда z_A дан фойдаланади. Тасдиқловчи A – томон 3) – босқичда B – фойдаланувчидан олган маълумотни дешифрлаб 1) ва 2) – босқичда узтилган тасодифий сонларнинг тўғри эканлигини текширади [2].

Ньюман-Стабблайн протоколи

1) A – фойдаланувчи ўз исми ва тасодифий сонини бирлаштириб, натижани B – фойдаланувчига узатади:

$$A \rightarrow B\colon A,\ R_B.$$

2) B – фойдаланувчи A – фойдаланувчининг исми, тасодифий сони ва вақт белгисини бирлаштириб умумий калит билан шифрлайди. Ўзининг исми ва тасодифий сонини натижага қўшиб, ишончли томонга узатади:

$$B \rightarrow W\text{: } B,\ R_B,\ E_B(A,\ R_A,\ T_W).$$

3) Ишончли томон тасодифий сеанс калитни генерация қилади ва иккита маълумотни ҳосил қилади, биринчиси – B – фойдаланувчининг исми, A – фойдаланувчининг тасодифий сони, сеанс калит ва вақт белгисидан иборат. Бу маълумотни умумий калит билан шифрлайди. Иккинчиси – A – фойдаланувчининг исми, сеанс калит ва вақт белгисидан ташкил топади. Бу маълумотни B – фойдаланувчининг умумий калити билан шифрлайди. Иккала шифрматнни ва B – фойдаланувчининг тасодифий сонини A – фойдаланувчига узатади:

$$W \rightarrow A\text{: } E_A(B,\ R_A,\ k,\ T_B),\ E_B(A,\ k,\ T_B),\ R_B.$$

4) A – фойдаланувчи ишончли томоннинг умумий калити билан шифрланган маълумотни очиб k – калитни олади ва R_A нинг ҳақиқий эканига ишонч ҳосил қилади. A – фойдаланувчи B – фойдаланувчига иккита маълумот узатади. Биринчиси – B – фойдаланувчининг калити билан шифрланган арбитр маълумоти. Иккинчиси эса сеанс калит ёрдамида шифрланган R_B маълумот:

$$A \rightarrow B\text{: } E_A(A,\ k),\ E_k(R_B).$$

5) B – фойдаланувчи бу маълумотни дешифрлайди ва k ни олади. T_B ва R_B ни 2 – босқичда юборилгани билан солиштиради ва маълумотнинг ҳақиқий эканига ишонч ҳосил қилади.

Агар иккита тасодифий сонлар ва вақт белгиси мос келса, A ва B – фойдаланувчилар сеанс калитни ўзаро ишлатаётганларига ишонч ҳосил қиладилар.

Шундан сўнг A ва B – фойдаланувчилар бир-бирларининг ҳақ эканликларини ишончли томоннинг иштирокисиз текширишлари мумкин.

1) A – фойдаланувчи 3 – босқичда ишончли томон унга юборган маълумотни ва янги тасодифий сонни B – фойдаланувчига узатади:

$$A \rightarrow B\text{: } E_B(A,\ k,\ T_B),\ R_A.$$

2) B – фойдаланувчи A – фойдаланувчи юборган янги тасодифий сонни сеанс калит билан шифрлайди. У A – фойдаланувчига янги тасодифий сонни ва шифрматнни узатади:

$$B \rightarrow A: R_B, E_k(R_A).$$

3) A – фойдаланувчи B – фойдаланувчининг янги тасодифий сонини сеанс калит билан шифрлаб унга узатади:

$$A \rightarrow B: E_k(R_B).$$

Ушбу ҳолат эса янги тасодифий сонлар такроран узатилиши мумкин бўлган ҳужумни бартараф этади.

SKID протоколи

SKID протоколи симметрик криптография асосида қурилган идентификация масаласини таъминлашга қаратилган бўлиб, унинг SKID2 ва SKID3 турлари мавжуд. Қуйида B – фойдаланувчига ўзининг ҳақлигини кўрсатиш имконини берувчи SKID2 протоколини кўриб чиқилади.

1) A – фойдаланувчи тасодифий R_A – сонини В – фойдаланувчига узатади.

2) B – фойдаланувчи тасодифий R_B – сонини олади. У R_B, $H_k(R_A, R_B, B)$ ни A – фойдаланувчига узатади. Бу ерда H_k – MAC код, B эса B – фойдаланувчининг исми.

3) A – фойдаланувчи $H_k(R_A, R_B, B)$ ни хисоблайди ва қабул қилгани билан солиштиради. Агар натижалар тенг бўлса, у ҳолда A – фойдаланувчи B – фойдаланувчи билан тўғри боғланганига ишонч ҳосил қилади.

SKID3 протоколи эса A ва B – фойдаланувчиларнинг тўлиқ аутентификациясини таъминлайди. 1, 3 – босқичлар SKID2 протоколи каби бажарилади:

4) A – фойдаланувчи $H_K(R_B, A)$ ни B – фойдаланувчига узатади. Бу ерда A, A – фойдаланувчининг исми.

5) B – фойдаланувчи $H_K(R_B, A)$ ни хисоблайди ва қабул қилгани билан солиштиради. Агар натижалар тенг бўлса, B – фойдаланувчи айнан A – фойдаланувчи билан боғланганига ишонч ҳосил қилади.

Ушбу протокол «Ўртадаги киши» ҳужумига бардошли эмас.

Хулоса. Қуйидаги **10.1** ва **10.2-жадвал**ларда симметрик шифрлаш алгоритмидан фойдаланиб калит узатиш протоколларининг таҳлили (арбитр иштирок этмаган ва арбитр иштирок этган ҳол учун мос равишда) келтирилган.

10.1-жадвал. Симметрик шифрлаш алгоритмидан фойдаланиб калит узатиш протоколлари таҳлили (арбитр иштирок этмаган ҳол учун)

№	Протокол номи	A	B	rA	rB	k	t	Бўлиши мумкин бўлган ҳужум
1.	Нидхем-Шредер протоколи	–	–	+				Арбитр маълумотни кимдан келгани ва кимга юбориш кераклиги ҳақида ҳеч нарса билмайди.
		+	+	–		+		А фойдаланувчи маълумотни арбитрдан келганига тўла ишонч ҳосил қилолмайди
		–				+		В фойдаланувчи калитни ким томонидан юборилганини билмайди
		+				+		Криптоаналитик бу калитни маълум вақтдан сўнг қайта узатиши мумкин
2.	Wide-Mouth Frog протоколи	+	+				–	Криптоаналитик бу маълумотни *B* фойдаланувчига такроран узатиши мумкин
		–	–				+	Фойдаланувчилар маълумотнинг кимдан келганини билмайдилар
3.	Yahalom протоколи	+	+	–	–	+		Фойдаланувчилар ўртасида ўзаро идентификация таъминланмайди
4.	Kerberos протоколи	Тизим вақтининг синхрон равишда ишлаши талаб этилади						

10.2-жадвал. Симметрик шифрлаш алгоритмидан фойдаланиб калит узатиш протоколлари таҳлили (арбитр иштирок этган ҳол учун)

№	Протокол номи	r_A	r_B	k_A	B	t_A	Бўлиши мумкин бўлган ҳужум
1.	Вақт меткасидан фойдаланиб калитни узатиш протоколи	+			+	–	Криптоаналитик бу маълумотни қайта узатиши мумкин
		+			–		Криптоаналитик бу маълумотни *A* фойдаланувчига қайта узатиши мумкин

№	Протокол номи	r_A	r_B	k_A	B	t_A	Бўлиши мумкин бўлган хужум
2.	Вакт меткаси ва хешфункциядан фойдаланиб калитни узатиш протоколи				+	–	Криптоаналитик бу маълумотни B фойдаланувчига қайта узатиши мумкин
					–	+	Криптоаналитик бу маълумотни А фойдаланувчига қайта узатиши мумкин
3.	Тасодифий сондан фойдаланиб калитни узатиш протоколи	+			–		Криптоаналитик бу маълумотни А фойдаланувчига қайта узатиши мумкин
		–			+		В фойдаланувчи калитнинг янги эканига ишонч ҳосил қилмайди
4.	Тасодифий сон ва хеш-функциядан фойдаланиб		–		+		В фойдаланувчи умуман бошка калитни ҳосил қилади
			+		–		В фойдаланувчи умуман бошка калитни ҳосил қилади
5.	Шамир протоколи						Криптоаналитик $E_{k_4}(k)$ ни ва $E_{k_4}(E_{k_4}(k))$ ни билган ҳолда криптоанализ усули билан k_B ни топиши мумкин
6.	Тасодифий сонлардан фойдаланиб калитни биргаликда ҳосил қилиш протоколи	+	–	+	+		Фойдаланувчилар бир-бирларини идентификация қилмайдилар
		–	+	+	+		Фойдаланувчилар бир-бирларини идентификация қилмайдилар
		+	+	+	–		Фойдаланувчилар сеанс калитни генерация қила олмайдилар
		+	+	–	+		Фойдаланувчилар сеанс калитни генерация қила олмайдилар

§10.5. Асимметрик калитли алгоритмлар ёрдамида махфий алоқани ташкил қилиш протоколлари

Асимметрик калитли алгоритмлардан фойдаланиб ҳужжатли маълумотларнинг махфийлигини таъминловчи протоколлар

Очиқ калитли шифртизимларга асосланган идентификация протоколларида узатувчи томонларнинг махфий калитга эга эканлигини қуйидаги икки усул ёрдамида кўриш мумкин:

1) Идентификация қилиниши керак бўлган томоннинг очиқ калити билан шифрланган сўровни дешифрлаш орқали.

2) Сўровнома давомига ўзининг электрон рақамли имзосини бирлаштириш орқали.

Бу икки усулни мукаммал кўриб чиқамиз.

Қуйдаги белгилашларни киритамиз:

h – бирор қайтмас функция ;

E_A – A – фойдаланувчининг шифрлаш алгоритми;

D_A – эса А – фойдаланувчининг дешифрлаш алгоритмлари; бўлсин.

Биринчи усул қуйидаги протоколга асосланган:

$$B \rightarrow A\text{: } h(z),\ id(B),\ E_A(z,\ id(B)),$$
$$A \rightarrow AB\text{: } z.$$

текширувчи B тасодифий сон z ни ҳосил қилади, $h(z)$ ни ва $c=E_A(z, id(B))$ – сўровномани ҳисоблайди. Ўзини ҳақиқатдан ҳам А – фойдаланувчи эканини исбот қилувчи A – томон сўровномани дешифрлайди ва хэш-функция қиймати ҳамда идентификация параметрларининг тўғрилигини текширади. Агар тафовутни аниқласа, у ҳолда A – фойдаланувчи дарҳол протоколни тўхтатади. Акс ҳолда A – фойдаланувчи z ни текширувчи B га узатади. Агар A – фойдаланувчидан қабул қилинган z – сони B – фойдаланувчига маълум сонга тенг бўлса, B – фойдаланувчи A ни идентификациялайди.

Қуйида электрон рақамли имзодан фойдаланувчи протоколни кўриб чиқилади.

Белгилашлар киритилади:

z_A – A – фойдаланувчининг мос ҳолда тасодифий сони ;

t_A – A – фойдаланувчининг мос ҳолда тасодифий вақт белгиси;

S_A – A – фойдаланувчининг электрон рақамли имзосини англатади.

Электрон рақамли имзони текшириш алгоритми B – фойдаланувчига маълум деб ҳисобланади.

Асимметрик шифрлаш алгоритми ёрдамида идентификацияни таъминлаш учун қуйидаги 3 та проколдан фойдаланиш мумкин:

1. Вақт белгисидан фойдаланиб бир томонлама идентификацияни таъминлаш

Бу протоколда А – фойдаланувчи B – фойдаланувчига вақт белгиси, В нинг идентификация рақами ва уларга қўйилган электрон рақамли имзони узатади:

$$A \rightarrow B\text{: } t_A,\ id(B),\ E_A(t_A,\ id(B)).$$

Протоколга мувофиқ, B – фойдаланувчи маълумотни қабул қилиб, вақт белгисини белгиланган ораликда эканини, $id(B)$ ни ўзининг идентификация рақами билан тенг эканлигини ва иккита параметрга қўйилган электрон рақамли имзонинг тўғрилигини ҳам текширади.

Бу протоколни тасодифий сондан фойдаланиб ҳам амалга ошириш мумкин.

2. Тасодифий сондан фойдаланиб бир томонлама идентификацияни таъминлаш

B – фойдаланувчи ўзининг тасодифий сонини *A* – фойдаланувчига очиқ ҳолда узатади:

$$B \rightarrow A : z_B,$$

A – фойдаланувчи бу тасодифий сонни қабул қилади ва *B* – фойдаланувчига қуйидаги маълумотни узатади:

$$A \rightarrow B: z_A, id(B), S_A(z_A, z_B, id(B)).$$

B – фойдаланувчи бу маълумотни қабул қилгач $id(B)$ ни ўзининг идентификация рақами билан тенг эканини текширади. Бундан ташқари $(z_A, z_B, id(B))$ – маълумотга қўйилган рақамли имзонинг тўғри эканини текширади.

Тасодифий сондан фойдаланиб ўзаро икки томонлама идентификацияни ҳам таъминлаш мумкин.

3. Тасодифий сондан фойдаланиб ўзаро идентификацияни таъминлаш

Бу протколда ҳам юқоридаги каби *B* – фойдаланувчи ўзининг тасодифий сони z_B ни *A* – фойдаланувчига узатади:

$$B \rightarrow A : z_B$$

A – фойдаланувчи ўзининг тасодифий сони z_A ни, z_B ва $id(B)$ ларни бирлаштириб рақамли имзо қўяди. *A* – фойдаланувчи ўзининг тасодифий сони, *B* нинг идентификация рақами ва рақамли имзо қўйилган маълумотни бирлаштириб, *B* – фойдаланувчига узатади:

$$A \rightarrow B : z_A, id(B), S_A(z_A, z_B, id(B))$$

B – фойдаланувчи имзонинг ҳақиқийлигини текширади. Энди у ўзи тасодифий сони z_B ни, *A* – фойдаланувчининг тасодифий сони ва идентификация рақамини бирлаштириб рақамли имзо қўяди. *B* – фойдаланувчи *A* нинг идентификация рақамини ва рақамли имзо қўйилган маълумотни *A* – фойдаланувчига узатади:

$$B \rightarrow A : id(A), SB(z_B, z_A, id(A)).$$

A – фойдаланувчи бу маълумотни қабул қилиб, $id(A)$ ни ўқийди ва маълумот унга узатилганлигини аниқлайди. Сўнгра рақамли имзонинг ҳақиқий эканини текширади. Агар электрон рақамли имзо

ҳақиқий бўлса, A–фойдаланувчи B–фойдаланувчи билан алоқа боғлаганига ишонч ҳосил қилади. Шу тарзда абонентлар ўртасида идентификация таъминланади.

§ 10.6. Асимметрик шифрлаш алгоритмидан фойдаланиб калитларни алмашиш протоколлари

Асимметрик шифрлаш алгоритмлари симметрик шифрлаш алгоритмларига нисбатан секин ишлайди. Бу тафовут катта хажмдаги матнларни шифрлашда яққол намоён бўлади. Шу сабабли матнлар асосан симметрик шифрлаш алгоритми ёрдамида шифрланади. Симметрик шифрлаш алгоритмининг махфий калитини алмашишда асиммметрик шифрлаш алгоритмидан фойдаланилади. Бошқача қилиб айтганда калитнинг хажми кичик бўлгани учун у асимметрик шифрлаш алгоритмидан фойдаланиб шифрланади. Матнлар эса катта хажмга эга бўлгани учун симметрик шифрлаш алгоритми ёрдамида шифрланади.

Электрон рақамли имзодан фойдаланмаган ҳолда калитни узатиш протоколлари:

Сеанснинг махфий бўлган k – калитини узатиш учун куйидаги қадамдан иборат бўлган протоколни кўриб ўтамиз. Ушбу протокол бизга бир томонлама идентификацияни таъминлаш учун хизмат қилади:

$$A \rightarrow B\colon E_{k_B}(k, t, A).$$

Бу ерда :

E – асимметрик шифрлаш алгоритми;

t – вақт белгиси.

A – фойдаланувчи симметрик шифрлаш алгоритмининг калитини, вақт меткасини ва ўзининг идентификацион рақамини бирлаштириб, B – фойдаланувчининг очиқ калити ёрдамида шифрлайди ва унга узатади. B – фойдаланувчи ўзининг ёпиқ калити ёрдамида бу маълумотни дешифрлайди. Натижада k – калит, t – вақт меткаси ва A – фойдаланувчининг идентификация рақамига эга бўлади. B – фойдаланувчи t – вақт меткасини текширади, агар тўғри бўлса k – калитни ҳақиқий деб қабул қилади.

Томонлар ўртасида ўзаро **идентификация**ни таъминлаш учун эса қуйидаги протоколдан фойдаланиш мумкин. Бу протокол қуйидагича амалга оширилади:

A – фойдаланувчи k_1 – калитни ва ўзининг идентификация рақамини бирлаштириб, B – фойдаланувчининг очиқ калити билан шифрлайди ва B га узатади

$$A \rightarrow B: E_B(k1, A).$$

B – фойдаланувчи бу маълумотни дешифрлайди, натижада k_1 ва A га эга бўлади. У k_1 – калитни олиб унга k_2 – калитни бирлаштириб, A – фойдаланувчининг очиқ калити билан шифрлайди ва A га узатади

$$B \rightarrow A: E_A(k_1, k_2).$$

A – фойдаланувчи бу маълумотни ўзининг ёпиқ калити билан дешифрлаб, k_1 ва k_2 ларга эга бўлади. k_1 калитни текшириб, B – фойдаланувчини идентификациялайди. Энди B – фойдаланувчи уни идентификациялаши учун k_2 – калитни B – фойдаланувчининг очиқ калити ёрдамида шифрлаб, B – фойдаланувчига узатади:

$$B \rightarrow A: E_A(k_1, k_2).$$

B – фойдаланувчи бу маълумотни ўзи ёпиқ калити билан дешифрлаб эса k_2 – калитни олади. Агар бу калит 2 – босқичда юборилган k_2 – калитга тенг бўлса, B – фойдаланувчи А ни идентификациялайди. Натижавий k – калит бирор

$$k=f(k_1, k_2)$$

функция ёрдамида ҳисоблаб топилади.

Электрон рақамли имзодан фойдаланиб калитни алмашиш протоколлари

Электрон рақамли имзодан фойдаланиб генерация қилинган калитни асимметрик шифрлаш алгоритмидан фойдаланиб алмашиш протоколлари уч турга бўлинади. Бу протоколларда электрон рақамли имзони текшириш алгоритми иккала томонга ҳам маълум деб ҳисобланади.

1) Рақамли имзо қўйилган калитни шифрлаш:

$$A \rightarrow B\ E_B(k, t, S_A(B, k, t).$$

A – фойдаланувчи B нинг идентификация рақамини, генерация қилинган k – калитни ва t – вақт белгисини бирлаштириб, бу маълумотга ўзининг ёпиқ калити билан электрон рақамли имзо қўяди. Бундан сўнг генерация қилинган k – калитни, t – вақт белгисини ва рақамли имзо қўйилган маълумотини бирлаштириб, B – фойдаланувчининг очиқ калити билан шифрлайди. Ҳосил бўлган

шифрматнни B га узатади. B – фойдаланувчи ўзининг ёпиқ калити билан бу маълумотни дешифрлаб, генерация қилинган k – калитга, t – вақт белгисига ва рақамли имзо қўйилган маълумотга эга бўлади. У A – фойдаланувчининг очиқ калити ёрдамида электрон рақамли имзони текширади. Агар рақамли имзо тўғри бўлса, бу маълумотни A – фойдаланувчи узатганига ишонч ҳосил қилади ва k – калитни ҳақиқий деб тан олади.

2) Калитни шифрлаш ва калитга рақамли имзо қўйиш:

$$A \rightarrow B\ E_B(k, t)\ S_A(B, k, t).$$

A – фойдаланувчи генерация қилинган k – калитни ва t – вақт белгисини бирлаштириб, бу маълумотни B – фойдаланувчининг очиқ калити билан шифрлайди. Шундан сўнг у, B – фойдаланувчининг идентификация рақами, генерация қилинган k – калит ва t – вақт белгисини бирлаштириб, бу маълумотга ўзининг очиқ калити ёрдамида электрон рақамли имзо қўяди. Кейин эса шифрматнни ва имзоланган маълумотни бирлаштириб B – фойдаланувчига узатади. B – фойдаланувчи бу маълумотни қабул қилади ва маълумотларни очишга киришади. Аввало у ўзининг ёпиқ калитидан фойдаланиб шифрматнни дешифрлайди ва генерация қилинган k – калит ва t – вақт меткасига эга бўлади. Сўнгра A – фойдаланувчининг очиқ калити ёрдамида электрон рақамли имзони текширади. Агар электрон рақамли имзо тўғри бўлса, бу маълумотни ҳақиқатдан ҳам A – фойдаланувчи узатганига ишонч ҳосил қилади ва k – калитни ҳақиқий деб қабул қилади.

3) Шифрланган калитга электрон рақамли имзо қўйиш:

$$A \rightarrow B\colon t, E_B(A, k), S_A(B, t, E_B(A, k))$$

A – фойдаланувчи ўзининг идентификация рақамини ва генерация қилинган k – калитни бирлаштириб, B – фойдаланувчининг очиқ калити билан шифрлайди. Шундан сўнг у, B – фойдаланувчининг идентификация рақамини, t – вақт белгисини ва шифрматнни бирлаштириб, бу маълумотга электрон рақамли имзо қўяди. Сўнгра t – вақт белгиси, шифрматн ва рақамли имзо қўйилган маълумотни бирлаштириб B – фойдаланувчига узатади. B – фойдаланувчи бу малумотни олади ва уни очишга киришади. У шифрматнни ўзининг ёпиқ калити билан очади, натижада A – фойдаланувчининг идентификация рақами ва генерация қилинган k –калитга эга бўлади. Сўнгра у рақамли имзо қўйилган маълумотни A – фойдаланувчининг очиқ калити ёрдамида текширади. Агар рақамли имзо тўғри бўлса, генерация қилинган k – калитни ҳақиқий деб қабул қилади.

§ 10.7. Ўзаро маълумот алмашувчи субъектлар аутентификациясида асимметрик калитли алгоритмлар ёрдамида махфий алоқани ташкил қилиш протоколлари

SKEY дастури

Маълумотнинг хавфсизлигини таъминлаш учун SKEY (маълумотнинг ҳақиқийлигини текширувчи) дастуридан фойдаланиш мумкин. Бу дастур қуйидагича амалга оширилади.

A – фойдаланувчи аутентификация масаласини ҳал қилиш учун тасодифий R сонини киритади.

Компьютер $f(R), f(f(R)), f(f(f(R)))$, ... қийматларини ҳисоблайди. Бу қийматларни мос холда $x_1, x_2, x_3 \ldots x_{100}$ деб белгилаймиз. А фойдаланувчи бу рўйхатни қоғозга ёзиб олади ва беркитади. Бундан ташқари, компьютер x_{101} қийматни шифрланмаган холда сақлайди.

A – фойдаланувчи тизимга биринчи марта кириши учун ўз исмини ва x_{101} қийматини киритади. Компьютер $f(x_{100})$ нинг қийматини ҳисоблайди ва x_{101} билан солиштиради. Агар қийматлар тенг бўлса, хақиқатдан ҳам A – фойдаланувчи эканлигини тасдиқлайди. Сўнгра компютер маълумотлар базасидаги x_{101} қийматни x_{100} билан алмаштириб қўяди. A – фойдаланувчи эса x_{100} нинг қийматини ўз рўйхатидан ўчиради.

Кейинчалик A – фойдаланувчи ҳар сафар тизимга киришида охирги ўчирилмаган сонни киритади, масалан i. Компютер $f(x_i)$ қийматни ҳисоблайди ва маълумотлар базасида сақланаётган x_{i+1} сон билан солиштиради. SKEY дастурида ҳар бир сон бир марта иштирок этади. Бундай холатда эса криптоаналитик ҳеч қандай фойдали маълумотга эга бўла олмайди.

MTI протоколи

MTI протоколининг номи унинг муаллифлари ҳисобланган *Т. Мацумото И. Такашима ва Х. Имаи*лар шарафига қўйилган. Бу протокол ҳам Диффи-Хеллман протоколига ўхшаш бўлиб, унинг криптобардошлилиги чекли майдонда дискрет логарифмлашга асосланган [14, 20]. Бироқ ундан фарқли томони шундаки, MTI протоколида криптобардошлилигини ошириш мақсадида қўшимча a ва b ўзгарувчилардан фойдаланилади. Ушбу протоколнинг амаллар кетмакетлиги қуйидагича бажарилади. Энг аввало A ва B – фойдаланувчилар катта туб сон p ва унинг примитив илдизи a нинг қиймати ҳақида келишиб оладилар.

A – фойдаланувчи ўз махфий калити a, $1 \le a \le p-2$ ни генерация қилади ва бу калит ёрдамида

$$z_A = \alpha^a \bmod p$$

ифодани хисоблайди. A – фойдаланувчи ҳосил бўлган қийматни B – фойдаланувчига узатади:

$$A \rightarrow B: z_A = \alpha^a \bmod p,$$

B – фойдаланувчи бу маълумотни қабул қилади. У ўзининг ёпиқ калити b, $1 \le b \le p-2$ ни генерация қилади. Бу ёпиқ калит ёрдамида

$$z_B = \alpha^b \bmod p$$

ифодани ҳисоблайди ва натижани A – фойдаланувчига узатади:

$$B \rightarrow A: z_B = \alpha^b \bmod p.$$

A – фойдаланувчи z_B ни қабул қилади. A ва B – фойдаланувчилар умумий махфий калитни генерация қилиш учун мос ҳолда ўзларининг x, $1 \le x \le p-2$ ва y, $1 \le y \le p-2$ тасодифий сонларини генерация қилишлари зарур. A – фойдаланувчи ўзининг тасодифий x – сонини генерация қилиб,

$$\alpha^x \bmod p$$

ифодани ҳисоблайди ва уни B – фойдаланувчига узатади:

$$A \rightarrow B: \alpha^x \bmod p$$

B – фойдаланувчи бу маълумотни қабул қилади. У ўзининг тасодифий y – сонини генерация қилиб, $\alpha^y \bmod p$ ифодани ҳисоблайди. Ҳосил бўлган натижани A – фойдаланувчига узатади. Шу вақтдан бошлаб, B – фойдаланувчи α^x ва z_A – маълумотларга эга. Энди у ўзининг тасодифий сони ва ёпиқ калитидан фойдаланиб қуйидаги ифодани ҳисоблайди:

$$k = (\alpha^x)^b \cdot z_A^y,$$
$$B \rightarrow A: \alpha^y \bmod p.$$

A – фойдаланувчи бу маълумотни қабул қилади. Энди А – фойдаланувчи α^y ва z_B – маълумотларга эга. У ўзининг тасодифий сони ва ёпиқ калитидан фойдаланиб ушбу ифодани ҳисоблайди: $k = (\alpha^y)^a \cdot z_B^x$,

Натижавий калитнинг умумий кўриниши эса қуйидагича:

$$k = (\alpha^y)^a \cdot z_B^x = (\alpha^x)^b \cdot z_A^y = \alpha^{xb+ya} \bmod p.$$

MTI протоколи шу тартибда амалга оширилади. Унда криптоаналитикнинг ихтиёрий алмаштириши томонлардаги калитнинг қиймати турлича бўлишига олиб келади. Бу эса узатилаётган маълумотни ўқиш имкониятини бутунлай йўқотади.

Қуйида MTI протоколи учун ҳам мисол келтирилади.

$p=9531$
$\alpha=1647$
$A: a=126$
$A: Z_a=\alpha^a \bmod p=1647^{126}=\bmod 9531=3375$
$A\rightarrow B$: $Z_a=3375$.
$B: b=98$
$B: Z_B=\alpha^b \bmod p=1647^{98} \bmod 9531=8775$
$B\rightarrow A$: $Z_B=8775$.
$A: x=8643$
$A: X=\alpha^x \bmod p=1647^{98} \bmod 9531=972$
$A\rightarrow B$: $X=972$.
$B: k_1=(\alpha^x)^b Z_A^y \bmod p=X^b Z_A^y \bmod p=972^{98}\cdot 3375^{6983} \bmod 9531=3564$
$B: y=6983$
$B: Y=\alpha^y \bmod p=1647^{6983} \bmod 9531=4131$
$B\rightarrow A$: $Y=4131$
$B: k_2=(\alpha^y)^a Z_B^x \bmod p=X^b Z_B^x \bmod p=4131^{126}\cdot 8775^{972} \bmod 9531=3564$
жавоб: $k_1=k_2=k=3564$.

Арбитр иштирокида асимметрик шифрлаш алгоритми ёрдамида калитларни алмашиш

Dass протоколи

Dass протоколи очиқ калитли шифрлаш алгоритми ва симметрик шифрлаш алгоритмидан фойдаланади. Шундай қилиб, A ва B – фойдаланувчилар ўзларининг ёпиқ калитларига эгалар. Арбитр уларнинг очиқ калитларига рақамли имзо қўяди.

1) A – фойдаланувчи B – фойдаланувчининг исмини арбитрга узатади:

$$A \rightarrow \mathrm{W}: B.$$

2) Арбитр B – фойдаланувчининг исмини ва очиқ калитини бирлаштириб ўзининг ёпиқ калити билан рақамли имзо қўяди. Ҳосил бўлган шифрматнни A – фойдаланувчига узатади:

$$W \rightarrow A: S_W(B, k_B).$$

3) A – фойдаланувчи ҳақиқатдан ҳам B – фойдаланувчининг очиқ калитини олганлигини аниқлаш учун арбитрнинг рақамли имзосини текширади. У тасодифий сеанс калит k ва тасодифий очиқ/ёпиқ жуфтлик калит k_p ни генерация қилади. A – фойдаланувчи вақт белгисини тасодифий сеанс калит ёрдамида шифрлайди. Сўнгра ҳаётий вақт, ўзининг исми ва очиқ/ёпиқ жуфтлик калитларни бирлаштириб ўзининг ёпиқ калити k_A ёрдамида рақамли имзо қўяди. Ниҳоят у k – сеанс калитни B – фойдаланувчининг очиқ калити ёрдамида шифрлайди ва унга жуфтлик k_p – калит ёрдамида рақамли имзо қўяди. A – фойдаланувчи буларнинг барчасини B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_k(t_A),\ S_{k_A}(L, A, k_p),\ S_{k_p}(E_{k_B}(k)).$$

4) B – фойдаланувчи A – фойдаланувчининг исмини арбитрга узатади:

$$B \rightarrow W\colon A.$$

5) Арбитр A – фойдаланувчининг исми ва унинг очиқ калитини бирлаштириб, ўзининг ёпиқ калити билан рақамли имзо қўяди. Натижани B – фойдаланувчига узатади:

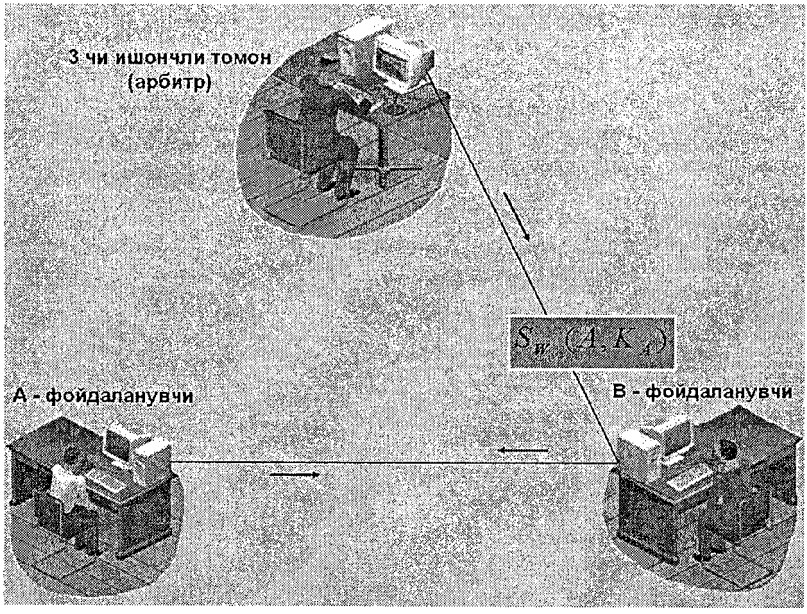
$$W \rightarrow B\colon S_W(A, k_A).$$

6) B – фойдаланувчи арбитрнинг ҳақиқатдан ҳам A – фойдаланувчининг очиқ калитини олганлигини аниқлаш учун арбитрнинг рақамли имзосини текширади. Сўнгра у, A – фойдаланувчи узатган маълумотдаги A – фойдаланувчининг рақамли имзосини текширади. Имзо ҳақиқий бўлса, очиқ/ёпиқ жуфтлик k_p – калитни ҳақиқатдан A – фойдаланувчи узатганига ишонч ҳосил қилади. Сўнгра ушбу калит ёрдамида иккинчи рақамли имзони текширади. Агар бу имзо ҳам ҳақиқий бўлса, шифрматнни ўзининг ёпиқ калити ёрдамида дешифрлайди ва натижада k – тасодифий сеанс калитга эга бўлади. Энди у сеанс калит ёрдамида биринчи шифрматнни дешифрлайди ва вақт белгисини олади. Агар бу вақт белгиси белгиланган вақт интервали оралиғида бўлса, маълумотнинг ҳақиқийлигига, қайта узатилмаган эканлигига яна бир карра ишонч ҳосил қилади.

7) Агар икки томонлама идентификация талаб қилинса, В – фойдаланувчи янги вақт белгисини сеанс калит ёрдамида шифрлаб, А – фойдаланувчига узатади:

$$B \rightarrow A\colon E_k(t_B).$$

8) A – фойдаланувчи k – сеанс калит ёрдамида t_B – вақт белгисини дешифрлайди ва маълумотнинг хозирги пайтда узатилганига ишонч хосил қилади. Ана шу ҳолда икки томонлама идентификация таъминланади.



Denning-Sacco протоколи

Ушбу протоколда ҳам асимметрик шифрлаш алгоритмидан фойдаланилади. Арбитр барча фойдаланувчиларнинг очиқ калитларини сақловчи маълумотлар базасини бошқаради.

1) A – фойдаланувчи ўзининг ва B – фойдаланувчининг исмини арбитрга узатади:

$$A \to W\colon A, B.$$

2) Арбитр B – фойдаланувчининг исмини ва очиқ калитини бирлаштириб ўзининг ёпиқ калити ёрдамида рақамли имзо қўяди. Шунингдек у A – фойдаланувчининг исмини ва очиқ калитини бирлаштириб ўзининг ёпиқ калити билан рақамли имзо қўяди. Арбитр иккала маълумотни ҳам A – фойдаланувчига узатади:

$$W \to A\colon S_W(B, k_B), S_W(A, k_A)$$

1) A – фойдаланувчи арбитрнинг очиқ калити ёрдамида рақамли имзони текширади. Агар имзо тўғри бўлса, B – фойдаланувчининг очиқ калитини ҳақиқий деб қабул қилади. У тасодифий сеанс калит ва вақт меткасини бирлаштириб, ўзининг ёпиқ калити ёрдамида рақамли имзо қўяди. Натижани B – фойдаланувчининг очиқ калити ёрдамида шифрлайди. A – фойдаланувчи бу шифрматнга арбитрдан қабул қилган иккита маълумотни бирлаштириб B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_B(S_A(k, t_A))\ S_W(A, k_A),\ S_W(B, k_B).$$

2) B – фойдаланувчи арбитрнинг очиқ калитидан фойдаланиб унинг рақамли имзосини текширади. Агар имзо тўғри бўлса, A – фойдаланувчининг очиқ калитини ҳақиқий деб қабул қилади. У ўзининг ёпиқ калити билан шифрматнни дешифрлайди. Ҳосил бўлган маълумотдан A – фойдаланувчининг рақамли имзосини текширади. Агар имзо ҳақиқий бўлса, сеанс калит сифатида k ни қабул қилади. Шунингдек, B – фойдаланувчи вақт меткасини текшириб, маълумотнинг яқин вақт ичида узатилганига ишонч ҳосил қилади.

Шу вақтдан бошлаб, А ва B – фойдаланувчилар сеанс калитга эга бўлдилар ва хавфсиз сеанс алоқасини ўрнатишлари мумкин бўлади.

Бироқ бу протоколнинг қуйидагича камчилиги мавжуд. B – фойдаланувчи А – фойдаланувчининг номидан иш кўриши мумкин.

1) B – фойдаланувчи арбитрга ўзининг ва 3 чи C – фойдаланувчининг исмини узатади:

$$B \rightarrow W\colon B, C.$$

2) Арбитр B – фойдаланувчининг исмини ва очиқ калитини бирлаштириб ўзининг ёпиқ калити ёрдамида рақамли имзо қўяди. Шунингдек у C – фойдаланувчининг исмини ва очиқ калитини бирлаштириб ўзининг ёпиқ калити билан рақамли имзо қўяди. Арбитр иккала маълумотни ҳам B – фойдаланувчига узатади:

$$W \rightarrow B\colon S_W(B, k_B),\ S_W(C, k_C).$$

3) B – фойдаланувчи арбитрнинг очиқ калити ёрдамида рақамли имзони текширади. Агар имзо тўғри бўлса, C – фойдаланувчининг очиқ калитини ҳақиқий деб қабул қилади. У A – фойдаланувчидан олган рақамли имзо қўйилган маълумотни C – фойдаланувчининг очиқ калити ёрдамида шифрлайди. B – фойдаланувчи бу шифрматнга арбитрдан қабул қилган иккита маълумотни бирлаштириб, C – фойдаланувчига узатади:

$$B \rightarrow C\colon E_C(S_A(k, t_A))\ S_W(A, k_A),\ S_W(C, k_C).$$

4) C – фойдаланувчи арбитрнинг очиқ калитидан фойдаланиб унинг рақамли имзосини текширади. Агар имзо тўғри бўлса, A – фойдаланувчининг очиқ калитини ҳақиқий деб қабул қилади. У ўзининг ёпиқ калити билан шифрматнни дешифрлайди. Ҳосил бўлган маълумотдан A – фойдаланувчининг рақамли имзосини текширади. Агар имзо хақиқий бўлса, сеанс калит сифатида k ни қабул қилади. Шунингдек C – фойдаланувчи вақт белгисини текшириб, маълумотнинг яқин вақт ичида узатилганига ишонч ҳосил қилади.

Энди C – фойдаланувчи ўзини A – фойдаланувчи билан алоқа ўрнатган деб ҳисоблайди. B – фойдаланувчи эса уни осонгина алдади. Ҳақиқатдан ҳам вақт белгиси ўзининг вақт интервалидан ўтгунгача B – фойдаланувчи тармоқдаги ихтиёрий фойдаланувчини алдаши мумкин. Лекин буни осонгина бартараф этиш мумкин. Бунинг учун 3) – босқичда шифрланиши керак бўлган маълумотга фойдаланувчиларнинг исмларини қўшиш керак:

$$E_A(S_A(A, B, k, t_A))\ S_W(A, k_A),\ S_W(B, k_B).$$

Энди B – фойдаланувчи эски маълумотни C – фойдаланувчига такроран узата олмайди, чунки бу фақат A ва B – фойдаланувчилар сеанс алоқаси учун яратилгани яққол намоён бўлади. Ана шу тарзда *Denning – Sacco* протоколининг камчилиги бартараф этилади.

Woo-Lam протоколи

Бу протоколда ҳам асимметрик шифрлаш алгоритмидан фойдаланилади.

1) A – фойдаланувчи ўзининг ва B – фойдаланувчининг исмини арбитрга узатади:

$$A \rightarrow W\colon A, B$$

2) Арбитр B – фойдаланувчининг очиқ калитига ўзининг ёпиқ калити билан рақамли имзо қўяди ва A – фойдаланувчига узатади:

$$W \rightarrow A\colon S_W(k_B).$$

3) A – фойдаланувчи арбитрнинг имзосини текширади. Агар имзо тўғри бўлса, B – фойдаланувчининг очиқ калитини хақиқий деб қабул қилади. Сўнгра у ўзининг тасодифий сонини B – фойдаланувчининг очиқ калити билан шифрлайди. A – фойдаланувчи ўзининг исмини ва шифрматнни B – фойдаланувчига узатади:

$$A \rightarrow B\colon A, E_B(R_A).$$

4) B – фойдаланувчи ўзининг ёпиқ калити ёрдамида шифрматнни дешифрлайди ва натижада A – фойдаланувчининг тасодифий сонига эга бўлади. Энди у тасодифий сонни арбитрнинг очиқ калити ёрдамида шифрлайди. Сўнгра A – фойдаланувчининг, ўзининг исми ва шифрматнни арбитрга узатади:

$$B \rightarrow W\colon A, B, E_{k_T}(R_A).$$

5) Арбитр A – фойдаланувчининг очиқ калити k_A га ўзининг ёпиқ калити билн рақамли имзо қўяди. У A – фойдаланувчининг тасодифий сони, сеанс калит, A ва B – фойдаланувчиларнинг исмларини бирлаштириб, ўзининг ёпиқ калити билан рақамли имзо қўяди ва B –

фойдаланувчининг очиқ калитидан фойдаланиб шифрлайди. Арбитр иккала маълумотни ҳам B – фойдаланувчига узатади:

$$W \rightarrow B\colon S_W(k_A),\ E_{k_B}(S_W(R_A,\ k,\ A,\ B)).$$

6) B – фойдаланувчи арбитрнинг рақамли имзосини текширади. Агар рақамли имзо тўғри бўлса, A – фойдаланувчининг очиқ калитини ҳақиқий деб қабул қилади. Сўнгра у шифрматнни ўзининг ёпиқ калити билан дешифрлайди. Ҳосил бўлган маълумотдаги абитрнинг рақамли имзосини текширади. Агар имзо тўғри бўлса, k – сеанс калитни хақиқий деб қабул қилади. Энди B – фойдаланувчи 5) – босқичда арбитрдан қабул қилган рақамли имзо қўйилган маълумотга ўзининг тасодифий сонини бирлаштириб, A – фойдаланувчининг очиқ калити билан шифрлайди. Ҳосил бўлган шифрматнни A – фойдаланувчига узатади:

$$B \rightarrow A\colon E_{k_A}(S_W(R_A,\ k,\ A,\ B),\ R_B).$$

7) A – фойдаланувчи шифрматнни ўзининг ёпиқ калити ёрдамида дешифрлайди. У арбитрнинг рақамли имзосини текширади. Агар имзо тўғри бўлса, k – сеанс калитни ҳақиқий деб қабул қилади. Сўнгра B – фойдаланувчининг тасодифий сонини k – сеанс калит ёрдамида шифрлаб B – фойдаланувчига узатади:

$$A \rightarrow B\colon E_k(R_B).$$

8) B – фойдаланувчи сеанс калит ёрдамида шифрматнни дешифрлайди. Ҳосил бўлган ўзининг тасодифий сонини ўзгарган ёки ўзгармаганлигини аниқлайди. Агар у ўзгармаган бўлса, A – фойдаланувчи билан алоқа ўрнатилганига ишонч ҳосил қилади. Ана шу тарзда икки томонлама идентификация таъминланади.

10.3-жадвал. Асимметрик шифрлаш алгоритмидан фойдаланиб калит узатиш протоколлари таҳлили (арбитр иштирок этмаган ҳол учун)

№	Протокол номи	A	B	k_A	k_B	k	t	S	Бўлиши мумкин бўлган хужум
1.	Вақт меткасидан фойдаланиб калитни узатиш протоколи	+				+	–		Криптоаналитик бу маълумотни маълум вақтдан сўнг В фойдаланувчига қайта узатиши мумкин
		–				+	+		В фойдаланувчи бу маълумотни ким томонидан узатилганини билмайди

Давоми

№	Протокол номи	A	B	k_A	k_B	k	t	S	Бўлиши мумкин бўлган хужум
2.	ЭРИ ёрдамидаги протоколлар	+	+				+	–	В фойдаланувчи бу маълумотни ким томонидан узатилганини билмайди
		+	+				–	+	Криптоаналитик бу маълумотни маълум вақтдан сўнг В фойдаланувчига қайта узатиши мумкин
3.	Тасодифий сонлардан фойдаланиб калитни генерация қилиш протоколи	+		–	–				Протокол бажарилиши натижасида сеанс калит генерация қилинмайди
4.	Диффи-Хеллман протоколи	«Ўртадаги киши» хужумига бардошли эмас							
5.	Гуруҳдаги 3 та фойдаланувчилар учун калитларни генерация қилиш протоколи	«Ўртадаги киши» хужумига бардошли эмас							

10.4-жадвал. Асимметрик шифрлаш алгоритмидан фойдаланиб калит узатиш протоколлари таҳлили (арбитр иштирок этган ҳол учун)

№	Протокол номи	A	B	K_D	k_B	R_A	t	S	Бўлиши мумкин бўлган хужум
1.	Dass протоколи	+	+	+	+		+	–	Томонлар ўртасида ўзаро идентификация таъминланмайди
		+	+	–		+	+	+	В фойдаланувчи ЭРИ ни текшириш имконига эга бўлмайди.
2.	Ву-Лам протоколи	+	+			–		+	А фойдаланувчи арбитрни идентификация қилмайди
		+	+			–		+	Фойдаланувчилар бир-бирларини идентификация қилмайдилар
3.	Деннинг-Сакко протоколи	Протокол якунлангандан сўнг B фойдаланувчи бошқа C фойдаланувчи билан алоқа ўрнатиши учун A фойдаланувчининг номидан иш кўриши мумкин							

10-боб бўйича хулосалар

Ушбу бобда:

1. Танланган криптотизим қанчалик криптобардошли ва ишончли бўлмасин, ундан амалда фойдаланиш жараёнлари калитларини бошқариш:

– барча калитларнинг ўзаро боғлиқ ҳолда, яъни бир бутун ҳолда ишлаш жараёнини таъминлаш (*калитлар генерацияси*);

– калитлар тўпламининг мақсадли кенгайиб боришини таъминлаш (*калитларнинг тўпланиши*);

– калитларларни фойдаланувчилар доирасида тақсимлаш (*калитларнинг тақсимланиши*) жараёнлари масалалари билан боғлиқлиги изоҳланди.

2. Бир томонлама функция сифатида олинган $f(x)=\alpha^x(\mathrm{mod}\ p)$ акслантириш асосида махфий калитларни очиқ тақсимлашнинг Диффи-Хеллман алгоритми криптографик асослари ёритилди.

3. Умумфойдаланадиган ахборот – коммуникация тармоғида махфий алоқа тизими фойдаланувчиларига калитларни тарқатишнинг криптографик протоколларининг асослари муҳокама қилиниб, очиқ калитлар тўпламининг муҳофазасининг ҳам зарурлиги ва уларни фойдаланувчиларга муҳофазаланган ҳолда етказиш алгоритми протоколи шифрлаш, хэшлаш ва ЭРИ алгоритмлари воситаларидан фойдаланиб қандай амалга оширилиши кўрсатилди.

4. Ушбу воситалар:

E – симметрик шифрлаш алгоритми;

t_A – вақт белгиси;

r_A – A – фойдаланувчининг тасодифий сони;

n_A – A – фойдаланувчининг генерация қилиш тартиб рақами;

B – B – фойдаланувчининг идентификацион рақами;

k_{AB} – иккала томонга ҳам маълум бўлган калит орқали калитларни алмашувчи протоколлар ва уларнинг криптоҳужумга заиф томонлари таҳлил қилинди.

5. Асимметрик калитли алгоритмлардан фойдаланиб ҳужжатли маълумотларнинг махфийлигини таъминловчи протоколлар криптографик асосларининг моҳиятлари изоҳланди.

6. Асимметрик шифрлаш алгоритмидан фойдаланиб калитларни алмашиш протоколларининг тузилишлари таҳлил қилинди.

ХУЛОСА

Маълумотларни муҳофазалаш услубларининг амалда қўлланиши масаласи қуйидаги ўзининг муҳим томонларига эга:

– криптографик алгоритмларнинг амалда қўлланилишини таъминловчи воситаларни яратиш ёки ишлаб чиқариш;

– яратилган ёки ишлаб чиқилган воситалардан фойдаланиш услублари.

Ҳар бир криптографик услуб бирор алгоритмик тилда дастур тузишга ёки шу услубни амалдаги қўлланишини таъминловчи асбоб–ускуна жиҳозлари кўринишидаги яратилишнинг табиий омилларини туғдириши керак.

Криптографик алгоритмнинг дастурлаш воситаси амалда қўлланилганда формал математик алмаштириш амалларининг бирор чекли кетма-кетлигига асосланади.

Шифрлаш ва дешифрлаш амалларининг асбоб-ускуналар воситасида таъминланиши махсус электрон схемалар асосида амалга оширилади ва бунда гаммалаштириш услубини қўшиб олиб борилади. Чунки гаммалаштириш услуби юқори даражадаги криптобардошлиликни таъминлаб, нисбатан соддароқ амалий қўлланиш имкониятларига ҳам эга. Гаммалаштиришнинг асосини ташкил этувчи тасодифий битлар кетма-кетлигини ишлаб чиқарувчи генератор сифатида чизиқли ва чизиқли бўлмаган сонли акслантириш амалларини бажарувчи кўчиш (силжиш) регистрларидан фойдаланилади.

Бундай бошқаришнинг моҳияти, электрон қурилмаларда бирор ҳажмдаги маълумотни шифрлаб бўлингандан сўнг, электрон қурилманинг ҳотира регистрларига жойлашган мана шу шифрланган маълумотнинг белгиларини бирор услуб билан даврий равишда ўзгартирилиши билан боғлиқ. Чизиқли бўлмаган акслантириш амалларини бажарувчи кўчиш регистрларидан фойдаланиш тасодифий битлар кетма-кетлигидан иборат бўлган гамманинг мураккаб ҳолда вужудга келиши билан боғлиқ бўлиб, криптоанализ масалаларининг ечилишини мураккаблаштиради.

Дастурлаш услубларидан фойдаланиб маълумотларни муҳофазалашнинг қулайлиги ва ютуғи, унинг асосини ташкил этувчи криптографик шифрлаш алгоритмининг маълумотларни математик амаллар билан тез ва мақсадли ўзгартиришни таъминлаш имкониятларининг мавжудлигидадир.

Дастурлаш услубининг асосий камчилиги эса, унинг электрон схемалар кўринишдаги асбоб-ускуналар воситасида шифрлаш ва дешифрлашга нисбатан секин ишлашидадир.

Дастурлаш ва асбоб-ускуналар воситаларини қўшиб олиб боришга асосланган шифрлаш ҳамда дешифрлаш криптографик услублари компьютерлар тизими билан боғлиқ бўлган ахборотларни муҳофазалаш соҳасида компьютерларга «криптографик қўшимча-процессор»ларнинг ишлаб чиқарилиши ва ўрнатилиши билан боғлиқ. Бу «криптографик қўшимча-процессор» ҳисоблаш қурилмаси бўлиб, криптографик амаллар: модул бўйича қўшиш, регистрларнинг силжишини таъминлаш ва бошқа шу каби вазифаларни бажаради. Бундай қурилмалар учун дастурлаш алгоритмини ўзгартириш билан шифрлаш жараёнида фойдаланилаётган услубларнинг ўзгаришини бошқариш мумкин.

Шундай қилиб, бирор ахборотлар тизимида маълумотларни криптографик услублар билан муҳофазалаш, шу ахборотлар тизими соҳасининг ўзига хос муҳим бўлган хусусиятли томонларини чуқур ва кенг таҳлил қилган ҳолда мос келувчи криптографик муҳофаза услубини танлашни тақазо этади.

Катта ҳажмдаги маълумотларни шифрлаш масалалари мультимедия ва юқори даражадаги ўтказиш ҳамда узатиш имкониятларига эга бўлган алоқа тармоқлари воситаларининг вужудга келиши билан боғлиқ.

Ахборотларни муҳофаза қилиш тўғрисида гап борганда кенг маънодаги бирор алифбода ифодаланган маълумот тушунилади. Аммо ҳозирги замонавий ахборот тизимларидаги маълумотлар матн кўринишдаги ҳусусиятдан фарқли ўлароқ юқори даражада ривожланиб бораётган алоқа технологияларининг маҳсули бўлган:

– факс, тасвир ва сўз алоқа тизимлари;

– овозли почта;

– тасвирли конференциялар ўтказишда тасвирли (видио) алоқа тизимлари ва шу кабиларни ўз ичига олади. Бундан эса криптология ютуқларидан фойдаланиб шу алоқа тизимлари маълумотларни муҳофазалашнинг ўзига хос услубларини ишлаб чиқиш масалаларини ечиш вазифалари келиб чиқади.

Қуйида келтирилаётган жадвалда ахборотларни шифрлаш, кодлаштириш ва сиқиш жараёнларининг мақсад нуқтаи назаридан турлича эканлиги кўринади.

Ахборотни акслантириш тури	Мақсади	Акслантиришдан сўнг ахборот ҳажмининг ўзгариши
Шифрлаш	– махфий маълумотни алоқа канали тизимида жўнатиш; – маълумотни рақиб томонидан ўзгартирилишига йўл қўймасдан унинг ҳақиқийлигини таъминлаш	Одатда ўзгармайди, фақат рақамли сигнатура ва имзо ҳисобига кўпаяди
Ҳар-хил ҳалақит берилишларига бардошли бўлган кодлаштириш	Маълумотларни алоқа канали тизимида ҳар-хил ҳалақит беришлар эвазига ўзгаришидан муҳофазалаш	Кўпаяди
Сиқиш (компрессиялаш)	Сақланаётган ва узатилаётган маълумотлар ҳажмини қисқартириш	Камаяди

Кўриниб турибдики, бу уч турдаги маълумотларни акслантириш бир-бирини тўлдиради ва уларнинг ҳаммасидан биргаликда фойдаланиш алоқа тармоғи (канали) тизимларидан самарали фойдаланишга олиб келади.

Сизга тақдим этилаётган ушбу китобда криптология фанининг вужудга келиш жараёнларининг бизга очиқ манбалардан маълум бўлган баъзи математик асослари, асосий тушунча ва таърифлари, криптотизимларга қўйиладиган талаблар, криптоалгоритмларнинг хусусиятларига кўра таснифлаш (классификациялаш) ва уларнинг криптобардошлилигининг асослари, криптографик хэш-функциялар, электрон рақамли имзо алгоритмлари ва уларнинг моҳияти, криптографик алгоритмларни амалда – алоқа тармоқларида қўлланилишида тармоқнинг иккита ва ундан ортиқ фойдаланувчилари қатнашадиган ахборотлар алмашинуви жараёнлари тартиб ва қоидаларини ифодаловчи – криптографик протоколлар, криптоалгоритмлар учун бардошли калитлар ишлаб чиқиш ва уларни криптографик протоколлар асосида фойдаланувчиларга тақсимлаш ва криптографиянинг бошқа масалалари ҳамда уларнинг ечимлари ҳақида сўз юритилди.

Ахборот тизимини ташкил этувчи маълумотлар тўпламининг қандай тузилишда эканлиги ҳамда маълумотнинг муҳим ва аҳамиятлилигидан келиб чиқадиган махфийлик даражаси ва шу каби ўзига хос криптомуҳофаза хусусиятларини таҳлил қилиб, муҳофаза мақсадини ифодаловчи шарт (критерий) асосида, ҳар бир алоҳида ахборотлар тизимига мос келувчи криптографик услуб танланади. Мақсадни ифодаловчи шартдан келиб чиқиб, мос келувчи криптографик тизимни танлашнинг ягона услуби мавжуд бўлмасада, қуйидаги:

– калитларни аниқлаш мумкинлигининг эҳтимоли;

– мумкин бўлган калитлар тўпламининг қуввати, яъни мумкин бўлган калитлар тўплами элементларининг сони ва шулар каби криптобардошлилик билан боғлиқ бўлган сонли баҳолаш имкониятини берувчи тушунчалардан фойдаланиш мумкин.

Умуман олганда, муҳофаза мақсадини ифодаловчи шарт, қуйидаги асосли ҳолатларни (факторларни) назарда тутувчи, криптотизимга қўйиладиган талабларни:

– шифрматн тузилишига асосланиб шифрни очиш ёки онгли равишда маълумотларни ўзгартириш (модификациялаш) имкониятининг йўқлиги;

– муҳофазалашда фойдаланиладиган тартиб-қоидаларни (протоколни) такомиллаштириб бориш имконининг мавжудлигини таъминлашни;

– калит сифатида фойдаланилган маълумотнинг ҳажмини мумкин қадар озайтириш имконияти борлигини;

– мумкин қадар кам сарф–ҳаражат билан амалда қўлланилишини таъминлаш мумкинлигини;

– юқори даражадаги иш унумдорлигини, яъни етарли даражада тез шифрлаш ва дешифрлаш алгоритмларига асосланишини ўз ичига олган бўлиши керак.

Ҳар қандай криптографик алгоритмлар тизими ахборотлар тизимида маълумотлар алмашувининг узлуксиз жараёнида: қулайлик, ишончлилик, криптобардошлилик ва шу каби биз юқорида кўриб ўтган қатор талабларни қаноатлантириши лозим.

АДАБИЁТЛАР

1. *Жельников В.* Криптография от папируса до компьютера. М., ABF, 1997. – 336 с.

2. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии: Учебное пособие, 2-е изд. – М.: Гелиос АРВ, 2002. –480 с.

3. *Vernam G.S.* Cipher printing telegraph systems for secret wire and radio telegraphic communications, «J. Amer. Inst. Elec. Eng., vol. 55, pp. 109–115, 1926.

4. *Шенон К.Э.* Теория связи в секретных тизимх. В кн.: Шенон К.Э. Работы по теории информации и кибернетике. – М.: ИЛ, 1963, том 1. – С. 333–402.

5. *Шенон К.Э.* Теория связи в секретных тизимх. В кн.: Шенон К.Э. Работы по теории информации и кибернетике. – М.: ИЛ, 1963, том 2. – С. 243–332.

6. *Diffie W. and Hellman M.E.* «New directions in cryptography» IEEE Trans. Informat. Theory, vol. IT–22, pp. 644–654, Nov. 1976.

7. *R. C. Merkle* «Secure communication over insecure channels», Comm. ACM, pp. 294–299, Apr. 1978.

8. Дейтель Г. Введение в операционные системы. Том 2. – М.: Мир, 1987, с. 357–371.

9. *Феллер В.* Введение в теорию вероятностей и ее приложения. Том 2. – М.: Мир, 1984.

10. *Кнут Д.* Искусство программирования для ЭВМ. Том 1. Основные алгоритмы. М.: Мир, 1976.

11. *Гэри М. , Джонсон Д.* Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982.

12. *Simmons G. J.* «Authentication theory/coding theory, in Advances in Cryptology, Proceedings of CRYPTO 84, G. R. Blakley and D. Chaum, Eds. Lecture Notes in Computer Sciense, No. 196. New York, NY: Springer, 1985, pp. 411–431

13. *Бабаш А. В., Шанкин Г. П.* Криптография. – М.: Лори Гелиос АРВ, 2002. – 512 с.

14. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: издательство ТРИУМФ, 2003 – 816 с.

15. *Молдовян Н.А., Молдовян А.А., Еремеев М.А.* Криптография: от примитивов к синтезу алгоритмов. –СПб.: БХВ – Петербург, 2004. – 448 с.

16. *Новиков П.С.* Элементы математической логики. – М.: ИЛ, 1973.

17. *Логачёв О.А., Сальников А.А., Ященко В.В.* Булевы функции в теории кодирования и криптологии. – М.: Изд. МЦНМО, 2004. – 470 с.

18. *Фомичев В.М.* Дискретная математика и криптология. – М., «ДИАЛОГ-МИФИ», 2003. – 400 с.

19. *Коблиц Н.* Курс теории чисел и криптографии. – М.: Научное изд-во ТВП, 2001 г. – 261 стр.

20. *Харин Ю.С., Берник В.И., Матвеев Г.В., Агиевич С.Г.* «Математические и комьютерные основы криптологии» ООО «Новое знание» 2003 г. 381 стр.

21. *Молдавян А.А., Молдавян Н.А., Гуц Н.Д., Изотов Б.В.* «Криптография. Скоростные шифры» Санкт-Петербург. «БХВ – Петербург» 2002 г. 439 стр.

22. *Молдавян А.А., Молдавян Н.А.* Введение в криптосистемы с открытым ключом. Санкт – Петербург «БХВ – Петербург» 2005 г. 288 с.

23. *Ростовцев А.Г., Маховенко Е.Б.*, Теоретическая криптография. НПО «Профессионал», Санкт-Петербург. 2004 г. – 478 стр.

24. *Василенко О. Н.* Теоретико-числовые алгоритмы в криптографии. – М., МЦНМО, 2003. – 328 с.

25. *Зензин О.С., Иванов М.А.* Стандарт криптографической защиты – AES.

26. Конечные поля /Под ред. *М.А. Иванова* – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с.

26. *Акбаров Д.Е.* Криптография, Стандарты алгоритмов криптографической защиты информации и их приложения. – Т., 2007. – 188 с.

27. *Венбо Мао.* Современная криптография. Теория и практика. –Москва–Санкт-Петербург–Киев: Лори Вильямс, 2005. – 768 с.

28. *Иванов М.* Криптографические методы защиты информации в компьютерных тизимх и сетях. – М., «Кудиц-Образ», 2001, – 368с.

29. *Столлингс В.* Криптография и защита сетей: принципы и практика. – М., Изд. дом «Вильямс», 2001. – 672 с.

30. *Анохин М.И., Варновский Н.П., Сидельников В.М., Ященко В.В.* Криптография в банковском деле. – М., Изд. МИФИ, 1997.

31. O'z DSt 1106: 2006. Государственный Стандарт Узбекистана. Информационная технология. Криптографическая защита информации. Функция хэширования. – Ташкент. Узбекское агентство стандартизации, метрологии и сертификации. 2006.

32. Shafi Goldwasser, Mihir Bellare. Lecture Notes on Cryptography. Cambridge, Massachusetts, August, 1999. – 268 p.

33. *Menezes A., P. van Oorshot, Vanstone S.* Handbook of Applied Cryptography. CRC Press, 1996. – 780 p.

34. *Молдовян А. А., Молдовян Н. А., Советов Б. Я.* Криптография. – Санкт-Петербург, Изд. «Лань», 2001. – 224 с.

35. ГОСТ Р 34.11 – 94. Государственный Стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. Функция хэширования. – М.: Издательство стандартов, 1994.

36. Federal Information Processing Standards Publication 180–2. Secure Hash Standard. 2002 August 1.

37. Federal Information Processing Standards Publication 198. The Keyed-Hash Message Authentication Code (HMAC). 2002 March 6.

38. *Ященко В.В. и др.* Введение в криптографию. – М., МЦНМО, 2000. – 288 с.

39. *Чмора А.* Современная прикладная криптография. – М., Гелиос АРВ, 2002. – 256 с.

40. *Аграновский А.В., Хади Р.А.* «Практическая криптография» –М.: СОЛОН-Пресс. 2002 г. – 254 стр.

41. *Ростовцев А., Михайлова М.* Методы криптоанализа классических шифров.

42. *Хасанов Х. П.* Такомиллашган диаматрицалар алгебралари ва параметрли алгебра асосида криптотизимлар яратиш усуллари ва алгоритмлари. – Т., 2008. – 208 б.

МУНДАРИЖА

V боб. УЗЛУКСИЗ ШИФРЛАШ АЛГОРИТМЛАРИНИНГХОССАЛАРИ

VI боб. АСИММЕТРИК ШИФРЛАШ АЛГОРИТМЛАРИВА УЛАРНИНГ КРИПТОГРАФИЯ МАСАЛАЛАРИНИ ЕЧИШДА ҚЎЛЛАНИЛИШИ

VII боб. ХЭШ-ФУНКЦИЯ ВА УНИНГ АХБОРОТНИМУҲОФАЗА ҚИЛИШ МАСАЛАЛАРИНИЕЧИШДА ҚЎЛЛАНИЛИШИ

VIII боб. ЭЛЕКТРОН РАҚАМЛИ ИМЗО АЛГОРИТМЛАРИ

IX боб. БАРДОШЛИ КАЛИТЛАР ИШЛАБ ЧИҚИШ (ГЕНЕРАЦИЯЛАШ)

X боб. КАЛИТЛАРНИ ТАҚСИМЛАШНИ БОШҚАРИШ АЛГОРИТМЛАРИ (ПРОТОКОЛЛАРИ)

Акбаров Давлатали Егиталиевич

АХБОРОТ ХАВФСИЗЛИГИНИ ТАЪМИНЛАШНИНГ КРИПТОГРАФИК УСУЛЛАРИ ВА УЛАРНИНГ ҚЎЛЛАНИЛИШИ

Тошкент – «Ўзбекистон маркаси» нашриёти – 2009

Муҳаррир	*С. Хашимова*
Бадиий муҳаррир	*Ж. Гурова*
Техник муҳаррир	*А. Салихов*
Мусаҳҳиҳ	*Г. Азимова*
Компьютерда тайёрловчи	*Б. Бабаходжаева*

Босишга 03.02.09. да рухсат этилди. Бичими $60\times90^{1}/_{16}$.
«Times New Roman» гарнитурада офсет босма усулида босилди.
Шартли б.т. 27,0. Нашр-ҳисоб т. 27,2. Адади 500 нусха.
38- рақамли буюртма. Бепул.

«Niso Poligraf» ШКда, чоп этилди.
100182, Тошкент, Ҳ. Бойқаро кўчаси, 41.