M.M. KARIMOV, J.T. ARZIYEVA, Z.T. XUDOYQULOV

BIR MARTALI PAROLLAR ASOSIDA FOYDALANUVCHILARNI AUTENTIFIKATSIYALASH PROTOKOLLARI

MONOGRAFIYA

O‘ZBEKISTON RESPUBLIKASI AXBOROT
TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI
RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

M.M. Karimov, J.T. Arziyeva, Z.T. Xudoyqulov

BIR MARTALI PAROLLAR ASOSIDA FOYDALANUVCHILARNI AUTENTIFIKATSIYALASH PROTOKOLLARI

MONOGRAFIYA

TOSHKENT
«IQTISOD-MOLIYA»
2021

UDK: 004.057.4
KBK: 32.81

Taqrizchilar: *t.f.d., dots.* B.B.Mo'minov,
PhD B.K.Yusupov

K 25 Bir martali parollar asosida foydalanuvchilarni autentifikatsiyalash protokollari: *Monografiya* / M.M. Karimov, J.T. Arziyeva, Z.T. Xudoyqulov / – T.: "Iqtisod-Moliya", 2021. – 120 b.

Monografiya axborot kommunikatsiya tizimlarida autentifikatsiya usullari, ularda mavjud tahdidlar, bir martali parollarga asoslangan autentifikatsiya usullarining tahlili, psevdotasodifiy sonlar generatori va unga asoslangan bir martali parollarni generatsiyalash usuli, ularning xavfsizligini baholash masalalariga bag'ishlangan. Bundan tashqari, monografiyada bir martali parollarga asoslangan qator autentifikatsiya usullari takomillashtirilgan, savol-javob mexanizmiga asoslangan yangi ikki tomonlama autentifikatsiya usuli ishlab chiqilgan hamda xavfsizlik, samaradorlik omillari bo'yicha baholangan.

Ushbu monografiya axborot xavfsizligi va ruxsatlarni nazoratlash, xususan, psevdotasodifiy sonlar generatorini ishlab chiqish va bir martali parolga asoslangan autentifikatsiyalash sohasida ilmiy izlanish olib borayotgan mutaxassislar uchun tavsiya etiladi hamda mazkur sohada oliy ta'lim muassasalari talabalari va magistrlarining keng doirasi ham foydalanishi mumkin.

UDK. 004.057.4
KBK: 32.81

ISBN 978-9943-13-987-9

KIRISH

Jahonda autentifikatsiyaning an'anaviy usullarida zaifliklarni ortib borishi natijasida tashkilotlarda qat'iy yoki ko'p faktorli autentifikatsiya usullaridan foydalanish ko'lamini kengaytirishga alohida e'tibor qaratilmoqda. Axborot – kommunikatsiya tizimlari rivojining hozirgi zamon bosqichida foydalanuvchilarni haqiqiyligini tekshirishda ko'p faktorli autentifikatsiya usullaridan foydalanish muhim hisoblanadi. Xususan, Mircosoft kompaniyasining xavfsizlik bo'yicha xodimi, Aleks Vaynert so'zlariga ko'ra "Ko'p faktorli autentifikatsiya usullaridan foydalanish natijasida avtomatik ravishda amalga oshiriluvchi hujumlarning 99.9%i bloklangan"[1]. Bu yo'nalishda rivojlangan mamlakatlarda, jumladan, AQSh, Angliya, Yaponiya, Fransiya, Niderlandiya va boshqa davlatlarda foydalanuvchilarni kafolatli autentifikatsiyasini ta'minlovchi apparat va dasturiy vositalarni ishlab chiqish muhim ahamiyat kasb etmoqda.

Jahonda foydalanuvchi haqiqiyligini kafolatli tekshirish imkoniyatini beruvchi, foydalanishga qulay va kam xarajatli ko'p faktorli autentifikatsiya usullarini yaratishga yo'naltirilgan ilmiy tadqiqot ishlari olib borilmoqda. Bu borada, amalda keng qo'llaniluvchi parolga asoslangan autentifikatsiya usulida mavjud xavfsizlik muammolarini bartaraf etishda qo'shimcha faktor sifatida bir martali parollardan foydalanishning samarali va xavfsiz usullarini ishlab chiqish muhim vazifalardan biri hisoblanadi. Shu bilan birga, bir martali parollarni yaratish usullarini va unga asoslangan mavjud autentifikatsiya usullarining bardoshligini tahlil qilish hamda takomillashtirish zarur bo'lmoqda.

Respublikamizda davlat va xo'jalik boshqaruv organlarida foydalanuvchilarni haqiqiyligini tekshirishda va moliyaviy operatsiyalarni kafolatli amalga oshirishda jarayon xavfsizligini ta'minlashga qaratilgan keng qamrovli chora-tadbirlar amalga oshirilmoqda. 2017–2021-yillarda O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasida, jumladan "...axborot xavfsizligini ta'minlash va axborotni himoyalash tizimini takomillashtirish, axborot sohasidagi tahdidlarga qarshi o'z vaqtida va

[1] https://www.zdnet.com/article/microsoft-using-multi-factor-authentication-blocks-99-9-of-account-hacks/

munosib qarshilik ko‘rsatish”[2] vazifalari belgilangan. Mazkur vazifalarni bajarishda foydalanuvchilarni haqiqiyligini tekshirishning samarali, xavfsiz va parolga asoslangan usullari hamda vositalarini ishlab chiqish muhim vazifalardan biri hisoblanadi.

Monografiyaning birinchi bobi axborotni himoyalashda autentifikatsiyalash usullarining o‘rni, autentifikatsiya usullari va ularda mavjud tahdidlar tahlili va bir martali parollarga asoslangan autentifikatsiya usullarining tahliliga bag‘ishlangan.

Monografiyaning ikkinchi bobida bir martali parollarni generatsiyalash usullari tahlil qilingan va psevdotasodifiy sonlar generatori, unga asoslangan bir martali parollarni generatsiyalash usuli va algoritmi taklif etilgan.

Monografiyaning uchinchi bobida bir martali parollar generatori asosida autentifikatsiya usullaridagi xavfsizlik muammolarini bartaraf etishga va ularni tahlil qilishga bag‘ishlangan bo‘lib, dastlab amalda foydalanilayotgan bir martali parollarga asoslangan autentifikatsiya usullaridagi xavfsizlik muammolari keltirilgan va shundan so‘ng, ularni bartaraf etish usuli taklif etiladi.

Monografiyaning to‘rtinchi bobi yuqorida keltirilgan bir martali parollarga asoslangan autentifikatsiya protokollarining tahliliga va tatbiqiga bag‘ishlangan bo‘lib, taklif etilgan usul, algoritmlarning samaradorligi va xavfsizligi asoslangan.

[2] O‘zbekiston Respublikasi Prezidenti 2017-yil 7-fevraldagi PF-4947-son “O‘zbekiston Respublikasini yanada rivojlantirish bo‘yicha Harakatlar strategiyasi to‘g‘risida”gi Farmoni

I BOB. AXBOROT-KOMMUNIKATSION TIZIMLARIDA AUTENTIFIKATSIYA MUAMMOLARI

1.1. Axborot-kommunikatsion tizimlarida autentifikatsiya usullari va vositalarining o'rni

Ma'lumotlarni uzatishda ochiq kanallardan foydalanish buzg'unchilik harakatlari uchun potensial imkoniyatlarni yaratadi. Shuning uchun foydalanuvchiga bog'liq holda axborot xavfsizligini ta'minlashning asosiy vazifalaridan biri – tomonlardan birini boshqasiga ishonishiga xizmat qiluvchi usul va vositalardan foydalanishdir. Ushbu muammoni bartaraf etish uchun tomonlarning haqiqiyligini tasdiqlash imkonini beruvchi maxsus usullardan foydalaniladi.

Kompyuter tizimlarida ro'yxatga olingan har bir subyektga (foydalanuvchi yoki uning nomidan harakat qiluvchi jarayonga) tegishli bo'lgan axborot mavjud bo'lib, u subyektni bir ma'noli identifikatsiyalashga xizmat qiladi. Bu axborot subyektga tegishli bo'lgan raqam yoki belgilardan iborat bo'lishi mumkin va u *identifikator* deb yuritiladi. Agar foydalanuvchi identifikatorga ega bo'lsa, u tarmoqda qonuniy deb hisoblanadi. Identifikatorga ega bo'lmagan foydalanuvchilar esa noqonuniy foydalanuvchi hisoblanadi. Shu sababli, foydalanuvchilar tizimda biror resursga murojaatni amalga oshirishidan oldin, kompyuter tizimi bilan hamkorlikda *identifikatsiya* va *autentifikatsiya* deb ataluvchi birlamchi jarayondan o'tishi talab etiladi.

Identifikatsiya – foydalanuvchini identifikatori yordamida tanib olish muolajasi bo'lib, ushbu funksiya foydalanuvchi tarmoqda yoki tizimda kirishga urinish vaqtida amalga oshiriladi. Bunda foydalanuvchi o'zining identifikatori yordamida tizimga xabar beradi va uni bazada mavjudligi haqidagi tekshiruv natijasi qaytariladi.

Autentifikatsiya – murojaat qilgan foydalanuvchi, qurilma yoki jarayonning haqiqiyligini tekshirish muolajasi bo'lib, ushbu tekshiruv foydalanuvchini (qurilma yoki jarayonni) o'zini kim sifatida e'lon qilganiga ishonchli ko'rsatishga xizmat qiladi. Ushbu muolajada foydalanuvchi va tizim orasida ma'lum turdagi axborot almashinuvi amalga oshirilib, u foydalanuvchining identifikatori va boshqa foydalanuvchilar bilmaydigan o'ziga tegishli axborot (masalan, parol yoki sertifikat)dan tashkil topgan bo'ladi [1, 5].

Identifikatsiya va autentifikatsiya o'zaro bog'liq bo'lgan jarayonlar bo'lib, subyektni tizimga tanitish va haqiqiyligini tekshirishni bir vaqtda

amalga oshiradi. Aynan ushbu jaryonlar foydalanuvchini tizimdan foydalanishga huquqi mavjudligini belgilab beradi. Subyektni identifikatsiyalash va autentifikatsiyalash amalga oshirilgandan so'ng avtorizatsiya amalga oshiriladi.

Avtorizatsiya – suyektga mazkur tizim ichida ma'lum huquqlarni va resurslardan foydalanish imtiyozini taqdim etish muolajasi bo'lib, aynan ushbu jarayon tizim ichida haqiqiy bo'lgan foydalanuvchilar uchun foydalanishning turli darajasini taqdim etadi. Boshqacha aytganda, ushbu muolaja har bir qonuniy foydalanuvchi uchun tashkilotning qat'iy talabini aniqlashga xizmat qiladi.

Autentifikatsiya va avtorizatsiya muolajalari bilan birgalikda keng qo'llanuvchi tushunchalardan biri – *ma'murlash* bo'lib, u foydalanuvchining tarmoqdagi resurslarga murojaatlari va harakatlarini ro'yxatga olishni maqsad qiladi. Ro'yxatga olingan ma'lumotlar asosida xavfsizlik insidentlarini tahlil qilish imkoniyatiga ega bo'linadi.

Yuqoridagi keltirilgan muolajalar ketma-ketligidan ko'rish mumkinki, autentifikatsiya qolganlari uchun muhim o'tishni ta'minlab beruvchi jarayon hisoblanib, uni xavfsizlik talabiga javob berishi keyingi jarayonlarning ishonchli amalga oshirilishini kafolatlaydi. Mos autentifikatsiya tashkilotga o'rnatilgan xavfsizlik talablarini aniqlaydi. Masalan, web-serverlardan foydalanish uchun foydalanuvchining anonim bo'lishi yoki oddiy "mehmon" tipida bo'lishining o'zi yetarli bo'lsa, moliyaviy tranzaksiyalar yoki bank amaliyotlarini bajarishda foydalanuvchilarni qat'iy autentifikatsiyadan o'tishi talab etiladi.

Foydalanuvchilarni autentifikatsiyalash jarayoni bir yoki bir nechta autentifikatsiya faktorlariga asoslanadi. Inson ko'rinishidagi foydalanuvchilarni autentifikatsiyalashda asosan quyidagi 4 ta faktor keng qo'llaniladi [6-8]:

1. *Foydalanuvchi bilgan biror bilim asosida:* parol, maxfiy ibora, PIN (Personal Identification Number) kod, foydalanuvchiga tegishli biror ma'lumot va h.k.;

2. *Foydalanuvchiga mavjud biror narsa asosida:* USB tokenlar, telefon, smart karta, dasturiy ko'rinishdagi token va h.k.;

3. *Foydalanuvchini xarakterlovchi biror narsa asosida:* barmoq izi, DNK fragmenti, yuz tasviri, ko'z qorachig'i va h.k.;

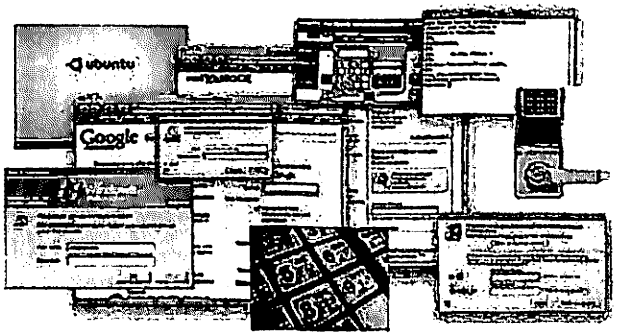
4. *Foydalanuvchi qila oladigan bir narsa asosida:* imzolash, harakat tarzi va h.k.

Foydalanuvchini autentifikatsiyalashda birdan ortiq faktorlardan foydalanish ba'zida yuqori xavfsizlikni ta'minlash uchun yetarlidek

ko‘rinsada, autentifikatsiyaning bardoshligi foydalanilgan usullarning bardoshligi bilan belgilanadi va ular odatda “ikki faktorli autentifikatsiya” deb ham yuritiladi. Masalan, ikki faktorli autentifikatsiyada foydalanuvchiga tegishli ma’lumot (masalan, uning onasining familiyasi) va oddiy PIN koddan foydalanilgan bo‘lsin. Bu holda autentifikatsiyada ikki faktordan foydalanilgan bo‘lsada, uni bardoshli deb aytib bo‘lmaydi.

Yuqorida keltirilgan faktorlarga asoslangan ko‘plab usullar mavjud bo‘lib, ularning barchasi o‘ziga xos afzallik va kamchiliklarga ega. Quyida keng qo‘llaniluvchi autentifikatsiya usullari bilan tanishib chiqiladi:

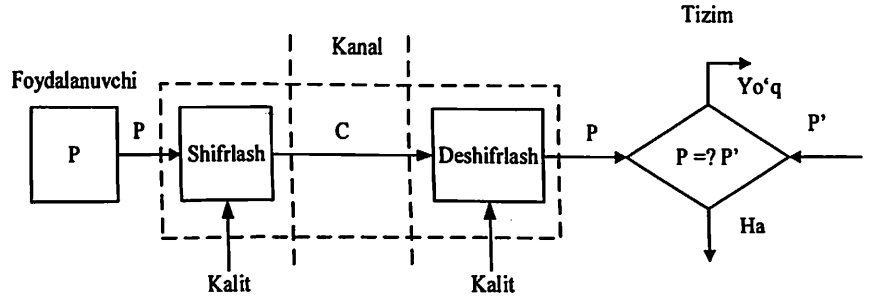
Taqsimlangan sirga asoslangan statik autentifikatsiya. PIN/ parolga asoslangan statik autentifikatsiya usuli amalda keng tarqalgan bo‘lib (1.1-rasm), ko‘p martali parollarga asoslangan autentifikatsiya deb ham ataladi. Autentifikatsiyaning mazkur usuli foydalanishga juda qulay bo‘lib, ortiqcha infratuzilma, vosita yoki qo‘shimcha narxni talab etmaydi. Bunda, autentifikatsiyadan o‘tishda tizim tomonidan zarur bo‘lgan maxfiy sirni (parolni yoki PIN kodni) kiritish talab etiladi. Kiritilgan sirni tizimdagisi bilan solishtirish natijasida foydalanuvchining haqiqiyligi aniqlanadi.



1.1-rasm. **Foydalanuvchilarni autentifikatsiyalashda keng qo‘llaniluvchi statik autentifikatsiya usuli**

Statik autentifikatsiyada foydalanilgan maxfiy sir parol, ibora, PIN kod, matn bo‘lmagan parol (grafik parol) ko‘rinishlarida bo‘lishi mumkin. Umumiy holda taqsimlangan maxfiy sirga asoslangan autentifikatsiya usulining sodda ko‘rinishi 1.2-rasmda keltirilgan. Bunga ko‘ra, foydalanuvchi o‘zi bilgan maxfiy sir P ni tizimga kiritadi. Foydalanuvchi va tizim tarmoq bilan bog‘langan (masofadan) yoki ular

bir tizimda bo'lishi mumkin. Kiritilgan maxfiy sir ba'zida ochiq holatda uzatilsa (masalan, PAP (Password Authentication Protocol) protokoli [69]), ba'zida bir tomonlama o'zgartirish (masalan, xesh-funksiyaga asoslangan CHAP (Challenge Handshake Authentication Protocol) protokoli [70]) yoki kriptografik akslantirish (shifrlash) asosida (natijasi shifrmatn deb ataladi, C) uzatiladi. Kiritilgan maxfiy sir va ro'yxatga olish jarayonida kiritilgani o'zaro mos kelsa, foydalanuvchi autentifikatsiyadan muvaffaqiyatli o'tgan hisoblanadi. Aks holda, foydalanuvchi haqiqiy emas deb topiladi.



1.2-rasm. **Statik autentifikatsiyalashning sodda ko'rinishi**

Amalda keng ko'lamda qo'llanilishiga qaramasdan, statik autentifikatsiya usulida quyidagi jiddiy xavfsizlik muammolari mavjud:

– parol onlayn tarzda o'g'irlanishi yoki tizimdan ruxsat olish uchun foydalanilishi (takrorlash hujumi) mumkin;

– PIN kodni kiritish vaqtida kameralar orqali yoki *keylogger* turidagi dasturiy yoki qurilma ko'rinishidagi vositalar orqali kompyuterdan kiritilgan parollarni muammosiz qo'lga kiritish mumkin;

– parollarni farazga asoslangan hujumlar, "qo'pol kuch" hujumi, lug'at bo'yicha hujum orqali qo'lga kiritish mumkin (hattoki, bu hujumni muvaffaqiyatsiz urinishlar sonini cheklash imkoni bo'lsada);

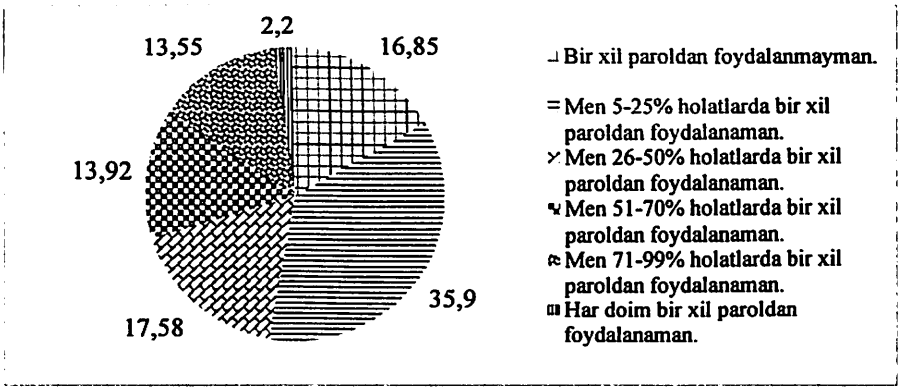
– o'g'irlangan parolning xesh qiymati vaqt o'tishi bilan "orqaga qaytarilishi" mumkin (masalan, keng tarqalgan parollarning xesh qiymatlariga solishtirish orqali, rainbows-tables [71]);

– parollar bir foydalanuvchi tomonidan boshqasiga oson oshkor qilinishi mumkin.

Statik autentifikatsiya usulini "qo'pol kuch" hujumi yoki lug'atga asoslangan hujumlarga qarshi himoyalashning imkoni mavjud emas. Mazkur hujumlar parollarni uzunligi va entropiyasi yetarli darajada

emasligiga asoslanadi. Boshqacha aytganda, murakkab (yuqori entropiyaga ega) va uzun ("qo'pol kuch" hujumiga bardoshli bo'lishi uchun) parollarni inson xotirasida saqlash murakkab.

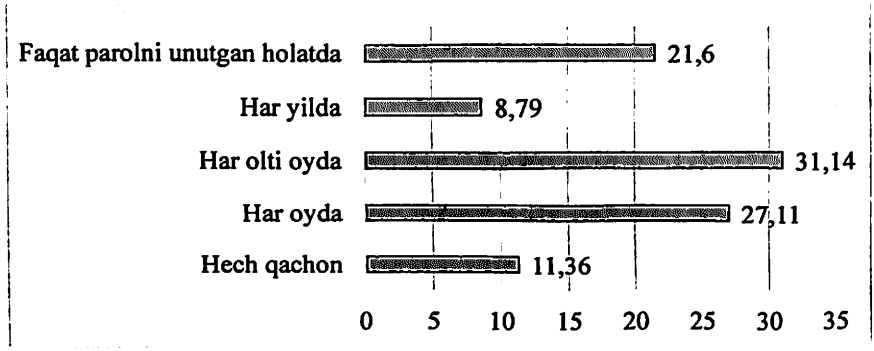
Bundan tashqari, foydalanuvchi qayd yozuvini, xususan, parolni oshkor bo'lishiga olib keladigan sabablardan biri – ko'plab qayd yozuvlarida bitta paroldan foydalanishdir. Murakkab parollarni va ko'p sonli parollarni inson xotirasida saqlash insondan kuchli xotirani talab etadi. Shu sababli, foydalanuvchilar tomonidan ko'p hollarda turli qayd yozuvlari uchun aynan bir xil parol foydalaniladi. 1.3-rasmda Cyclonis Password Security Report tomonidan web sahifalar/ onlayn xizmatlar uchun bitta paroldan foydalanishga oid statistik ma'lumotlar keltirilgan [72].



1.3-rasm. **Bitta paroldan foydalanishga oid Cyclonis Password Security Report statistik natijalari (%)**

Diagramma qayd etgani kabi 83,15% holatda foydalanuvchilar tomonidan bir qancha web sahifalar va onlayn xizmatlarda yagona paroldan foydalanish holatlari kuzatilgan. Bundan tashqari, qayd yozuvlarida parollarni o'rnatishda odatda foydalanuvchiga tegishli bo'lgan umumiy ma'lumotlardan foydalanish holatlari ko'p uchraydi. Umumiy holda parol sifatida keng foydalanilayotgan ma'lumotlar turi 1-ilovada keltirilgan [72].

Statik parollar bilan bog'liq bo'lgan yana bir muammolardan biri – ularni uzoq vaqt davomida o'zgartirilmasligi yoki uzaytirish muddatining uzunligidir. Ushbu holatni 1.4-rasmda keltirilgan natijalar tasdiqlaydi [72].



1.4-rasm. **Statik parollarning o'zgartirish muddatlari (%)**

Bundan tashqari, parollarni tarmoq orqali yuborishda jiddiy muammolar kelib chiqishi mumkin. Aksariyat hollarda parollar tarmoqda ochiq tarzda uzatilgani bois, kanal himoyalanmagan taqdirda jiddiy muammoga sabab bo'ladi.

Bir martali parollarga asoslangan autentifikatsiya. Statik yoki ko'p martali parollarga asoslangan autentifikatsiya usullari yetarlicha xavfsizlikni ta'minlay olmaydi. Bunda, mavjud muammolarni bartaraf etish uchun bir martali parollardan (One time password, OTP) foydalanish yuqori samara beradi [9].

Bir martali parolga asoslangan autentifikatsiyalash sxemasi resursga bo'lgan har bir murojaat uchun yangi paroldan foydalanishga asoslanadi. Hosil qilingan paroldan tizimda faqat bir marta foydalaniladi. Ushbu parol qo'lga kiritilgan taqdirda ham foydasiz hisoblanadi. Umumiy holda bir martali parollarga asoslangan autentifikatsiya masofadagi foydalanuvchini haqiqiyligini tekshirishda foydalaniladi.

Bir martali parollar *token* deb yuritiluvchi qurilmada yoki dasturiy vositalar orqali generatsiya qilinadi. Tokenlar odatda ikkinchi faktor sifatida qaralgan PIN kodlar bilan himoyalangan bo'lib, ularda bir martali parolni generatsiyalash tugmasi mavjud bo'ladi. Generatsiya qilingan bir martali parol token monitorida namoyon bo'ladi.

Bir martali parollardan foydalanishning qator, sanoqni sinxronizatsiyalashga, vaqtni sinxronizatsiyalashga, xavfsiz kanalni o'z ichiga olgan yoki taqsimlangan parollar ro'yxatidan foydalanuvchi usullari mavjud.

Keng tarqalgan bir martali parollarga asoslangan autentifikatsiya protokollaridan biri – S/Key Internet protokolidir [73]. Ushbu protokol

ko‘plab tizimlarda amalga oshirilgan, xususan, Cisco tashkiloti tomonidan ishlab chiqilgan TACACS+ tizimida foydalanilgan. Bir martali parollarga asoslangan autentifikatsiya protokollarining tahlili bilan 1.3-bo‘limda batafsil tanishib o‘tiladi.

Bir martali parollarga asoslangan autentifikatsiya usullari biror narsani bilishga asoslangan autentifikatsiya guruhiga emas, balki, foydalanuvchiga mavjud biror narsa asosida autentifikatsiya guruhiga taalluqli. Ya’ni, tokendan foydalanganda parol generatsiya qilingunga qadar u monitorga va foydalanuvchiga ko‘rinmaydi.

Shaxsiy identifikatorni tasdiqlovchi (Personal Identity Verification, PIV) kartalar biror narsaga egalik qilishga asoslangan yana bir keng tarqalgan autentifikatsiya usuli bo‘lib, egasining haqiqiyligini tasdiqlash uchun xizmat qiladi [10].

Biometrik autentifikatsiya. Insonning biometrik parametrlari hisoblash tizimlarida ruxsatlarni nazoratlash va identifikatsiyani boshqarish uchun foydalanilib, ular quyidagi afzalliklar bilan xarakterlanadi [1]:

– biometrik xususiyatlarga (unikalligi sababli) asoslangan autentifikatsiyaning yuqori aniqligi;

– biometrik xususiyatlarning egasidan ajratib bo‘lmasligi;

– biometrik xususiyatlarni qalbakilashtirishning murakkabligi.

Hozirda quyidagi biometrik xususiyatlarga asoslangan autentifikatsiya va identifikatsiya usullari keng qo‘llanilmoqda:

– barmoq iziga asoslangan;

– qo‘lning geometrik shakliga asoslangan;

– yuz shakli va o‘lchamiga asoslangan;

– ovozga asoslangan;

– ko‘z qorachig‘i va to‘r pardasi xususiyatiga asoslangan.

Biometrik xususiyatlarga asoslangan autentifikatsiya usullarini amalga oshirishda xususiyatlarni qabul qiluvchi turli vositalar, maxsus skanerlar, kamera, diktafon va boshqalar talab etiladi. Iste’molchi nuqtayi nazaridan biometrik autentifikatsiya tizimlarining samaradorligi quyidagi ikki parametr (xatolik ko‘rsatkichi) bilan xarakterlanadi:

– yolg‘ondan rad etish xatoligi (False rejection rate, FRR);

– yolg‘ondan tasdiqlash xatoligi (False acceptance rata, FAR).

Yolg‘ondan rad etish xatoligi autentifikatsiya tizimining haqiqiy foydalanuvchini rad etish holatini xarakterlasa, yolg‘ondan tasdiqlash xatoligi esa haqiqiy bo‘lmagan foydalanuvchini tizim tomonidan haqiqiy deb topish darajasini xarakterlaydi. Mazkur sohada olib borilayotgan

ilmiy tadqiqotlarning asosiy maqsadi ularning qiymatini nolga olib kelish bo'lsada, amalda ularning ko'rsatkichlari turli biometrik xususiyatlar uchun turlichadir.

Yuqorida tahlil qilingan barcha autentifikatsiya usullari turlicha xususiyatga ega bo'lib, ularning barchasi bir xil sharoitda foydalanish uchun mos emas. Bundan tashqari, bir turdagi autentifikatsiyani turlicha amalga oshirilishida turli natijalarga erishish mumkin. Boshqacha aytganda, tahlil qilingan barcha autentifikatsiya usullarining o'ziga xos bo'lgan afzallik va kamchiliklari mavjud. Quyidagi 1.1-jadvalda bir qancha autentifikatsiya usullari keltirilgan omillar bo'yicha tahlil keltirilgan.

1.1-jadval

Mavjud autentifikatsiya usullarining qiyosiy tahlili [11]

Usul	Ommaviylik	Qurilma narxi	Infratuzilma narxi	Foydalanishning osonligi	Umumiy xavfsizlik	Afzallik	Kamchilik
Sodda parolga asoslangan	3	-	1	3	1	Kam xarajat, portativ, foydalanuvchiga qulay	Past xavfsizlik darajasi, esda saqlash muammosi
Murakkab parolga asoslangan	2	-	1	2	2	Sodda parollarga qaraganda xavfsizroq	Esda saqlash murakkab
Sinxronlashgan OTP tokenga asoslangan	1	3	3	3	3	Yaxshi xavfsizlik, foydalanishga qulay	Qurilma va infratuzilma narxi yuqori, birga olib yurish zaruriyati
Barmoq iziga asoslangan	1	2	3	3	3	Umumiy holda yaxshi xavfsizlik darajasiga ega	Yuqori narx, amalga oshirish xatoliklari, almashtirishning imkonsizligi

Har bir omil bo'yicha 3 ta darajada (*yuqori – 3, o'rta – 2, past – 1*) baho berilgan:

– *ommaviylik* – autentifikatsiyani qolganlariga qaraganda amalda keng foydalanish darajasiga egaligini ifodalaydi;

– *qurilma narxi* – autentifikatsiyani amalga oshirishda talab etiladigan vositalarning narxini ifodalaydi;

– *infratuzilma narxi* – axborot tizimlari arxitekturasida autentifikatsiya usulini o'rnatish narxini ifodalaydi;

– *foydalanishning osonligi* – autentifikatsiya usulining umumiy foydalanuvchanligini ifodalaydi;

– *umumiy xavfsizlik* – autentifikatsiya usulini turli major hollarda xavfsizlikni ta'minlash bahosini ifodalaydi.

Autentifikatsiya usullari orasida sodda parollarga asoslangan usulning ommaviyligi va foydalanish uchun osonlik darajasi yuqori hisoblansada, juda past xavfsizlik darajasiga ega bo'lib, esda saqlash zaruriyatini talab etadi. Tokenga asoslangan autentifikatsiya usullari esa yuqori xavfsizlik darajasini ta'minlasada, infratuzilma va qurilma narxining yuqoriligi sababli ommaviylik darajasi past hamda har doim birga olib yurishni talab etadi. Biometrik parametrlarga asoslangan autentifikatsiyalash usullari esda saqlash va olib yurish kabi zaruriyatlarni talab etmasada, Web tizimlarda foydalanishning noqulayligi, ommaviylik darajasining pastligi va yuqori narxga ega.

Autentifikatsiya usullari orasida parol va tokenlarga asosan ko'rinishi masofadan turib web resurslardan foydalanishga qulay bo'lib, qolganlariga qaraganda keng ommalashgan. Shuningdek, xavfsizlikni yuqori darajada talab etgan tizimlar tokenlarga asoslangan autentifikatsiya usulidan keng foydalanadi. Bu tizimlar yuqori narxga ega bo'lsada, barqaror ishlash (biometrikka qaraganda) va yuqori xavfsizlikni ta'minlaydi.

1.2. Parolga asoslangan autentifikatsiya usullariga qaratilgan tahdidlar va himoya usullari

Autentifikatsiyada tizimdan foydalanish huquqining to'g'riligini tekshiruvchi muhim jarayon hisoblangani sababli, ular foydalanuvchilarning shaxsiy ma'lumotlarini qo'lga kiritishga qaratilgan hujumlardan himoyalangan bo'lishi zarur. Umumiy holda hujumchining maqsadi foydalanuvchi ma'lumotlarini ruxsatsiz qo'lga kiritish hisoblanib, bunda u turli usullardan foydalanadi. Biror narsani bilishga

asoslangan autentifikatsiya usullariga qaratilgan hujumlarni quyidagi ikki guruhga ajratish mumkin [2, 12]:

- qayd yozuvini boshqarishga qaratilgan hujumlar;
- parolni topishga qaratilgan hujumlar.

Qayd yozuvini boshqarishga qaratilgan hujumlar. Ushbu guruhga tegishli hujumlar bevosita parolni qo'lga kiritishni maqsad qilishi bilan xarakterlanadi. Quyidagi hujumlar mazkur guruhga tegishli:

Zararli dasturlarga asoslangan. Zararli dasturlar ixtiyoriy ruxsat etilmagan dastur bo'lib, foydalanuvchining ruxsatisiz tizimga o'rnatiladi. Ularning asosiy maqsadi foydalanuvchiga tegishli konfidensial axborotni to'plash bo'lib, ularga klaviatura orqali kiritilgan axborotni to'plovchilarni (keylogger), sichqoncha harakatini qayd etuvchilarni, ekran va xotiraning joriy holatidan nusxa oluvchilarni misol keltirish mumkin.

Mazkur hujum turiga misol tariqasida 2018-yilning 25-sentyabrida Hindistonda Zoho kompaniyasining 3 millionga yaqin foydaluvchilariga tegishli bo'lgan qayd yozuvlarini o'g'irlanishini keltirish mumkin [74]. Buning uchun xakerlar tomonidan *keylogger* turidagi dasturiy vositadan foydalanilgan. Bundan tashqari, 2010-yilda Buyuk Britaniyada bank mijozlarining taxminan 31 mln. $ mablag'ini o'g'irlanishiga Zeus nomli *keylogger* dasturi sababchi bo'lgan [75].

"Spufing"(spoofing) hujumlariga asoslangan. Spufing hujumi deb biror imtiyozni noqonuniy qo'lga kiritish uchun hujumchi o'zini boshqa kabi muvaffaqiyatli maskirovka qilish holatiga aytiladi. Yuqorida keltirilgan hujumlarni amalga oshishi uchun "qurbon" kompyuteri zararli dastur bilan zararlanishi talab etilsa, mazkur hujumda esa bu talab etilmaydi. Fishing (Phishing) va farming (Pharming) hujumlari mazkur hujumlarga yaqqol misol bo'ladi.

Fishing hujumlarida hujumchi o'z (qalbaki) serveriga murojaatni amalga oshirish imkoniyatini beruvchi havola (URL)dan iborat bo'lgan spam xabarni "qurbon"ga yuboradi. Ushbu URLda "qurbon"ning konfidensial axborotini (masalan, paroli) talab etuvchi biror xizmat taqdim etiladi. Fishing hujumi ijtimoiy muhandislik (sotsial injineriya) va "o'rtada turgan odam" hujumlarining mujassamlashgan ko'rinishi bo'lib, turli shaxsiy ma'lumotlarni olishni maqsad qiladi. Ushbu hujumni amalga oshirilishida foydalanilgan kriptografik autentifikatsiya usulining (masalan, TLS/SSL protokoli) zaifligi emas, balki umumiy holda tashkil etilgan autentifikatsiya yechimining zaifligi sabab bo'ladi [2].

Pochta ma'lumotlari (login va parol)ni o'g'irlashga qaratilgan, 2018-yildagi eng yirik fishing hujumlari sifatida "MailChimp", "Social Network", "Account Verification", "ICO", "Tax W2" va "Gmail Scam"larni misol keltirish mumkin. Xususan, "Bee Token" kompaniyasiga ta'sir qilgan "ICO" fishing hujumi natijasida 85000 $ zarar yetkazilgan [76].

Farming hujumlari "qurbon"ni hujumchini web sahifasiga murojaatini amalga oshirishda turli xiylalardan foydalanadi. Bunda, u zararlangan DNS (Domain Name System) serverlar xizmatidan foydalanadi. Buning uchun, foydalanuvchi platformasi dastlab zararlantiriladi (masalan, biror JavaScript va JavaApplet ilovalarini bilmasdan yuklash orqali) va shundan so'ng amalga oshirilgan barcha murojaatlar zararlangan DNSlar orqali hujumchi saytiga yo'naltiriladi.

Farming hujumi katta zarar keltirishi mumkin bo'lgan hujum turlariga kiradi. Xususan, 2017-yilning fevral oyida AQSh, Yevropa va Osiyo-tinch okeani hududlarida 50 dan ortiq moliyaviy tashkilotlarda farming hujumi kuzatilgan. Ushbu tashkilotlar qatoriga Barclays banki, Shotlandiya banki, PayPal, eBay, Discover Card va American Express tashkilotlari ham mavjud [77].

"O'rtaga turgan odam" hujumi. Ushbu hujum turi tarmoq orqali himoyalanmagan ko'rinishda parolni uzatish natijasida amalga oshiriladi. Mazkur holda hujumchi mijoz va server orasida uzatiluvchi axborotni tutib oladi hamda kontentni tahlil qilish orqali muhim axborot, login, parol yoki pochta xabarlarini qo'lga kiritishi mumkin bo'ladi. Ushbu hujum xavfsiz bo'lmagan autentifikatsiya protokollaridan, PAP yoki Telnet, foydalanilganda katta samara beradi. Biroq, mazkur hujumni hattoki xavfsiz protokollar (masalan, SSL/TLS)da ham amalga oshirish imkoniyati mavjud.

Olib borilgan izlanishlar natijasida SSL/TLS protokollaridan foydalanishning o'zi "O'rtada turgan odam" hujumidan himoyalanish uchun yetarli emasligi aniqlandi. Boshqacha aytganda, aksariyat web severlarda fishing va "O'rtada turgan odam" hujumini bartaraf etish imkonini beruvchi HTTP Strict Transport Security (HSTS) xususiyatdan foydalanilmagan [95].

Takrorlash hujumi. Mazkur hujumga asosan hujumchi qonuniy foydalanuvchini autentifikatsiya jarayonini kuzatadi va seans ma'lumotlarini ko'chirib oladi. Ma'lum vaqtdan so'ng esa ushbu ma'lumotlarni qayta yuborish orqali haqiqiy foydalanuvchi nomidan

muvaffaqiyatli autentifikatsiyadan o'tadi. Mazkur hujum usuli hattoki foydalanilgan parollar xeshlangan ko'rinishda bo'lganda ham samara beradi.

"O'rtada turgan brauzer"(Man-In-The-Browser attack, MITB) hujumi. Ushbu hujum Internet hujumlari sirasiga kirib, "O'rtada turgan odam" hujumiga o'xshash hisoblansada, tahdidni "qurbon" platformasida amalga oshirilishi bilan farqlanadi. MITB hujumida zararli fayl web brauzerni maqsad qiladi va o'zini foydalanuvchi hamda web brauzer o'rtasida joylashtiradi. MITB hujumining asosiy maqsadi foydalanuvchi va xizmat serveri provayderiga bildirmasdan onlayn bank tizimlarining moliyaviy tranzaksiyalarini hosil qilish yoki modifikatsiya qilish sanaladi. Bunda, hujum autentifikatsiya sessiyasi amalga oshirilib bo'lgunga qadar amalga oshirilmaydi. Bu hujum hattoki xavfsiz protokollar (masalan, SSL/TLS) amalga oshirilgan taqdirda ham amalga oshirilishi mumkin. Ushbu hujumni amalga oshirilishiga olib keluvchi quyidagi 5 ta sabab Filip Guxring tomonidan keltirilgan [13]:

1. *Browser Helper Objects* – Windows operatsion tizimida Internet Explorer tomonidan dinamik kutubxonalarning (Dynamically Loaded Libraries, DLL) yuklanishi sababli tahdidchi to'liq nazoratni ta'minlashi mumkin.

2. *Extensions* – ushbu sabab ham birinchisi kabi bo'lib, u boshqa brauzerlar, masalan, FireFox, uchun o'rinli bo'lishi bilan farqlanadi.

3. *Foydalanuvchi skriptlari (UserScripts)* – brauzerda yuklanuvchi foydalanuvchi tomonidan yuklangan skriptlar.

4. *API-Hooking* – bu hujum brauzer va brauzer DLL fayllari (Extensions va operatsion tizim DLL fayllari) orasida amalga oshiriladi. Masalan, bu hujum orqali SSL protokoli jarayonida brauzer va SSL orqa fondi mexanizmlari orasidagi aloqa modifikatsiya qilinishi mumkin.

5. *Virtualizatsiya* – operatsion tizim virtual muhitda yuklanganda barcha xavfsizlik usullarini osonlik bilan chetlab o'tish mumkin bo'ladi.

Turli web brauzerlar uchun MITB hujumini amalga oshirishni maqsad qilgan turli zararli dasturlar mavjud. Xususan, Windows OTlarda Internet Explorer va Firefox brauzerlari uchun yaratilgan Agent.DBJP, Bugat, Carberp, Goz [78]lar yordamida MITB hujumi amalga oshirish mumkin.

"Xizmat ko'rsatishdan voz kechishga undash"(Denial of Service attack, DOS) hujumi. Ushbu hujum natijasida qayd yozuvi ma'lumotini qo'lga kiritishga harakat qilish o'rniga tizimni foydalanuvchanlik xususiyatini yo'qqa chiqarishni maqsad qilinadi. Odatda parollarga

asoslangan autentifikatsiya usullarida qo'pol kuch hujumi yoki lug'atga asoslangan hujumlar bo'lishidan uni oldini olish uchun muvaffaqiyatsiz urinishlar soni bo'yicha tizimni vaqtinchalik bloklab qo'yish amalga oshiriladi. Boshqacha aytganda, hujumchi foydalanuvchi nomidan talab etilgan muvaffaqiyatsiz urinishlarni amalga oshiradi va qayd yozuvini blokka tushiradi. Bu holda haqiqiy foydalanuvchi parolni to'g'ri kiritgan taqdirda ham tizimdan foydalanish imkoniyatiga ega bo'lmaydi.

Parolni topishga qaratilgan hujumlar. Yuqorida keltirilgan hujum turlaridan farqli ravishda parolga qaratilgan hujumlar faqat foydalanuvchiga tegishli parolni topishni maqsad qiladi. Parolga qaratilgan hujumlar asosan parollarni faraz qilish, uzatish yoki saqlash davomida ularni tiklashga asoslanadi. Bular ichida eng ommalashgan usul bu – parolni taxminiy kiritib ko'rish. Quyida mazkur turga tegishli qator hujum usullarining tavsifi keltirilgan.

Zaif kriptografik tizimdan foydalanish. Parolni saqlashda yoki tarmoq orqali uzatishda zaif kriptografik protokoldan foydalanilganda hujumchi tomonidan uni aniqlash imkoniyati yuzaga keladi. Masalan, Telnet kabi protokollar foydalanuvchi parolini ochiq holda uzatishi sababli, jiddiy muammolarga sabab bo'ladi. Bundan tashqari, parollarni xesh qiymatlarini saqlashda kriptobardoshli funksiyalardan foydalanmaslik natijasida ham parolni topish ehtimoli ortadi.

Faraz qilish asosida hujum. Odatda foydalanuvchilar esda saqlash muammosidan qochish uchun o'zlariga tegishli shaxsiy axborotdan parol sifatida foydalanishga harakat qilishadi. Bunga ularning tug'ulgan kunlari, telefon raqamlari, turmush o'rtog'ining ismi va boshqa ma'lumotlarni misol keltirish mumkin. Bu hujumni amalga oshirishda hujumchi "qurbon"ga tegishli shaxsiy ma'lumotlarni qo'lga kiritish uchun sotsial injineriyadan keng foydalanadi [90].

Lug'atga asoslangan hujum. Parolni topishning mazkur usuli ehtimollikga asoslanadi. Bunda hujumchi eng keng tarqalgan parollar lug'atidan ko'ra ehtimoli eng yuqori bo'lganlaridan birin-ketin foydalanadi. Ko'plab insonlar aynan bir turdagi parollardan foydalanishlari mumkinligi sababli, aksariyat hollarda ushbu hujum muvaffaqiyatga erishadi.

Butun dunyo bo'yicha xakerlar guruhi yoki analitika bilan shug'ullanuvchi tashkilotlar tomonidan har yilgi eng ko'p foydalanilgan parollar ro'yxati e'lon qilinib boriladi. 2020-yil uchun TechRepublic Premium tashkiloti tomonidan taqdim etilgan ma'lumotga asosan eng

ko‘p foydalanilgan parollar beshligini: “123456”, “123456789”, “picture1”, “password” va “12345678”lar tashkil etgan [79].

Qo‘pol kuch hujumi. Bu hujumda parolning bo‘lishi mumkin bo‘lgan barcha kombinatsiyalari ko‘rib chiqiladi. Bunda asosiy faktor sifatida parolning uzunligi va uni qanday belgilardan iborat ekanligi qaraladi. Mumkin bo‘lgan belgilar sonining va parol uzunligini ortishi ushbu hujumni amalga oshirilish ehtimolini yo‘qqa chiqaradi. Shuning uchun, ushbu usuldan odatda qisqa uzunlikdagi parollarni topishda keng foydalaniladi. Bundan tashqari, barcha parollar kombinatsiyasini hisoblash vaqtini kamaytirishda parallel hisoblash texnologiyalaridan keng foydalaniladi. Qo‘pol kuch hujumining offlayn va onlayn turlari farqlanib, agar hujumchi himoyalangan resursga bog‘lanmasdan hujumni amalga oshirsa offlayn hujum, aks holda onlayn hujum deb ataladi. Onlayn turidagi qo‘pol kuch hujumini oldini olish oson bo‘lib, muvaffaqiyatsiz urinishlar soniga ko‘ra qayd yozuvi bloklanadi.

Oldindan hisoblashga asoslangan hujumlar. Mazkur hujum turiga ko‘ra lug‘atdagi har bir parol so‘z o‘zining biror xesh-funksiyadagi qiymati bilan saqlanadi hamda ikki ustundan iborat bo‘lgan: hisoblangan xesh qiymatlar va unga mos parol so‘zlar shaklidagi jadval ko‘rinishida shakllantiriladi. Agar yangi xesh qiymat kiritilsa, jadvaldagi xesh qiymatlar ustunidan aniqlanib, unga mos bo‘lgan parol topiladi.

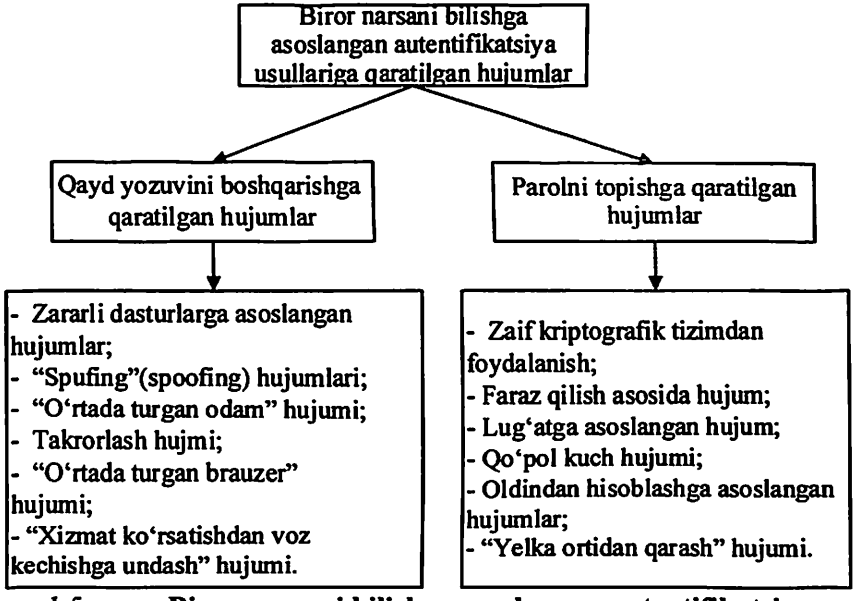
Ushbu hujumni oldini olish uchun hosil qilingan xesh qiymatlarni turlicha bo‘lishini ta’minlash zarur. Buning uchun “tuz” (salt) deb ataluvchi tasodifiy qiymatdan foydalaniladi. Bu usulga ko‘ra har bir parol uchun tasodifiy “tuz” qiymati hosil qilinadi va u parolga biriktirilgan holda xeshlanadi. Parollar bazasida esa parolning xesh qiymati va unga mos “tuz” qiymati saqlanadi.

Mazkur usul oldindan hisoblashga asoslangan hujumni amalga oshirilishini murakkablashtiradi. Boshqacha aytganda, “tuz” qiymati asosida parolni xeshlamasdan turib, uni taqqoslashning foydasi bo‘lmaydi. Buning natijasida oldindan hisoblashga asoslangan hujumning samaradorligi amaliy jihatdan yo‘qqa chiqadi.

Mazkur hujumni amalga oshirish uchun ko‘plab vositalar mavjud bo‘lib, ular orasida RainbowCrack [80], Aircrack-ng [81], John the Ripper [82]lar keng qo‘llaniladi. Xususan, turli uzunlikdagi va belgilardan iborat bo‘lgan parollarning LM, NTLM, MD5 va SHA1 xesh funksiyalardagi qiymatlari RainbowCrack vositasida oldindan generatsiya qilingan [71].

"Yelka ortidan qarash" hujumi. Mazkur hujum usulidan foydalanuvchi qayd yozuvi ma'lumotlariga bevosita qarash yoki bilvosita kamera yozuvlaridan foydalangan holda bilib olishni maqsad qiladi. Mazkur hujum usuli jarayon va foydalanilgan vositaga bog'liq holda samaradorlik darajasini qayd etadi.

Umumiy holatda biror narsani bilishga asoslangan autentifikatsiya usullariga qaratilgan hujumlar tasnifini 1.5-rasmdagi kabi ifodalash mumkin.



1.5-rasm. **Biror narsani bilishga asoslangan autentifikatsiya usullariga qaratilgan hujumlar tasnifi**

Biror narsani bilishga asoslangan autentifikatsiya usullariga qaratilgan hujumlarni oldini olishning yagona usuli mavjud emas. Hujumchining malakasi, foydalanilgan vosita va muhitga mos ravishda yetkaziluvchi zarar miqdori va xavf darajasi ham turlicha bo'ladi. 2-ilovada biror narsani bilishga asoslangan autentifikatsiya usullariga qaratilgan hujumlarni oldini olishga xizmat qiluvchi usul va choralar keltirilgan.

Biror narsani bilishga asoslangan autentifikatsiya usullariga qaratilgan hujumlarga qarshi taklif etilgan usullar to'laqonli ravishda himoyani ta'minlamaydi. Biroq, keltirilgan tavsiyalardan OTPga

asoslangan autentifikatsiya usullari mavjud hujumlarga qarshi muhim vosita ekanligini ko'rish mumkin. Boshqacha aytganda, parolga asoslangan autentifikatsiya usullarini amalda keng qo'llanilish darajasini saqlab qolishda ulardagi xavfsizlik muammolarini bir martali parollardan ikkinchi faktor sifatida foydalanish muhim hisoblanadi.

1.3. Bir martali parollarga asoslangan autentifikatsiya protokollarining tahlili

OTP tarmoq yoki xizmatdan foydalanishda bir martali parollarga asoslangan mexanizm bo'lib, u qo'lga kiritilgan login/paroldan qayta foydalana olmaslikni kafolatlaydi. Mazkur holatda, foydalanuvchining logini o'zgarmasdan, har bir urinishda uning paroli o'zgarib turadi.

Bir martali parollar odatda bir tomonlama funksiyalar deb ataluvchi – *xesh funksiyalar* asosida hosil qilinadi yoki xesh funksiyalarga asoslangan autentifikatsiya protokoli ishlab chiqiladi. Shunga ko'ra, xesh funksiyaga asoslangan bir martali parol yordamida autentifikatsiya sxemalarini ikki guruhga ajratish mumkin [14, 15]:

I. ***Savol-javobga asoslangan bir martali parol yordamida autentifikatsiya protokollari.*** Ushbu usulga ko'ra keyingi OTP o'zidan oldingisi asosida hosil qilinadi. Bu holda bajarilishi kerak bo'lgan ketma-ketlik oldindan aniqlangan bo'ladi. Ushbu guruhga tegishli OTPni generatsiyalashda Lamport sxemasi alohida ahamiyatga ega [16]. Lamport sxemasiga asoslangan autentifikatsiya usullari sodda hisoblansada, har bir autentifikatsiya sessiyasida foydalanuvchidan ko'p marta xeshlash amalini bajarish talab etiladi. Bu esa o'z navbatida yuqori vaqt sarfi va qurilma imkoniyatini talab etadi [17]. M.Sandirigama va boshqalar tomonidan ishlab chiqilgan SAS (Simple and Secure) autentifikatsiya protokoli saqlash, yuqori amal bajarish va tarmoq orqali uzatish zaruriyatini talab etmasligi, "O'rtada turgan odam" hujumini kamaytirish imkoniyatini bersada [18], takrorlash va DOS hujumlariga bardoshsizdir [19]. Mazkur muammolar OSPA (Optimal Strong-Password Authentication) [19] va SAS protokolining qayta ko'rib chiqilgan versiyasida [20] bartaraf etilgan bo'lsada, kichik imkoniyatga ega qurilmalarda amalga oshirish murakkab hisoblanadi (SAS protokolining qayta ko'rib chiqilgan versiyasida (SAS-R) autentifikatsiyaning har bir sessiyasida 5 marta xeshlash amalini bajarish talab etiladi). Bir tomonlama funksiyalardan foydalanish sonini kamaytirishga harakat qilingan yoki ikki tomonlama autentifikatsiyani

ta'minlovchi SAS-2 protokoli va uning savol-javob mexanizmiga asoslangan ko'rinishlari [21] (maxfiy kattalikni, masalan, tasodifiy qiymatni) ishlab chiqilgan bo'lsada, ular ham bir qancha ma'lumotni saqlashni va ma'lum hisoblashlarni talab etadi. Bundan tashqari, Lamport, SAS sxemalariga o'xshash bir qator autentifikatsiya usullarida [22, 23] ham ushbu muammolarni ko'rish mumkin.

Mavjud savol-javobga asoslangan bir martali parol yordamida autentifikatsiya protokollarini tahlillashda zarur bo'lgan omillarni aniqlash muhim hisoblanadi. Protokollarga qo'yilgan talablardan kelib chiqib, bir martali parollarga asoslangan autentifikatsiya protokollarini xavfsizlik va samaradorlik xususiyatlari bo'yicha tahlil qilish muhimdir.

Bir martali parollarga asoslangan autentifikatsiya protokollarini ishlab chiqishda qator izlanuvchilar tomonidan xavfsizlik talablarini aniqlashga harakat qilingan [24-26]. Xavfsizlik talablari bir martali parollarga asoslangan autentifikatsiya usulini turli hujumlarga qarshi tura olishini ta'minlasa, qo'yilgan maqsadlar esa autentifikatsiya usulini amalga oshirishdagi qo'shimcha erishilishi zarur bo'lgan xususiyatlarni tavsiflaydi.

Ishlab chiqiladigan bir martali parollarga asoslangan autentifikatsiya protokolini tekshirish uchun quyidagi xavfsizlik talablari tanlab olindi [27, 28]:

1. "Xizmat ko'rsatishdan vos kechishga undash (Denial of Service)" hujumiga bardoshli bo'lish. Ushbu hujumda tahdidchi qonuniy foydalanuvchiga tegishli bo'lgan ma'lumotlarni almashtirish orqali uning keyingi urinishini muvaffaqiyatsiz bo'lishiga sababchi bo'ladi. Shundan so'ng, qonuniy foydalanuvchi ortiq autentifikatsiyadan o'ta olmaydi.

2. "Qalbakilashtirish" hujumiga bardoshli bo'lish. Hujumchi qonuniy foydalanuvchini qalbakilashtirish uchun eshitiluvchi aloqani modifikatsiyalaydi va bu orqali tizimga kiradi.

3. "O'rtada turgan odam" hujumiga bardoshli bo'lish. Mazkur holda hujumchi ikkita ketma-ket seansda aloqa liniyasiga ulanish orqali tizimga kirishga muvaffaq bo'ladi. Mazkur holda aloqa liniyasini eshitishdan, modifikatsiyalashdan va zararlashdan himoyalash muhim hisoblanadi.

4. Takrorlash hujumiga bardoshli bo'lish. Ushbu hujumda hujumchi protokolni tinglashdan olingan ma'lumotdan qayta foydalanish orqali boshqa qonuniy foydalanuvchilarni aldaydi yoki obro'sizlantiradi.

5. "Faraz qilish bo'yicha" hujumga bardoshli bo'lish. Autentifikatsiyada foydalanilgan parolning entropiyasi yuqori bo'lishi talab etiladi. Mazkur hujumda onlayn yoki offlayn tarzda foydalanuvchi parolini aniqlashga xizmat qilinadi.

6. "O'g'irlangan verifikator" hujumiga bardoshli bo'lish. Hujumchi serverdan parol-verifikatorlarni (masalan, xeshlangan parollar) o'g'irlash orqali haqiqiy foydalanuvchini obro'sizlantiradi.

7. Ikki tomonlama autentifikatsiyani ta'minlash. Foydalanuvchi va server bir-birini haqiqiyligini tekshirish imkoniyatiga ega bo'ladi. Ya'ni, faqat server foydalanuvchini autentifikatsiyadan o'tkazib qolmasdan, foydalanuvchi ham serverni haqiqiyligini tekshirish imkoniyatiga ega bo'ladi [86].

8. Hisoblash xarajatlarini imkoni boricha kam bo'lishi. Ishlab chiqilgan autentifikatsiya protokolini amalga oshirganda kam xarajatli hisoblash imkoniyatini talab etishi zarur.

Yuqorida keltirilgan tahdid turlari yoki xavfsizlik talablari bir martali parollarga asoslangan autentifikatsiya protokollariga qo'yilgandan eng muhimlari hisoblanadi. Bundan tashqari, ba'zi tahdid turlari mavjudki, ularni bir martali parolga asoslangan autentifikatsiya protokollarining barcha ko'rinishida qo'llashning imkoniyati mavjud emas.

Bir martali parollarga asoslangan autentifikatsiya protokollarini samaradorligini baholashda ularda foydalanilgan xeshlashlar soni, saqlanuvchi ma'lumotlarning hajmi va tarmoq orqali uzatiluvchi ma'lumotlar hajmini o'z ichiga olgan quyidagi omillar tanlandi:

– server qismida bajarilgan xeshlashlar soni va saqlanuvchi ma'lumotlar;

– mijoz qismida bajariluvchi xeshlashlar soni va saqlanuvchi ma'lumotlar;

– mijozdan serverga uzatiluvchi ma'lumotlar hajmi ($L()$ – orqali foydalanilgan xesh funksiya asosida olingan xesh qiymat yoki biror kattalikning bitdagi uzunligi belgilanadi).

Yuqorida keltirilgan xavfsizlik va samaradorlik talablarini o'z ichiga olgan omillar orqali mavjud bir martali parollarga asoslangan autentifikatsiya protokollarining tahlili mos holda 1.2 va 1.3-jadvallarda keltirilgan (bu yerda, P – past, O' – o'rta va Y – yuqori).

1.2-jadval

Mavjud bir martali parollarga asoslangan autentifikatsiya usullarining xavfsizlik omillari bo'yicha tahlili

	1-talab	2-talab	3-talab	4-talab	5-talab	6-talab	7-talab	8-talab
Lampord sxemasi [16]	+	+	+	-[30]	+	+	-	P
CINON [29]	-	+	-[19]	-	+	-	-	O'
PERM [31]	+	-[32]	-[33]	+	+	-	-	O'
SAS [18]	-[19]	+	+	-[19]	-[34]	-[35]	-	Y
SAS-1 [49]	-[47]	+	+	-	+	-[47]	-	Y
OSPA [19]	+	-[36]	-[37]	+	-	-[35]	-	O'
OSPA-1 [38]	-[39]	+	+	-[39]	+	+	-	O'
ROSI [34]	-[40]	+	+	+	+	+	+	O'
SAS-2 [41]	-[34]	-[42]	+	+	-[34]	-[34]	+	Y
CLH [45]	-[44]	+	+	+	+	+	+	O'
SPAPA [46]	+	+	-[43]	-[43]	+	-[43]	+	O'
SAS-3 [33]	+	+	#	+	+	#	+	Y
JA [25]	#	+	+	+	+	+	+	O'
Ku [47]	+	+	+	-[48]	-[49]	+	-	O'

1.3-jadval

Mavjud bir martali parollarga asoslangan autentifikatsiya usullarining samaradorlik omillari bo'yicha tahlili

Usul nomi	Serverda		Mijozda		Mijoz→ Server	
	Xeshlashlar soni (marta)	Saqlanuvchi ma'lumotlar	Xeshlashlar soni (marta)	Saqlanuvchi ma'lumotlar	Uzatilishlar soni	Uzatiluvchi ma'lumot hajmi
1	2	3	4	5	6	7
Lampord sxemasi [16]	1	n	$M-n$	n	1	$L(H)$
CINON [29]	2	N_i, N_{i+1}	5	E_i^1, E_{i+1}^1, M_n	1	$3L(H)$
PERM [31]	1	$A, h^2(P\|\|N_n), N_n$	5	–	2	$L(A)$ $+ L(Ser.req)$ $+ 2L(h)$ $+ L(N_n)$
SAS [18]	2	ID, V_i	5	N_i	1	$L(ID)$ $+ 2L(H)$
SAS-1 [36]	4	$A, h^2(P \oplus n), n$	9	–	2	$L(A)$ $+ L(Ser.Req)$ $+ 3L(H)$
OSPA [19]	4	$ID, h^2(P \oplus N)$	5	Token (K, N)	1	$L(ID)$ $+ 2L(H)$
OSPA-1 [38]	3 (*mutual* 6)	$ID, h^2(S\|\|N_i)$	4	Token $(R, h(S\|\|N_i))$	1	$L(ID)$ $+ 2L(H)$
ROSI [34]	1 (*mutual* 2)	ID, V_i	3 (*mutual* 4)	N_i	1	$L(ID)$ $+ 2L(H)$
SAS-2 [41]	4 (*mutual* 6)	$ID, h^2(PW_i \oplus N)$	4	Token $(N, h(x\|\|ID))$	1 (*mutual* 2)	$L(ID)$ $+ 4L(H)$

1	2	3	4	5	6	7
CLH [45]	5	$A, h(A\|\|N\|\|P)$, $h^2(P \oplus N)$	7	Token (K, N)	1	$5L(H)$
SPAPA [46]	$\frac{1+n}{2}n$	B, UID, P, n	$\frac{1+n}{2}n+n-c$	n	2	$2L(IUD)$ $+2L(k)$ $+L(c)+L(Z)$
SAS-3 [33]	$m_{i+1}+2$	$ID, N_i, N_{i+1}, m_i, m_i$ $T, SV_U^{N_i}, K_{ID}^T$	$m_i+2m_{i+1}+4$	Token ($ID, N_i, N_{i+1}, m_i, m_i$ $T, SV_U^{N_i}$)	2	$L(ID)$ $+L(log.req)$ $+3L(H)$
JA [25]	4	$sv^{(N)}, T, N$	6	–	2	$L(log.req)$ $+3L(H)$
KU [47]	4	$sv^{(N)}, T, N, K_S$	6	–	1	$L(log.req)$ $+3L(H)$

Olingan tahlil natijalari asosida xeshlashga asoslangan bir martali parollar orqali autentifikatsiya protokollarida quyidagi umumiy bo'lgan xavfsizlik muammolari va kamchiliklar aniqlandi:

1. Aksariyat autentifikatsiya protokollarida ikki tomonlama autentifikatsiya mavjud emas (ya'ni, server haqiqiyligini tekshirish amalga oshirilmagan).

2. Aksariyat autentifikatsiya protokollari amalga oshirilish imkoniyati yuqori bo'lgan "xizmat ko'rsatishdan voz kechishga undash (Denial of Service)", offlayn "faraz qilish" va "o'g'irlangan verifikator" tahdidlariga bardoshsiz.

3. Mijoz va server tomonda saqlanuvchi ma'lumotlar hajmining va xeshlashlar sonining ko'pligi.

II. ***Bir tomonlama bir martali parol yordamida autentifikatsiya protokollari.*** Ushbu autentifikatsiya protokollarida har ikkala mijoz va serverda bir xil "sanoq qiymati" bo'lib, mijoz ushbu sanoq qiymati, maxfiy kalit yordamida bir martali parolni generatsiya qiladi. Shundan so'ng, sanoqning qiymati oshiriladi. Generatsiya qilingan bir martali parol foydalanuvchi tomonidan serverga kiritiladi. O'z navbatida server ham mijozniki bilan bir xil bo'lgan sanoq qiymati, maxfiy kalit yordamida bir martali parolni generatsiya qiladi va mijoz tomonidan kiritilgan parol bilan solishtiradi. Taqqoslash natijasi ijobiy bo'lganda foydalanuvchi autentifikatsiyadan o'tgan hisoblanadi. Aks holda esa autentifikatsiyadan o'ta olmaydi.

Bir yo'nalishli bir martali parollarga asoslangan autentifikatsiya protokollarida OTP foydalanuvchi tomonidan serverga quyidagicha yetkazilishi mumkin [50, 51]:

a) Mijozda ham serverda mavjud bo'lgan OTP generatori va bir xil sanoq qiymati (va maxfiy kalit) mavjud bo'ladi. Bunda, odatda qurilma (yoki dasturiy vosita) ko'rinishida bo'lgan *shaxsiy token* orqali

hosil qilinadi. Qurilmada (masalan, qurilmaning ekranida namoyon bo'lgan) yoki dasturiy vositalarda (masalan, maxsus mobil ilovalarida) hosil bo'lgan OTP mijoz tomonidan "qo'lda" kiritiladi. Bularga RSA Security tashkilotning SecurID [83] qurilmasini yoki TOTP (yoki HOTP)ga asoslangan mobil ilovalarni (Google Authenticator, Microsoft Authenticator) misol keltirish mumkin.

HOTP protokoli. HOTP (An HMAC-Based One-Time Password Algorithm [84]) algoritmi ikki tomonda taqsimlangan kalit K va bir xil sanoq holati C bo'lganda sinxron ravishda ishlaydi. Quyida ushbu algoritmning ketma-ketligi keltirilgan:

1. HMAC-SHA1 qiymatini hosil qilish: $HS = HMAC - SHA1(K, C)$. Ushbu qiymat 160 bit yoki 20 ($HS[0], \dots, HS[19]$) bayt axborot. $HMAC - SHA1()$ – funksiyasi $SHA1$ xesh funksiyasi asosida $HMAC$ (Hashed Message authentication code)ni anglatadi.

2. HMAC-SHA1 qiymat asosida 4 baytli qatorni hosil qilish (dinamik qisqartirish (Dynamic Truncation, DT) orqali): $Sbits = DT(HS)$. Dinamik qisqartirish quyidagicha amalga oshiriladi:

I. $HS[19]$ ning eng kichik 4 bitini ajratish: $offset = HS[19]\&0xF$. Bu holda 10 lik sanoq tizimida $offset \in [0,15]$ oraliqda bo'ladi.

II. Shundan so'ng $P = HS[offset] \parallel HS[offset + 1] \parallel HS[offset + 2] \parallel HS[offset + 3]$ hisoblanadi.

III. P qiymatning oxirgi 31 biti $Sbits$ ni tashkil etadi.

3. $Sbits$ bitni 10 lik sanoq tizimidagi ko'rinish ($Snum$)ga o'tkaziladi.

4. Talab qilingan uzunlik ($digit$)dagi parol mos holda $D = Snum \bmod 10^{digit}$ tenglik orqali hosil qilinadi.

TOTP protokoli. TOTP (Time-Based One-Time Password Algorithm [85]) algoritmida HOTP dan farqli ravishda sanoq qiymati sifatida vaqtdan foydalaniladi. Vaqt zonalari bilan mavjud muammolardan qochish uchun 1970-yil 1-yanvardan boshlanuvchi sekundda ifodalangan UNIX vaqt belgisidan foydalanilgan. Ushbu vaqt qiymati sekundda ifodalangani uchun har 30 sekundda TOTP uchun kiruvchi T parametr hosil bo'ladi: ($T = \left[\frac{UNIX\ ВАҚТИ}{30}\right]$). TOTPning ishlash prinsipi yuqorida ifodalangan HOTP algoritmiga o'xshash bo'lib, sanoq o'rniga vaqt parametri T dan foydalaniladi [52]. Amalda ko'plab ilovalarda TOTP algoritmidan foydalaniladi (masalan, Google Authenticator, Microsoft Authenticator). Token sifatida xususiy holda maxsus dasturiy vosita o'rnatilgan mobayl qurilmalar olinishi mumkin.

OTPPni generatsiyalashning mazkur usullari yordamida olingan parollar har doim ham yuqori tasodifiylik darajasini qayd etmaydi. Xususan, Google Authenticator ilovasida hosil qilingan OTPlar hech qachon 0 bilan boshlanmasligi va buning natijasida kalit maydoni 6 xonali parollar uchun 10^6 dan 10^6-10^5 gacha tushishi aniqlangan [53].

Amalda vaqtni sinxronlashga va sanoqni sinxronlashga asoslangan OTP protokollari juda keng foydalaniladi. Ushbu ikki yondashuvga asoslangan autentifikatsiya protokollaridagi afzallik va kamchiliklar 1.4-jadvalda aks ettirilgan.

1.4-jadval

Vaqtni va sanoqni sinxronlashga asoslangan OTP protokollarining tahlili

Sinxron-lanuvchi kattalik	**Xavfsizlik**	**Foydalanishda**
Vaqtga	*Afzal:* OTP qiymati qisqa vaqt uchun mavjud	*Afzal:* OTPni qurilma ekranidan osonlik bilan o‘qish mumkin
	Kam: OTP qiymati kuzatuvchi tomonidan osonlik bilan olinishi mumkin	*Kam:* OTP qiymati u kiritilgunga qadar o‘zgarishi mumkin
Sanoq	*Afzal:* OTPni generatsiya qilish uchun buzg‘unchi qurilmani boshqarish talab etiladi	*Afzal:* OTP qiymati foydalanuvchi so‘roviga ko‘ra generatsiya qilinadi; ma’lum vaqtdan keyin uning qiymati o‘zgarib qolmaydi
	Kam: OTPni yangi OTP foydalangunga qadar foydalanish mumkin bo‘ladi	*Kam:* OTPni generatsiya qilish uchun foydalanuvchi “generatsiya qilish” tugmasini bosishi talab etiladi

Har ikkala yondashuvning ham o‘ziga xos afzallik va kamchiliklari mavjud bo‘lsada, xavfsizlik nuqtayi nazaridan qisqa vaqt davomida amal qilishi hamda foydalanishdagi qulaylik sababli vaqtni sinxronlashga asoslangan autentifikatsiya usullaridan foydalanish maqsadga muvofiq hisoblanadi.

b) OTP server tomonidan hosil qilinadi va uzatish tarmoqlari orqali mijozga yetkaziladi. Bularga misol tariqasida mobil qurilmalariga SMS (Short Message Service) orqali yuborish, elektron pochta orqali yuborish yoki "qog'ozga" yozib olish usullarini keltirish mumkin.

SMS orqali uzatish bank-moliya sohasida to'lovlarni amalga oshirishda keng qo'llaniluvchi va barcha turdagi mobil qurilmalar uchun joiz bo'lgan usul hisoblansada, jiddiy xavfsizlik muammosiga ega. Ya'ni, SMS xabarlarni ochiq holatda yuborilishi yoki SS7 marshrutlash protokolidagi xavfsizlik muammolari sababli [86], ularda yuborilgan OTPni osonlik bilan qo'lga kiritish mumkin.

Aksariyat ikki faktorli autentifikatsiya tizimlarida generatsiya qilingan OTPlar foydalanuvchining elektron pochtasiga yuboriladi. Xususan, Microcosm tashkiloti tomonidan ishlab chiqilgan SmartSign mahsuloti OTPlarni uzatishda foydalanuvchining elektron pochtasidan foydalanadi [87].

Ba'zi davlatlardagi onlayn bank tizimlari foydalanuvchilarga qog'ozda chop etilgan ko'rinishda oldindan generatsiya qilingan va raqamlangan OTPlarni taqdim etadi. Ba'zilarida esa, foydalanuvchilarga plastik karta ko'rinishdagi kartalar taqdim etiladi va undan OTPni aniqlash uchun "o'chirishni" talab etadi. Har ikkala holda ham mavjud OTPlar bankda mavjud tizim talabiga ko'ra tasodifiy yoki ketma-ket tarzda foydalanilishi mumkin. Germaniya, Brazilya va Avstriya kabi davlatlarda bu turdagi OTPlarni TAN (Transaction authentication numbers) deb yuritiladi. Ba'zi davlatlarda esa TANlar foydalanuvchilarni mobayl telefonlariga SMS tarzida yuboriladi [22].

c) Server tomonidan generatsiya qilingan OTPlar mijozga web brauzerga maxsus ko'rinishda ham taqdim etilishi mumkin. OTPni yetkazishning mazkur usuliga webga asoslangan usulni misol keltirish mumkin. Bu usul foydalanuvchining tasodifiy taqdim etilgan tasvirlar orasidan oldindan tanlanganlarini tanish qobilyatiga asoslangan. Foydalanuvchi dastlab ro'yxatdan o'tishda taqdim etilgan narsalarning (mushuk, mashina, gullar va h.k.) maxfiy guruhini tanlaydi. Autentifikatsiyadan o'tish davomida tizim unga tasodifiy tanlangan narsalarni (albatta oldin tanlagan narsalar ham ular orasida bo'ladi) taqdim etadi. Har bir taqdim etilganlar orqa fondida takrorlanmas raqam mavjud bo'ladi. Foydalanuvchi tomonidan taqdim etilganlar orasidan oldindan tanlanganlari belgilanadi va bu holda orqa fonddagi raqamlar tasodifiy OTPni tashkil etadi [88].

Bundan tashqari, Confident Technologies tomonidan ishlab chiqilgan webga asoslangan OTP tizimida autentifikatsiyadan o'tish vaqtida mobayl telefonga rasmlarni tanlash imkoniyatini beruvchi havolani yuboradi. Ushbu havolaga murojaat bo'lganda web brauzerda tasvirlar orasidan keraklisini tanlash va OTPni generatsiya qilish amalga oshiriladi [89].

Bir yo'nalishli bir martali parollarga asoslangan autentifikatsiya usullarida OTPlarni yetkazish usullarining qiyosiy tahlili 3-ilovada keltirilgan.

Foydalanishdagi keng qamrovligi nuqtayi nazardan SMS yuborishga asoslangan yetkazish usuli qolganlaridan ajralib tursada, undagi xavfsizlik muammolari sababli kritik holatlarda foydalanish cheklanishi mumkin. Qurilma ko'rinishidagi tokenlar yuqori xavfsizlikni ta'minlasada, uning yuqori narxi va doim birga olib yurish zaruriyati sababli noqulayliklar tug'diradi. Hozirgi kunga kelib mobil qurilmalarning keng tarqalishi va insonlarning "ajralmas sherigiga" aylanishi dasturiy ko'rinishdagi tokenlardan foydalanish imkoniyatini oshiradi. Biroq, mobil operatsion tizimlarning xavfsiz emasligi yoki "yelka ortidan qarash" kabi hujumlar natijasida qo'shimcha xavfsizlik choralarini ko'rish talab etiladi.

Pochta orqali yuborish yoki qog'ozda yozgan holda olib yurish mos holda qo'shimcha bilimni (pochta parolini) yoki doimo yonda xavfsiz saqlash zaruriyatini talab etadi. Webga asoslangan usullar esa hozirgi kunga kirib kelayotgan yangi yondashuv hisoblanib, inson xotirasida oson saqlash imkoniyatini taqdim etishga harakat qiladi.

II BOB. BIR MARTALI PAROLLARNI GENERATSIYALASHNING SAMARALI USULI VA ALGORITMI

2.1. Bir martali parollarni generatsiyalash usullarining tahlili

Biror narsani bilishga asoslangan autentifikatsiyada "kerakli bilimni" (masalan, parolni) yaratish muhim ahamiyat kasb etadi. Boshqacha aytganda, hosil qilinuvchi parollarning tasodifiylik darajasini yuqori bo'lishi ishonchli xavfsizlikni ta'minlaydi. Parollarni hosil qilishda parollar generatori deb nomlanuvchi vositalardan foydalanilib, ular umumiy holda quyidagi tashkil etuvchilardan iborat [54]:

1. Kiruvchi qiymatlar to'plami. Ushbu qiymatlar parollarni generatsiya qilish uchun zarur bo'lgan parametrlar hisoblanadi va ular parolga qo'yilgan talabga ko'ra turlicha bo'lishi mumkin. Masalan, saytga bog'liq ("site-specific") parollarni [55] generatsiya qilishda, saytga aloqador parametrlar (masalan, URL nomi) va foydalanuvchi tomonidan kiritilgan ma'lumotlardan foydalaniladi.

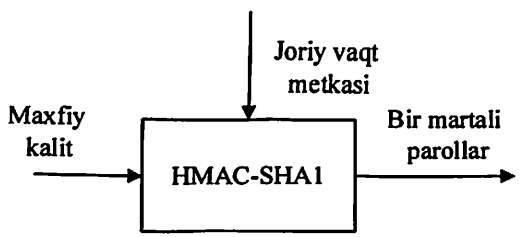
2. Parolni generatsiyalash funksiyasi. Ushbu funksiya kiruvchi qiymatlarni mujassamlashtirgan holda mos parolni generatsiya qiladi. Mazkur funksiyaga parolga qo'yiladigan barcha talablarni qanoatlantirish sharti qo'yiladi. Masalan, faqat raqamdan iborat bo'lgan parollar yoki raqam/belgilardan iborat bo'lgan parollar. Bu holda parolning uzunligi ham parol generatorlari uchun qo'yilgan muhim talablardan hisoblanadi. Bundan tashqari, parolni generatsiyalash funksiyasi uchun yuqori takrorlanish darajasiga ega bo'lish muhim hisoblanadi.

3. Parolni chop etish usuli. Mazkur tashkil etuvchi talab etilgan joyda kiritish uchun generatsiya qilingan parolni foydalanuvchiga namoyish etish yoki autentifikatsiyalash saytiga transfer qilishni amalga oshiradi.

Parollarni generatsiya qilish foydalanish mumkin bo'lgan joyga bog'liq holda turlicha amalga oshiriladi. Masalan, biror saytga yoki pochta tizimlariga kirishni amalga oshirganda tizim xususiyatlaridan foydalanish foydalanuvchi uchun esda saqlashda muhim ahamiyat kasb etadi. Shuningdek, bir martali parollarni generatsiya qilish ham o'ziga xos xususiyatlarga ega. Masalan, ularda tasodifiylik darajasi muhim hisoblanib, esda saqlash zaruriyati talab etilmaydi.

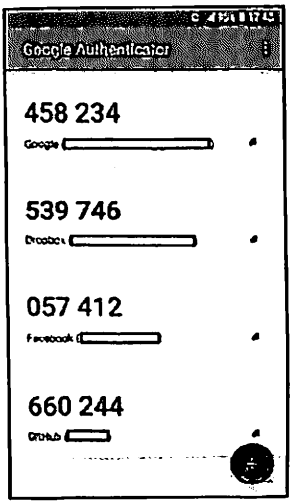
Bir martali parollarni generatsiya qilishda turli usullardan foydalanish mumkin bo'lib, ularni umumiy holda quyidagi guruhlarga ajratish mumkin [56]:

1. *Vaqt metkasidan kiruvchi parametr sifatida foydalanuvchi psevdotasodifiy sonlar generatoriga asoslangan.* Amalda keng qo'llanilayotgan OTPni generatsiyalash usullari vaqtni sinxronlashga asoslangan bo'lib, ular orasida TOTP algoritmi muhim ahamiyatga ega. TOTPga asoslangan uchinchi tomon ilovalari hozirda amalda keng qo'llanilmoqda. Ularga Google Authenticator, Microsoft Authenticator kabilarni misol keltirish mumkin. Xususan, hozirda Google Authenticator ilovasidan ko'plab tizimlarda (masalan, Gmail.com, facebook.com, GitHub, Twitter, Dropbox) ikkinchi faktor sifatida qo'llanilmoqda. Mazkur ilovalarni amalda turli tizimlarda foydalanish imkoniyatining mavjudligi TOTP algoritmida taqsimlangan parametr sifatida vaqt parametridan foydalanishga asoslanadi. 2.1-rasmda TOTP algoritmining umumiy ko'rinishi keltirilgan (TOTPning batafsil tavsifi 1.3-bo'limda keltirilgan).

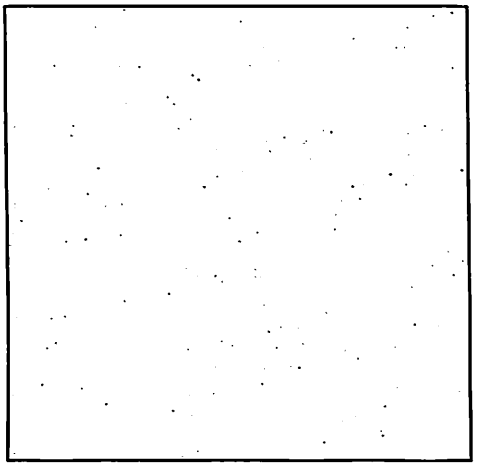


2.1-rasm. **TOTP algoritmining umumiy ko'rinishi**

Google Authenticator ilovasida 6 ta belgidan iborat bo'lgan OTPlar generatsiya qilinadi. Ushbu ilova yordamida generatsiya qilingan OTPlarni tasodifiylik darajasini tekshirish uchun mazkur ilova Gmail, Facebook, Dropbox va Github tizimlari uchun foydalanildi (2.2a - rasm). Ushbu tizimlarda vaqt parametri bir xil bo'lsada, turli kalitlardan foydalanilgani bois turli OTPlarni hosil qiladi. Ushbu ilovalar uchun generatsiya qilingan 1000 ta OTP to'plandi. Har bir OTPlar 6 ta raqamdan iborat bo'lgani sababli, uning grafik ko'rinishi 2.2b-rasmda keltirilgan.



a) Google Authenticator ilovasida generatsiya qilingan OTPlar

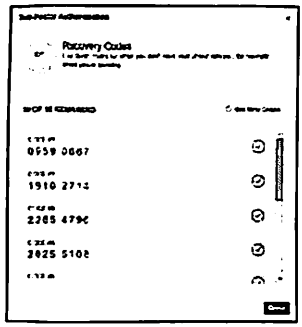


b) OTPlarning grafik tasvirlanishi

2.2-rasm. **OTP va uning grafik tasvirlanishi**

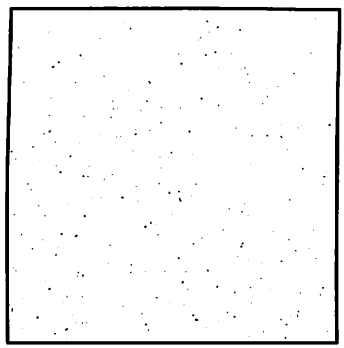
OTPlar asosida grafik ko'rinishni hosil qilish uchun, har bir OTP ikki qismga ajratildi. Masalan, agar OTP $= 458234$ ga teng bo'lsa, birinchi qism $X = 458$ ga va ikkinchi qism $Y = 234$ ga teng bo'ldi. Ikki qism asosida esa nuqta $P(X, Y)$ ikki o'lchovli koordinatalar tizimida $(999, 999)$ ifodalangan. Grafik ko'rinishda akslantirilgani kabi Google Authenticator ilovasi yordamida hosil qilingan parollarning bo'lishi mumkin bo'lgan variantlari 10^6 ga teng bo'ladi. Google Authenticator ilovasida OTPlarni generatsiya qilishda vaqt metkasidan foydalaniladi va ma'lum og'ishlarni oldini olish uchun OTP har 30 yoki 60 sekundda almashib turadi.

Faqat raqamlardan iborat bo'lgan OTPlarni tizimga kiritish oson bo'lsada, ularning bardoshliligi yuqori sanalmaydi. Shuning uchun odatda OTPlarni generatsiya qilganda (faqat raqamdan iborat bo'lgan hollarda) ularning uzunligiga e'tibor qaratiladi. Masalan, Facebook ijtimoiy tarmog'ida foydalanuvchilarga tarqatiluvchi oldindan generatsiya qilingan OTPlar (TAN) 8 ta raqamdan iborat (2.3-rasm).



2.3-rasm. **Facebook tizimidagi TAN ro'yxati**

2. Sanoqdan kiruvchi parametr sifatida foydalanuvchi psevdotasodifiy sonlar generatoriga asoslangan. Mazkur OTPni generatsiya qilish usuliga yaqqol misol bo'lib HOTP algoritmi xizmat qiladi. Ushbu algoritmning ishlashi 2.1-rasmdagi kabi bo'lib, farqli ravishda vaqt metkasining o'rniga ortib boruvchi sanoq qiymati kiritiladi. Quyidagi 2.4-rasmda HOTP asosida yagona kalitdan hosil qilingan dastlabki 1000 ta OTPlarni grafik ravishda tasvirlashdan olingan natija keltirilgan.

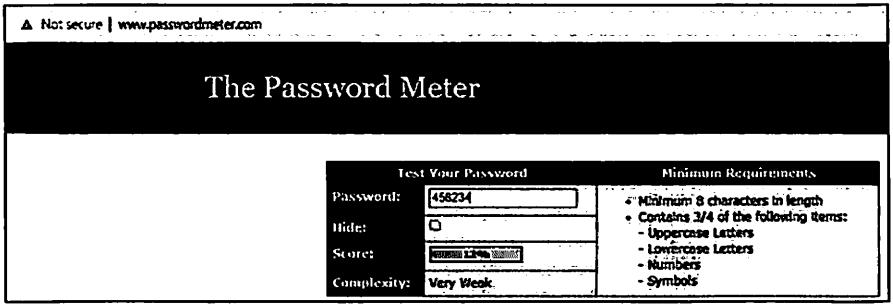


2.4-rasm. **HOTP algoritmi natijasining grafik ko'rinishi**

Bundan tashqari, mazkur guruhga tegishli OTPlarni generatsiyalash algoritmlarida "savol-javob" (challenge-response) toifasidagi bir martali parollarga asoslangan autentifikatsiya protokollaridan ham keng foydalaniladi. Bunda, sanoq o'rniga "savol" qiymati kiritiladi va uning "javob"ini solishtirish orqali autentifikatsiya jarayoni amalga oshiriladi.

3. Ma'lum belgilar to'plamidan foydalanib parollarni generatsiyalashga asoslangan. Faqat raqamlardan iborat bo'lgan OTPlar

to‘liq tanlash hujumiga bardoshsiz hisoblanadi va odatda “juda zaif” deb topilgan [90] (2.5-rasm).



2.5-rasm. **http://www.passwordmeter.com/ tizimida OTPning bahosi[91]**

Bundan tashqari, yuqori xavfsizlik darajasini talab etuvchi tizimlarda kichik lotin harflari (26 ta), katta lotin harflari (26 ta) va raqamlardan (10 ta) iborat bo‘lgan OTPlar ham qo‘llaniladi [82]. Mazkur holatda agar parol uzunligi 6 ta teng bo‘lsa, bo‘lishi mumkin bo‘lgan parollar sohasi $62^6 \approx 5{,}68 \times 10^{10} \approx 2^{35{,}7}$ ga teng bo‘ladi va “kuchli” parol deb qaraladi. Bu turdagi OTPlardan ko‘plab tizimlarda, xususan, Gmail, Dropbox kabilar o‘z TAN ro‘yxatini shakllantirishda foydalangan. Bu turdagi OTPlardan statik parollar sifatida ham foydalanish mumkin bo‘ladi.

Bundan tashqari, ma’lum belgilar to‘plamidan foydalangan holda parollarni generatsiyalashda talaffuz qilinishi va esda qolishi oson bo‘lgan, biroq, yuqori tasodifiylik darajasiga ega bo‘lgan generatorlarni ishlab chiqishga alohida e’tibor beriladi. Bularga misol qilib, Lionard va boshqalar tomonidan ishlab chiqilgan PRONOUNCE3 generatorini olish mumkin [57]. Ushbu generatorda unli (a, e, i, o, u) va undosh (b, c, ch, d, f, g, h, j, k, l, m, n, p, ph, r, s, st, v, w, x, y, z) bo‘lgan harflardan foydalanib entropiyasi 30,8 bitga teng bo‘lgan OTPlarni generatsiya qilish mumkin.

Olingan belgilar to‘plamidan tasodifiy tanlashda turli dasturlash tillarida mavjud bo‘lgan funksiyalar (Mersenne Twister algoritmiga asoslangan *rand()*, *random()* funksiyalari [58])dan keng foydalaniladi. Generatordan olingan tasodifiy baytlar ma’lum algoritm asosida belgilarga akslantiriladi [59].

4. *Tasodifiy sonlar generatoridan parollarni generatsiyalashga asoslangan.* Mazkur turdagi generatorlar odatda kam tarqalgan bo‘lib,

tasodifiy yoki psevdotasodifiy sonlar generatori mavjud bo'lgan hollarda foydalaniladi. Masalan, quyida *C* dasturlash tilida *rand()* psevdotasodifiy sonlar generatoridan foydalangan holda 8 ta uzunlikdagi parollarni generatsiya qilish algoritmi keltirilgan (*Algoritm_C*):

```
1.  #include<stdio.h>
2.  int main(void)
3.  {
4.  unsigned short int length = 8;
5.  srand((unsigned int) time(0));
6.  while(length--)
7.  {
8.  putchar(rand()%94 + 33);
9.  }
10. printf("\n");
11. return 0;
12. }
```

5. Bundan tashqari, aksariyat operatsion tizimlarda bardoshli tasodifiy sonlar generatori mavjud bo'lib (masalan, Unix oilasi uchun */dev/random* va */dev/urandom* yoki Windows uchun *CryptGenRandom*), ular yordamida yuqori tasodifiylik darajasiga ega parollarni generatsiyalash mumkin. Quyida Python 3.6 muhitida operatsion tizim generatoridan foydalangan holda bardoshli parollarni generatsiyalash algoritmi keltirilgan (*Algorithm_Ph*):

```
1.  import random
2.  import string
3.  my_random = random.SystemRandom()
4.  length = 10
5.  alphabet = string.ascii_letters +
string.digits
6.  password =
str().join(my_random.choice(alphabet) for
_ in range(length))
7.  print(password)
```

Parollar bardoshligining tahlili. Generatorlardan hosil bo'lgan parollarning bardoshligini tahlil qilish muhim ahamiyat kasb etadi. Parollar statik yoki bir martali ekanligiga qarab turli usullar yordamida tahlil qilinishi mumkin.

Generatsiya qilingan parolni bardoshligini tekshirishda uning tarkibi, uzunligi va eng keng tarqalgan parollar ro'yxatida mavjud emaslik xususiyatlariga e'tibor qaratiladi. Ushbu talablarga mos kelgan parollar "qo'pol kuch" va lug'at asosida amalga oshiriluvchi hujumlarga bardoshli hisoblanadi.

Bundan tashqari, generatsiya qilingan parollarni biror hujumga (masalan, to'liq tanlash hujumiga) bardoshliligini tekshirishda uning entropiyasini o'lchash amalga oshiriladi. Agar paroldagi belgilar mustaqil va biror qonuniyatga asoslanmagan bo'lsa, uning entropiyasi quyidagi tenglik bilan aniqlanadi [60]:

$$H = L\log_2 N = L\frac{\log N}{\log 2}$$

Bu yerda, N– bo'lishi mumkin bo'lgan belgilar soni bo'lsa, L – esa paroldagi belgilar sonini anglatadi. H– kattalik esa битlarda o'lchanadi.

2.1 – jadvalda turli sondagi belgilar to'plami uchun bir belgiga tushuvchi entropiyaning qiymati keltirilgan.

2.1-jadval

Turli belgilar to'plami uchun bir belgiga to'g'ri keluvchi entropiya

Belgilar to'plami	**To'plamdagi belgilar soni, *N***	**Bir belgiga mos entroniya, *H* (bit)**
0–9	10	3,32
0–9, *A–F*	16	4,00
a–z yoki A–Z	26	4,70
a–z yoki A–Z, 0–9	36	5,17
a–z, A–Z	52	5,70
a–z, A–Z, 0–9	62	5,95
ASCII ning barcha bosma qilinuvchi belgilari	94	6,55

Yuqorida keltirilgan belgilar to'plamidan foydalangan holda turli murakkablik darajasiga ega OTPlarni to'liq tanlash usulida topish uchun sarflangan vaqt 4-ilovada keltirilgan. Bunda http://password-checker.online-domain-tools.com/ onlayn parollarni tekshirish tizimidan foydalanildi.

Bundan tashqari, parollar bardoshligini tekshirishda ko'plab avtomatlashgan vositalardan keng foydalaniladi. Ularda parollar bardoshliligi bo'yicha turli toifalarga ajratiladi [61].

Bir martali parollar statik parollarga qaraganda qisqa vaqtda amal qilishi sababli, ularni bardoshligini tekshirishda ham qo'shimcha talablar mavjud. Quyida bir martali parollarni bardoshligini tekshirishda zarur bo'lgan omillar bilan tanishib chiqiladi.

Tasodifiylik. OTP generatorlari cheksiz uzunlikdagi parollarni generatsiya qilgani va ular qisqa vaqtda foydalanilgani bois, ular uchun tasodifiylik darajasi muhim hisoblanadi. OTP generatorlarida tasodifiylik darajasi – bir parolni qayta generatsiyalash vaqti yuqori bo'lishi bilan xarakterlanadi. Bundan tashqari, hosil bo'lgan parollar orasida chiziqli bog'liqlik bo'lmasligi talab etiladi.

Uzunlik. Ushbu kattalik generatsiya bo'ladigan OTP uchun muhim ahamiyatga ega. Aksariyat OTPga asoslangan autentifikatsiya tizimlarida 6-8 ta belgi uzunligidagi parollardan foydalaniladi. Masalan, Google Authenticator, Microsoft Authenticator, Ipak Yo'li Bank mobil ilovalarida 6 ta raqamdan iborat bo'lgan OTPlar hosil qilinsa, Facebook tizimidagi TAN ro'yxati 8 ta raqamdan iborat.

Ko'p sonli belgilar. Aksariyat OTPga asoslangan tizimlarda parollar faqat raqamlardan iborat bo'ladi (masalan, Google Authenticator, Ipak Yo'li Bank mobil ilovasida). Umumiy holda esa OTP generatorlari turli (0...9, a...z, A...Z) belgilar kombinatsiyasidan iborat bo'lgan parollarni generatsiyalash imkoniyatiga ega bo'lishi shart.

2.2-jadvalda amalda keng qo'llanilayotgan OTP generatorlaridan hosil bo'lgan parollar va ularning takrorlanish darajalari, uzunligi va tarkibidagi belgilari keltirilgan (bu yerda, *N/A* – aniq emas).

2.2-jadval

Mavjud OTP generatorlari va ularning xususiyatlari

OTP generatorlari	Tasodifiyligi	Uzunligi	Tarkibi	Guruhi	Muammo
1	2	3	4	5	6
Google Authenticator	1000 tadan 0	6	0...9	1	OTP kuzatuvchi tomonidan osonlik bilan olinishi mumkin
Facebook TAN	1000 tadan 0	8	0...9	*N/A*	OTPni yangi OTP foydalangunga qadar foydalanish mumkin
HOTP (RFC 4226)	1000 tadan 1	6	0...9	2	OTPni foydalanil-magunga qadar

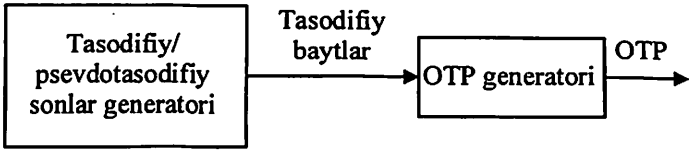
1	2	3	4	5	6
					yuborish
Ipak Yo'li Bank	-	6	0...9	*N/A*	OTPni foydalanil-magunga qadar yuborish
UPAY	-	6	0...9	*N/A*	OTPni foydalanil-magunga qadar yuborish
PAYME	-	6	0...9	*N/A*	OTPni foydalanil-magunga qadar yuborish
id.gov.uz	-	6	0..9	*N/A*	OTPni foydalanil-magunga qadar yuborish

2.2. Kompyuter resurslari asosida psevdotasodifiy sonlarni generatsiyalash usuli

Ixtiyoriy autentifikatsiya tizimi mijoz – server arxitekturasida ishlaydi va bunda server tekshiruvchi vazifasini bajaradi. Xususan, parollarga asoslangan autentifikatsiya mexanizmida ham foydalanuvchi tomonidan kiritilgan parol serverda ro'yxatdan o'tish jarayonidagisi bilan taqqoslanadi. Boshqacha aytganda, mijoz-server arxitekturasiga asoslangan autentifikatsiyada server tomoni ishonchli deb qaraladi va barcha jarayonlarni boshqaruvchisi hisoblanadi.

Bir martali parollarga asoslangan autentifikatsiya usulida ham yuqoridagi ssenariy o'zgarmas bo'lib qoladi va ular ikki holatda bo'lishi mumkin. Birinchi holatda, bir martali parol serverda hosil qilinadi va mijozga uzatiladi. Ikkinchi holatda esa mijozda alohida token deb ataluvchi qurilma mavjud bo'lib, unda generatsiya qilingan OTP serverga uzatiladi va bunda server tekshiruvchi vazifasini o'taydi.

Birinchi holatga asoslangan usulda serverda maxsus generator mavjud bo'lib, u doimiy ravishda ishonchli OTPlarni generatsiya qiladi. Generatsiya qilingan OTPlar mijozga biror uzatish usuli yordamida uzatiladi. Bunda pochta yoki SMS xabar ko'rinishida yuborish keng qo'llaniladi. Mazkur holda OTPlarni generatsiya qilishda tasodifiy/ psevdotasodifiy sonlar generatoridan (PTSG) foydalaniladi (2.6-rasm).

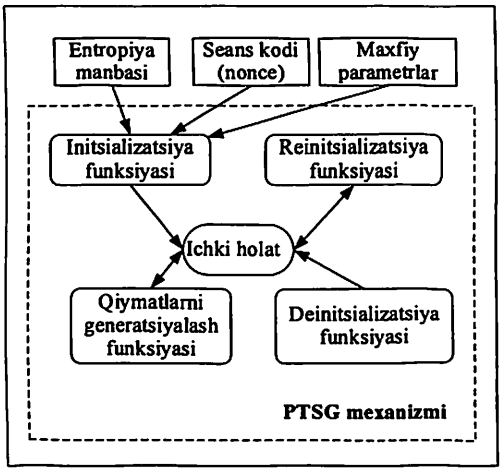


2.6-rasm. **PTSG asosida OTPlarni generatsiyalash**

Taklif etilgan yondashuvga asosan OTPni generatsiyalash usuli uchun psevdotasodifiy ketma-ketliklarni hosil qiluvchi generatorning funksional modeli 2.7-rasmda keltirilgan.

PTSG funksional modelining asosiy tashkil etuvchilari quyidagilar:

- entropiya manbasi;
- seans kodi;
- maxfiy parametrlar;
- ichki holat;
- ichki funksiyalar to'plami.



2.7-rasm. **PTSGning funksional modeli**

Entropiya manbasi ixtiyoriy PTSG uchun tasodifiylik manbasi bo'lib, turli hodisalar manbalaridan olinadi. PTSG ishga tushirish va ichki holatini yangilab turishda entropiya manbasi muhim ahamiyatga ega bo'lib, amaliyotda biror fizik hodisa manbasidan foydalanishga harakat qilinadi. Turli operatsion tizimlarda esa PTSG uchun foydalanish mumkin bo'lgan turli hodisa manbalari mavjud. Bularga magnit diskdan foydalanishdagi shovqinni, videokameradan olingan oqimni, mikrafon orqali qabul qilingan ovozni misol keltirish mumkin.

Taklif etilayotgan PTSG uchun to'planishi oson bo'lgan sichqonchaning joriy holati, operatsion tizimning yuqori aniqlikdagi vaqt metkasi, foydalanuvchi ishchi stolining joriy holati kabi parametrlar entropiya manbasi sifatida olindi.

Qurilmaga asoslanmagan PTSG uchun odatda qo'shimcha kirish qiymatlaridan foydalanish tavsiya etiladi. *Seans kodi* ham qo'shimcha kirish qiymatiga tegishli bo'lib, faqat joriy seans uchun mavjud bo'ladi va qolgan seansda takrorlanishni minimallashtirishga xizmat qiladi. Taklif etilayotgan PTSG uchun kriptografik bo'lmagan yoki operatsion tizimdagi PTSGdan olingan 128 bitli seans kodidan *(nonce)* foydalanilmoqda.

PTSGni dastlab ishga tushirishda maxfiy kattaliklardan (qiymatlardan) foydalanish talab etilib, bu talab turli muhitlar uchun unikal bo'lgan ketma-ketliklarni olish imkonini beradi. Shuning uchun, maxfiy parametr sifatida imkoni boricha sirli bo'lgan va xususiy bo'lgan qiymatlardan foydalanish mumkin. Taklif etilgan PTSG uchun maxfiy parametr sifatida qurilmaning MAC manzili, IP manzili, qattiq diskning seriya raqami kabi kattalikdan foydalanildi.

Ichki holat PTSGning xotirasi bo'lib, unda PTSG ishlashi uchun barcha parametrlar saqlanadi. Xususiy holda, ichki holat PTSGning joriy qiymatini o'zida saqlaydi.

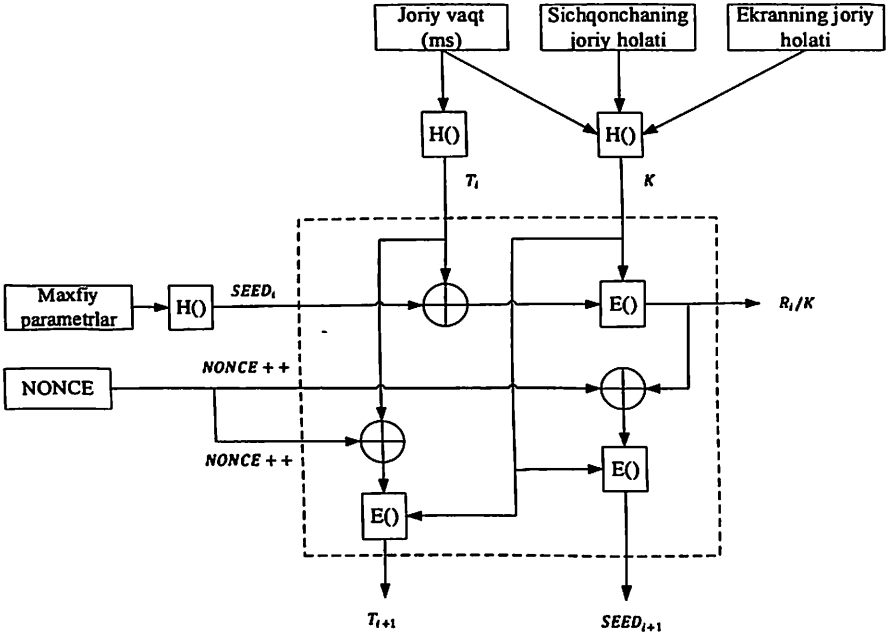
PTSG uchun ichki holatni boshqarish muhim ahamiyatga ega bo'lib, u ichki deb ataluvchi funksiyalardan foydalanadi. PTSG ichki holatini boshqarishda foydalanish zarur bo'lgan funksiyalarni shakllantirish haqida NIST SP 800-90A [62] manbasida keltirib o'tilgan. Ushbu tavsiyalarga asoslangan holda taklif etilayotgan PTSGning ichki funksiyalarining ko'rinishi 2.8-rasmda keltirilgan.

Initsializatsiya funksiyasi. Ushbu funksiya generator ishga tushishidan oldin bajariladi va kalit/ sanoq qiymati kabi parametrlarni nol holatda o'rnatadi. Shundan so'ng, entropiya manbalaridan olingan ma'lumotlar asosida foydalanilgan parametrlarning dastlabki qiymatlari shakllantiriladi.

Initsializatsiya funksiyasi 2.8-rasmdan kelib chiqib biror-bir tomonlama funksiya $H()$ yordamida joriy vaqt T_i qiymati, joriy vaqt, sichqonchaning joriy holati va ekranning joriy holati asosida kalit K qiymati, maxfiy parametrlar asosida $SEED_i$ parametr qiymati va biror kriptografik bo'lmagan sonlar generatori (yoki foydalanuvchi kiritgan) asosida $NONCE$ qiymati shakllantiriladi:

– $T_i = H(Joriy\ vaqt)$;

– $SEED_i = H(maxfiy\ parametrlar)$;
– $K = H(joriy\ vaqt \parallel sichqonchaning\ joriy\ holati \parallel ekranning\ joriy\ holati)$.



2.8-rasm. **Taklif etilgan PTSG funksional ko'rinishi**

Reinitsializatsiya funksiyasi. Mazkur funksiyada PTSG entropiya manbalaridan yangi qiymatlarni oladi va uni joriy qiymatlar bilan aralashtiradi. Natijada, keyingi qiymatlarni hosil qilish uchun PTSG yangi ichki holatga keltiriladi.

Qiymatlarni generatsiyalash funksiyasi. Ushbu funksiya talab qilingan hajmdagi psevdotasodifiy ketma-ketliklarni joriy ichki holatdan foydalangan holda hosil qiladi va keyingi so'rov uchun ichki holatni yangilaydi. 2.8-rasmga asoslangan holda qiymatlarni generatsiyalash funksiyasini bir blok uchun quyidagicha ifodalash mumkin:

– $R_i = E_K(T_i \oplus SEED_i)$;
– $SEED_{i+1} = E_K(R_i \oplus NONCE++)$;
– $T_{i+1} = E_k(T_i \oplus NONCE++)$.

Bu yerda, $E_K()$ – bardoshli simmetrik shifrlash algoritmi bo'lib, unda K – bitli kalitdan foydalaniladi va kalit K ning uzunligi $H()$ – bir tomonlama (xesh funksiya) funksiya qiymatiga teng bo'lishi talab

qilinadi. Masalan, 128 – bit xesh qiymatni hosil qiluvchi MD5 xesh funksiyasi va 128 – bit kalit o'lchamli ixtiyoriy simmetrik blokli shifrlardan foydalanish mumkin. Yoki, $H()$ – bir tomonlama xesh funksiya qiymatidan $E()$ – simmetrik blokli shifrlash uchun kalitni hosil qilishning alternativ usulidan foydalanish talab etiladi.

Har bir talab qilingan baytlar ketma-ketligini generatsiya qilingandan so'ng keyingi so'rov uchun $SEED_j$, T_j parametrlar hosil qilinadi va yangi kalit K quyidagicha hisoblanadi:

$$K = E_K(T_j \oplus SEED_j).$$

Bir tomonlama funksiya $H()$ va simmetrik blokli shifr $E()$ tanlashda kriptobardoshlilik va tezkorlik xususiyatlariga e'tibor berish talab etiladi.

Deinitsializatsiya funksiyasi. Mazkur funksiya PTSGni ichki holatini tozalash uchun foydalaniladi va uning asosiy maqsadi PTSGni tahlil qilishni oldini olishga qaratilgan.

PTSGning xavfsizlik tahlili. Kriptografik PTSGlarini baholashning qator usullari mavjud. PTSGlardan hosil bo'lgan ketma-ketliklarni baholashda "tasodifiylik" kriteriyasi muhim hisoblanib, u PTSGdan hosil bo'lgan qiymatlarni haqiqatda tasodifiy ekanligini ko'rsatadi [63].

PTSGlardan hosil bo'ladigan ketma-ketliklarni tasodifiylikka tekshirishning qator usullari mavjud bo'lib, ular orasidan statistik testlarga asoslangan usullar keng qo'llaniladi [64]. Shu sababli, 2.8-rasmda keltirilgan PTSGdan olingan natijalarning tasodifiylik darajasini tekshirish uchun NIST SPECIAL PUBLICATION 800-22 statistik testlar to'plamidan foydalanildi. Buning uchun PTSGdan 5 marta 2 million bitdan bo'lgan namunalar olindi. Taklif etilgan (AES shifrlash algoritmi va MD5 xesh funksiyasidan foydalanilgan) va mavjud generatorlarning testlashdan olingan natijalar 2.3-jadvalda keltirilgan.

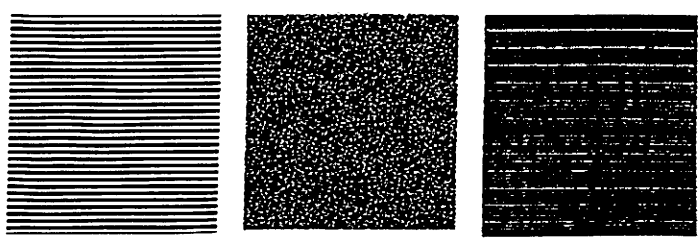
Mazkur testlar to'plamida 15 ta test mavjud bo'lib, testlash natijasi jadvalda a/b ko'rinishida berilgan. Bunda b – jami testlar soni, ya'ni 15 ga teng. a – ketma-ketlikning nechta testdan o'tganini anglatadi. Agar $14/15$ – bo'lsa, demak, ketma-ketlik 15 ta testdan 14 tasidan o'tganini bildiradi. Natijalarning to'liq ko'rinishi 5 – ilovaga keltirilgan. Taklif etilgan PTSGni amalga oshirishda $H()$ – funksiya sifatida $MD5$ – xesh funksiyasidan va $E()$ – shifrlash funksiyasi sifatida AES simmetrik blokli shifrlash standartidan foydalanildi.

2.3-jadval

Generatorlarning testlash natijalari

Generator nomi	Testlash namunalari				
	1	2	3	4	5
CryptGenRandom	14/15	15/15	14/15	13/15	15/15
/dev/urandom	15/15	15/15	15/15	15/15	15/15
Java Random()	15/15	15/15	15/15	15/15	14/15
Python Random()	14/15	15/15	15/15	15/15	15/15
Taklif etilgan PTSG	14/15	15/15	15/15	15/15	15/15

Mazkur testlar to'plamidagi har bir test usuli ma'lum maqsadga qaratilgan bo'lib, ular umumiy holda 6-ilovada keltirilgan. Statistik testlardan tashqari, PTSGdan hosil bo'lgan qiymatlarni grafik tasvirlash (randogramma) orqali ham undagi tasodifiylik darajasiga baho berish mumkin. Xususan, 2.9-rasmda turli tasodifiylik darajasi qayd etgan PTSGning randogrammasi keltirilgan [65].

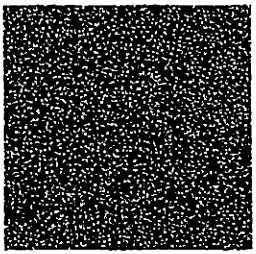


a) "Yomon" tasodifiy *b) Haqiqiy tasodifiy* *c) Yaxshi tasodifiy*

2.9-rasm. **PTSGlarni tasodifiylik darajasini randogramma orqali tasvirlash**

Taklif etilgan PTSG uchun randogrammani hosil qilish uchun 125000 bayt hosil qilindi va undan 62500 baytlar jufti hosil qilindi. Ushbu baytlar juftini [0, 255] oraliqdagi X va Y koordinatalar shaklida tasvirlab, 2.10-rasmda keltirilgan randogramma hosil qilindi.

Statistik testlar natijasi va randogramma orqali taklif etilgan PTSGning tasodifiylik darajasi mavjudlari orasida yuqoriligini ko'rish mumkin.

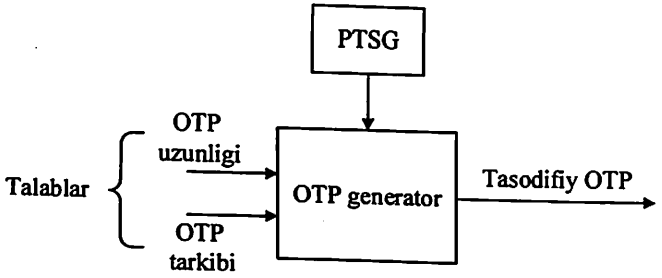


2.10-rasm. **PTSG tasodifiyligini randogramma orqali ko'rsatish**

Har bir taklif etilgan PTSGlarni baholashda turli kriteriyalarga asoslaniladi. Ular orasida *tekis taqsimlanganlik (равномерность), mustaqillik (независимость)* va *qat'iylik (стохастичность)* kriteriyalari muhim ahamiyat kasb etadi. Tekis taqsimlanganlikni NIST SPECIAL PUBLICATION 800-22 statistik testlar to'plamidagi *bitlar bo'yicha chastotalar testi (The Frequency Monobit Test), kumulyativ yig'indi testi (The Cumulative Sums Test)* va *bloklar bo'yicha chastota testi (Test for Frequency within a Block)* yordamida aniqlanadi. Ketma-ketlikdagi bitlarning bir-biridan mustaqilligini baholashda esa *davriylik testi (The Serial Test)* va *seriyali test (Runs Test)* qat'iylikni tekshirish uchun foydalaniladi. Boshqacha aytganda, NIST SPECIAL PUBLICATION 800-22 statistik testlar to'plami o'zida PTSGni baholashda zarur bo'lgan barcha kriteriyalarni jamlagan.

2.3. Psevdotasodifiy sonlarga asoslangan bir martali parollarni generatsiyalash usuli va algoritmi

Bir martali parollar foydalanuvchilarini autentifikatsiyalashda asosan ikkinchi faktor sifatida amalda keng qo'llaniladi. Bunda ikki holat: OTP bir tomonda generatsiya qilinib, ikkinchi tomonga uzatilishi yoki ikkala tomonda bir vaqtning o'zida generatsiya qilinishi mumkin. Birinchi usul amalda keng qo'llanilib, serverda generatsiya qilingan OTPlar mijozga SMS yoki pochta orqali yuboriladi. Mazkur usulda OTPlarni generatsiya qilish biror sanoq yoki vaqtning sinxron bo'lishini talab etmaydi. Ikkinchi usulda OTPni generatsiya qilish ikki tomonda sanoq yoki vaqtni sinxronlashga asoslanadi. Bunda odatda "savol-javob"ga asoslangan generatorlardan foydalaniladi. Birinchi usulda OTPni generatsiyalashning umumiy ko'rinishi 2.11-rasmda keltirilgan [66].



2.11-rasm. **Tizim PTSGga asoslangan OTP generatorining umumiy ko'rinishi**

OTP generatorlari uchun hosil bo'layotgan bir martali parollarni *tekis taqsimot* funksiyasiga (continuous uniform distribution) bo'ysunishi muhim hisoblanadi. *Tekis taqsimot* yoki *to'rtburchak taqsimot* bu – o'zgarmas ehtimollikka ega bo'lgan taqsimot bo'lib, chiquvchi qiymatlarni aniq bir oraliqda yotishini anglatadi. Ushbu oraliq a – minimal qiymat va b – maksimal qiymat parametrlari bilan belgilanib, $U(a,b)$ shaklida belgilanadi. OTP generatori uchun esa hosil bo'layotgan bir martali parollarni paydo bo'lish ehtimolini bir xil bo'lishini anglatadi. Uzluksiz tekis taqsimotning ehtimollik zichlik funksiyasi (Probability density function, PDF) quyidagiga teng:

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a,b] \\ 0, & x \notin [a,b] \end{cases} \qquad (2.1)$$

(2.1) tenglikda $a = 0$ va $b = 1$ ga teng bo'lsa, u holda $U(a,b)$ taqsimot standart tekis taqsimot funksiyasi deb ataladi.

Odatda simmetrik blokli shifrlar va xesh funksiyalar tekis taqsimot funksiyasiga bo'ysunmaydi. Shu sababli, ulardan hosil bo'lgan qiymatlarni tekis taqsimot qonuniyatiga bo'ysunishi uchun qo'shimcha usullardan foydalanish talab etiladi. Tekis taqsimot qonuniyatiga asoslangan ketma-ketliklarni olishning qator usullari mavjud bo'lib, ular orasidan irratsional sonlarga asoslangan yondashuv alohida ahamiyatga ega. Xususan, D.I. Golenko tomonidan psevdotasodifiy sonlarni generatsiyalashning Neyman usuli aynan irratsioanal sonlar xususiyatidan kelib chiqqan holda modifikatsiyalangan [3]. Bunda muallif tomonidan nazariy tekis taqsimot funksiyasidan kichik farq qiluvchi quyidagi funksiyani keltirgan:

$$\xi_i = \{i\Theta\} \qquad (2.2)$$

Bu yerda, Θ – irratsional son bo‘lib, muallif tomonidan unga misol sifatida $\sqrt{2}$, $\frac{\sqrt{2}}{2}$, $\frac{\sqrt{3}}{3}$, $\sqrt{3}$ va $\frac{\sqrt{5}-1}{2}$ irratsional sonlari keltirilgan. Mazkur yondashuvchi to‘g‘riligini tekshirish uchun quyidagi tajriba o‘tkazildi:

Tajriba 1. $1 \leq i \leq 1000000$ oraliqda Θ – irratsional sonning yuqorida keltirilgan qiymatlari va tahlillar asosida tanlab olingan $\sqrt{7}$, $\frac{\sqrt{7}}{7}$ irratsional sonlari va π soni uchun (2.2) tenglik qiymatlari hisoblanib, olingan natijalarning kasr qismlaridan 6 ta va 7 tadan sonlar olinib, ularni tekis taqsimot qonuniga asoslanishi tekshirildi. Olingan natijalar 2.4-jadvalda keltirilgan.

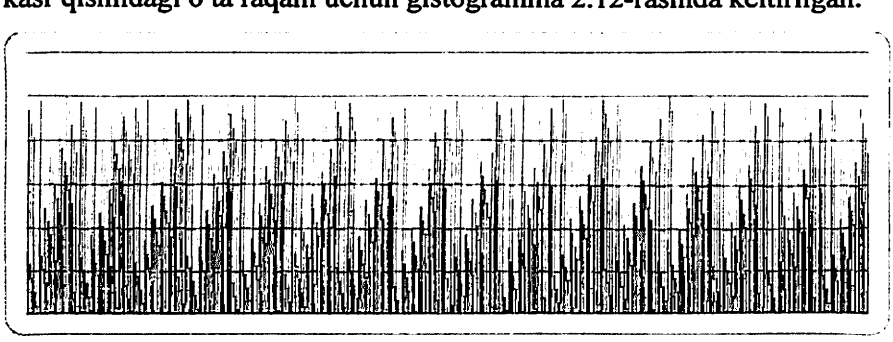
2.4-jadval

Tekis taqsimot qonuniga bo‘ysunishni aniqlash

Irratsional son (Θ)	**1 marta takrorlanishlar soni**	**2 marta takrorlanishlar soni**	**3 va undan ortiq takrorlanishlar soni**
1	**2**	**3**	**4**
6 ta uzunlikdagi kasr qismi uchun			
$\sqrt{2}$	736340	131830	0
$\frac{\sqrt{2}}{2}$	657502	171249	0
$\frac{\sqrt{3}}{3}$	859182	70409	0
$\sqrt{5}$	459738	270131	0
$\frac{\sqrt{5}-1}{2}$	717992	141004	0
$\sqrt{7}$	883440	58280	0
$\frac{\sqrt{7}}{7}$	385430	307285	0
$\pi = 3.141592\ldots$	476474	261763	0
7 ta uzunlikdagi kasr qismi uchun			
$\sqrt{2}$	1 milllion	0	0
$\frac{\sqrt{2}}{2}$	1 milllion	0	0
$\frac{\sqrt{3}}{3}$	1 milllion	0	0

1	2	3	4
$\sqrt{5}$	1 milllion	0	0
$\frac{\sqrt{5}-1}{2}$	1 milllion	0	0
$\sqrt{7}$	1 milllion	0	0
$\frac{\sqrt{7}}{7}$	1 milllion	0	0
π $= 3.141592 \ldots$	1 milllion	0	0

Tajriba natijasi tanlab olingan irratsional sonlar orasidan $\sqrt{7}$ soni qolganlariga qaraganda tekis taqsimot funksiyasiga eng ko'p bo'ysunishini ko'rish mumkin. 7 va undan ortiq raqamlar olinganda $1 \leq i \leq 1000000$ oraliqda takrorlanish holatlari kuzatilmadi. Xususan, $\sqrt{7}$ irratsional son uchun $1 \leq i \leq 1000$ oraliqda ξ_i funksiya qiymatining kasr qismidagi 6 ta raqam uchun gistogramma 2.12-rasmda keltirilgan.



2.12-rasm. $\sqrt{7}$ irratsional son uchun $1 \leq i \leq 1000$ oraliqda ξ_i funksiya qiymati gistogrammasi

Olingan natijalar $1 \leq i \leq 1000000$ oraliq tekis taqsimot funksiyasiga bo'ysunishini ko'rsatsada, amalda foydalanish uchun i tasodifiy tanlanmasa, hosil bo'luvchi OTPlarni faraz qilish mumkin bo'ladi. Shu sababli, (2.2) tenglikka asoslangan holda bir martali parollarni generatsiyalashda i tasodifiy tanlash muhim. Quyida mazkur masala ko'rib o'tiladi.

Taklif etiladigan OTP generatori yordamida parollarni generatsiyalashda quyidagi farazlar va shartlar olindi:

1. Talab etiladigan bir martali parolning uzunligi kamida 6 ta va ko'pi bilan 10 ta bo'lishi. Ushbu shart OTPlarga qo'yilgan standart talab hisoblanadi [84, 85].

2. Talab etiladigan bir martali parol turli $[0,\dots,9]$, $[0,\dots,9,A,\dots,Z]$ va $[0,\dots,9,a,\dots,z,A,\dots,Z]$ belgilar to'plamidan iborat bo'lishi. OTPlar qisqa vaqtda foydalanilishi va kiritishdagi noqulayliklar sababli maxsus belgilar (masalan, @,?,!) kiritilmadi.

Olingan faraz va shartlar asosida bir martali parollarni generatsiyalash ketma-ketligi bilan quyida tanishib chiqiladi. Buning uchun quyidagi belgilanishlar qabul qilindi:

1. $S_0 = \{0,\dots,9\}, S_1 = \{0,\dots,9,A,\dots,Z\}, S_2 = \{0,\dots,9,a,\dots,z,A,\dots,Z\}$ belgilar to'plami. Bunda har bir to'plamdagi belgilar soni $len(S_0) = 10, len(S_1) = 36, len(S_2) = 62$ ga teng.

2. $R = \{L, S_i\}$ – bir martali parol uchun talablar bo'lib, L – bir martali parol uzunligini ($L \in [6, 10]$), S_i – belgilar to'plamini bildiradi ($i \in [0,2]$).

3. C_j^l – PTSGdan hosil bo'lgan j – tasodifiy qiymat bo'lib, l – uning bitdagi o'lchamini ko'rsatadi.

4. P_j – generatsiya qilingan j – bir martali parol.

5. $F(R, C_j^l)$ – bir martali parolni hosil qilish funksiyasi bo'lib, bu yerda, $P_j = F(R, C_j^l) = F(\{L, S_i\}, C_j^l)$ tenglik o'rinli.

6. $DT(N_j, L)$ – dinamik qisqartirish funksiyasi bo'lib, uning tavsifi quyida keltirilgan.

7. $unique$ – mantiqiy qiymat bo'lib, hosil bo'layotgan OTPlarni takrorlanmasligini anglatadi. Mazkur parametr OTP generatoridan offlayn (ya'ni, ma'lum sondagi OTPlar talab qilinganda) holatda foydalanishda ishlatiladi. Agar $unique = true$ bo'lganida, OTP generatori foydalanuvchiga offlayn ravishda talab qilingan sondagi unikal OTPlarni taqdim etadi. $unique = false$ holatida esa, OTP generatori uzluksiz ravishda ishlaydi, ya'ni, har murojaatda bitta OTPni hosil qiladi.

$DT(N_j, L)$ – dinamik qisqartirish funksiyasi bo'lib, OTP uzunligiga L bog'liq holda P_j bir martali parolni hosil qilib beradi:

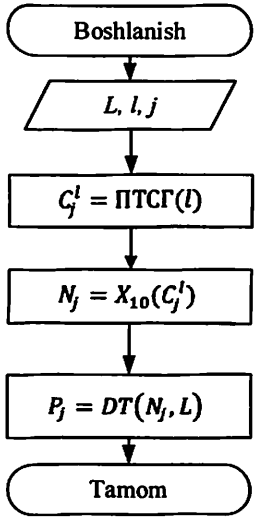
$$DT(N_j, L) = S(M(N_j * \Theta), L) \qquad (2.3)$$

Bu yerda, $M()$ – funksiya, $Nonce$ tasodifiy qiymatni Θ irratsional songa ko'paytirib, kasr qismini qaytaradi. $S()$ – funksiya esa, $M()$ funksiya natijasining o'ng tomonidan L ta raqamni qaytaradi.

Taklif etilayotgan OTP generatori uchta S_0, S_1 ва S_2 to'plamdagi belgilardan iborat bo'lgan parollarni generatsiya qiladi. Shuning uchun, har bir to'plamga tegishli belgilarni generatsiya qilish jarayoni alohida ko'rib o'tiladi.

1 – algoritm. **S_0 to'plam belgilaridan iborat parollarni hosil qilish.**

Faqat raqamdan iborat bo'lgan OTPlar amalda eng keng qo'llaniladi. S_0 to'plam elementlaridan iborat bo'lgan OTPlarni generatsiya qilish funksiyasi $F(R, C_j^l) = F(\{L, S_0\}, C_j^l)$ga teng. Bu holda talab etiluvchi OTPni uzunligi L ning ixtiyoriy mumkin bo'lgan qiymatlari uchun PTSGdan bir blok uzunligidagi tasodifiy qiymat olinadi. Bu holda OTPni generatsiya qilishning umumiy ketma-ketligi quyidagicha (2.13-rasm):



2.13-rasm. **1 – algoritm. S_0 to'plam belgilaridan iborat parollarni hosil qilish usulining blok sxemasi**

a. Talab etilgan P_j bir martali parolni generatsiya qilish uchun PTSGidan l bit uzunligidagi C_j^l generatsiya qilinadi.

b. C_j^l bitlar ketma-ketligidan o'nlik sanoq tizimidagi $N_j = X_{10}(C_j^l)$ hosil qilinadi.

c. Talab etilgan P_j bir martali parolni hosil qilish uchun $P_j = DT(N_j, L)$ tenglikdan foydalaniladi.

d. Agar generatordan unikal bo'lgan OTPlarni hosil qilish talab qilinsa (ya'ni, offlayn holatda foydalanilganda $unique = true$), u holda hosil bo'lgan OTPlar saqlanib borilib, keyingisini hosil qilishda o'zidan oldingilariga tengligi tekshiriladi. Agar oldindan aniq bo'lmagan sondagi OTPlarni generatsiyalash talab qilinsa, ya'ni, OTP generatoridan uzluksiz holatda foydalanilganda ushbu parametr $unique = false$ deb qaraladi.

2-algoritm. **S_1 to'plam belgilaridan iborat parollarni hosil qilish.**

S_1 to'plam elementlaridan iborat bo'lgan OTPlarni generatsiya qilish funksiyasi $F(R, C_i^l) = F(\{L, S_1\}, C_i^l)$ga teng. Bu holda talab etiluvchi OTPni uzunligi L ning ixtiyoriy mumkin bo'lgan qiymatlari uchun PTSGdan l bit uzunligidagi tasodifiy bitlar ketma-ketligi olinadi. Bu holda OTPni generatsiya qilish ketma-ketligi quyidagicha (7-ilova):

a. Talab etilgan P_i bir martali parolni generatsiya qilish uchun PTSGidan l bit C_i^l generatsiya qilinadi.

b. C_j^l bitlar ketma-ketligidan o'nlik sanoq tizimidagi $N_j = X_{10}(C_j^l)$ hosil qilinadi.

c. N_j va L asosida G_j kattalik quyidagicha hosil qilinadi: $G_j = DT(N_j, L')$. Bu yerda, $L' = len(len(S_1)^L)+1$ ga teng bo'lib, $len(S_1) = 36$ ga tengligini inobatga olib, $L' = len(36^L)+1$ yozish mumkin. $len(A)$ – funksiya esa A sonning o'nlik sanoq tizimidagi uzunligini ifodalaydi. Shundan kelib, chiqib, $G_j = DT(N_j, len(36^L) + 1)$ni yozish mumkin.

d. Ajratilgan G_j qiymat asosida P_j bir martali parolni hosil qilishda $P_j = base36encode(G_j)$ funksiyasidan foydalaniladi. Ushbu funksiyaning psevdokodi quyida keltirilgan:

```
def base36encode(integer):
    chars =
    '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    password = ''
    while integer > 0:
    integer, remainder = divmod(integer, 36)
    password = chars[remainder] + password
    return password
```

e) Agar generatordan unikal bo'lgan OTPlarni hosil qilish talab qilinsa (ya'ni, offlayn holatda foydalanilganda $unique = true$), u holda

hosil bo'lgan OTPlar saqlanib borilib, keyingisini hosil qilishda o'zidan oldingilariga tengligi tekshiriladi. Agar oldindan aniq bo'lmagan sondagi OTPlarni generatsiyalash talab qilinsa, ya'ni, OTP generatoridan uzluksiz holatda foydalanilganda ushbu parametr $unique = false$ deb qaraladi.

3-algoritm. **S_2 to'plam belgilaridan iborat parollarni hosil qilish.**

S_2 to'plam elementlaridan iborat bo'lgan OTPlarni generatsiya qilish funksiyasi $F(R, C_j^l) = F(\{D, S_2\}, C_j^l)$ga teng. Bu holda talab etiluvchi OTPni uzunligi L ning ixtiyoriy mumkin bo'lgan qiymatlari uchun PTSGdan bir blok uzunligidagi (l bit) tasodifiy bitlar ketma-ketligi olinadi. Bu holda OTPni generatsiya qilishning umumiy ketma-ketligi quyidagicha (8-ilova):

a. Talab etilgan P_i bir martali parolni generatsiya qilish uchun PTSGidan l bit C_i^l generatsiya qilinadi.

b. C_j^l bitlar ketma-ketligidan o'nlik sanoq tizimidagi $N_j = X_{10}(C_j^l)$ hosil qilinadi.

c. N_j va L asosida G_j kattalik quyidagicha hosil qilinadi: $G_j = DT(N_j, L')$. Bu yerda, $L' = len(len(S_2)^L)+1$ ga teng bo'lib, $len(S_2) = 62$ ga tengligini inobatga olib, $L' = len(72^L)+1$ yozish mumkin. $len(A)$ – funksiya esa A sonning o'nlik sanoq tizimidagi uzunligini ifodalaydi. Shundan kelib chiqib, $G_j = DT(N_j, len(62^L) + 1)$ni yozish mumkin.

d. Ajratilgan G_j qiymat asosida P_j bir martali parolni hosil qilishda $P_j = base62encode(G_j)$ funksiyasidan foydalaniladi. Ushbu funksiyaning psevdokodi quyida keltirilgan:

```
def base62encode(integer):
    chars =
    '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgh
    ijklmnopqrstuvwxyz'
    password = ''
    while integer > 0:
    integer, remainder = divmod(integer, 62)
    password = chars[remainder] + password
    return password
```

e) Agar generatordan unikal bo'lgan OTPlarni hosil qilish talab qilinsa (ya'ni, offlayn holatda foydalanilganda $unique = true$), u holda

hosil bo'lgan OTPlar saqlanib borilib, keyingisini hosil qilishda o'zidan oldingilariga tengligi tekshiriladi. Agar oldindan aniq bo'lmagan sondagi OTPlarni generatsiyalash talab qilinsa, ya'ni, OTP generatoridan uzluksiz holatda foydalanilganda ushbu parametr $unique = false$ deb qaraladi.

Taklif etilgan OTP ni generatsiya qilish usulini afzalligini tasdiqlash uchun quyidagi tajriba amalga oshirildi.

Tajriba 2. Taklif etilgan 1- algoritm ($\Theta = \sqrt{7}$ uchun) asosidagi OTP generatorining takrorlanmaslik darajasi quyidagi mavjud generatorlar bilan taqqoslandi:

1. HOTP generatori.
2. Turli dasturlash tillarida mavjud $random()$ funksiyasi.

Har bir algoritm yordamida 6 va 7 uzunlikdagi 1 millionta OTP hosil qilindi. Olingan tajriba natijalari 2.5-jadvalda aks ettirilgan.

2.5-jadval

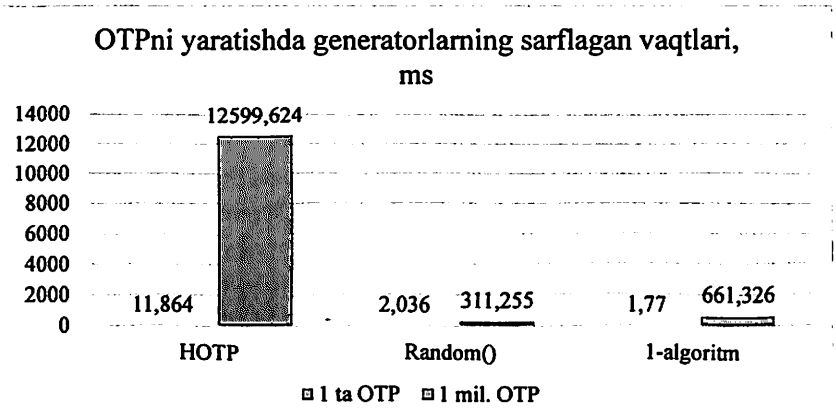
S_0 to'plam elementlarini hosil qilish natijalari

OTP generatorlari (uzunligi)	Takrorlanishlar soni										Takrorlanmaslik
	2 marta	3 marta	4 marta	5 marta	6 marta	7 marta	8 marta	9 marta	9 dan ortiq	jami	
HOTP (6)	184085	61208	15331	3080	490	71	11	0	0	264276	367957
random()(6)	183757	61646	15322	3078	517	72	8	2	0	264402	367182
1-algoritm (6)	68394	2898	0	0	0	0	0	0	0	71292	854518
HOTP (7)	44964	1547	43	0	0	0	0	0	0	46554	905253
random()(7)	44923	1472	40	0	0	0	0	0	0	46435	905578
1-algoritm (7)	0	0	0	0	0	0	0	0	0	0	1000000

Olingan tajriba natijasi taklif etilgan OTPni generatsiyalash usuli mavjudlariga qaraganda yuqori takrorlanmaslik darajasiga egaligini ko'rsatadi. Agar biror tashkilot uchun kunlik OTPning zaruriyati 1 millionga yaqin bo'lsa, u holda 7 xonali OTPlardan foydalanish tavfsiya etiladi. Kunlik talab etiladigan OTPlar soni 1 milliondan ancha kam bo'lganda 6 xonali OTPlardan foydalanish mumkin bo'ladi.

OTP generatorlari uchun tezkorlik muhim ahamiyatga ega. Shu sababli, 2 – tajriba doirasida olingan generatorlarning 1 ta va 1 millionta 7 xonali bir martali parolni generatsiyalash uchun sarflagan vaqtlari hisoblandi. Hisoblash, 2.5 GGs chastotaga ega Core(TM) i5-7300HQ prosessor, 8 GB tezkor xotiraga ega Windows 10x64 muhitida ANTS Performance Profiler 10 ilovasidan foydalangan holda amalga oshirildi. Hisoblash natijalari diagramma ko'rinishida 2.14-rasmda keltirilgan.

Tahlil natijasi taklif etilgan 1-algoritmni bitta OTPni hosil qilishda yuqori tezkorlikni qayd etgan.



2.14-rasm. **OTP generatorlarining tezkorlik tahlili**

Bundan tashqari, taklif etilgan bir martali parollarni generatsiyalash usuli yordamida S_2 to'plam elementlaridan iborat bo'lgan 1 million bir martali parollar generatsiya qilindi va mazkur holatda to'liq takrorlanmaslik darajasini qayd etdi (9-ilova). Olingan natijalar to'liq kafolatni ta'minlay olmasada (sababi 10^8 parolni generatsiyalash va testlash talab etiladi), amalda foydalanish uchun yetarli bardoshlikka ega. Shu sababli, taklif etilgan parollarni generatsiyalash algoritmini amalda foydalanish mumkin.

III BOB. BIR MARTALI PAROLGA ASOSLANGAN AUTENTIFIKATSIYA PROTOKOLLARI

3.1. Bir martali parollarga asoslangan autentifikatsiya protokollarini takomillashtirish

Mazkur bo'lim 2.3-bo'limda keltirilgan bir martali parollar generatori asosida autentifikatsiya usullaridagi xavfsizlik muammolarini bartaraf etishga va ularni tahlillashga bag'ishlangan. Ushbu bo'limda amalda foydalanilayotgan bir martali parollarga asoslangan autentifikatsiya usullaridagi xavfsizlik muammolari keltirilib, so'ng ularni bartaraf etish usuli taklif etiladi.

*1. **HOTP yoki TOTP algoritmiga asoslangan usullar.*** Vaqtni (yoki sanoqni) sinxronlashga asoslangan TOTP (yoki HOTP) algoritmi hozirgi kunda ijtimoiy tarmoqlarda va aksariyat web tizimlarda foydalanilmoqda. Xususan, ushbu algoritmlar asosida ishlab chiqilgan Google Authenticator va Microsoft Authenticator bularga yaqqol misol bo'ladi. Ushbu ikki algoritm asosida ishlab chiqilgan mobil ilovalarda quyidagi kamchiliklar va xavfsizlik muammolari mavjud:

– QR – kod orqali taqdim etilgan taqsimlangan kalitning xavfsizligi ta'minlanmagan [92];

– TOTP protokoliga asoslangan ilovalarda odatda OTP ko'p vaqt davomida (kamida 30 sek) foydalanish uchun joiz bo'lib, bu vaqtda uni ikki marta kiritsa bo'ladi [93];

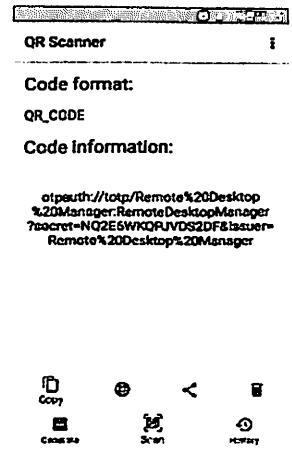
– TOTP protokoli "o'rtaga turgan odam", fishing va zararli dasturlardan foydalanib amalga oshirilgan hujumlarga bardoshsiz [92].

Yuqorida keltirilgan birinchi xavfsizlik muammosi shundan iboratki, Google Authenticator ilovasidan foydalanish mumkin bo'lgan tizimlarda taqdim etilgan QR kod ochiq ko'rinishda bo'lib, ixtiyoriy QR kodni skanerlash vositasi yordamida taqsimlangan (maxfiy) kalitni aniqlashi mumkin. 3.1-rasmda Google Authenticator ilovasi tomonidan taqdim etilgan QR kodni oddiy QR Scanner ilovasidan foydalanib qo'lga kiritish natijasi keltirilgan.

Quyidagi muammoni o'zida bartaraf etgan, yangi TOTP algoritmiga asoslangan autentifikatsiya usulida ro'yxatga olish jarayonining umumiy ko'rinishi 3.2-rasmda keltirilgan. Bunga ko'ra,

foydalanuvchi dastlab biror xizmat provayderidan ro'yxatdan o'tadi. Bunda, foydalanuvchi ro'yxatdan o'tish davomida birinchi faktor sifatida *login* va *password*ni tanlaydi. Shundan so'ng, ikki faktorli autentifikatsiyani yoqish uchun xizmat provayderiga murojaat qiladi. Umumiy holda taklif etilgan usulda ro'yxatdan o'tish quyidagi tartibda amalga oshiriladi: $K_{SH} = PTSG()$

a) QR kod

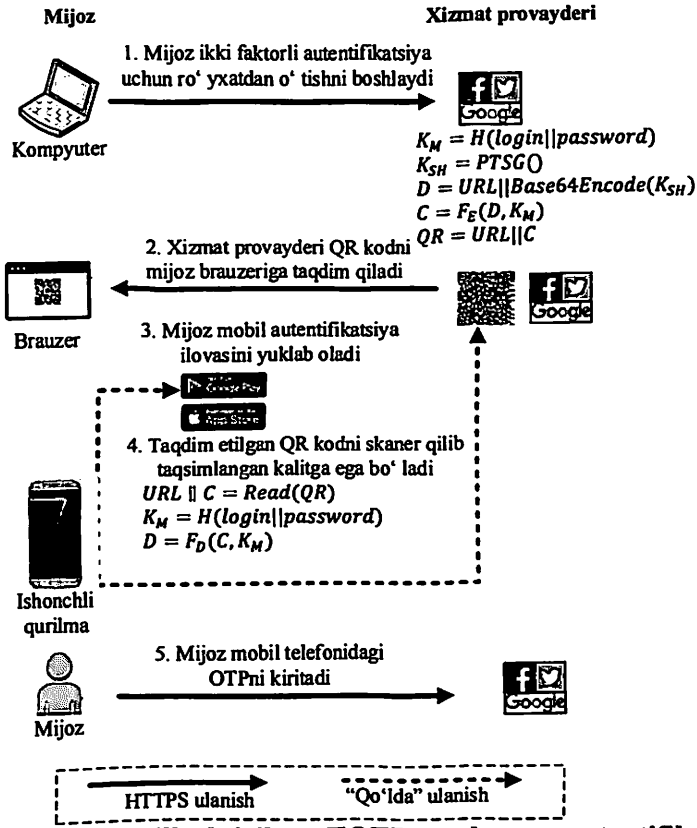


b) QR Scanner ilovasidan o'qishdan keyingi natija

3.1-rasm. **Ochiq ko'rinishda taqdim etilgan QR kod**

1. Xizmat provayderi foydalanuvchining birinchi faktori (*login* va *password*) orqali bosh kalit K_M ni hosil qiladi: $K_M = H(login||password)$. Bu yerda, H – bir tomonlama xesh funksiya bo'lib, uning tavsifi quyida keltirilgan.

2. Shundan so'ng, 2.2-bo'limda keltirilgan PTSG yordamida taqsimlangan kalit – K_{SH} generatsiya qilinadi. Hosil qilingan taqsimlangan kalit bayt ko'rinishda bo'lsa, $Base64$ kodlash orqali qator ko'rinishiga keltiriladi va u xizmat turini anglatuvchi (URL) nom bilan birlashtiriladi: $D = URL||Base64Encode(K_{SH})$.



3.2-rasm. **Takomillashtirilgan TOTP asoslangan autentifikatsiya protokolida ro'yxatdan o'tish bosqichi**

3. Hosil bo'lgan ma'lumot D bosh kalit K_M yordamida biror simmetrik blokli shifrlash algoritmi F orqali shifrlanadi: $C = F_E(D, K_M)$. Bu yerda, F_E – shifrlash algoritmi F dagi shifrlash funksiyasi.

4. Hosil qilingan shifrmatnni *URL* bilan birlashtirib, *QR* kod ko'rinishida foydalanuvchi brauzerida taqdim etiladi: $QR = URL||C$. Bundan asosiy maqsad foydalanuvchi tomonidan onlayn bo'lmagan holda *QR* ni to'g'ri deshifrlanganini tekshirishdir.

5. Foydalanuvchi kerakli bo'lgan mobil ilovani o'rnatgandan so'ng, xizmat provayderi uchun zarur bo'lgan birinchi faktorni (*login, password*) kiritadi. Kiritilgan kattaliklar orqali bosh kalit K_M yuqorida keltirilgani kabi hisoblanadi.

6. QR kodni skanerlash orqali esa *URL* va shifrmatn C ga ega bo'linadi. Shifrmatn C ni K_M yordamida deshifrlab, *URL* va K_{SH} olinadi.

Agar olingan ikki *URL* qiymatlar teng bo'lgan taqdirda foydalanuvchining birinchi autentifikatsiya faktorlari (*login, password*) to'g'ri deb aniqlanadi va taqsimlangan kalit K_{SH} qabul qilinadi.

7. Shundan so'ng, qurilmada hosil qilingan OTPni xizmat provayderiga kiritish orqali ikki tomonning to'g'ri sinxronlashgani tekshiriladi. 3.3-rasmda takomillashtirilgan TOTP usulida hosil qilingan QR kod va uni ixtiyoriy QR kod skanerlaridan olingan natijasi keltirilgan.

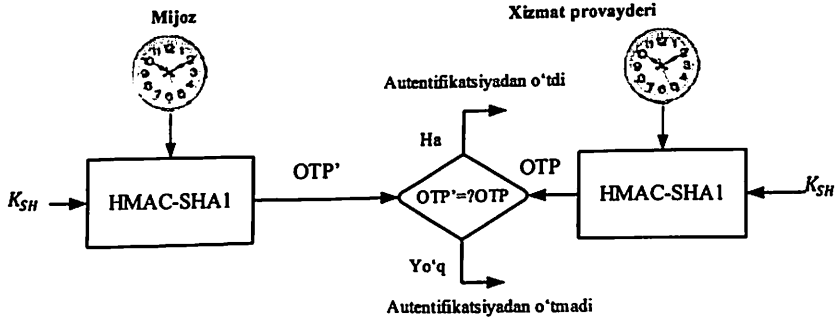
3.3-rasm. **Takomillashtirilgan TOTP algoritmiga asoslangan autentifikatsiya protokolida QR kod va uni o'qib olish natijasi**

H – funksiya. $H()$ – funksiya sifatida ixtiyoriy bardoshli bo'lgan xesh – funksiyalardan foydalanish mumkin. Amalga oshirishdagi qulaylik uchun 256 bit xesh qiymat qaytaruvchi xesh funksiyalarni tanlash maqsadga muvofiq (masalan, SHA256). Ushbu funksiya yagona kirish qiymati M dan $H(M)$ shaklida yoki kirish qiymati ikkita bo'lsa, ularni birlashtirgan holda, $(H(M_1||M_2)$ shaklida) hisoblanadi.

F – funksiya. Ushbu funksiya sifatida ixtiyoriy bardoshli simmetrik blokli shifrlash algoritmidan foydalanish mumkin bo'lib, amalga oshirishdagi qulayligi va xavfsizlik nuqtayi nazaridan 256 bitli kalitni madadlovchi algoritmlarni tanlash maqsadga muvofiq hisoblanadi. Masalan, AES, ГОСТ 28147-89 va h.k.

TOTP algoritmini takomillashtirishda foydalanilgan xesh funksiya va blokli simmetrik algoritmlar umumiy holda keltirilgan bo'lib,

loyihalovchi talabidan kelib chiqib tanlanishi mumkin. Autentifikatsiyadan o'tish jarayonining umumiy ko'rinishi esa quyidagi 3.4-rasmda aks ettirilgan.



3.4-rasm. **TOTP algoritmi asosida kirish bosqichi**

Taklif etilgan mazkur takomillashtirilgan usul taqsimlangan kalitni maxfiyligini ta'minlashni maqsad qilib, qolgan xavfsizlik muammolarini bartaraf etmaydi. Xususan, TOTP autentifikatsiya algoritmida mavjud bo'lgan – "o'rtada turgan odam" hujumini oldini olish uchun, HTTPS tarmoqdan foydalanishning o'zi yetarli bo'lsa, fishing hujumlarini oldini olish uchun ikki tomonlama autentifikatsiyani amalga oshirish va buning natijasida xizmat provayderini haqiqiyligini ta'minlash amalga oshiriladi.

Bundan tashqari, TOTP algoritmida mavjud bo'lgan jiddiy muammolardan yana biri bu – mobil qurilmalarni zararli dasturiy vositalar orqali zararlanishi. Xususan, zararli dasturiy vositalar TOTP algoritmining taqsimlangan kalitini qo'lga kiritishni maqsad qiladi. Mazkur holatda himoyani tashkil qilishda mobil qurilmadagi operatsion tizim imkoniyatining o'zi ham aksariyat hollarda yetarli hisoblanadi. Masalan, Android OTda kriptografik kalitlarni xavfsiz saqlash imkonini beruvchi AndroidKeyStore [67], Java Keystores [68] kabi imkoniyatlar mavjud.

Masalan, kriptografik kalitlarni va parollarni saqlash uchun Android operatsion tizimida AndroidKeyStore deb nomlanuvchi provayder mavjud bo'lib, u Android 4.3 (API level 18) va undan yuqori versiyalarda mavjud [94]. Ushbu provayderda saqlangan kalitlarni olish jarayoni quyidagi keltirilgan:

```
    KeyStore ks =
KeyStore.getInstance("AndroidKeyStore");
    ks.load(null);
    Enumeration<String> aliases = ks.aliases();
```

Bundan tashqari, ushbu provayder turli kriptografik algoritmlar uchun kalitlar va parametrlarni generatsiya qilish, import va eksport qilish imkoniyatiga ham ega.

2. SMS xabar yordamida OTPni yuborishga asoslangan usul. Bir martali parollarga asoslangan autentifikatsiyaning mazkur usuli onlayn bank xizmatlarida va to'lovlarni tasdiqlashda keng qo'llaniladi. Autentifikatsiyaning mazkur usuliga asoslangan ko'plab tizimlarni o'rganish natijasida quyidagi xavfsizlik muammolari aniqlandi:

– bir martali parol SMS xabarda ochiq tarzda yetkazilgani va SS7 [86] protokolidagi zaiflik sababli uni ruxsatsiz qo'lga kiritish mumkin;

– SMS xabarda ochiq holda yuborilgan OTPlarni SIM kartani klonlash, o'g'irlash va telefonda himoya bo'lmagan holda ko'chirib olish kabi hujumlar orqali qo'lga kiritish mumkin [95];

– bir martali parollar xizmat provayderlari tomonidan generatsiya qilinadi va odatda yuqori takrorlanmaslik darajasiga ega emas;

– aksariyat tizimlarda (UPAY, PAYME) OTP foydalanilmagunga qadar takror uzatiladi va buning natijasida uni qo'lga kiritish imkoniyati ortadi.

SMS xizmati orqali yetkazilgan OTPlarga asoslangan autentifikatsiya usullaridan asosan bank-moliya sohasida to'lovlarni tasdiqlash uchun keng foydalaniladi. Bundan tashqari, aksariyat ijtimoiy tarmoqlarning mobil versiyalarida ham foydalanuvchi telefon nomerini mavjudligini tekshirish uchun OTPlardan foydalanadi. Hozirda SMS orqali OTPni uzatish usulining amalda keng tarqalganiga asosiy omil bu – faqat ixtiyoriy ko'rinishdagi mobil qurilmasini o'zi bo'lishini yetarligi. Boshqa OTPga asoslangan autentifikatsiya usullari, masalan, HOTP, TOPT yoki pochta orqali uzatish usullaridan foydalanish uchun mobil qurilmalarga ma'lum talablar qo'yiladi. Masalan, kamera bo'lishi yoki Internet tarmog'iga ulangan bo'lishi shular jumlasidan.

SMS xabar orqali OTPni uzatishda uning maxfiyligi ta'minlanmagani va hattoki, bu hol hozirda mobil banking ilovalarida ham mavjudligi sababli, quyida SMS xabar orqali OTPni xavfsiz uzatish imkonini beruvchi usul keltirilgan. Taklif etilgan OTPni uzatishning mazkur usuli mavjud mobayl banking ilovalarida qo'shimcha bir modul

ko'rinishida yoki boshqa hollarda alohida mobayl ilova ko'rinishida amalga oshirilishi mumkin.

Ro'yxatdan o'tish bosqichi. Oddiy SMS xizmatiga asoslangan ikki faktorli autentifikatsiya usulini yoqish juda oddiy bo'lib, ro'yxatdan o'tish davomida foydalanuvchiga tegishli bo'lgan mobil telefon raqamini kiritishning o'zi yetarlidir. Biroq, taklif etilgan usulda ro'yxatdan o'tish alohida ilova yoki mavjud mobayl banking ilovalarida qo'shimcha modul ko'rinishida bo'lgani bois, quyidagi ketma-ketliklar asosida amalga oshiriladi:

1. Foydalanilayotgan xizmat sozlanishidan ikkinchi faktorga asoslangan autentifikatsiya tanlanadi.

2. Foydalanuvchiga tegishli bo'lgan mobayl telefon raqami kiritiladi va shundan so'ng yuqorida keltirilgan takomillashtirilgan TOTP autentifikatsiya algoritmi kabi tomonlar orasida taqsimlangan kalit K_{SH} ga ega bo'linadi.

Alohida mobayl ilova ko'rinishida bo'lganida dastlab ilova mobil qurilmasida o'rnatiladi va ro'yxatdan o'tish jarayoni davomida sozlanadi. Taqsimlangan kalitlarni xavfsiz saqlash ham yuqorida keltirilgan kabi amalga oshiriladi. Bunda, ro'yxatdan o'tkazish jarayonida himoyalangan kanaldan (masalan, HTTPS) foydalanish talab etiladi.

Kirish bosqichi. Taklif etilgan usul asosida autentifikatsiyadan o'tish ketma-ketligi 3.5-rasmda keltirilgan:

1. Foydalanuvchi dastlab xizmat provayderida biror operatsiyani amalga oshiradi. Masalan, kirishda birinchi faktordan so'ng ikkinchi faktor sifatida OTPdan foydalanganda yoki to'lovni amalga oshirish tasdig'i talab etilganda.

2. Xizmat provayderi 2.3-bobda keltirilgan OTP generatori orqali zarur bo'lgan OTPni generatsiyalaydi va u asosida umumiy yuboriluvchi xabar M ni hosil qiladi. Xabarni foydalanuvchiga ro'yxatdan o'tish jarayonida taqsimlangan kalit K_{SH} bilan biror simmetrik blokli shifrlash algoritmi F_E asosida shifrlaydi: $C = F_E(M, K_{SH})$. Bu yerda, F_E – simmetrik blokli shifrlash algoritmining shifrlash funksiyasi.

3. Ushbu bosqichda shifrlangan xabar C biror SMS shlyuziga yuboradi.

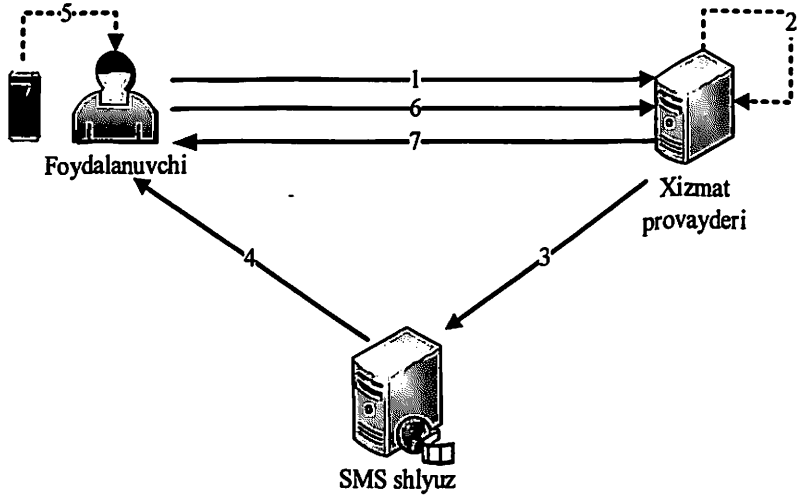
4. SMS shlyuz orqali shifrmatn C foydalanuvchi mobayl qurilmasiga uzatiladi.

5. Mobil ilovadagi maxsus ilova yoki mobayl banking ilovasining moduli orqali shifrlangan xabar C deshifrlanadi: $M = F_D(C, K_{SH})$. Bu

yerda, F_D – simmetrik blokli shifrlash algoritmining deshifrlash funksiyasi.

6. Deshifrlangan xabar M dan OTP so'ralgan manzilga kiritiladi.

7. Ushbu bosqichda xizmat provayderidan olingan autentifikatsiya natijasi foydalanuvchiga taqdim etiladi.



3.5-rasm. **Xavfsiz SMS xabarga asoslangan autentifikatsiya protokolining umumiy ko'rinishi**

SMS orqali OTPni yuborish usulini takomillashtirish uchun taklif etilgan usul, hozirda mobil qurilmalarning yetarlicha imkoniyatga egaligiga asoslanadi. Xususan, moqil qurilmada talab etilgan ilovani o'rnatish va undan foydalanish imkoniyati mavjud bo'lishi kerak.

Taklif etilgan usulda ham taqsimlangan kalit K_{SH} ni xavfsizligini ta'minlash muhim ahamiyatga ega. Umumiy holda, SMS orqali OTPni yuborish usuli qator davlat standartlaridan chiqarib tashlanayotan bir vaqtda [4], taklif etilgan usul uning foydalanish davrini qisman bo'lsa ham ortishiga sabab bo'ladi.

3.2. "Savol-javob" mexanizmiga asoslangan bir martali parol yordamida autentifikatsiyalash protokoli

Ushbu bo'limda vazifasi nuqtayi nazaridan HOTP yoki TOTP protokollariga o'xshash, biroq "savol-javob" mexanizmiga asoslangan bir martali parollar asosida autentifikatsiya usulini ishlab chiqish masalasi ko'rilgan. Taklif etilayotgan mazkur autentifikatsiya usulida

foydalanish uchun qulay bo'lishini inobatga olib, QR kod texnologiyasidan foydalanilgan.

Bir martali parollarga asoslangan autentifikatsiyaning mazkur usulini yaratishda HOTP va TOTP protokollarini amalga oshirishda kuzatiladigan quyidagi zaifliklarni va noqulayliklarni bartaraf etish maqsad qilib olindi:

1. HOTP va TOTP protokollari mos holda sanoq va vaqtni sinxronlashga asoslangani bois, ba'zida sinxronlashdagi kamchiliklar kuzatiladi. Masalan, TOTP protokolida OTPni kiritish davomida uning amal qilish muddatini tugashi yoki HOTP protokolida sanoq qiymatini ikki tomonda turlicha bo'lishi kuzatiladi.

2. HOTP va TOTPga asoslangan ilovalarda taqsimlangan kalitlarning maxfiyligi ta'minlanmagan.

3. HOTP va TOTPga asoslangan ilovalarda urinishlar soni cheklanmaganligi natijasida "qo'pol kuch" hujumiga zaif bo'lishi mumkin.

Keltirilgan birinchi kamchilikni bartaraf etishda "savol-javob" usuli samarali hisoblanadi. Ushbu usulda "savol" har doim serverdan (xizmat provayderidan) mijozga uzatilgani bois, sinxronlashni talab etmaydi. "Savol"ni mijozga yetkazishning ko'plab usullari mavjud. Masalan, pochta orqali, SMS xabarda va h.k. Biroq, taklif etilayotgan usulda "savol"ni yetkazishda QR kod texnologiyasidan foydalanish maqsad qilindi. QR kod texnologiyasi mashina tomonidan oson o'qish imkoniyatini bergani bois, ortiqcha sarf-xarajat va noqulayliklarni oldini oladi.

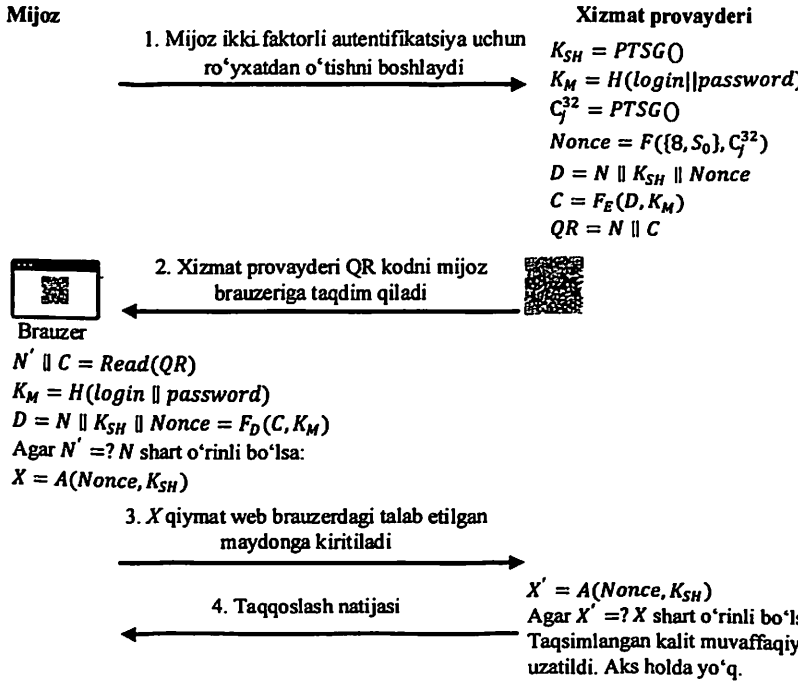
Keltirilgan ikkinchi kamchilik bo'yicha 3.1-bobda keltirilgan TOTP protokolini takomillashtirish usulidan foydalaniladi. Ya'ni, foydalanuvchining birinchi parametri asosida hosil qilingan *master_key* asosida taqsimlangan kalit shifrlanadi.

"Qo'pol kuch" hujumi kalit yoki parolni bo'lishi mumkin bo'lgan barcha variantini hisoblashga asoslanadi. HOTP yoki TOTPga asoslangan aksariyat ilovalarda foydalanuvchi tomonidan qisqa vaqtda kiritilishi uchun 6 xonali OTPlar generatsiya qilinadi. Bu agar amalga oshirilgan tizimda urinishlar soni bo'yicha cheklovlar mavjud bo'lmaganda, "qo'pol kuch" hujumini amalga oshirilishi imkoniyatini ortishiga sababchi bo'ladi. Shuni hisobga olgan holda, mavjud kamchilikni bartaraf etishda OTPni uzunligini va belgilar tarkibini

oshirish hamda nojoiz urinishlar bo'yicha cheklovlarni o'rnatish maqsad qilindi.

Umumiy holda, taklif etilayotgan bir martali parollarga asoslangan autentifikatsiya jarayoni: *ro'yxatdan o'tish* va *kirish* bosqichlaridan iborat.

Ro'yxatdan o'tish bosqichi. Ushbu bosqich 3.1-bobda keltirilgan bo'lib, dastlab foydalanuvchi xizmat provayderidan birinchi faktor bo'yicha (login va parol) ro'yxatdan o'tadi. Shundan so'ng, ikki faktorli autentifikatsiya xizmatini yoqish uchun xizmat provayderida quyidagi sozlanishlarni amalga oshiradi (3.6-rasm):



3.6-rasm. **"Savol-javob" mexanizmiga asoslangan bir martali parol yordamida autentifikatsiya usulining ro'yxatdan o'tish bosqichi**

1. Oldindan tanlangan bir tomonlama funksiya A uchun 2.2-bobda keltirilgan PTSG tomonidan taqsimlangan kalit K_{SH} hosil qilinadi. A funksiyaning tavsifi quyida keltirilgan.

2. Foydalanuvchining birinchi autentifikatsiya faktorlari (login va parol) asosida bir tomonlama $H()$ funksiya asosida $F()$ shifrlash

algoritmi uchun bosh kalit K_M hosil qilinadi. $H()$ va $F()$ funksiyalarning tavsifi yuqorida keltirilgan.

3.2.3 - bobda keltirilgan *1-algoritm* asosida uzunligi 8 ga teng (L = 8) bo'lgan bir martali parol - $Nonce$ hosil qilinadi.

4. Xizmat provayderining nomi N taqsimlangan kalit K_{SH} va $Nonce$ bilan birlashtiriladi va D kattalik hosil qilinadi: $D = N||K_{SH}||Nonce$.

5. Shundan so'ng, quyidagi shifrmatn hosil qilinadi: $C = F_E(D, K_M)$. Bu yerda, $F_E()$ – funksiya $F()$ shifrlash algoritmining shifrlash funksiyasi.

6. Xizmat provayderi nomi N bilan shifrmatnni biriktirish orqali QR kod hosil qilinadi: $QR = N||C$.

7. Hosil bo'lgan QR kod xizmat provayderi tomonidan foydalanuvchi brauzerida taqdim etiladi.

8. Taqdim etilgan QR kod ostida esa OTPni kiritish bandi bo'lib, undan QR kod asosida taqsimlangan kalit to'g'ri uzatilganini tekshirish uchun foydalaniladi.

Keltirilgan jarayonlar server tomonda amalga oshirilgani bois, ikki faktorli autentifikatsiya usuli tanlanganda avtomatik taqdim etiladi. Taqdim etilgan QR kodni o'qib olish va uni mobil qurilmada saqlash quyidagicha amalga oshiriladi:

1. Foydalanuvchi dastlab zarur bo'lgan mobayl ilovani yuklab oladi.

2. Mobayl ilovada talab qilingan birinchi autentifikatsiya faktorlarini (login va parol) kiritib, QR kodni o'qib olish tugmasini bosadi.

3. Shundan so'ng, $QR = N||C$ ga teng bo'lgan ma'lumot o'qib olinadi va uning asosida quyidagi hisoblanishlar amalga oshiriladi:

a. Birinchi autentifikatsiya faktorlari asosida bosh kalit K_M serverdagi kabi hisoblanadi.

b. Hosil qilingan bosh kalit asosida shifrmatn C deshifrlanadi: $D = F_D(C, K_M)$. Bu yerda, F_D shifrlash algoritmi F dagi deshifrlash funksiyasi.

c. Deshifrlangan matn D uchta tashkil etuvchiga ajratiladi: $D = N||K_{SH}||Nonce$.

d. Agar QR kodni bevosita o'qishdan olingan N va c bosqichdan olingan N qiymatga o'zaro teng bo'lsa, kiritilgan birinchi faktorlar to'g'ri va deshifrlash muvaffaqqiyatli amalga oshirilganini anglatadi.

e. Agar d bosqichdan qoniqarli natija olinsa, taqsimlangan kalitni to'g'ri uzatilganini tasdiqlash uchun foydalanilgan *Nonce* va K_{SH} asosida $X = A(Nonce, K_{SH})$ hisoblanadi.

4. Olingan X qiymat web brauzerdagi tasdiqlash maydoniga kiritiladi. Agar kiritilgan X qiymat serverdagi kabi bo'lsa, har ikkala tomonda taqsimlangan kalit K_{SH} bir xilligi va ikkinchi faktor bo'yicha autentifikatsiyalash muvaffaqiyatli amalga oshirilganini anglatadi.

5. Shundan so'ng, taqsimlangan kalit K_{SH} mobil qurilmada qo'shimcha himoya usullari asosida xavfsiz saqlanadi.

A – funksiya. Ushbu funksiya kiritilgan ikki qiymatlar $(Nonce, K_{SH})$ asosida "savol-javob" mexanizmidagi "javob"ni shakllantiradi. Ushbu funksiya ikki qismdan iborat bo'lib, dastlab kirish qiymatlari $C_1^{32} = HMAC\_H256(Nonce, K_{SH})$ funksiyaga kiritiladi. $HMAC\_H256()$ – funksiyasi ixtiyoriy bardoshli va amalga oshirish nuqtayi nazaridan tanlangan 256 bit (32 bayt) xesh qiymat qaytaruvchi xesh funksiyaga asoslangan *HMAC* funksiyasidir (masalan, *HMAC_SHA256()*) va shuning uchun natija C_1^{32} tarzida ifodalandi. A funksiyaning ikkinchi tashkil etuvchisi esa, hosil bo'lgan 256 bitli C_1^{32} qiymatdan bir martali parolni hosil qiladi. Foydalanuvchi tomonidan tizimga kiritishda qulay bo'lishi uchun faqat raqamdan iborat bo'lgan 8 ta belgidan iborat OTP generatsiyalash sharti qo'yildi va bunda funksiyaning ikkinchi tashkil etuvchisi sifatida 2.3-bobda keltirilgan *1-algoritm*dan foydalanish mumkin: $Responce = F(\{L, S_0\}, C_i^l) = F(\{8, S_0\}, C_1^{32})$. Umumiy holda esa $Responce = A(Nonce, K_{SH})$ ko'rinishda bo'ladi.

Yuqoridagi ketma-ketliklar muvaffaqiyatli amalga oshirilgandan so'ng, ro'yxatdan o'tish jarayoni tugagan hisoblanadi. "Savol-javob" mexanizmiga asoslangan bir martali parol asosida autentifikatsiya usulining autentifikatsiyadan o'tish bosqichi esa quyidagicha amalga oshiriladi:

Kirish bosqichi. Taklif etilgan autentifikatsiya usuli biror web xizmat tizimiga kirishdagi ikkinchi faktor sifatida yoki mobayl banking tizimida to'lovlarni tasdiqlashda foydalanish mumkin. Buning uchun quyidagi ketma-ketliklarni amalga oshiriladi (3.7-rasm):

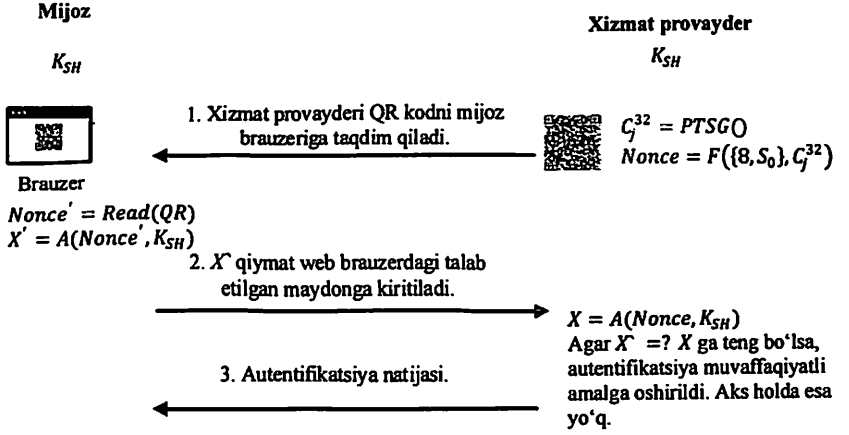
1. Xizmat provayderi tomonidan 2.3-bobga keltirilgan OTP generatori asosida 8 ta xonadan iborat va tanlovga ko'ra faqat raqamdan iborat bo'lgan OTP – *Nonce* hosil qilinadi: $(R = \{L, S_i\} = \{8, S_0\})$.

2. Hosil qilingan *Nonce* xizmat provayderi tomonidan foydalanuvchi web brauzerida QR kod ko'rinishida taqdim etiladi.

3. Taqdim etilgan QR koddagi *Nonce* "savol-javob" mexanizmi uchun "savol" vazifasini bajargani bois, unga mos "javob"ni kiritish uchun web brauzerga maxsus band taqdim etiladi.

4. Foydalanuvchi tomonidan maxsus ilova mavjud mobayl qurilma yordamida web brauzerdagi QR kod o'qiladi va olingan *Nonce* asosida A funksiya asosida "javob" hisoblanadi: $Responce = A(Nonce, K_{SH})$.

5. Agar web brauzerda kiritilgan *Responce* server tomonidan to'g'ri deb topilsa, autentifikatsiya muvaffaqiyatli deb hisoblanadi. Aks holda autentifikatsiya jarayoni muvaffaqiyatsiz deb topiladi.



3.7-rasm. **"Savol-javob" mexanizmiga asoslangan bir martali parol yordamida autentifikatsiya usulining kirish bosqichi**

"Savol-javob" mexanizmiga asoslangan ikkinchi faktor sifatida foydalaniluvchi mazkur autentifikatsiya usuli qurilma ko'rinishidagi tokenga asoslangan usulga o'xshash bo'lib, vaqt/sanoqni sinxronlash o'rniga "savol" web brauzerdan olinadi hamda unga mos hisoblangan "javob" tegishli manzilga kiritiladi.

3.3. Bir martali parollarga asoslangan xavfsiz autentifikatsiya protokoli

Birinchi bobning 3-bo'limi "savol-javob"ga asoslangan bir martali parollar yordamida autentifikatsiya protokollarining tahliliga bag'ishlanb, unda mazkur sohada mavjud bo'lgan ochiq xavfsizlik muammolari hamda amalga oshirishdagi kamchiliklar sanab o'tilgan edi. Ushbu bo'limda, sanab o'tilgan kamchiliklarni bartaraf etuvchi va amalga oshirishda samara beruvchi, savol-javob mexanizmidagi bir

martali parollarga asoslangan autentifikatsiya protokolini ishlab chiqish masalasi ko'rib chiqiladi.

Taklif etilayotgan autentifikatsiya protokolida W.C.Ku tomonidan ishlab chiqilgan xeshlashga asoslangan autentifikatsiya usuli asos qilib olindi [47]. Shuning uchun, dastlab mazkur autentifikatsiya protokoli bilan tanishib o'tiladi.

W.C. Ku tomonidan ishlab chiqilgan autentifikatsiya usuli mijozdan biror ma'lumotni (paroldan tashqari) xavfsiz saqlashni talab etmaydi.

Mazkur protokolda quyidagi belgilanishlardan foydalanilgan:

- U foydalanuvchini, S serverni va A hujumchini anglatadi.
- $h()$ kriptografik xesh-funksiyani bildiradi. $h(m)$ orqali m ma'lumotni bir marta xeshlanganligi tushinilsa, $h^2(m)$ esa m ma'lumotni ikki marta xeshlanganini bildiradi, ya'ni, $h^2(m) = h(h(m))$.
- N butun son bo'lib, U ni ro'yxatdan o'tkazish uchun 1 dan boshlanadi va har bir autentifikatsiyadan o'tish jarayonida ortib boradi.
- P foydalanuvchi U ning bardoshli paroli.
- K_S server S ning maxfiy kaliti.
- T vaqt metkasi bo'lib, foydalanuvchi U ro'yxatdan o'tkazilgan yoki qayta o'tkazilgan vaqtni bildiradi.
- $\oplus$ – belgisi XOR amalini va || - belgisi birlashtirish amalini anglatadi.
- $A \rightarrow B: X$ belgilanish A tomon X xabarni B tomonga xavfsiz bo'lmagan kanal orqali yuborganini anglatadi.
- $A \Rightarrow B: X$ belgilanish A tomon X xabarni B tomonga xavfsiz kanal orqali yuborganini anglatadi.

Ushbu protokol ikki: *ro'yxatdan o'tish* va *kirish* bosqichidan iborat.

Ro'yxatdan o'tish bosqichi. Ushbu muolaja foydalanuvchi U serverga ro'yxatdan o'tishni talab etganda amalga oshiriladi (3.8-rasm).

1-cўpoв. $U \longrightarrow S$: Ro'yxatdan o'tish so'rovi.

2-cўpoв. $S \longrightarrow U: N, T$.

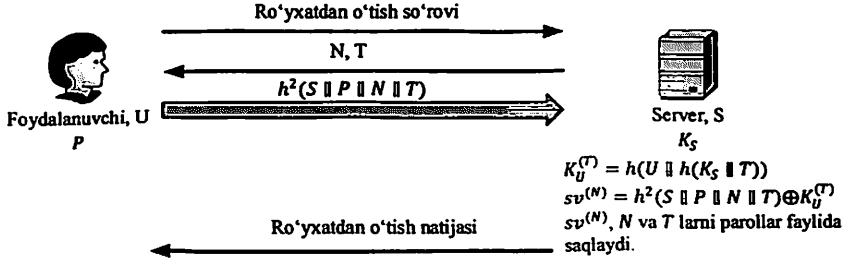
S o'zida T ni joriy vaqt metkasi sifatida o'rnatadi. Agar ushbu so'rov ro'yxatga olish jarayonida uzatilayotgan bo'lsa, server $N = 1$ holatni, aks holda $N = N + 1$ ni o'rnatadi. Shundan so'ng, S tomoni N, T ni U ga uzatadi.

3-cўpoв. $U \Rightarrow S: h^2(S \parallel P \parallel N \parallel T)$.

U tomoni tekshirishda foydalaniluvchi $h^2(S \parallel P \parallel N \parallel T)$ ni generatsiya qiladi va uni S ga yuboradi.

4-cўров. $S \longrightarrow U$: Ro'yxatdan muvaffaqiyatli o'tganlik haqidagi ma'lumot.

S tomoni foydalanuvchi U ma'lumotini saqlash uchun $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ kalitni va foydalanuvchini to'g'riligini tasdiqlovchi $sv^{(N)} = h^2(S \parallel P \parallel N \parallel T) \oplus K_U^{(T)}$ hisoblaydi hamda S tomon $sv^{(N)}$, N va T larni parollar faylida saqlaydi.



3.8-rasm. **W.C. Ku protokolining ro'yxatdan o'tish bosqichi**

Kirish bosqichi. Ushbu muolaja U tomoni S ga autentifikatsiyadan o'tish uchun amalga oshiradi (3.9-rasm).

1-cўров. $U \longrightarrow S$: Autentifikatsiya so'rovi.

2-cўров. $S \longrightarrow U: r, n, t.$

S tomoni bir martali parol r ni generatsiya qiladi va parollar faylidan olingan $n = N$ va $t = T$ larga qo'shib U ga yuboradi.

3-cўров. $U \longrightarrow S: c_1, c_2, c_3.$

U tomoni S ga quyidagilarni yuboradi:

$$c_1 = h^2(S \parallel P \parallel n \parallel t) \oplus h(S \parallel P \parallel n \parallel t)$$
$$c_2 = h^2(S \parallel P \parallel n+1 \parallel t) \oplus h(S \parallel P \parallel n \parallel t)$$
$$c_3 = h(h^2(S \parallel P \parallel n+1 \parallel t) \parallel r)$$

4-cўров. $S \longrightarrow U$: Autentifikatsiyadan o'tganlik yoki o'ta olmaganlik haqida ma'lumot.

S tomoni $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ ni hisoblaydi va u orqali saqlangan $sv^{(N)}$ dan $h^2(S \parallel P \parallel n \parallel t)$ ni ajratib oladi, ya'ni:

$$h^2(S \parallel P \parallel n \parallel t) = sv^{(N)} \oplus K_U^{(T)}.$$

Shundan so'ng, S tomoni u_1 va u_2 ni quyidagicha hisoblaydi:

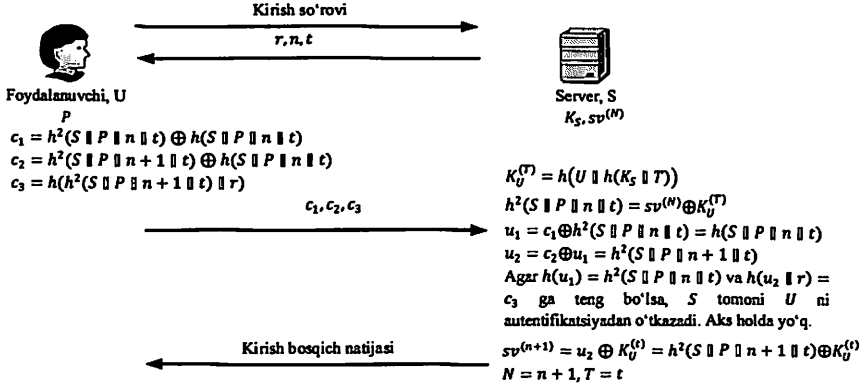
$$u_1 = c_1 \oplus h^2(S \parallel P \parallel n \parallel t) = h(S \parallel P \parallel n \parallel t)$$
$$u_2 = c_2 \oplus u_1 = h^2(S \parallel P \parallel n+1 \parallel t).$$

Agar $h(u_1) = h^2(S \parallel P \parallel n \parallel t)$ va $h(u_2 \parallel r) = c_3$ ga teng bo'lsa, S tomoni U ni autentifikatsiyadan o'tkazadi. Aks holda, S tomoni A ning autentifikatsiyadan o'tish so'rovini rad etadi va sessiyani tugatadi.

Muvaffaqiyatli autentifikatsiyadan so'ng esa, S tomoni keyingi autentifikatsiya so'rovi uchun tasdiqlovchi kattalikni hisoblaydi:

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S \parallel P \parallel n+1 \parallel t) \oplus K_U^{(t)}$$

va U ning keyingi kirishi uchun $sv^{(N)}$ ni $sv^{(n+1)}$ bilan almashtiradi va $N = n+1$ kabi o'rnatadi. T ning qiymati o'zgarmas saqlanadi, ya'ni, $T = t$.



3.9-rasm. **W.C. Ku protokolining kirish bosqichi**

W.C. Ku protokolining tahlili. Mazkur autentifikatsiya protokoli ko'plab olimlar tomonidan tahlil qilingan [48, 49] bo'lib, aksariyat hujumlar aynan $h^2(S \parallel P \parallel n \parallel t)$ tekshiruvchini o'g'irlashdan keyin amalga oshirilgani aytib o'tilgan. Biroq, mazkur ishni amalga oshirishda S ning maxfiy kalitini bilish talab etiladi (ya'ni, K_S ni bilish). Agar hujumchi S ning maxfiy kalitini bilsa, unda protokolni buzishdan ko'ra jiddiyroq muammolar paydo bo'lishi mumkin. Bundan tashqari, W.C.Ku tomonidan taqdim etilgan protokolda quyidagi xavfsizlik muammolari mavjud:

1. Sodda paroldan foydalangan taqdirda "faraz qilish" bo'yicha hujumga bardoshsiz. Ya'ni, hujumchi tutib olingan c_1 xesh qiymat asosida parol P ni taxmin qilgan holda quyidagi tenglikni tekshirishga harakat qiladi:

$$c_1 = h^2(S \parallel P \parallel n \parallel t) \oplus h(S \parallel P \parallel n \parallel t).$$

Agar tenglikni qanoatlantiruvchi P ning qiymati aniqlansa, u holda protokolning xavfsizligi yo'qqa chiqadi.

2. Bundan tashqari, W.C.Ku protokoli ikki tomonlama autentifikatsiyani ta'minlay olmaydi. Ya'ni, foydalanuvchi server haqiqiyligini tekshira olmaydi.

3. Mazkur protokol faqat autentifikatsiyani amalga oshiradi. Seans kalitlarini almashinish imkoniyatini taqdim etmaydi.

4. Qolgan autentifikatsiya usullariga qaraganda yuqori samaradorlikka ega emas (1.5-jadvalga qaralsin).

Quyidagi keltirilgan kamchiliklarni o'zida bartaraf etgan W.C. Ku usulining modifikatsiyalangan shakli va SAS-2 protokoli ko'rinishidagi shakllari keltirilgan.

W.C. Ku usulining modifikatsiyalangan shakli. Ushbu usul uchta: *ro'yxatdan o'tish, kirish* va *parolni almashtirish* bosqichlaridan iborat [25].

Ro'yxatdan o'tish. Ushbu muolaja foydalanuvchi U serverga ro'yxatdan o'tishni talab etganda amalga oshiriladi (3.10-rasm).

1-cўpoв. $U \longrightarrow S$: Ro'yxatdan o'tish so'rovi.

2-cўpoв. $S \longrightarrow U: N, T.$

S o'zida T ni joriy vaqt metkasi sifatida o'rnatadi. Agar ushbu so'rov ro'yxatga olish jarayonida uzatilayotgan bo'lsa, server $N = 1$ holatni, aks holda $N = N + 1$ ni o'rnatadi. Shundan so'ng, S tomoni N, T ni U ga uzatadi.

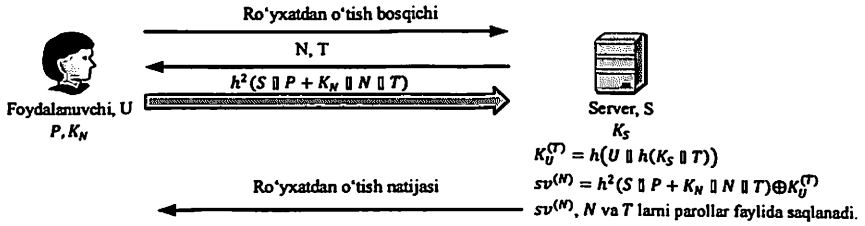
3-cўpoв. $U \Rightarrow S: h^2(S \parallel P + K_N \parallel N \parallel T).$

U tomonini tekshirishda foydalaniluvchi $h^2(S \parallel P + K_N \parallel N \parallel T)$ ni generatsiya qiladi va S ga yuboradi. Bu yerda, K_N – tasodifiy son bo'lib, mijoz tomonda generatsiya qilinadi va xavfsiz saqlanadi.

4-cўpoв. Ro'yxatdan muvaffaqiyatli o'tganlik haqidagi ma'lumot.

S tomoni foydalanuvchi U ma'lumotini saqlash uchun $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ kalitni va foydalanuvchini to'g'riligini tasdiqlovchi

$sv^{(N)} = h^2(S \parallel P + K_N \parallel N \parallel T) \oplus K_U^{(T)}$ hisoblaydi va S tomon $sv^{(N)}$, N va T ni parollar faylida saqlaydi.



3.10-rasm. **Ro'yxatdan o'tish bosqichi**

Kirish bosqichi. Ushbu bosqichda U tomoni S tomonidan autentifikatsiyadan o'tish uchun amalga oshiradi (3.11-rasm).

1-cўpoв. $U \longrightarrow S$: Autentifikatsiya so'rovi.

2-cўpoв. $S \longrightarrow U: r, n, t.$

S tomoni bir martali parol r ni generatsiya qiladi va parollar faylidan olingan $n = N$ va $t = T$ larga qo'shib U ga yuboradi.

3-cўpoв. $U \longrightarrow S: c_1, c_2, c_3.$

U tomoni K_{N+1} generatsiya qiladi va quyidagilarni hisoblaydi:

$$c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$
$$c_2 = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$
$$c_3 = h(h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \parallel r)$$

S ga c_1, c_2, c_3 larni yuboradi.

4-cўpoв. $S \longrightarrow U: c_4.$

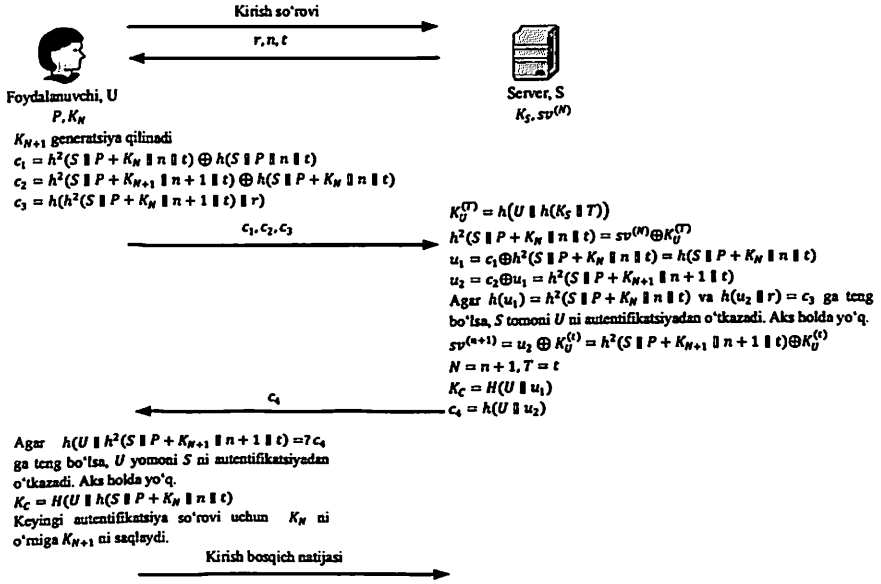
S tomoni $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ ni hisoblaydi va u orqali saqlangan $sv^{(N)}$ dan $h^2(S \parallel P + K_N \parallel n \parallel t)$ ni ajratib oladi, ya'ni:

$$h^2(S \parallel P + K_N \parallel n \parallel t) = sv^{(N)} \oplus K_U^{(T)}.$$

Shundan so'ng, S tomoni u_1 va u_2 ni quyidagicha hisoblaydi:

$$u_1 = c_1 \oplus h^2(S \parallel P + K_N \parallel n \parallel t) = h(S \parallel P + K_N \parallel n \parallel t)$$
$$u_2 = c_2 \oplus u_1 = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t).$$

Agar $h(u_1) = h^2(S \parallel P + K_N \parallel n \parallel t)$ va $h(u_2 \parallel r) = c_3$ ga teng bo'lsa, S tomoni U ni autentifikatsiyadan o'tkazadi.



3.11-rasm. **Kirish bosqichi**

Muvaffaqiyatli autentifikatsiyadan so'ng, S tomoni keyingi autentifikatsiya so'rovi uchun tasdiqlovchi kattalikni hisoblaydi:

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \oplus K_U^{(t)}$$

U ning keyingi kirishi uchun $sv^{(N)}$ ni $sv^{(n+1)}$ bilan almashtiradi va $N = n + 1$ kabi o'rnatadi. T ning qiymati o'zgarmas saqlanadi, ya'ni, $T = t$.

Autentifikatsiya jarayonidan so'ng foydalanish uchun sessiya kaliti K_C quyidagicha hosil qilinadi:

$$K_C = H(U \parallel u_1).$$

Bu yerda, $H()$ – bir tomonlama funksiya bo'lib, keyinchalik foydalanish uchun talab etilgan sessiya kalitini hosil qilishda foydalaniladi. Shuning uchun, $H()$ – funksiyani tanlaganda shifrlash funksiyasi talabiga va xavfsizlik darajasiga binoan tanlanadi.

U autentifikatsiyadan o'tganidan so'ng, foydalanuvchi uni haqiqiyligini tekshirish uchun quyidagini hisoblaydi va U ga yuboradi:

$$c_4 = h(U \parallel u_2).$$

5-cўpoв. $U \longrightarrow S$: Autentifikatsiyadan o'tganlik yoki o'ta olmaganligi haqidagi so'rovni yuboradi.

Agar $h(U \parallel h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) =? c_4$ ga teng bo'lsa, U tomoni S ni autentifikatsiyadan o'tkazadi va sessiya kaliti K_C ni quyidagicha hisoblaydi:

$$K_C = H(U \parallel h(S \parallel P + K_N \parallel n \parallel t).$$

Shundan so'ng, keyingi autentifikatsiya so'rovi uchun K_N ni o'rniga K_{N+1} ni saqlaydi.

Parolni almashtirish bosqichi. Mazkur jarayon bir martali parollarga asoslangan autentifikatsiya usullari uchun zarur talab hisoblanadi va odatda statik parolga asoslangan autentifikatsiya usulidagi kabi osonlik bilan amalga oshirilmaydi. Shuning uchun ham, aksariyat bir martali parollarga asoslangan autentifikatsiya usullarida parolni almashtirish jarayoni mavjud emas yoki muallif tomonidan e'tiborga olinmagan. W.C. Ku usulida ham muallif tomonidan parolni almashtirish jarayoni keltirilmagan. Taklif etilayotgan modifikatsiyalangan usulda esa parolni almashtirish jarayoni mavjud bo'lib, u quyidagicha amalga oshiriladi (3.12-rasm):

1-cўpoв. $U \longrightarrow S$: Parolni almashtirish so'rovi.

2-cўpoв. $S \longrightarrow U: r, n, t.$

S tomoni bir martali parol r ni generatsiya qiladi va parollar faylidan olingan $n = N$ va $t = T$ larga qo'shib U ga yuboradi.

3-cўpoв. $U \longrightarrow S: c_1, c_2, c_3.$

U tomonidan tasodifiy soni K_{N+1} va yangi parol $\acute{P}$ generatsiya qilinadi va quyidagilar hisoblanadi:

$$c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$
$$c_2 = h^2(S \parallel \acute{P} + K_{N+1} \parallel n + 1 \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$
$$c_3 = h(h^2(S \parallel \acute{P} + K_{N+1} \parallel n + 1 \parallel t) \parallel r)$$

S ga c_1, c_2, c_3 larni yuboradi.

4-cўpoв. $S \longrightarrow U$: Parol almashganligi haqidagi xabar va c_4.

S tomoni $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ ni hisoblaydi va u orqali saqlangan $sv^{(N)}$ dan $h^2(S \parallel P + K_N \parallel n \parallel t)$ ni ajratib oladi, ya'ni:

$$h^2(S \parallel P + K_N \parallel n \parallel t) = sv^{(N)} \oplus K_U^{(T)}.$$

Shundan so'ng, S tomoni u_1 va u_2 ni quyidagicha hisoblaydi:

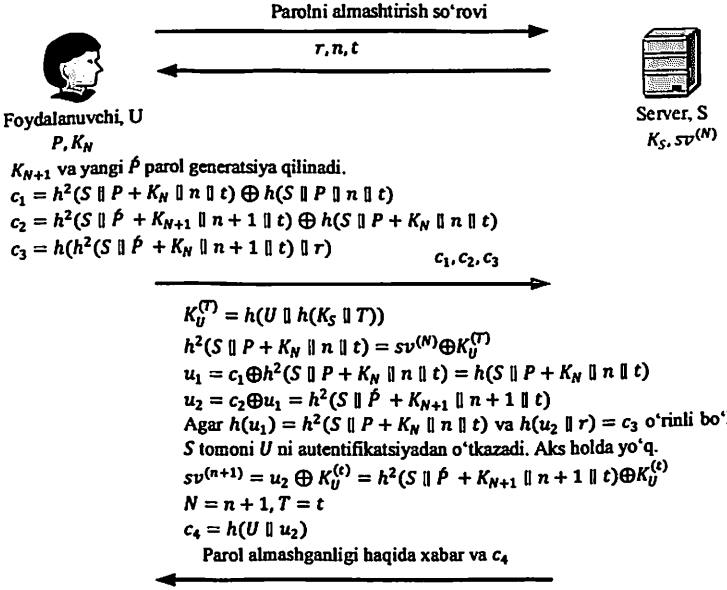
$$u_1 = c_1 \oplus h^2(S \parallel P + K_N \parallel n \parallel t) = h(S \parallel P + K_N \parallel n \parallel t)$$

$$u_2 = c_2 \oplus u_1 = h^2(S \parallel \acute{P} + K_{N+1} \parallel n+1 \parallel t).$$

Agar $h(u_1) = h^2(S \parallel P + K_N \parallel n \parallel t)$ va $h(u_2 \parallel r) = c_3$ ga teng bo'lsa, S tomoni U ni autentifikatsiyadan o'tkazadi va yangi parol asosida hosil qilingan tasdiqlovchi kattalikni hisoblaydi:

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S \parallel \acute{P} + K_{N+1} \parallel n+1 \parallel t) \oplus K_U^{(t)}$$

Shundan so'ng, autentifikatsiya jarayoni uchun $sv^{(N)}$ ni $sv^{(n+1)}$ bilan almashtiradi va $N = 1$ kabi o'rnatadi. Joriy vaqt metkasi esa o'zgarmas saqlanadi.



3.12-rasm. **Parolni almashtirish bosqichi**

Foydalanuvchi uchun server autentifikatsiyasini ta'minlash uchun quyidagi hisoblanadi va parol muvaffaqiyatli almashtirilganligi haqidagi xabarga qo'shib U ga yuboriladi:

$$c_4 = h(U \parallel u_2).$$

Agar $h(U \parallel h^2(S \parallel P + K_{N+1} \parallel n+1 \parallel t) = ? c_4$ ga teng bo'lsa, U tomoni S ni autentifikatsiyadan o'tkazadi va parolni muvaffaqiyatli almashganligi haqidagi xabarga ega bo'ladi.

W.C. Ku usulining SAS-2 protokoli ko'rinishidagi shakli. Ushbu bir martali parollarga asoslangan autentifikatsiya usuli ham uchta: *ro'yxatdan o'tish, kirish* va *parolni almashtirish* bosqichlaridan iborat.

Ro'yxatdan o'tish. Mazkur jarayon foydalanuvchi U birinchi marta server S ga ro'yxatdan o'tishi davomida amalga oshiradi (3.13-rasm).

1-cўpoв. $U \rightarrow S$: Ro'yxatdan o'tish so'rovi.

2-cўpoв. $S \rightarrow U: N, T.$

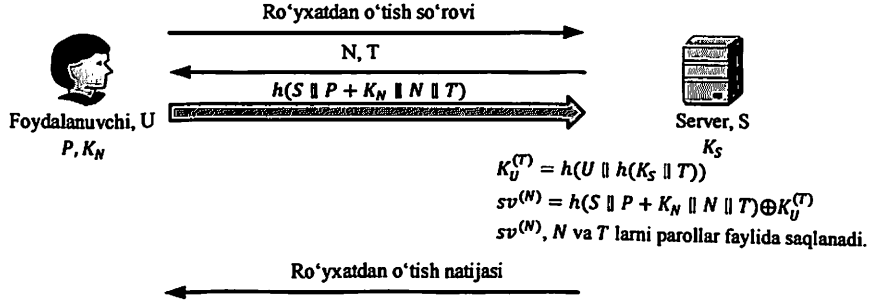
S o'zida T ni joriy vaqt metkasi sifatida o'rnatadi. Agar ushbu so'rov ro'yxatga olish jarayonida uzatilayotgan bo'lsa, server $N = 1$ holatni, aks holda $N = N + 1$ ni o'rnatadi. Shundan so'ng, S tomoni N, T ni U ga uzatadi.

3-cўpoв. $U \Longrightarrow S: X.$

U tomonini tekshirishda foydalaniluvchi $X = h(S \parallel P + K_N \parallel N \parallel T)$ ni generatsiya qiladi va uni S ga yuboradi. Bu yerda, K_N – tasodifiy son bo'lib, mijoz tomonda generatsiya qilinadi va xavfsiz saqlanadi.

4-cўpoв. $S \longrightarrow U$: Ro'yxatdan o'tib bo'lganlik haqidagi ma'lumot.

S tomoni foydalanuvchi U ma'lumotini saqlash uchun $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ kalitni va foydalanuvchini to'g'riligini tasdiqlovchi $sv^{(N)} = X \oplus K_U^{(T)}$ hisoblaydi hamda S tomon $sv^{(N)}$, N va T ni parollar faylida saqlaydi.



3.13-rasm. **Ro'yxatdan o'tish bosqichi**

Kirish bosqichi. Ushbu muolaja U tomoni S ga autentifikatsiyadan o'tishida amalga oshiriladi (3.14-rasm).

1-сўров. $U \longrightarrow S$: Autentifikatsiyadan o'tish so'rovi.

2-сўров. $S \longrightarrow U: r, n, t.$

S tomoni bir martali parol r ni generatsiya qiladi va parollar faylidan olingan $n = N$ va $t = T$ larga qo'shib U ga yuboradi.

3-сўров. $U \longrightarrow S: \alpha, \beta.$

U tomoni K_{N+1} generatsiya qiladi va quyidagilarni hisoblaydi:

$$X = h(S \parallel P + K_N \parallel n \parallel t)$$
$$C = h(S \parallel P + K_{N+1} \parallel n \parallel t)$$
$$h(C) = h(S \parallel C)$$
$$\alpha = C \oplus (h(C) + X)$$
$$\beta = h(C) \oplus X$$

S ga α, β larni yuboradi.

4-сўров. $S \longrightarrow U: \gamma.$

S tomoni $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ ni hisoblaydi va u orqali saqlangan $sv^{(N)}$ dan X ni ajratib oladi, ya'ni:

$$X = sv^{(N)} \oplus K_U^{(T)}.$$

Shundan so'ng, S tomoni quyidagilarni hisoblaydi:

$$h(C) = \beta \oplus X$$
$$C = \alpha \oplus (h(C) + X)$$

Mazkur holda $h(C) =? h(S, C)$ ga teng bo'lsa, S tomoni U ni autentifikatsiyadan o'tkazadi. Aks holda U autentifikatsiyadan o'ta olmaydi.

Autentifikatsiya jarayonidan so'ng foydalanish uchun sessiya kaliti K_C quyidagicha hosil qilinadi:

$$K_C = H(U \parallel h(C) || N).$$

Bu yerda, $H()$ – bir tomonlama funksiya bo'lib, keyinchalik foydalanish uchun talab etilgan sessiya kalitini hosil qilishda foydalaniladi. Shuning uchun, $H()$ – funksiya shifrlash funksiyasi talabiga va xavfsizlik darajasiga binoan tanlanadi.

U autentifikatsiyadan o'tganidan so'ng, foydalanuvchi uni haqiqiyligini tekshirish uchun quyidagini hisoblaydi va U ga yuboradi:

$$\gamma = h(S \parallel h(C)).$$

Muvaffaqiyatli autentifikatsiyadan so'ng esa, S tomoni keyingi autentifikatsiya so'rovi uchun tasdiqlovchi kattalikni hisoblaydi:

$$sv^{(n+1)} = C \oplus K_U^{(t)}.$$

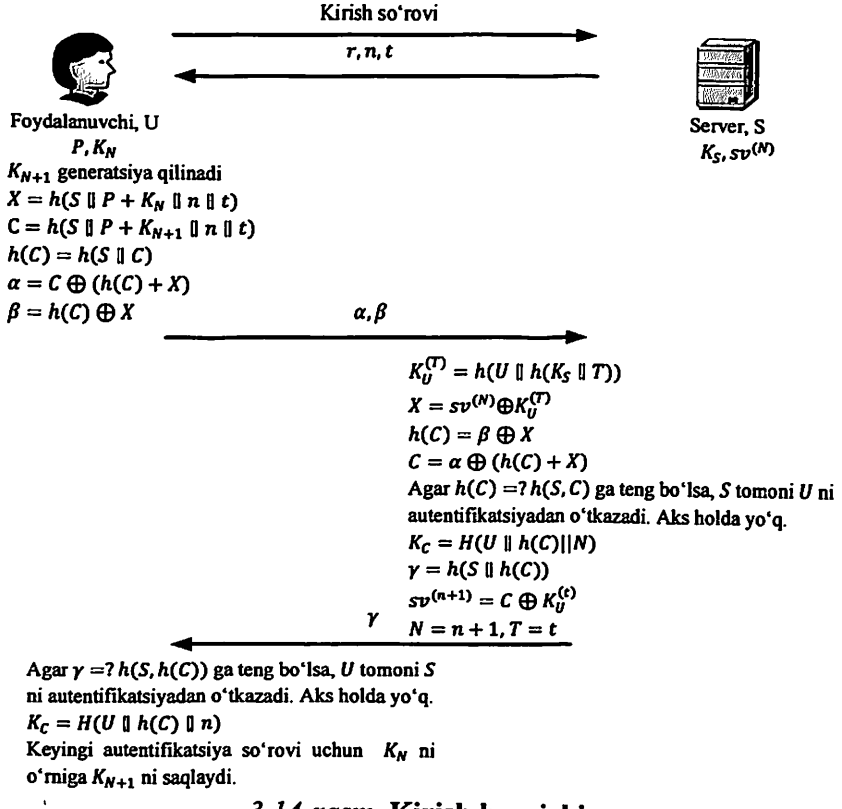
U ning keyingi kirishi uchun $sv^{(N)}$ ni $sv^{(n+1)}$ bilan almashtiradi va $N = n + 1$ kabi o'rnatadi. T ning qiymati o'zgarmas saqlanadi, ya'ni, $T = t$.

5-cўpoв. $U \longrightarrow S$: Autentifikatsiyadan o'tganlik yoki o'ta olmaganligi haqidagi so'rovni yuboradi.

Agar $\gamma =? h(S, h(C))$ ga teng bo'lsa, U tomoni S ni autentifikatsiyadan o'tkazadi va sessiya kaliti K_C ni quyidagicha hisoblaydi:

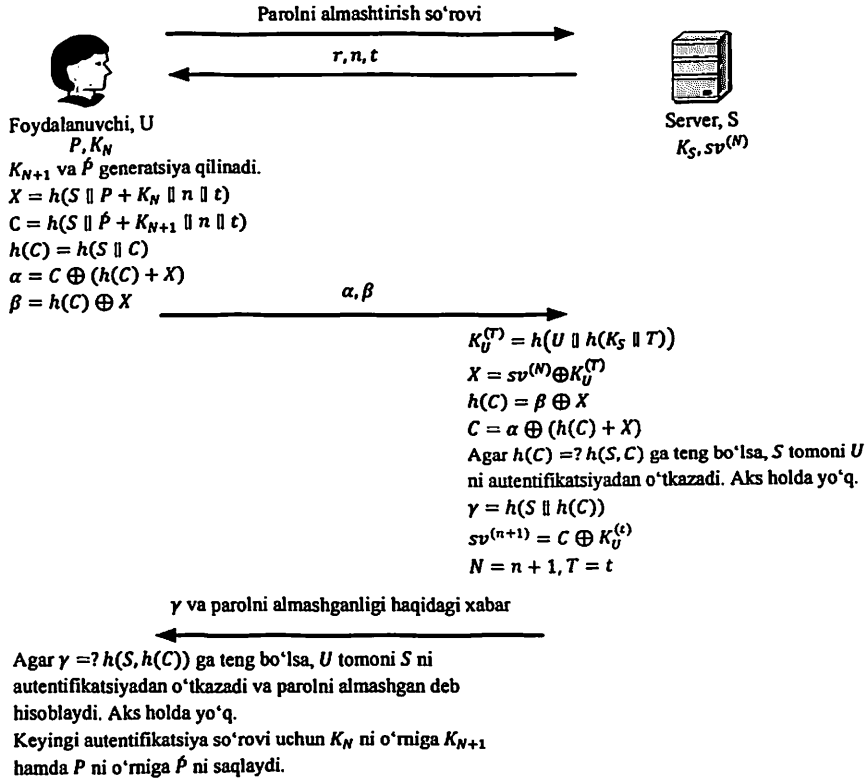
$$K_C = H(U \parallel h(C) \parallel n).$$

Shundan so'ng, keyingi autentifikatsiya so'rovi uchun K_N ni o'rniga K_{N+1} ni saqlaydi.



3.14-rasm. **Kirish bosqichi**

Parolni almashtirish. U talabi bilan parolni almashtirish quyidagicha amalga oshiriladi (3.15-rasm).



3.15-rasm. **Parolni almashtirish bosqichi**

1-cўpoв. $U \longrightarrow S$: Parolni almashtirish so'rovi.

2-cўpoв. $S \longrightarrow U: r, n, t.$

S tomoni bir martali parol r ni generatsiya qiladi va parollar faylidan olingan $n = N$ va $t = T$ larga qo'shib U ga yuboradi.

3-cўpoв. $U \longrightarrow S: \alpha, \beta.$

U tomoni yangi parol $\acute{P}$ va K_{N+1} generatsiya qiladi va quyidagilarni hisoblaydi:

$$X = h(S \parallel P + K_N \parallel n \parallel t)$$
$$C = h(S \parallel \acute{P} + K_{N+1} \parallel n \parallel t)$$
$$h(C) = h(S, C)$$
$$\alpha = C \oplus (h(C) + X)$$
$$\beta = h(C) \oplus X$$

S ga α, β larni yuboradi.

4-сўров. $S \longrightarrow U$: γ va parolni almashganligi haqidagi xabar.

S tomoni $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ ni hisoblaydi va u orqali saqlangan $sv^{(N)}$ dan X ni ajratib oladi, ya'ni:

$$X = sv^{(N)} \oplus K_U^{(T)}.$$

Shundan so'ng, S tomoni quyidagilarni hisoblaydi:

$$h(C) = \beta \oplus X$$
$$C = \alpha \oplus (h(C) + X)$$

Mazkur holda $h(C) =? h(S, C)$ ga teng bo'lsa, S tomoni U ni autentifikatsiyadan o'tkazadi va parolni almashgan deb hisoblaydi. Aks holda U autentifikatsiyadan o'ta olmaydi va parol o'zgarishsiz qoladi.

U autentifikatsiyadan o'tganidan so'ng, foydalanuvchi uni haqiqiyligini tekshirish uchun quyidagini hisoblaydi va U ga yuboradi:

$$\gamma = h(S \parallel h(C)).$$

Shundan so'ng esa, S tomoni keyingi autentifikatsiya so'rovi uchun tasdiqlovchi kattalikni hisoblaydi:

$$sv^{(n+1)} = C \oplus K_U^{(t)}.$$

va U ning keyingi kirishi uchun $sv^{(N)}$ ni $sv^{(n+1)}$ bilan almashtiradi va $N = 1$ kabi o'rnatadi. T kattalik esa o'zgarmas saqlanadi.

Agar $\gamma =? h(S, h(C))$ ga teng bo'lsa, U tomoni S ni autentifikatsiyadan o'tkazadi va parolni muvaffaqiyatli almashgan deb hisoblaydi.

Shundan so'ng, keyingi autentifikatsiya so'rovi uchun K_N ni o'rniga K_{N+1} ni saqlaydi.

W.C. Ku tomonidan yaratilgan protokolning ikki takomillashtirilgan versiyasi ham ikki tomonlama autentifikatsiyani va seans kalitini almashinish imkoniyatini taqdim etadi. Bundan tashqari, parolni almashtirish imkoniyatini mavjudligi protokoldan amalda foydalanish imkoniyatini yanada kengaytiradi. Ushbu ikki versiyaning xavfsizlik va hisoblash bo'yicha samaradorlik tahlillari keyingi bobda keltirilgan.

IV BOB. AUTENTIFIKATSIYA PROTOKOLLARINING XAVFSIZLIK VA SAMARADORLIK TAHLILI

4.1. Takomillashtirilgan va "savol-javob" mexanizmiga asoslangan autentifikatsiya usullarining tahlili

Ushbu bo'lim 3.1 va 3.2-bo'limlarda keltirilgan bir martali parollarga asoslangan autentifikatsiya protokollarining tahliliga bag'ishlangan.

Takomillashtirilgan TOTP/HOTP algoritmining tahlili. TOTP/HOTP algoritmlariga asoslangan mobil ilovalar amalda keng qo'llaniladi. Ular haqida birinchi bo'limda ma'lumotlar keltirilgan edi. TOTP/HOTP algoritmlari asosida ishlab chiqilgan mobil ilovalarda quyidagi umumiy bo'lgan xavfsizlik muammolari aniqlangan:

1. QR kodni o'g'irlash. Ushbu hujum hujumchi foydalanuvchi kompyuterini boshqarish imkoniga ega bo'lganda joiz bo'lib, web brauzer kesh xotirasi tozalanmagan holda amalga oshiriladi. Google Chrome va qolgan brauzerlarda kesh ma'lumotlari oldindan belgilangan manzillarda saqlanadi. Masalan, Google Chrome brauzerida (66-versiyagacha bo'lgan variantda) URL qatorida *chrome://cache* buyrug'ini yozish bilan barcha kesh xotiradagi ma'lumotlarni ko'rish mumkin. Xususan, ular orasidan QR kod URL manzilini topish uchun *chart?chtr=qr* buyrug'idan foydalanish mumkin. Agar qidirish natijasi muvaffaqiyatli bo'lsa, u holda QR kod mavjud URL manzilga ega bo'lishi mumkin. Bu esa taqsimlangan kalitni qo'lga kiritish imkonini beradi.

Bundan tashqari, ushbu hujumni "yelka orqali qarash" hujumi yoki turli zararli dasturiy vositalar ko'magida ham amalga oshirish mumkin.

Takomillashtirilgan TOTP/HOTP algoritmiga asoslangan mobil ilovada esa ushbu hujumni oldi olingan. Buning uchun foydalanuvchining birinchi autentifikatsiya faktori (parol) asosida hosil qilingan kalit bilan taqsimlangan kalit shifrlanib, QR kod ko'rinishida taqdim qilinadi. Bu esa ma'lum vaqtdan so'ng kesh xotiradan mos bo'lgan URLni topgan taqdirda ham taqsimlangan kalitni bilish imkoniyatini yo'qqa chiqaradi.

2. "Mobil qurilmani o'g'irlash" hujumi. Bir martali parolni generatsiyalashda foydalanilgan taqsimlangan kalit odatda mobil qurilmadagi biror manzilda ochiq tarzda saqlanadi. Qurilmani *root*

rejimida ishga tushirib, ma'lum dasturlar (masalan, *adbd Insecure*) yordamida kalit saqlangan ma'lumot bazasini ajratish mumkin:

adb pull
/data/data/com.google.android.apps.authenticator2/databases/
databases
sqlite3 ./database
*select * from accounts*

Ushbu muammoni oldini olish uchun TOTP/HOTP algoritmlari asosida ishlab chiqilgan mobil ilovada taqsimlangan kalitni xavfsiz manzilda saqlash amalga oshirilgan. Masalan, Android OTda kriptografik kalitlarni xavfsiz saqlash imkonini beruvchi AndroidKeyStore [94], Java Keystores [68] kabi imkoniyatlardan foydalanib ularni osonlik bilan amalga oshirish mumkin.

SMS xabar yordamida OTPni yuborishga asoslangan autentifikatsiya usulining tahlili. Taqdim etilgan autentifikatsiya usuliga ko'ra SMS xabar "nuqtadan-nuqtaga" tarzida shifrlanadi. Buning uchun server va mijoz (mobil ilova) taqsimlangan OTPni shifrlash kalitiga ega bo'lishi shart. Mazkur jarayon ro'yxatdan o'tish bosqichida amalga oshiriladi.

Umumiy holda taqdim etilgan usul asosida quyidagi hujumlarni oldini olish imkoniyati mavjud bo'ladi:

– SMS xabarni tutib olish, tinglash va qo'lga kiritish hujumi;

– generatsiya qilingan OTP ni qo'lga kiritishni maqsad qilgan qo'pol kuch hujumi;

– taqsimlangan kalitni topishga qaratilgan hujumlar;

– SIM kartani qalbakilashtirishga asoslangan hujumlar.

"Savol-javob" mexanizmiga asoslangan bir martali parol yordamida autentifikatsiya usulining tahlili. Taklif etilgan bir tomonlama autentifikatsiya protokoli TOTP/HOTP protokollariga analog bo'lib, ular kabi "savol-javob" mexanizmiga asoslanadi. Mazkur protokolda mobil qurilma xavfsiz token vazifasini bajaradi. Bunda, ro'yxatdan o'tish jarayonida taqsimlangan kalit xavfsiz ravishda mijoz mobil ilovasiga yetkaziladi. Taqsimlangan kalit asosida, har bir sessiyasida server tomonidan web serverga taqdim etilgan QR kod ko'rinishidagi bir martali parol (savol) uchun A – funksiya yordamida "javob" hisoblanadi.

Taqdim etilgan ushbu autentifikatsiya protokoli quyidagi hujumlarga bardoshli:

Takrorlash hujumi. Har bir seans uchun takrorlanmas *Nonce* generatsiya qilingani sababli, ushbu hujumning oldi olingan. *Nonce* sifatida 2.3-bo'limda keltirilgan bir martali parollarni generatsiyalash algoritmlaridan foydalaniladi.

Qalbakilashtirish hujumi. Berilgan "savol"ga javob berish faqat mos taqsimlangan kalitni bilgan tomon uchun o'rinli bo'lgani bois, hujumchi biror *Nonce* uchun to'g'ri javobni hisoblay olmaydi.

O'rtaga turgan odam hujumi. Ushbu hujumni amalga oshirish uchun hujumchidan web saytni to'liq qalbakilashtirish talab qilinadi. Boshqacha aytganda, hujumchi foydalanuvchini birinchi faktori bo'yicha muvaffaqiyatli autentifikatsiyasini qo'lga kiritishi talab qilinadi. Ushbu hujumning samarasi talab qilingan "javob"ni topishga qaraganda yuqori bo'lgani sababli, ushbu hujum amalga oshirilishi nazarda tutilmagan.

Taqsimlangan kalitni bilishga qaratilgan hujumlar. Ushbu hujumda hujumchi foydalanuvchi mobil qurilmasida saqlangan taqsimlangan kalitni bilishga harakat qiladi. Yuqorida keltirilgan kabi, taqsimlangan kalitlar mobil ilovada belgilangan xavfsizlik sharoitida saqlanadi va shuning uchun ushbu hujumning bo'lish ehtimolini pasaytiradi.

4.2. Bir martali parollarga asoslangan xavfsiz autentifikatsiya protokolining tahlili

Ushbu bo'limda 3.3-bo'limda keltirilgan bir martali parolga asoslangan xavfsiz autentifikatsiya protokollarining tahlil natijalari keltirilgan.

W.C. Ku usulining modifikatsiyalangan ko'rinishining tahlili.

"Xizmat ko'rsatishdan vos kechishga undash (Denial of Service)" hujumi. Ushbu hujum ma'lumotni o'zgartirish orqali haqiqiy foydalanuvchi uchun keyingi sessiyani amalga oshira olmasligini maqsad qiladi. Ya'ni, mazkur holatda hujumchi c_2 ni o'zgartirishga harakat qiladi. Biroq, taqdim etilgan protokolda u_2 ni hisoblashda foydalanilgan c_1 va c_2 ning butunligi c_3 yordamida himoyalangan. Shuning uchun, c_1, c_2 va c_3 larga bo'lgan ruxsat etilmagan ixtiyoriy o'zgarish S tomonidan osonlik bilan aniqlanadi. Boshqacha aytganda, hujumchi U ning qayd yozuvini bloklay olmagani sababli, modifikatsiyalangan protokol "xizmat ko'rsatishdan voz kechishga undash" hujumiga bardoshli hisoblanadi.

"Qalbakilashtirish" hujumi. Modifikatsiyalangan autentifikatsiya protokolida qalbakilashtirish hujumini amalga oshirish uchun hujumchidan berilgan n va r sonlari uchun autentifikatsiya xabarlarini generatsiyalash talab etiladi. Hujumchi P, K_N va $h(S \parallel P + K_N \parallel n \parallel t)$ bilmasdan turib, S tomonidan qabul qilinishi kerak bo'lgan $\{c_1, c_2, c_3\}$ qiymatlarni taqdim qila olmaydi. Shuning uchun, modifikatsiyalangan usul qalbakilashtirish hujumiga bardoshli hisoblanadi.

"O'rtada turgan odam" hujumi. Ushbu hujumda hujumchi foydalanuvchi va server o'rtasida joylashib, ular tomonidan uzatilgan xabarni o'qish va o'zgartirish orqali autentifikatsiyadan o'tishga harakat qiladi. Autentifikatsiyadan o'tishdagi 3-so'rovda c_1 ning o'zgarishi server tomonidan c_2 dan mos bo'lgan $h(S \parallel P + K_N \parallel n \parallel t)$ qiymatni tiklay olmaslikga sabab bo'lsa, c_2 ning o'zgarishi $h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t)$ qiymatni to'g'ri hisoblay olmaslikka sabab bo'ladi. Mazkur holatda S tomoni $h(h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \parallel r)$ ni c_3 ga tengligini tekshirish uchun $h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t)$ ni tiklay olmagini sababli, autentifikatsiya so'rovini rad etadi. Boshqacha aytganda, c_1 va c_2 ni bir vaqtda o'zgartirish uchun hujumchidan $h(S \parallel P + K_N \parallel n \parallel t)$ va $h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t)$ ni bilishi talab etiladi. Bu esa o'z navbatida hujumchidan foydalanuvchining paroli P va bir marta foydalanuvchi qiymat K_{N+1} ni bilish zaruriyatini qo'yadi. Shuning uchun modifikatsiyalangan protokolni "o'rtada turgan odam" hujumiga bardoshli deb aytish mumkin.

Takrorlash hujumi. Faraz qilinsin, $N = n$ ga va hujumchi $i = 1, 2, \dots, n-1$ uchun foydalanuvchining barcha autentifikatsiya ma'lumotlari, $\{c_1^{(i)}, c_2^{(I)}, c_3^{(i)}\}$ ni tutib olgan bo'lsin. Foydalanuvchi U ning server S da saqlangan joriy verifikatori $h^2(S \parallel P + K_N \parallel n \parallel t)$ ga tengligi sababli, hujumchi $\{c_1^{(i)} = h^2(S \parallel P + K_N \parallel i \parallel t) \oplus h(S \parallel P + K_i \parallel i \parallel t), c_2^{(i)}, c_3^{(i)}\}$, $1 \le i \le n-1$ dan foydalangan holda server S da autentifikatsiyadan o'ta olmaydi. Alternativ holatda, agar hujumchi uzatiluvchi $c_2^{(n)}$ ва $c_3^{(n)}$ larni $c_2^{(i)}$ ва $c_3^{(i)}$ bilan almashtirsa (bu yerda $i = 1, 2, \dots, n-1$), foydalanuvchi U ning tizimga kirishdagi urinishi davomida S tomoni $h((c_2^{(i)} \oplus \left(c_1^{(n)} \oplus h^2(S \parallel P \oplus K_n \parallel n \parallel t)\right)) \oplus r^{(n)})$ qiymat $c_3^{(i)} = h(h^2(S \parallel P \oplus K_{n+1} \parallel n + 1 \parallel t) \parallel r^{(i)}))$ ga teng bo'lmagani bois, xavfni aniqlaydi. Bundan tashqari, agar hujumchi server S ni aldab foydalanuvchi U ning verifikatori $h^2(S \parallel P + K_n \parallel n \parallel$

t) ni $h^2(S \parallel P + K_i \parallel i \parallel t)$ bilan (bu yerda $1 \leq i \leq n-1$) almashtirgan taqdirda ham, $r^{(n)} \neq r^{(i)}$ va buning natijasida $h((c_2^{(i)} \oplus \left(c_1^{(n)} \oplus h^2(S \parallel P + K_n \parallel n \parallel t)\right)) \oplus r^{(n)}) \neq c_3^{(i)}$ uzatilgani bois hujumchi foydalanuvchini obro'sizlantira olmaydi. Shu sababli, modifikatsiyalangan protokolni takrorlash hujumiga bardoshli deb aytish mumkin.

"Faraz qilish bo'yicha" hujum. Ushbu hujum foydalanuvchi yoki server tomonidan uzatilgan ma'lumotlardan foydalanuvchi parolini bilishga harakat qiladi. Mazkur hujum usuli aksariyat statik parollarga asoslangan autentifikatsiyalash protokollari uchun o'rinli hisoblanadi. Masalan, $X = h(S \parallel P \parallel n \parallel t)$ ko'rinishdagi tenglik orqali hisoblangan qiymat, X asosida eng keng tarqalgan parollar lug'atidan foydalangan holda P parolni topish ehtimoli mavjud. Ushbu holatning matematik tomondan murakkabligini quyidagicha hisoblash mumkin.

Misol 1. Faraz qilinsin, olingan parol uzunligi 8 ga teng va foydalanilgan alifbo 128 ta belgidan iborat. Bu holda jami parollarning soni $128^8 = 2^{56}$ ga teng bo'ladi va buzg'unchi eng keng tarqalgan 2^{20} ta paroldan iborat lug'atga ega bo'lsin. Bundan tashqari, ixtiyoriy berilgan parolni buzg'unchining parollar lug'atidan topilish ehtimoli ¼ ga teng bo'lsin. Mazkur holda, buzg'unchining bajaradigan ishi xeshlashlar soni bilan o'lchansin.

Holat 1. Agar foydalanuvchi parollar lug'atidan foydalanmasa, u holda buzg'unchi o'rtacha $\frac{2^{56}}{2} = 2^{55}$ hisoblashni amalga oshirishi talab etiladi.

Holat 2. Agar foydalanuvchi paroli ¼ ehtimollik bilan buzg'unchining lug'atidan topilishi mumkin bo'lsa, u holda buzg'unchi o'zidagi parol lug'atini yarmini, 2^{19} ta xeshlash amalini hisoblaganidan keyin uni aniqlashi mumkin bo'ladi.

Faraz qilinsin, hujumchi tomonidan foydalanuvchi U ning barcha autentifikatsiya ma'lumotlari: c_1, c_2, c_3 tutib olingan bo'lsin. Bu holatda hattoki hujumchi U ning parolini $c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$ tenglikdan foydalanib bilishga harakat qilgan taqdirda ham tasodifiy bir martali qiymat K_N ni bilmasdan buni amalga oshira olmaydi. Shuning uchun modifikatsiyalangan protokol, offlayn holatdagi faraz qilish bo'yicha hujumga bardoshli hisoblanadi.

"O'g'irlangan verifikator" hujumi. Faraz qilinsin, S ning parollar fayli hujumchi tomonidan qo'lga kiritildi. Ya'ni, foydalanuvchi U ga

tegishli $T(=t)$, $N(=n)$ va verifikator $sv^{(N)} = h^2(S \parallel P + K_N \parallel N \parallel T) \oplus K_U^{(T)}$ hujumchiga ma'lum. Bu holda K_S asosida hosil qilingan $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ kattalik hujumchiga ma'lum bo'lgan holdagina $sv^{(N)}$ dan $h^2(S \parallel P + K_N \parallel N \parallel T)$ ni ajratish mumkin bo'ladi. Protokolni ishlab chiqishda qo'yilgan farazga ko'ra K_S kalit qattiq himoyalangan va shuning uchun modifikatsiyalangan protokolni o'g'irlangan verifikator hujumiga bardoshli deb o'ylash mumkin. Boshqacha aytganda, agar tizim tomonidan maxfiy tutilgan K_S obro'sizlantirilsa, u holda butun tizim protokolni buzilishiga qaraganda jiddiy hujum ostida qolishi mumkin.

Ikki tomonlama autentifikatsiyani amalga oshirish. W.C.Ku usulining taklif etilgan modifikatsiyalangan ko'rinishida ikkinchi tomonni, ya'ni, serverni ham autentifikatsiyadan o'tkazish imkoniyati mavjud. Buning uchun server tomoni $c_4 = h(U \parallel u_2)$ ni yuboradi. Bu yerda, $u_2 = c_2 \oplus u_1 = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t)$ va $u_1 = c_1 \oplus h^2(S \parallel P + K_N \parallel n \parallel t) = h(S \parallel P + K_N \parallel n \parallel t)$ ga teng, U – esa foydalanuvchi nomi. Foydalanuvchi U tomonidan qabul qilingan c_4 asosida serverning haqiqiyligini $h(U \parallel h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) = ? c_4$ tenglikni tekshirish bilan aniqlaydi. Bu yerda serverni qalbakilashtirib bo'lmaslik holatining oldi c_4 ni hisoblash uchun u_2 ni talab etilishi bilan xarakterlanadi.

Seans kalitini hisoblash. W.C.Ku tomonidan taklif etilgan autentifikatsiya protokoli faqat bir tomonlama autentifikatsiyani ta'minlash bilan chegaralangan. Takomillashtirilgan variantida esa nafaqat serverni ham haqiqiyligini tekshirish, balki keyingi xavfsizlik amallari uchun zarur bo'lgan seans kalitini hosil qilish imkoniyati ham mavjud. Seans kalitini hosil qilish server S tomonida $K_C = H(U \parallel u_1)$ shaklida hosil qilinsa, foydalanuvchi U esa $K_C = H(U \parallel h(S \parallel P + K_N \parallel n \parallel t)$ tenglikdan foydalanadi. Bu holda ikkita tenglikning bir xil ekanligini quyidagicha isbotlash mumkin ($c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$ ga tengligini inobatga olib):

$$u_1 = c_1 \oplus h^2(S \parallel P + K_N \parallel n \parallel t) = h(S \parallel P + K_N \parallel n \parallel t)$$

Seans kalitini hosil qilishda foydalanilgan $H()$ - funksiya sifatida talab etilgan kalit uzunligiga mos bo'lgan ixtiyoriy bardoshli bir tomonlama funksiyalardan (masalan, SHA-256, SHA-384, SHA-512 va hak.) foydalanish mumkin.

Seans kalitini generatsiyalash ifodasida kiruvchi parametrda ikkita o'zgaruvchi, K_N va n lar har bir seansda o'zgargani bois, hosil bo'luvchi qiymatlarning ham takrorlanish darajasi past bo'ladi. Bundan tashqari, foydalanilgan bir tomonlama funksiya kalitning statistik tomondan tasodifiy bo'lishini ta'minlaydi.

W.C.Ku usulining SAS-2 protokoli ko'rinishidagi shaklining tahlili.

"Xizmat ko'rsatishdan vos kechishga undash (Denial of Service)" hujumi. Xizmat ko'rsatishdan voz kechishga undash hujumi odatda yuborilgan ma'lumot o'zgartirilsa va bazada saqlangan vaqtda amalga oshgan hisoblanadi. Natijada, qonuniy foydalanuvchi keyingi seansda tizimdan muvaffaqqiyatli autentifikatsiyadan o'ta olmaydi. Faraz qilinsin, 3-so'rovda yuborilgan ikki xabar quyidagicha o'zgartirilgan bo'lsin:

$$\alpha' = c \oplus (h(c) + x)$$
$$\beta' = h(c) \oplus x$$

Ushbu ikki xabar server S tomonida qabul qilingandan so'ng $h(C) = \beta \oplus X$ hisoblanadi. Bu yerda, X kattalik server tomonda mavjud bo'lib, faqat β' va α' kattaliklarda foydalanilgan bo'lsagina keyingi sessiya uchun bazada saqlanishi mumkin. Bundan tashqari, c kattalik α' da foydalanilgani bois hujumchi β' uchun biror $y \oplus A$ ni yubora olmaydi (bu yerda, y va A hujumchi tomonidan tasodifiy tanlangan qiymatlar). Shuning sababli, xizmat ko'rsatishdan voz kechishga undash hujumini ham mazkur holatda amalga oshirishning imkoni mavjud emas.

"Qalbakilashtirish" hujumi. Mazkur hujumda foydalanuvchi tomonidan uzatilgan α, β larni hisoblash talab qilinadi. Bu esa o'z navbatida hujumchidan berilgan n va r sonlari uchun autentifikatsiya xabarlarini generatsiyalashni talab etadi. Buning uchun hujumchi P va K_N kattaliklarni bilishi talab qilinadi. Bundan tashqari, agar hujumchi X kattalikdan foydalanish asosida buni amalga oshirmoqchi bo'lsa, undan $K_U^{(T)}$ maxfiy kattalikni bilishi talab qilinadi. Bu esa, takomillashtirilgan usulni qalbakilashtirish hujumiga bardoshliligini ko'rsatadi.

"O'rtada turgan odam" hujumi. Agar hujumchi foydalanuvchi U ni obro'sizlantirishni istasa, u holda quyidagini yuboradi:

$$\alpha' = c \oplus (h(c) + x)$$
$$\beta' = h(c) \oplus x$$

Shundan so'ng, foydalanuvchi β' uchun c va x larni tanlaydi. Bu asosida α' ni ham hosil qilishi mumkin. Biroq α' va β' lar server tomonidan qabul qilinganda, haqiqiy X kattalik asosida ularni qalbakilashtirilganini osonlik bilan aniqlash mumkin bo'ladi. Bundan tashqari, agar β' ni o'zi qalbakilashtirilsa, uni α bilan tekshirish mumkin bo'ladi. Shu sababli, mazkur protokolni "o'rtada turgan odam" hujumiga bardoshli deb aytish mumkin.

Takrorlash hujumi. Faraz qilinsin, hujumchi $i-$seansgacha bo'lgan barcha α_n va β_n qiymatlariga ega bo'lsin. Bu yerda, $1 \leq n \leq i-1$ tenglik o'rinli. Hujumchi $i-$ seans uchun zarur bo'lgan α_i va β_i o'rniga α_{i-2} va β_{i-2} qiymatlarni yuborgan bo'lsin. Server tomonidan foydalanuvchini autentifikatsiyalash uchun $i-$seansda $X_i = h(S \parallel P + K_i \parallel n_i \parallel t)$ foydalaniladi va shuning uchun α_{i-2} va β_{i-2} qiymatlar uchun to'g'ri yechim bo'la olmaydi. Bu holni α_n va β_n larning boshqa qiymatlari uchun ham ko'rish mumkin. Boshqacha aytganda, har bir seans uchun yangi X qiymatlarning hosil bo'lishi natijasida, takrorlash hujumining oldi olingan.

"Faraz qilish bo'yicha" hujum. Mazkur hujumga bardoshlik tahlili yuqorida keltirilgan bo'lib, SAS-2 sxemasi uchun ham o'rinli bo'ladi.

"O'g'irlangan verifikator" hujumi. Ro'yxatdan o'tishda foydalanuvchi tomonidan uzatilgan X qiymat $K_U^{(T)}$ kalit yordamida XOR amalida qo'shilib, foydalanuvchini to'g'riligini tasdiqlovchi $sv^{(N)}$ sifatida ($sv^{(N)} = X \oplus K_U^{(T)}$) serverdagi parollar faylida saqlanadi. Agar ushbu parollar fayli buzg'unchining qo'liga tushib qolganda ham, undan foydalanish uchun $K_U^{(T)}$ kalit talab qilinadi. Bu esa o'z navbatida serverning maxfiy kaliti K_S ni bilishni talab qiladi. Protokolni ishlab chiqishda qo'yilgan shartlar va farazlarga ko'ra, buning imkoni yo'q. Shuning uchun takomillashtirilgan protokol "o'g'irlangan verifikator" hujumiga bardoshli hisoblanadi.

Ikki tomonlama autentifikatsiyani amalga oshirish. Taqdim etilgan SAS-2 ko'rinishidagi protokolda ham serverni haqiqiyligini tekshirish imkoniyati mavjud bo'lib, foydalanuvchi tomonda $\gamma = ?h(S, h(C))$ tenglikni tekshirish orqali amalga oshiriladi.

Seans kalitini hisoblash. Original SAS-2 protokolidan farqli ravishda takomillashtirilgan shaklida seans kalitini taqsimlash imkoniyati mavjud bo'lib, bu kalit server va foydalanuvchi autentifikatsiya jarayoni muvaffaqiyatli tugagandan so'ng $K_C = H(U \| h(C)||N)$ tenglik orqali uni hisoblashi mumkin bo'ladi.

Har ikkala holat uchun samaradorlik va xavfsizlik talablari uchun olingan natijalar quyidagi 4.1 va 4.2-jadvallarda aks ettirilgan.

4.1-jadval

Takomillashtirilgan bir martali parollarga asoslangan autentifikatsiya usullarining xavfsizlik omillari bo'yicha tahlili

	1-talab	2-talab	3-talab	4-talab	5-talab	6-talab	7-talab	8-talab
Tak. 1.	+	+	+	+	+	+	+	O'
Tak. 2.	+	+	+	+	+	+	+	Y

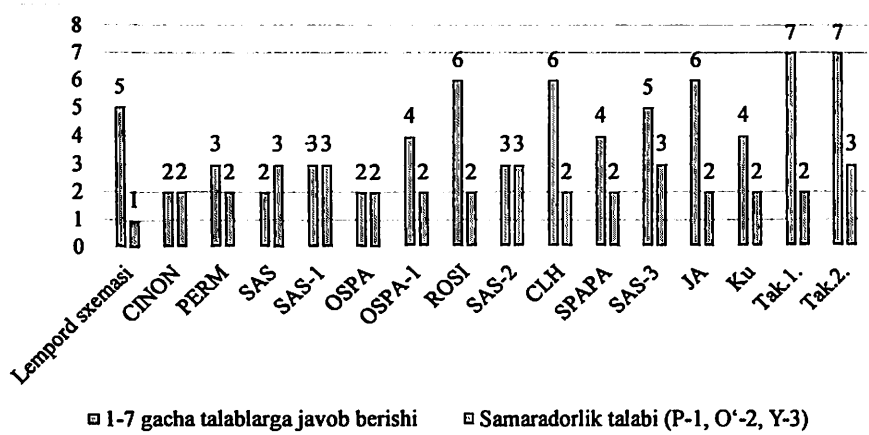
4.2-jadval

Takomillashtirilgan bir martali parollarga asoslangan autentifikatsiya usullarining samaradorlik omillari bo'yicha tahlili

	Serverda		Mijozda		Mijoz→ Server	
	Xeshlashlar soni (marta)	Saqlanuvchi ma'lumotlar	Xeshlashlar soni (marta)	Saqlanuvchi ma'lumotlar	Uzatilishlar soni	Uzatiluvchi ma'lumot hajmi
Tak. 1.	4 (*mutual* 6)	$sv^{(N)}, T, N, K_S$	6 (*mutual* 7)	K_N	1	$L(log.req) + 3L(H)$
Tak. 2.	3 (*mutual* 6)	$sv^{(N)}, T, N, K_S$	3 (*mutual* 4)	K_N	1	$L(ID) + 2L(H)$

Bundan tashqari, mavjud bir martali parollarga asoslangan autentifikatsiya usullar va takomillashtirilgan variantlarning xavfsizlik va samaradorlik bo'yicha tahlil natijalari 4.1-rasmda aks ettirilgan. Takomillashtirilgan birinchi ko'rinishdagi protokol samaradorlik nuqtayi nazaridan bir xil (mijozdan serverga uzatilgan ma'lumot hajmi

$L(log.req) + 3L(H)$) bo'lsada, ikki tomonlama autentifikatsiya va seans kalitini almashinish imkoniyati mavjud. Boshqa tomondan, ikkinchi variantdagi takomillashtirilgan protokolda esa mijozdan serverga uzatilgan ma'lumot hajmi $L(ID) + 2L(H)$) yoki boshqacha aytganda 33,3% kamroq ma'lumotni uzatishni talab qilgan.



4.1-rasm. **Mavjud va takomillashtirilgan algoritmlarning tahlili**

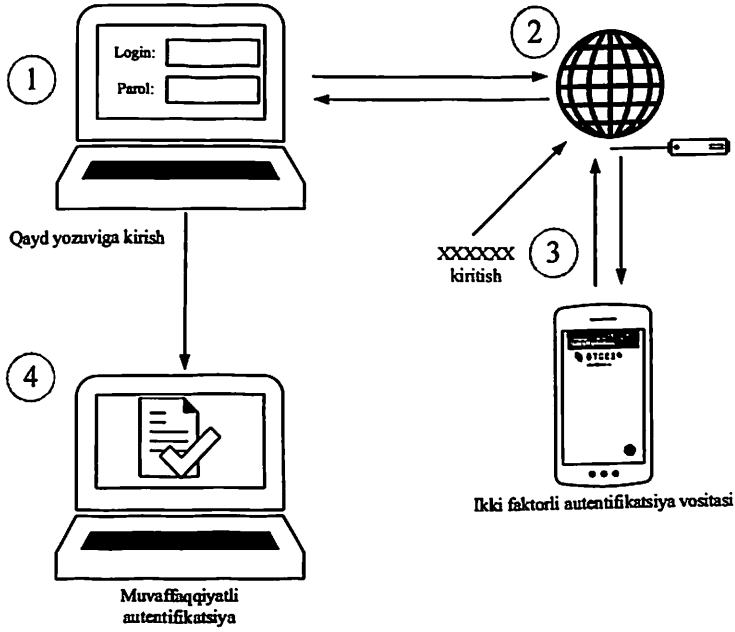
Bundan tashqari, taqdim etilgan har ikkala takomillashtirilgan protokollar versiyasida ham original ko'rinishlaridan farqli o'laroq, parollarni almashtirish imkoniyati mavjud. Ushbu imkoniyat foydalanuvchining paroli zaif deb topilgan taqdirda yoki xavfsizlik siyosatiga mos holda amalga oshirilishi mumkin.

4.3. Taklif etilgan autentifikatsiya protokollarining amalda tatbiqi

Ishlab chiqilgan va takomillashtirilgan bir martali parollarga asoslangan autentifikatsiya protokollari mobil ilova ko'rinishidagi dasturiy vosita shaklida amalga oshirildi. Quyida ularning har biri bilan batafsil tanishib chiqiladi.

Takomillashtirilgan TOTP/HOTP protokoli asosidagi dasturiy ta'minot. Taqdim etilgan protokol asosida Android OT uchun mobil

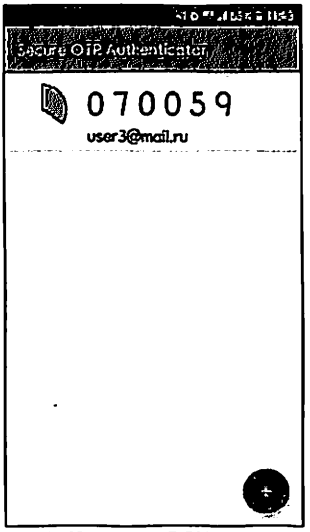
ilova ishlab chiqilgan bo'lib, tizimning umumiy ko'rinishi quyidagi 4.2-rasmda keltirilgan.



4.2-rasm. **Takomillashtirilgan TOTP protokoli asosida autentifikatsiyalash sxemasi**

Ishlab chiqilgan mobil dasturiy ta'minotda taqsimlangan kalitni shifrlashda AES algoritmidan foydalanilgan bo'lsa, TOTP algoritmini hosil qilishda HMAC-SHA1 algoritmidan foydalanilgan.

Foydalanuvchi dastlab biror tizimdan birinchi faktor bo'yicha ro'yxatdan o'tadi va zarur bo'lgan mobil ilovani o'rnatadi. Ro'yxatdan o'tish davomida foydalanuvchining birinchi autentifikatsiya omillari asosida shifrlangan ko'rinishdagi taqsimlangan kaliti QR kod shaklida taqdim etiladi. Mobil ilovada dastlab QR kod o'qib olinadi va foydalanuvchi tomonidan kiritilgan birinchi faktor parametrlari asosida deshifrlanadi. Shundan so'ng, taqsimlangan kalit mobil ilovaning belgilangan manzilda saqlanadi va joriy vaqt asosida bir martali parollarni generatsiyalash boshlanadi (4.3-rasm).

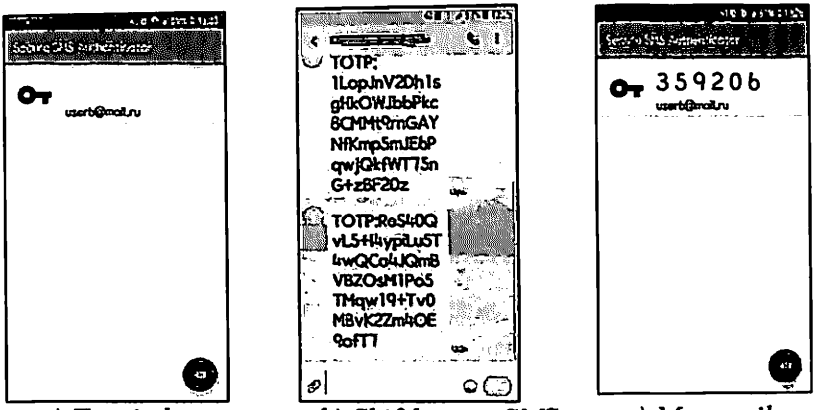


4.3-rasm. **Takomillashtirilgan TOTP protokoli asosidagi mobil ilova dasturi**

Har bir generatsiya qilingan OTPlar 30 sekund davomida foydalanish uchun yaroqli bo'ladi. Mazkur dasturiy vosita bir vaqtning o'zida bir nechta akkauntlar uchun OTPlarni generatsiya qilib beradi. Ushbu dasturiy vositalardan umumiy holda ixtiyoriy ijtimoiy tarmoqlarda ikkinchi faktor sifatida foydalanish mumkin.

SMS xabar yordamida OTPni yuborishga asoslangan usulning dasturiy ta'minoti. Serverdan yuborilgan shifrlangan ko'rinishdagi SMS ni o'quvchi va uni deshifrlangan holda saqlab qo'yish uchun Android OTda maxsus mobil dasturiy ilova yozilgan. Ilovada dastlab server bilan taqsimlangan kalit o'qib olinadi va shundan so'ng qabul qilingan shifrlangan ko'rinishdagi SMS xabarni deshifrlash imkoniyati taqdim etiladi. 4.4-rasmda ushbu holatlar to'liq aks ettirilgan.

Har safar tizimga kirish amalga oshirilganda yangi OTP yuboriladi. Qabul qilingan shifrlangan ko'rinishdagi SMS xabar oddiy SMS ilovasida shifrlangan ko'rinishda taqdim qilinsa, ishlab chiqilgan maxsus mobil dasturiy ilovada esa u ochiq ko'rinishda bo'ladi. Mazkur usul SMS xabarni mobil aloqani ta'minlashdagi SS7 protokolidagi zaiflik va ruxsat etilmagan o'qish kabi xavfsizlik muammolarini oldini oladi.



a) Taqsimlangan kalitga ega bo'lgan holat

b) Shifrlangan SMS xabarni SMS ilovasida ko'rinishi

c) Maxsus ilovada OTP ni ko'rinishi

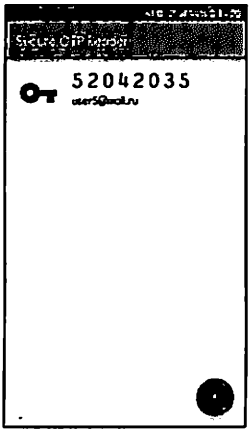
4.4-rasm. **OTPni shifrlangan ko'rinishda uzatish usulining umumiy bosqichlari**

Dasturiy ta'minotda taqsimlangan kalitlar yuqorida keltirilgan kabi xavfsiz manzilda saqlanadi va bu turli zararli dasturlar tomonidan qo'lga kiritilishidan himoyalaydi. Ishlab chiqilgan dasturiy ta'minotda taqsimlangan kalit va SMS xabarni shifrlashda AES shifrlash algoritmidan foydalanilgan.

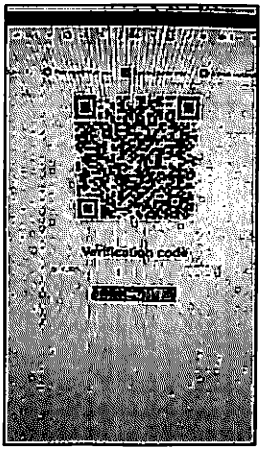
Taqdim etilgan usuldan hozirda SMS xabarga asoslangan ikki faktorli autentifikatsiya usullarining o'rniga foydalanish mumkin. Bu oddiy holatga nisbatan mavjud xavfsizlik muammolarini bartaraf etishga imkon beradi.

"Savol-javob" mexanizmiga asoslangan bir martali parol yordamida autentifikatsiya usulining dasturiy ta'minoti. QR kod ko'rinishida taqdim qilingan OTP ("savol")ni o'qib, u asosida tegishli "javob"ni hisoblashga asoslangan protokolning Android OT uchun mobil ilovasi ishlab chiqilgan bo'lib, yuqorida dasturlardagi kabi taqsimlangan kalitga ega bo'linadi. Shundan so'ng, har bir kirishda birinchi autentifikatsiya faktoridan o'tilgandan so'ng foydalanuvchiga QR kod ko'rinishida OTP taqdim etiladi. Foydalanuvchi mobil ilovadan foydalangan holda QR kodni o'qib oladi va ilova avtomatik ravishda mos "javob"ni foydalanuvchiga taqdim etadi. Agar tizimga kiritilgan "javob" server tomonidan maqbul deb topilgan taqdirda, foydalanuvchi tizimga kirish huquqiga ega bo'ladi. Umumiy holda mobil ilovaning ko'rinishi quyidagi 4.5-rasmda keltirilgan.

a) Taqdim etilgan QR kod ko'rinishidagi OTP



b) Maxsus dastur yordamida hisoblangan "javob"



s) "Javob"ni tizimga kiritish jarayoni

4.5-rasm. **"Savol-javob" mexanizmiga asoslangan bir martali parol yordamida autentifikatsiya usulining dasturiy ta'minoti**

Ishlab chiqilgan mobil ilovada taqsimlangan kalitni shifrlashda AES algoritmidan foydalanilgan bo'lsa, "savol-javob" mexanizmi esa HMAC-SHA1 asosida amalga oshirilgan. Taqdim etilgan autentifikatsiya usulidan ijtimoiy tarmoqlarda ikkinchi faktor sifati foydalanish mumkin bo'lib, yuqorida keltirilgan HOTP/TOTP algoritmiga asoslangan dasturiy mobil ilovalarga analog hisoblanadi.

XULOSA

Ushbu monografiyada olib borilgan tadqiqotlar natijasida quyidagi natijalarga erishildi:

Bardoshli simmetrik blokli shifrlash algoritmlari asosida psevdotasodifiy sonlar generatori ishlab chiqildi. Ishlab chiqilgan psevdotasodifiy sonlar generatori yuqori tasodifiylik darajasini ko'rsatdi.

Ishlab chiqilgan psevdotasodifiy sonlar generatorlari asosida irratsional sonlar xususiyatidan foydalangan holda bir martali parollarni hosil qilish usuli va algoritmi ishlab chiqildi. Ishlab chiqilgan generator 85,4% to'liq takrorlanmaslik darajasiga ega 6 xona uzunlikdagi parollarni hosil qilish imkonini berdi.

Taqsimlangan kalitlarni xavfsiz uzatish va xavfsiz saqlash orqali TOTP/HOTP algoritmi asosida ishlab chiqilgan autentifikatsiya usuli va algoritmi takomillashtirildi. Takomillashtirilgan usul turli zararli dasturiy vositalar yordamida amalga oshiriladigan hujumlarga bardoshlikni oshirish imkonini berdi.

Bir martali parollarni shifrlash orqali SMS xabar yordamida yetkazish usuli takomillashtirildi. Takomillashgan usul ruxsat etilmagan tutib olish va SS7 protokolidagi zaiflik natijasida amalga oshiriluvchi tahdiddan himoyalash imkonini berdi.

Bir martali parolni QR kod ko'rinishida taqdim etish orqali yetkazishga asoslangan autentifikatsiya usuli va algoritmi ishlab chiqildi. Ishlab chiqilgan usul bir martali parol asosida autentifikatsiya usullariga qaratilgan hujumlarni oldini olish va qulay foydalanish imkonini berdi.

W.C. Ku tomonidan taqdim etilgan bir martali parolga asoslangan autentifikatsiya protokoli yangi tasodifiy kattalikni kiritish va ikki tomonlama autentifikatsiyalash imkoniyatini yaratish orqali takomillashtirildi. Takomillashtirilgan usul aslida mavjud bo'lgan takrorlash hujumi va parolni faraz qilish hujumini oldini olish imkonini bergan.

FOYDALANILGAN ADABIYOTLAR

1. Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей : Учеб. пособие. М.: ФОРУМ: ИНФРА-М, 2017. 416 с.

2. Alzomai M.H. Identity management: strengthening one-time password authentication through usability: dis. Queensland University of Technology, 2011.

3. Голенко Д.И. Моделирование и статистический анализ псевдослучайных чисел на электронных вычислительных машинах. М.: Наука, Главная редакция физико-математической литературы, 1965.

4. Grassi P., Garcia M., Fenton J. DRAFT NIST Special Publication 800-63-3 Digital Identity Guidelines //National Institute of Standards and Technology. Los Altos, CA. 2017.

5. Арзиева Ж.Т. Об одном методе удостоверения подлинности пользователей в инфокоммуникационных системах // Республиканский семинар: Информационные технологии и проблемы телекоммуникаций. Т., 2012. С. 196-197.

6. Syed Zulkarnain Syed Idrus, Estelle Cherrier, Christophe Rosenberger, Jean-Jacques Schwartzmann. A Review on Authentication Methods. Australian Journal of Basic and Applied Sciences. 2013. 7 (5). P.95-107.

7. Каримов М.М., Арзиева Ж.Т. Метод аутентификации объектов инфо-коммуникационных систем // ILMIY xabarnoma. Т., 2011. №4. Р. 17-18.

8. Utewliev N., Arzieva J.T. Direct search methods to solve stochastic optimization problem // International conference on IT Promotion in Asia 2009. T., 2009. P. 101-105.

9. M.M. Karimov, K.A. Tashev, J.T. Arziyeva, A.A. Abdurakhmonov, A.T. Imamaliyev. About one of the authentication methods // TUIT BULLETIN. T., 2013. №3. P. 5-12.

10. Alliance S.C. Fips 201 and physical access control: An overview of the impact of fips 201 on federal physical access control systems. 2005.

11. E. De Cristofaro et al. A comparative usability study of two-factor authentication //arXiv preprint arXiv:1309.5344-2013.

12. Каримов М.М., Худойқулов З.Т., Арзиева Ж.Т. Атаки, направленные на методы аутентификации по паролю // Muhammad al-Xorazmiy avlodlari. T., 2019. №4(10). P. 156-159.
13. Gühring P. Concepts against man-in-the-browser attacks. 2006.
14. Каримов М.М., Худойқулов З.Т., Арзиева Ж.Т. Анализ метода аутентификации на основе одноразовых паролей // Muhammad al-Xorazmiy avlodlari. T., 2019. №4(10). P. 3-6.
15. Arziyeva J.T. OTP asosida bir yo'nalishli autentifikatsiya protokollarining tahlili // Respublika ilmiy-texnik konferensiya: "Axborot texnologiyalari va kommunikatsiyalari sohasida axborot xavfsizligi muammolari". T., 2019. P.63-67.
16. Lamport L. Password Authentication with Insecure Communication. In: Comm. ACM. Vol. 24. No 11. 1981. P. 770-772.
17. Takasuke TSUJI. A One-Time Password Authentication Method. Master's thesis. January 31. 2003.
18. Sandirigama M., Shimizu A., Noda M.T. Simple and secure password authentication protocol (SAS) // IEICE Transactions on Communications. 2000. T. 83. №. 6. C. 1363-1365.
19. Lin C.L., Sun H.M., Hwang T. Attacks and solutions on strong-password authentication // IEICE transactions on communications. 2001. T. 84. №. 9. C. 2622-2627.
20. Kamioka T. The examination of the security of SAS one-time password authentication // IEICE Technical Report. 2001. T. 435. C. 53-58.
21. T.Tsuji, T.Kamioka, and A.Shimizu. Simple and secure password authentication protocol, ver.2 (SAS-2). IEICE Technical Report, OIS2002-30. Vol.102. No.314. September 2002.
22. Jo H.S., Youn H.Y. A secure user authentication protocol based on one-time-password for home network // International Conference on Computational Science and Its Applications. Springer, Berlin, Heidelberg, 2005. C. 519-528.
23. Haller Bellcore N. The S/KEY One-Time Password System. Network Working Group, February. 1995.
24. Chwei-Shyong, Tsai & Cheng-Chi, Lee & Hwang, Min-Shiang. (2006). Password Authentication Schemes: Current Status and Key Issues. International Journal of Network Security.

25. Jan M.S., Afzal M. Hash chain based strong password authentication scheme //2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST). IEEE, 2016. C. 355-360.

26. Liao I.E., Lee C.C., Hwang M.S. A password authentication scheme over insecure networks //Journal of Computer and System Sciences. 2006. T. 72. №. 4. C. 727-740.

27. Каримов М.М., Худойкулов З.Т., Арзиева Ж.Т. Атаки, направленные на методы аутентификации по паролю // Muhammad al-Xorazmiy avlodlari. T., 2019. №4(10). P. 156-159.

28. Karimov M.M., Arziyeva J.T. Bir martali parollarga asoslangan autentifikatsiyada mavjud muammolar // Respublika ilmiy-texnika anjumani: "Iqtisodiyotning tarmoqlarini innovatsion rivojlanishida axborot-kommunikatsiya texnologiyalarining ahamiyati". T., 2019. P. 190-192.

29. Shimizu A. A dynamic password authentication method using a one-way function //Systems and computers in Japan. 1991. T. 22. №. 7. C. 32-40.

30. Chen L., Mitchell C.J. Comments on the S/KEY user authentication scheme //ACM Operating Systems Review. 1996. T. 30. №. 4. C. 12-16.

31. Shimizu A., Horioka T., Inagaki H. A password authentication method for contents communications on the Internet //IEICE transactions on communications. 1998. T. 81. №. 8. C. 1666-1673.

32. Mangipudi K.V., Katti R.S. A Hash-based Strong Password Authentication Protocol with User Anonymity //IJ Network Security. 2006. T. 2. №. 3. C. 205-209.

33. Weragama N.S., Sandirigama M. SAS-3: A polynomial based strong password authentication protocol //2007 International Conference on Industrial and Information Systems. IEEE, 2007. C. 41-46.

34. Chien H.Y., Jan J.K. Robust and simple authentication protocol//The computer journal. 2003. T. 46. №. 2. C. 193-201.

35. Chen C.M., Ku W.C. Stolen-verifier attack on two new strong-password authentication protocols //IEICE Transactions on communications. 2002. T. 85. №. 11. C. 2519-2521.

36. Tsuji T., Shimizu A. An impersonation attack on one-time password authentication protocol OSPA //IEICE Transactions on Communications. 2003. T. 86. №. 7. C. 2182-2185.

37. Yoon E.J., Ryu E.K., Yoo K.Y. Fixing problems in Lin et al.'s OSPA protocol //Applied mathematics and computation. 2005. T. 166. №. 1. C. 46-57.

38. Lin C.W., Shen J.J., Hwang M. S. Security enhancement for optimal strong-password authentication protocol //ACM SIGOPS Operating Systems Review. 2003. T. 37. №. 2. C. 7-12.

39. Ku W.C., Tsai H.C., Chen S.M. Two simple attacks on Lin-Shen-Hwang's strong-password authentication protocol //ACM SIGOPS Operating Systems Review. 2003. T. 37. №. 4. C. 26-31.

40. Ku W.C., Tsai H.C., Tsaur M.J. A Common Weakness of Password Authentication Schemes Requiring Synchronous Update of Stored Data //Proceedings of the 2004 International Computer Symposium, Taiwan. 2004. C. 849-852.

41. Tsuji T., Kamioka T., Shimizu A. Simple and secure password authentication protocol, ver. 2 (SAS-2) //ITE Technical Report 26.61. The Institute of Image Information and Television Engineers, 2002. C. 7-11.

42. Arziyeva J.T. Autentifikatsiya usullarining tahlili // Respublika ilmiy-texnik anjumani: "Axborot-kommunikatsiya texnologiyalari va telekommunikatsiyalarning zamonaviy muammolari va yechimlari". Farg'ona, 2019. P. 343-345.

43. Mitchell C.J., Ng S.L. Comments on the security of the SPAPA strong password authentication protocol. 2007.

44. Chiang M.H., Ku W.C. Weaknesses of two SAS-like password authentication schemes //IEICE transactions on communications. 2006. T. 89. №. 2. C. 594-597.

45. Chen T.H., Lee W.B., Horng G. Secure SAS-like password authentication schemes //Computer Standards & Interfaces. 2004. T. 27. №. 1. C. 25-31.

46. Mangipudi K.V., Katti R.S. A Hash-based Strong Password Authentication Protocol with User Anonymity //IJ Network Security. 2006. T. 2. №. 3. C. 205-209.

47. Ku W.C. A hash-based strong-password authentication scheme without using smart cards //ACM SIGOPS Operating Systems Review. 2004. T. 38. №. 1. C. 29-34.

48. Kim M., Koç C.K. A Simple Attack on a Recently Introduced Hash-based Strong-password Authentication Scheme //IJ Network Security. 2005. T. 1. №. 2. C. 77-80.

49. Kumar M. On the security vulnerabilities of a hash based strong password authentication scheme //Organization. 2009. T. 4. C. 9.

50. Xudoyqulov Z.T., Arziyeva J.T., Ortiqboyev A.M. Bir martali parol generatorlarining tahlili // Axborotkommunikatsiyalar: Tarmoqlar, Texnologiyalar, Yechimlar.T., 2019. №3(51). P. 48-55.

51. Ташев К.А., Арзиева Ж.Т., Насруллаев Н.Б. Анализ протокола аутентификации объектов инфокоммуникационных систем // Международная конференция "Актуальные проблемы развития инфокоммуникаций и информационно общества". Т., 2012. С. 718-723.

52. Арзиева Ж.Т., Шыхыев Р.М. Аутентификация в системах с разделением времени // Актуальные проблемы прикладной математики и информационных технологий. Аль-Хорезми 2016. Бухара, Узбекистан, 2016. P. 147-149.

53. Dmitrienko A. et al. Security analysis of mobile two-factor authentication schemes //Intel Technology Journal. 2014.T. 18. №.4.

54. Al Maqbali F., Mitchell C.J. AutoPass: An automatic password generator //2017 International Carnahan Conference on Security Technology (ICCST). IEEE, 2017. C. 1-6.

55. Halderman J.A., Waters B., Felten E.W. A convenient method for securely managing passwords //Proceedings of the 14th international conference on World Wide Web. ACM, 2005. C. 471-479.

56. Арзиева Ж.Т. Об одном способе применения генераторов паролей в задачах аутентификации пользователей // ТУИТ вестник. Т., 2012. №2. P. 23-27.

57. Leonhard M, Venkatakrishnan V.N. A comparative study of three random passwords generators. Electro/ Information Technology, IEEE International conference. 2007. P. 227-6.

58. Jagannatam A. Mersenne Twister–A Pseudo Random Number Generator and its Variants //George Mason University, Department of Electrical and Computer Engineering. 2008.

59. Каримов М.М., Арзиева Ж.Т., Худойкулов З.Т. Выбор, соответствующих генераторов псевдослучайных чисел для генераторов одноразовых паролей // Международно-научно-практическая конференция: «Наука в современном мире (Science in the modern world)». Душанбе, 2019. С. 21-31.

60. Knuth D.E. Art of computer programming, volume 2: Semi numerical algorithms. Addison-Wesley Professional, 2014.

61. X.D.C. De Carnavalet et al. From Very Weak to Very Strong: Analyzing Password-Strength Meters //NDSS. 2014. T. 14. C. 23-26.

62. Barker E., Kelsey J. NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators. 2012.

63. Ganiev S.K., Khudoykulov Z.T., Halimtaeva I.U. Computer's source based (Pseudo) Random Number Generation. International Conference on Information Science and Communications Technologies ICISCT. 2017. C.1-6.

64. Арзиева Ж.Т., Утепбергенова Г. Тестирование получаемых псевдослучайных чисел в системах аутентификации // Республиканская научно-практическая конференция по смежным вопросам математической физики и современного анализа. Бухара, 2015. С. 317-318.

65. M.E.O'Neill. PCG: A family of simple fast space-efficient statistically good algorithms for random number generation. *ACM Trans. Math. Softw.* 2014.

66. M.M. Karimov, K.A. Tashev, S.S. Kim, A.R. Ishmuratov, J.T. Arzieva. The authentication method based on the random number generator // International Journal of Ubiquitous Computing and Internationalization. 2011. Vol.3. No.2. P. 35-40.

67. Cooijmans, T., de Ruiter, J., & Poll, E. Analysis of secure key storage solutions on Android. In Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices. 2014. November. P. 11-20. ACM.

68. R. Focardi, F. Palmarini, M. Squarcina, G. Steel, & M. Tempesta. Mind Your Keys? A Security Evaluation of Java Keystores. In NDSS. 2018. February.

69. PPP Authentication Protocols [sayt]: https://tools.ietf.org/html/rfc1334 (murojaat vaqti: 25.02.2021).

70. PPP Challenge Handshake Authentication Protocol (CHAP) [sayt]: https://www.ietf.org/rfc/rfc1994.txt (murojaat vaqti: 25.02.2021).

71. Rainbow jadvallar ro'yxati [sayt]: http://project-rainbowcrack.com/table.htm (murojaat vaqti: 25.02.2021).

72. Password Security Report [sayt]: https://www.cyclonis.com/report-83-percent-users-surveyed-use-same-password-multiple-sites/ (murojaat vaqti: 25.02.2021).

73. The S/KEY One-Time Password System [sayt]: https://tools.ietf.org/html/rfc1760 (murojaat vaqti: 25.02.2021).

74. Hackers use keyloggers to steal the information from 3 million Zoho users [sayt]: https://blog.360totalsecurity.com/en/hackers-use-keyloggers-to-steal-the-information-from-3-million-zoho-users/ (murojaat vaqti: 25.02.2021).

75. Hackers infected thousands of PCs with Zeus trojan to steal millions [sayt]: http://news.techworld.com/security/3241594/police-arrest-gang-behind-20-million-online-bank-fraud/ (murojaat vaqti: 25.02.2021).

76. Top Phishing Email Attacks Worldwide in 2018 [sayt]: https://www.duocircle.com/phishing-protection/top-phishing-email-attacks-worldwide-in-2018 (murojaat vaqti: 25.02.2021).

77. Top Phishing Email Attacks Worldwide in 2018 [sayt]: https://www.adaware.com/blog/50-banks-in-pharming-attack (murojaat vaqti: 25.02.2021).

78. Man-in-the-browser [sayt]: https://en.wikipedia.org/wiki/Man-in-the-browser (murojaat vaqti: 25.02.2021).

79. "123456" tops list of most common passwords for 2020 [sayt]: https://www.techrepublic.com/article/123456-tops-list-of-most-common-passwords-for-2020/ (murojaat vaqti: 25.02.2021).

80. RainbowCrack [sayt]: http://project-rainbowcrack.com/ (murojaat vaqti: 25.02.2021).

81. Aircrack-ng [sayt]: http://www.aircrack-ng.org/ (murojaat vaqti: 25.02.2021).

82. John the Ripper password cracker [sayt]: http://www.openwall.com/john/ (murojaat vaqti: 25.02.2021).

83. RSA SecurID® Access [sayt]: https://www.rsa.com/en-us/products/rsa-securid-suite/rsa-securid-access/securid-hardware-tokens (murojaat vaqti: 25.02.2021).

84. HOTP: An HMAC-Based One-Time Password Algorithm [sayt]: https://tools.ietf.org/html/rfc4226 (murojaat vaqti: 25.02.2021).

85. TOTP: Time-Based One-Time Password Algorithm [sayt]: https://tools.ietf.org/html/rfc6238 (murojaat vaqti: 25.02.2021).

86. SS7 NETWORK SECURITY ANALYSIS REPORT [sayt]: https://positive-tech.com/knowledge-base/research/ss7-network-security-analysis-2020/ (murojaat vaqti: 25.02.2021).

87. Email-Based OTP [sayt]: https://www.microcosm.com/products/smartsign/authentication-methods/email-otp (murojaat vaqti: 25.02.2021).

88. Images Could Change the Authentication Picture [sayt]: https://www.darkreading.com/risk/images-could-change-the-authentication-picture/d/d-id/1134705 (murojaat vaqti: 25.02.2021).
89. Confident Multi-Factor-Authentication Demo [sayt]: http://confidenttechnologies.com/confident-multi-factor-authentication-demo/ (murojaat vaqti: 25.02.2021).
90. Safe online banking [sayt]: https://www.commerzbank.de/portal/en/englisch/products-offers/services/secure-internet-banking/banking.html (murojaat vaqti: 25.02.2021).
91. The Password Meter [sayt]: http://www.passwordmeter.com/ (murojaat vaqti: 25.02.2021).
92. TOTP: (way) more secure than SMS, but more annoying than Push [sayt]: https://www.allthingsauth.com/2018/04/05/totp-way-more-secure-than-sms-but-more-annoying-than-push/ (murojaat vaqti: 25.02.2021).
93. Time-based One-time Password algorithm [sayt]: https://en.wikipedia.org/wiki/Time-based_One-time_Password_algorithm (murojaat vaqti: 25.02.2021).
94. Android keystore system [sayt]: https://developer.android.com/training/articles/keystore#java (murojaat vaqti: 25.02.2021).
95. SMS-based two-factor authentication is not safe — consider these alternative 2FA methods instead [sayt]: https://www.kaspersky.com/blog/2fa-practical-guide/24219/ (murojaat vaqti: 25.02.2021).
96. Random String Generator [sayt]: https://www.random.org/strings/ (murojaat vaqti: 25.02.2021).
97. PWGen [sayt]: http://pwgen-win.sourceforge.net/ (murojaat vaqti: 25.02.2021).

ILOVALAR

1-ilova

Parol sifatida foydalanish uchun afzal ko'riladigan ma'lumotlar

№	Ma'lumot turi	Ulushu, %
1.	Foydalanuvchiga yoqqan ism (nom)	34,07
2.	Bir turdagi belgilar kombinatsiyasi (masalan, 12345, 111 va h.)	33,70
3.	Keng tarqalgan so'zlardan foydalangan holda ulardan keyin "1" yoki "!" ni qo'shib qo'yish	29,30
4.	Qarindoshlar ismi	27,47
5.	Tug'ulgan kun	22,71
6.	Eng yaqin sinfdosh ismi	17,58
7.	Yoqtirgan sport jamoasi nomi	17,58
8.	Brauzerlarni parollarni boshqarish imkonidan foydalanish	16,85
9.	Mashhur atletchi ismi yoki raqamidan foydalanish	14,65
10.	Maktabni, kollejni tugatgandan yoki turmush qurgandan hozirgacha bo'lgan vaqt (yil)	13,92
11.	Boshqa ma'lumotlar	19,41

2-ilova

Hujum turlari va ularga qarshi himoya usullari

№	Hujum turlari	Ularga qarshi himoya usullari
	Qayd yozuvini boshqarishga qaratilgan hujumlar	
1.	Zararli dasturlarga asoslangan hujumlar	Zararli dasturlarga qarshi himoyani o'rnatish, OTPdan foydalanish
2.	"Spufing" (spoofing) hujumlari	Statik bo'lmagan autentifikatsiya (OTP)dan foydalanish
3.	"O'rtada turgan odam" hujumi	Shifrlangan aloqa kanallaridan foydalanish
4.	Takrorlash hujumi	Shifrlash va tasodifiy qiymat/ vaqt metkasidan foydalanish
5.	"O'rtada turgan brauzer" hujumi	Autentifikatsiyani tranzaksiya sathida amalga oshirish
6.	"Xizmat ko'rsatishdan voz kechishga undash" hujumi	Kompleks chora-tadbirlar
	Parolni topishga qaratilgan hujumlar	
1.	Zaif kriptografik tizimdan foydalanish	Bardoshli kriptografik xesh funksiyalardan foydalanish
2.	Faraz qilish asosida hujum	OTP dan foydalanish
3.	Lug'atga asoslangan hujum	Avtomatik bloklash, muvaffaqiyatsiz urinishlar sonini cheklash, "tuz"dan foydalanish
4.	Qo'pol kuch hujumi	Ko'p faktorli autentifikatsiyadan, CAPTCHA tizimlaridan, "ishonchli manzil" texnologiyasidan foydalanish
5.	Oldindan hisoblashga asoslangan hujumlar	OTP, ko'p faktorli autentifikatsiya yoki "tuz"dan foydalanish
6.	"Yelka ortidan qarash" hujumi	Grafik parollardan, OTPdan foydalanish

3-ilova

OTPlarni yetkazish usullarining tahlili

№	OTPni yetkazish usuli	Kamchiligi	Afzalligi	Xavfsizlik muammosi	Mavjud hujumlar	Misollar
1	2	3	4	5	6	7
1.	Mobil telefon orqali (SMS)	Mobil qurilma talab etadi	Foydalanish-ga qulay, past narx	Mobil qurilma tizimining xavfsiz emasligi, SMS ni uzatish muhitini xavfsiz emasligi.	Qurilmani fizik nazoratlash, mobayl tarmoqni tinglash, zararli dasturlar asosida hujumlar, "yelka ortidan qarash" hujumi	Telegram, WhatsApp
2.	Shaxsiy tokenlar (qurilma ko'rinishida)	Doim birga olib yurish, yuqori narx, cheklangan ishlash muddati	Foydalanish-ga qulay, yuqori xavfsizlik	Doimo xavfsiz saqlash va olib yurish	O'g'irlatish yoki yo'qotib qo'yish orqali xavfsizlikni to'liq ta'minlanmasligi, "yelka ortidan qarash" hujumi	SecurID
3.	Shaxsiy tokenlar (dasturiy ko'rinishda)	Mobil qurilma talab etadi	O'rtacha narx, cheklanma-gan ishlash muddati	Mobil qurilma tizimining xavfsiz emasligi	Zararli dasturlar asosida hujumlar, spufing hujumlari, "yelka ortidan qarash" hujumi	Google Authenticat or
4.	Webga asoslangan usullar	Esda saqlash zaruriyati	Ortiqcha qurilma talab etmasligi,	Mobil qurilma tizimining xavfsiz emasligi	Spufing hujumlari, "yelka ortidan qarash" hujumi	Confident Multi-Factor-

1	2	3	4	5	6	7
			past narx	(foydalanilgan bo‘lsa)		Authentication [89]
5.	“Qog‘ozga” yozib olish	Doim birga olib yurish	Ortiqcha qurilma talab etmaydi, past narx	Doimo xavfsiz saqlash va olib yurish	Fishing, “o‘rtada turgan odam” hujumi, o‘g‘irlatish, “yelka ortidan qarash” hujumi	Klassik TAN
6.	Pochta orqali yuborish	Pochta tizimi parolini bilish	Ortiqcha qurilma talab etmaydi, past narx	Pochta tizimi qayd yozuvini doim xavfsiz saqlanmasligi	Spufing hujumlari, “o‘rtada turgan odam” hujumi, “yelka ortidan qarash” hujumi	SmartSign [87]

4-ilova

Turli parollarni to'liq tanlash usulida topish uchun sarflangan vaqt

	(0...9) dan iborat 10 belgi uzunligidagi (5689743664)	**(0...9, a...z, A...Z) dan iborat 10 ta belgi (j63o1f9Avu)**	**(0...9, a...z, A...Z, ?,/,~,!,(,),| ...) dan iborat 10 ta belgi (c?Kxar/XM7)**
Standart ShK	2 min	3 000 yil	208 000 yil
Tezkor ShK	25 sek	67 yil	52 000 yil
GPU	10 sek	27 yil	21 000 yil
Tezkor GPU	5 sek	13 yil	10 000 yil
Parallel GPU lar	1 sek	1 yil	87 yil
O'rtacha sonli botlar	<1 sek	2 soat	6 kun
Bardoshlik (100 dan)	26 %	55%	62%

NIST SPECIAL PUBLICATION 800-22 statistik testlar to'plamidan olingan natijalar

CryptGenRandom 1

monobit_test	0.061540729218848234 PASS
frequency_within_block_test	0.45509594185633667 PASS
runs_test	0.6475642341504244 PASS
longest_run_ones_in_a_block_test	0.3528109578164638 PASS
binary_matrix_rank_test	0.33330219975911546 PASS
dft_test	0.19436591080310917 PASS
non_overlapping_template_matching_test	0.9993490178308497 PASS
overlapping_template_matching_test	0.005173739045216826 FAIL
maurers_universal_test	0.9992418718038792 PASS
linear_complexity_test	0.46511039917819896 PASS
serial_test	0.04052468521717285 PASS
approximate_entropy_test	0.05452573789558518 PASS
cumulative_sums_test	0.09574808337565521 PASS
random_excursion_test	0.5970248063880195 PASS
random_excursion_variant_test	0.3542918290328953 PASS

CryptGenRandom 2

monobit_test	0.30521937621459216 PASS
frequency_within_block_test	0.267939486040662 PASS
runs_test	0.4214289744608908 PASS
longest_run_ones_in_a_block_test	0.16721113509729008 PASS
binary_matrix_rank_test	0.703159743935546 PASS
dft_test	0.6731887478154436 PASS
non_overlapping_template_matching_test	0.9999999917577034 PASS
overlapping_template_matching_test	0.6579898571968237 PASS
maurers_universal_test	0.9996331710302226 PASS
linear_complexity_test	0.37484272127811236 PASS
serial_test	0.20618807009725704 PASS
approximate_entropy_test	0.20637200966389013 PASS
cumulative_sums_test	0.20476087477134453 PASS
random_excursion_test	0.08239858278413338 PASS
random_excursion_variant_test	0.06716037254224258 PASS

CryptGenRandom 3

monobit_test	0.8496973365380595 PASS
frequency_within_block_test	0.7123307913023494 PASS
runs_test	0.004866735862014582 FAIL
longest_run_ones_in_a_block_test	0.6799682364842367 PASS
binary_matrix_rank_test	0.08870585294291476 PASS
dft_test	0.3636458223803358 PASS
non_overlapping_template_matching_test	0.9999996343966502 PASS
overlapping_template_matching_test	0.05552985823706506 PASS
maurers_universal_test	0.999682383907673 PASS
linear_complexity_test	0.4329754450743705 PASS
serial_test	0.024307114235540935 PASS
approximate_entropy_test	0.02381926648853401 PASS
cumulative_sums_test	0.41905243152809213 PASS
random_excursion_test	0.13612149997909762 PASS
random_excursion_variant_test	0.08166982597207062 PASS

CryptGenRandom 4

monobit_test	0.33978297435581895 PASS
frequency_within_block_test	0.9396711245310864 PASS
runs_test	0.40689287895724013 PASS
longest_run_ones_in_a_block_test	0.8842902498806724 PASS
binary_matrix_rank_test	0.7120992710639026 PASS
dft_test	0.14251711752569796 PASS
non_overlapping_template_matching_test	1.0000965761102598 PASS
overlapping_template_matching_test	0.7949571523320188 PASS
maurers_universal_test	0.9993779816178165 PASS

```
linear_complexity_test                    0.9232694447344479 PASS
serial_test                               0.035022629545118616 PASS
approximate_entropy_test                  0.11461538766655344 PASS
cumulative_sums_test                      0.310838663585292  PASS
random_excursion_test                     0.004412328783076444 FAIL
random_excursion_variant_test             0.0                FAIL
                          CryptGenRandom 5
monobit_test                              0.9402518535555923 PASS
frequency_within_block_test               0.602068849853921  PASS
runs_test                                 0.4201819021700258 PASS
longest_run_ones_in_a_block_test          0.35570879991220244 PASS
binary_matrix_rank_test                   0.7528813156020931 PASS
dft_test                                  0.02691832269283094 PASS
non_overlapping_template_matching_test    1.000000004410003  PASS
overlapping_template_matching_test        0.3825997905015421 PASS
maurers_universal_test                    0.9992897900273794 PASS
linear_complexity_test                    0.16078427886240693 PASS
serial_test                               0.7339437725385058 PASS
approximate_entropy_test                  0.9380924265642778 PASS
cumulative_sums_test                      0.5834973443296709 PASS
random_excursion_test                     0.06415602413012561 PASS
random_excursion_variant_test             0.14148028355875486 PASS
                          /dev/urandom 1
monobit_test                              0.8013829984038171 PASS
frequency_within_block_test               0.3742752931031935 PASS
runs_test                                 0.9343200559795215 PASS
longest_run_ones_in_a_block_test          0.8825602333880895 PASS
binary_matrix_rank_test                   0.8393701289574338 PASS
dft_test                                  0.28712374326829326 PASS
non_overlapping_template_matching_test    0.999991533237183  PASS
overlapping_template_matching_test        0.4608396934193636 PASS
maurers_universal_test                    0.9997402193997992 PASS
linear_complexity_test                    0.6519220294428996 PASS
serial_test                               0.1170215409732085 PASS
approximate_entropy_test                  0.24275067242791418 PASS
cumulative_sums_test                      0.8140509095657333 PASS
random_excursion_test                     0.01753988430386064 PASS
random_excursion_variant_test             0.02891574659831201 PASS
                          /dev/urandom 2
monobit_test                              0.9409548837291478 PASS
frequency_within_block_test               0.6104891411605241 PASS
runs_test                                 0.44960463729983846 PASS
longest_run_ones_in_a_block_test          0.7387869470161018 PASS
binary_matrix_rank_test                   0.49314203763931586 PASS
dft_test                                  0.2406102901827999 PASS
non_overlapping_template_matching_test    1.0004803568809206 PASS
overlapping_template_matching_test        0.680105193721871  PASS
maurers_universal_test                    0.999663104832164  PASS
linear_complexity_test                    0.4011324454633697 PASS
serial_test                               0.6535061630948787 PASS
approximate_entropy_test                  0.8776456522401267 PASS
cumulative_sums_test                      0.6686593932267484 PASS
random_excursion_test                     0.07583759883694292 PASS
random_excursion_variant_test             0.022645540682891915 PASS
                          /dev/urandom 3
monobit_test                              0.025530953983317707 PASS
frequency_within_block_test               0.3908906820727942 PASS
runs_test                                 0.11540790260437706 PASS
longest_run_ones_in_a_block_test          0.19935571644365457 PASS
binary_matrix_rank_test                   0.9580613734880038 PASS
dft_test                                  0.48860117060332464 PASS
non_overlapping_template_matching_test    0.9999975345283598 PASS
overlapping_template_matching_test        0.10752279852598208 PASS
maurers_universal_test                    0.9996872222380958 PASS
```

```
linear_complexity_test                    0.717340918833691  PASS
serial_test                               0.27856318192625246 PASS
approximate_entropy_test                  0.2791179761333817 PASS
cumulative_sums_test                      0.01706818706336577 PASS
random_excursion_test                     0.22857497213742767 PASS
random_excursion_variant_test             0.019553208238583517 PASS
                                /dev/urandom 4
monobit_test                              0.37109336952269767 PASS
frequency_within_block_test               0.057584876938821236 PASS
runs_test                                 0.8939358783159861 PASS
longest_run_ones_in_a_block_test          0.7073557676249743 PASS
binary_matrix_rank_test                   0.7924136945961584 PASS
dft_test                                  0.812457035634175  PASS
non_overlapping_template_matching_test    1.000000045409078  PASS
overlapping_template_matching_test        0.014353207299332028 PASS
maurers_universal_test                    0.9999155713477748 PASS
linear_complexity_test                    0.3848420922163466 PASS
serial_test                               0.2557322549349314 PASS
approximate_entropy_test                  0.2594760482879653 PASS
cumulative_sums_test                      0.33894200498075877 PASS
random_excursion_test                     0.10161494105875651 PASS
random_excursion_variant_test             0.2363157270671385 PASS
                                /dev/urandom 5
monobit_test                              0.6099792984024024 PASS
frequency_within_block_test               0.37345386807404735 PASS
runs_test                                 0.64981476162645   PASS
longest_run_ones_in_a_block_test          0.043226640751716006 PASS
binary_matrix_rank_test                   0.2142586375494195 PASS
dft_test                                  0.1061124964529202 PASS
non_overlapping_template_matching_test    0.999999990144254  PASS
overlapping_template_matching_test        0.8102917255127788 PASS
maurers_universal_test                    0.9993033031155459 PASS
linear_complexity_test                    0.9886280441212981 PASS
serial_test                               0.4515102475226974 PASS
approximate_entropy_test                  0.7644720184220585 PASS
cumulative_sums_test                      0.5408446831930025 PASS
random_excursion_test                     0.04660022144696309 PASS
random_excursion_variant_test             0.07585826061362605 PASS
                                JAVA 1
monobit_test                              0.8942439364709419 PASS
frequency_within_block_test               0.518160875200195  PASS
runs_test                                 0.8964911449591423 PASS
longest_run_ones_in_a_block_test          0.4110636895767209 PASS
binary_matrix_rank_test                   0.45107895536256176 PASS
dft_test                                  0.03609099520422324 PASS
non_overlapping_template_matching_test    1.000015643893229  PASS
overlapping_template_matching_test        0.6246713565955363 PASS
maurers_universal_test                    0.9986312033058913 PASS
linear_complexity_test                    0.5277537481432787 PASS
serial_test                               0.014204039046356491 PASS
approximate_entropy_test                  0.09481320309488134 PASS
cumulative_sums_test                      0.7626327621505542 PASS
random_excursion_test                     0.08258717219433295 PASS
random_excursion_variant_test             0.01351845176089688 PASS
                                JAVA 2
monobit_test                              0.051152225938006946 PASS
frequency_within_block_test               0.18051043062439714 PASS
runs_test                                 0.8142871267669853 PASS
longest_run_ones_in_a_block_test          0.5214391923006133 PASS
binary_matrix_rank_test                   0.7251212459403048 PASS
dft_test                                  0.8052359468183318 PASS
non_overlapping_template_matching_test    0.9999999953331739 PASS
overlapping_template_matching_test        0.41272964918444577 PASS
maurers_universal_test                    0.99834710307316   PASS
```

```
linear_complexity_test                    0.2514506425828426 PASS
serial_test                               0.15275037889524276 PASS
approximate_entropy_test                  0.1535387633479182 PASS
cumulative_sums_test                      0.0636222076551396 PASS
random_excursion_test                     0.06493753499583524 PASS
random_excursion_variant_test             0.19094065395649334 PASS
                                    JAVA 3
monobit_test                              0.8976002309746519 PASS
frequency_within_block_test               0.38653337811938515 PASS
runs_test                                 0.6754946533865663 PASS
longest_run_ones_in_a_block_test          0.684076926265762  PASS
binary_matrix_rank_test                   0.5715483339928238 PASS
dft_test                                  0.5946662523346172 PASS
non_overlapping_template_matching_test    0.999999891756343  PASS
overlapping_template_matching_test        0.9031464345276996 PASS
maurers_universal_test                    0.9983928142169938 PASS
linear_complexity_test                    0.9712617327332375 PASS
serial_test                               0.39765610334632645 PASS
approximate_entropy_test                  0.6155009602342296 PASS
cumulative_sums_test                      0.921986765071739  PASS
random_excursion_test                     0.039894536594631116 PASS
random_excursion_variant_test             0.08080509898094734 PASS
                                    Java 4
monobit_test                              0.41774228018229564 PASS
frequency_within_block_test               0.5664151464473817 PASS
runs_test                                 0.7102845464212569 PASS
longest_run_ones_in_a_block_test          0.6654637778561973 PASS
binary_matrix_rank_test                   0.7815181654602418 PASS
dft_test                                  0.4361783622988469 PASS
non_overlapping_template_matching_test    0.9999999134429278 PASS
overlapping_template_matching_test        0.5790998101420868 PASS
maurers_universal_test                    0.9996462890119561 PASS
linear_complexity_test                    0.640534554142144  PASS
serial_test                               0.3757431033605115 PASS
approximate_entropy_test                  0.3744923439592698 PASS
cumulative_sums_test                      0.48797096718071575 PASS
random_excursion_test                     0.17008615301878408 PASS
random_excursion_variant_test             0.014736188992246296 PASS
                                    JAVA 5
monobit_test                              0.7567793907879514 PASS
frequency_within_block_test               0.5612401856807804 PASS
runs_test                                 0.17059165657859518 PASS
longest_run_ones_in_a_block_test          0.862237974573052  PASS
binary_matrix_rank_test                   0.10881665983752357 PASS
dft_test                                  0.8405777193396354 PASS
non_overlapping_template_matching_test    1.000146190432456  PASS
overlapping_template_matching_test        0.2988996051302883 PASS
maurers_universal_test                    0.9998140346517829 PASS
linear_complexity_test                    0.6440443635370146 PASS
serial_test                               0.10806879636730239 PASS
approximate_entropy_test                  0.26928387650819907 PASS
cumulative_sums_test                      0.7049727916386821 PASS
random_excursion_test                     0.15741870893156196 PASS
random_excursion_variant_test             0.005360562674188974 FAIL
                                    PYTHON 1
monobit_test                              0.5487551485387947 PASS
frequency_within_block_test               0.7643635786213017 PASS
runs_test                                 0.0050594198055186335 FAIL
longest_run_ones_in_a_block_test          0.3452455957669108 PASS
binary_matrix_rank_test                   0.8965181906818852 PASS
dft_test                                  0.7603842243797873 PASS
non_overlapping_template_matching_test    0.9999993777859705 PASS
overlapping_template_matching_test        0.5638945143107165 PASS
maurers_universal_test                    0.9999269022668253 PASS
```

linear_complexity_test	0.9583408426740642 PASS
serial_test	0.06637262870828518 PASS
approximate_entropy_test	0.06655210484355568 PASS
cumulative_sums_test	0.34150300037459913 PASS
random_excursion_test	0.22545422715411953 PASS
random_excursion_variant_test	0.04791324799760892 PASS
PYTHON 2	
monobit_test	0.7492622755103526 PASS
frequency_within_block_test	0.9512832331071555 PASS
runs_test	0.1511860743028433 PASS
longest_run_ones_in_a_block_test	0.5945191016445435 PASS
binary_matrix_rank_test	0.17234592537514534 PASS
dft_test	0.201141976570284 PASS
non_overlapping_template_matching_test	1.0000004244795118 PASS
overlapping_template_matching_test	0.5579749229524441 PASS
maurers_universal_test	0.9996518620583371 PASS
linear_complexity_test	0.0418941404328522 PASS
serial_test	0.7543357731342827 PASS
approximate_entropy_test	0.7548578662420481 PASS
cumulative_sums_test	0.8952013161822883 PASS
random_excursion_test	0.047100799895076075 PASS
random_excursion_variant_test	0.19210299481523294 PASS
PYTHON 3	
monobit_test	0.17683194950763342 PASS
frequency_within_block_test	0.6141044447764846 PASS
runs_test	0.6317291917162263 PASS
longest_run_ones_in_a_block_test	0.6214120277863623 PASS
binary_matrix_rank_test	0.27362966514165304 PASS
dft_test	0.27852485041817376 PASS
non_overlapping_template_matching_test	0.9966891508837562 PASS
overlapping_template_matching_test	0.39740834864038443 PASS
maurers_universal_test	0.9998125282213026 PASS
linear_complexity_test	0.26340122865845356 PASS
serial_test	0.6013683425317707 PASS
approximate_entropy_test	0.7545340575094706 PASS
cumulative_sums_test	0.10213606495171512 PASS
random_excursion_test	0.05731350056475351 PASS
random_excursion_variant_test	0.2220023525510689 PASS
PYTHON 4	
monobit_test	0.6837924318441025 PASS
frequency_within_block_test	0.7264044655689849 PASS
runs_test	0.8619982948994267 PASS
longest_run_ones_in_a_block_test	0.7920345451179281 PASS
binary_matrix_rank_test	0.2950406756700544 PASS
dft_test	0.10756463861441073 PASS
non_overlapping_template_matching_test	0.9997989865915445 PASS
overlapping_template_matching_test	0.3114320471956003 PASS
maurers_universal_test	0.9992389047993402 PASS
linear_complexity_test	0.47749989564510703 PASS
serial_test	0.5396186103389221 PASS
approximate_entropy_test	0.5387999650117913 PASS
cumulative_sums_test	0.4828977603302489 PASS
random_excursion_test	0.028712899684548016 PASS
random_excursion_variant_test	0.554700196225229 PASS
PYTHON 5	
monobit_test	0.17592711779534634 PASS
frequency_within_block_test	0.7356774875517608 PASS
runs_test	0.45445976423128426 PASS
longest_run_ones_in_a_block_test	0.35552498613596545 PASS
binary_matrix_rank_test	0.2969462735936953 PASS
dft_test	0.6127648061387001 PASS
non_overlapping_template_matching_test	1.00000004232126 PASS
overlapping_template_matching_test	0.8915845453588737 PASS
maurers_universal_test	0.9982561125552846 PASS

```
linear_complexity_test                    0.42362740063334386 PASS
serial_test                               0.5464414853971798 PASS
approximate_entropy_test                  0.5856084278379095 PASS
cumulative_sums_test                      0.1831455869199925 PASS
random_excursion_test                     0.27505015889892265 PASS
random_excursion_variant_test             0.06640158940746632 PASS

                          Taklif etilgan PTSG
                               1-na'muna
-------
monobit_test                              0.15768058479781322 PASS
frequency_within_block_test               0.11453130677850276 PASS
runs_test                                 0.9154110583905278 PASS
longest_run_ones_in_a_block_test          0.5400186652862622 PASS
binary_matrix_rank_test                   0.5590371767755881 PASS
dft_test                                  0.5255431399121548 PASS
non_overlapping_template_matching_test    0.9999999942879848 PASS
overlapping_template_matching_test        0.15539891435101494 PASS
maurers_universal_test                    0.99953250196402   PASS
linear_complexity_test                    0.18220914573354957 PASS
serial_test                               0.28699982619419406 PASS
approximate_entropy_test                  0.38906682444016544 PASS
cumulative_sums_test                      0.18461772810457866 PASS
random_excursion_test                     0.10049650410891338 PASS
random_excursion_variant_test             0.0034342570972031275 FAIL
                               2- na'muna
-------
monobit_test                              0.3763484553051467 PASS
frequency_within_block_test               0.36213308469589217 PASS
runs_test                                 0.539893180880937  PASS
longest_run_ones_in_a_block_test          0.8140633236790145 PASS
binary_matrix_rank_test                   0.8687805946576294 PASS
dft_test                                  0.021332977000548845 PASS
non_overlapping_template_matching_test    1.0001363571964823 PASS
overlapping_template_matching_test        0.31485508354779207 PASS
maurers_universal_test                    0.9990727696981969 PASS
linear_complexity_test                    0.6289329506963944 PASS
serial_test                               0.17843075664839758 PASS
approximate_entropy_test                  0.1789962946917316 PASS
cumulative_sums_test                      0.29217495894249446 PASS
random_excursion_test                     0.11305125471596977 PASS
random_excursion_variant_test             0.07801080992678706 PASS
                               3- na'muna
-------
monobit_test                              0.7970646653468292 PASS
frequency_within_block_test               0.7803804124941501 PASS
runs_test                                 0.30699043639190327 PASS
longest_run_ones_in_a_block_test          0.6590453863502214 PASS
binary_matrix_rank_test                   0.7387520552500334 PASS
dft_test                                  0.6034807571193761 PASS
non_overlapping_template_matching_test    0.9999999332568198 PASS
overlapping_template_matching_test        0.7283300102753313 PASS
maurers_universal_test                    0.9993109007764321 PASS
linear_complexity_test                    0.8408587275021514 PASS
serial_test                               0.04844397630798777 PASS
approximate_entropy_test                  0.13571865179403234 PASS
cumulative_sums_test                      0.7276215105301072 PASS
random_excursion_test                     0.28978031284464534 PASS
random_excursion_variant_test             0.010040466534645876 PASS
                               4 - na'muna
monobit_test                              0.705923571822622  PASS
frequency_within_block_test               0.9280679751866501 PASS
runs_test                                 0.6307616080798263 PASS
longest_run_ones_in_a_block_test          0.018079288538172603 PASS
```

```
binary_matrix_rank_test                  0.9580613734880038 PASS
dft_test                                 0.43781031855499863 PASS
non_overlapping_template_matching_test   0.9999922126812182 PASS
overlapping_template_matching_test       0.030782457393763903 PASS
maurers_universal_test                   0.998607916519982  PASS
linear_complexity_test                   0.3229359985800194 PASS
serial_test                              0.041375443760419524 PASS
approximate_entropy_test                 0.1748502940762445 PASS
cumulative_sums_test                     0.6286368592354918 PASS
random_excursion_test                    0.41907854721691395 PASS
random_excursion_variant_test            0.042624646216904594 PASS
                          5 - na'muna
monobit_test                             0.10349584044027604 PASS
frequency_within_block_test              0.9536304581777542 PASS
runs_test                                0.1394990465030953 PASS
longest_run_ones_in_a_block_test         0.3508288193036804 PASS
binary_matrix_rank_test                  0.8874835399970811 PASS
dft_test                                 0.18018545975265288 PASS
non_overlapping_template_matching_test   0.9999718030967704 PASS
overlapping_template_matching_test       0.22100396714394427 PASS
maurers_universal_test                   0.9993863673723593 PASS
linear_complexity_test                   0.08599709414866186 PASS
serial_test                              0.030274564609006616 PASS
approximate_entropy_test                 0.030424674720146302 PASS
cumulative_sums_test                     0.13827985254186737 PASS
random_excursion_test                    0.20066307170007402 PASS
random_excursion_variant_test            0.025691749776935398 PASS
```

6-ilova

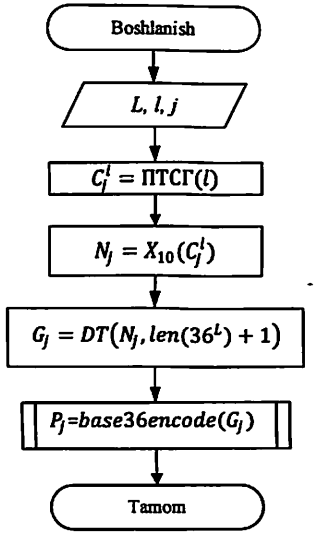
NIST SPECIAL PUBLICATION 800-22 statistik testlar maqsadi

№	Test nomi	Maqsadi
1	2	3
1.	Bitlar bo'yicha chastotalar testi (The Frequency Monobit Test)	Mazkur test ketma-ketlikdagi nollar va birlar sonini muvozanatda ekanligini tekshiradi
2.	Kumilativ yig'indi testi (The Cumulative Sums Test)	Mazkur testlash usuli bitlar ketma-ketliklarini (0 ni -1 va 1 ni 1 shaklida) kumuly ativ yig'indisini noldan maksimal og'ishini ko'rsatadi
3.	Bloklar bo'yicha chastota testi (Test for Frequency within a Block)	Belgilab olingan ketma-ketlikning bloklari ustida chastotalar testini amalga oshiradi
4.	Seriyali test (Runs Test)	Ketma-ketlikdagi seriyali bir xil bitlarni taqsimotini tasodifiylikdan og'ishini ko'rsatadi
5.	Matritsaning rangi testi (Random Binary Matrix Rank Test)	32×32 bitli matritsaning qatorlari orasidagi chiziqli bog'liqlikni ko'rsatadi
6.	Spektral test (Spectral Test)	Ketma-ketlikdagi takrorlanuvchi blokni aniqlaydi
7.	Bir birini takrorlamaydigan davriy shablonlar testi (Non-overlapping Template Matching Test)	Oldingan berilgan shablondagi bitlar qatorini ketma-ketlikda mavjudligini ko'rsatadi
8.	Davriy shablon testi (Overlapping (Periodic) Template Matching Test)	Oldingan berilgan shablondagi bitlar qatorini ketma-ketlikda mavjudligini ko'rsatadi. Oldingi testdan farqi moslik topilganda siljishida
9.	Universal statik testi (Maurer's Universal Statistical Test)	Ikkita shablon o'rtasidagi bitlar sonini ko'rsatadi va ketma-ketlikni siqiluvchanligini aniqlashga xizmat qiladi

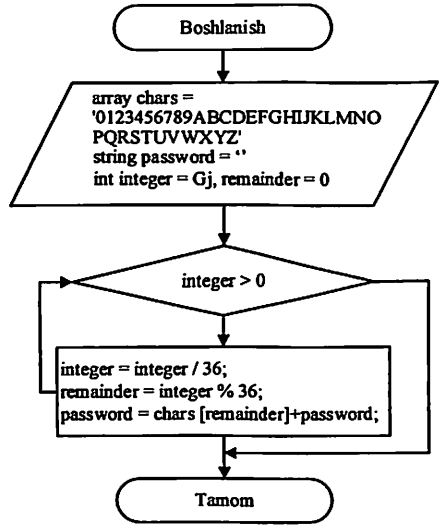
1	2	3
10.	Taxminiy entropiya testi (The Approximate Entropy Test)	Ketma-ketlikdagi m va $m+1$ bitli ikki kesishuvchi bloklar chastotasini tasodifiy bo'lgan analogidagi kesishuvchi bloklar chastotasini taqqoslaydi
11.	Ixtiyoriy og'ish testi (The Random Excursions Test)	Kiruvchi to'liq tasodif ketma-ketlikda ma'lum bir holatli tashriflar soni bir xil miqdordan farqlanishini aniqlash
12.	Ixtiyoriy og'ishning boshqa testi (The Random Excursions Variant Test)	Ketma-ketlikda ixtiyoriy aylanishda kutilayotgan turli holatdagi tashriflar sonidan og'ishini ko'rsatadi
13.	Davriylik testi (The Serial Test)	m bitli bir-biriga mos keluvchi shablonlarning paydo bo'lish ehtimoli haqiqiy tasodifiy ketma-ketlikdagi kabi 2^m ga yaqinligini ko'rsatadi
14.	Lempel-Ziv kompleks testi (Lempel-Ziv Complexity Test)	Ketma-ketlikdagi juft so'zlarning sonini va shu orqali ketma-ketlikning siqiluvchanligini ko'rsatadi
15.	Chiziqli murakkablik testi (Linear Feedback Shift Register)	Ketma-ketlikning chiziqli murakkabligini chiziqli teskari aloqali siljitish registridan olingan ketma-ketlik sifatida ko'rib, davrini hisoblash

7-ilova

2 - algoritm. S_1 to'plam belgilaridan iborat parollarni hosil qilish usulining blok sxemasi

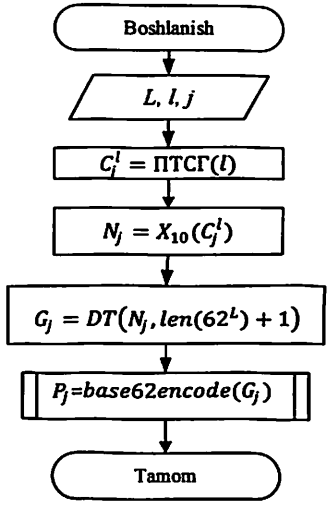


a) Umumiy ko'rinishi

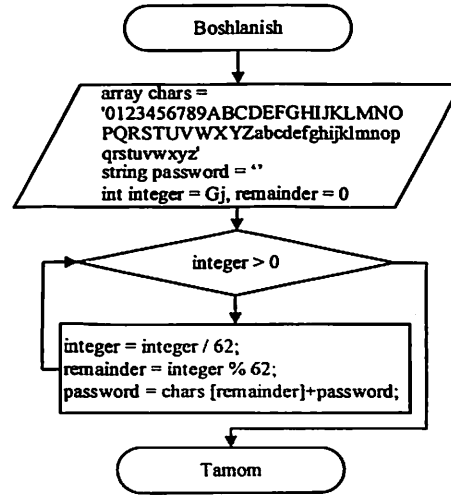


a) base36encode() funksiya blok sxemasi

3 - algoritm. S_2 to'plam belgilaridan iborat parollarni hosil qilish usulining blok sxemasi



a) Umumiy ko'rinishi



b) base62encode() funksiya blok sxemasi

9-ilova

S_2 to'plam elementlaridan iborat parollarni turli generatorlarda hosil qilish natijalari

Parol generatori	8 ta belgidan iborat parol	10 ta belgidan iborat parol
random.org [96]	0	0
PWGen dasturi [97]	0	0
Algorithm_Ph algoritmi	0	0
3 - algoritm	0	0

MUNDARIJA

M.M. Karimov, J.T. Arziyeva, Z.T. Xudoyqulov

BIR MARTALI PAROLLAR ASOSIDA FOYDALANUVCHILARNI AUTENTIFIKATSIYALASH PROTOKOLLAR

MONOGRAFIYA

Muharrir **Sh. Bazarova**
Badiiy muharrir **K. Boyxo'jayev**
Kompyuterda sahifalovchi **Z. Ulug'bekova**

Nashr. lits. AI № 305.
Bosishga ruxsat 23.04.2021-yilda berildi.
Bichimi 60x84 $^{1}/_{16}$. Ofset qog'ozi №2.
"Times New Roman" garniturasi.
Shartli b.t. 6,9. Nashr hisob t. 7,2.
Adadi 50 dona. 28-buyurtma.

«IQTISOD-MOLIYA» nashriyoti
100000, Toshkent, Amir Temur, 60 «A».

«DAVR MATBUOT SAVDO»
MChJ bosmaxonasida chop etildi.
100198, Toshkent, Qo'yliq, 4-mavze, 46.