Journal of Circuits, Systems, and Computers
Vol. 19, No. 7 (2010) 1399–1421

DOI: 10.1142/S0218126610006724

World Scientific
www.worldscientific.com

REAL-TIME STEGANALYSIS OF LSB-REPLACEMENT IN DIGITAL AUDIO STREAMS*

MOJTABA MAHDAVI* and SHADROKH SAMAVI†

Department of Electrical & Computer Engineering, Isfahan University of Technology, Isfahan, Iran
**mahdavi@ec.iut.ac.ir*
†*samavi96@cc.iut.ac.ir*

SORINA DUMITRESCU

Department of Electrical & Computer Engineering, McMaster University, Hamilton, Ontario, L8S4L8, Canada
sorina@mail.ece.mcmaster.ca

FERESHTEH AALAMIFAR

Department of Electrical & Computer Engineering, Queens University, Kingston, Ontario, K7L 3N6, Canada
fereshteh.alamifar@queensu.ca

PARISA ABEDIKHOOZANI

Department of Electrical & Computer Engineering, Isfahan University of Technology, Isfahan, Iran
parisa.abedi@ec.iut.ac.ir

Received 8 June 2009
Accepted 7 April 2010

Data hiding in the LSB of audio signals is an appealing steganographic method. This is due to the large volume of real-time production and transmission of audio data which makes it difficult to store and analyze these signals. Hence, steganalysis of audio signals requires online operations. Most of the existing steganalysis methods work on stored media files. In this paper, we present a steganalysis technique that can detect the existence of embedded data in the least significant bits of natural audio samples. The algorithm is designed to be simple, accurate, and to be hardware implementable. Hence, hardware implementation is presented for the proposed algorithm. The proposed hardware analyzes the histogram of an incoming stream of audio signals by using a sliding window strategy without needing the storage of the signals. The algorithm is mathematically modeled to show its capability to accurately predict the amount of embedding in an incoming stream of audio signals. Audio files with different amounts of embedded data were used to test the algorithm and its hardware implementation. The experimental results prove the functionality and high accuracy of the proposed method.

*This paper was recommended by Regional Editor Majid Ahmadi.

Keywords: Steganography; audio signals; LSB flipping; steganalysis; real-time; embedded systems.

1. Introduction

Applications such as copyright management, secret communications, and embedding executable files for access control, have produced a demand for steganography in audio signals. One of the main requirements on the part of the steganography is that the presence of the secret data in the stego audio stream be perceptually imperceptible and statistically undetectable. The second requirement is the size of the payload to be sufficiently high.[1]

A popular steganography technique for audio signals is the least significant bit (LSB) embedding. The sender and receiver of the messages share a private secret key that creates a random sequence of samples of a digital signal. The secret message, which is compressed and encrypted, is embedded in the LSBs of the samples of the stream. The LSB plane of the audio media has a low power additive white Gaussian noise (AWGN) characteristic and is similar to the characteristics of the secret message after compression and encryption. In the field of psychoacoustics it is shown that the human auditory system (HAS) is not sensitive to AWGN embedding message bits in the least significant bit plane, making the embedded media to be indiscernible from the original audio signals.[2] The extraction process retrieves the secret massage by extracting the LSBs of the audio stego stream. The receiver needs all of the samples of the stego audio.

There are two main methods for LSB embedding which are LSB replacement and LSB matching. In the first category, the intended data replaces LSBs of the samples. In the LSB matching method, one unit is added to or subtracted from each sample to generate LSBs that match with the intended data.[3] Each of these two steganographic methods has its applications and steganalysis attacks. The aim of this paper is to detect the LSB replacement steganography.

A number of algorithms have been proposed which embed data beyond the least significant bits of samples. See Refs. 4–6 for more detail. It has also been proposed to use the LSB plane of the wavelet coefficients of audio signals to increase the security of the algorithm.[7]

Steganalysis of LSB embedding has been addressed by numerous references too. See Refs. 8–10 for more details. Many of the steganography tools that are available use some forms of LSB replacement, but their results are highly vulnerable to statistical analysis. There is an abundance of detection methods which usually use the structural or combinatorial properties of the LSB replacement algorithms.[11] One of the earliest steganalytic attacks for LSB replacement is introduced by Westfeld and Pfitzmann[12] which was based on pairing of interrelated bins. This attack is effective when the full capacity of media is used for embedding or when the position of samples

containing the hidden data is known. To alleviate this shortcoming, Provos[13] generalized the Westfeld's attack using a sliding window.

Fridrich[14] has proposed a very efficient steganalysis method called RS. Also, Dumitrescu[15] has introduced a powerful steganalysis method which is called sample pair. Sample pair (SP) is capable of detect very low embedding rates. These two mentioned detection methods and their extensions are the most popular and effective attacks for the LSB replacement steganography.

Most of steganalysis methods require working on the statistical characteristics of the digital data. Most of the steganalysis algorithms are applied offline and their computational complexity is not of importance. The main concern of these methods is the algorithm's accuracy. In this paper, we present a simple steganalysis method for LSB embedding of digital signals. Online steganography in audio stream does not give the attacker the luxury of storing the signals for the offline analysis of the stored data. The widespread and high volume of audio transmissions forbids the storing of the audio signals. Our main concern in this paper is the simplicity of the algorithm while keeping the accuracy at an acceptable level. Three histogram parameters are computed to estimate the amount of embedding. We mathematically proved the relation between the extracted parameters and the amount of embedded data. These parameters can be easily computed for an incoming audio stream using a first in first out (FIFO) hardware queue. The algorithm is implemented in hardware by minimizing the computational overheads. A number of complex operations, such as histogram generation and subtraction of histogram bins, are performed by simple digital counters. This simple architecture has the potential of becoming an integral part of any network router or telephone switch where multiple copies of the hardware can work in parallel on different channels.

The paper is organized such that in Sec. 2 the proposed steganalysis for detection of hidden data in the least significant bits of a media streams, which is called *Sigma Delta* technique, is introduced. Sec. 3 contains the details of the proposed hardware architecture of Sigma Delta algorithm. The implementation results to proof the functionality and accuracy of the algorithm and its hardware implementation are presented in Sec. 4 of the paper.

2. Sigma Delta Steganalysis

Here, we assume that the digital audio signals are 8-bit values. We denote the histogram of a piece of an audio stream as $h^p = \{h_0^p, \ldots, h_{255}^p\}$ where $0 \leq p \leq 1$ is the amount of embedding in the piece. The histogram of an audio piece with no embedding is represented by h^0 and the histogram of a piece that is completely embedded is symbolized by h^1.

In Fig. 1 it is shown that embedding 0 in odd numbers or embedding 1 in even samples changes the LSB of the signal otherwise the embedding process does not produce any change in the audio signal.

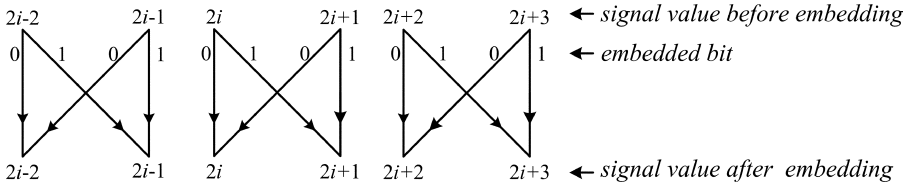


Fig. 1. Changes in odd−even valued samples as a result of LSB-flipping.

Since the values of signals with $2i$ and $2i+1$ intensities can interchange due to the embedding process we call the histogram bins corresponding to these intensities as the *interrelated* bins. On the other hand, $2i$ and $2i-1$ bins of a histogram are *unrelated*.

When an audio piece is completely embedded *pairing* occurs in its histogram. The value h_k^0 is a function of the distribution of the signals of the original audio stream. The embedding process causes some changes in the histogram due to changes that are made to the least significant bits of the audio stream. The final value of h_{2i}^1 is equal to the number of 0's embedded in signals with $2i$ and $2i+1$ intensities. With the same token, the number of 1's that are embedded in signals with intensities $2i$ and $2i+1$ makes up the value of h_{2i+1}^1. In fact this is equivalent to performing binomial trials with the total number of $n_i = h_{2i}^0 + h_{2i+1}^0 = h_{2i}^1 + h_{2i+1}^1$. To present our steganalysis method let us consider the sample histogram of Fig. 2.

We notice that due to the embedding process the distance between the sizes of every two interrelated bins shrinks while that distance for the unrelated bins expands. We exploit this fact to come up with an algorithm which we refer to as Sigma Delta. To formulate the changes that are just mentioned we define α^p as the sum of the absolute differences between interrelated bins of a histogram. Also β^p is the sum of the absolute differences between unrelated bins in the histogram of a

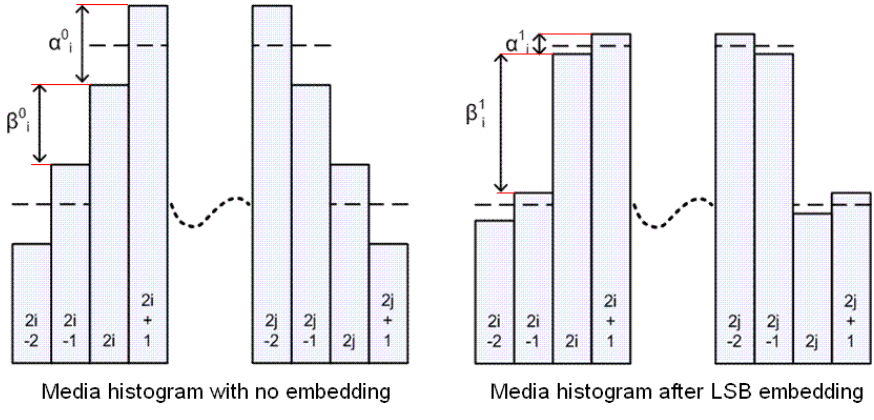


Fig. 2. Effect of embedding on the relative value of interrelated and unrelated bins.

stego-audio stream with an embedding ratio of p. Formally, we have

$$\begin{aligned}
\alpha^p &= \sum_{i=0}^{127} \alpha_i^p, \quad \alpha_i^p = |h_{2i}^p - h_{2i+1}^p|\,, \qquad 0 \le i \le 127\,, \\
\beta^p &= \sum_{i=0}^{127} \beta_i^p, \quad \beta_i^p = \begin{cases} |h_{2i}^p - h_{2i-1}^p|\,, & 0 < i \le 127\,, \\ |h_0^p - h_{255}^p| & i = 0 \end{cases} \\
\beta^p - \alpha^p &= \sum_{i=0}^{127} \delta_i^p, \quad \delta_i^p = \alpha_i^p - \beta_i^p\,.
\end{aligned} \tag{1}$$

For ease of representation we refer to $\beta^p - \alpha^p$ as $\sum \delta^p$. We will show that $\sum \delta^p$ is a linear function of p. By calculating $\sum \delta^p$ from the histogram of the medium and by knowing $\sum \delta^1$ from the fully embedded file, we can estimate the embedding ratio, p. Characteristics of a file, where LSB's of all of its signals are replaced by random bits, corresponds to $\sum \delta^1$.

2.1. *Expected value of α^1 and β^1*

Let H_k^p denote the random variable representing the number of samples with intensity of k, when the audio stream has embedding ratio of p. For $p = 0, H_k^0$ takes only one value of h_k^0 for a given image. Because the secret message is random with equal probability of embedding 0 or 1, $\text{Prob}(0) = \text{Prob}(1) = 0.5$, it follows that when the embedding ratio is p, the expected number of 0's embedded in samples of same value k equals the expected number of 1's embedded in samples of value k, further equals to $h_k^0 \frac{p}{2}$. Thus using the transitions illustrated in Fig. 2, we obtain:

$$E[H_{2i}^p] = h_{2i}^0\Big(1 - \frac{p}{2}\Big) + h_{2i+1}^0 \frac{p}{2}, \quad 0 \le i \le 127\,. \tag{2}$$

$$E[H_{2i+1}^p] = h_{2i}^0 \frac{p}{2} + h_{2i+1}^0\Big(1 - \frac{p}{2}\Big), \quad 0 \le i \le 127\,. \tag{3}$$

Suppose the sum of the values of two interrelated bins is n_i. Then

$$\begin{aligned}
&n_i = h_{2i}^0 + h_{2i+1}^0\,. \\
&\alpha_i^1 = |h_{2i}^1 - n_i + h_{2i}^1| = 2|h_{2i}^1 - n_i/2| = 2|h_{2i}^1 - E[H_{2i}^p]|\,.
\end{aligned} \tag{4}$$

It happens that α_i^1 follows the definition of the absolute deviation and therefore $E[A_i^1]$ represents mean deviation of the distribution, where A_i^p is the random variable for the distribution of α_i^p. For a binomial distribution the mean deviation is calculated as

$$E[A_i^1] = \begin{cases} \dfrac{n_i!!}{(n_i - 1)!!} & \text{for odd } n_i\,. \\ \dfrac{(n_i - 1)!!}{(n_i - 2)!!} & \text{for even } n_i\,. \end{cases} \tag{5}$$

where !! denotes double factorial.

The expected value of H^p_{2i} is equal to the initial number of $2i$ signals plus number of 0's embedded in $2i+1$ and $2i$ signals minus the number of 1's embedded in $2i$ signals. This imply that

$$E[H^p_{2i}] = h^0_{2i} + p\left(\frac{h^0_{2i} + h^0_{2i+1}}{2}\right) - p \cdot h^0_{2i} = h^0_{2i} + \frac{p}{2}(h^0_{2i+1} - h^0_{2i})\,. \tag{6}$$

H^p_{2i} is a normal random variable with mean value of (3−3) and variance of

$$\mathrm{Var}[H^p_{2i}] = \frac{p}{4N}(h^0_{2i+1} + h^0_{2i})(2N - (h^0_{2i+1} + h^0_{2i})) + \frac{p}{N}h^0_{2i}(N - h^0_{2i})\,. \tag{7}$$

where N is the total number of signals in the media file.

Now for the normal random variable $(H^p_{2i} - H^p_{2i+1})$ using (6) and (7) mean (μ_A) and variance (σ^2_A) after simplification will be:

$$\begin{aligned} E[H^p_{2i} - H^p_{2i+1}] &= \mu_A = (1-p)(h^0_{2i+1} - h^0_{2i}) \\ \mathrm{Var}[H^p_{2i} - H^p_{2i+1}] &= \sigma^2_A = \frac{p}{N}(h^0_{2i+1} + h^0_{2i})(2N - (h^0_{2i+1} + h^0_{2i})) \\ &\quad - \frac{4pN(1 - h^0_{2i}/h^0_{2i+1})}{h^0_{2i+1}}\,. \end{aligned} \tag{8}$$

It is shown that the absolute value of a normal distribution with mean μ_A and variance σ^2_A has a mean of

$$E[|H^p_{2i} - H^p_{2i+1}|] = E[A^p_i] = \sigma_A\sqrt{\frac{2}{\pi}}e^{-\frac{1}{2}(\mu_{A/\sigma_A})^2} + \mu_A \times \mathrm{erf}\left(\frac{\mu_A}{\sqrt{2}\sigma_A}\right). \tag{9}$$

Calculating the expected value of α^1 using (9) is more practical than using (5).

$$E[\alpha^1] = \sum_{i=0}^{127} E[A^1_i] = E\left[\sum_{i=0}^{127} A^1_i\right]. \tag{10}$$

To calculate $E[|H^1_{2i} - H^1_{2i-1}|]$ we consider that the absolute value of the difference between two random variables X and Y is $E[|X-Y|] = \sum_j \sum_k |x_j - y_j| p_x(x_j) p_y(y_k)$. Then B^1_i, which is the random variable of β^p_i, has an expected value of

$$\begin{aligned} E[B^1_i] &= \sum_j^{n_i} \sum_k^{h_i - 1} |j-k| p_i(j) p_{i-1}(k) \\ &= \sum_j^{n_i} \sum_k^{h_i - 1} |j-k| \binom{n_i}{j} \left(\frac{1}{2}\right)^{n_i} \binom{n_{i-1}}{k} \left(\frac{1}{2}\right)^{n_{i-1}}. \end{aligned} \tag{11}$$

For large values of n_i and n_{i-1} it is difficult to calculate $E[B^1_i]$ using Eq. (11). Hence, we use the normal distribution to find a close approximation for $E[B^1_i]$. Ross[15] points out that any binomial distribution with parameters (n, p) for large values of n can be approximated by $\mathrm{Norm}(np, np(1-p))$, where Norm indicates the normal distribution. For random variables H^p_{2i} and H^p_{2i-1} with mean values of $n_i/2, n_{i-1}/2$ and

standard deviation values of $n_i/4, n_{i-1}/4$, the value of B_i^1 is represented by:

$$B_i^1 = |\mathrm{Norm}(n_i/2, n_i/4) - \mathrm{Norm}(n_{i-1}/2, n_{i-1}/4)|\,.$$

Therefore, B_i^1 has a distribution of the form Norm $(\frac{n_i - n_{i-1}}{2}, \frac{n_i + n_{i-1}}{4})$ and its expected value is

$$E[B_i^1] = \int_{-\infty}^{\infty} \left(|x| \frac{1}{\sqrt{2\pi\left(\frac{n_i+n_{i-1}}{4}\right)}} e^{-\frac{(2x-n_i+n_{i-1})^2}{2(n_i+n_{i-1})}} \right) \mathrm{d}x\,. \tag{12}$$

Performing numerical simulation using (12) is much more practical than that of (11). Hence

$$E[\beta^1] = E\left[\sum_{i=1}^{127} B_i^1\right] = \sum_{i=1}^{127} E[B_i^1]\,. \tag{13}$$

2.2. *Estimated parameters*

The expected values of β^1 and α^1 that are offered in the above relations are dependent on the characteristics of the cover media files. In the following, we present estimation of these parameters to experimentally calculate the amount of the embedded data in a media file.

Assuming that the variance of $H_{2i}^p - H_{2i+1}^p$ is low, we approximate $E[|H_{2i}^p - H_{2i+1}^p|] \approx |E[H_{2i}^p] - E[H_{2i+1}^p]|$. Then, we can approximate the expectation of A_i^p, which is the random variable for the distribution of α_i^p, as: $E[A_i^p] \approx |h_{2i}^0 - h_{2i+1}^0|$ $(1-p)$ which by summing up over all i will result: $E[\sum_{i=0}^{127} A_i^p] \approx (\sum_{i=0}^{127} \alpha_i^0)(1-p)$

Hence,

$$\alpha^p \approx \alpha^0(1-p)\,. \tag{14}$$

Equation (14) shows that α^p is a linear function of p and also reveals the coefficients of proportionality. Now, we want to prove the linearity of β^p. Equation (3) implies that:

$$E[H_{2i-1}^p] = h_{2i-2}^0 \frac{p}{2} + h_{2i-1}^0 \left(1 - \frac{p}{2}\right). \tag{15}$$

Using Eqs. (2) and (15) implies:

$$E[H_{2i}^p - H_{2i-1}^p] = h_{2i}^0 - h_{2i-1}^0 + \frac{p}{2}[(h_{2i+1}^0 - h_{2i}^0) + (h_{2i-1}^0 - h_{2i-2}^0)]\,. \tag{16}$$

We use the following critical assumption that is supported by experiments and relies on the piecewise linearity of the media histogram for almost all k.

$$h_{k-1}^0 - h_k^0 \approx h_k^0 - h_{k+1}^0\,. \tag{17}$$

This assumption implies that:

$$h_{2i-1}^0 - h_{2i-2}^0 \approx h_{2i}^0 - h_{2i-1}^0 \approx h_{2i+1}^0 - h_{2i}^0\,. \tag{18}$$

Equation (17) is true for almost all i. Then Eqs. (16) and (18) imply that for almost all values of i

$$E[H^p_{2i} - H^p_{2i-1}] \approx (h^0_{2i} - h^0_{2i-1})(1+p)\,.$$

Therefore, by approximating

$$E[|H^p_{2i} - H^p_{2i-1}|] \approx |E[H^p_{2i}] - E[H^p_{2i-1}]|\,.$$

We further obtain

$$E[|H^p_{2i} - H^p_{2i-1}|] \approx |h^0_{2i} - h^0_{2i-1}|(1+p)\,. \tag{19}$$

Let S denote the set of i's for which the above holds. Then

$$E\left[\sum_{i\in S} B^p_i\right] \approx \left(\sum_{i\in S} \beta^0_i\right)(1+p)\,. \tag{20}$$

where B^p_i is the random variable for the distribution of β^p_i. Since S contains almost all i's, we may assume that $E[\sum_{i\in S} B^p_i]/E[\sum_{i\in S} B^p_i]$ is very small. Then, we can extend Eq. (20) to the summation over all i's, which implies:

$$E[\beta^p] \approx \beta^0(1+p)\,. \tag{21}$$

$$\beta^p \approx \beta^0 + \beta^0 \times p\,. \tag{22}$$

From Eqs. (1) and (22) it is concluded that $\Sigma\delta^p = \beta^p - \alpha^p \approx \beta^0 - \alpha^0 + (\beta^0 + \alpha^0)p$. Thus, the embedding ratio p can be approximated as

$$p \approx \frac{\beta^p - \alpha^p - (\beta^0 - \alpha^0)}{\beta^0 + \alpha^0}\,. \tag{23}$$

Based on the piecewise linearity assumption for the media histogram (Eq. (7)) we should have $\beta^0 \approx \alpha^0$ or $\beta^0 - \alpha^0 \approx 0$. Also from (14) and (22) we see that for $p = 1$ we have $\alpha^1 \approx 0, \beta^1 \approx 2\beta^0$ and $\Sigma\delta^1 = \beta^1 - \alpha^1 \approx 2\beta^0 \approx \beta^0 + \alpha^0$. Further, substituting in (23) we obtain

$$p \approx \frac{\Sigma\delta^p}{\Sigma\delta^1}\,. \tag{24}$$

where $\Sigma\delta^p$ is calculated from the initial stego file, while $\Sigma\delta^1$ is calculated after the stego file is 100% embedded by replacing LSB's of all of the signals by random bits.

However, the estimation of Eq. (24) is not reliable enough. The error in the approximation $\beta^0 \approx \alpha^0$ has high impact on the quality of the estimator. Therefore to improve the accuracy of the estimation method we propose a different approach to calculate $\beta^0 - \alpha^0$ and $\beta^0 + \alpha^0$ to be used in Eq. (23). For this we form a second audio file by adding 1 intensity level to all signals in the stego file that is to be analyzed, taking 256 as 0. The histogram of the second file is the same as the first one except all of the bins are shifted one place to the right. We call this second file as the shifted file.

The measured initial parameters of the shifted file are denoted $\tilde{\alpha}^p, \tilde{\beta}^p$ and $\Sigma\tilde{\delta}^p$. Also the shifted file is fully embedded to result $\Sigma\tilde{\delta}^1$.

Due to the shifting of the histogram we get $\tilde{\alpha}^p = \beta^p, \tilde{\beta}^p = \alpha^p$, and $\Sigma\tilde{\delta}^p = -\Sigma\delta^p$. With these two audio files we perform a number of experiments, which their results are conceptually illustrated in Fig. 3.

The following are done to both the original and shifted files. In the first experiment, the value of $\Sigma\delta$ is calculated and we see $\Sigma\delta^p = -\Sigma\tilde{\delta}^p$. Then 10% of the signals of both files are embedded and $(\Sigma\delta, \Sigma\tilde{\delta})$ are calculated. The signals to be embedded are selected using an embedding key so that the embedded signals are uniformly distributed throughout the stream. We call this embedding as the *embedding for steganalysis* and call its amount Z which is independent of p. The next experiment is similarly performed except Z is increased to 0.2. We keep increasing Z in the following experiments. It is seen that as Z is increased the calculated values of $\Sigma\delta$ and $\Sigma\tilde{\delta}$ increase too, except for files that initially have 100% embedding where $\Sigma\delta$ stays constant. The results of these experiments are plotted as $\Sigma\delta$ and $\Sigma\tilde{\delta}$ graphs in Fig. 3. Part (a) of Fig. 3 shows the results of a series of experiment on an audio file (and its shifted version) which initially had no embedding. Same series of experiments can be performed on an audio file with an arbitrary p% initial embedding to produce graphs of Fig. 3(b). Figure 3(c) shows a situation that if the file under analysis had 100% embedded data then applying Z amount of embedding to it would not change $\Sigma\delta$ of the file but $\Sigma\tilde{\delta}$ would start with a large negative number and reaches $\Sigma\tilde{\delta}^1$ at $Z = 1$. Hence the $\Sigma\delta$ graph has the following line equation: $\Sigma\delta = (\Sigma\delta^1 - \Sigma\delta^p)Z + \Sigma\delta^p - d$ where $d = (\Sigma\delta^1 - \Sigma\tilde{\delta}^1)/2$.

Figure 3 is a graphical representation of the possible outcomes of actual experiments. For media files with $\beta^0 = \alpha^0$ the $\Sigma\delta$ graph will be the dotted line with $\Sigma\delta^1 = \Sigma\tilde{\delta}^1$ and d becomes zero. For cases that $\beta^0 \neq \alpha^0$ graphs of $\Sigma\delta$ and $\Sigma\tilde{\delta}$ do not intersect at $Z = 1$.

In the followings, we show that the $\Sigma\delta$ parameters obtained from the original and shifted files can be used to modify Eq. (24) to generate an accurate estimation of p.

Let g_k^p denote the size of the kth histogram bin of the shifted file. Then, we obtain

$$g_k^p = h_{k-1}^p \,. \tag{25}$$

Then, we have

$$\begin{aligned} \Sigma\tilde{\delta}^1 &\approx \sum_{i=1}^{127} |g_{zi}^1 - g_{zi-1}^1| \approx \sum_{i=1}^{127} \left| \frac{g_{zi}^p + g_{zi+z}^p}{2} - \frac{g_{zi-z}^p + g_{zi-z}^p}{2} \right| \\ &= \sum_{i=1}^{227} \left| \frac{h_{zi-z}^p + h_{zi}^p}{2} - \frac{h_{zi-z}^p + h_{zi-z}^p}{2} \right| . \end{aligned} \tag{26}$$

The second equality in (26) is due to the fact that after full embedding the number of samples with $2i$ and $2i + 1$ are almost equal. The third equality follows from Eq. (25).

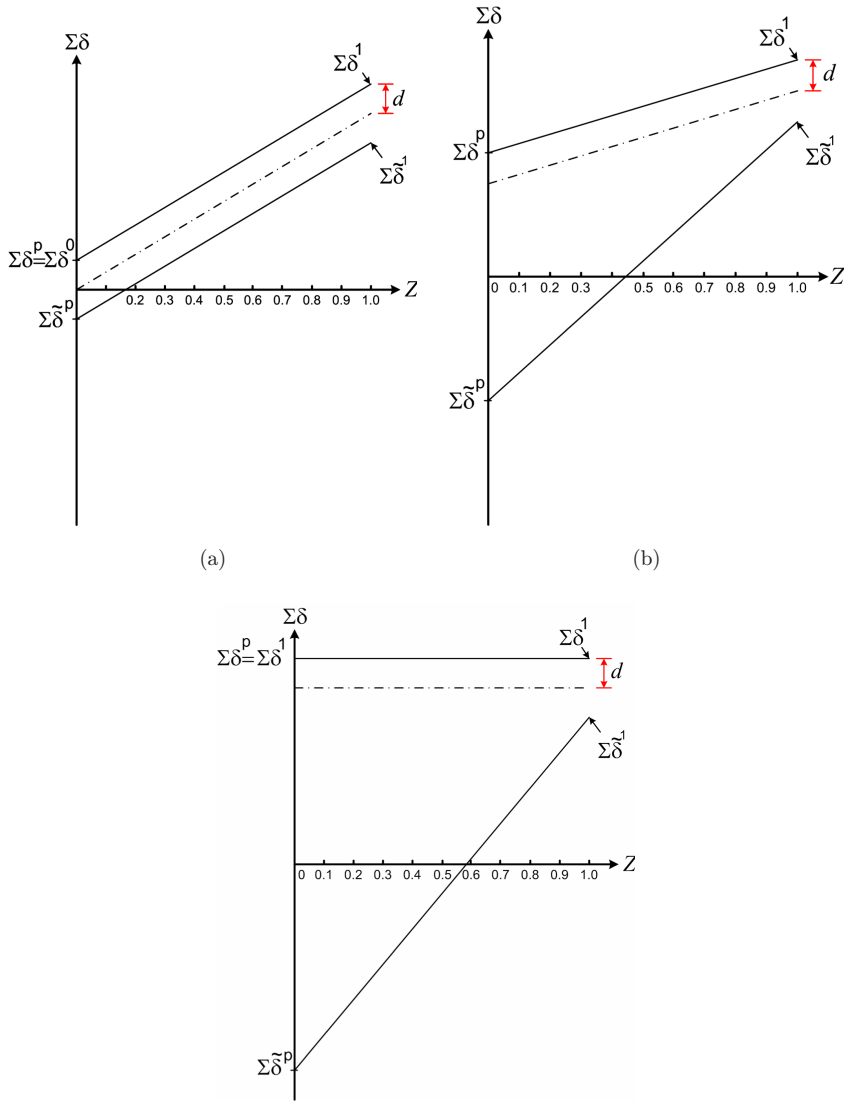


Fig. 3. Relation between value of $\Sigma\delta$ and the amount of embedding for analysis (z). (a) File under analysis has no initial embedding. (b) File under analysis has an unknown P embedding ratio. (c) File under analysis has 100% embedded data.

Denoting that $h^0_{2i+1} - h^0_{2i} = a^0_i, h^0_{2i} - h^0_{2i-1} = b^0_i$, by using Eqs. (14) and (26) we get

$$\Sigma\tilde{\delta}^1 = \sum_{i=1}^{127} \frac{1}{2} \left| h^0_{2i} + \frac{p}{2} a^0_i + a^0_{i-1}(1-p) - h^0_{2i-3} + \frac{p}{2} a^0_{i-2} \right|.$$

With piecewise linearity assumption of $\alpha^0_{i-1} = \alpha^0_i = \alpha^0_{i-1}$ (Eq. (17)) we get

$$\begin{aligned}\Sigma\tilde{\delta}^1 &\approx \frac{1}{2}\sum_{i=1}^{127}|h^0_{2i} - h^0_{2i-1} + 2a^0_{i-1} + h^0_{2i-2} - h^0_{2i-3}| \\ &\approx \frac{1}{2}\sum_{i=1}^{127}|\beta^0_i| + \frac{2}{2}\sum_{i=1}^{127}|\alpha^0_{i-1}| + \frac{1}{2}\sum_{i=1}^{127}|\beta^0_{i-1}| \approx \beta^0 + \alpha^0\,.\end{aligned}$$

Therefore, we can define $(\beta^0 - \alpha^0)$ and $(\beta^0 + \alpha^0)$ as

$$\begin{aligned}(\beta^0 - \alpha^0) &\approx \frac{\Sigma\delta^1 - \Sigma\tilde{\delta}^1}{2} \\ (\beta^0 + \alpha^0) &\approx \frac{\Sigma\delta^1 + \Sigma\tilde{\delta}^1}{2}\,.\end{aligned}$$

Substituting the above relations in (23) results in

$$p \approx \frac{\Sigma\delta^p - \left(\dfrac{\Sigma\delta^1 - \Sigma\tilde{\delta}^1}{2}\right)}{\dfrac{\Sigma\delta^1 + \Sigma\tilde{\delta}^1}{2}}\,. \tag{27}$$

which is the proposed estimation equation. We see that the series of experiments that were mentioned to generate graphs of Fig. 3 are just to prove the claim that $\Sigma\delta$ increases linearly with increase in the amount of embedding for steganalysis, Z. Otherwise, to estimate the amount of LSB embedding that is present in a file, it is just necessary to generate the shifted file and apply $Z = 1$ to both files to calculate $\Sigma\delta^p, \Sigma\delta^1$ and $\Sigma\tilde{\delta}^1$.

3. Proposed Architecture

In this section, we introduce a hardware implementation which is devised to implement the Sigma Delta steganalysis. Sigma Delta, similar to many other steganalysis methods that have been designed for LSB-embedding, analyzes the histogram of a file. When the size of the file increases the number of mathematical operations needed to form the histogram and perform the analysis increases too. Our proposed hardware architecture is designed to work on a number of audio signals and its complexity is independent of the size of the audio stream under the analysis. The general concept of the architecture is shown in Fig. 4. As it is shown, this hardware contains three blocks for calculating $\Sigma\delta$. The first block calculates $\Sigma\delta^p$ by taking the audio samples and without any modifications works on its histogram. The second block is for calculating $\Sigma\delta^1$. This is done by replacing all of the LSBs of the audio samples of a stream with random data. The random bit stream is produced by a

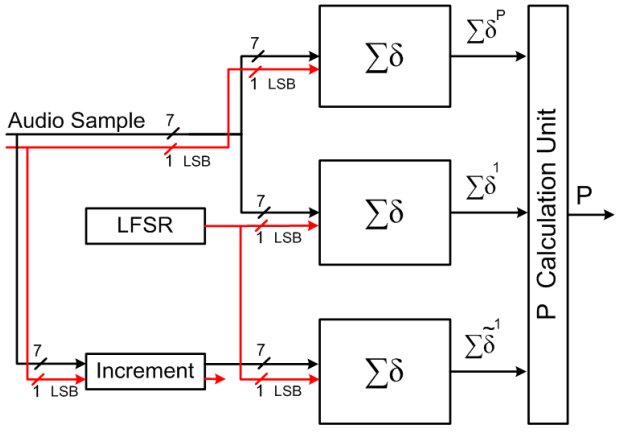


Fig. 4. Main block diagram of the proposed architecture.

linear feedback shift register (LFSR). The third block is to find $\Sigma\bar{\delta}^1$ by generating the shifted file. This is done by incrementing all of the signals of the original stream and fully embedding it by replacing the LSBs of the signals. Finally, the values of $\Sigma\delta^p$, $\Sigma\delta^1$ and $\Sigma\tilde{\delta}^1$ are fed into a block which calculates p. We do not go into the details of the p calculation unit since its operation is straightforward and it is made of standard elements.

3.1. *FIFO memory*

We need a real-time steganalysis of the voice. Therefore instead of analyzing an audio file with an unknown size, we only receive one signal at a time and analysis a slice of the incoming stream. Using a first-in-first-out (FIFO) memory unit we designed a sliding window structure. Initially the memory is empty but as the stream of audio signals arrive the memory fills up. In fact a queue is formed. From there on, with the arrival of one new sample at the end of the queue one old sample from the head of the queue is discarded. The samples are thoroughly analyzed before being discarded. At each point of time, the histogram of the samples inside of the FIFO is analyzed. With every rising-edge of the clock, one sample is added to the FIFO and one is removed. To speed up the operation of the hardware and to avoid redundancy in computations, each time the window slides we only consider the effect of the newly added sample and the effect of the departed sample in the computation of $\Sigma\delta$.

The frequency of the FIFO's clock should be half of the frequency of the system's clock. During every pulse of the system's clock, two operations should be done:

(1) adding the effect of the new sample; (2) removing the effect of the old sample. The FIFO block is shown in Fig. 5.

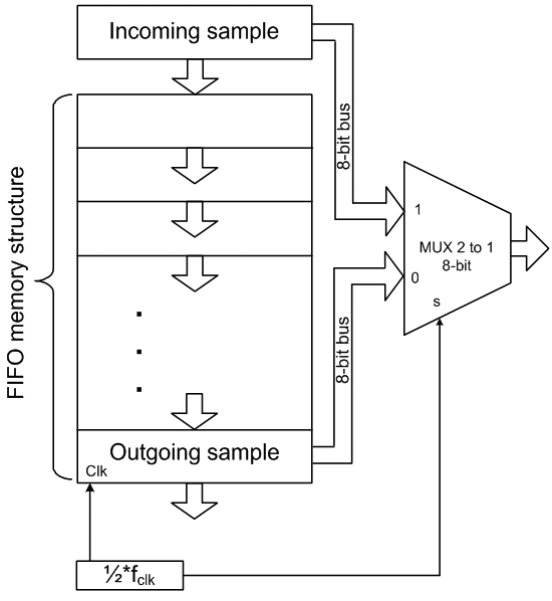


Fig. 5. FIFO memory block.

During the high half cycle of the clock the 2×1 multiplexer transfers the new sample to the input register and decrease block of Sec. 2. In contrast, in negative level, it transfers the old sample.

3.2. *Sigma Delta computation*

Computation of $\Sigma\delta$ according to Eq. (17) requires the generation of histograms for the original and the shifted files, computation of the differences between adjacent bins, summation of all differences to form (α^p, β^p), and the final subtraction of α^p from β^p. Performing these tasks for real-time process is especially difficult if we take large number of samples for higher accuracy. From hardware point of view, additions and subtractions are slow operations. To circumvent this problem we came up with the circuit of Fig. 6. In this architecture $\Sigma\delta$ is computed without the use of any adders or subtractors. A sample comes to this unit from the FIFO. There is an incoming sample for half a clock cycle and an outgoing sample arrives in the second half cycle. We add the incoming sample to the histogram and delete the outgoing sample. The histogram is preserved by the 256 counters. Each incoming sample increments one of the counters and an outgoing sample decrements its corresponding counter. The adjacent bins of the histogram are continuously compared to see the effect that the sample has on α^p and β^p. Each even numbered comparator look at two interrelated bins to test the inequality of $h_{2k} \geq h_{2k+1}$ while an odd numbered comparator examines two unrelated bins and checks the inequality $h_{2k} \geq h_{2k-1}$. The output of odd and even

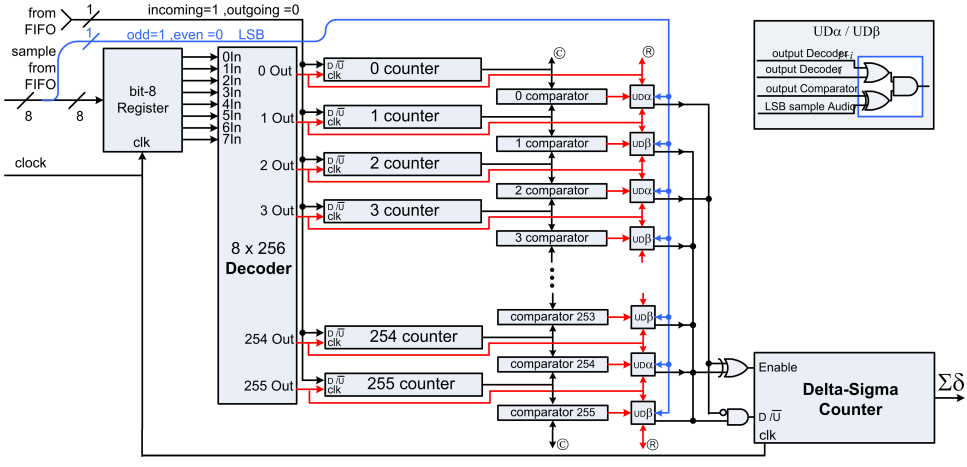


Fig. 6. Sigma Delta () Unit.

numbered comparators can be expressed as:

$$\text{out}_{\text{even comparator}} = \begin{cases} 1, & \text{if } h_{2k} \geq h_{2k+1}\,. \\ 0, & \text{otherwise}\,. \end{cases}$$

$$\text{out}_{\text{odd comparator}} = \begin{cases} 1, & \text{if } h_{2k} \geq h_{2k-1}\,. \\ 0, & \text{otherwise}\,. \end{cases}$$

We use the LSB of the sample to differentiate between odd and even samples. Two simple logic blocks, called UDα and UDβ, are used to decide that the sample has an increasing or decreasing effect on α^p and β^p. If the output of UDα is 1 then α should go Up, otherwise it should go Down. Same can be said for the UDβ unit. The outputs of one UDα and one UDβ are activated each time a sample is added or is eliminated from the histogram. A UDα block, using the output of an even numbered comparator and based on the sample being odd or even, increases or decreases the value of α^p. If an 8-bit audio sample S is being analyzed, then the function of UDα can be defined as:

if $h_{2k} \geq h_{2k+1}$ then

$$\alpha^p + (-1)^S$$

else

$$\alpha^p - (-1)^S$$

end if

Similarly, we can scrutinize the value of two unrelated adjacent bins and decide on the effect of the sample (S) on β^p. Hence the function of UDβ can be

formulated as:

$$\begin{aligned}&\text{if } h_{2k} \geq h_{2k-1} \text{ then}\\&\qquad \beta^p + (-1)^S\\&\text{else}\\&\qquad \beta^p - (-1)^S\\&\text{end if}\end{aligned}$$

Again to speed up the operation of the circuit, we do not store the values of α and β separately to subtract them and come up with the final value of $\Sigma\delta$. For each sample we produce one UDβ and one UDα outputs and since we want to calculate, we can decide on the effect that the sample has on the final value of $\Sigma\delta$. If both α and β are either going Up or Down, then $\Sigma\delta = \beta^p - \alpha^p$ does not change and the Sigma Delta counter of Fig. 5 is disabled. This is implemented by (UD$\alpha \oplus$ UDβ) enabling the counter. If β^p is going up and α^p is going down then the value of $\Sigma\delta$ should increase by 2 but we only increment the counter by 1. This is because eventually we need to calculate $p \approx [\Sigma\delta^p - (\Sigma\delta^1 - \Sigma\tilde{\delta}^1/2)]/[(\Sigma\delta^1 + \Sigma\tilde{\delta}^1)/2]$ and if both numerator and denominator are half of their actual values the ratio would still be correct. If β^p is going down and α^p is going up then the value of $\Sigma\delta$ is decremented. This operation is implemented by an AND gate at the Up/$\overline{\text{Down}}$ control of the counter. Hence, we see that with every arriving sample of the audio signal we modify the value of $\Sigma\delta$ which is a statistical characteristic of a stream of signals.

4. Implementation Results

In this section, we intend to explain the results of the implementation of the proposed algorithm and its hardware. We used uncompressed 8-bit mono natural audio files. An audio file or stream consists of consecutive audio signals or samples. Figure 7

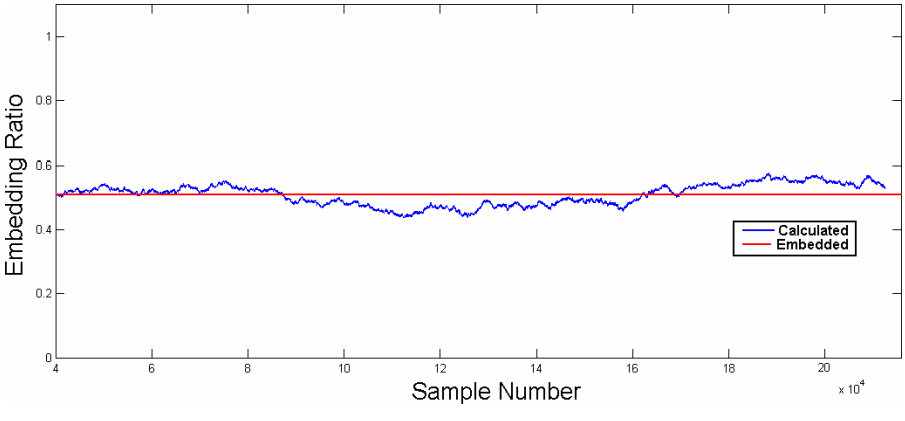


Fig. 7. Calculated value of P for an audio stream with 50% embedded signals.

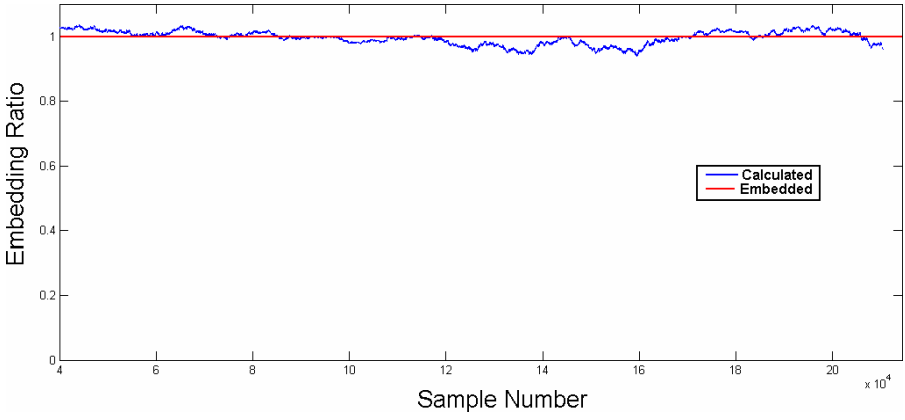


Fig. 8. Calculated value of P for an audio stream with 100% embedded signals.

shows the results of p computation on a stream of audio signals with 50% stationary embedding. The process of calculating p is same as mentioned in Sec. 2. We took the file with an unknown amount of embedding (in this case 50%) and formed a shifted file. Then the LSB of all of the signals were replaced with a random data ($Z = 1$). Parameters $\Sigma\delta^p$, $\Sigma\delta^1$ and $\Sigma\bar{\delta}^1$ were calculated and fed to Eq. (17) to calculate p. This was done for all of the signals of the stream.

Also Fig. 8 shows the results from a similar experiment where the same audio stream has an initial 100% embedding. The size of FIFO in both experiments is 60,000 samples. About 210,000 signals are in the audio stream.

We performed a total of 11 experiment on this file with embedding ratios of 0%, 10%,...,100%. The produced mean value of p for each of the 11 experiments is computed and the results are plotted in Fig. 9. The mean error for this experiment is 0.83%.

Embedding was performed in the span of zero to 100% of the capacity of an audio stream. Figures 10 and 11 show the results of about 120 experiments performed on 40 files. Figure 10 present the results obtained from the simulation process. The horizontal axis shows the actual embedding ratios and the simulation results are shown on the vertical axis. This means that, $\Sigma\delta^1$ and $\Sigma\tilde{\delta}^1$ are obtained after replacing LSB's of all of the signals in a file by random bits. Figure 11 illustrates the results obtained from the mathematical model of the algorithm. To obtain results from the model, for each file, $\Sigma\delta^p$ is calculated. But to find its $\Sigma\delta^1$ and $\Sigma\tilde{\delta}^1$ we used (9), (10), (12) and (13). The mean absolute error for the experiments shown in Fig. 10 is 5.06% with a variance of 0.22%. Also, the mean absolute error for the results shown in Fig. 11 is 3.44% with a variance of 0.08%.

Using ISE Foundation Design Suite 10.1 the proposed hardware was implemented on Spartan3:XC3S4000L FPGA. Table 1 shows the implementation results and utilization of FPGA assets.

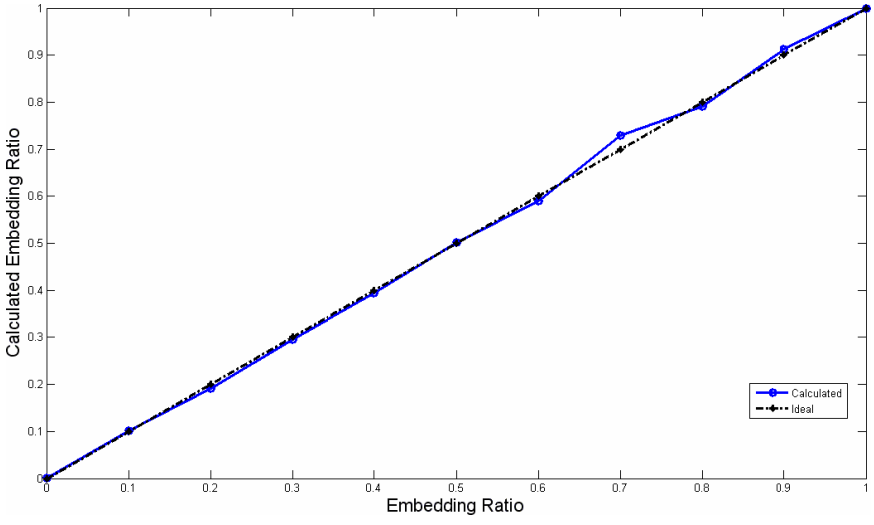


Fig. 9. Calculated vs. actual embedding ratios.

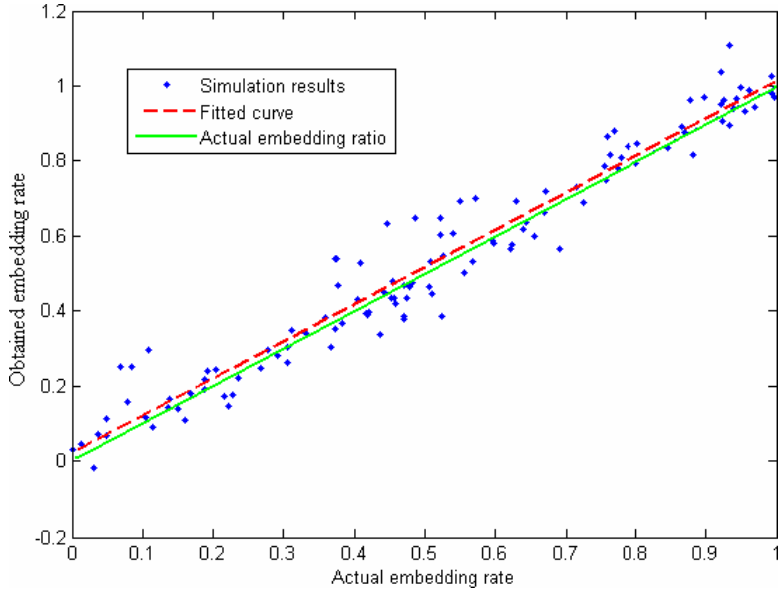


Fig. 10. Actual embedding and computed embedding ratio by simulation.

Figure 12 shows the waveforms obtained from the implemented circuit. At each clock pulse of the FIFO one sample arrives but the effect of the outgoing sample is considered too. That is why the frequency of the Sigma Delta block is twice that of the FIFO's clock. The correct operation of UDα and UDβ are shown.

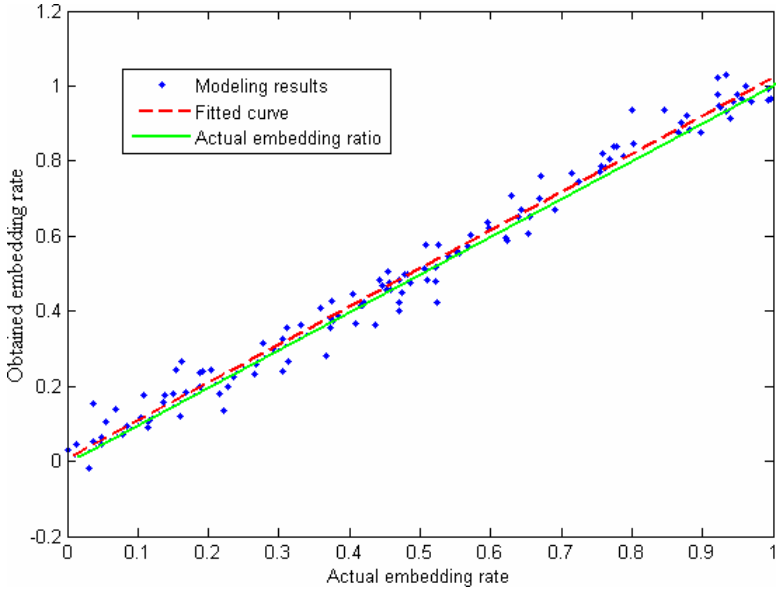


Fig. 11. Actual embedding and computed embedding ratio by mathematical modeling.

Table 1. List of FPGA device utilization.

Asset name	Utilized	Total available	Percent utilized
Slices	21972	27648	79%
Slice flip flops	17994	55296	32%
4 Input LUTs	42248	55296	76%
IOs	42	—	—
Bonded IOBs	42	633	6%
BRAMs	32	96	33%
MULT18X18s	2	96	2%
GCLKs	1	8	12%

5. Comparison with other Steganalysis Algorithms

In this section, we compare Sigma Delta (SD) with two other effective steganalysis methods that are designed to detect LSB replacement. For this purpose, we choose RS[14] and Sample Pair (SP)[15] algorithms. For audio steganalysis, it was found out that RS performs better than both SP and SD. This may be different for other media types such as image. To perform the experiments, we used a total of 90 audio files (all 8 bits), obtained from Ref. 17. Then for a total of 2672 times, we selected at random one of the test audio files and embedded a random amount of data bits in random places of the stream. Every time we recorded the size of the embedded data and the amount that was detected by RS, SD, and SP. Figure 13 shows the obtained results

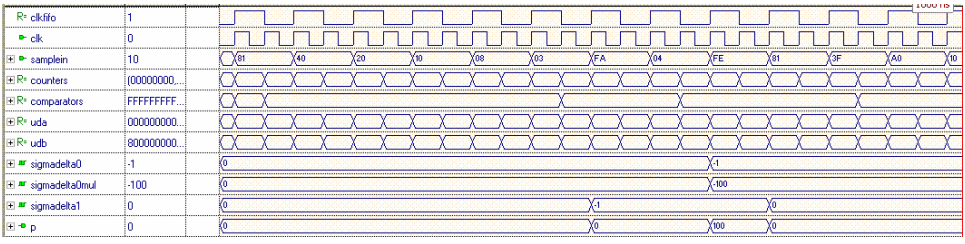
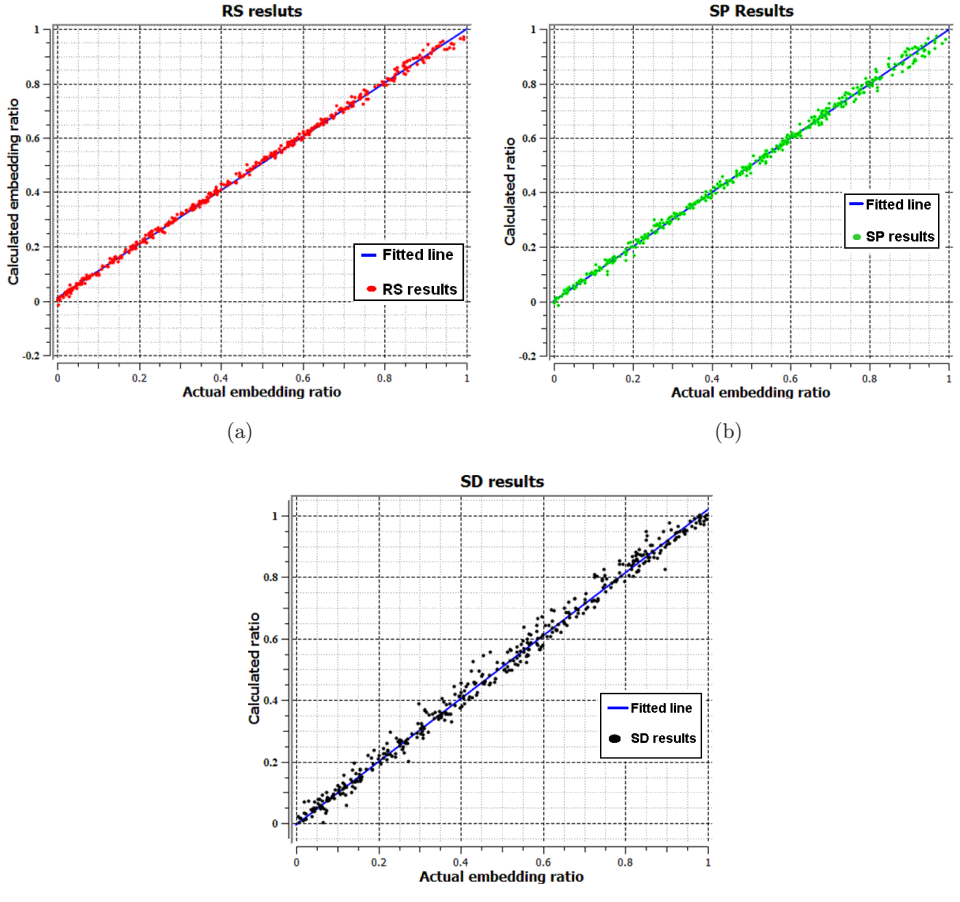


Fig. 12. Operation of a Sigma Delta () block for 12 incoming audio signals.



(a) (b)

(c)

Fig. 13. Results of RS and SP and SD. (a) RS results. (b) SP results. (c) SD results.

Table 2. Mean and variance of error and number of unsuccessful results.

Name	Mean of error	Variance of error	Unsuccessful results	Mean of absolute error	Variance of absolute error
RS	0.005	0.001	5.7%	0.01	0.0009
SP	−0.001	0.0002	5.3%	0.01	0.0001
SD	0.007	0.0007	0%	0.02	0.0003

from RS, SP and SD steganalysis methods. We see that all three steganalysis method are accurate while the precision of RS is the highest for the tested audio streams.

Table 2 shows the mean and variance of error for each attack. Also the number of unsuccessful results is shown in Table 2.

As we see, the accuracy of SD is comparable with RS and SP. There are situations that RS and SP do not produce any results. On the other hand, Sigma Delta always produces a result. When using RS and SD, about 5−6% of cases produce no results. Although the precision of SP and RS are higher than SD, the ease of hardware implementation makes SD a more attractive algorithm.

6. Scalability of Sigma Delta

So far we only mentioned 8-bit audio signals. Other types of audio files with 12-bit or 16-bit signals are widely used. In this section, we want to show that Sigma Delta can be applied to these audio signals too.

Generally, the proposed method is independent of the signals bit resolution. We implemented a 16-bit version of Sigma Delta and tested it with some samples. Figure 14 shows the result of these experiments. Random numbers of data bits are embedded in random files. Then SD is applied to the audio stream and results are recorded.

Implementing the 16-bit version in hardware requires more assets than the 8-bit version. Large FIFOs are required and the number of comparators and counters should increase to 65536. To alleviate this problem, we suggest ignoring some of the more significant bits of the samples. We experimented with a number of different bit resolutions and found out that ignoring some of the higher order bits of each signal would have minor effect in the precision of Sigma Delta. Same experiments as explained in Sec. 5 were performed but with a variable number of sample bits. Table 3 shows the result of this experiment for number of bits varying from 2 to 8 bits.

As we see, with ignoring more bits of the most significant part of the signals, the algorithm holds its accuracy while its precision degrades. The advantage of ignoring most significant bits is the reduction in the amount of required hardware for implementing *Sigma Delta*. If the audio stream under analysis has 16-bit samples then the same reduction technique can be applied. We performed a set of experiments on 16-bit audio streams and varied the number of tested bits from 16 to 8. The results of these experiments are listed in Table 4.

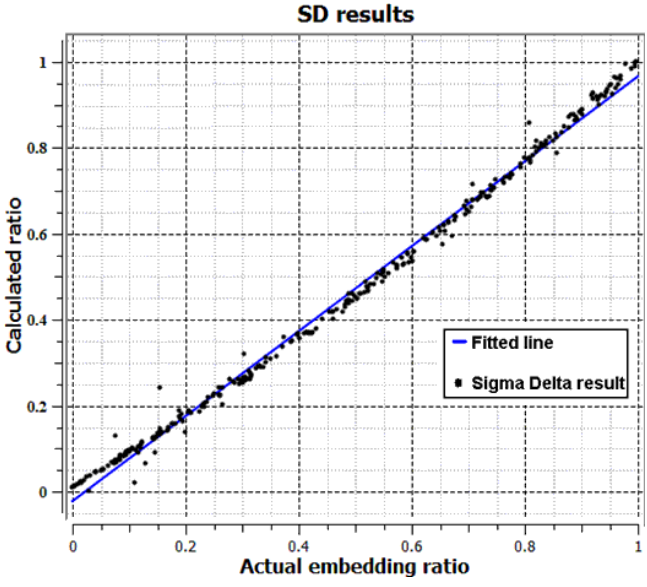


Fig. 14. Results of SD attack on 16 bit audio files.

Table 3. The result of ignoring some most significant bits.

Number of LSB bits	Mean of error	Variance of error
8	0.007	0.0007
7	0.007	0.0011
6	0.004	0.0016
5	0.0007	0.0039
4	−0.004	0.0081
3	−0.009	0.0110
2	0.068	0.0393

Table 4. The result of ignoring some most significant bits in 16 bit audio files.

	Mean of		Variance	
Number of tested bits	Error	Absolute error	Error	Absolute error
16	−0.02869	0.032067	0.000576	0.000370
15	−0.02789	0.031426	0.000552	0.000342
14	−0.02853	0.031196	0.000453	0.000293
13	−0.0293	0.030936	0.000372	0.000273
12	−0.02744	0.030360	0.000462	0.000292
11	−0.02484	0.031027	0.000827	0.000481
10	−0.02311	0.030630	0.000799	0.000393
9	−0.01569	0.025484	0.000705	0.00030
8	−0.01533	0.026332	0.000815	0.000355

The reduction of tested sample bits would not affect the overall accuracy of the algorithm but the precision of the steganalysis is slightly reduced. Therefore, it is deduced that *Sigma Delta* can be scaled for different sample resolutions of 8, 12, or 16 bits. The amount of hardware for any resolution can be reduced by testing only the lower order bits of the audio stream signals.

7. Conclusions

Audio stream is an easy medium to embed data in. Online nature of audio communication makes LSB flipping still an effective steganographic method. High volume of communication can make offline steganalysis of audio signal an impossible task. In this paper, a steganalysis algorithm was presented which extracts statistical parameters from the histogram of a media file. The extracted parameters are linearly dependent on the amount of embedding. Analytical proof of the claim was provided too. The algorithm was designed to have low computational complexity in order to be hardware implementable. A novel architecture was then presented for the real-time application of the algorithm. We presented a strategy that performs statistical analysis on incoming audio signals with very low hardware utilizations. Our implementation further reduced the computational complexity of the algorithm and performed the histogram analysis by a number of simple counters. The implementation results proved the functionality and accuracy of the proposed steganalysis and its hardware implementation. Using this technique, we could find the embedding ratio in audio streams. The proposed strategy is applicable for natural 8-bit mono audio streams which have embedded data placed in the least significant bit in each sample. The algorithm and its hardware implementation can be adapted for compressed data streams or embedding in transfer domains too.

Two powerful steganalysis methods of RS[14] and SP[15] were used for comparison purposes. It was shown that the accuracy of the proposed method is comparable with these algorithms while the advantage of our method is its ease of hardware implementation. The proposed method can be applied to non-8-bit audio streams of 12- or 16-bit signals. This can be done by scaling of the hardware. To reduce the amount of the required hardware in such cases we showed that the MSB bits of the samples could be ignored. We showed that ignoring these bits would have minor effects on the accuracy of the method and causes major reduction in the amount of hardware.

References

1. T. M. M. Cedric, R. W. Adi and L. Mclougblin, Data Concealment in Audio Using a Nonlinear Frequency Distribution of PRBS Coded Data and Frequency Domain LSB Insertion, Vol. 1 (TENCON, 2000), pp. 275−218.
2. K. Gopulan, Audio steganography using bit modification, *Proc. Int. Conf. Multimedia and Expo*, Vol. 1 (ICME, 2003), pp. 629−632.

3. T. Holotyak, J. Fridrich and D. Soukal, *Stochastic Approach to Secret Message Length Estimation in ±k Embedding Steganography* (Security, Steganography, and Watermarking of Multimedia Contents, 2005), pp. 673−684.
4. A. D. Ker, Steganalysis of LSB matching in grayscale images, *Signal Process. Lett.* **12** (2005) 441−444.
5. S. S. Agaian, D. Akopian, O. Caglayan and S. A. D'Souza, Lossless adaptive digital audio steganography, *Proc. 29th Asilomar Conf. Signals, Systems and Computers* (2005), pp. 903−906.
6. X. Huang, R. Kawashima, N. Segawa and Y. Abe, Design and implementation of synchronized audio-to-audio steganography scheme, *Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing* (2008), pp. 331−334.
7. N. Cvejic and T. Seppanen, A wavelet domain LSB insertion algorithm for high capacity audio steganography, *Proc. Digital Signal Processing Workshop* (2002), pp. 53−55.
8. S. Dumitrescu and X. Wu, A new framework of LSB steganalysis of digital media, *IEEE Trans. Signal Process.* **53**(10, Part 2) (2005) 3936−3947.
9. S. Lyu and H. Farid, Steganalysis using higher-order image statistics, *IEEE Trans. Inform. Forensics and Security* **1** (2006) 111−119.
10. M. Goljan, J. Fridrich and T. Holotyak, New blind steganalysis and its implications, *Proc. Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, Vol. 6072 (San Jose, Canada, 2006), pp. 1−13.
11. A. Ker, A general framework for structural steganalysis of LSB replacement, *Proc. 7th Information Hiding Workshop*, Vol. 3727 of LNCS (Springer, 2005), pp. 296−311.
12. A. Westfeld and A. Pfitzmann, *Attacks on Steganographic Systems*, Lecture Notes in Computer Science, Vol. 1768 (Springer, Berlin, Heidelberg, 2000), pp. 61−67.
13. N. Provos and P. Honeman, *Detecting Steganographic Content on the Internet*, Vol. 1 (CITI Technical Report, 2001), pp. 1−11.
14. J. Fridrich, M. Goljan and R. Du, Reliable detection of lsb steganography in color and grayscale images, *Proc. ACM Workshop on Multimedia Security* (2001), pp. 27−30.
15. S. Dumitrescu, X. Wu and Z. Wang, Detection of LSB steganography via sample pair analysis, *IEEE Trans. Signal Process.* **51** (2003) 1995−2007.
16. S. M. Ross, *Stochastic Processes* (John Wiley and Sons, New York, 1996).
17. http://www.voiceage.com/other.php.