British Journal of Mathematics & Computer Science
5(2): 190-203, 2015, Article no.BJMCS.2015.014
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

A Real Time Secured Voice Framework Using the Totally Reconstructed Discrete Wavelet Transformation

Ahmad Azzazi[1*]

[1]*Computer Information Systems Department, Applied Science University, Amman, Jordan.*

Article Information

DOI: 10.9734/BJMCS/2015/11626
Editor(s):
(1) Tian-Xiao He, Department of Mathematics and Computer Science, Illinois Wesleyan University, USA.
Reviewers:
(1) M. Lokesha, Mechanical and Industrial Engineering, Caledonian College of Engineering, Muscat, Oman.
(2) Anonymous, King Saud University, Saudi Arabia.
(3) Anonymous, Mepco Schlenk Engineering College, Sivakasi, India.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=726&id=6&aid=6704

Original Research Article

Received: 26 May 2014
Accepted: 14 October 2014
Published: 29 October 2014

Abstract

In our life today and with the upraise of the mobile computation and mobile phones we are using more voice applications. The voice application could then send voice information with different information over different communication channels, with an increasing need to protect this information without any delay when applying any modification on it. To protect this voice information, we encrypt it. In this paper, we are proposing a Framework using the totally reconstructed Discrete Wavelet Transformation (DWT) to encrypt the voice information in real time, where the sample rate of the encrypted voice signal is equal to the original voice signal sample rate. The framework is designed and tested with sample tests to measure the performance of this Framework, showing the speed and complexity of this Framework.

Keywords: Discrete wavelet transformation, sound/ voice encryption, real time application.

1 Introduction

There has been increasing interest in research involving the security of information systems. In in recent years the number of computer attacks increased dramatically. With the use of the internet and smart mobile devices, there is a need to secure the communication channels against possible threats. Securing voice communication became a very important Challenge to information technology researchers to find suitable ways in protecting this type of information.

**Corresponding author: a_azzazi@asu.edu.jo;*

In the world of new technologies we have more and more applications depending on sounds/voices. A variety of modern applications are taking sounds/voices as input, storing, processing and transferring it over different communication channels. The new era with the mobile computing gives us many challenges to handle the different multimedia elements. With sounds and voices as a major and important input to these devices, there is a need to develop the right applications for it. This shows us the need for fast applications if applied on mobile devices, and then the user of a mobile device should not wait when doing the application. This means that there is a need of a real time application and application processing. Therefore, if we develop applications for such type of devices or other devices we need not only to think about the application itself but also about the processing speed of the developed application. In real time application, we have to develop the application with results appearing to the user or other systems without any delay. Therefore, since we have to choose carefully the algorithm with a minimum time complexities and fast execution times. Another point of view we are considering is that the sound or voice we are using is transferred over different communication channels, therefore we using these channels we are facing different threads, we give us the need of securing the sound/voice over these channels. In our paper we will use a methodology of encrypting/ decrypting the sound/voice, we the property of having all things done in real time.

The rest of this paper is organized as follows: literature review and related work are described in Section 2. Section 3 describes the Mechanism of the proposed method. In section 4 a sample encryption of the proposed method is shown. Section 5 shows the performance analysis of the proposed method. Finally, Conclusions and future Work are drawn in section 5.

2 Related Works

Secure voice/ sound had been studied in many published papers; those papers discussed many secured methods, technologies and challenges. The encryption mechanism of voice / sound, which consists of the different filter banks as keys for the different stages of the filtering process to encrypt the voice signal.

There are many ways to secure voice, one of these methods is to scramble the voice [1,2], where one have to transpose or invert the voice signal. Another way is to make the so called Base band inversion [3] or phase inversion, with it the spectrum is inverted at a single preset never changing frequency. One further way to deal with secure voice is the Amplitude modification [3], where it is not really changing the original signal. Among further methods is the use of the linear predictive coding (LPC) [4]. The application of cryptographic algorithm to voice parameters results as clearly encrypted voice signal.

One of the pioneered concepts to secure voice was the SIGSALY System (X System, Project X, Ciphony I) developed by Bell Labs [5]. It was a secure voice system used in the Second World War for the highest-level Allied. This concept was built by adding noise to voice signals to prevent listening to the original voice signals by the enemy forces.

After that the United States Department of Defense uses the MELP or enhanced-MELP (Mixed Excitation Linear Prediction). The MELP is a voice/speech coding standard [5]. Another standard for speech coding was created by Texas Instruments known as MIL-STD-3005 [5].

Many Encryption Systems based on the hardware were built, where they are voice encryption schemes that use hardware to encrypt the speech without the use of a computer. They are used in Cell Phones and Radio to make them more secure. Hardware encryption systems are relatively expensive and difficult to acquire.

Siddeeq Y. Ameen et al. [6] showed in their work that using a segmental spectral signal-to-noise ratio to test the performance of their proposed system. They concluded that the built system has a high security performance with a distinguish between male and female speakers.

Another related work is the work of Tin Lai Win and Co. [7]. They presented in their work an overview of the symmetric cryptography. They used the Linear Feedback Shift Register (LFSR) to examine the Voice over IP Security. They found that the Linear Feedback Shift Register (LFSR) algorithm is suitable for encrypting/decrypting of streams of data.

Jay M. Joshi et al. [8] added in their work another security level to the existing security standard for mobile communication. This added security level was done using the Discrete Wavelet Transformation. They showed a comparison between their approach and existing approaches for mobile communication. They also suggested a Hardware implementation of their approach. They showed that their proposed method with the allowed mobile communication delay standards and their method need only 20% of CPU load, a maximum size of about 15k bytes, a minimum size of about 13k bytes and other capacity results.

Hemlata Kohad et al. [9] discussed in their work the different methods used for the encryption and decryption of the speech signals. Among the discussed methods are the Time Domain Scramblers, the Frequency Domain Scramblers and the Pseudo Noise Sequences Encryption. They showed also the different modes of encryption systems. They concluded that the Speech encryption using a method called the Kasami sequence is better with the time domain and frequency domain as they compared with the other pseudorandom noise methods. They showed also that enhancement of the security of Speech encryption is possible using the kasami method.

3 The Mechanism of the Proposed Method

A wavelet is from its name a small wave, which oscillates beginning at the zero value of the amplitude, then increasing, and at the end decreasing towards zero value. Wavelets are considered as functions, which can act as a Bandpass filter with a variation between high frequency and low frequency [10].

The wavelet ψ has the following property:

$$\int_{-\infty}^{+\infty} \Psi \, dt = 0 \tag{1}$$

There are many types of commonly used wavelets, among them the Haar wavelet, Daubechies wavelets, Symlets and many others [11]. Fig. 1 shows a sample of a wavelet signal.

The Discrete Wavelet Transform (DWT) is a wavelet transform, where the wavelets are discretely used, with the advantage that it can capture the frequency information and the time information of the signal (location information) [12].

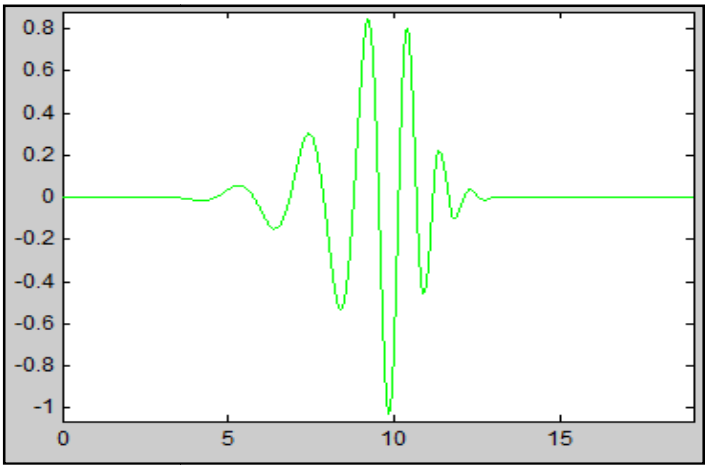

Fig. 1. Wavelet example

The Discrete Wavelet Transformation is totally reconstructed without any loss of information [12].

The Multiscale decomposition analysis is composed of a discrete time low pass and band pass filter banks. It divides the discrete time signal into a high band and a low band one. The signal x is down sampled by the factor of two, Fig. 2.

$$y_{\text{low}} = (x * g) \downarrow 2 \tag{2}$$

$$y_{\text{high}} = (x * h) \downarrow 2 \tag{3}$$

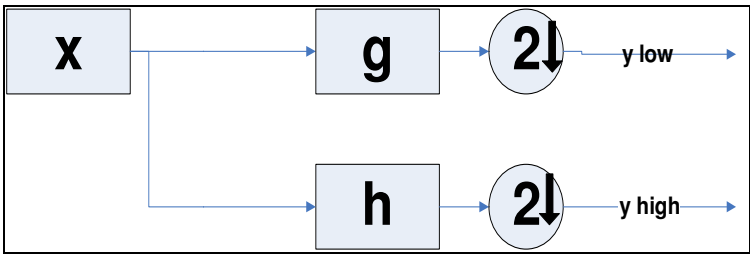

Fig. 2. The multiscale decomposition analysis with DTW

This process is repeated with the low band output, where the fast discrete wavelet transform produces a stream of samples octaves. Half the sampling rate, such a one again halved sampling rate for the 1 octave lower lying bound.

The proposed voice encryption method steps are shown in Fig. 3.

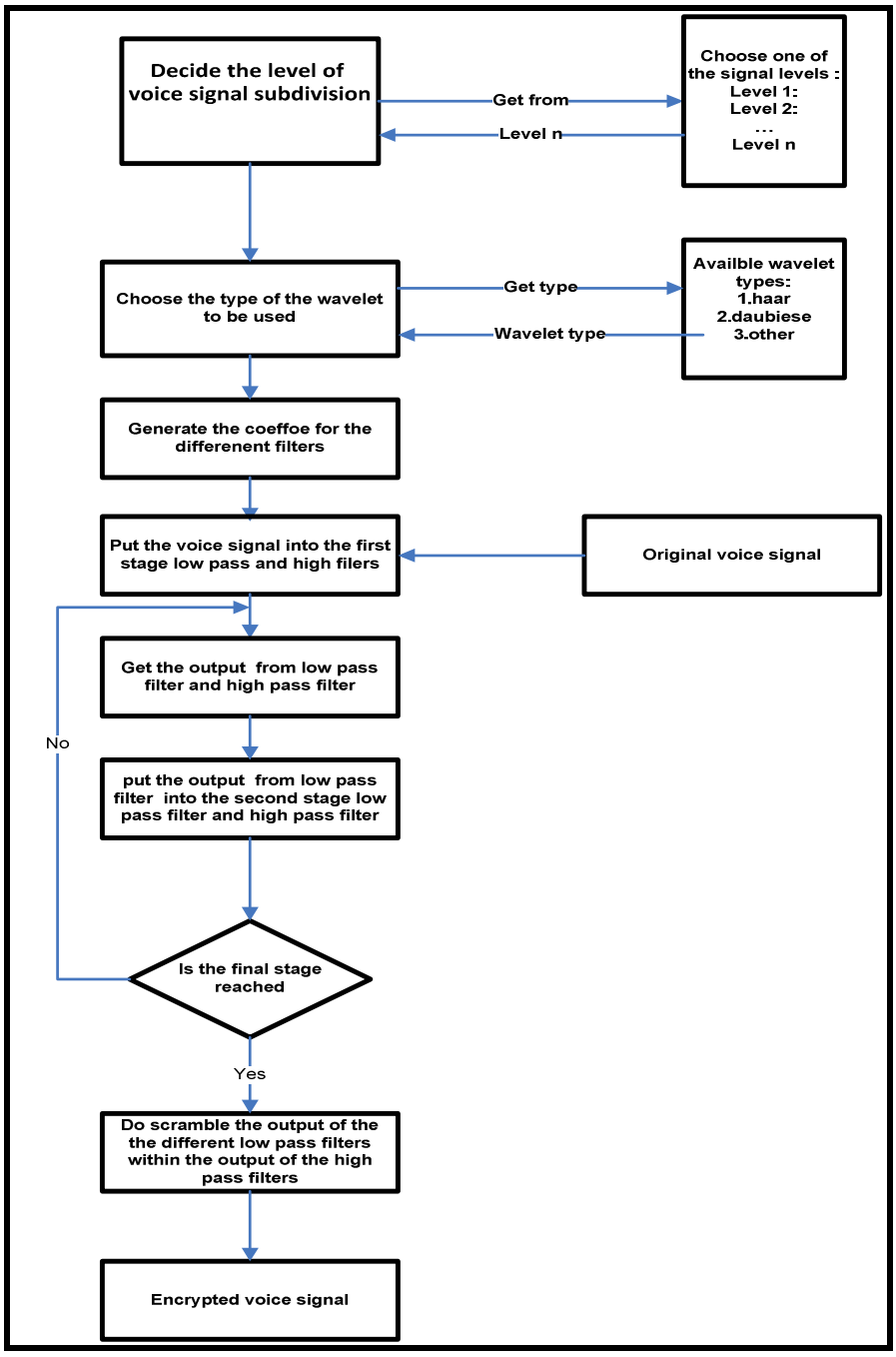

Fig. 3. The proposed voice encryption method

At the first step of the proposed method one decides about the level of voice signal subdivision which means the number of octaves to be used (which is equal to n). In this step the voice signal is divided in different sub signals, where the signal level should be selected from the available signal levels (namely n levels). The more the value of this selection is, and then more sub signals are produced, which makes the encryption more efficient.

The next step is to choose the type of the wavelet to be used for the discrete wavelet transformation from a given list of the available wavelet types like the Haar wavelet, Daubechies wavelets, etc. For more encryption complexity one can develop his own wavelets and then add them to the available wavelet list.

After this step the generation of the coefficients for the different filter banks is done. These coefficients are stored as a vector for further processing.

We put then the original voice signal into the first stage of the low pass and high filters.

At the next step we get the output from low pass filter and high pass filter, where the output of the high pass filter is stored.

The output from low pass filter is inputted again like the original voice signal into the second stage low pass filter and high pass filter. The last two steps are repeated until the required level of sub signal is reached.

If the final stage reached, then we do scramble the output of the low pass filter within the output of the different high pass filter. After all, we have the encrypted voice signal with the same sample rate as the original voice signal. This encrypted voice signal is then transmitted through an appropriate transmission channel.

The different parameters (n levels, type of the wavelet, filter banks parameters) are acting as the key for this encryption method.

4 A Sample Encryption of the Proposed Method

We used in this work the Matlab program to do all the programming needed for this research.

In this section, we are showing an example of the model.

At the production of the voice signal we have a continuous representation of the voice signal with a common sampling rate like 8 kHz, see Fig. 4.

At the first step we have to choose the level of voice signal subdivision which means the number of octaves to be used (namely n). In this example we will choose n equal to three, therefore we should have 4 different signal outputs like in Fig. 5.

The next step is to choose the type of the wavelet to be used for the discrete wavelet transformation from the given list of the available wavelet. For this example we choose a Daubechies wavelet with 8 filter coefficients (see Figs. 6 and 7), where the filter coefficients are generated from a library containing all the possible used wavelet types and their coefficients.

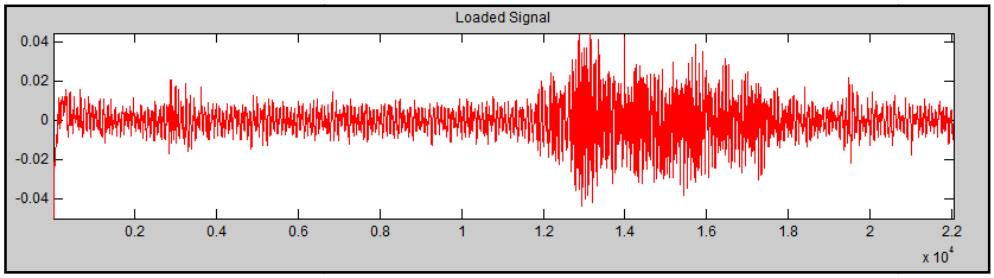

Fig. 4. Original voice signal

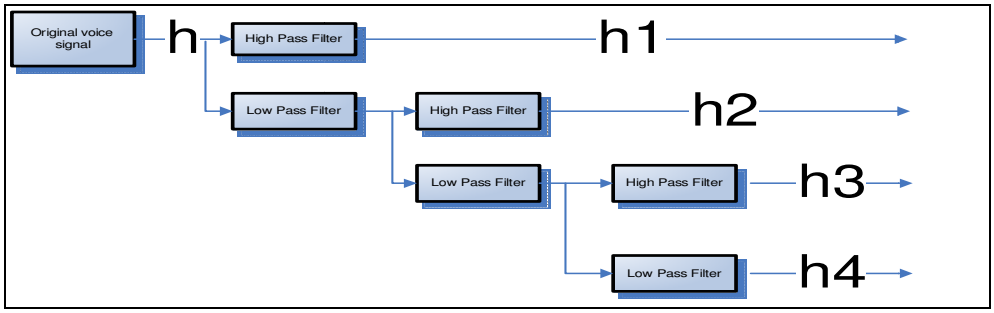

Fig. 5. The 4 produced sub signals from the original voice signal by choosing n=3

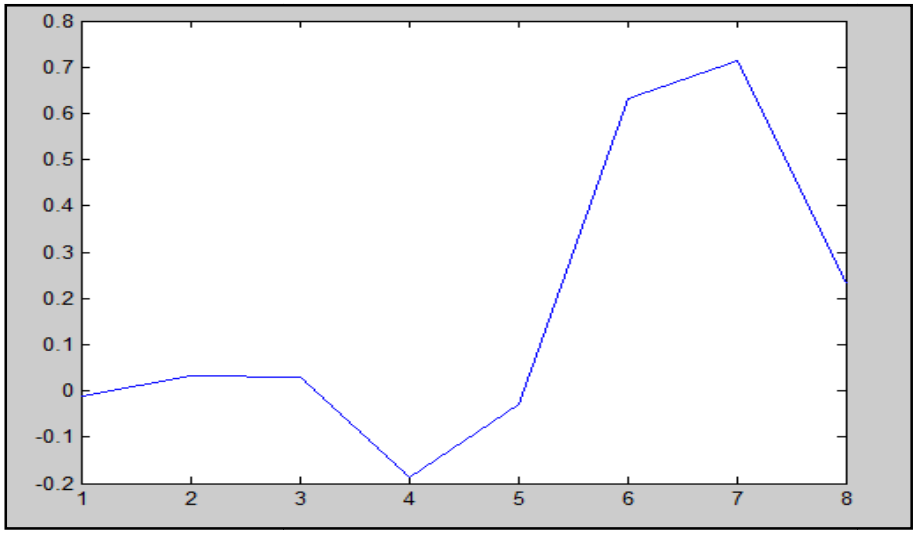

Fig. 6. Generated low pass filter

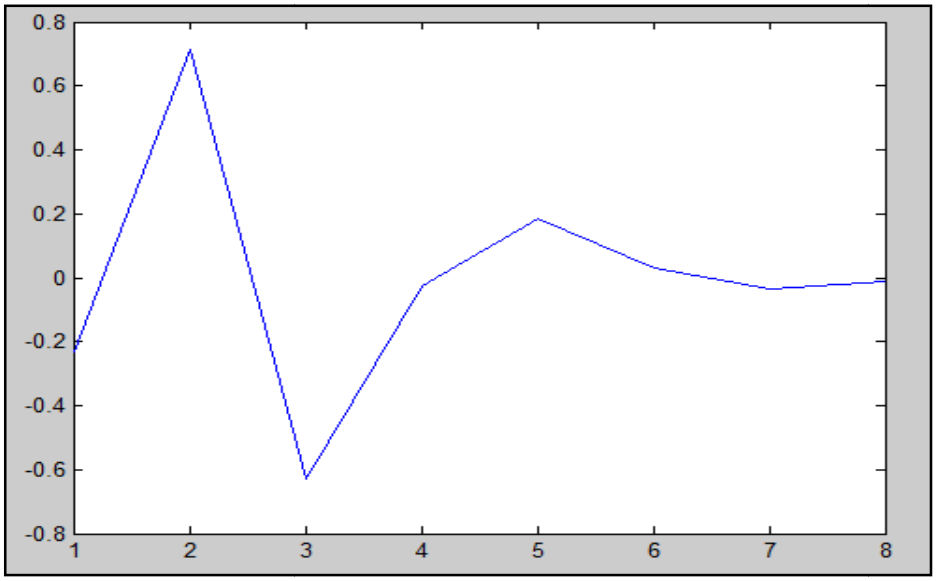

Fig. 7. Generated high pass filter

The generation of the different coefficients for the 6 different filter banks is done. Then we put the original voice signal into the first stage of the low pass and high filters and the output from low pass filter is putted into the second stage filtering repeatedly until we reach the final level, having the following outputs shown in Fig. 8.

h1	**h1**	**h1**	**h1**	**h1**	**h1**	**h1**	**h1**	**h1**	**...**
h2		**h2**		**h2**		**h2**		**h2**	**...**
h3				**h3**				**h3**	**...**
h4								**h4**	**...**

Fig. 8. Different outputs for different signal levels with different sample rates.

In Fig. 9, we have the original signal "h" and the 4 generated output signals (h1, h2, h3, h4).

If we have an original voice signal with the sample rate k= 8k Hz, then we have then the output signals having the following sample rates: h1 with the sample rate k/2= 4 kHz, h2 with the sample rate k/4= 2 k Hz, h3 with sample rate k/8 = 1 kHz and h4 = 1 kHz with the sample rate k/8. Adding all the sub signals' sample rates together gives the original sample rate, namely 8 kHz.

The next step is to scramble the four signals, where the scrambling mechanism is shown in Fig. 10.

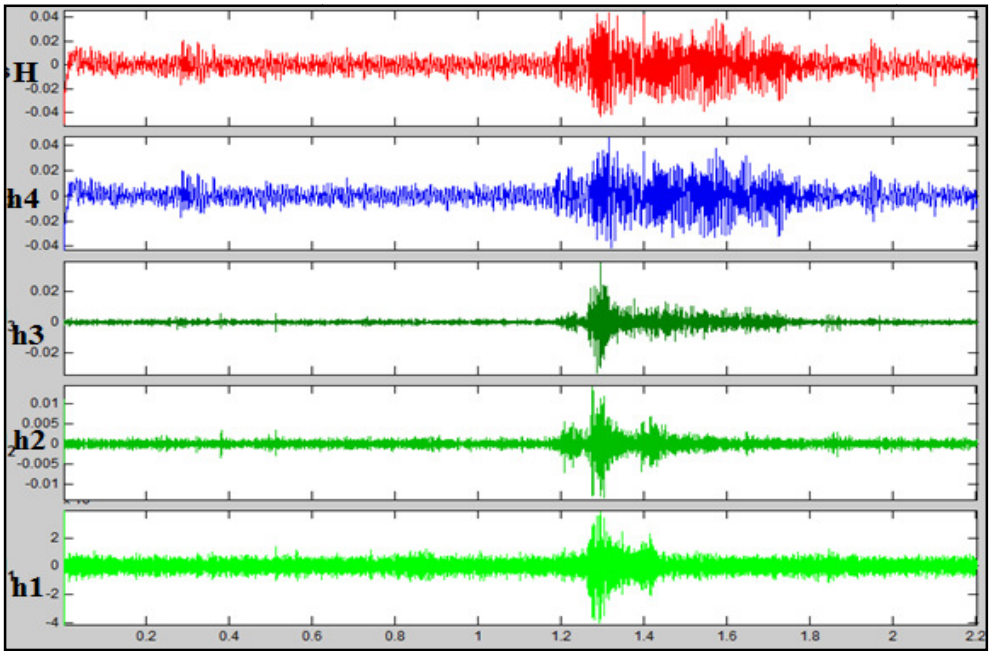

Fig. 9. Original signal "h" and the 4 generated output signals (h1,h2,h3,h4).

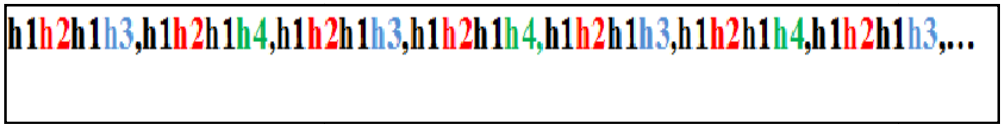

Fig. 10. Scrambling the different output signals

This is then the final scrambled signal with the originals sampling rate to be transmitted (Fig. 11).

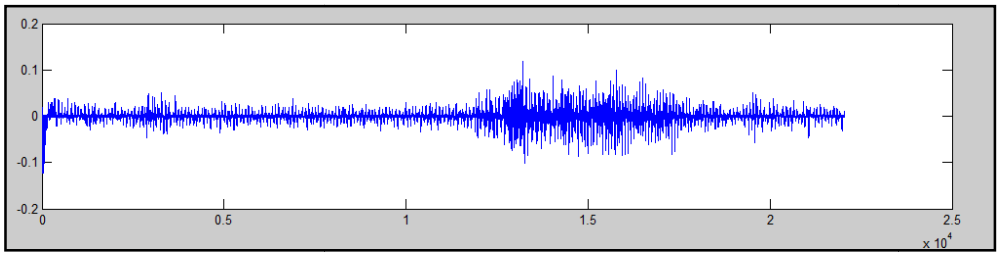

Fig. 11. Scrambled signal with the original sampling rate to be transmitted

We do scramble the different 4 signals' output in a way, that if a signal output is ready, then it should be transmitted. In such a way, we are gaining the time, so our entire framework for the

encryption should take only the length of the filter operation, which dependences on the length of the filter coefficients.

The filter operation multiplies each filter coefficient with a signal sample and adds all the multiplication with each other. This entire operation takes a cycle length equal to the coefficient number of the wavelet filter. Such an operation is on most today available processors optimized, so there will be no problem to generate the output without any delay, so the encrypted signal is then in real time generated.

The first 50 values of the original signal, the different output signals and the encrypted, are shown in the tables below. Tables 1, 2, 3, 4, 5, 6 are showing the first 50 values of the different signals produced from the sample encryption.

Table 1. The first 50 values of the original signal

-0.0277	-0.05	-0.0443	-0.0468	-0.0454	-0.0439	-0.0429	-0.0418	-0.0419	-0.0403
-0.0391	-0.0391	-0.0374	-0.0368	-0.0356	-0.0343	-0.032	-0.0316	-0.0328	-0.0313
-0.0301	-0.0293	-0.0276	-0.0266	-0.0256	-0.0246	-0.0269	-0.0259	-0.0239	-0.0216
-0.0202	-0.02	-0.0211	-0.0218	-0.0213	-0.0203	-0.0196	-0.0206	-0.0214	-0.0229
-0.0224	-0.0217	-0.0221	-0.0216	-0.022	-0.0219	-0.021	-0.0213	-0.0219	-0.0205

Table 2. The first 50 values of the signal "h1"

0.0028	0.0066	-0.0004	-0.0003	-0.0007	0.0004	0.0005	0	0.0007	-0.0014
0.0005	0.0002	-0.0001	-0.0016	0.0008	-0.0002	-0.0005	0.0004	0.0003	0
0.0005	-0.0005	-0.0002	0.0006	-0.0008	0.0001	0.0007	-0.0002	-0.0002	-0.0003
0.0007	0.0001	0	-0.0003	-0.0001	-0.0003	0.0005	-0.0003	0	0.0004
0.0003	-0.0003	-0.0006	0.0006	-0.0001	-0.0005	0.0006	0.0003	-0.0006	-0.0003

Table 3. The first 50 values of the signal "h2"

0.0061	0.0191	-0.004	0.0013	-0.0005	0.0012	-0.0009	0.0013	-0.0021	0.0006
0.0014	-0.0018	0.0003	-0.0006	0.0017	-0.0011	0.0007	0	0.0003	-0.001
0.0011	-0.0007	-0.0008	0.0013	-0.0023	0.0004	-0.0009	0.0017	-0.0005	-0.0008
0.0009	0.0012	0.0003	0.0019	-0.0001	0.0002	0.0002	-0.0011	0	-0.0012
0.0022	-0.0002	0.0003	-0.0004	-0.0006	0.0012	0.0014	-0.0005	-0.0002	0.0006

Table 4. The first 50 values of the signal "h3"

-0.0037	-0.0123	-0.0011	0.0022	-0.0012	-0.0024	-0.0002	-0.0007	0.0006	-0.0008
-0.0002	0.0013	0.0024	0.0006	0.0009	-0.0016	0.0007	-0.0047	0.0043	-0.0008
0	-0.0031	0	0.0012	0.0043	-0.0034	0.0012	-0.0025	0.0022	0.001
-0.0003	0.0018	0.001	-0.0002	0.0015	0.0021	0	-0.002	0.002	0.0038
0.0022	-0.0009	0.0016	-0.0025	0.0004	-0.0019	-0.0006	-0.0006	0.0003	-0.003

Table 5. The first 50 values of the signal "h4"

-0.1198	-0.1206	-0.1227	-0.1222	-0.1194	-0.1225	-0.1025	-0.0824	-0.0624	-0.0606
-0.0617	-0.0466	-0.0349	-0.0301	-0.0242	-0.0304	-0.0257	-0.0049	0.0044	0.0025
0.0028	0.0155	0.0292	0.0156	0.0074	-0.0081	-0.0053	0.0239	0.0256	0.0254
0.0243	0.0273	0.0299	0.0292	0.0193	0.008	0.0156	0.0156	0.0141	0.0202
0.0278	0.0278	0.0386	0.0374	0.0357	0.0313	0.0087	-0.0035	-0.0129	-0.0212

Table 6. The first 50 values of the encrypted signal

0.0031	0.0061	0.0046	-0.0037	-0.0004	0.0191	-0.0072	-0.1198	-0.0007	-0.004
0.0004	-0.0123	0.0005	0.0013	0	-0.1206	0.0007	-0.0005	-0.0014	-0.0011
0.0005	0.0012	0.0002	-0.1227	-0.0001	-0.0009	-0.0016	0.0022	0.0008	0.0013
-0.0002	-0.1222	-0.0005	-0.0021	0.0004	-0.0012	0.0003	0.0006	0	-0.1194
0.0005	0.0014	-0.0005	-0.0024	-0.0002	-0.0018	0.0006	-0.1225	-0.0008	0.0003

At the end of this section, we are showing a sample of the used Matlab code in Fig. 12.

```
Fs = 8000;
y = wavrecord (2*Fs, Fs, 'double');
[LO_D,HI_D,LO_R,HI_R] = WFILTERS ('db4') ;
[A1,D1] = dwt(y,LO_D,HI_D);
[A2,D2] = dwt(A1,LO_D,HI_D);
[A3,D3] = dwt(A2,LO_D,HI_D);
zero_p=1;
D1_new=zeros(1, (zero_p+1)*length(D1)-zero_p);
D1_new(1:(zero_p+1):end)=D1;
D1_new=D1_new';
zero_p=3;
D2_new=zeros(1, (zero_p+1)*length(D2)-zero_p); D2_new(1:(zero_p+1):end)=D2;
D2_new=D2_new';
D2_new_shifted=circshift(D2_new,1);
zero_p=7;
D3_new=zeros(1, (zero_p+1)*length(D3)-zero_p);
D3_new(1:(zero_p+1):end)=D3;
D3_new=D3_new';
D3_new_shifted=circshift(D3_new,3);
zero_p=7;
A3_new=zeros(1, (zero_p+1)*length(A3)-zero_p); A3_new(1:(zero_p+1):end)=A3;
A3_new=A3_new';
A3_new_shifted=circshift(A3_new,7);
 y_new=D1_new(1:22050)+D2_new_shifted(1:22050)+
D3_new_shifted(1:22050)+A3_new_shifted(1:22050);
```

Fig. 12. Sample of the used Matlab code

5. Performance Analysis of the Proposed Method

A comparison of the performance of the proposed method with the most known encryption methods is done in this section. The code for the comparison was written using the Crypto++ library, which is a free open source library for encryption algorithms. The code was running on an Intel core i5-2400 @3.1 GHz Processor under windows 7 with 4G RAM.

The algorithms compared with our method are the Rijndael (AES) algorithm, the 3DES algorithm, DES algorithm, Blowfish algorithm and Whirlpool algorithm.

The Rijndael (ASE) Algorithm is the Advanced Encryption Standard. The mode used in this study is the ASE-CCM for this Algorithm (counter mode of encryption). It is an encryption algorithm with solid authentication and confidentiality mechanisms. This CCM mode is designed for blocks of the length of 128 bits.

The Whirlpool algorithm consists of a hash function. It is a modification of the Advanced Encryption Standard (AES). It can take messages of length not greater than 2256 bits.

The DES (Data Encryption Standard) algorithm is one of the most known symmetric encryption algorithms. The mode used in this study is the DES-CTR (Counter Mode) mode.

The 3DES algorithm is an enhancement of DES (Triple DES).It is like the DES encryption standard but applied three times after each other.

The Blowfish algorithm is a public domain encryption algorithm, which is one of the most common used algorithms. The mode used in this study is the CTR mode.

A Data Set of 100 voice files of different sizes was created to run this comparison test, an average of ten different size categorizes was taken to summarize the results of the comparison for the performance in this paper.

Table 7 shows the results of this comparison.

Table 7. Comparison results between the proposed method the most common encryption algorithm for the execution times (in seconds)

Average data input in Kbytes	DES-CTR	ASE-CCM	3DES	Blowfish	Whirlpool	The proposed method
10	0.1251	0.2457	0.4231	0.1192	0.121	0.091
15	0.1875	0.3524	0.662	0.177	0.21	0.11
19	0.2451	0.489	0.877	0.212	0.252	0.198
30	0.4234	0.662	1.457	0.3661	0.388	0.3
46	0.6137	1.024	2.11	0.5136	0.62	0.416
57	0.846	1.254	2.92	0.7812	0.876	0.503
76	1.312	1.7845	3.654	0.881	0.903	0.732
129	2.141	3.145	6.27	1.812	1.912	1.232
170	2.712	4.124	7.814	2.211	2.311	1.913
230	3.817	5.563	10.254	3.012	3.452	2.87

The average execution time for the proposed method is 0.8365 seconds for the average file size of 78.2 kB. The results show that the proposed method has a better execution times than the other compared algorithms when varying the size of the encrypted file. The Blowfish algorithm comes in the second place in this comparison in terms of the processing time. It shows also that the AES-CCM is the worst one in terms of execution times for big data.

6. Conclusion and Future Work

In this work we proposed a new framework for the real time encryption and scrambling of speech signals. The new framework gets the original signal with the original sample rate a gives this signal as a scrambled one using Discrete Wavelet Transformation (DWT).

The proposed method is implemented using standard available software with the feature that the encrypted signal is retrieved in real time. The method is a good way to encrypt voice for real time applications. The original voice signal is preserved and totally reconstructed. The results show that the proposed method has a better execution times than the other compared algorithms when varying the size of the encrypted file.

Acknowledgments

The author is grateful to the Applied Science Private University, Amman, Jordan, for the Full Financial support granted to this research project.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Matsunaga A, Koga K, Ohkawa M. An analog speech scrambling system using the FFT technique with high-level security. IEEE Communication. 1989;7:4.

[2] Goldburg B, Sridharan S, Dawson E. Design and cryptanalysis of transform-based analog speech scramblers. IEEE Communication. 1993;11:5.

[3] Beker HJ, Piper FC. Secure speech communications. Academic, Ondlon, UK; 1985.

[4] Jeremy Bradbury. linear Predictive Coding; 2000.

[5] William R. Bennett. Secret telephony as a historical example of spread-spectrum communications. IEEE Transactions on Communications. 1983;31:1.

[6] Siddeeq Y. Ameen, Abbas A. Al-Shalchi, Muhanad D. Al-Bayati. Design and hardware implementation of a speech cipher system. Journal (NUCEJ). 2007;10:1.

[7] Tin Lai Win, Nant Christina Kyaw. Speech Encryption and decryption using Linear Feedback Shift Register (LFSR). World Academy of Science, Engineering and Technology. 2008;2.

[8] Jay M. Joshi, Upena Dalal. Hardware implementation of wavelet based speech encryption for end-to-end security in mobile communication system. Int. J. of Recent Trends in Engineering and Technology. 2010;4:4.

[9] Hemlata Kohad, Ingle VR, Gaikwad M. An overview of speech encryption techniques. International Journal of Engineering Research and Development. 2012;3:4.

[10] Paul S. Addison. The illustrated wavelet transform handbook. Institute of Physics; 2002. ISBN 0-7503-0692-0.

[11] Martin V, Herley C. Wavelets and filter banks: Theory and design. Signal Processing, IEEE Transactions; 1992.

[12] Stephane M. A Wavelet Tour of Signal Processing; 1998.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=726&id=6&aid=6704