A Covert Communication Model Based on Least Significant Bits Steganography in Voice over IP

Hui Tian[1], Ke Zhou[1], Yongfeng Huang[2], Dan Feng[1], Jin Liu[1]
[1]*Wuhan National Laboratory for Optoelectronics, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China*
[2]*Network and Human-machine Communication Research Institute, Department of Electronic Engineering and Technology, Tsinghua University, Beijing, China*
{huitian, jinliu}@smail.hust.edu.cn, {k.zhou, dfeng}@hust.edu.cn, yfhuang@tsinghua.edu.cn

Abstract

Steganography, as one of alternative techniques for secure communications, has drawn more and more attentions. This paper presents a covert communication model based on least significant bits (LSB) steganography in Voice over IP (VoIP). The model aims at providing nice security of secret messages and real-time performance that is vital for VoIP. Therefore, we employ a simple encryption of secret messages before embedding them. This encryption strikes a good balance between adequate short-term protection for secret messages and low latency for VoIP. Furthermore, we design a structure of embedded messages. It can provide flexible length and avoid effectually both extraction attack and deceptive attack. We evaluate the model with ITU-T G.729a as the codec of the cover speech in StegTalk, our platform for study on covert communications theory in VoIP. In this case, the proposed model can provide two optional covert transmission speeds, i.e. 0.8kb/s and 2.6 kb/s, where the maximum payload ratio is 99.98%. The experimental results show that our method has negligible effects on speech quality and well meets the real-time requirement of VoIP.

Keywords: Covert communication, steganography, least significant bits, voice over IP.

1. Introduction

With the drastic development of applications based on Internet, secure communications have been becoming increasingly important. As one of the alternative techniques, steganography, an art and science of information hiding has drawn more and more attentions. It provides a secure communication by embedding secret messages into digital media and making them unconspicuous for eavesdroppers. In contrast with traditional cryptographies, whose purpose is to hide the content of secret messages being exchanged between two communicating parties, the purpose of steganography is to hide not only the content but also its very existence. Therefore, it can provide a securer protection for secret messages in certain ways.

In recent years, steganography has been carried out on image [1], video [2], audio [3], text [4], etc. The techniques with steganography at the core have been applied popularly and successfully in covert exchange of information, copyright protecting, etc. However, the most of previous steganography works are carried out on storage cover media [5], and by contrast with them the area of steganography in real-time systems remains mostly unexplored. In fact, due to their instantaneity, real-time systems can often provide a better security for hidden messages.

Voice over IP (VoIP) is a typical real-time system that allows users to make telephone calls via a broadband Internet connection. In recent years, due to its advantages of low cost and flexible advanced digital features, VoIP has become a popular alternative to public-switched telephone network (PSTN) [6]. Many applications of VoIP technology (e.g. Skype etc.) have been developed and are currently under development. Therefore, embedding hidden messages in VoIP is a very interesting task and may become the subject of further studies [7]. Some researchers have carried out fruitful research works [7-10], though there are few studies of steganography in VoIP compared with other carriers. From them, we can learn following facts:

- VoIP is a good carrier in which steganography can be applied [7-10]. The main reasons are: 1) the conversation that is carrying on can offer a nice camouflage for secret messages, because it

978-0-7695-3398-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ICYCS.2008.394

would be considered naturally as the only data channel in VoIP by the most of observers; 2) A VoIP connection, which is often of short duration, does not give eavesdroppers enough time to detect possible abnormity.

- Least Significant Bits (LSB) steganography technique can provide a tradeoff between adequate requirement (nice security and enough capacity) for information hiding and low latency for the real-time service [7-10], so it could be used to construct a prototypical covert channel.
- It is a good choice for improving security to encrypt messages before embedding them [7, 8]. In fact, steganalysis algorithm proposed by J. Dittmannet et al. can detect direct embedding messages in voice samples with a success rate of approximately 98.60 percent, but not detect encrypted messages [7].
- Cryptography is the most popular technique for encrypting messages. However, it does not suit this case well for two reasons: 1) Because of its time-consumed character, cryptography can be responsible for the increased latency that may degrade speech quality drastically. Therefore, we often face the necessity of tradeoff between providing adequate security and low latency for real-time service. 2) The distribution of key is another consequent problem that further increases the complexity of the system. Moreover, if it is not dealt with well, it may weaken security.

Motivated by the views, this paper presents a covert communication model based on LSB steganography (LSBCCM) in VoIP. Our model aims at providing nice security of secret messages and good real-time performance that is vital for VoIP. In this model, we adopt a simple encryption of secret messages before embedding them. The encryption strikes a good balance between effective short-term protection for secret messages and low latency for VoIP. Moreover, we design a structure of embedded messages. It can provide flexible length and avoid both extraction attack (attempt to extract the secret message) and deceptive attack (attempt to replace the secret message with other message) effectually.

Furthermore, we evaluate the model with ITU-T G.729a [11] as the codec of the cover speech in StegTalk that is our platform for study on covert communication in VoIP. In this case, the proposed model provides two optional covert transmission speeds, i.e. 0.8kb/s and 2.6 kb/s, where the maximum payload ratio is 99.98%. The experimental results demonstrate that our technique has negligible impacts on the speech quality (That is to say, it provides a good shelter for secret messages), and well satisfies the real-time requirement of the VoIP system.

2. Proposed Model

2.1. Model Overview

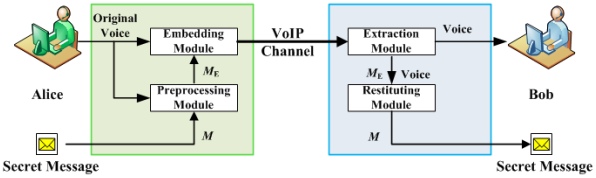

Figure 1. General covert communication model in Voice over IP

Figure 1 depicts the general covert communication model in VoIP. Let us assume that Alice (as the sender) wants to transmit a secret message to Bob (as the receiver), while they are talking about some inconspicuous topics via the VoIP system. The transmission process can be described as follows: Alice provides original voice (V) and Secret Message (M) as inputs to the Preprocessing Module (PM); after preprocessing, an embedding Message (M_E) can be obtained; M_E is the final object embedded into V by Embedding Module (EM_A); After the voice with M_E is sent through the VoIP channel, Bob extracts M_E via Extraction Module (EM_B) and retrieves M via Restituting Module (RM). In LSBCCM, EM_A just fills all binary bits of M_E into LSB of the cover speech. And EM_B, which is the inverse of EM_A, extracts all binary bits of M_E from LSB of the cover speech. They are quite clear-cut, so the rest in this section will mainly focus on the design of PM and RM.

2.2. Design of PM and RM

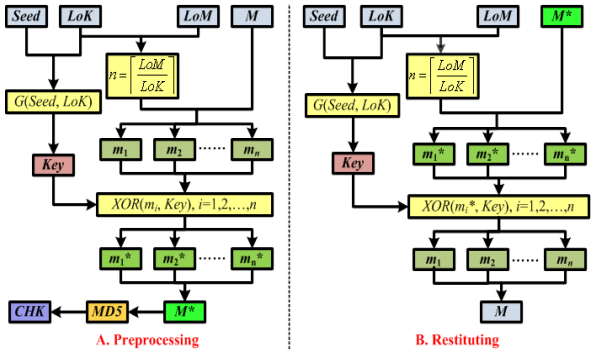

Figure 2. Preprocessing and Restituting

The purpose of PM is to hide the content of secret messages. We design a simple preprocessing that is illustrated by Figure 2-A. In the beginning, the sender must provide four parameters, i.e. *Seed*, the Length of Key (*LoK*), secret Message (*M*), and the Length of secret Message (*LoM*). *Seed* and *LoK* are the inputs of the key generator denoted by *G*(*Seed*, *LoK*) . The key generator can produce a pseudo random key with

arbitrary length. There are many generation algorithms of pseudo random numbers (PRNs). Mersenne Twister (MT) is one of the best algorithms. It has three advantages that have not been achieved by other algorithms as follows [12]: 1) Astronomical period of $2^{19937} - 1$; 2) 623 dimensional equidistribution with up to 32 bit accuracy; 3) A working area of only 624 words. In addition, its performance exceeds that of integer-large-modulus generators. However, MT can only generate PRN with 32 bits length (MT-PRN), so we derive our key generator with arbitrary length from MT as follows:

$$G(Seed, Lok) = \begin{cases} MT(Seed) >> (32 - Lok), \text{ If } Lok \le 32 \\ MT(Seed) + G(S(Seed), Lok - 32), \text{ Else} \end{cases} \quad (1)$$

where, $MT(Seed)$ denotes the original MT, and $S(Seed)$ indicates a seed generator that can generate another new seed used to reset the starting point of MT and keep our key unpredictable. The main idea of our method is generating MT-PRNs according to the required size and jointing them as the needed PRN.

Furthermore, M is divided into n parts, (i.e. m_i, i = 1, 2,…, n). And parameter n can be calculated as follows:

$$n = \lceil LoM / LoK \rceil \quad (2)$$

Formula (2) means that the length of m_i should equal to LoK. However, if LoM is indivisible by LoK, the length of m_n equals to LoM - (n - 1) · LoK. Next, for each part m_i, we can execute XOR between itself and the key (for m_n, we should extract anterior LoM - (n - 1) · LoK bits of the key), and gain a resulting secure and skimble-skamble form m_i*. Consequently, we can obtain an integrated secure form M* of M by combining all m_i*. To sum up, the final secure form of M* can be obtained via following formula:

$$M^* = \sum_{i=1}^{n} m_i^* = \sum_{i=1}^{n} XOR(G(Seed, LoK), m_i) \quad (3)$$

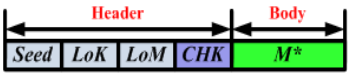

Figure 3. Structure of M_E

In the end, we employ MD5 to produce a short hash value of M* that acts as the checksum (CHK). That is a mean of detecting errors during transmission. If there are some errors, we should adopt a reasonable retransmission mechanism. About this, this paper does not refer to its detail. However, if the communication takes place in the secure LAN and other nice network or the message is very short, this operation could be elided. After the preprocessing, we can get M_E, as Figure 3 shows. In the header, the value of $Seed$, LoK, LoM and CHK should have a fixed size so as to ensure the correct parsing for them at the receiver side. Let us assume that their sizes are L_{Seed}, L_{LoK}, L_{LoM} and L_{CHK} respectively. Then, we can get the fixed length of the header $LoH = L_{Seed} + L_{LoK} + L_{LoM} + L_{CHK}$. Accordingly, we can define payload ratio (PR) as follows:

$$PR = LoM / (LoH + LoM) \quad (4)$$

In virtue of this structure, we accomplish the protection for the content of M (avoiding extraction attack). In addition, another issue we would emphasize is why we place the CHK in the header instead of the end of the structure. The reason is that we want to eliminate the possible "deceptive attack". Let us consider the case that the CHK is located in the end of M_E. After knowing the structure of M_E (especially LoM), the attackers may replace both M* and its CHK with other deceitful message and relevant CHK, which is referred to as deceptive attack. However, if the CHK is placed in the header, even though the attackers have known LoM, they could replace M* but not replace the CHK, because it has been transmitted immediately to the receiver behind LoM. Therefore, the receiver can distinguish the deceitful message by comparing the received CHK and the calculated CHK of received M*.

RM is the inverse of PM. Due to the sender and the receiver share the same knowledge of LoH and key generator, the receiver can reconstruct M as the sender constructs M*. Figure 2-B shows the whole process. However, before this step, the M* should be checked with the CHK. If there are some errors, the receiver could end RM and turn to other mechanisms (e.g. retransmission mechanism, etc.).

3. Case study with ITU-T G.729a

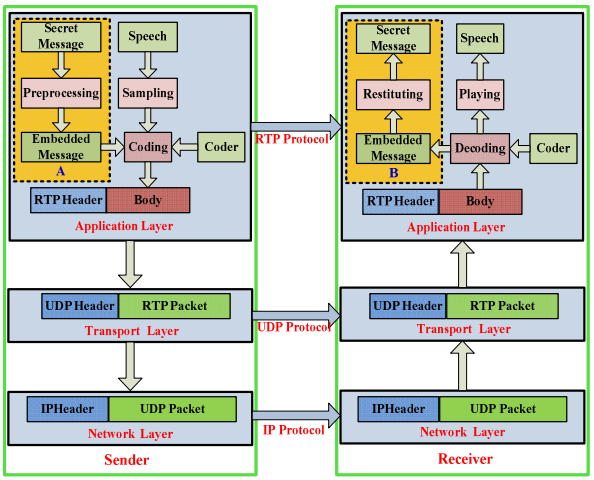

Figure 4. Architecture of StegTalk

We evaluate LSBCCM in StegTalk that is a covert communication system based on VoIP. Figure 4 shows its architecture. The VoIP communication in StegTalk is constructed based on JVoIPLIB [13] that is a free library providing a primitive VoIP model. In addition,

StegTalk supports typical coders, such as ITU-T G.711, G.723.1, G.729a, etc. In this case, we choose G.729a as the codec of the cover speech. G.729a is an algorithm for the coding of speech signals at 8kbit/s using conjugate-structure algebraic-code-excited linear prediction, and operates on speech frames of 10ms corresponding to 80 samples at a sampling rate of 8000 samples per second [11, 14].

In this case, LSBCCM is mainly applied in module A and B, as showed in Figure 4. In the following text, we introduce two key issues in detail, i.e. choosing of LSB and design of M_E.

3.1. Selection of LSB

Table 1. The definition of covert rates

L_0	L_1	L_1	L_1	L_1	L_1	L_1	L_1
L_2	L_2	L_2	L_2	L_2	L_3	L_3	L_3
L_3	L_3	P_1	P_1	P_1	P_1	P_1	P_1
P_1	P_1	P_0	C_1☆	C_1☆	C_1★☆	C_1☆	C_1☆
C_1★☆	C_1☆	C_1☆	C_1★☆	C_1☆	C_1☆	C_1☆	C_1★☆
S_1	S_1	S_1	S_1	GA_1	GA_1	GA_1	GB_1
GB_1	GB_1	GB_1	P_2	P_2	P_2	P_2	P_2
C_2☆	C_2☆	C_2★☆	C_2☆	C_2☆	C_2★☆	C_2☆	C_2☆
C_2★☆	C_2☆	C_2☆	C_2☆	C_2★☆	S_2	S_2	S_2
S_2	GA_2	GA_2	GA_2	GB_2	GB_2	GB_2	GB_2

L_0: Switched MA predictor of LSP quantizer
L_1: First stage vector of quantizer
L_2: Second stage lower vector of LSP quantizer
L_3: Second stage higher vector of LSP quantizer
P_0: Parity bit for pitch delay
P_i: Pitch delay the ith subframe (i = 1 or 2)
C_i: Fixed codebook the ith subframe (i = 1 or 2)
S_i: Signs of fixed-codebook pulses the ith subframe (i = 1 or 2)
GA_i: Gain codebook (stage 1) the ith subframe (i = 1 or 2)
GB_i: Gain codebook (stage 2) the ith subframe (i = 1 or 2)
★: 8 bits per frame; ☆: 26 bits per frame

Ya-min Su and Yongfeng Huang have analyzed the noisy resistance of each parameter in G.729a and evaluated their capability of carrying secret messages. It is found that the parameters of fixed codebook have the best transparency for information hiding [15]. Motivated by this observation, we examine the impacts of these parameters, and define two covert rates, i.e. 8 bits per frame (0.8 kb/s) and 26 bits per frame (2.6 kb/s). They are showed in Table 1. Between them, the former has better speech quality but lower capacity.

3.2. Design of M_E

We design the size of each parameter in the header of M_E as follows:

***Seed*:** 9 bits. It indicates that the range of *seed* value is from 1to 512.

***LoK*:** 3 bits, we define its unit is 64 bits in consideration of both enough security and reasonable number of bits. That is to say, the range of *LoK* value is from 64 ((0+1) * 64) to 512 ((7+1) * 64) bits.

***LoM*:** 20 bits, its unit is byte. Thus, the maximum length of the secret message is $2^{20} - 1$ Bytes. In the other words, the maximum transmission time is 10486 seconds with 8 bits per frame and 3227 seconds with 26 bits per frame.

***CHK*:** 128 bits, which is defined by MD5.

To sum up, the total number of bits in the header is 160. Consequently, the maximum payload ratio is 99.98% by formula (4).

4. Test and Analysis

In this section, we describe and analyze the experimental results. Our experiments focus on two key issues, i.e. speech quality with embedded messages and processing time of LLBCCM.

4.1. Test on Speech Quality

In order to test effects on speech quality induced potentially by steganography, we contrast the spectrograms of the cover speech without steganography and with steganography at 8 bits per frame (R_1) and 26 bits per frame (R_2) respectively. We choose phrase "Hello" (P_1) and phrase "It is nice to meet you" (P_2) as the cover speeches, and the text in the Abstract as secret message. In addition, we set *seed* as 200 and *LoK* as 128 bits. After encoded by G.729a, P_1 has 50 frames, and P_2 126 frames. Consequently, P_1 conceals 400 bits of message at R_1 and 1300 bits at R_2; P_2 conceals 1008 bits of message at R_1 and 3276 bits at R_2.

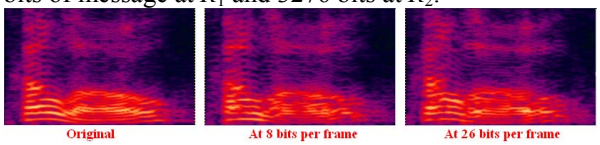

Figure 5. Spectrograms contrast of P_1

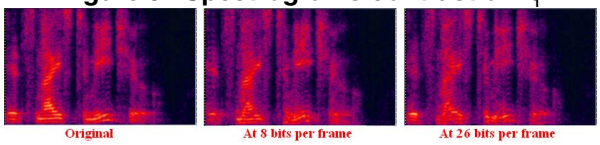

Figure 6. Spectrograms contrast of P_2

Figure 5 and 6 show the spectrograms of the original speeches and their steganographical versions with secret messages embedded at R_1 and R_2. From them, we could find that the spectrograms at R_1 have very slight differences from the original spectrograms, which indicates that the degradation of speech quality at R_1 is nearly negligible; the spectrograms at R_2 have a few visible differences from the original spectrograms,

which indicates that the quality degradation at R_2 may be perceived faintly, but not impact the understanding of speeches. These conclusions are also proved by the Mean Opinion Score (MOS) experimental result (See Table 2). MOS [16] is used popularly as a metric for speech quality and measured by subjective perceptual evaluation of speech quality. And each MOS is a mapping of perceived levels of distortion into either the descriptive terms "excellent, good, fair, poor, unsatisfactory".

Table 2. Result of MOS Experiments

Speech	Original	At R_1	At R_2
MOS	4.0	3.7	3.0

4.2. Test on Processing Time

Another crucial issue is that the performance of LSBCCM should meet the real-time requirement of VoIP. Processing time at the sender side (T_S) involves: (1) time of generating key (t_{SK}), which is relative with *LoK*; (2) time of generating M^* (t_{SM}), which depends on *LoM* and *LoK*; (3) time of generating *CHK* (t_{SC}), which is relative with *LoM*; and (4) time of embedding M_E (t_{SE}), which is relative with *LoM*. Accordingly, processing time at the receiver side (T_R) involves: (1) time of generating key (t_{RK}), which is relative with *LoK*; (2) time of restituting M (t_{RM}), which depends on *LoM* and *LoK*; (3) time of checking *CHK* (t_{RC}), which is relative with *LoK*, and involves the time of generating and comparing *CHK*; and (4) time of extracting M_E (t_{RE}), which is relative with *LoM*. Because embedding process is just an operation of bit substitution that can be merged into encoding process, its processing time t_{SE} can be often ignored. Likewise, t_{RE} can be ignored, because extracting operation can be often merged into decoding process with scarcely any time cost. Thus, we could assume that $T_S = t_{SK} + t_{SM} + t_{SC}$, $T_R = t_{RK} + t_{RM} + t_{RC}$. For the same *LoK* and *LoM*, if the experiments at the sender side and the receiver side are carried out in the same environments, we can deduce that $t_{SK} = t_{RK}$, $t_{SM} = t_{RM}$, and t_{RC} is slightly larger than t_{SC}. In this test, we aim at examining the possible values of T_S and T_R with different values of *LoK* and *LoM*, and evaluate their effects on the VoIP system. The experiments are carried out on Intel Celeron 2.66GHZ computers with 512M DDR2 SD RAM. And the experiments mainly focus on the typical values of *LoK* (i.e. 64bit, 128bit, 256 bit) and *LoM* (i.e. 1KB, 64KB, 128KB, 256KB, 512KB, 1MB). From Figure 7, we could observe that:

(1) Our deductions mentioned above are proved. t_{SK} and t_{RK} are exactly equal for the same value of *LoK*, which indicates that our key generation algorithm has low time and space cost; there is an error between t_{SM} and t_{RM}, which is mainly induced by disk read or write operations. However, the error span is from 0ms to 0.0681ms, so it is also considered that $t_{SM} = t_{RM}$. In addition, the deduction that t_{RC} slightly larger than t_{SC} is also correct. And their difference is negligible, if *LoM* is lager than 256KB.

(2) Total time is mainly depended on *LoM*. Though they are only relative to *LoK*, t_{SK} and t_{RK} are so little that they can be ignored in total time. Compared with *LoM*, *LoK* has little influence on t_{SM} and t_{RM}. Furthermore, t_{SC} and t_{RC}, which are the most parts in T_S and T_R respectively, lie upon *LoM* only. Therefore, we can consider total time is a function of *LoM* approximately.

(3) The maximum of T_S and T_R are 4.6193ms and 4.6296ms respectively. In other words, LSBCCM may increase the latency of the VoIP system with no more than maximum 4.7ms. That is very small, compared with allowable maximum of a 150ms one-way latency ITU-T G.114 recommends [17]. Therefore, LSBCCM well meets the real-time requirement of VoIP.

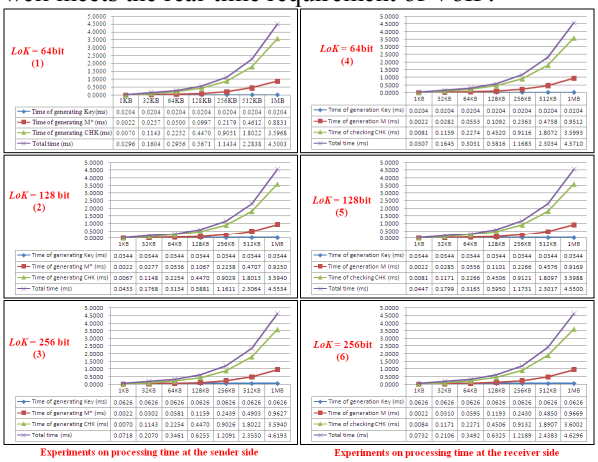

Figure 7. Results of experiments on processing time

5. Conclusion and Future Work

In this paper, we presented a LSBCCM to construct covert communications in VoIP. In this model, we designed a structure of embedded messages. It can provide flexible length of the secret message, avoid effectively both extraction attack and deceptive attack, and offer the maximum payload ratio 99.98%. In addition, in order to provide good security for secret messages without sacrificing real-time performance of VoIP, we proposed an encryption to protect secret messages. Although this encryption is simple and not effective for all security works, it is very effective for the short-term secure protection in LSBCCM and the reduction of the computing complexity. Furthermore, we evaluated the model with G.729a as the coder of cover speeches in StegTalk. In this case, the covert communication system can run with optional lengths

of the key and optional covert transmission speeds. The experimental results demonstrate that the proposed LSBCCM provides good transparency for transmitting secret messages, and well meets the real-time requirement of VoIP.

Our future work in this project is proceeding along three lines. Firstly, our implementation in this paper only supports maximum 1MB of secret message. That is not enough for exchanging large secret messages. Therefore, we are designing a subsection mechanism that may be a promising solution. Secondly, we are studying the retransmission mechanism that may be triggered by some errors. Finally, except LSB steganography model, we would like to locate other effective steganography models that can be employed in VoIP.

Acknowledgement

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2004CB318201, the National High Technology Research and Development Program of China (863 Program) under Grant No. 2006AA01Z444 and the Program for New Century Excellent Talents in University NCET-06-0650.

References

[1] J. Fridrich, T. Tevný, J. Kodovský. “Statistically undetectable JPEG steganography: dead ends challenges, and opportunities”, *Proceedings of the 9th ACM workshop on Multimedia & Security*, Dallas, Texas, USA, Sep. 2007, pp. 3-14.

[2] S. Badura, S. Rymaszewski. "Transform domain steganography in DVD video and audio content", *Proceedings of 2007IEEE International Workshop on Imaging Systems and Techniques*, 5, May, 2007, pp.1-5.

[3] Mohammad Pooyan, Ahmad, Delforouzi. "LSB-based audio steganography method based on lifting wavelet transform", *Proceedings of 2007 IEEE International Symposium on Signal Processing and Information Technology*, 15-18 Dec. 2007, pp. 600-603.

[4] Yuling Liu, Xingming Sun, etc. "An efficient linguistic steganography for Chinese text", *Proceedings of 2007 IEEE International Conference on Multimedia and Expo*, 2-5 July 2007, pp.2094-2097.

[5] M. Shirali-Shahreza. "A new method for real-time steganography", *Proceedings of the 8th International Conference on Signal Processing*, 2006, vol. 4, pp. 16-20.

[6] B. Goode. “Voice over Internet protocol (VoIP)”, *Proceedings of the IEEE*, vol. 90, Issue 9, Sept. 2002, pp. 1495 - 1517.

[7] C. Kratzer, J. Dittmann, T. Vogel etc. “Design and evaluation of steganography for voice-over-IP”, *Proceedings of 2006 IEEE International Symposium on Circuits and Systems,* 21-24 May 2006, pp.2397-2340.

[8] J. Dittmann, D. Hesse, etc. "Steganography and steganalysis in voice-over IP scenarios: operational aspects and first experiences with a new steganalysis tool set", *Proceedings of SPIE, Volume 5681, Security, Steganography, and Watermarking of Multimedia Contents VII*, Edward J. Delp III, Ping W. Wong, Editors, March 2005, pp. 607-618.

[9] Chungyi Wang, Quingcy Wu. "Information Hiding in Real-Time VoIP Streams", *Proceedings of the 9th IEEE International Symposium on Multimedia*, 10-12 Dec. 2007, pp. 255-262.

[10] W. Mazurczyk, Z. Kotulski, "Covert channel for improving VoIP security", *Advances in Information Processing and Protection*, Springer, Berlin, 2007, pp.271-280.

[11] R. Salami, C. Laflamme, etc. “Description of ITU-T Recommendation G.729 Annex A: reduced complexity 8 kbit/s CS-ACELP codec”, *Proceedings of 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 21-24, April 1997, pp. 775-778.

[12] Makoto Matsumoto, Takuji Nishimura. "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator", *ACM Transactions on Modeling and Computer Simulations: Special Issue on Uniform Random Number Generation*, vol. 8, Issue 1, Jan.1998, pp. 3-30.

[13] Jori Liesenborgs. Jori's Voice over IP Libraty (JVoIPLIB), http://reserach.edm.luc.ac.be/jori/jvoiplib/jvoiplib.html.

[14] ITU-T, Recommendation G.729. “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)”, Jan. 2007.

[15] Ya-min Su, Yongfeng Huang, etc. “Steganography-Oriented Noisy Resistance Model of G.729a”, *Proceedings of IMACS Multi-conference on Computational Engineering in Systems Applications*, vol. 1, 4-6 Oct. 2006, pp.11- 15.

[16] ITU-T Recommendation P.800. "Methods for subjective determination of transmission quality", Aug. 1996.

[17] ITU-T Recommendation G.114. “One-way transmission time, SERIES G: transmission systems and media, digital system and networks”, May 2003.