Computer Communications 34 (2011) 2236–2247

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Adaptive partial-matching steganography for voice over IP using triple *M* sequences

Hui Tian [a,c,*], Hong Jiang [b], Ke Zhou [c], Dan Feng [c]

[a] College of Computer Science and Technology, National Huaqiao University, Xiamen, Fujian 361021, China
[b] Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0150, USA
[c] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

ARTICLE INFO

Article history:
Received 23 November 2010
Received in revised form 30 April 2011
Accepted 15 July 2011
Available online 22 July 2011

Keywords:
Steganography
Information hiding
Voice over IP
M sequence
Partial matching

ABSTRACT

Although steganographic transparency and steganographic bandwidth are believed to be two conflicting objectives in the design of steganographic systems, it is possible and necessary to strike an optimal balance between them. This paper presents an adaptive partial-matching steganography for voice over IP (VoIP). We introduce the notion of partial similarity value (PSV) to evaluate the partial matching between covers and secret messages. By properly setting a low threshold of PSV and a high threshold of PSV, we can adaptively balance steganographic transparency and bandwidth. Moreover, we employ triple m sequences to eliminate the correlation among secret messages, guide the adaptive embedding process, and encrypt synchronization signaling patterns. In addition, we introduce an improved strategy that takes into account the similarity between not only covers and encrypted messages but also covers and original messages. We evaluate the proposed approach and its improved strategy with ITU-T G.729a as the codec of the cover speech in StegVoIP that is a prototypical covert communication system based on VoIP and compare them with some existing approaches. The experimental results demonstrate that the proposed approaches can provide a better balance between steganographic transparency and bandwidth. Furthermore, the results of delay tests show that they adequately meet the real-time requirement of VoIP.

1. Introduction

In recent years, steganography, which is an art and science of information hiding, has been widely applied successfully in covert exchange of information [1,2], copyright protection [3], etc. However, most of the existing studies on steganography are based on storage cover media [4]. In contrast, the area of steganography in streaming media is largely unexplored, in part due to the fact that the real-time characteristic of streaming media is a double-edged sword. While the real-time nature potentially offers better security for secret messages by virtue of its instantaneity, it does not allow many complex operations, which increases the difficulty in assuring security. Nevertheless, given its potential advantages, steganography for real-time streaming media may soon become a worthy subject of further studies. In this study, we will focus on a typical streaming media, Voice over IP (VoIP), as a possible carrier to apply steganography to enhance security for transmitting secret messages (covert communication) while maintaining good performance for real-time services of VoIP.

VoIP is a promising technique to enable telephone calls via a broadband Internet connection. Owing to its advantages of low cost and advanced flexible digital features, VoIP has become a popular alternative to the public-switched telephone network (PSTN), and extensive research on it has been conducted [5]. Our main motivations for VoIP-based steganography study are threefold. First, the ongoing conversation of VoIP can offer an ideal camouflage for secret messages, because the stream-like voice data is naturally assumed to be the only data carried in a given VoIP channel. Second, a typically short VoIP connection does not give eavesdroppers sufficient amount of time to detect possible abnormity due to hidden messages. Third, VoIP can often be considered a multidimensional carrier in which both the packet protocol headers and the payload data can be used to hide data.

Some researchers have noticed these advantages and proposed various useful steganographic approaches for VoIP [11–15,18–24] that are reviewed in the next section. Most of studies focused on the least-significant-bits (LSBs) steganography techniques for VoIP [11–15]. Fig. 1 illustrates the traditional LSBs steganography. Usually, LSBs steganography can render relatively high capacity and acceptable security, but direct LSBs substitu-

* Corresponding author at: College of Computer Science and Technology, National Huaqiao University, Xiamen, Fujian 361021, China. Tel.: +86 18959267996; fax: +86 592 6162556.
E-mail addresses: htian@hqu.edu.cn (H. Tian), jiang@cse.unl.edu (H. Jiang), k.zhou@hust.edu.cn (K. Zhou), dfeng@hust.edu.cn (D. Feng).

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.07.003

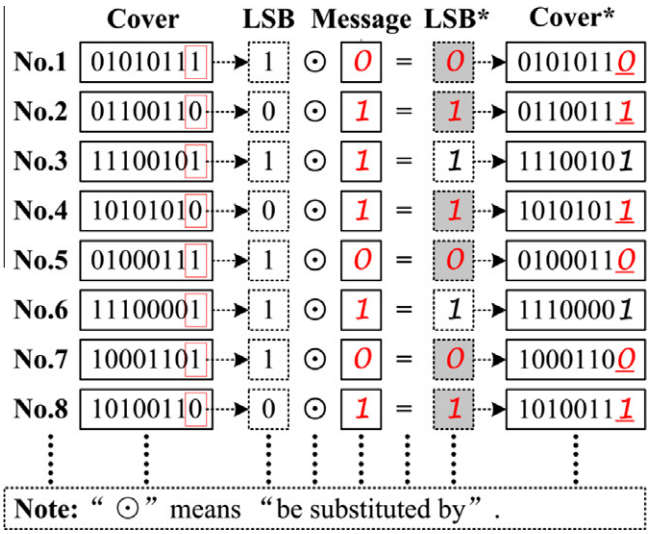

Fig. 1. The traditional LSBs steganography.

tion is vulnerable to detection by the steganalysis algorithm proposed by Dittmann et al. [13] based on the fact that the distribution of the LSBs in the stego-speech is not uniform. Therefore, Dittmann et al. [13] and Kratzer et al. [14] suggested that messages be encrypted prior to embedding to improve security. However, traditional encryptions, due to their time-consuming characteristic, are not well suited for the real-time steganography over VoIP. In fact, for the real-time covert communication we must strike an acceptable balance between providing adequate security and maintaining low latency for real-time services. In addition, as in other steganographies, steganographic transparency to non-authenticated entities and steganographic capacity (in covert communication also using steganographic bandwidth that is the steganographic capacity per unit time) for carrying secret messages are two key criteria, and it is popularly believed that they are inversely related. However, according to our previous work [24], we find that it is possible to balance steganographic transparency and steganographic capacity (bandwidth) objectives.

In this paper, we present a novel adaptive partial-matching steganography (APMS) scheme for VoIP, which aims at providing good security for real-time covert communication without sacrificing the real-time performance of VoIP and adaptively balancing the requirements for good steganographic transparency and sufficient bandwidth. To achieve these design goals, we introduce two key techniques as follows:

(i) Instead of traditional LSBs steganographic methods, we propose an adaptive steganographic method based on partial matching (partial similarity) that is defined by partial similarity value (PSV). Differing from our previous work [24], the embedding process is not only governed by two thresholds of PSV but also guided by an m sequence.
(ii) We employ two other m-sequences to eliminate the correlation among secret messages and synchronization signaling patterns to resist the statistical detection mentioned above and provide short-term protection for them.

It has been demonstrated by analysis and tests that APMS can adaptively balance steganographic transparency and steganographic bandwidth while maintaining good performance for real-time services of VoIP. Moreover, APMS can provide a better steganographic performance than the existing approaches.

The remainder of this paper is organized as follows. Section 2 reviews some previous VoIP-based steganographic techniques. Section 3 explains our motivation for this study and introduces the principle. The proposed APMS is described in Section 4, which is followed by the evaluation of APMS and its test results that are presented in Section 5. Finally, concluding remarks are given in Section 6.

2. Related work

Generally, the existing steganographic approaches for VoIP can be classified into three categories.

First, steganographic approaches for VoIP can be derived from the protocol steganography techniques, since VoIP is constructed on the Internet Protocol. Generally, the protocol steganography techniques have two main implementations [5–10]. One implementation utilizes the fact that few headers in packets are changed during transmission and embeds the secret messages into unused or optional fields of protocol headers, such as IP headers, TCP headers, UDP headers, RTP/RTCP headers, etc. This implementation can support relatively high steganographic bandwidth if all specific fields are used, but at the cost of potentially disabling the protocols for some specific functions. The other implementation encodes secret messages by varying packet rates, which is equivalent to modulating the packet timing (the inter-packet times). This implementation offers relatively high transparency, but sacrifices covert data rate and potentially degrades QoS of the network.

Second, steganography over VoIP can be implemented by embedding secret messages into the payload of VoIP packets. To the best of our knowledge, such approaches are mainly based on steganographic techniques that can strike a balance between security and complexity for specific speech codec. Due to its relatively high capacity and acceptable security, LSBs steganography, replacing the LSBs with binary bits of secret messages (as Fig. 1 shows), is popularly employed [11–13]. However, Dittmann et al. pointed out that direct substitution is vulnerable to detection by their steganalysis algorithm [13] and suggested that messages be encrypted prior to embedding to improve security [13,14]. Motivated by this view, they later proposed a scheme that introduces the cryptographies (i.e., Twofish, Tiger) for embedded messages [14]. This encryption operation is designed to be carried out offline before the embedding process, which is efficient for transmitting bulk covert messages. However, it is not suitable for real-time exchanging of secret short messages that are popular in interactive scenarios. To satisfy the requirement of real-time operation, Tian et al. [15] proposed an approach that uses a simple encryption to provide short-term but sufficient protection for secret messages. The encryption is based on an exclusive OR (XOR) operation between secret messages and a very large pseudo random number (PRN) generated by an improved Mersenne Twiste algorithm [16]. If the size of the secret message is no more than 1 MB, the total delay induced by the approach is still acceptable considering the allowable maximum of 150 ms one-way latency recommended by ITU-T G.114 [17]. However, the encryption operation is still totally carried out before the embedding process, so its scalability is not very well. Obviously, the extra delay increases with the size of the secret message. Thus, in this paper, the encryption operation is integrated into the embedding process instead of carried out beforehand, and thereby only induces a fixed and negligible delay for each frame.

Another challenge for LSBs steganography is to reduce the distortion of the covers (or the change of the covers) as much as possible. Huang et al. [18] introduced an LSB matching steganography to enhance transparency. The key point of this approach is to employ a pseudo random sequence (PRS) consisting of "0" and "1" to guide the embedding process. If the current value of the PRS is "1",

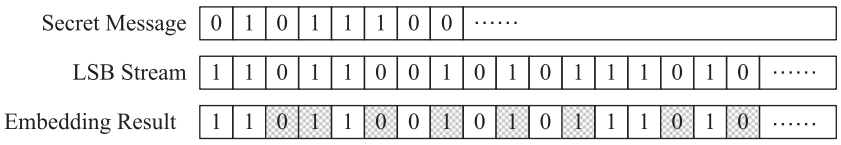

Fig. 2. An ideal steganography instance, in which we only employ the portions of LSBs that are equal to the binary bits of secret messages.

the corresponding LSB is replaced with one bit of secret messages; otherwise, the corresponding LSB is not modified. Due to the small amount of substitutions, this approach can reduce the distortion of the cover speech in comparison with traditional LSBs steganography. However, the steganographic bandwidth of this approach is consequently halved. In addition, Aoki [19] proposed a lossless steganographic approach for μ-law of G. 711 (PCMU), which embed secret messages into "0" speech samples by exploiting the characteristic that a "0" speech sample may be represented by two codes (namely "+0" and "−0"). If "0" ("1") is required to be embedded into such a "0" speech sample, the sign of the speech sample is modified to be "−" ("+"). Evidently, its steganographic capacity depends on the number of "0" speech samples. While this approach is a careful discovery, its applicability is limited.

Third, there are hybrid steganographic approaches that combine the above two approaches. For example, The authors in Refs. [20–22] proposed a steganography named LACK for VoIP using lost audio packets, which modifies both the content of VoIP packets and their time relations. In this approach, the payload of some intentionally delayed packets is used to transmit secret messages to the receiver, which is invisible to the unaware eavesdroppers. Its steganographic bandwidth is lower than that of those approaches that only modify packets, and higher than that of those approaches that only modify packets' time relations.

In our previous work, we also proposed two hybrid approaches. One [23] utilizes protocol steganography techniques to transmit the synchronization patterns and introduces the m-sequence to encrypted secret messages prior to the LSBs substitution to resist detection by the statistical steganalysis algorithm and provide a short-term security protection. The other [24] introduces the notion of partial similarity value (PSV) that measures the similarity between LSBs and embedded messages. By properly setting the threshold PSV, this approach can adaptively balance steganographic transparency and bandwidth. In the approach, protocol steganography techniques are employed to transmit some signaling bits securely. This paper substantially extends above works and presents a comprehensive adaptive partial-matching steganography (APMS) scheme for VoIP. In APMS, two thresholds of PSV and an m sequence are collectively employed to guide the embedding process to balance the steganographic transparency and bandwidth; moreover, other two m sequences are used to encrypt secret messages and signaling bits, which can provide good security for real-time covert communication while maintaining the real-time performance of VoIP. Table 1 shows a comparison of APMS and previous steganographic algorithms for VoIP.

3. Motivation and principle

A well established objective for steganography is to guarantee the perceptual similarity between the cover and the corresponding steg-object (the cover embedded with secret messages). The similarity function used to describe the perceptual similarity can be defined as follows [25]:

Similarity function: Let C be a non-empty set. A function sim: $C^2 \in (-\infty, 1]$ is called the similarity function on C, for $x, y \in C$, if $x = y$, $sim(x,y) = 1$; otherwise, $sim(x,y) < 1$.

Further, the transparency criterion for steganography can be formally described as one that, for $c \in C$ and $m \in M$, maximizes the value of the following function:

$$f(c,m) = sim(c, E(c,m)) - 1 \quad (1)$$

The LSBs substitution has a premise that the modification of LSBs is insufficient to induce perceptual distortion. However, the selection of LSBs often largely depends on subjective opinions. Therefore, the potential ineffectiveness of simple LSBs substitution on covers cannot be ignored, especially when employed in applications with high security requirements.

Clearly, if the LSB stream of the cover speech is exactly the same as the bit stream of secret messages to be embedded, the steganography has the best transparency without any induced distortion to the speech quality, i.e., $f(c,m)$ reaches its maximum value of 0. This observation suggests that an ideal cover may be obtained if its LSBs match perfectly with the binary bits of the secret messages. However, it is impossible to find such an ideal match for most given secret messages, making this approach impractical.

Another possible approach is to only employ the portions of LSBs that are equal to the binary bits of secret messages, as Fig. 2 shows. This approach can also induce no distortion to the speech quality. However, the selection of LSBs in this approach cannot be fixed beforehand and may even be uncertain, making it very difficult, if not impossible, for the receiver to determine which LSBs conceal the secret messages. In addition, it will inevitably sacrifice the steganographic bandwidth in favor of good transparency, which may not be a significant concern if the cover speech is sufficiently long. Unfortunately, the length of the cover speech depends on the conversation, which is often short and unpredictable. In fact, we must strike an acceptable balance between steganographic transparency and bandwidth. Therefore, we prefer to exploit the similarity among LSB stream and secret messages instead of their exact matching.

Table 1
Comparison of existing algorithms and APMS

Algorithms	Comparison items			
	Security	Transparency	Bandwidth	Self-adapting
Traditional LSBs steganography (Traditional-LSB)	Fair	Fair	Max.	No
LSB matching steganography (LSB-Matching) [18]	Good	Good	Middle	No
Adaptive LSBs steganography with one thresold (Adaptive-LSB)[24]	Good	Fair/Good/Excellent (Variable)	Min. ∼ Max. (Variable)	Yes (Good)
		A good balance between transparency and bandwidth		
APMS	Excellent	Fair/Good/Excellent (Variable)	Min. ∼ Max. (Variable)	Yes (Excellent)
		A better balance between transparency and bandwidth		

For measuring similarity, we define a similarity function as follows:

$$\varepsilon(X,Y) = \text{count}(x_i = y_i) \tag{2}$$

where, $X = \{x_1, x_2, \ldots, x_L\}, Y = \{y_1, y_2, \ldots, y_L\}$, L is the length of X and Y, $x_i, y_i = 0$ or 1, $i = 1,2,\ldots,L$. $\varepsilon(X,Y)$ represents the number of identical bits between X and Y. If X and Y are LSBs stream and secret messages respectively, the value of $\varepsilon(X,Y)$ is called the *similarity value* (SV). Accordingly, the transparency function (Eq. (1)) can be converted into:

$$f(X,Y) = \varepsilon(X,Y)/L - 1 \tag{3}$$

where the transparency value (TV) $f(X,Y) \in [-1, 0]$. If X and Y are one part of LSB set (B) and secret messages (M) respectively, the value of $\varepsilon(X,Y)$ is called the *partial similarity value* (PSV). Accordingly, the value of $f(X,Y)$ is called the partial transparency value (PTV). In the embedding process, the LSB stream and the secret message are divided into many parts with identical length, and PSV is employed to evaluate each LSB part and corresponding secret message part. Accordingly, through properly setting the substitution thresholds, we can adaptively achieve a good balance between steganographic transparency and bandwidth.

Moreover, we must provide sufficient protection for secret messages, which is necessary for resisting the statistical steganalysis and avoiding unauthorized extraction. However, this measure involves another tradeoff between security and quality of real time services [23], because the delay induced by encryption could have negative impact on real-time services of VoIP. Motivated by such a consideration, we introduce m sequence to encrypt secret messages.

To make our paper self-contained, we would like to breifly review the m sequence that is a typical pseudorandom binary sequence with the longest period. It has two properties that make it similar to zero-mean white noise: the numbers of ones and zeros are almost equal, and the autocorrelation function is as near to a delta function as we hope for [26]. Therefore, it has been widely employed in the code division multiple access (CDMA) technique and other spread spectrum communications.

M sequence is often generated by a linear feedback shift register (LFSR). Fig. 3 depicts an n-degree LFSR. In the figure, $a_i = 0$ or 1 ($i = 0,1,\ldots,n-1$), which indicates the state of the ith shift register; $c_i = 0$ or 1, which indicates the state of the ith feedback line. If $c_i = 1$, the ith feedback line is connected; otherwise, it is disconnected. M sequence consists of all the output bits. In addition, the vector $\{a_0, a_1, \ldots, a_{n-1}\}$ indicates the current state of LFSR. Generally, if the current state of LFSR is $\{a_{k-n}, a_{k-n+1}, \ldots, a_{k-1}\}$, the next state is determined by the following steps: (1) output a_{k-n}; (2) shift right all other bits by one position; and (3) calculate a new input of the nth register (a_k) by the following equation:

$$a_k = c_1 a_{k-1} \oplus c_2 a_{k-2} \oplus \cdots \oplus c_n a_{k-n} = \sum_{i=1}^{n} c_i a_{k-i} \bmod 2 \tag{4}$$

That is, the new state of LFSR depends on the following characteristic polynomial:

$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n = \sum_{i=0}^{n} c_i x^i \tag{5}$$

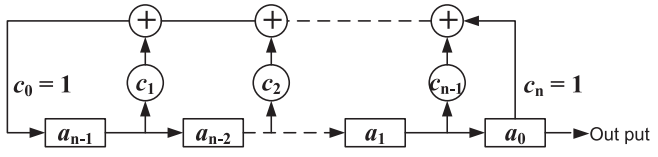

Fig. 3. The n-degree linear feedback shift register.

If we choose one of the primitive polynomials as the characteristic polynomial, we can obtain the m sequence with the longest period $P = 2^n - 1$, i.e., $\{a_0, a_1, \ldots, a_{p-1}, a_0, a_1, \ldots, a_{p-1}, a_0, a_1, \ldots\}$. In addition, we can generate different m sequences with different initial states for an LFSR with a given degree. The total quantity of m sequences generated by an n-degree LFSR can be calculated via the following equation:

$$q(n) = (2^n - 1) \cdot \phi(2^n - 1)/n \tag{6}$$

where $\emptyset(n)$ is the Euler's function that can be stated as follows:

$$\phi(n) = \begin{cases} 1, & n = 1 \\ \prod_{i=1}^{m} p_i^{e_i - 1} \cdot (p_i - 1), & n = \prod_{i=1}^{m} p_i^{e_i} \end{cases} \tag{7}$$

where each $p_i (i = 1,2,\ldots,m)$ is a unique prime number that is different from all the others. Furthermore, if the degree of LFSR is chosen from 1 to N, the quantity of m sequences generated by LFSR with different degrees can be calculated by the following equation:

$$Q(N) = \sum_{n=1}^{N} q(n) \tag{8}$$

In order to resist the statistical detection mentioned above, we attempt to make the embedded messages uniformly distributed and completely random using the m sequence. For that, we first introduce the following proposition and present its proof.

Theorem 1. *Given an arbitrary binary sequence X and an m sequence Y, Z is the result of executing the bit-wise XOR operation between them, then the entropy H(Z) of Z reaches the maximum of 1.*

Proof. Because the numbers of ones and zeros in an m sequence are almost equal, the probabilities $P_Y(1)$ and $P_Y(0)$ of a bit in Y being 1 and 0 are equal, i.e., $P_Y(1) = P_Y(0) = 0.5$. Moreover, let the probabilities $P_X(1)$ and $P_X(0)$ of a bit in X being 1 and 0 be α and β respectively, then $P_X(1) = \alpha$, $P_X(0) = \beta$, and $\alpha + \beta = 1$. For the arbitrary bits $x \in X$ and $y \in Y$, $P(x = 1) = \alpha$, $P(x = 0) = \beta$, $P(y = 1) = 0.5$, $P(y = 0) = 0.5$. Then, the process of executing the XOR operation (denoted by "$\oplus$") between x and y can be described as follows:

$$z = x \oplus y = \begin{cases} 1 \oplus 1 = 0 & P(x=1) \times P(y=1) = 0.5\alpha \\ 0 \oplus 0 = 0 & P(x=0) \times P(y=0) = 0.5\beta \\ 0 \oplus 1 = 1 & P(x=0) \times P(y=1) = 0.5\alpha \\ 1 \oplus 0 = 1 & P(x=1) \times P(y=0) = 0.5\beta \end{cases}$$

Therefore, $P(z = 1) = 0.5 \times (\alpha + \beta) = 0.5$, $P(z = 0) = 0.5 \times (\alpha + \beta) = 0.5$, namely, $P_Z(1) = P_Z(0) = 0.5$, then $H(Z) = (P_Z(1) \times \log_2(P_Z(1)) + P_Z(0) \times \log_2(P_Z(0))) = 1$. In other words, Z is a completely random and uniformly distributed binary sequence. □

In addition, according to Eq. (8), the total quantity of m sequences that can be generated depends on the maximum of LFSR degree N. In our practical system, let N be 60, then the total quantity of m sequences that can be generated is 1.4854×10^{34}, an astronomically large space in which the potential attacker can hardly decipher the encrypted messages in a short time.

For these two reasons, we consider that it is reasonable and feasible to eliminate the correlation among embedded messages and provide short-term protections for them using m sequences.

4. Proposed APMS scheme

4.1. Overview

An overview of APMS scheme is shown in Fig. 4. In APMS scheme, "divide and rule" strategy is adopted. That is, the secret message and the LSBs of the cover are divided into many parts with the same length. Moreover, we employ LFSR to obtain three m se-

quences, namely, S, S^*, and $S^\dagger$. In the embedding process, the secret message part is first encrypted using m sequence S; then the partial similarity between the message part and corresponding LSB part is evaluated, which is followed by embedding decision operation. If the substitution condition is satisfied, we can replace the LSB part with the message part, namely substitute each LSB with the message bit one by one; otherwise, the LSB part is maintained unchanged. In any case, the flag bits must be set, which is significant for the receiver to exactly extract the message part. In this operation, m sequence $S^\dagger$ is employed to encrypt the flag bits to resist the statistic attack.

Due to abandoning the unsuitable LSB parts, APMS effectively reduces the change of the cover, and thereby achieve a better steganographic transparency than traddtional LSB steganographic techniques. Further, by properly setting the substitution conditions, APMS can adptively achieve a good balance between steganographic transparency and bandwidth. The following sections describe APMS scheme in further detail and discuss some important issues, such as synchronization, parameter settings, etc.

4.2. General APMS

Let us assume that the sender wants to send L_M bits of secret messages $M = \{m_i = 0 \text{ or } 1 | i = 0, 1, \ldots, L_M - 1\}$; the n-degree LFSR is adopted to produce three m sequences $S = \{s_i = 0 \text{ or } 1 | i = 0, 1, \ldots, P-1, P = 2^n - 1\}$, $S^\dagger = \{s_i^\dagger = 0 \text{ or } 1 | i = 0, 1, \ldots, P-1, P = 2^n - 1\}$ and $S^* = \{s_i^* = 0 \text{ or } 1 | i = 0, 1, \ldots, P-1, P = 2^n - 1\}$, where P is the period of the m sequence; The LSBs set of each frame of the speech coded by a given codec is $B = \{b_i = 0 \text{ or } 1 | i = 0, 1, \ldots, L_B - 1\}$, where L_B is the total number of the LSBs in each frame. In APMS scheme, B is divided into R parts, namely, $B = \{\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{R-1}\}$, where $\mathcal{B}_i = \{\lfloor_{i0}, \lfloor_{i1}, \ldots, \lfloor_{i(n-1)}\}$, $i = 0, 1, \ldots, R-1$. $n = L_B/R$, $\lfloor_{ij} = b_{(i \times n + j)}, j = 0, 1, \ldots, n-1$, and M is divided into Q parts, i.e., $M = \{\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_{Q-1}\}$, where $\mathcal{M}_i = \{\Updownarrow_{i0}, \Updownarrow_{i1}, \ldots, \Updownarrow_{i(n-1)}\}, i = 0, 1, \ldots, Q-1; Q = L_M/n; \Updownarrow_{ij} = m_{(i \times n + j)}, j = 0, 1, \ldots, n-1$. For the given secret message part $\mathcal{M}_i$ and corresponding LSB part $\mathcal{B}_j$, the embedding process can be described as follows.

Step 1. Encryption: As mentioned above, to improve the security of secret messages, we first encrypt $\mathcal{M}_i$ using m sequence S, which can be represented as follows:

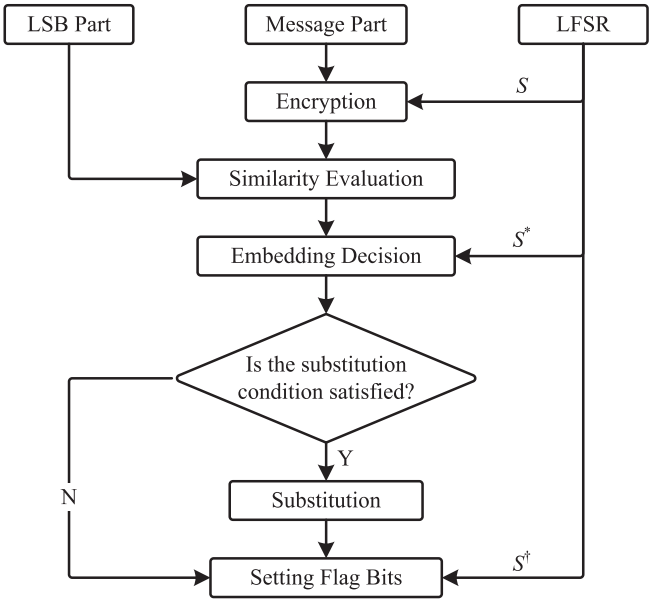

Fig. 4. Overview of APMS scheme.

$$\mathcal{M}_i^* = E(\mathcal{M}_i, S) = \sum_{j=0}^{n-1} \mathcal{M}_{ij}^* = \sum_{j=0}^{n-1} (\mathcal{M}_{ij} \oplus s_k) \tag{9}$$

where $\mathcal{M}_i^* = \{\Updownarrow_{i0}^*, \Updownarrow_{i1}^*, \ldots, \Updownarrow_{i(n-1)}^*\} (i = 0, 1, \ldots, Q-1)$ is the secure form of $\mathcal{M}_i$, and $k = (i \times n + j) \bmod P$.

Step 2. Similarity evaluation: Further, we calculate the PSV between $\mathcal{B}_j$ and $\mathcal{M}_i^*$, namely, $\varepsilon(\mathcal{B}_j, \mathcal{M}_i^*)$.

Step 3. Decision on embedding: In order to adaptively decide how and where to embed the messages, we define two threshold PSVs, namely η_1 and η_2, where $0 \leqslant \eta_1 \leqslant \eta_2 \leqslant n$. Moreover, incorporating another m sequence S^*, we design the replacing strategy based on partial matching as follows:

$$\phi(\mathcal{B}_j, \mathcal{M}_i^*) = \begin{cases} \mathcal{B}_j, & \text{If } \varepsilon(\mathcal{B}_j, \mathcal{M}_i^*) < \eta_1 \\ (1 - s_k^*) \cdot \mathcal{B}_j + s_k^* \cdot \mathcal{M}_i^*, & \text{If } \eta_1 \leqslant \varepsilon(\mathcal{B}_j, \mathcal{M}_i^*) < \eta_2 \\ \mathcal{M}_i^*, & \text{If } \varepsilon(\mathcal{B}_j, \mathcal{M}_i^*) \geqslant \eta_2 \end{cases} \tag{10}$$

where $s_k^* \in S^*$. Its means that it can be replaced under either of the two conditions, namely, $\varepsilon(\mathcal{B}_j, \mathcal{M}_i^*) \geqslant \eta_2$ or both $\eta_1 \leqslant \varepsilon(\mathcal{B}_j, \mathcal{M}_i^*) < \eta_2$ and $s_k^* = 1$. Obviously, if $\eta_1 = \eta_2 = n$, the embedding process has the best transparency but the smallest bandwidth; if $\eta_1 = \eta_2 = 0$, the embedding process can achieve the maximum bandwidth but the minimum transparency. However, we can adaptively balance the steganographic transparency and bandwidth by properly setting η_1 and η_2 for a given n. We will discuss the issule of parameters setting shortly in the following text. Fig. 5 illustrates this adaptive embedding process.

Step 4. Signaling mechanism: As showed in Fig. 5, to correctly extract the embedded parts at the receiver side, we set a flag bit for each LSB part to indicate whether it is used to hide secret messages. According to the aforementioned definition, we need R flag bits for each speech frame, denoted by $FB = \{fb_1, fb_2, \ldots, fb_R\}$. $fb_j \in FB (j = 1, 2, \ldots, R)$ can be determined as follows:

$$fb_j = \begin{cases} 1, & \text{If } \varepsilon(\varphi(\mathcal{B}_j, \mathcal{M}_i^*), \mathcal{M}_i^*) = n \\ 0, & \text{If } \varepsilon(\varphi(\mathcal{B}_j, \mathcal{M}_i^*), \mathcal{B}_j) = n \end{cases} \tag{11}$$

where $fb_j = 1$ indicates that the jth LSB part has been replaced, otherwise, the jth LSB part has not been changed. For the sake of privacy, we can encrypt the FB set using m sequence $S^\dagger$ in the same manner as Eq. (9). Moreover, as proposed in our previous work [24], we can distribute the encrypted flag bits among the unused and/or optional fields of the header of a certain packet in a predetermined manner. Because the number of flag bits is often small and altered continually, such a transmission of flag bits is potentially hard to discover.

Apparently, if B is divided into more parts, more flag bits will be needed. As far as the number of flag bits is concerned, a smaller R (a larger n) is preferable. However, the number of parts can also impact the performance of the embedding operation, because the embedding transparency and bandwidth depend on the values of parameters n, η_1 and η_2. For the convenience of discussion, we first give the definition of embedding rate (ER) and bit-change rate (BCR). ER (denoted by μ) can be calculated as follows:

$$\mu = \frac{N_M}{N_C} \tag{12}$$

where, N_M denotes the practical steganographic capacity, namely, the number of secret messages embedded into the cover; N_C denotes the number of cover bits, such as LSBs. ER is also called the usage rate

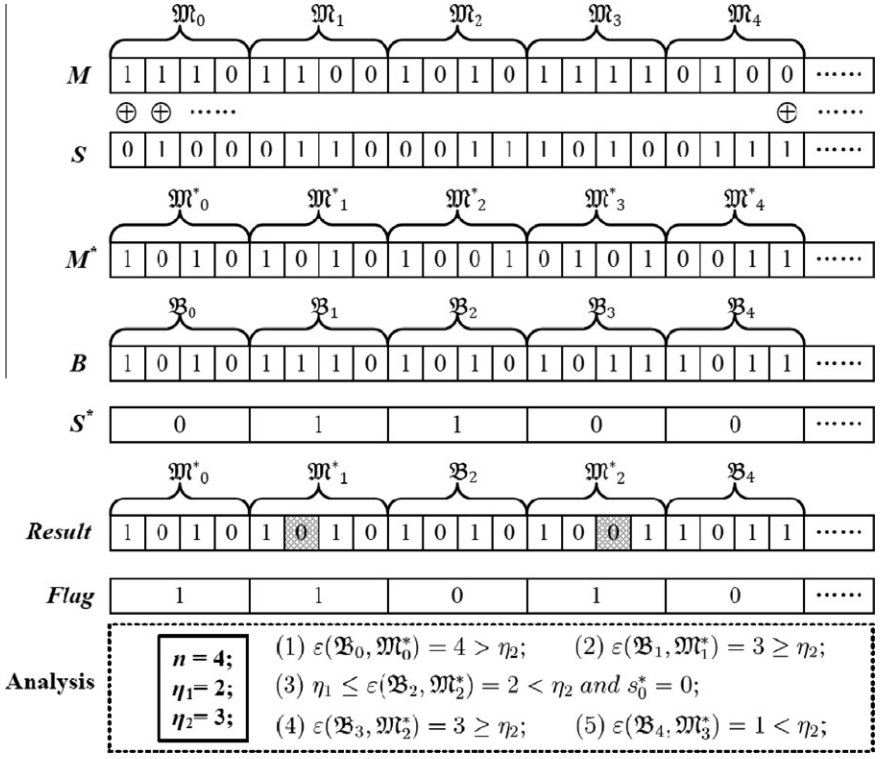

Fig. 5. The adaptive embedding process of APMS.

of the cover and often employed to measure the practical steganographic capability (bandwidth or capacity) of a given method.

BCR (denoted by ρ) is the ratio of the number of changed bits to the number of the cover bits, which can be defined as follows:

$$\rho = \frac{N_D}{N_C} \tag{13}$$

where N_D is the number of the changed bits and indicates the distortion of the cover. For APMS, BCR can be further described as follows:

$$\rho = \frac{\sum_{i=1}^{N_P}(n-\eta_i)}{N_C} \tag{14}$$

where, η_i is the PSV of the ith replaced parts; N_P is the number of the replaced cover parts, namely $N_P = N_M/n$, n is the part size. Essentially, BCR is a normalized distortion value, which can be used to evaluate the steganographic transparency of a given method.

In APMS, all the LSB parts with PSV values smaller than η_1 are not used to hide secret messages, so the substitution probability sp_1 is 0; all the LSB parts with PSV values larger than η_2 are employed to hide secret messages, so the substitution probability sp_3 is 1.0; and we determine whether the parts with PSV values between η_1 and η_2 are employed according to the values of the m sequence. Since m sequence obeys uniform distribution, the substitution probability sp_2 in this case is 0.5. If we exactly know the appearance probabilities of all LSB parts for each PSV value, namely, $ap_0, ap_1, ap_2, \ldots, ap_n$, then we can obtain the following equations:

$$\begin{aligned}\mu &= sp_1 \cdot \sum_{i=0}^{\eta_1-1} ap_i + sp_2 \cdot \sum_{i=\eta_1}^{\eta_2-1} ap_i + sp_3 \cdot \sum_{i=\eta_2}^{n} ap_i \\ &= \frac{1}{2}\sum_{i=\eta_1}^{\eta_2-1} ap_i + \sum_{i=\eta_2}^{n} ap_i \end{aligned} \tag{15}$$

$$\begin{aligned}\rho &= \sum_{j=1}^{3}\left(sp_j \cdot \sum_{i=\eta_{j-1}}^{\eta_j-1}(ap_i \cdot (n-i))\right) \\ &= \frac{1}{2}\sum_{i=\eta_1}^{\eta_2-1}(ap_i \cdot (n-i)) + \sum_{i=\eta_2}^{n}(ap_i \cdot (n-i)) \end{aligned} \tag{16}$$

where $\eta_0 = 0$ and $\eta_3 = n$. In this case, for given ER and BCR, we can obtain η_1 and η_2 exactly. Unfortunately, we cannot learn the appearance probabilities of all cover parts before starting the covert communication, so it is hard to give an exact guideline for setting η_1 and η_2. If all PSV values of cover parts are considered as evenly distributed between 0 to n, namely, $ap_0 = ap_1 = ap_2 = \cdots = ap_n = 1/(n+1)$, then we can obtain

$$\mu = \frac{\eta_2 - \eta_1}{2(n+1)} + \frac{n - \eta_2 + 1}{n+1} = 1 - \frac{\eta_1 + \eta_2}{2(n+1)} \tag{17}$$

$$\rho = \frac{1}{2}\sum_{i=\eta_1}^{\eta_2-1}\frac{n-i}{n+1} + \sum_{i=\eta_2}^{n}\frac{n-i}{n+1} \tag{18}$$

Accordingly, we can get referenced values of η_1 and η_2 by solving the above two formulas. Note that their practical configurations are also relevant to the adopted codec, the length of the conversation, the parameters setting of the covert communication system, etc. Therefore, the practical values of these parameters should be adjusted according to the feedback of previous tests.

To this end, we can complete the embedding process step by step. However, in order to successfully transmit the secret message from one end to the other, we need to solve two other crucial problems as follows.

The first problem is the negotiation of adopted m sequences. Although the sender and the receiver share the same knowledge about the generation algorithm of m sequences, they should also agree on the degree and the initial state (*IS*) of LFSR in order to get the same m sequences. In our work, for each degree, we design an *IS* pool that includes many *ISs*. We determine the degree and the

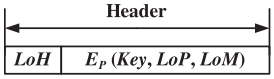

Fig. 6. The structure of the header.

IS of each m sequence by a pseudo random algorithm derived from the Mersenne Twiste algorithm [16]. As is well known, the result of the pseudo random algorithm often depends on the seed. In other words, if the seed is identical, the result is consequentially identical. In this work, the seed, denoted by *Key*, is sent to the receiver in real time, which will be detailed in the following paragraph. Consequently, we can obtain the same m sequences in both sides. It is worth noting that the receiver only needs two m sequences, i.e., S and $S^{\dagger}$, because the embedded parts are indicated by the flag bits.

The second problem is the synchronization of covert communication. For this problem, we introduced the synchronization mechanism based on protocol steganography techniques in our previous work [23]. We also define two synchronization patterns (*SPs*): *beginning of header* (*BoH*) and *beginning of message* (*BoM*), marking the start of the header and the start of the message respectively. *SPs* are distributed among the unused and/or optional fields of the header of the first embedded packet in a predetermined manner. Only upon detecting *SPs* can the receiver begin to extract the hidden data and carry out other relevant operations. It is also important to note that there are other two key operations: (1) The sender must ensure that *SPs* do not appear in the header of each VoIP packet, if he (she) does not want to send secret messages; (2) Whenever not receiving any hidden information, the receiver must continuously check *SPs* so as to detect new transmissions of secret messages in a timely manner. In addition, considering the actual need in this paper, the header is organized as shown in Fig. 6. In the figure, *LoH* (*length of header*) that indicates the total length of the header is often set as a fixed size in the interest of exact parsing; *Key* is the seed used to randomly choose the degree and the *IS* of m sequences; *LoP* (*length of part*) indicates the length of each part and *LoM* (*length of message*) indicates the length of secret messages. Moreover, for the sake of privacy, the parameters, *Key*, *LoP* and *LoM* are encrypted with a public key cryptography (denoted by E_P). E_P(*Key*,*LoP*,*LoM*) consequently denotes the cipher of the three parameters.

Accordingly, the receiver can restitute the secret messages by the following steps:

Step 1. Extraction of the header: Upon detecting *BoH* the receiver parses *Key*, *LoP* and *LoM* from the succeeding header, and produces the m sequences S and $S^{\dagger}$.

Step 2. Extraction of the secret message: After detecting *BoM*, the receiver starts to restitute the secret message. For each VoIP packet, the receiver extracts the encrypted flag bits in the header and decipher them using m sequence $S^{\dagger}$. According to the flag bits, the receiver extracts the embedded parts in the payload and deciphers them using m sequence S. Combining all parts, the receiver can obtain the whole secret message.

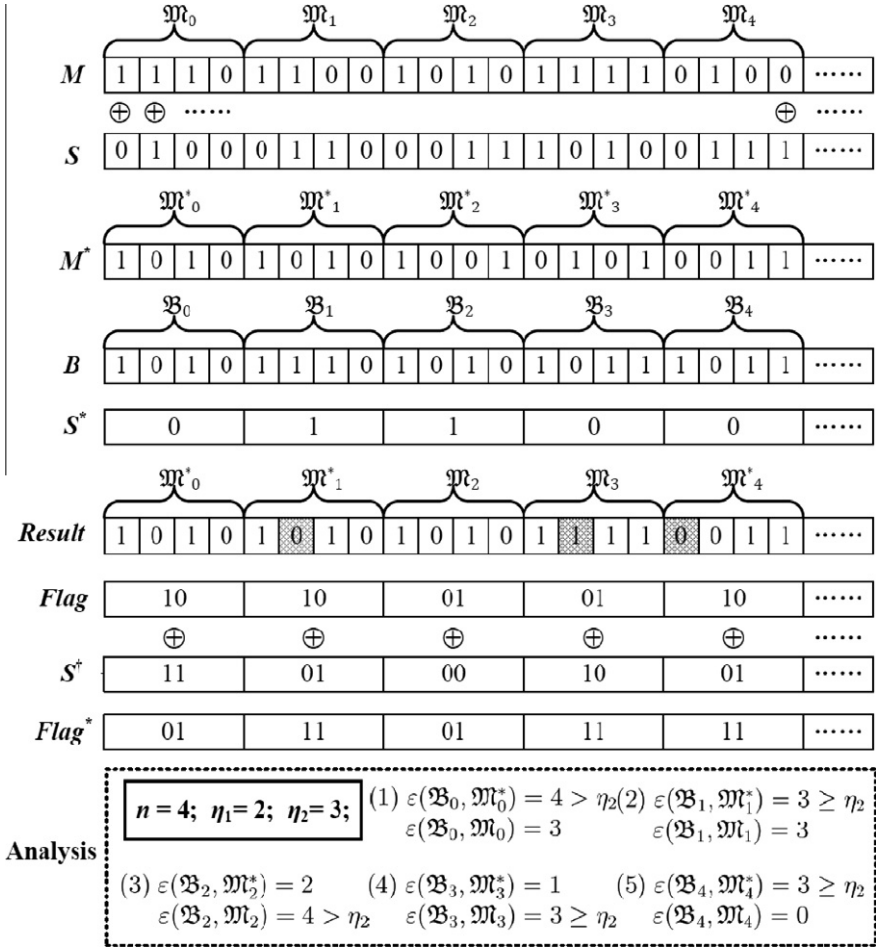

Fig. 7. The embedding process of Improved-APMS.

Table 2
The meaning of flags in improved-APMS.

Flags	Meaning
00	The corresponding part is not replaced
01	The corresponding part is replaced but not encrypted
10	The corresponding part is replaced and encrypted
11	Unused

Table 3
Test modes.

Mode	Algorithms	Related Parameters
1	Traditional LSBs steganography (Traditional-LSB)	—
2	LSB matching steganography [18] (LSB-Matching)	—
3	Adaptive LSBs steganography with one threshold PSV [24] (Adaptive-LSB)	$n = 4$, $\eta = 2$
4	Adaptive LSBs steganography with one threshold PSV [24] (Adaptive-LSB)	$n = 4$, $\eta = 3$
5	Adaptive LSBs steganography with one threshold PSV [24] (Adaptive-LSB)	$n = 4$, $\eta = 4$
6	APMS [Section 4.2]	$n = 4$, $\eta_1 = 2$, $\eta_2 = 3$
7	APMS [Section 4.2]	$n = 4$, $\eta_1 = 2$, $\eta_2 = 4$
8	APMS [Section 4.2]	$n = 4$, $\eta_1 = 3$, $\eta_2 = 4$
9	APMS [Section 4.2]	$n = 4$, $\eta_1 = 4$, $\eta_2 = 4$
10	Improved-APSM [Section 4.3]	$n = 4$, $\eta_1 = 2$, $\eta_2 = 3$
11	Improved-APSM [Section 4.3]	$n = 4$, $\eta_1 = 2$,$\eta_2 = 4$
12	Improved-APSM [Section 4.3]	$n = 4$, $\eta_1 = 3$, $\eta_2 = 4$
13	Improved-APSM [Section 4.3]	$n = 4$, $\eta_1 = 4$, $\eta_2 = 4$

4.3. An optional improved-APMS

This section presents an optional, improved strategy that considers the partial similarities between not only M^* and B but also M and B. In other words, we calculate both $\varepsilon(\mathcal{B}_j, \mathcal{M}_i^*)$ and $\varepsilon(\mathcal{B}_j, \mathcal{M}_i)$, and decide whether to replace or not by choosing the larger of these two to compare with the threshold PSVs. Therefore, the replacement strategy can be formalized as follows:

$$\phi(\mathcal{B}_j, \overline{\mathcal{M}}_i) = \begin{cases} \mathcal{B}_j, & \text{If } \varepsilon(\mathcal{B}_j, \overline{\mathcal{M}}_i) < \eta_1 \\ (1 - s_k^*) \cdot \mathcal{B}_j + s_k^* \cdot \overline{\mathcal{M}}_i, & \text{If } \eta_1 \leqslant \varepsilon(\mathcal{B}_j, \overline{\mathcal{M}}_i) < \eta_2 \\ \overline{\mathcal{M}}_i, & \text{If } \varepsilon(\mathcal{B}_j, \overline{\mathcal{M}}_i) \geqslant \eta_2 \end{cases} \tag{19}$$

where, $s_k^* \in S^*$, and if $\varepsilon(\mathcal{B}_j, \mathcal{M}_i^*) \geqslant \varepsilon(\mathcal{B}_j, \mathcal{M}_i), \overline{\mathcal{M}}_i = \mathcal{M}_i^*$; otherwise, $\overline{\mathcal{M}}_i = \mathcal{M}_i$. Fig. 7 illustrates this improved embedding process. In addition, the strategy needs two bits for the flag instead of one bit. Table 2 defines the meaning of each flag. Comparing with the above general embedding strategy, the improved strategy can support higner steganographic bandwidth while maintaining a reasonably good transparency. However, it increases the complexity of the whole algorithm. In fact, the operation time must be far less than the coding time of each frame for the specific speech codec. Therefore, which strategy will be chosen is depended on the adopted speech codec and the specific requirement for steganographic security and bandwidth.

5. Performance evaluation

This section evaluates APMS in StegVoIP [15,23] that is a prototypical covert communication system based on VoIP. StegVoIP supports typical coders, such as ITU-T G.711, G.723.1, G.729a, etc. In order to compare with the previous works, we choose G. 729a [27] as the codec of the cover speech, while APMS can also be applied with other coders used typically in VoIP. Similarly, we choose 8 LSBs (the bits with the least replaced impact on the speech quality) in each G. 729a frame based on the observation that the parameters of fixed codebook in G.729a have the best transparency for information hiding [23,24]. In the experiments, we mainly focus on two key issues, i.e., (1) performance comparison among the traditional LSBs steganography (Traditional-LSB), the LSB matching steganography (LSB-Matching) [18], the adaptive LSBs steganography with one threshold PSV [24] (Adaptive-LSB), APMS and Improved-APMS; and (2) the additional delay due to the proposed APMS algorithms.

5.1. Comparison of APMS and other algorithms

For the sake of performance comparison, we define thirteen test modes as shown in Table 3. For LSB-Matching, we randomly choose an m sequence to guide the embedding operation. For Adaptive-LSB, APMS and Improved-APMS, all m sequences that will be used are randomly generated. We first choose the English phrase "Huazhong University of Science and Technology" (denoted by P_E) as the covert speech and the introduction of Huazhong University of Science and Technology [28] as the secret message. After being encoded by G. 729a, P_E has 260 frames. Table 4 shows the steganographic bandwidth (bits per frame) in each mode. Fig. 8 shows the spectrograms of the original speech and its steganographic versions with the secret message embedded in thirteen different steganographic modes respectively. From them, we can find that: (1) speech spectrograms in mode 5, mode 9 and mode 13 are identical to the original spectrogram, because the embedding processes of these modes do not replace any bits. In other words, these three modes provide the best embedding transparency. (2) Other speech spectrograms respectively show slight differences from the original spectrogram, which indicate that degradations in the speech quality exist in varying degrees in these modes. Although the steganographic bandwidth impacts greatly on the degree of spectrogram difference, the modes with APMS and Improved-APMS can effectively decrease the difference by increasing the similarity of the embedding secret message and the cover speech.

To further evaluate and compare the performances of APMS and the other approaches, we collect 320 ten-second[1] speech samples. These samples consist of two categories[2]: English speech (including male speech and female speech) and Chinese speech (including male speech and female speech). All samples are PCM coded files with 8 kHz sampling rate, 16 bits quantization and mono. For each sample, we perform the corresponding steganographic experiment on its G. 729a coded file in the thirteen modes respectively. The secret message (also choosing the introduction of Huazhong University of Science and Technology [28], possibly only some forward parts) can be successfully embedded and retrieved in any case. Furthermore, for evaluating the speech quality, we employ the perceptual evaluation of speech quality (PESQ) method described in the ITU-T P. 862 Recommendation [29,30]. PESQ compares an original signal

[1] Although ITU-T P.862 recommends that the length range of each test speech sample is 8 to 30 s, PESQ is validated in ITU-T for use with signals that are mostly 8–12 s long [30]. Therefore, we typically choose the 10s long audio samples.

[2] So far, PESQ has not been validated with music as input to a codec [29], so the test samples do not include music.

Table 4
Steganographic bandwidth in each mode (bits per frame).

Mode	1	2	3	4	5	6	7	8	9	10	11	12	13
Bandwidth	8.00	4.03	5.68	2.55	0.60	3.88	2.91	1.51	0.62	5.97	3.98	2.58	1.05

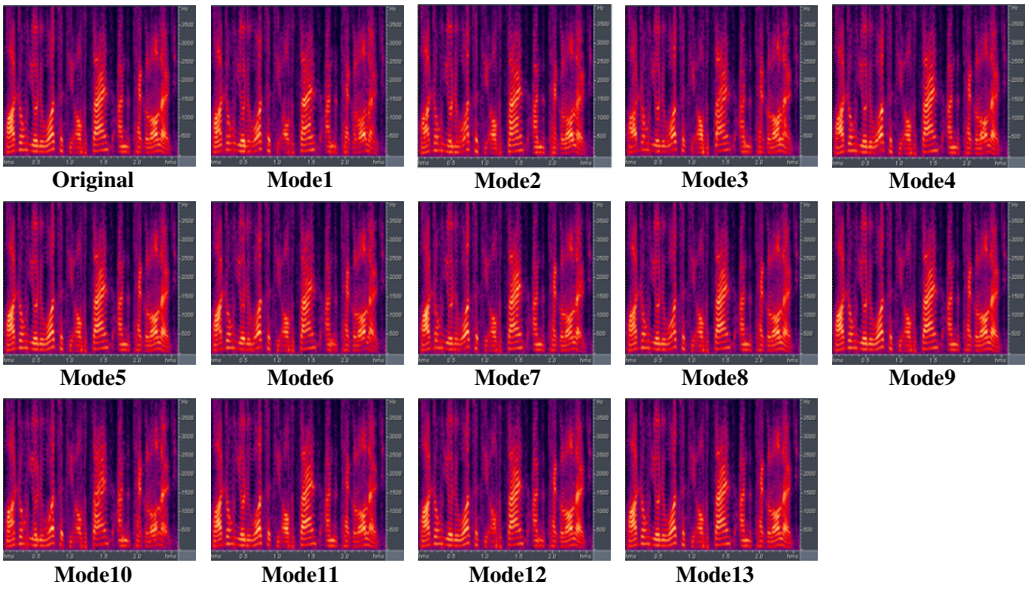

Fig. 8. Spectrogram contrast.

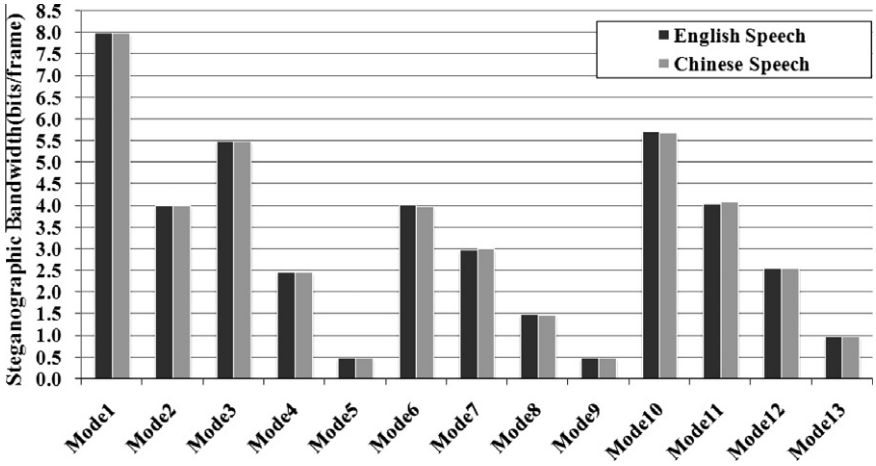

Fig. 9. Test results of steganographic bandwidth.

with a degraded signal and outputs a PESQ score as a prediction of the perceived quality. The range of the PESQ score is-0.5 (worst) to 4.5 (best). Moreover, the PESQ score can be converted to mean opinion score-listening quality objective (MOS-LQO). The range of MOS-LQO is 1.017 (worst) to 4.549 (best), which more closely matches the range of subjective mean opinion score (MOS) [31].

We convert all the steganographic G. 729a coded files into PCM encoded files as the degraded signals and perform the PESQ test with the orgianl samples as the reference signals. In the process, we also pay attention to the steganographic bandwidth and the effective bit-change rate (EBCR) for each sample. EBCR (denoted by ω) for a given sample is the ratio of the changed bits to the embedded bits, which can be defined as follows:

$$\omega = \frac{\rho}{\mu} = \frac{N_D}{N_M} \tag{20}$$

where, ρ is BCR; μ is ER; N_D is the number of the changed bits; N_M is the number of the embedded bits.

Figs. 9–11 show the statistical results of the steganographic bandwidth, the mean EBCR and the mean MOS-LQO value for all thirteen modes respectively. From these charts, we can observe four facts.

First, Chinese speech samples and English speech samples have the same steganographic bandwidth, which may mean that the language used in the speech has little impact on the steganographic bandwidth. However, for the same mode, the mean MOS-LQO of

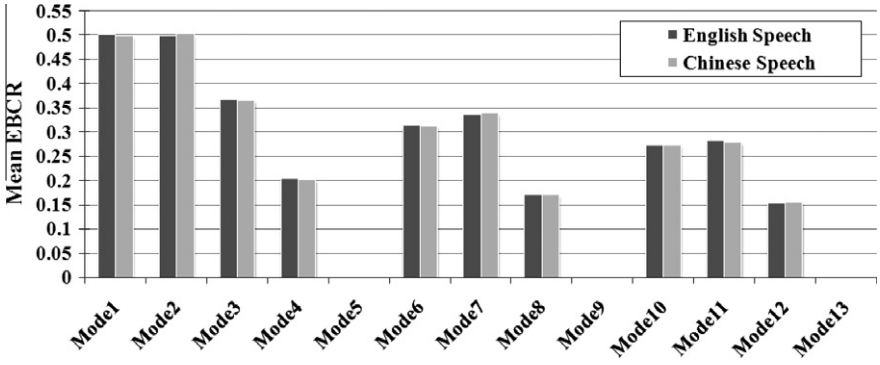

Fig. 10. Test results of effective bit-change rate (EBCR).

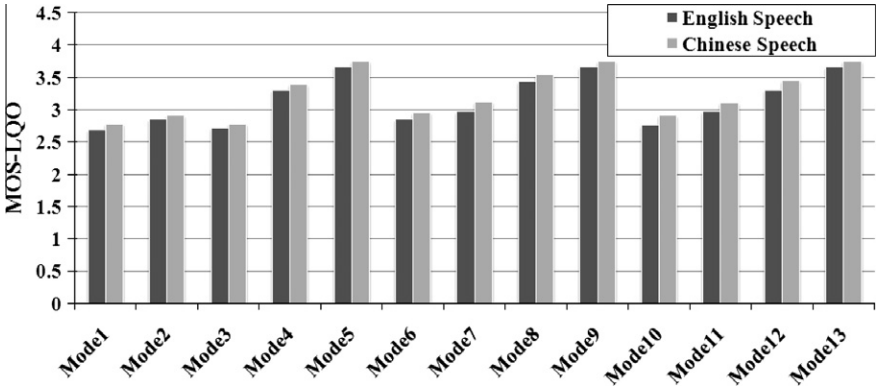

Fig. 11. Test results of MQS-LQO.

Chinese speech samples is slightly larger than that of English speech samples, which means that Chinese speech samples may have better steganographic transparency than English speech samples.

Second, Traditional-LSB can provide the largest steganographic bandwidth but the worst embedding transparency; LSB-Matching provides better embedding transparency than Traditional-LSB but only half steganographic bandwidth of Traditional-LSB; In contrast to the above two static algorithms, the other three algorithms can adaptively balance steganographic transparency and steganographic bandwidth by adjusting some parameters. Moreover, the latter three algorithms offer better performance than the former two algorithms. For instance, Adaptive-LSB in mode 3 offers better transparency and higher bandwidth than LSB-Matching; for the same steganographic bandwidth, APMS in mode 6 can provide better steganographic transparency than LSB-Matching; for the same steganographic transparency, Improved-APMS in mode 10 can support larger steganographic bandwidth than LSB-Matching (about 1.42 times), etc.

Third, APMS provides better steganographic transparency than Adaptive-LSB by decreasing the substitutions under the lower partial matching conditions. However, if the two threshold PSVs are set as the same value, APMS degrades to Adaptive-LSB (e.g. Mode 9 and Mode 5). In addition, Improved-APMS can provide a larger steganographic bandwidth than APMS while maintaining the same steganographic transparency or very slight degradation for the same parameters setting. Typically, for the same MOS-LQO the steganographic bandwidth of Mode 13 is nearly 2 times of that of Mode 9 and Mode 5. As mentioned above, these three modes provide the best steganographic transparency, because they do not change any bits, i.e., mean EBCR = 0.

Four, although the steganographic bandwidth of Improved-APMS is clearly larger than that of APMS for the same parameters setting, there are no significant differences between their MOS-LQOs. The reason is that Improved-APMS can more effectively decrease EBCR than APMS. From this, we learn that decreasing EBCR can enhance the steganographic transparency, which is our main motivation to balance steganographic transparency and steganographic bandwidth.

To sum up, while the pursuit of high steganographic transparency (bandwidth) is bound to decrease steganographic bandwidth (transparency), it is necessary and feasible to strike an optimal balance between them. In contrast to previous works, APSM and Improved-APSM provide a better balance between steganographic transparency and bandwidth by properly choosing embedding parts and consequently decreasing EBCR.

5.2. Additional delay

To evaluate the additional delay on the real-time services of VoIP due to APMS and Improved-APSM, we setup the experiment scenario as follows: two StegVoIP clients are run on Intel Pentium IV 2.8 GHZ computers with 1 GB DDR2 SDRAM that are linked with multi-hop network connection through CERNET (China education and research network). In APMS and Improved-APSM operations, the additional delays are mostly induced by the embedding algorithm at the sender side and the restituting algorithm at the receiver side. Therefore, we mainly focus on the average embedding time (*AET*) per frame and the average restituting time (*ART*) per frame. From Section 3, we learn that the degrees of m sequences are the key factors impacting the additional delays. Intuitively, the larger the degrees are, the larger the additional delays will

Table 5
Test results of *AET* and *ART* for APMS and improved-APMS.

Algorithm	Delay per frame	Parameters setting											
		n	η_1	η_2	n	η_1	η_2	n	η_1	η_2	n	η_1	η_2
		4	2	3	4	2	4	4	3	4	4	4	4
APMS	***AET***	6.852 8 us			6.766 2 us			6.710 4 us			6.704 8 us		
	ART	5.592 9 us			5.579 8 us			5.571 7 us			5.568 2 us		
Improved-APMS	***AET***	7.056 7 us			6.810 9 us			6.668 5 us			6.548 3 us		
	ART	5.741 0 us			5.657 1 us			5.587 3 us			5.565 0 us		

be. To consider the maximum additional delays, we set all the degrees as the maximum value 60. Since we do not know whether the setting of parameters n, η_1 and η_2 impacts the additional delays, we measure the delays respectively in the different parameters settings shown in Table 5.

The data in Table 5 demonstrates that the impact of parameters settings on *AET* and *ART* is negligible. Moreover, the maximum *AET*s of APMS and Improved-APMS are approximately 7us, the maximum *ART*s of APMS and Improved-APMS are approximately 6us. They are three orders of magnitude smaller than the allowable algorithmic delay of 15 ms for each frame [27], and four orders of magnitude smaller than allowable maximum of a 150 ms one-way latency ITU-T G.114 recommends [17]. Actually, one-way latencies including the embedding time and restituting time in our test are not more than 80 ms. Therefore, we can safely conclude that both APMS and Improved-APMS very adequately meet the real-time requirement of VoIP services.

6. Conclusions

In this paper, we presented an adaptive partial-matching steganography (APMS) scheme for VoIP. The notion of partial matching in APMS is quantified by a partial similarity value (PSV). By properly setting the threshold PSVs, we can adaptively balance steganographic transparency and bandwidth. Moreover, we employ three m sequences in APMS. The first one is used to eliminate the correlation among secret messages and provide short-term security protection. The second one is used to guide the adaptive embedding process under the condition that the current PSV is between the low threshold PSV and the high threshold PSV. The third one is employed to encrypt the synchronization signaling patterns that indicate the embedded parts. Furthermore, we introduce an improved strategy by taking into account the similarity between not only LSBs and encrypted messages but also LSBs and original messages. We evaluated APMS and Improved-APSM with ITU-T G.729a as the codec of the cover speech in StegVoIP and compared them with some existing approaches. The experimental results show that APSM and Improved-APSM can provide a better balance between steganographic transparency and bandwidth, which also proves that it is necessary and feasible to strike an optimal balance between them. In addition, the results of delay measures demonstrate that they adequately meet the real-time requirement of VoIP services.

In addition, it is particularly worth noting that the proposed approaches are codec-independent and cover-independent, meaning that it can be applied in the presence of any other coders used in VoIP and deployed with any other streaming media, such as internet protocol television (IPTV), etc. Also, we would like to point out that, since the signaling mechanism and synchronization mechanism based on protocol steganography techniques are introduced, the security of the VoIP-based steganography is also affected by the employed protocol steganography techniques. We are now studying how to evaluate the security of the multidimensional steganography, which is beyond the scope of the current manuscript. However, we believe that this effect can be minimized effectively, because the signaling and synchronization information are often very small in its quantity and its manner and embedded location can be altered continually. Especially, for short bursty transmissions, the statistical steganalysis is nearly impotent. Furthermore, APMS offers another dependable security, in addition to steganographic transparency, in that the eavesdroppers are not able to extract the embedded secret messages, which is the final line of defence.

Acknowledgements

This work was supported in part by National Basic Research Program of China (973 Program) under Grant No. 2011CB302300, US National Science Foundation under Grant No. CCF-0621526, National High Technology Research and Development Program of China (863 Program) under Grant No. 2009AA01A402, Program for Changjiang Scholars and Innovative Research Team in University under Grant No. IRT-0725, Natural Science Foundation of Fujian Province of China under Grant No. 2011J05151 and Scientific Research Foundation of National Huaqiao University under Grant No. 11BS210. The authors also wish to thank anonymous reviewers for their valuable comments and suggestions that improved this paper.

References

[1] N. Provos, P. Honeyman, Hide and seek: an introduction to steganography, IEEE Security & Privacy Magazine 1 (3) (2003) 32–44.
[2] K. Bailey, K. Curran, An evaluation of image based steganography methods, Multimedia Tools and Applications 30 (1) (2006) 55–88.
[3] E. T. Lin, A.M. Eskicioglu, Advances in digital video content protection, in: Proceedings of the IEEE: Special Issue on Advances in Video Coding and Delivery, 93(1) 2005, pp. 171–183.
[4] M. Shirali-Shahreza, A new method for real-time steganography, in: Proceedings of 8th International Conference on Signal Processing, vol. 4, 2006, pp. 16–20.
[5] B. Goode, Voice over internet protocol (VoIP), in: Proceedings of the IEEE, 90(9), September 2002, pp. 1495–1517.
[6] S.J. Murdoch, S. Lewis, Embedding covert channels into TCP/IP, in: Proceedings of the 7th Information Hiding workshop, June, 2005, pp. 247–262.
[7] D. Llamas, C. Allison, A. Miller, Covert channels in internet protocols: a survey, in: Proceedings of the 6th Annual Postgraduate Symposium about the Convergence of Telecommunications, Networking and Broadcasting, PGNET 2005, June 2005.
[8] S. Zander, G. Armitage, P. Branch, A Survey of covert channels and countermeasures in computer network protocols, IEEE Communications Surveys and Tutorials 9 (3) (2007) 44–57.
[9] S. Zander, G. Armitage, P. Branch, Covert channels and countermeasures in computer network protocols, IEEE Communications Magazine 45 (12) (2007) 136–142.
[10] X. Luo, E.W.W. Chan, R.K.C. Chang, TCP covert timing channels: design and detection, in: Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), June 2008, pp. 420–429.
[11] C. Wang, Q. Wu, Information hiding in real-time VoIP streams, in: Proceedings 9th IEEE International Symposium on Multimedia, December 2007, pp. 255–262.
[12] N. Aoki, A band extension technique for G.711 speech using steganography, IEICE Transactions on Communications E89-B (6) (2006) 1896–1898.
[13] J. Dittmann, D. Hesse, R. Hillert, Steganography and steganalysis in voice over IP scenarios: operational aspects and first experiences with a new steganalysis tool set, in: Proceedings of SPIE, Security, Steganography, and Watermarking of Multimedia Contents VII, vol. 5681, March 2005, pp. 607–618.

[14] C. Kratzer, J. Dittmann, T. Vogel, R. Hillert, Design and evaluation of steganography for voice-over-IP, in: Proceedings of 2006 IEEE International Symposium on Circuits and Systems, May 21–24, 2006, pp. 2397–2340.
[15] H. Tian, K. Zhou, Y. Huang, J. Liu, D. Feng. A covert communication model based on least significant bits steganography in voice over IP, in: Proceedings of the 9th International Conference for Young Computer Scientists, November 2008, pp. 647–652.
[16] M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Transactions on Modeling and Computer Simulations: Special Issue on Uniform Random Number Generation 8 (1) (1998) 3–30.
[17] ITU-T Recommendation G.114. One-way transmission time, SERIES G: transmission systems and media, digital system and networks. May 2003.
[18] Y. Huang, B. Xiao, H. Xiao, Implementation of Covert Communication Based on Steganography, in: Proceedings of 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, August 2008, pp. 1512–1515.
[19] N. Aoki, A technique of lossless steganography for G.711 telephony speech, in: Proceedings of 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, August 2008, pp. 608–611.
[20] W. Mazurczyk, K. Szczypiorski, Steganography of VoIP Streams, in: Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Part II, LNCS 5332, pp. 1001–1018.
[21] W. Mazurczyk, J. Lubacz, LACK-a VoIP steganographic method, Springer's Telecommunication Systems Journal 45 (2-3) (2010) 153–163.
[22] J. Lubacz, W. Mazurczyk, K. Szczypiorski, Vice over IP, IEEE Spectrum 47 (2) (2010) 42–47.
[23] H. Tian, K. Zhou, H. Jiang, Y. Huang, J. Liu, D. Feng. An M-Sequence Based Steganography Model for Voice over IP, in: Proceedings of the 44th IEEE International Conference on Communications, Dresden, Germany, June 14–18, 2009, pp. 1–5.
[24] H. Tian, K. Zhou, H. Jiang, Y. Huang, J. Liu, D. Feng, An Adaptive Steganography Scheme for Voice over IP, in: Proceedings of the 2009 IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, May 24–27, 2009, pp. 2922–2925.
[25] S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House Press, Boston, USA, 1999.
[26] S. Engelberg, H. Benjamin, Pseudorandom sequences and the measurement of the frequency response, IEEE Instrumentation & Measurement Magazine 8 (1) (2005) 54–59.
[27] ITU-T Recommendation G.729, Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP), January 2007.
[28] Available at:<http://english.hust.edu.cn/about_overview.html>.
[29] ITU-T Recommendation P. 862, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, Feb. 2001.
[30] ITU-T Recommendation P. 862.3, Application guide for objective quality measurement based on Recommendations P.862, P.862.1 and P.862.2, November 2007.
[31] ITU-T Recommendation P. 800, Methods for subjective determination of transmission quality, August 1996.