Steganography integration into a low-bit rate speech codec

Huang, Yongfeng, Liu, Chenghao, Tang, Shanyu ORCID logoORCID: https://orcid.org/0000-0002-2447-8135 and Bai, Sen (2012) Steganography integration into a low-bit rate speech codec. IEEE Transactions on Information Forensics and Security, 7 (6). pp. 1865-1875. ISSN 1556-6013

http://dx.doi.org/10.1109/TIFS.2012.2218599

Steganography Integration into a Low-bit Rate Speech Codec

Yongfeng Huang, Chenghao Liu, Shanyu Tang, *Senior Member IEEE*, and Sen Bai

***Abstract*—Low bit-rate speech codecs have been widely used in audio communications like VoIP and mobile communications, so that steganography in low bit-rate audio streams would have broad applications in practice. In this paper, the authors propose a new algorithm for steganography in low bit-rate VoIP audio streams by integrating information hiding into the process of speech encoding. The proposed algorithm performs data embedding while pitch period prediction is conducted during low bit-rate speech encoding, thus maintaining synchronization between information hiding and speech encoding. The steganography algorithm can achieve high quality of speech and prevent detection of steganalysis, but also has great compatibility with a standard low bit-rate speech codec without causing further delay by data embedding and extraction. Testing shows, with the proposed algorithm, the data embedding rate of the secret message can attain 4 bits / frame (133.3 bits / second).**

***Index Terms*—Information hiding; Low bit-rate speech codec; VoIP; G.723.1; Pitch period prediction**

I. INTRODUCTION

Nowadays people are becoming more and more concerned about the security of private information transmitted over the Internet. Protecting the private information from being attacked is regarded as one of the major problems in the field of information security. Apart from encryption, digital steganography has been one of the solutions to protecting data transmission over the network [1].

Steganography is the science of covert communications that conceal the existence of secret

S. Tang is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (Corresponding author; tel: +86 27-6784-8563; e-mail: shanyu.tang@gmail.com).

information embedded in cover media over an insecure network. A great effort has been made to explore the methods for embedding information in cover media, such as plaintext [2], audio files in WAV or MP3 [3], and images with BMP or JPEG format [4]. In recent years, computer network protocols and streaming media like Voice over Internet Protocol (VoIP) audio streams were used as cover media to embed secret messages [5][6]. Dittmann *et al.* [5], for example, suggested the design and evaluation of steganography in VoIP, indicating possible threats as a result of embedding secret messages in such a widely used communication protocol.

The methods of speech steganography can be classified into three categories. The first is the least significant bit (LSB) replacement / matching method towards the pulse code modulation (PCM) format voice data [3]. The second hides a secret message in transform domain, firstly transforming the cover's data to the transform domain, and then modifying some parameters in the domain to embed the secret message, with often used transform including the Cepstrum transform [7], discrete cosine transform [8], and so on. The third is the Quantization Index Modulation (QIM)-based method firstly proposed by Xiao et al. [9]. The QIM hides the secret message by modifying the quantization vector, which is applicable to various digital media, such as speech, image and video. It is very suitable to information hiding in the media compression encoding process.

Although some methods have been suggested for speech steganography, most of which dealt with high bit-rate speech format like PCM. However, most codecs used in VoIP are those with low bit-rate, such as Internet low bit-rate codec (iLBC), G.723.1 and G.729A; this means existing steganographic methods do not necessarily meet all the requirements of information hiding in VoIP. Up to now, only little attention has been paid to steganography in low bit-rate VoIP audio streams. For example, in our preliminary work, we proposed a codebook partition algorithm called the

Complementary Neighbor Vertex (CNV) algorithm for optimally dividing the vector codebook into two sub-codebooks, which are needed by QIM embedding.

In general, it is more challenging to embed information in low bit-rate VoIP streams. The first reason is the requisite for real-time VoIP communications. Most previous steganographic algorithms have been designed for embedding data in image or audio files. These algorithms usually take relatively long time to process data embedding. So they are not suitable for steganography in VoIP streams. Secondly, only a few results have so far proved conventional steganographic algorithms could survive low bit-rate compression. Finally, data embedding is to replace the redundancy in the cover media with the secret message; the less the redundancy is, the more difficult information hiding becomes. Unfortunately, all low bit-rate codecs are based on analysis by synthesis (AbS) that uses effective methods such as linear predictive coding (LPC) to eliminate redundancy. So conventional steganographic algorithms, *i.e.* replacing LSBs with the secret message, are not necessarily suitable for steganography in low bit-rate VoIP audio streams.

To take on these challenges, we propose a new method for steganography in low bit-rate VoIP audio streams and design an enhanced speech codec to integrate the information hiding function.

The rest of the paper is organized as follows. In Section II, related work is briefly introduced. Section III describes the pitch period prediction method in the hybrid speech codec. Section IV presents a new pitch period prediction-based algorithm for steganography in low bit-rate VoIP streams, and an enhanced speech codec combined with information hiding. Experimental results are discussed in Section V. Finally, Section VI concludes with a summary and directions for future work.

II. RELATED WORK

Over the past few years, a number of attempts have been made to study steganography in low bit-rate audio streams. Some related works are introduced below.

Several MP3stego, AAC-based audio steganographic systems have been suggested in recent years [10][11][12]. Wang *et al*. [1] proposed a scheme to convey secret messages by embedding them in VoIP streams. The scheme divides the steganography process into two steps, compressing the secret message and embedding its binary bits into the LSBs of the cover speech encoded by G.711 codec. Dittmann *et al*. [5] presented a more general scheme for steganography in VoIP, which can be used for transmitting an arbitrary secret message. More recently, Huang and co-workers [7] suggested an M-Sequence based LSB steganographic algorithm for embedding information in VoIP streams encoded by G.729A codec. With their algorithm, embedding data in a speech frame takes less than 20 us on average, which is negligible in comparison with the allowable coding time of 15 ms for each frame in VoIP. In addition, Huang *et al.* [6] suggested an algorithm for embedding data in some parameters of the inactive speech frames encoded by G.723.1 codec. However, this algorithm is also based on the LSB substitution of encoded audio streams. Therefore, the algorithms above would lead to obvious distortion, which affects the quality of steganographic speech.

Xiao suggested a QIM-based steganography in low bit-rate speech while encoding [9]. The QIM method randomly divides the whole codebook into two parts, each colored with white or black. When a secret bit of ‘0’ is embedded, the white codeword is used; the black codeword is used when a secret bit of ‘1’ is embedded. On the receiving side, the hidden bit is extracted by checking which part of the codebook the codeword belongs to. It is the first attempt to perform steganography and compression operation in the same codec. However, this information hiding algorithm has a small hiding capacity, which is no use in practice.

Our work described in this paper is the first ever effort to explore a novel method for steganography in low bit-rate speech based on pitch period prediction while the speech is encoded. The steganographic algorithm can not only achieve much higher data hiding capacity than the QIM algorithm [9], but also assure a good quality of speech.

III. PITCH PERIOD PREDICTION IN HYBRID SPEECH CODEC

As pitch period prediction is required in almost all speech analysis-synthesis (vocoder) systems, the pitch period predictor is an essential component in all speech codecs of low bit-rate. Because of the importance of pitch period prediction, a variety of algorithms for pitch period prediction have been proposed in the speech processing literature [13]-[15]. However, accurate predictions about the pitch period of a speech signal from the acoustic pressure waveform alone is often exceedingly difficult due to the reasons below.

1) The glottal excitation waveform is not a perfect train of periodic pulses. Although finding the period of a perfectly periodic waveform is straightforward, predicting the period of the speech waveform can be quite difficult, as the speech waveform varies both in period and in the detailed structure of the waveform within a period.

2) The interaction between the vocal tract and the glottal excitation also makes pitch period prediction difficult. In some instances, the formants of the vocal tract can significantly alter the structure of the glottal waveform, so that the actual pitch period is unlikely to predict. Such an interaction is most deleterious to pitch period prediction during fast movements of articulators while the formants are also changed rapidly.

3) The problem of accurately predicting the pitch period is the inherent difficulty in defining the exact beginning and end of each pitch period during voiced speech segments. Choosing the beginning and ending locations of the pitch period is often quite arbitrary. The pitch period discrepancies are arisen from the quasiperiodicity of the speech waveform, but also the fact that peak measurements are sensitive to the formant structure during the pitch period, whereas zero crossings of the waveform are sensitive to the formants, noise, and any DC level in the waveform.

4) Another difficulty of pitch period prediction is how to distinguish between unvoiced speech and low-level voiced speech. In many cases, transitions between unvoiced speech segments and low-level voiced speech segments are very subtle, and so they are extremely hard to pinpoint.

Apart from the difficulties in measuring the pitch period discussed above, pitch period prediction is also impeded by other factors. Although it is difficult to predict the pitch period, a number of sophisticated algorithms have been developed for pitch period prediction. Basically, algorithms for pitch period prediction can be classified into three categories. The first category mainly utilizes the time-domain properties of speech signals, the second category employs the frequency-domain properties of speech signals, and the third category uses both the time- and frequency-domain properties of speech signals. Most low bit-rate speech encoders, such as ITU G.723.1 and G.729A, adopt the first type of algorithms. As an example, the pitch period prediction algorithm of ITU G.723.1 is introduced below.

ITU-T G.723.1 encoder operates on frames of 240 samples each, a speech frame is denoted by $S[M] = \{s[n]\}_{n=0\ldots239}$, equal to 30ms at an 8-kHz sampling rate. Each frame is divided into four subframes of 60 samples each. After accomplishing a series of processes, the input signal of a frame $S[M]$ is converted to the weighted speech signal $F[M] = \{f[n]\}_{n=0\ldots239}$. For every two subframes (120 samples), the open-loop pitch period, L_{OL}, is computed using the weighted speech signal $f[n]$. The pitch estimation is performed on blocks of 120 samples. The pitch period is searched in the range from 18 to 142 samples. Two pitch estimations are computed for every frame, one for the first two subframes and the other for the last two. The open-loop pitch period estimation, L_{OL}, is computed using the perceptually weighted speech $f[n]$. A cross-correlation criterion, namely $C_{OL}(j)$, calculated by using the maximization method [13], is used to determine the pitch period, as shown in (1).

$$C_{OL}(j) = \frac{\left(\sum_{n=0}^{119} f[n] \cdot f[n-j]\right)^2}{\sum_{n=0}^{119} f[n-j] \cdot f[n-j]} \qquad 18 \le j \le 142 \tag{1}$$

The index j which maximizes the cross-correlation, $C_{OL}(j)$, is selected as the open-loop pitch estimation for the appropriate two subframes. While searching for the best index, preference is given to smaller pitch periods to avoid choosing pitch multiples. Maximums of $C_{OL}(j)$ are searched for beginning with j = 18. For every maximum $C_{OL}(j)$ found, its value is compared to the best previous maximum found, $C_{OL}(j')$. The following pseudo code shows how it works:

```
if ($j < j'+18$)
 then (if ($C_{OL}(j) > C_{OL}(j')$)
        then (select $C_{OL}(j)$, $L_{OL} \leftarrow j$)
      )
 else (if ($C_{OL}(j)$ - $C_{OL}(j') > 1.25$dB)
        then (select $C_{OL}(j)$ , $L_{OL} \leftarrow j$)
      )
```

Using the pitch period estimation, L_{OL}, a closed-loop pitch predictor is computed. The pitch predictor in G.723.1 is a fifth order pitch predictor. The pitch prediction contribution is treated as a conventional adaptive codebook contribution. For subframes 0 and 2, the closed-loop pitch lag is selected from around the appropriate open-loop pitch lag in the range of ±1. For subframes 1 and 3, the closed-loop pitch lag is coded differentially using 2 bits and may differ from the previous subframe lag only by –1, 0, +1 or +2 [10].

IV. PITCH PERIOD PREDICTION-BASED STEGANOGRAPHY ALGORITHM

A. Embedding Algorithm

In the process of G.723.1 encoding, the open-loop pitch estimation is conducted first, followed by closed-loop pitch prediction. The open-loop pitch estimation computes the open-loop pitch period L_{OL} of a frame of speech signal $F[m] = \{f[n]\}_{n=0...239}$. For each frame, two pitch periods are computed by using the first two subframes and the last two subframes, respectively. The method for computing the open-loop pitch period is described below.

First, a cross-correlation criterion C_{OL} is computed by using (1), and then it searches for the open-loop pitch following the procedures below [13]:

1) Suppose $L_{OL} = 8$, $j = 18$, Max$C_{OL} = 0$;

2) Using (1), compute $C_{OL}(j)$. If

$$\sum_{n=0}^{119} f[n]\cdot f[n-j] > 0, \sum_{n=0}^{119} f[n-j]\cdot f[n-j] > 0 \tag{2}$$

and

$$\text{Max}C_{OL} < C_{OL}(j) \text{ and } L_{OL} - j < 18 \text{ or Max}C_{OL} < \frac{3}{4}C_{OL}(j) \tag{3}$$

then $L_{OL} \leftarrow j$, and Max$C_{OL} \leftarrow C_{OL}(j)$.

3) Set $j = j + 1$, if $j \le 142$, return to 2), otherwise stop.

Having obtained the pitch period L_{OL} of a frame of speech signal $F[m] = \{f[n]\}_{n=0...239}$, search for the closed-loop pitch period and embed information.

The closed-loop pitch period of a subframe is defined by L_i, i = 0, 1, 2, 3, and its open-loop pitch period is L_{OLi}, i = 0, 1, representing the open-loop pitch periods of the first two subframes and the last two subframes, respectively. Adjusting L_{OLi} yields L_{OLAi}

$$L_{OLA_i} = \begin{cases} 19 & , L_{OL_i} = 18 \\ L_{OL_i} & , 18 < L_{OL_i} \leq 140 \\ 140 & , L_{OL_i} > 140 \end{cases} \tag{4}$$

The closed-loop pitch period L_i is assigned a value close to the open-loop pitch period L_{OLi}. The L_i values for odd subframes and for even subframes are obtained from different ranges as shown in (5).

$$\begin{aligned} L_0 &\in U_0 = \{L_{OLA_0} - 1, L_{OLA_0}, L_{OLA_0} + 1\} \\ L_1 &\in U_1 = \{L_0 - 1, L_0, L_0 + 1, L_0 + 2\} \\ L_2 &\in U_2 = \{L_{OLA_1} - 1, L_{OLA_1}, L_{OLA_1} + 1\} \\ L_3 &\in U_3 = \{L_2 - 1, L_2, L_2 + 1, L_2 + 2\} \end{aligned} \tag{5}$$

The minimum value of L_i is 17, and its maximum is 143. The number of L_i is equal to the number of elements in U_i, denoting by dim(U_i). $U_i(j)$ represents the jth element in U_i, $0 \leq j \leq \dim(U_i)$.

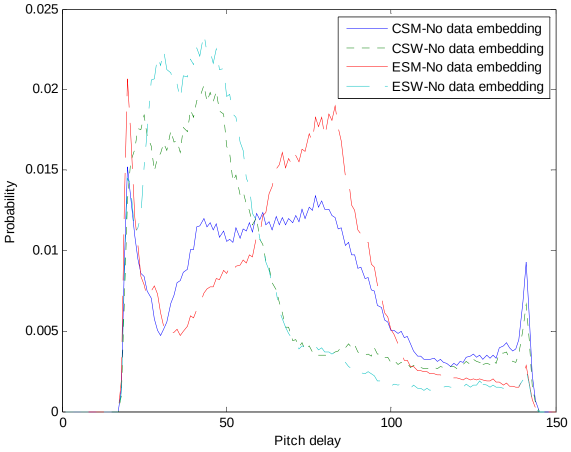

Fig. 1. Pitch distribution probabilities of four types of untouched G.723.1 VoIP speech samples

The pitch prediction contribution is treated as a conventional adaptive codebook contribution.

For subframes 0 and 2, the closed-loop pitch lag is selected around the appropriate open-loop pitch lag in the range ±1 and coded using 7 bits. For subframes 1 and 3, the closed-loop pitch lag is coded differentially using 2 bits and may differ from the previous subframe lag only by –1, 0, +1 or +2 [13]. The quantized and decoded pitch lag values are referred to as L_i from this point on. The pitch predictor gains are vector quantized using two codebooks with 85 or 170 entries for the high bit rate and 170 entries for the low bit rate. The 170 entry codebook is the same for both rates. For the high rate, if L_0 is less than 58 for subframes 0 and 1 or if L_2 is less than 58 for subframes 2 and 3, then the 85 entry codebook is used for the pitch gain quantization. Otherwise, the pitch gain is quantized using the 170 entry codebook. We studied the pitch distribution probabilities of closed-loop pitch period of untouched G.723.1 VoIP speeches, and Fig. 1 shows the pitch distribution probability results for four types of untouched G.723.1 VoIP speeches, each with 250 samples.

TABLE I
DATA EMBEDDING AT DIFFERENT EMBEDDING BIT-RATES

Steganography Solution (N_i)	Bit-rate	Embedding Subframes in $f[n]$
0	1 bit / frame	$F_0[m]$
1	1 bit / frame	$F_1[m]$
2	1 bit / frame	$F_2[m]$
3	1 bit / frame	$F_3[m]$
4	2 bits / frame	$F_0[m]$, $F_1[m]$
5	2 bits / frame	$F_0[m]$, $F_2[m]$
6	2 bits / frame	$F_0[m]$, $F_3[m]$
7	2 bits / frame	$F_1[m]$, $F_2[m]$
8	2 bits / frame	$F_1[m]$, $F_3[m]$
9	2 bits / frame	$F_2[m]$, $F_3[m]$
10	3 bits / frame	$F_0[m]$, $F_1[m]$, $F_2[m]$,
11	3 bits / frame	$F_0[m]$, $F_1[m]$, $F_3[m]$
12	3 bits / frame	$F_0[m]$, $F_2[m]$, $F_3[m]$
13	3 bits / frame	$F_1[m]$, $F_2[m]$, $F_3[m]$
14	4 bits / frame	$F_0[m]$, $F_1[m]$, $F_2[m]$, $F_3[m]$

In search for the closed-loop pitch period, data embedding is accomplished by adjusting the searching range U_i of the pitch prediction L_i of a subframe according to the secret bit information to be embedded. For instance, if the secret information to be embedded is ‘0’, the subframe search is performed on the even elements in U_i; if the secret information is ‘1’, the odd elements in U_i are searched. In G.723.1, each frame $F[m]$ has four subframes, $F[m] = \{F_0[m], F_1[m], F_2[m], F_3[m]\}$, all subframes require searching for the closed-loop pitch, so that data embedding can be performed on part of or all subframes. Therefore, we propose a series of solutions for steganography at four different embedding bit-rates, as shown in TABLE I, while the 15 strategies are randomly selected, the average data embedding rate is around 2.1 bits/frame, not 4 bits/frame.

On the basis of the steganography solutions listed in TABLE I, a new data embedding algorithm is proposed below.

Step 0: generate a random K, $k_i = \mathrm{mod}(K, 14)$, then choose a steganography solution N_i according to k_i and TABLE I.

Step 1: according to N_i, decide the embedding bit-rate and where to embed the secret bit stream $B = [b_0, b_1, b_2, ...]$, i.e. which i is the subframe in the m frame, $0 < i < 4$.

Step 2: suppose the bit b_i in the bit stream B is embedded in the $F_i[m]$ subframe of the frame m, data embedding is conducted by using the following algorithm.

Step 3: if $b_i = 0$, then data are embedded in the $F_i[m]$ subframe of the m frame, i.e. the pitch period ($l_i^{'}$) of the $F_i[m]$ subframe is searched upon $U_i^{'}$.

$$i = 0, l_0^{'} \in U_0^{'} \begin{cases} \{L_{OLA_0}\} & \text{if } \mathrm{mod}(L_{OLA_0}, 2) == 0 \\ \{L_{OLA_0} - 1, L_{OLA_0} + 1\} & \text{if } \mathrm{mod}(L_{OLA_0}, 2) == 1 \end{cases} \tag{6}$$

$$i = 1, l_1^{'} \in U_1^{'} = \begin{cases} \{L_0, L_0 + 2\}, & \text{if } \mathrm{mod}(L_0, 2) == 0 \\ \{L_0 - 1, L_0 + 1\}, & \text{if } \mathrm{mod}(L_0, 2) == 1 \end{cases}$$

$$i = 2, l_2^{'} \in U_2^{'} \begin{cases} \{L_{OLA_2}\} & \text{if } \mathrm{mod}(L_{OLA_2}, 2) == 0 \\ \{L_{OLA_2} - 1, L_{OLA_2} + 1\} & \text{if } \mathrm{mod}(L_{OLA_2}, 2) == 1 \end{cases} \tag{7}$$

$$i = 3, l_3^{'} \in U_3^{'} = \begin{cases} \{L_2, L_2 + 2\}, & \text{if } \mathrm{mod}(L_2, 2) == 0 \\ \{L_2 - 1, L_2 + 1\}, & \text{if } \mathrm{mod}(L_2, 2) == 1 \end{cases}$$

If $b_i = 1$, then data are embedded in the $F_i[m]$ subframe of the m frame, i.e. the pitch period of the $f_i[m]$ subframe is searched upon $U_i^{'}$.

$$
\begin{gathered}
i=0, l_0^{'} \in U_0^{'} \begin{cases} \{L_{OLA_0}\} & \text{if} \quad \mathrm{mod}(L_{OLA_0},2)==1 \\ \{L_{OLA_0}-1, L_{OLA_0}+1\} & \text{if} \quad \mathrm{mod}(L_{OLA_0},2)==0 \end{cases} \\
i=1, l_1^{'} \in U_1^{'} = \begin{cases} \{L_0, L_0+2\}, & \text{if} \quad \mathrm{mod}(L_0,2)==1 \\ \{L_0-1, L_0+1\}, & \text{if} \quad \mathrm{mod}(L_0,2)==0 \end{cases} \\
i=2, l_2^{'} \in U_2^{'} \begin{cases} \{L_{OLA_2}\} & \text{if} \quad \mathrm{mod}(L_{OLA_2},2)==1 \\ \{L_{OLA_2}-1, L_{OLA_2}+1\} & \text{if} \quad \mathrm{mod}(L_{OLA_2},2)==0 \end{cases} \\
i=3, l_3^{'} \in U_3^{'} = \begin{cases} \{L_2, L_2+2\}, & \text{if} \quad \mathrm{mod}(L_2,2)==1 \\ \{L_2-1, L_2+1\}, & \text{if} \quad \mathrm{mod}(L_2,2)==0 \end{cases}
\end{gathered} \tag{8}
$$

Step 4: repeat Step 3 until the completion of data embedding of the secret message $B = [b_0, b_1, b_2, ...]$.

For steganography using the data embedding algorithm above, errors in predicting speech pitch periods can be estimated in theory. As G.723.1 samples at 8 KHz, analysis of the closed-loop pitch period prediction shows data embedding would lead to one sampling-point error. So the absolute error ($g(x)$) in predicting pitch period caused by data embedding can be computed by

$$
g(x) = \begin{cases} (8000/x) - (8000/(x+1)) & x = 17,...,142 \\ (8000/x) - (8000/(x-1)) & x = 18,...,143 \end{cases} \tag{9}
$$

If the pitch period is $x = 17$, the maximum of $g(x)$ is 26.144Hz, and the relative error is 5.882%;

If the pitch period is $x = 142$, the maximum of $g(x)$ is 0.394Hz, and the relative error is 0.699%.

Therefore, the error in pitch frequency as a result of adjusting pitch prediction is proportional to the pitch frequency of speech signal, but the error has a little impact on speech synthesis, particularly for those speech signals with lower pitch frequency. In the literature [15], the average error of the most advanced algorithms for predicting pitch periods is found to be ± 0.5 samples, indicating that the pitch period prediction error arising from the data embedding algorithm is within the normal range.

B. Extracting Algorithm

The sender embeds the secret message in the low bit-rate speech streams encoded by G.723.1, and the bit streams containing the message are then sent to the receiver who extracts the secret message following the algorithm below.

Step 1: using a negotiating mechanism, the receiver acquires the data embedding algorithm (steganography solution) N_i for the current speech frame $F[m] = \{F_0[m], F_1[m], F_2[m], F_3[m]\}$.

Step 2: compute the pitch periods (L_i, $i = 0, 1, 2, 3$) of four subframes $F_0[m]$, $F_1[m]$, $F_2[m]$, $F_3[m]$ of the speech frame $f[m]$ decoded by G.723.1.

Step 3: according to the data embedding algorithm N_i, decide which of the four subframes $F_0[m]$, $F_1[m]$, $F_2[m]$, $F_3[m]$ contains the secret message, and determine the bits of the message using the following formula

$$\begin{aligned} &b_i=1, \text{ if mode}(L_i,2)=0 \\ &b_i=0, \text{ if mode}(L_i,2)=1 \end{aligned} \tag{10}$$

Step 4: repeat Step 3 until completion of decoding all speech frames, following by the bit streams of the secret message $B = \{b_0, b_1, \ldots b_i\}$ to be converted to the secret message $E = \{e_0, e_1, \ldots e_i\}$.

C. Design of the Coder with Steganography

A joint information embedding and lossy compression method is suggested in the literature [16], but no attempts have been made to study data embedding integrating into low bit-rate speech encoding. By using a data embedding algorithm based on pitch period prediction, we here develop the G.723.1 low bit-rate speech codec with data embedding functionality, i.e. the embedding and extracting of the secret message are integrated into G.723.1 speech codec.

To achieve data embedding while encoding in G.723.1, our specially designed secret information pre-processing module, steganography solution selecting module, $U_i^{'}$ updating module, and secret information bit stream framer module are inserted into a normal G.723.1 speech coder, as shown in Fig. 2. The pitch period prediction module in the codec is also modified so as to enable search for the closed-loop pitch upon the pitch period updating set, thus realising data embedding. Similarly, in order to achieve secret data extraction, the novel pitch period odd-even deciding module, steganography solution selecting module, secret data extraction module, and secret information post-processing module are built into the G.723.1 decoder, as shown in Fig. 3. Fig. 2 illustrates information embedding integrating into G.723.1 coder, whereas Fig. 3 shows information extraction along with G.723.1 decoding.

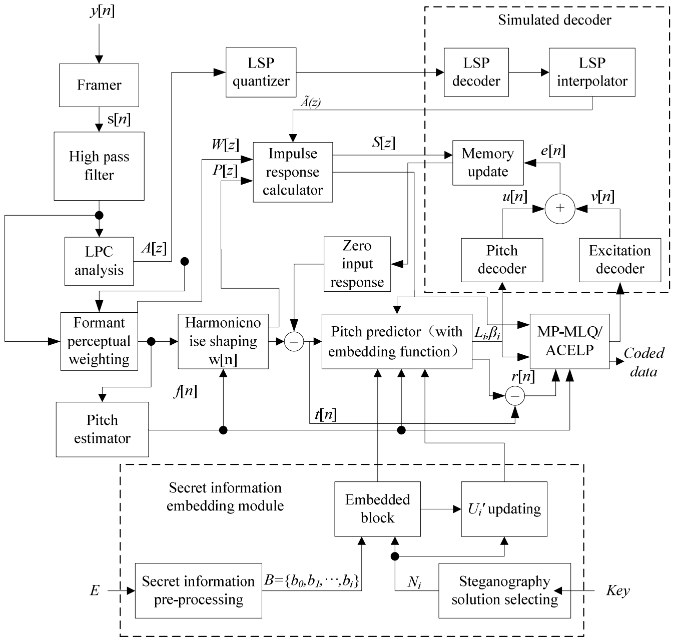

Fig. 2. G.723.1 coder with information embedding

In the process of information embedding and speech encoding, the secret message $E = \{e_0, e_1, \ldots e_i\}$ are compressed to form the secret data bit stream $B = \{b_0, b_1, \ldots b_i\}$, which is divided into segments according to the data embedding algorithm. The secret segments are then embedded into speech streams by adjusting pitch period prediction.

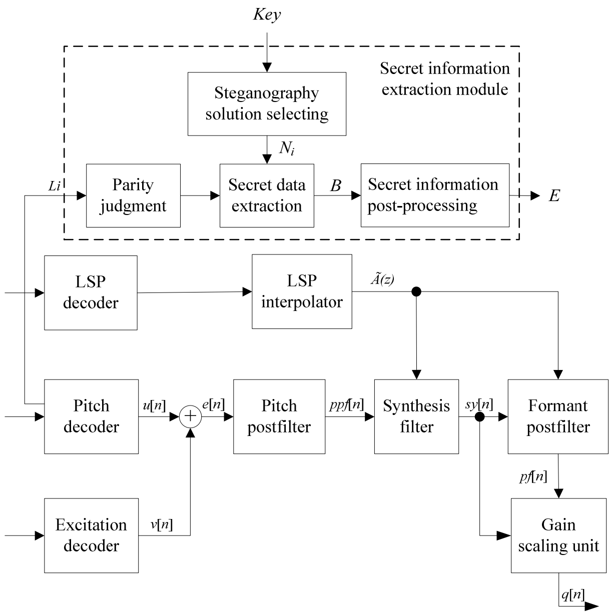

Fig. 3. G.723.1 decoder with information extraction

In the process of speech decoding and information extraction, G.723.1 decoder computes the pitch period of a subframe $F_{\mathrm{i}}[m]$, i = 0, 1, 2, 3, in the current frame $F[m]$, decides the odd-even nature of the pitch period L_{i} of the subframe by using the pitch period odd-even deciding module, determines the hidden data bit b_{i} according to the odd-even nature of L_{i} and the steganography solution N_{i}. The hidden data bit is then used to extract the secret information, $E'[n]$, by using the secret information post-processing module.

V. RESULTS AND DISCUSSION

A. Test Samples and Conditions

To evaluate the performance of the proposed steganographic algorithm, we employed different speech sample files with PCM format as cover media for steganography to conduct experiments. The speech samples are classified into four groups, Chinese Speech Man (CSM), Chinese Speech Woman (CSW), English Speech Man (ESM), and English Speech Woman (ESW). Each group contains 100 pieces of speech samples with length of 3 seconds, and 100 pieces of 10-second speech samples, and the four groups total 800 speech samples. Each speech sample was sampled at 8000 Hz and quantized to 16 bits, and saved in PCM format. Those speech samples with length of 3 seconds are defined as the 'Sample-3' sample set; the 'Sample-10' sample contains 10-second speech samples.

In our experiments, ITU G.723.1 codec operated at 6.3kbps, without silence compression. Fifteen solutions for data embedding proposed in TABLE I were used to conduct steganography at four different embedding bit-rates (1bit/frame, 2bits/frame, 3bits/frame, and 4bits/frame). Secret data were embedded into each audio frame by randomly choosing different embedding bit-rates and steganography solutions at equal probability.

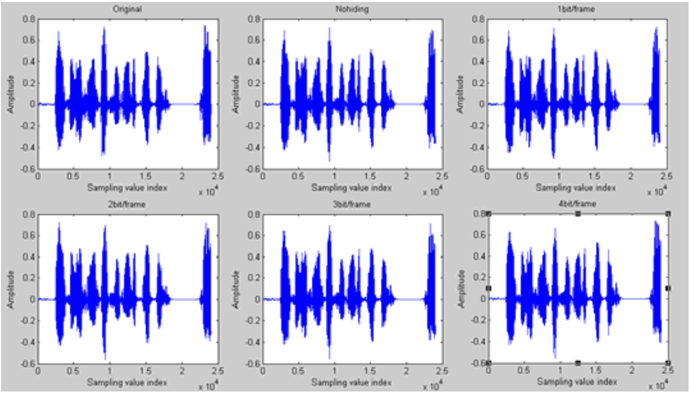

Fig. 4. Comparisons of time-domain amplitude plots of a 3-second CSM sample at different embedding bit-rates

B. Results and Analysis

Fig. 4 shows comparisons of the time-domain amplitude spectrum of an original 3-second CSM sample with those of the stego 3-second CSM samples at four different data embedding bit-rates. Almost no distortion occurred in the time domain as a result of data embedding in the speech sample;

no differences between the original speech sample and the stego speech samples in the time-domain spectrum were perceived, indicating that our proposed steganography algorithm had no or very little impact on the quality of the original speech.

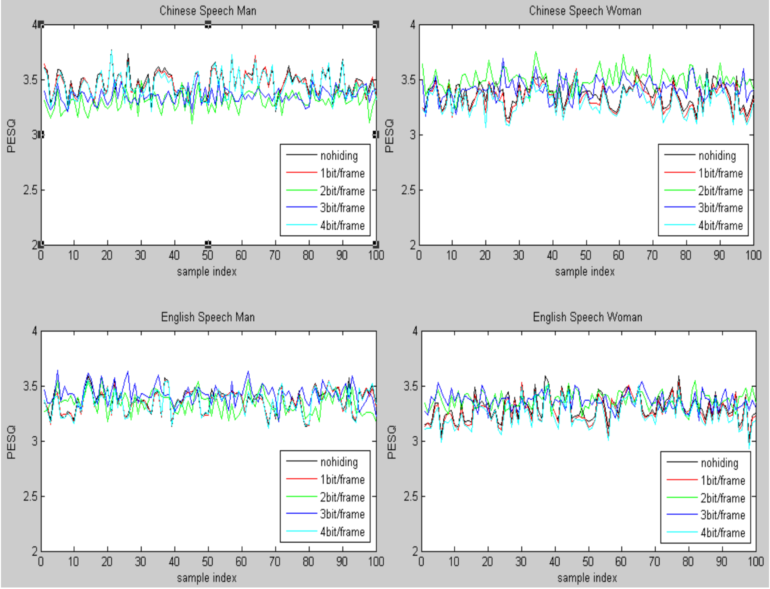

Fig. 5. PESQ values for 3-second samples using the proposed steganography algorithm

We used the perceptual evaluation speech quality (PESQ) value to assess the subjective quality of the stego speech samples. Fig. 5 and 6 shows the PESQ values for the original speech samples after G.723.1 codec without any data embedding and the stego speech files processed by G.723.1 with data embedding by means of the proposed steganography algorithm (detailed in Section IV), when the 3-second and the 10-second speech samples were used as cover media, respectively. The black curves are the PESQ values for the original speech samples without data hiding. Steganography was carried out at four different data embedding bit-rates (red curve: 1 bit/frame, green curve: 2 bits/frame, blue curve: 3 bits/frame, navy curve: 4 bits/frame,). As Figs 5 and 6 show, for the two types of speech cover media, the variations in PESQ between the original speech files and the stego speech files were so small, which means the proposed steganography algorithm has little effect on PESQ.

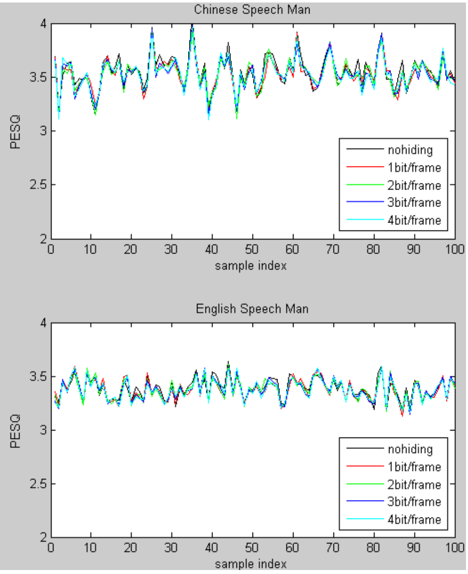

Fig. 6. PESQ values for 10-second samples using the proposed steganography algorithm

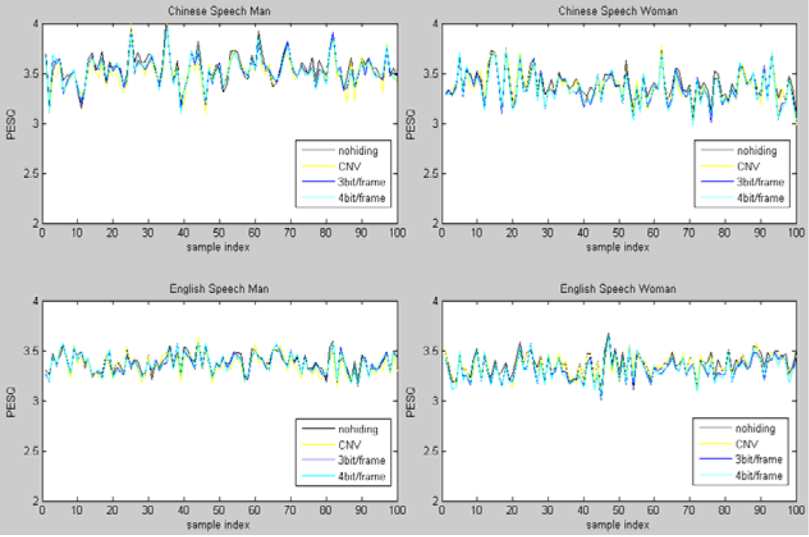

Fig. 7. Comparisons of PESQ values for 3-second samples between using the proposed steganography algorithm and using the CNV algorithm [9]

Figs. 7 and 8 show comparisons of PESQ values between using the proposed steganography algorithm and using the CNV algorithm (yellow curve) presented in the literature [9] for 3-second samples and 10-second samples, respectively. There were no obvious discrepancies in the PESQ value without (black curve: no hiding) and with data embedding at two different embedding bit-rates (blue curve: 3 bits/frame, navy curve: 4 bits/frame). As Figs. 7 and 8 show, the variations in PESQ between the original speech files and the stego speech files were so small, indicating that the proposed information hiding along with speech compression encoding had no or very little impact on the quality of the synthesized speech.

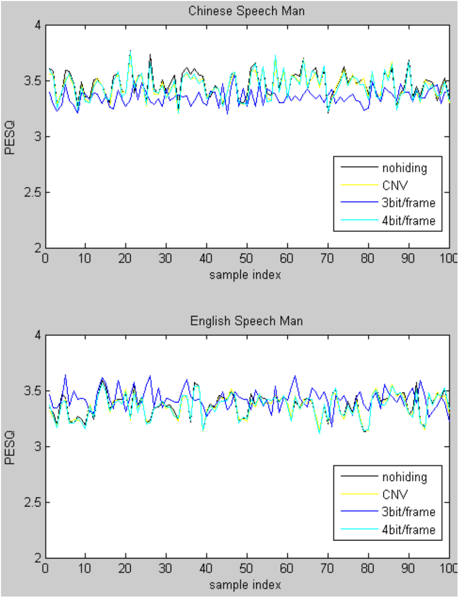

Fig. 8. Comparisons of PESQ values for 10-second samples between using the proposed steganography algorithm and using the CNV algorithm [9]

TABLES II to V list the PESQ values for the original speech samples and the stego speech files obtained by using the proposed steganography algorithm, when the 3-second and the 10-second speech samples were used as cover media, respectively. The statistical results were obtained for steganography experiments conducted at four different data embedding bit-rates. The PESQ values ranged from 2.9 to 4.1. On average, data hiding had less effect on the PESQ values of the male speech samples than the female speech samples. This is probably due to the fact that the pitch frequency of female speech has a greater range, and changes more quickly than male speech. Analysis of TABLES II to V shows, as the data embedding bit-rate increases, the average worsening change in PESQ increases - for 3s samples, 0.32% → 0.60% → 0. 96% → 1.22%; for 10s samples, 0.32% → 0.65% → 0. 94% → 1.22%. The maximum of the average worsening change in PESQ is 0.50%, and the average change in PESQ is within the standard error in PESQ for the speech samples without data hiding. This also means data hiding has a negligible effect on PESQ.

TABLE II
PESQ STATISTICS AT 1BIT/FRAME DATA EMBEDDING BIT-RATE

	Proposed Algorithm				Without Data Embedding				% Change in PESQ			
	3s Samples											
	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW
Average	3.53353	3.39355	3.37709	3.39173	3.53828	3.40712	3.38752	3.40776	-0.12%	-0.38%	-0.30%	-0.46%
Max	4.017	3.699	3.628	3.692	4.011	3.753	3.638	3.733	4.49%	3.46%	4.52%	3.07%
Min	3.179	3.108	3.055	3.033	3.19	3.103	3.03	3.075	-3.41%	-6.44%	-4.36%	-3.24%
	10s Samples											
	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW
Average	3.46297	3.34306	3.36003	3.28841	3.46875	3.35775	3.36626	3.30512	-0.16%	-0.44%	-0.18%	-0.50%
Max	3.74	3.619	3.591	3.584	3.784	3.604	3.603	3.591	1.95%	0.96%	1.63%	1.94%
Min	3.204	3.108	3.129	2.981	3.202	3.127	3.116	3.01	-2.23%	-2.00%	-1.96%	-2.90%
Note	‘Negative’ means a worse change in PESQ, ‘Positive’ means a better change in PESQ											

TABLE III
PESQ STATISTICS AT 2 BITS/FRAME DATA EMBEDDING BIT-RATE

	% Change in PESQ			
	3s Samples			
	CSM	CSW	ESM	ESW
Average	-0.28%	-1.01%	-0.16%	-0.94%
Max	5.78%	2.42%	3.47%	2.55%
Min	-3.71%	-7.42%	-2.61%	-3.78%
	10s Samples			
	CSM	CSW	ESM	ESW
Average	-0.42%	-0.93%	-0.22%	-1.04%
Max	2.20%	0.82%	1.37%	0.72%
Min	-2.29%	-2.82%	-1.81%	-2.86%

TABLE VI lists PESQ statistical results for the stego speech files obtained by using the steganography algorithm presented in [9], with cover media having the lengths of 3 and 10 seconds. Similarly, data embedding with the proposed algorithm led to a small change in PESQ, and the average change in PESQ is also within the standard error in PESQ for the speech samples without data hiding. However, the previous steganography algorithm [9] resulted in a larger change in PESQ than our proposed algorithm, and so it had a slightly high impact on PESQ.

TABLE IV
PESQ STATISTICS AT 3 BITS/FRAME DATA EMBEDDING BIT-RATE

	% Change in PESQ			
	3s Samples			
	CSM	CSW	ESM	ESW
Average	-0.59%	-1.63%	-0.28%	-1.35%
Max	4.14%	2.28%	3.28%	3.18%
Min	-4.17%	-8.12%	-2.96%	-6.23%
	10s Samples			
	CSM	CSW	ESM	ESW
Average	-0.52%	-1.42%	-0.35%	-1.47%
Max	1.51%	1.08%	2.23%	0.21%
Min	-2.32%	-4.02%	-2.40%	-4.12%

TABLE V
PESQ STATISTICS AT 4 BITS/FRAME DATA EMBEDDING BIT-RATE

	% Change in PESQ			
	3s Samples			
	CSM	CSW	ESM	ESW
Average	-0.84%	-1.88%	-0.38%	-1.76%
Max	4.85%	2.50%	3.24%	2.62%
Min	-5.71%	-5.99%	-4.05%	-5.17%
	10s Samples			
	CSM	CSW	ESM	ESW
Average	-0.71%	-1.83%	-0.48%	-1.86%
Max	2.04%	1.18%	1.50%	0.03%
Min	-2.93%	-4.54%	-2.07%	-4.52%

TABLE VI
PESQ STATISTICS USING THE STEGANOGRAPHY ALGORITHM PRESENTED IN [9]

	Algorithm Presented in [9]				Without Data Embedding				% Change in PESQ			
	3s Samples											
	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW
Average	3.50871	3.36577	3.35674	3.34671	3.53828	3.40712	3.38752	3.40776	-0.49%	-1.05%	-0.93%	-1.37%
Max	4.009	3.785	3.636	3.654	4.011	3.753	3.638	3.733	18.59%	15.50%	10.86%	15.19%
Min	3.098	2.979	3.137	2.998	3.19	3.103	3.03	3.075	-12.73%	-18.80%	-11.56%	-16.86%
	10s Samples											
	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW	CSM	CSW	ESM	ESW
Average	3.4508	3.31336	3.35771	3.26048	3.46875	3.35775	3.36626	3.30512	-0.62%	-1.44%	-0.29%	-1.22%
Max	3.713	3.569	3.553	3.53	3.784	3.604	3.603	3.591	1.43%	0.51%	1.30%	1.13%
Min	3.201	3.056	3.132	2.974	3.202	3.127	3.116	3.01	-2.44%	-4.92%	-1.82%	-4.40%

TABLE VII lists comparisons of changes in PESQ between the proposed steganography algorithm and the CNV algorithm presented in [9]. At the same embedding bit-rate with 3-second speech samples, the overall average standard error for the stego speech files using the proposed steganography algorithm was 1.60%, 4.04% less than the CNV algorithm, with both algorithms leading to 0.96% change in PESQ; for 10-second speech samples, the average worsening changes in PESQ of CSM and CSW with the proposed algorithm were smaller, those of ESM and ESW were bigger, the overall worsening change in PESQ was 0.05% larger, and the standard error (0.84%) was 0.02% larger in comparison with CNV. With the embedding bit-rate reaching 4 bits/frame, the average worsening change in PESQ of 3-second speech samples with the proposed algorithm was 0.26% larger, and the overall standard error (1.61%) was 4.03% smaller compared with CNV; for 10-second speech samples, the average worsening change in PESQ was 0.33% larger, and the overall standard error (0.90%) was 0.08% bigger than CNV.

TABLE VII
COMPARISONS OF CHANGES IN PESQ BETWEEN THE PROPOSED STEGANOGRAPHY ALGORITHM AND THE ONE PRESENTED IN [9]

Steganography Algorithm	Embedding Bit-rate (bits/frame)		3s Samples					10s Samples				
			CSM	CSW	ESM	ESW	Average	CSM	CSW	ESM	ESW	Average
Proposed Algorithm	3	Average	-0.59%	-1.63%	-0.28%	-1.35%	-0.96%	-0.52%	-1.42%	-0.35%	-1.47%	-0.94%
		St error	1.61%	1.80%	1.41%	1.57%	1.60%	0.87%	0.89%	0.77%	0.83%	0.84%
	4	Average	-0.84%	-1.88%	-0.38%	-1.76%	-1.22%	-0.71%	-1.83%	-0.48%	-1.86%	-1.22%
		St error	1.81%	1.76%	1.30%	1.57%	1.61%	0.93%	0.97%	0.77%	0.94%	0.90%
Algorithm Presented in [9]	3	Average	-0.49%	-1.05%	-0.93%	-1.37%	-0.96%	-0.62%	-1.44%	-0.29%	-1.22%	-0.89%
		St error	6.13%	6.53%	4.73%	5.17%	5.64%	0.76%	0.96%	0.68%	0.86%	0.82%

TABLE VIII lists differences in PESQ between normal en- and decoding and data hiding using different algorithms. When using the proposed steganography algorithm, the average worsening change in PESQ and the standard error of both 3s and 10s speech samples were within the range of the standard error of normal en- and decoding. For the algorithm presented in [9], this was the case for the 10s speech samples only. In comparison with the previous algorithm, the proposed algorithm had less impact on PESQ at lower data embedding bit-rates; when the data embedding bit-rate increased to 4 bits/frame, the average worsening change in PESQ was 0.295% larger, and the overall average standard error was 1.975% less than the previous algorithm.

TABLE VIII
DIFFERENCES IN PESQ BETWEEN NORMAL EN- AND DECODING AND DATA HIDING USING DIFFERENT ALGORITHMS

	Embedding Bit-rate (bits/frame)		3s Samples					10s Samples				
			CSM	CSW	ESM	ESW	Average	CSM	CSW	ESM	ESW	Average
Normal en- and Decoding	0	St error	0.1551	0.1518	0.1124	0.1214	0.1352	0.1234	0.1148	0.1109	0.1201	0.1173
Proposed Algorithm	3	Average	-0.0215	-0.0559	-0.0097	-0.0460	-0.0333	-0.0182	-0.0478	-0.0118	-0.0487	-0.0316
		St error	0.0570	0.0616	0.0476	0.0533	0.0549	0.0306	0.0301	0.0261	0.0279	0.0287
	4	Average	-0.0301	-0.0642	-0.0131	-0.0602	-0.0419	-0.0248	-0.0616	-0.0163	-0.0614	-0.0410
		St error	0.0641	0.0601	0.0436	0.0533	0.0553	0.0328	0.0329	0.0259	0.0312	0.0307
Algorithm Presented in [9]	3	Average	-0.0239	-0.0425	-0.0356	-0.0514	-0.0384	-0.0216	-0.0484	-0.0097	-0.0404	-0.0300
		St error	0.2163	0.2242	0.1608	0.1762	0.1944	0.0266	0.0324	0.0229	0.0287	0.0276

To evaluate the security of the proposed steganography algorithm, we employed the latest steganalysis method [17]-[20], which uses Derivative Mel-Frequency Cepstral Coefficients (DMFCC)-based Support Vector Machine (SVM) to detect audio steganography. SVM set RBF core function as its default parameter.

The test samples used were 501 CSM samples (300 as training samples, and 201 as test samples), 533 CSW samples (300 as training samples, and 233 as test samples), 819 ESM samples (600 as training samples, and 219 as test samples), 825 ESM samples (600 as training samples, and 225 as test samples), and Hybrid samples containing CSM, CSM, ESM and ESW samples. These five sorts of speech samples were used as the cover media in which data embedding at 4 bits / frame took place by using the proposed steganography algorithm and the one presented in [6]. The steganalysis results are listed in TABLES IX and X.

TABLE IX
STEGANALYSIS RESULTS OF THE LITERATURE [6] ALGORITHM USING DMFCC AT DIFFERENT DETECTION WINDOWS (DATA EMBEDDING RATE OF 3 BITS/FRAME)

Window Length（frames）	CSM（%）	CSW（%）	ESM（%）	ESW（%）	Hybrid（%）
1	53.2	55.6391	52.9268	51.2136	51.6442
10	63.6	66.9173	63.6585	64.3204	65.2466
20	71.2	81.5789	66.8293	70.6311	70.5531
40	77.2	85.3383	73.9024	74.7573	75.5605
80	78.4	92.1053	80.4878	84.9515	82.7354
150	81.6	95.8647	82.6829	91.2621	87.2945
200	86.0	95.8647	86.5854	93.4466	90.8072
250	88.4	97.3684	90.4878	94.6602	92.8699
300	91.6	97.7444	91.2195	94.4175	91.4798
333	93.2	98.4962	91.9592	95.3883	92.8996

In the experiments, we used LIBSVM Version 3.0 [21]. In the SVM-scale of LIBSVM, the lower is -1, the upper is 1, and the other parameters used are default values. In the SVM-train of LIBSVM, the svm_type is C-SVC, the kernel_type is RBF (radial basis function), the cost is 1000, the epsilon is 0.00001, and the other parameters used are default values.

TABLE X
STEGANALYSIS RESULTS OF THE PROPOSED ALGORITHM USING DMFCC AT DIFFERENT DETECTION WINDOWS (DATA EMBEDDING RATE OF 3 BITS/FRAME)

Window Length（frames）	CSM（%）	CSW（%）	ESM（%）	ESW（%）	Hybrid（%）
1	47.2	49.6241	49.0244	50	50.7474
10	48.8	49.2481	50.7317	48.7864	49.1031
20	47.2	56.391	48.5366	51.699	52.5411
40	51.2	53.7594	50.4878	51.9417	52.2422
80	51.2	55.2632	51.7073	55.0971	52.0179
150	50.4	51.5038	54.878	53.6408	51.7937
200	48.4	53.3835	51.4634	53.6408	53.2885
250	54	58.6466	49.0244	54.8544	55.2317
300	52.4	51.8797	52.439	53.8835	52.9895
333	50.8	57.5188	53.4146	58.7379	53.139
Average	50.16	53.72182	51.17072	53.22816	52.30941
Standard Variance	2.232686	3.221547	2.043203	2.816783	1.625442
Max	54	58.6466	54.878	58.7379	55.2317
Min	47.2	49.2481	48.5366	48.7864	49.1031

As TABLE IX shows, when the detection window length was 150 frames, the accuracy of DMFCC in detecting steganography using the algorithm suggested in [6] reached 80% for all the five types of speech samples, and increased further to over 90% at detection window length of 300 frames. This indicates that DMFCC is very effective in detecting the old steganography algorithm [6].

TABLE X shows the accuracy of DMFCC in detecting steganography with the proposed algorithm barely achieved 53% for five types of speech samples, with the maximum accuracy up to 56%, indicating that the proposed steganography algorithm is unlikely to be detected by DMFCC audio steganalysis.

TABLE XI

STEGANALYSIS RESULTS OF THE PROPOSED ALGORITHM USING THE MARKOV-DMFCC APPROACH [22] [23] AT DIFFERENT DETECTION WINDOWS (DATA EMBEDDING RATE OF 3 BITS/FRAME)

Windows Length (frames)	CSM (%)	CSW (%)	ESM (%)	ESW (%)	Hybrid (%)
1	48.4	46.6165	47.3171	45.8738	49.5516
10	48.8	48.1203	46.3415	46.3592	50.1495
20	49.2	48.4962	48.0488	47.8155	50.3737
40	50.8	48.8722	50.2439	47.8155	50.5232
80	51.6	49.2481	50.2439	48.0583	50.5979
150	51.6	50.7519	50.7317	48.0583	50.6726
200	52.4	51.1278	50.9756	51.699	51.42
250	52.4	51.8797	51.7073	52.1845	51.42
300	52.8	52.6316	52.1951	52.6699	52.1674
333	54	53.3835	52.439	53.1553	52.3916
Average	51.2	50.11278	50.02439	49.36893	50.92675
Standard Variance	1.866667	2.178093	2.099524	2.75128	0.901122
Max	54	53.3835	52.439	53.1553	52.3916
Min	48.4	46.6165	46.3415	45.8738	49.5516

We also adopted the latest DMFCC audio steganalysis, Second-order derivative-based Markov approach for audio steganalysis [22] [23], to detect VoIP steganography with the proposed steganographic algorithm, and the results are presented in TABLE XI. As TABLE XI shows, the average accuracy of Markov-DMFCC steganalysis in detecting steganography with the proposed algorithm just reached 51% for five different types of speech samples, with the maximum accuracy up to 54%, which means the proposed steganographic algorithm is unlikely to be detected by Markov-DMFCC steganalysis. This was probably due to the ineffectiveness of Markov-DMFCC steganalysis through analyzing Markov transition features, in detecting the proposed steganographic algorithm, which uses the pitch lag parameters substitution.

Fig. 9 shows comparisons of steganalysis results of two algorithms using DMFCC at different detection window lengths when Hybrid speech samples were used as cover media. As the detection window length increased, the accuracy of DMFCC in detecting the steganography algorithm presented in [6] improved significantly; the detection accuracy attained 90% when the detection window length reached 200 frames. By contrast, DMFCC was not effective in detecting the proposed steganography algorithm at different detection window lengths.

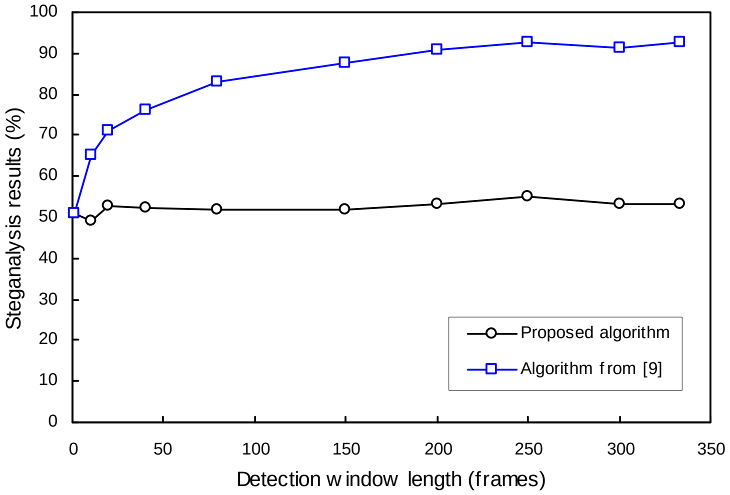

Fig. 9. Comparisons of steganalysis results of two algorithms using DMFCC at different detection window lengths

Fig. 10 shows the pitch distribution probabilities of G.723.1 VoIP samples (duration of 20 seconds) without and with data embedding. No obvious changes in the statistical property of the closed-loop pitch periods in the speech samples after G.723.1 codec without or with data embedding had been found for four types of VoIP audio samples, indicating that the proposed steganographic system retains the statistical property of original closed-loop pitch periods.

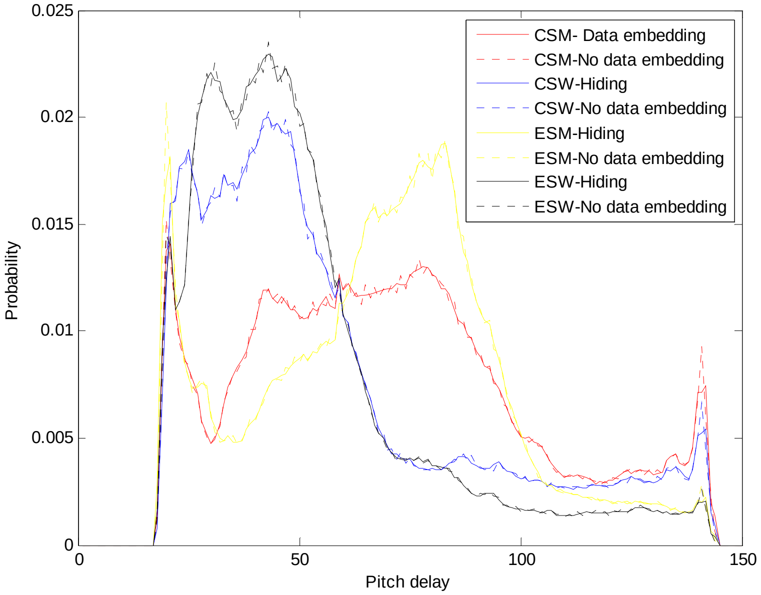

Fig. 10. Pitch distribution probabilities of G.723.1 VoIP samples (duration of 20 seconds) without and with data embedding

We carried out extra steganalysis experiments. As our proposed steganographic algorithm is based on pitch period prediction, pitch statistical characteristic-based steganalysis was specially designed in a way that suppose eavesdroppers know our steganographic algorithm (Kerckhoffs'-compliant), with VoIP samples of 3s, 5s, 10s, 20s and 30s in length with and without steganography being available, through analyzing pitch lag of VoIP samples with and without steganography eavesdroppers obtained the first-order pitch statistical characteristics, which were classified by using SVM (similar to DMFCC detection method in set-up), and the detection results are presented in TABLE XII. As the table shows, at five different detection window lengths, the accuracy in detecting steganography was below 70%, indicating that our proposed steganographic algorithm is capable of standing against steganalysis.

TABLE XII
STEGANALYSIS RESULTS OF THE PROPOSED ALGORITHM USING SVM AT DIFFERENT DETECTION WINDOWS

Window Length	CSM (%)	CSW (%)	ESM (%)	ESW (%)	Hybrid (%)
3s	60.8000	61.2782	60.2439	58.2524	59.5665
5s	64.8000	64.2857	63.6585	60.6796	63.9656
10s	67.2000	69.5489	66.3415	57.7670	64.3498
20s	68.8636	65.0000	66.5000	61.5909	66.8636
30s	69.8889	68.0000	67.8571	63.6500	68.6429

VI. CONCLUSIONS

In this paper, we have proposed a new method for steganography in low bit-rate VoIP streams based on pitch period prediction. On the basis of ITU G.723.1, a widely used low bit-rate speech codec, we have developed a much-improved G.723.1 speech codec with the information hiding functionality. Fifteen solutions for steganography have been suggested to perform on VoIP speech samples at four data embedding bit-rates taking into account the characteristics of G.723.1. The experimental results have shown that the worsening change in PESQ of the stego speech files obtained by using the proposed steganography algorithm was within 1.2%, indicating little impact on the quality of speech. In comparison with a previous algorithm [9], the proposed steganography algorithm has been found to have slightly larger effect on PESQ for 3s speech samples, but have less effect for 10s speech samples at 3 bits/frame data embedding rate; the worsening change in PESQ was 0.298% higher as the data embedding bit-rate reaching 4 bits/frame (33.3% increase than the old algorithm). Steganalysis tests using DMFCC-SVM have shown that the proposed steganography algorithm could prevent from being detected by steganalysis. Investigation into the applicability of the proposed algorithm to other low bit-rate speech codecs shall be the subject of future work. The steganalysis performance with different classifiers such as Fisher's linear classifier and logistic regression shall be part of future work.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61271392 and Grant 61272469.

REFERENCES

[1] C. Wang, and Q. Wu, "Information Hiding in Real-Time VoIP Streams," in *Proc. 9th IEEE International Symposium on Multimedia*, Taichung, Taiwan, 2007, pp. 255-262.

[2] S. Zander, G. Armitage, and P. Branch, "A Survey of Covert Channels and Countermeasures in Computer Network Protocols," *IEEE Communications Surveys and Tutorials*, vol. 9, no. 3, pp. 44-57, 2007.

[3] N. Aoki, "A technique of lossless steganography for G.711 telephony speech," in *Fourth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (IIHMSP2008), 2008, pp. 608-611.

[4] Po-Chyi Su, and C.-C. Jay Kuo, "Steganography in JPEG2000 Compressed images," *IEEE Transaction on Consumer Electronics*, vol. 49, no. 4, pp. 824-832, Nov. 2003.

[5] J. Dittmann, D. Hesse, and R. Hillert, "Steganography and steganalysis in voice over IP scenarios: operational aspects and first experiences with a new steganalysis tool set," in *Proc. SPIE, vol. 5681, Security, Steganography, and Watermarking of Multimedia Contents VII*, Mar. 2005, pp. 607-618.

[6] Y. F. Huang, Shanyu Tang, and Jian Yuan, "Steganography in Inactive Frames of VoIP Streams Encoded by Source Codec," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 296-306, 2011.

[7] Y. Su, Y. Huang, and X. Li, "Steganography-Oriented Noisy Resistance Model of G.729a," in *Proc. 2006 IMACS Multi-conference on Computational Engineering in Systems Applications*, Beijing, China, 2006, pp. 11-15.

[8] L. Liu, M. Li, Q. Li, and Y. Liang, "Perceptually Transparent Information Hiding in G.729 Bitstream," in *Proc. the 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, China, 2008, pp. 406-409.

[9] B. Xiao, Y. Huang, and S. Tang, "An Approach to Information Hiding in Low Bit-Rate Speech Stream," in *Proc. the 2008 IEEE Global Telecommunications Conference*, New Orleans, LA, USA, 2008, pp. 1-5.

[10] D. Yan, R. Wang, and L. Zhang, "Quantization Step Parity-based Steganography for MP3 Audio," *Journal Fundamental Informatics*, vol. 97, no.1-2, pp. 1-14, 2009.

[11] Fabien Petitcolas [Online]. Available: http://www.petitcolas.net/fabien/steganography/mp3stego/, accessed on 28 March 2012.

[12] M. Sheikhan, K. Asadollahi, and R. Shahnazi, "Improvement of Embedding Capacity and Quality of DWT-Based Audio Steganography Systems," *World Applied Sciences Journal*, vol. 10, no. 12, pp. 1501-1507, 2010.

[13] ITU.ITU-T Recommendation G.723.1.Dual Rate Speech Coder for Multimedia Communication Transmitting at 5.3 and 6.3 kbit/s, 1996. Available: http://www.itu.int/rec/T-REC-G.723.1-200605-I/en.

[14] ITU.ITU-T Recommendations G.729.Coding of speech at 8kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP), 2007. Available: http://www.itu.int/rec/T-REC-G.729/e.

[15] R. P. Ramachandran, and P. Kabal, "Pitch prediction filters in speech coding," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 37, no. 4, pp. 467-478, 1989.

[16] A. Maor, and N. Merhav, "On joint information embedding and lossy compression," *IEEE Transactions on Information*

Theory, vol. 51, no. 8, pp. 2998-3008, 2005.

[17] Y. Huang, S. Tang, C. Bao, and Y.J. Yip, “Steganalysis of compressed speech to detect covert voice over Internet protocol channels,” *IET Information Security*, vol. 5, no. 1, pp. 26-32, 2011.

[18] Qingzhong Liu, Andrew H. Sung, and Mengyu Qiao, “Temporal derivative-based spectrum and mel-cepstrum audio steganalysis,” *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 359-368, 2009.

[19] Y. Huang, S. Tang, and Y. Zhang, “Detection of covert voice over Internet protocol communications using sliding window-based steganalysis,” *IET Communications*, vol. 5, no. 7, pp. 929-936, 2011.

[20] Huang Yong-feng, Yuan Jian, and Chen Mingchao, “Key distribution in the covert communication based on VoIP,” *Chinese Journal of Electronics*, vol. 20, no. 2, pp. 357-361, 2011.

[21] Chih-Chung Chang, and Chih-Jen Lin, LIBSVM: a library for support vector machines [DB/OL]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/, assessed on 2 October 2011.

[22] Q. Liu, Andrew H. Sung, and M. Qiao, “Novel stream mining for audio steganalysis,” in *ACM Multimedia*, Beijing, China, 2009, pp. 95-104.

[23] Q. Liu, Andrew H. Sung, and M. Qiao, “Derivative-based audio steganalysis,” *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 7, no. 3, pp.18:1-18:9, 2011.