Steganography in Inactive Frames of VoIP Streams Encoded by Source Codec

Yong Feng Huang, Shanyu Tang, *Senior Member, IEEE*, and Jian Yuan

***Abstract*—This paper describes a novel high-capacity steganography algorithm for embedding data in the inactive frames of low bit rate audio streams encoded by G.723.1 source codec, which is used extensively in Voice over Internet Protocol (VoIP). This study reveals that, contrary to existing thought, the inactive frames of VoIP streams are more suitable for data embedding than the active frames of the streams; that is, steganography in the inactive audio frames attains a larger data embedding capacity than that in the active audio frames under the same imperceptibility. By analyzing the concealment of steganography in the inactive frames of low bit rate audio streams encoded by G.723.1 codec with 6.3 kb/s, the authors propose a new algorithm for steganography in different speech parameters of the inactive frames. Performance evaluation shows embedding data in various speech parameters led to different levels of concealment. An improved voice activity detection algorithm is suggested for detecting inactive audio frames taking into packet loss account. Experimental results show our proposed steganography algorithm not only achieved perfect imperceptibility but also gained a high data embedding rate up to 101 bits/frame, indicating that the data embedding capacity of the proposed algorithm is very much larger than those of previously suggested algorithms.**

***Index Terms*—Audio streams, inactive frames, steganography, Voice over Internet Protocol (VoIP).**

I. Introduction

STREAMING media, such as Voice over Internet Protocol (VoIP) streams, are broadcast live over the Internet and delivered to end-users. Security remains one of the main challenges with this new technology. With the upsurge of VoIP applications available for use in recent years, VoIP streams become one of the most interesting cover objects for modern steganography. Digital steganography in low bit rate audio streams is commonly regarded as a challenging topic in the field of data hiding.

There have been several steganography methods of embedding data in audio streams. For example, Wu *et al.* [1] suggested a G.711-based adaptive speech information hiding approach. Aoki [2] proposed a technique of lossless steganography in G.711 encoded speeches. Ma *et al.* [3] framed a steganography method of embedding data in G.721 encoded speeches. All these methods adopt high bit rate audio streams encoded by the waveform codec as cover objects, in which plenty of least significant bits exist.

However, VoIP are usually transmitted over low bit rate audio streams encoded by the source codec like ITU G.723.1 codec to save on network bandwidth. Low bit rate audio streams are less likely to be used as cover objects for steganography since they have fewer least significant bits than high bit rate audio streams. Little effort has been made to develop algorithms for embedding data in low bit rate audio streams. Chang *et al.* [4] embedded information in G.729 and MELP audio streams. Huang *et al.* [5] proposed a steganography algorithm for embedding information in low bit rate audio streams. But these steganography algorithms have constrains on the data embedding capacity; that is, their data embedding rates are too low to have practical applications. Thus the main focus of this study was to work out how to increase the data embedding capacity of steganography in low bit rate audio streams.

The rest of this paper is organized as follows. Section II summarizes some related work, discussing the possibility of embedding data in the inactive frames of low bit rate audio streams. In Section III, the imperceptibility of the steganography algorithm for embedding data in the inactive audio frames is analyzed. Our proposed steganography algorithm is presented in Section IV. Section V details the experimental setup and performance evaluation results. Finally, the paper ends with conclusions and directions for future work in Section VI.

II. Related Work

The analysis by synthesis (ABS)-based speech information hiding approach was adopted to embed speech data in an original speech carrier, with good efficiency in steganography and good quality of output speech [6]. Recently, linear predictive coefficients were substituted with secret speech data by using an ABS speech coding scheme [7], but the experimental results available are very limited.

Krätzer, Dittmann, and Vogel [8] argued that the inactive voice of a speech was not suitable for being used as a cover object for steganography owing to an obvious distortion of the original speech. By contrast, Huang *et al.* [9] suggested an algorithm for embedding information in some parameters of the speech frame encoded by ITU G.723.1 codec, without leading to distinction between inactive voices and active voices.

It seems that Krätzer, Dittmann, and Vogel's opinions [8] and Huang and coworkers' results [9] contradict each other. Such

Manuscript received August 13, 2010; revised January 11, 2011; accepted January 14, 2011. Date of publication January 28, 2011; date of current version May 18, 2011. This work was supported in part by the National Natural Science Foundation of China (Grant 60773140 and Grant 60703053), in part by the National Basic Research Program of China (Grant 2007CB310806), and in part by the British Government (Grant ktp6367). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wenjun Zeng.

Y. F. Huang and J. Yuan are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: yfhuang@tsinghua.edu.cn; jyuan@tsinghua.edu.cn).

S. Tang is with the Faculty of Computing, London Metropolitan University, London N7 8DB, U.K. (e-mail: s.tang@londonmet.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2011.2108649

1556-6013/$26.00 © 2011 IEEE

a contradiction can be attributed to the different speech codecs that were used to compress and encode audio signals. In [8], audio streams were encoded by a pulse-code modulation (PCM) codec; but an ITU G.723.1 source codec was used to encode audio streams in [9]. The PCM codec is based on the waveform model that samples, quantizes, and encodes audio signals directly; the sample value represents the original volume of the signal. In this case, the inactive voice cannot be used to embed information since it will lead to obvious distortion. However, the source codec is a hybrid codec, which is based on the source model. This codec compresses the speech at a very low bit rate and performs on a frame-by-frame basis; each frame is encoded into various parameters rather than the sample volumes. Thus the volume of the speech does not change imperceptibly even though their inactive audio frames contain hidden information.

The theoretical analysis above suggests that steganography in the inactive frames of low bit rate audio streams would attain a larger data embedding capacity if an appropriate steganography algorithm were used. The rest of this paper details our successful effort on such a new steganography algorithm for embedding data in the inactive frames of low bit rate audio streams encoded by ITU G.723.1 source codec.

III. Principle of Steganography in Inactive Audio Frames

A. Hangover Algorithm for Detecting Active Voices

To reduce network bandwidth in VoIP applications, some source codecs introduce silence compression during the inactive period of audio streams. The silence compression technique has two components: voice activity detection (VAD) and comfort noise generator [10]. The VAD is used to decide whether the current audio frame is an active voice by comparing the energy of the frame (Enr) with a threshold (Thr), as shown in (1)

$$\mathrm{VAD} = \begin{cases} 1, & \mathrm{Enr} \geq \mathrm{Thr} \\ 0, & \mathrm{Enr} < \mathrm{Thr}. \end{cases} \tag{1}$$

$\mathrm{VAD} = 0$ means the frame is an inactive voice; otherwise, the frame is an active voice.

The energy of the current frame Enr_t is computed by

$$\mathrm{Enr}_t = \frac{1}{80} \sum_{n=60}^{239} e_t^{'2}(n) \tag{2}$$

where $e_t'(n)$ is the output signal of the finite impulse response (FIR) filter whose input signal is the current frame $\{s[n]\}_{n=60,\ldots,239}$. The FIR filter computes $e_t'(n)$ using (3)

$$e_t'(n) = s[n] + \sum_{j=1}^{10} a_{no}[j] \cdot s[n-j] \quad n = 60 \rightarrow 239 \tag{3}$$

where $A_{\mathrm{no}}(Z) = \{a_{\mathrm{no}}[j] | j = 0, \ldots, 10\}$ is the autocorrelation coefficient vector of the filer.

The threshold in (1), Thr, is given by

$$\mathrm{Thr} = \begin{cases} 5.012, & \text{If Nlev} = 128 \\ 10^{0.7-0.05\log_2 \frac{\mathrm{nlev}}{128}}, & \text{If } 128 < \mathrm{Nlev} < 16384 \\ 2.239, & \text{If Nlev} \geq 16384 \end{cases} \tag{4}$$

Active voice	Vcnt	Hcnt	Silence compression
Enr ≥ Thresh	Enr < Thresh		
Normal encoding			Silence encoding

Fig. 1. Illustration of Hangover algorithm.

where Nlev_t is the noise size of the current frame, and is updated by its previous value, the energy of the previous frame Enr_{t-1}, and the self-adaptive flag Aen_t. Nlev_t is defined as follows:

$$\mathrm{Nlev}_t = \begin{cases} 0.25 \cdot \mathrm{Nlev}_{t-1} \\ \quad +0.75 \cdot \mathrm{Enr}_{t-1}, & \text{if } \mathrm{Nlev}_{t-1} > \mathrm{Enr}_{t-1} \\ \mathrm{Nlev}_{t-1}, & \text{else} \end{cases}$$
$$\mathrm{Nlev}_t = \begin{cases} 1.03125 \cdot \mathrm{Nlev}_t, & \text{if } \mathrm{Aen}_t = 0 \\ 0.9995 \cdot \mathrm{Nlev}_t, & \text{else} \end{cases} \tag{5}$$

where $\mathrm{Aen}_t = [0, 6]$, and Nlev_t is limited to a value between 128 and 131 071.

In general, the Hangover algorithm is used for detecting inactive voices to avoid noise peaks being extended [10]. If an audio frame is determined to be an inactive voice, the frame is encoded into a silence insert description (SID) frame by using the silence compression algorithm. Having received the SID frame, the decoder generates a comfortable noise at the receiving end. The Hangover algorithm is illustrated in Fig. 1.

The first row in Fig. 1 shows the classification of voice duration before silence compression. Hcnt is the Hangover-frame number of inactive voices when an active voice begins to change to an inactive voice in the speech. The second row is an estimate of the energy and the third row includes the corresponding codec algorithms.

An audio stream is actually divided into frames before being encoded. For instance, with G.723 codec the audio stream is divided into frames 30 ms in length. Suppose the audio stream F contains N frames, $F = \{f_i | i = 0, \ldots, N\}$. If the energy (Enr) of the frame f_i is less than the threshold value (Thr), $\mathrm{Enr} < \mathrm{Thr}$, the frame is the first frame of an inactive voice. This frame is defined as a Hcnt frame in Hangover algorithm and is then encoded by using the normal codec algorithm rather than the silence compression algorithm. If subsequent frames are still inactive voices, the Hangover algorithm will not perform silence compression until the sixth frame. In other words, the Hangover algorithm starts to encode the sixth frame of the inactive voice into a SID frame until the next active voice emerges. The first five frames (first to fifth) of the inactive voice are still encoded into Hangover frames, denoted by $f^{*\mathrm{Hcnt}}$. The active voice of the audio stream is encoded into active frames f^{*A} by using the normal codec algorithm.

According to the Hangover algorithm, audio frames are classified into three types, active voice frame f^A, Hangover frame f^{Hcnt}, and silence compression frame f^S. The audio speech F can be expressed as

$$F = \left\{ f_i^A, f_j^s | i = 0, \ldots, N_1, \;\; j = 0, \ldots, N_2, \;\; N = N_1 + N_2 \right\}. \tag{6}$$

The speech F is then encoded into F^* by using Hangover algorithm, which can be written as

$$\begin{aligned} F^* &= \varphi(F) \\ F^* &= \left\{ f_i^{*A}, f_j^{*\mathrm{Hcnt}}, f_l^{*\mathrm{SID}} | i = 1, \ldots, n_1, \;\; j = 1, \ldots, n_2, \right. \\ &\qquad \left. l = 1, \ldots, n_3, \right. \\ &\qquad \left. N = n_1 + n_2 + n_3 \right\}. \end{aligned} \tag{7}$$

B. Definitions of Inactive and Active Frames

The silence compression technique is an optional function for the source codec. In fact, most source codecs do not use silence compression in VoIP applications. All audio frames are encoded uniformly by using the normal encoding algorithm regardless of whether they are active voices or inactive voices. Thus two types of frames are outputted when the speech stream F is encoded by the source codec. For example, ITU G.723.1 codec encodes the speech into two types of frames, active frames and inactive frames, without using the silence compression algorithm.

Definition 1: The active frame f^{*A} is encoded by the source codec from the active voice of the speech. It is expressed as

$$f_i^{*A} = \varphi\left(f_i^A\right), \qquad i = 0, \ldots, N_1. \tag{8}$$

Definition 2: The inactive frame f^{*S} is encoded by the source codec from the inactive voice of the speech. It is expressed as

$$f_i^{*S} = \varphi\left(f_i^S\right), \qquad j = 0, \ldots, N_2. \tag{9}$$

As the speech is divided into inactive voices and active voices by VAD, all the voices are encoded uniformly by the source codec to form audio frames, in which inactive frames can be distinguished from active frames. Combining (6)–(9) yields

$$\begin{aligned} F^* = \left\{ f_i^{*A}, f_j^{*S} | i = 0, \ldots, N_1, \;\; j = 0, \ldots, N_2, \right. \\ \left. N = N + 1 + N_2 \right\}. \end{aligned} \tag{10}$$

C. Bit Distribution Patterns of Inactive Frames

This section discusses whether the "1/0" distribution pattern of an inactive frame is similar to that of an active frame if the inactive voice of a speech is encoded into inactive frames.

First, we analyzed the statistical probability of "1/0" presentation in inactive frames. Assuming an audio stream is divided into N frames, among them there are N_1 inactive frames and N_2 active frames, i.e., $N = N_1 + N_2$. The audio stream is denoted by $F^* = \{f_i^{*A}, f_j^{*S} | i = 0, \ldots, N_1, j = 0, \ldots, N_2\}$. Suppose each frame consists of M bits, namely $f_i^* = \{b_0, \ldots, b_i, \ldots, b_M | b_i = 0, 1\}$. The average probability of "1" presentation in all the inactive frames is computed by using

$$\rho_{b1} = \frac{1}{MN_1} \sum_{i=0}^{N_1 - 1} \sum_{j=0}^{M-1} b_{i,j} \tag{11}$$

where $b_{i,j}$ denotes the jth "1" in the ith inactive frame of the stream. So the average probability of "0" presentation in all the inactive frames is given by

$$\rho_{b0} = 1 - \rho_{b1}. \tag{12}$$

TABLE I
AVERAGE PROBABILITIES OF "1" PRESENTATION IN INACTIVE AND ACTIVE FRAMES

Speech	Active frame		Inactive frame	
file No	Mean	Variance	Mean	Variance
1	48.2	0.388	45.0	0.320
2	47.8	0.728	48.9	0.400
3	47.4	0.112	47.5	0.340
4	48.4	0.440	44.7	0.272
5	48.5	0.380	49.6	0.240
6	47.7	0.304	48.8	0.312
7	47.7	0.436	44.5	0.308
8	48.3	0.536	48.5	0.404
9	48.2	0.692	44.5	0.400
10	53.7	0.512	48.9	0.388

Table I lists the experimental results of the statistical probabilities of "1" presentation in the inactive frames and active frames encoded by G.723.1 codec, respectively. Ten speech sample files were used for the experiments, with each file being tested six times in order to work out the average probabilities of "1" presentation in the inactive and active frames.

The average probabilities shown in Table I indicate that there was no obvious difference in the "0/1" presentation probability between the active frames and inactive frames of low bit rate audio streams. In other words, we could not distinguish inactive frames from active frames by the "1/0" presentation probability.

Second, we examined the probability of "1/0" jumping in inactive frames. The "1/0" jumping probability expresses the chance that the bit "1" changes to "0" or "0" to "1" inversely will occur in an inactive frame. Similarly, the audio stream is denoted by $F^* = \{f_i^{*A}, f_j^{*S} | i = 0, \ldots, N_1, j = 0, \ldots, N_2\}$, and $f_i^{*s} = \{b_0, \ldots, b_i, \ldots, b_M | b_i = 0, 1\}$. Then the average probability of "1/0" jumping in all the inactive frames of the speech file is calculated by

$$\rho_c = \frac{1}{M \cdot N_2} \sum_{i=0}^{N_2} \sum_{j=0}^{M-1} |b_i - b_{i-1}| \times 100\% \tag{13}$$

where b_i denotes the jth bit in the inactive frame and M denotes the bit number of the inactive frame, such as $M = 192$ for G.723.1 codec with 6.3 kb/s.

Table II shows the average probabilities of "1/0" jumping in the inactive frames and active frames of low bit rate audio streams, respectively. Each mean probability is based on six repeated experiments on a speech file. The results suggest both inactive frames and active frames were indistinguishable in terms of the jumping probability.

Finally, we studied the run-length statistical character of "0/1" in inactive frames. The run-length statistical method was used to calculate the run-lengths of continuous "0" or "1" presentation in inactive frames. Assuming the audio stream is denoted by $F^* = \{f_i^{*A}, f_j^{*S} | i = 0, \ldots, N_1, j = 0, \ldots, N_2\}$ and $f_i^{*s} = \{b_0, \ldots, b_i, \ldots, b_M | b_i = 0, 1\}$, it satisfies the following equation:

$$\begin{cases} b_k \neq b_{k-1} \\ b_{k+i} = b_{k+i-1} \\ b_{k+R} \neq b_{k+R-1} \end{cases} \tag{14}$$

where $i = 0, 1, \ldots R$, and R denotes the run-length of b_k in an inactive frame. Then the run-length of b_k ($b_k = 0, 1$) in the

TABLE II
AVERAGE PROBABILITIES OF "1/0" JUMPING IN INACTIVE AND ACTIVE FRAMES

Speech	Active frame		Inactive frame	
file no	Mean	Variance	Mean	Variance
1	52.3	0.288	50.2	0.364
2	54.1	0.140	54.6	0.188
3	51.7	0.484	48.6	0.440
4	50.4	0.396	48.4	0.400
5	48.9	0.284	59.6	0.132
6	49.7	0.492	60.2	0.275
7	55.7	0.488	53.6	0.452
8	47.2	0.328	52.4	0.228
9	51.6	0.192	54.2	0.256
10	48.3	0.324	51.7	0.464

inactive frame is equal to the number of bits from b_k to b_{k+R-1}. The distribution pattern of the run-length in inactive frames is defined as the probability of various run-lengths presenting in all inactive frames of the speech file, given by

$$\rho_0(i) = \frac{1}{M \cdot N_2} \sum_{j=0}^{N_2} M_j^0(i) \times 100\%, \quad i = 0, \ldots R \quad (15)$$

$$\rho_1(i) = \frac{1}{M \cdot N_2} \sum_{j=0}^{N_2} M_j^1(i) \times 100\%, \quad i = 0, \ldots R \quad (16)$$

where $\rho_j(i), j = 0, 1$ denotes the percent of the run-length of the bit "0" or "1" being equal to i in all inactive frames, and $M^0(i)$ and $M^1(i)$ denote the numbers of the run-length of the bit "0" or "1" being equal to i in all inactive frames, respectively.

The Mann–Whitney–Wilcoxon (M-W-W) test, one of the best-known nonparametric significance tests, was used to evaluate whether the difference in run-length probability distributions between the inactive frames and active frames of a speech file is indistinguishable. To have 95% confidence, i.e., with a confidence coefficient $(1 - \alpha)$ of 0.95, where α is called the level of significance, we therefore require $z(1 - \alpha/2) = z(0.975) = 1.960$, where z is the percentile of the standard normal distribution. Hence, if the standardized test statistic $|z^*| \leq 1.960$, two distributions do not differ.

Table III describes the distribution patterns of the run-lengths of "0" and "1" in all inactive frames and active frames, and the M-W-W test results for comparing the probability distributions between the inactive frames and active frames for four speech samples, respectively. Since $|z^*| \leq 1.960$ for all the cases, we conclude that the probability distributions for both inactive frames and active frames do not differ, indicating that the inactive frames and active frames had a similar run-length pattern for each speech file.

To summarize, the above three experiments on bit distribution patterns indicate that the bit distribution of inactive frames is similar to that of active frames for the same speech files. Otherwise stated, it is highly unlikely to use the "1/0" distribution pattern to distinguish the inactive frames from active frames of low bit rate audio streams.

D. Steganography in Inactive Frames

The source codec like ITU G.723.1 is operated on a frame-by-frame basis. Each frame encoded by G.723.1 codec has 240 audio samples that are encoded according to PCM. First of all, each frame is filtered by a high-pass filter to remove the dc component and is then divided into four subframes of 60 samples each. A tenth order linear predictive coding (LPC) filter is computed using the unprocessed input signal for every subframe, and the last subframe is quantized using a predictive split vector quantizer. For every two subframes (120 samples), the weighted speech signal is used to compute the open-loop pitch period. A harmonic noise shaping filter is then constructed using the open-loop pitch period computed previously, and a closed-loop pitch predictor is constructed according to the impulse response created by the noise shaping filter. Finally, both the pitch period and the differential value are transmitted to the decoder and the nonperiodic component of the excitation is approximated. After completion of these operations, all speech parameters such as LPC, Pulse sign (Pamp), and Pulse position (Ppos) and so on, are obtained.

TABLE III
RUN-LENGTH PATTERNS IN INACTIVE AND ACTIVE FRAMES

Probability	Frame	Speech file no			
	type	1	2	3	4
$\rho_0(1)$	Inactive	11.5	9.9	11.2	13.4
	Active	11.5	8.9	10.4	11.5
$\rho_1(1)$	Inactive	9.9	10.9	8.7	12.2
	Active	7.2	7.8	9.9	12.5
$\rho_0(2)$	Inactive	10.4	14.6	12.5	7.3
	Active	8.3	11.5	8.5	9.2
$\rho_1(2)$	Inactive	14.6	10.4	8.5	11.5
	Active	14.6	12.5	10.4	10.5
$\rho_0(3)$	Inactive	14.06	7.8	9.8	11.7
	Active	4.7	10.9	9.4	9.9
$\rho_1(3)$	Inactive	4.7	9.4	11.3	7.8
	Active	3.6	6.25	12.4	8.7
$\rho_0(4)$	Inactive	4.2	8.3	10.5	7.5
	Active	10.4	4.2	7.2	4.2
$\rho_1(4)$	Inactive	6.25	8.3	11.8	9.3
	Active	2.1	6.25	8.4	8.3
$\rho_0(>4)$	Inactive	6.25	9.9	8.6	3.0
	Active	27	10.9	8.7	10.5
$\rho_1(>4)$	Inactive	18.2	10.4	6.9	16.3
	Active	2.6	20.3	14.2	10.6
Test statistic ($\|z^*\|$)		0.605	0.076	0.378	0.227

According to (6)–(10), if the speech, $F = \{f_i^A, f_j^S | i = 0, \ldots, N_1, j = 0, \ldots, N_2, N = N_1 + N_2\}$, is inputted into G.723.1 codec, then the bit stream $F^* = \{f_i^{*A}, f_j^{*S} | i = 0, \ldots, N_1, j = 0, \ldots, N_2, N = N_1 + N_2\}$ is outputted with two types of frames, inactive frames and active frames. Moreover, the bit allocation of the inactive frame is similar to that of the active frame. The bit allocation of G.723.1 codec with 6.3 kb/s is listed in Table IV.

The next step is to determine which speech parameters of inactive frames are suitable for data embedding. All the speech parameters are sorted into three imperceptibility levels of steganography in terms of the distance of signal-to-noise ratio (DSNR), which is defined as the difference in signal-to-noise ratio (SNR) between the original speech and stego speech. Close analysis of the data in Table V shows the imperceptibility levels of steganography for different parameters of the inactive frames are widely different. So it is possible to choose different parameters and various parameter bits to embed data on demand of practical applications. In short, the parameters marked with level 1–2 are suitable cover objects for steganography.

TABLE IV
BIT ALLOCATION OF G.723.1 CODEC WITH 6.3 kb/s

Parameters	Sub-frame 0	Sub-frame 1	Sub-frame 2	Sub-frame 3	Subtotal (bits)
Adaptive codebook lags (Olp/Aclg)	7	2	7	2	18
LPC indices (Lsf)	-	-	-	-	24
Grid index (Grid)	1	1	1	1	4
All the gains combined (Mamp)	12	12	12	12	48
Pulse positions (Ppos)	20	18	20	18	73
Pulse signs (Pamp)	6	5	6	5	22
Total	-	-	-	-	189

TABLE V
IMPERCEPTIBILITY LEVELS OF STEGANOGRAPHY IN VARIOUS PARAMETERS OF INACTIVE FRAMES

Number of bits	Olp (s1)	Lsf (s2)	Aclg (s3)	Grid (s4)	Mamp (s5)	Ppos (s6)
7	-	3	-	-	-	1
6	3	3	-	-	-	1
5	3	3	-	-	-	1
4	3	2	-	-	3	1
3	2	2	-	-	3	1
2	2	2	-	-	2	1
1	1	1	2	-	2	1
0	1	1	2	1	1	1

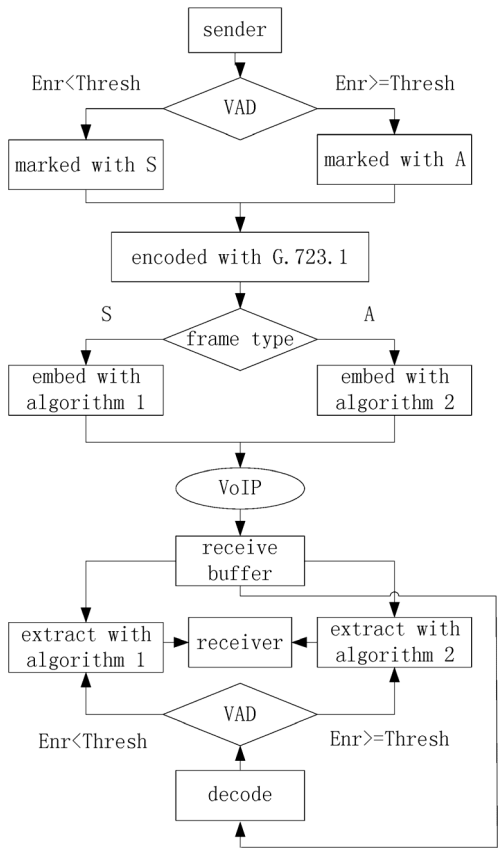

Fig. 2. Flowchart of steganography in inactive and active frames.

IV. OUR PROPOSED ALGORITHM FOR STEGANOGRAPHY IN INACTIVE FRAMES

Our steganography model is illustrated in Fig. 2, where VAD, data embedding, and audio frame encoding are carried out sequentially in the speech coding process. The sender samples an audio signal and encodes it into a PCM formatted audio stream, $F = \{f_i | i = 0, \ldots, N\}$. The VAD algorithm is then used to detect the inactive voice in the stream. If the current frame f_i is an inactive voice, the frame is marked with S; otherwise, it is marked with A. As a result, the audio stream is divided into a sequence of frames, $F = \{f_i^A, f_j^S | i = 0, \ldots, N_1, j = 0, \ldots, N_2, N = N_1 + N_2\}$. All the frames are then encoded uniformly by G.723.1 codec into a low bit rate stream, which is called the original speech, $F^* = \{f_i^{*A}, f_j^{*S} | i = 0, \ldots, N_1, j = 0, \ldots, N_2, N = N_1 + N_2\}$.

The low bit rate stream contains two types of frames, inactive frames and active frames. According to the frame type, two different steganography algorithms are then used, respectively, to embed the secret information, $S = (s_1, s_2, \ldots s_i, \ldots s_n)$, $s_i \in (0, 1)$, in the stream. That is, the algorithm 1 suggested below is used to embed information in inactive frames; the algorithm 2 presented in [5] is used for steganography in active frames. The low bit rate stream with hidden information is called the stego speech, denoted by $\widetilde{F} = \{\widetilde{f}_1, \widetilde{f}_2, \ldots, \widetilde{f}_i\}$, which is transmitted using VoIP. Afterwards, the receiver receives the stego speech, from which the secret information is finally extracted.

To sum up, the steganography process has three subprocesses, voice activity detection, data embedding, and extracting. The corresponding algorithms are detailed below.

A. Improved VAD Algorithm

Hangover algorithm is normally used for voice activity detection in the speech coding process. To synchronize the embedding and extraction in steganography, it is very important to keep the VAD result consistent between the sender and receiver because an inconsistent VAD result will result in errors in the extracting process. Some factors, such as packet loss, steganography and so on, may have an impact on the VAD result. So an improved VAD algorithm called the residual energy method is suggested below.

The residual energy method adopts the autocorrelation coefficient, which is not affected by the state of the codec, to detect the inactive voice in the speech. The coefficient vector of the FIR filter on (3), $A_{no}(Z)$, is computed by Levision–Durbin algorithm as follows:

$$\begin{bmatrix} \overline{R}_p[0] & \overline{R}_p[1] \cdots \overline{R}_p[10] \\ \overline{R}_p[1] & \overline{R}_p[0] \cdots \overline{R}_p[9] \\ \vdots & \vdots \ddots \vdots \\ \overline{R}_p[10] & \overline{R}_p[9] \cdots \overline{R}_p[0] \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ -a_{10} \end{bmatrix} = \begin{bmatrix} E_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (17)$$

As (17) reveals, the autocorrelation sum of the frame $\overline{R}_p[j]$ must be computed in advance by using (18) in order to obtain $A_{no}(Z)$

$$\overline{R}_p[j] = \sum_{k=t-3}^{t-1} R_i^k[j], \qquad j = 0, \ldots, 10 \tag{18}$$

where $R_i[j]$ denotes the autocorrelation of the subframe i.

As a frame consists of four subframes, each of which has 11 autocorrelation coefficients, all the autocorrelation coefficients for the frame can be described as $R_i[j]$, $j = 0, \ldots, 10$, $i = 0, \ldots, 3$. To compute the coefficients for the first subframe, it needs to obtain the data of three continuous subframes. The continuous subframes, $(i-1)$th, ith, $(i+1)$th subframes, can be combined to form a sequence, $\mathrm{ThreeSubFrm}_i$, which contains 180 samples.

When $i = 0$, the $(i-1)$th subframe belongs to the previous frame. If the predecessor of the current frame is lost, an error will occur in calculating the autocorrelation coefficients of the current frame. This is because Hangover algorithm has memory and error propagation would result from lost or delayed packets. In an attempt to solve this problem, we suggest an improved stateless algorithm for computing the autocorrelation $R_i[j]$. The algorithm is described in detail below.

First, $\mathrm{ThreeSubFrm}H_i(n)$ is computed through windowing/applying a Hamming window in the sequence of frames, given by

$$\begin{aligned}\mathrm{ThreeSubFrm}H_i(n) &= \mathrm{ThreeSubFrm}_i(n) \\ &\times \mathrm{HammiWindow}(n), \quad n = 0, \ldots, 179\end{aligned} \tag{19}$$

Second, the autocorrelation coefficients of the subframe are computed by

$$\begin{aligned}R_i[n] = \frac{1}{180 \times 180} \sum_{j=0}^{179-n} &\mathrm{ThreeSubFrm}H_i(j) \\ &\times \mathrm{ThreeSubFrm}H_i(n+j)\end{aligned} \tag{20}$$

where $n = 0, \ldots, 10$, which is the number of autocorrelation.

Third, a white noise is used to adjust the first coefficient, $R_i[0]$, as shown as follows:

$$R_i[0] = R_i[0] \times \frac{1025}{1024}. \tag{21}$$

And a binomial window is used to adjust the other coefficients by means of the following equation:

$$R_i[n] = R_i[n] \times \mathrm{Binormial}[n], \quad n \neq 0. \tag{22}$$

Equation (22) indicates that 180 samples needed for computing $R_i[n]$ are located in two continuous frames, the previous frame and the current frame. As the speech has the short-term stationary property, the samples in the previous frame can be replaced with one of the current frame in (19). Therefore, even if the previous frame is lost, it will not affect the computational results of autocorrelation coefficients for the current frame.

Finally, once all the autocorrelation coefficients of four subframes are obtained for a frame, the residual energy of the frame can be computed by the Yule–Walker equation:

$$\begin{bmatrix} R^t[0] & R^t[1] \cdots R^t[10] \\ R^t[1] & R^t[0] \cdots R^t[9] \\ \vdots & \vdots \ddots \vdots \\ R^t[10] & R^t[9] \cdots R^t[0] \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ -a_{10} \end{bmatrix} = \begin{bmatrix} E \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{23}$$

where $R^t[j]$ $(j = 0, \ldots, 10)$ denotes the autocorrelation coefficients, and a_j $(j = 1, \ldots, 10)$ are the LPC coefficients. The algorithm for solving the Yule–Walker equation is described below.

Step 1) Initialization: $i = 0$, $E = R^t[j]$, $a_j = 0$ and $j = 1, \ldots, 10$.

Step 2) Compute k using (24)

$$k = \frac{R^t[i+1] - \sum_{j=0}^{i-1} a_{j+1} \cdot R^t[i-j]}{E}. \tag{24}$$

If $i - 1 < 0$, then $\sum_{j=0}^{i-1} a_{j+1} \cdot R^t[i-j] = 0$.

Step 3) Compute a_j by

$$a_j \Leftarrow \begin{cases} k, & j = i+1 \\ a_j - k \cdot a_{i-j}, & 1 \leq j \leq i. \end{cases} \tag{25}$$

Step 4) Compute E as follows:

$$E \Leftarrow (1 - k^2) \cdot E. \tag{26}$$

And $i = i + 1$, if $i < 10$, go to Step 2. Otherwise, the final residual energy E_t yields; $E_t = E$ when $i = 10$.

A new method of detecting active voices is then suggested here, that is, comparing the threshold with the residual energy of the frame rather than the energy of the frame, as shown in (27)

$$\begin{cases} \text{Inactive frame}, & \text{if } E_t < \mathrm{Thr}_t \\ \text{Active frame}, & \text{if } E_t \geq \mathrm{Thr}_t \end{cases} \tag{27}$$

where E_t denotes the residual energy of the current frame, and Thr_t denotes the threshold, which is an empirical value obtained from experiments.

The above VAD method that is based on the residual energy instead of the frame energy is only related to the $R^t[j]$ $(j = 0, \ldots 10)$ of the current frame when the residual energy is computed. So the improved VAD method is not affected by packet loss, thereby guaranteeing the VAD result to be consistent between the sender and receiver.

B. Embedding Algorithm

The embedding process is divided into four steps as shown in Fig. 2.

Step 1) Voice activity detection. The speech with PCM format is divided into frames, $F = \{f_1, \ldots, f_i\}$. Each frame f_i is inputted into the VAD detector that adopts the residual energy algorithm above. The frame is marked with "A" if it is determined to

be an active voice; otherwise, the frame is marked with "S." The frames are defined as

$$\begin{cases} f_i = f_i^A, & \text{if } f_i \text{ is an active voice} \\ f_i = f_i^S & \text{else.} \end{cases} \tag{28}$$

The sequence of the frames with marks is then obtained, given by

$$F = \left\{ f_i^A, f_j^S | i = 0, \ldots, N_1, \;\; j = 0, \ldots, N_2 \right\}. \tag{29}$$

Step 2) Encoding all frames by G.723.1 codec. Regardless of the frame type, all the frames, f_i^A and f_j^S, are encoded by using the standard G.723.1 algorithm with 6.3 kb/s. The resulting low bit rate audio stream containing active and inactive frames is then outputted from the codec. The low bit rate audio stream is expressed as

$$\begin{aligned} F^* &= \left\{ f_i^{*A}, f_j^{*S} | i = 1, \ldots, N_1, \;\; j = 0, \ldots, N_2 \right\} \\ F^* &= \varphi(F) \\ &= \left\{ \varphi\left(f_i^A\right), \varphi\left(f_j^S\right) | i = 1, \ldots, N_1, \;\; j = 0, \ldots, N_2 \right\}. \end{aligned} \tag{30}$$

Step 3) Embedding information in frames. According to the frame type, two different steganography algorithms are used to embed information in the frames. They are expressed as

$$\begin{aligned} \widetilde{f}_i &= \varphi_1\left(f_i^*, S\right) = f_i^{*S} \otimes S, \quad \text{if } f_i^* = f_i^{*S} \\ \widetilde{f}_i &= \varphi_2\left(f_i^*, S\right) = f_i^{*A} \otimes S, \quad \text{if } f_i^* = f_i^{*A}. \end{aligned} \tag{31}$$

The expression $\varphi_1(f_i^*, S)$ means the algorithm 1 is used to embed the secret information S in the inactive frame. The expression $\varphi_2(f_i^*, S)$ denotes the algorithm 2 is used to embed the information in the active frame. So the stego speech is given by

$$\widetilde{F} = \{\widetilde{f}_1, \widetilde{f}_2, \ldots, \widetilde{f}_i\}. \tag{32}$$

Step 4) Encapsulation and sending. The inactive frames and active frames with hidden information are encapsulated in VoIP packets $P = \{p_i | p_i = \phi(\widetilde{f}_i), i = 1, \ldots, n\}$, which are transmitted over the Internet.

C. Extracting Algorithm

The extraction of secret information from the stego speech is the inverse process of the embedding algorithm, and it is divided into the following three steps.

Step 1) Receiving and decapsulation. The VoIP packets, $P = \{p_1, p_2, \ldots, p_n\}$, are received, buffered, and then decapsulated by the receiver. The decapsulation algorithm is described as

$$\widetilde{F} = \left\{ \widetilde{f}_i | \widetilde{f}_i = \phi^{-1}(p_i), \;\; i = 1, \ldots, n \right\}. \tag{33}$$

Step 2) Decoding and active frame detection. The buffered frames $\widetilde{F} = \{\widetilde{f}_i | i = 1, \ldots, n\}$ are copied to the decoding buffer and decoded into the PCM formatted audio stream $F' = \{f_i' | i = 1, \ldots, n\}$. The improved VAD method is then used to distinguish between inactive frames and active frames, $F' = \{f_i'^A, f_i'^S | i = 1, \ldots, N_1, j = 1, \ldots, N_2\}$.

Step 3) Extracting secret information. The inactive and active frames of the low bit rate audio stream $\widetilde{F} = \{\widetilde{f}_i | i = 1, \ldots, n\}$ are identified by referring to $F' = \{f_i'^A, f_i'^S | i = 1, \ldots, N_1, j = 1, \ldots, N_2\}$. The secret information is then extracted from $\widetilde{F} = \{\widetilde{f}_i | i = 1, \ldots, n\}$ by using Algorithms 1 and 2. Algorithms 1 and 2 are used to extract the secret information from the inactive frames and the active frames, respectively.

V. Performance Analysis of Steganography in Inactive Frames

In our experiments, voice activity detection, data embedding, and audio encoding operations were conducted in sequence for each speech sample by means of the corresponding algorithms detailed in Section IV.

Two parameters, imperceptibility and data embedding capacity, were used to evaluate the performance of the proposed steganography algorithm. Twenty speech samples files with PCM format were employed as cover objects for steganography, and they are classified into four groups, Group 1, Group 2, Group 3, and Group 4 (Table VI). Secret information was embedded in the inactive frames of the speech files, the imperceptibility of the resulting stego files was then evaluated, and the data embedding capacity was estimated accordingly for each speech file. The experimental results are discussed in detail below.

To verify the imperceptibility of steganography in various parameters of inactive frames, the same secret information was embedded in each parameter of the 20 speech files encoded by G.723.1 codec, and the DSNR values of the resulting stego speech files were then computed. The DSNR is defined as the difference in SNR between the original speech and the stego speech, given by

$$\mathrm{DSNR} = |\mathrm{SNR}_b - \mathrm{SNR}_a| \tag{34}$$

where SNR_b and SNR_a are the SNRs of the original speech and the stego speech, respectively.

Fig. 3 shows the results of experiments on the 20 speech files listed in Table VI, with the horizontal axis representing the number of bits of the parameter that are replaced by secret information. Experiments indicate that in most instances the DSNR value between the original speech and the stego speech was so small that the distortion of the stego speech was unlikely to be perceived as long as appropriate parameter bits of inactive frames were used to embed the secret information. The overall trend in DSNR was upward with increasing bit numbers of embedding. The parameters with DSNR values of less than 0.5 dB were chosen to embed information.

As shown in Fig. 3, when the bit number of hidden information in the Lsf parameter was not more than 3 bits, the DSNR value was under 0.5 dB; however, DSNR rose significantly

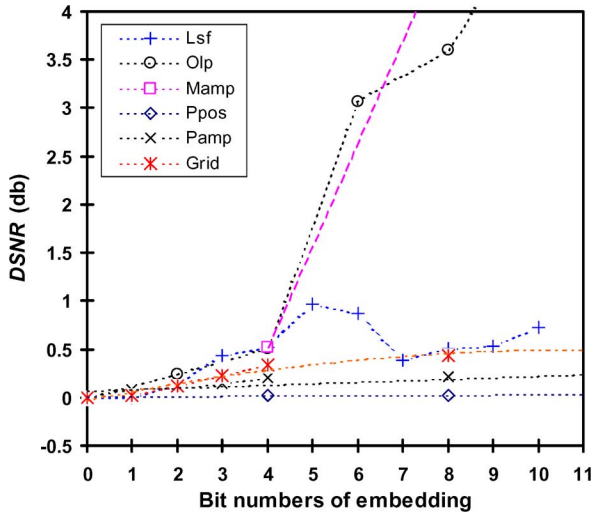

Fig. 3. DSNR for steganography in various parameters of inactive frames.

TABLE VI
NUMBERS OF INACTIVE FRAMES OF 20 PCM SPEECH FILES

Group	Speech file name	File length (s)	Number of inactive frames (30 ms)	Average no of inactive frames
Group 1	MC1	10	183	138
	MC2	10	101	
	MC3	10	121	
	MC4	10	120	
	MC5	10	166	
Group 2	WC1	10	139	125
	WC2	10	125	
	WC3	10	110	
	WC4	10	147	
	WC5	10	106	
Group 3	ME1	10	69	52
	ME2	10	59	
	ME3	10	48	
	ME4	10	42	
	ME5	10	45	
Group 4	WE1	10	51	59
	WE2	10	53	
	WE3	10	56	
	WE4	10	66	
	WE5	10	69	

when more than 4 bits of information were embedded. This means no more than 3 bits of information should be embedded in the Lsf parameter. For the Olp and Mamp parameters, even replacing 1 bit in each subframe with secret information (amounting to 4 bits hidden information per frame) resulted in a larger DSNR value, indicating that both parameters are not suitable for data embedding. By looking at the DSNR curves in Fig. 3 and the imperceptibility levels of steganography in Table V, we realized that all bits of the Ppos, Pamp, and Grid parameters could be used to embed information in inactive frames. We, therefore, selected five parameters of inactive frames (Table VII) to carry out further steganography experiments.

As a frame of G.723.1 with 6.3 kb/s has 192 bits, and the total number of replaceable parameter bits in an inactive frame is 101 bits, the data embedding capacity ratio C_r for the inactive frame is determined by

$$C_r = \text{Embedding bits/Total bits} = 101/192 = 52.6\%. \tag{35}$$

TABLE VII
PARAMETERS OF THE INACTIVE FRAME PERFECTLY SUITABLE FOR DATA EMBEDDING

Parameter name	Lsf	Grid	H_Ppos	L_Ppos	Pamp	Total bits
Number of bits	2	4	13	60	22	101

TABLE VIII
PERCENTAGES OF FAILURES USING $A/B/X$ TEST

	Group 1	Group 2	Group 3	Group 4
Tester 1	55%	30%	50%	55%
Tester 2	30%	55%	65%	30%
Tester 3	60%	30%	60%	35%
Tester 4	55%	45%	50%	60%
Tester 5	45%	40%	60%	55%
Average	49%	40%	57%	47%

A. Imperceptibility

According to the improved VAD algorithm, we counted the number of inactive frames for each speech file (Table VI), and encoded these files into low bit rate streams using G.723.1 codec with 6.3 kb/s. Five parameters of the inactive frame (Table VII) were selected to embed information. We then evaluated the imperceptibility of the stego speech files in terms of subjective quality and objective quality.

Subjective Quality: The "$A/B/X$" test method, ITU P.860 recommendation [12], was utilized to assess the subjective quality of the stego speech files. This method is described in detail as follows. Suppose there are three types of speech files, denoted by A, B, and X, respectively. A represents the stego speech file containing hidden information, B denotes the original speech file without any hidden information, and X is either A or B. Five evaluators were employed to listen the speech files, and then asked to decide whether X is A or B.

Speech samples were chosen randomly from the four groups of speech files listed in Table VI. Each tester made 20 judgments in total, some of which were successful and the other judgments were failures. These failure judgments include negative failures and positive failures. Table VIII shows the percentage of failures to identify the stego speech file.

Close analysis of the data listed in Table VIII shows the average percentage of failure judgments was 48.25%. This means it was impossible to distinguish the stego speech from the original speech by using the $A/B/X$ testing method when secret information was embedded in inactive frames. The results also indicate that the subjective quality of the proposed algorithm for steganography in inactive frames was close to that of the original speech.

We also adopted the ITU P.862 recommendation to measure the subjective quality of the stego speech. The recommendation describes an objective method for predicting the subjective quality of narrowband speech codecs. It uses the perceptual

TABLE IX
TESTING RESULTS WITH ITU P.862

Group	Speech file name	MOSLQO value	Average MOSLQO
	MC1	4.250	
	MC2	4.384	
Group 1	MC3	4.371	4.345
	MC4	4.438	
	MC5	4.305	
	WC1	4.258	
	WC2	4.417	
Group 2	WC3	4.278	4.315
	WC4	4.387	
	WC5	4.236	
	ME1	4.215	
	ME2	4.400	
Group 3	ME3	4.401	4.379
	ME4	4.431	
	ME5	4.450	
	WE1	4.413	
	WE2	4.413	
Group 4	WE3	4.264	4.383
	WE4	4.398	
	WE5	4.428	

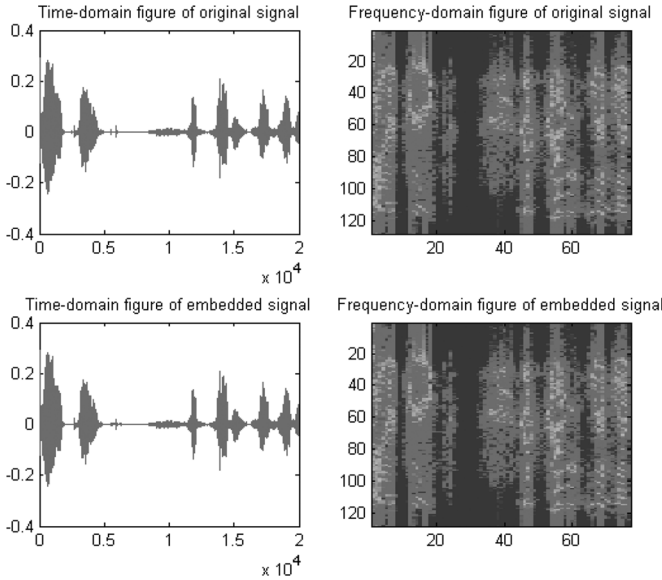

Fig. 4. Spectrum comparisons in the time- and frequency-domain.

evaluation speech quality (PESQ) value to assess the subjective quality of the stego speech. As the PESQ is not well matched with mean opinion score (MOS), PESQ-listening quality objective (LQO) is recommended to evaluate the quality of the stego speech. The PESQ is then mapped to the MOSLQO value. The testing results with the ITU P.862 method are listed in Table IX.

According to the ITU P.862 standard, the MOSLQO value of the original speech is equal to 4.5. As shown in Table IX, the average MOSLQO value of the stego speech was estimated to be 4.375. So the difference in MOSLQO between the original speech and the stego speech was so minor (3.18%) that distortion resulted from steganography in inactive frames was imperceptible.

Objective Quality: To evaluate further the imperceptibility of the stego speech, we compared the spectrum between the original speech and the stego speech in the frequency and time domain. For instance, the spectrums of the MC1 speech file having 183 inactive frames with and without hidden information are shown in Fig. 4.

Careful analysis of Fig. 4 shows very little distortion occurred in the time domain as a result of data embedding in inactive frames; however, we could not perceive any differences between the original speech and the stego speech in the frequency domain. This suggests steganography in inactive frames at a data embedding rate of 101 bits/frame had no or very little impact on the quality of the original speech.

The mean cepstrum distortion (MCD) metric [11] was used to measure the objective quality of the stego speech. The MCD is defined as

$$\mathrm{MCD} = \frac{1}{N_f}\sum_{k=1}^{N_f}\frac{10}{\ln 10}\sqrt{2\sum_{i=1}^{p}\left(c(i)-\widetilde{c}(i)\right)^2} \qquad (36)$$

where N_f is the number of audio frames, and $c(i)$ and $\widetilde{c}(i)$ are the cepstrum coefficients of the original speech and the stego speech, respectively, and p is the order of $c(i)$.

TABLE X
MCD VALUES OF STEGO SPEECH FILES

Group	Speech file name	Average MCD	MCD Mean	MCD Variance
	MC1	1.769		
	MC2	0.980		
Group 1	MC3	1.157	1.353	0.1124
	MC4	1.222		
	MC5	1.639		
	WC1	1.391		
	WC2	1.209		
Group 2	WC3	1.164	1.268	0.0208
	WC4	1.450		
	WC5	1.124		
	ME1	0.796		
	ME2	0.606		
Group 3	ME3	0.620	0.613	0.0127
	ME4	0.522		
	ME5	0.519		
	WE1	0.550		
	WE2	0.579		
Group 4	WE3	0.655	0.644	0.0061
	WE4	0.731		
	WE5	0.705		

Table X lists the MCD results of the stego speech files with information embedded in the inactive frames; each average MCD value in the third column is the arithmetic mean of MCDs obtained from six repeated experiments on an original speech file. As Table X shows, all the MCD values and variances of the stego speech files were relatively small, indicating that the proposed steganography algorithm for embedding information in the inactive frames achieved perfect imperceptibility.

B. Data Embedding Capacity

Using (34), we computed the data embedding capacity for each inactive frame. The length of the inactive frame encoded by G.723.1 codec at 6.3 kb/s was 192 bits; among them, 101 bits were used to embed information. The following paragraphs describe how to determine the average data embedding capacity

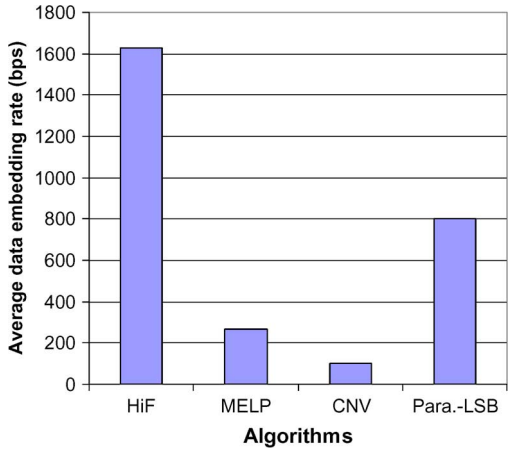

Fig. 5. Comparisons in data embedding rates between the proposed algorithm "HiF" and other algorithms.

for steganography in the active frames and inactive frames of low bit rate audio streams, respectively.

Suppose the frame number of the original speech is L, and the number of inactive frames is D, then the number of active frames is $L - D$. And, N bits of the inactive frame, such as $N = 101$, are used to embed information. Meanwhile, M bits of the active frame are used to embed information. The average data embedding capacity of the speech file can then be defined as the data embedding rate v in bits per second (b/s), given by

$$v = (D \times N + (L - D) \times M) / (L \times 192). \qquad (37)$$

Several other algorithms, such as CNV [5], MELP [8], [13], and parameter-LSB [9], [14]–[16], were previously suggested for embedding information in low bit rate audio streams encoded by ITU-T G.723.1. However, these algorithms are suitable for steganography in active frames only, achieving different levels of data embedding. For comparison purposes, these previously suggested algorithms and our proposed steganography algorithm were adopted to embed data in the speech sample files listed in Table VI, respectively. Fig. 5 shows the comparisons in data embedding capacity between our proposed algorithm (denoted as "HiF") and the other algorithms.

As Fig. 5 shows, the data embedding rate of our proposed algorithm "HiF" was much higher than those of the other algorithms. This is because the proposed steganography algorithm made good use of the redundancy in the inactive frames of low bit rate audio streams.

It is worth mentioning that the data embedding capacity of steganography in inactive frames is limited by the number of inactive frames of the original speech file. Research found 30%–50% of a VoIP session were inactive frames, so steganography in the inactive frames could attain a higher data embedding rate than other algorithms, which is in agreement with our experiment results.

VI. Conclusion

In this paper, we have suggested a high-capacity steganography algorithm for embedding data in the inactive frames of low bit rate audio streams encoded by G.723.1 source codec. The experimental results have shown that our proposed steganography algorithm can achieve a larger data embedding capacity with imperceptible distortion of the original speech, compared with other three algorithms. We have also demonstrated that the proposed steganography algorithm is more suitable for embedding data in inactive audio frames than in active audio frames. However, before the proposed algorithm comes into practical use in covert VoIP communications, it is necessary to explore how to assure the integrity of hidden messages in the case of packet loss, which shall be the subject of future work.

References

[1] Z. Wu and W. Yang, "G.711-based adaptive speech information hiding approach," *Lecture Notes Comput. Sci.*, vol. 4113, pp. 1139–1144, 2006.

[2] N. Aoki, "A technique of lossless steganography for G.711 telephony speech," in *Proc. 2008 4th Int. Conf. Intelligent Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Harbin, Aug. 2008, pp. 608–611.

[3] L. Ma, Z. Wu, and W. Yang, "Approach to hide secret speech information in G.721 scheme," *Lecture Notes Comput. Sci.*, vol. 4681, pp. 1315–1324, 2007.

[4] P. Chang and H. Yu, "Dither-like data hiding in multistage vector quantization of MELP and G.729 speech coding," in *Proc. Conf. Rec. 36th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2002, vol. 2, pp. 1199–1203.

[5] B. Xiao, Y. F. Huang, and S. Tang, "An approach to information hiding in low bit rate speech stream," in *Proc. IEEE GLOBECOM 2008*, Dec. 2008, pp. 371–375, IEEE Press.

[6] Z. Wu, W. Yang, and Y. Yang, "ABS-based speech information hiding approach," *Electron. Lett.*, vol. 39, no. 22, pp. 1617–1619, Oct. 2003.

[7] Z. Wu, W. Gao, and W. Yang, "LPC parameters substitution for speech information hiding," *J. China Univ. Posts Telecommun.*, vol. 16, no. 6, pp. 103–112, 2009.

[8] C. Krätzer, J. Dittmann, T. Vogel, and R. Hillert, "Design and evaluation of steganography for voice-over-ip," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 2397–3234.

[9] C. Bao, Y. F. Huang, and C. Zhu, "Steganalysis of compressed speech," in *Proc. IMACS Multiconf. Computational Eng. Syst. Applicat. (CESA)*, Oct. 2006, pp. 5–10.

[10] *Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*, ITU-T Recommendation G.723.1 Annex A, 2009 [Online]. Available: http://www.itu.int/net/itu-t/sigdb/speaudio/AudioForm-s.aspx?val=1117231

[11] N. Kitawaki, H. Nagabuchi, and K. Itoh, "Objective quality evaluation for low-bit-rate speech coding systems," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 2, pp. 242–248, Feb. 1988.

[12] Z. M. Lu, B. Yan, and S. H. Sun, "Watermarking combined with CELP speech coding for authentication," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 2, pp. 330–334, 2005.

[13] 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 *Security, Steganography, and Watermarking of Multimedia Contents VII*. San Jose, CA: Electronic Imaging Science and Technology, 2005, pp. 607–618.

[14] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-lsb data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[15] H. Tian, K. Zhou, Y. F. Huang, D. Feng, and J. Liu, "A covert communication model based on least significant bits steganography in voice over IP," in *Proc. 9th Int. Conf. For Young Comput. Scientists*, Nov. 2008, pp. 647–652.

[16] L. Y. Bai, Y. F. Huang, G. Hou, and B. Xiao, "Covert channels based on jitter field of the RTCP header," in *Proc. IEEE Int. Conf. Intelligent Inf. Hiding Multimedia Signal Process.*, 2008, pp. 1388–1391.

Yong Feng Huang received the Ph.D. degree in computer science and engineering from Huazhong University of Science and Technology in 2000.

He is an Associate Professor in the Department of Electronic Engineering, Tsinghua University, Beijing. His research interests include VoIP, P2P IP TV, multimedia network security, and next-generation Internet. He has published five books and over 70 research papers on computer network and multimedia communications. As one of the principal researchers, he has designed and constructed the China Education and Research Network (CERNET), which is the second largest computer network in China.

Dr. Huang is the principal/joint grant holder of ten externally funded research projects.

Shanyu Tang (A'08–M'08–SM'10) received the Ph.D. degree from Imperial College London in 1995.

He is a senior lecturer in informatics and multimedia in the Faculty of Computing at London Metropolitan University (U.K.). He is dedicated to adventurous research in fractal computing methods for covert communications, network security, data mining, and bio-informatics.

Dr. Tang is the principal grant holder of three externally funded research projects. He has contributed to 53 scientific publications—33 refereed journal papers including IEEE TRANSACTIONS and IEE/IET journal papers.

Jian Yuan is working toward the Ph.D. degree at the Department of Electronic Engineering, Tsinghua University, Beijing.

His research interests mainly focus on network security and complex networks.