Noname manuscript No.
(will be inserted by the editor)

High Capacity Speech Steganography for the G723.1 Coder Based on Quantised Line Spectral Pairs Interpolation and CNN Auto-Encoding

Hamza Kheddar · David Megías

Received: 27 May 2021 / Accepted: 18 Oct 2021

Abstract In this paper, a novel steganographic method for Voice over IP applications —called Steganography-based Interpolation and Auto-Encoding (SIAE)— is proposed. The aim of the proposed scheme is to securely transmit a secret speech hidden within another (cover) speech coded with a G723.1 coder. SIAE embeds the steganograms in four interpolated and quantised line spectral pairs (QLSP) vectors. In order to minimize the changes in the cover speech, the proposed approach uses a 1D auto-encoder to compress the payload, and this scheme only requires embedding eight bits in about 30% of the packets. At the receiver side, the secret data can be successfully expanded to its original size upon decoding. This represents a significant reduction in the number of modified bits compared to state-of-the-art schemes, and results in enhanced undetectability and decreased steganographic quality loss. The results show that the proposed auto-encoder scheme has a very high performance since it can compress the embedded data up to 80 times from its original size, leading to a steganographic capacity that exceeds one kilobit per second (kpbs). In terms of imperceptibility, which is a relevant property for speech-in-speech steganography, the proposed SIAE method entails a very imperceptible distortion, with an average steganographic quality loss not greater than 0.19 in terms of mean opinion scores (MOS). Last but not least, the proposed method evades steganalysis specifically targeted at speech steganography. The tested steganalytic methods fail in detecting the steganographic content produced with the proposed SIAE method, yielding classification results that are indistinguishable from random guessing.

Hamza Kheddar
LSEA Laboratory, Faculty of technology, Department of Electrical Engineering, University of Medea, Medea, 26000, Algeria.
Tel.: +213-550226503
E-mail: kheddar.hamza@univ-medea.dz

David Megias
Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), CYBERCAT-Center for Cybersecurity Research of Catalaonia, Av. Carl Friedrich Gauss, 5, 08860 Castelldefels, Barcelona, Spain.
E-mail: dmegias@uoc.edu

Keywords Auto-encoder · Convolutional Neural Networks · Multi-pulse Maximum Likelihood Quantisation · G.723.1 · Speech Steganography · Interpolation.

1 Introduction

Information hiding, watermarking, and steganography are three closely related fields that overlap and share numerous technical approaches. Nevertheless, there are major relevant distinctions that influence the requirements and, consequently, the technical solution design in each case.

Information hiding (also called data hiding) is a common expression that encompasses a wide range of techniques, in addition to that of concealing secret information in the carrier. Herein, the word "hiding" refers to either preserving the presence of the communication secret, as in steganography, or making the information imperceptible, as in watermarking.

Steganography is a technical term derived from the Greek words *steganos* and *graphia* that mean, respectively, *covered* and *writing*. Steganography is the technique of concealing communication. The very presence of communication is confidential.

Systems for embedding messages can be classified into *non-watermarking* frameworks, in which the message is not associated to the carrier, and *watermarking* frameworks, in which the message is associated to the carrier (cover). They can also be individually broken down into *steganographic* frameworks, in which the presence of the communication is kept confidential, and *non-steganographic* frameworks, in which the presence of the communication is not confidential [4].

By differentiating between embedded data that depends on the cover and embedded data that does not depend on the cover, we can detail various applications and requirements of the data hiding methods. The existing mechanisms may be very similar or, in some cases, identical. However, with the arrival of the Internet and TCP/IP protocols, new methods of steganography are available; hence, the above categories have been extended with different sub-categories.

In this article, we introduce a Steganography-based Interpolation and Auto-Encoding (SIAE) method that mainly focuses on embedding a secret speech within cover speech coded with a G723.1 coder [23], which is widely used in voice over IP (VoIP) communication. The speech analysis of the coder is based on dividing the frame of 240 ms into four sub-frames. The motivation behind our approach is the fact that the Quantised Line Spectral Pairs (QLSP) are calculated only in the fourth sub-frame, while the QLSP coefficients of the remaining three sub-frames are estimated, by linear interpolation, to embed the secret speech. The QLSP is then transformed into codes (using codebook indexes) and transmitted to its destination. The last step is deleting the hidden data at the point of origin. The authors of [23] carried out the challenge by superimposing one element on each QLSP array with one hidden speech sample that has the minimum euclidean distance (D) to calculate the error array (E) and extract the position (index). Then, the obtained E is approximated with Lagrange interpolation. The obtained polynomial coefficients are transmitted in the least significant bit (LSB) of the pulse position index (PPI). The transmission of E needs a large amount of LSBs of the target signal to be successfully received, which affects the imperceptibility and rises the suspicions of potential eavesdroppers who may be listening to the channel. For that reason, we have employed a 1D auto-encoder to compress E and transmit, at most, eight bits per packet and expand them to their original size at the decoding stage.

1.1 Contributions and Plan of the Paper

As detailed in Section 2, which is devoted to overview VoIP stegnaographic schemes, most of the existing steganographic schemes for low-rate speech coders, such as G723.1, adaptive multi-rate (AMR) or algebraic-code-excited linear-prediction (ACELP), roughly leverage on the same ideas, that is, either pitch delay or codebook partitioning. Moreover, most of the existing methods do not extract the steganogram from QLSP at the reception due to the erasure carried out during conversion, predictive split vector quantiser (PSVQ) to indexes, and vice versa. This latter operation severely limits their steganographic applicability in real time, due to the fact that it significantly reduces the capacity and the security of the scheme.

With the aim of overcoming these issues, to the best of our knowledge, this paper offers the following novel contributions:

- No work, in the state-of-the-art methods, embeds either the steganogram nor the stego keys in G723.1 pulse positions indexes (PPI) codes.
- For the first time, the steganograms are extracted from QLSP without partitioning the codebook at the receiver, skipping the step of converting the QLSP into codes to preserve the secret message.
- We ensure that the legitimate and illegitimate steganographic end users receive a G723.1 bitstream with negligible changes for most QLSPs coefficients compared to the original ones.
- We integrate a convolutional neural network (CNN) deep auto-encoder into a G731.1 voice coder to compress the steganogram into a small amount of bits, so-called feature map, to avoid affecting the speech signal. The applied CNN is a type of unsupervised deep learning, which means that neither legitimate or illegitimate steganographic end users have the secret or cover speech database. This latter concept significantly increases the security of the proposed SIAE scheme.

The remainder of this paper is organised as follows. As remarked above, Section 2 overviews the state of the art of VoIP steganography. In Section 3, we recall the G723.1 coding concepts and the linear interpolation and characteristics of PPI, which are necessary for understanding the subsequent sections. Section 4 defines the proposed SIAE method. Section 5 provides a thorough experimental validation and comparison with existing methods. Finally, Section 6 concludes the paper.

2 Related Work

To date, in VoIP, a remarkable number of steganographic algorithms have been developed, most of which were surveyed by Mazurczyk in [30]. In general, they can be split into two classes: protocol steganography approaches and payload steganography approaches.

Protocol steganography approaches, commonly known as ***covert channels***, make use of the particular protocols of VoIP as carriers. Generally speaking, this type of covert channels involves two major techniques. The first one encodes the secret messages by modulating the inter-packet delays, which is equivalent to altering packet rates. The second one uses the fact that only a few packets' headers are modified during transmission and embed the secret messages into optional or unused fields of the protocol headers. In the VoIP scenario, the applications of the techniques that rely on packet re-ordering are introduced in [18,53]. Those techniques are based on changing the order of sent or received messages, or on methods that modify the inter-packet delay [2,8]. The second technique has been usually applied to TCP/IP [47,34], real-time transport protocol (RTP) [44], and session initiation protocol (SIP) with session description protocol (SDP) [33]. The latter category of covert channel techniques has been applied to new protocols such as [9], where the Stream Control Transmission Protocol (SCTP) replaced the TCP and UDP protocols, and in [3], where IPv6 replaced IPv4. Finally, a hybrid steganographic method, Lost Audio Packets Steganography (LACK)[31], is suggested for VoIP, which uses some artificially delayed packets to send secret messages in the payload.

Payload steganography approaches include the methods based on the modification of the original message's content carried in the payload field. In VoIP, they use the digital speech signal as the carrier. This class is divided into three sub-categories, namely, hiding in the *temporal domain*, hiding in *frequency/wavelet domains* and hiding in the *coded domain*. Fig. 1 illustrates the classification of steganography techniques used for VoIP transmission.

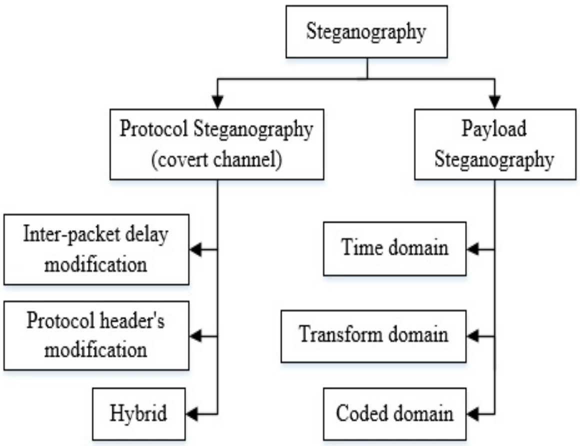

Steganography
Protocol Steganography (covert channel)
Payload Steganography
Inter-packet delay modification
Protocol header's modification
Hybrid
Time domain
Transform domain
Coded domain

Fig. 1 Categories of VoIP steganography schemes.

In the temporal domain, among the speech steganography schemes, replacing the LSBs with the binary bits of the secret messages is the most popularly employed method, which results in a high embedding capacity and weak security. Recently, improved versions of LSB techniques have been proposed to enhance the security level and the imperceptibility of such steganographic schemes. For example, LSB combining steganography with VoIP-based covert communication was introduced by Huang et al. [20]; an adaptive LSB approach with secret speech scrambling to reduce the embedding distortion and increase security was proposed by Ballesteros and Renza [1]; Miao et al. [36] presented an adaptive steganography strategy to enhance the security by selecting higher embedding bit rates in the sharp blocks and lower embedding bit rates in the flat blocks; Liu et al. [28] adopted least-significant-digits (LSDs), instead of LSBs, to secretly embed messages that can enhance the steganographic capacity to approximately 30% and lead to less embedding distortion.

Several techniques in the transformed domain have been proposed in the literature. To attain imperceptibility, those techniques take advantage of the frequency masking effect of the human auditory system (HAS) explicitly by altering only masked regions [7], or implicitly [14] by applying a small change to the samples of the audio signal. The spread spectrum (SS) technique [5] is an approach that spreads the concealed information over the frequency spectrum. The discrete wavelet transform (DWT) is another technique that hides in the transform domain. DWT-based speech steganography is introduced in [6]. The discrete spring transform (DST) is used in [39] within a scheme called DST-LACK, which is a speech-in-speech steganography that reduces the effect of packet loss caused by the LACK covert channel scheme [31] to make the latter method less detectable.

In addition to the previous sub-categories, steganography in coded domains is considered in data hiding for real time communications. Speech encoders such as the AMR, ACELP, Speex and the mixed-excitation linear prediction (MELP), at their respective encoding rates, are employed. We have identified two main approaches: post-encoder and in-encoder. For example, the steganographic scheme based in-coder, detailed in [29], embeds the confidential speech into the fractional pitch delay parameters while maintaining the integer pitch delay parameters unaltered. Efficient steganographic strategies for speech post-coding are: echo steganography [12] and embedding in the variation caused after a low coding bit-rate of a high coded bit-rate cover, which is also called transcoding [32]. Besides that, post-encoding steganography can be used to protect steganographic systems from man-in-the-middle attacks, like in [38], where the cover objects are coded using the G.711 codec.

Different works in the coded domain sub-category have been proposed recently. Kheddar et al. [27] employed a random LSB embedding in specific parameters of the mixed-excitation linear prediction (MELP) coder bitstream to increase security and preserve imperceptibility. Another example, also belonging to the same sub-category, is the use quantization-index-modulation (QIM) steganography as described in [52,48]. Steganography can also be applied in inactive speech frames [37,21] for confidential information exchange in IP telephony.

Table 1: Summary of the related work.

Scheme	Characteristics	Comments
[25]	Is based on the use of *interpolation* to approximate the parameter F0, which is in charge of embedding secret information about the pitch of the speech signal	Scheme applied on the Speex speech coder. It causes a significant speech distortion.
[15]	Uses the *Lagrange interpolation* to alter the threshold (k, n) secret sharing method. This is applied to the LACK steganographic algorithm [31] to increase the unreliability and undetectability level.	The interpolation is used to enhance the security of an existing scheme. It applies encryption on the payload.
[19]	The suggested method performs information hiding while *the prediction of pitch period* is performed, during low bitrate (G.723.1) speech encoding, hence preserving synchronisation between data embedding and the process of speech encoding.	Achieves a high quality of speech and prevents detection using steganalysis.
[42]	The proposed approach implements an Adaptive Multi-Rate (AMR) Fixed CodeBook (FCB) Adaptive steganography scheme (AFA). The introduction of a cost function and the additive distortion function, merged with Syndrome-Trellis codes (STC), enhances the hiding and statistical undetectability.	Better concealment schemes and statistical security than the existing AMR FCB steganographic schemes.
[16]	Is based on the algorithm of *diameter-neighbour codebook partition* that provides the AMR-WB coder with flexibility in steganographic capacity with various iterative parameters.	Security against statistical steganalysis is better than that obtained with schemes based on neighbour-index-division (NID) codebook partition.
[13]	Scheme for AMR low bit-rate speech stream secure against pitch delay steganalysis that benefits from effective STCs and *suboptimal pitch delay searches.*	It is an enhanced version of steganographic pitch delay-based schemes.
[45]	Concentrating on steganography in internet low bit-rate codec (iLBC) speech streams. It is based on *gain quantisation*, which is enclosed in the vector quantisation of the dynamic lookup table (codebook). The hidden data is leaked by quantising the gain value while changing the range of the gain codebook.	High embedding capacity, low signal distortion, and provides high resistance against steganalysis.

[29]	It is a *pitch delay-based steganographic* algorithm, It embeds the hidden data into the fractional pitch delay parameters. The scheme provides excellent performance and preserves the integrity of integer pitch delay parameters.	Method applied on ACELP speech coder. Provides larger embedding capacity. Resists the state-of-the-art steganalytic methods.
[43]	The scheme named pulse distribution model AMR FCB steganography (PDM-AFS) is based on the pulse distribution model, which is obtained from the distribution characteristics of the FCB value in the cover audio.	It is implemented on the AMR coder and provides high capacity. In addition, it is very resistant against steganalysis.
[35]	The proposed scheme is based on changing the position of the last FCB parameter in each track, and the embedded message is computed based on the relationship between the parameters in the same track.	It is implemented on G.711 coder. Provides a very high stenographic capacity. Resists to RS steganalysis algorithm. The quality may decrease by 0.5 MOS.
[11]	In this scheme, the second FCB parameter in each track is modified, and the embedded message is hidden in the first and second FCB parameters in the track.	Scheme applied on ACELP speech codec. It provides high capacity, the speech quality is slightly affected.
[41]	The scheme pitch delay on unvoiced speech (PDU-AAS) is based on changing the pitch delay of the embedded positions, using the distribution characteristic of the adjacent pitch delay.	The method is realised on AMR coder. Resists to CEC and calibrated matrix of the second-order difference of the pitch delay (C-MSDPD) steganalysis methods.

Table 1 summarises the most recent related work of steganography using interpolation and coded domains, considering speech encoders in the signals context and information hiding for VoIP communications.

3 Background

Several techniques that constitute the basis of the proposed method are introduced in the next few sections.

3.1 Basics on G723.1 Low-Rate Coding

The G723.1 low bit rate speech coder can compress data at 5.3 and 6.3 kbps coding rates. It makes it possible to change the encoding rate at any interval of 30 ms. The multi-pulse maximum likelihood quantisation (MP-MLQ) and ACELP are the forms of the excitation for the higher and lower rates, respectively. The coder depends on the concept of straight linear prediction (LP) analysis-by-synthesis coding and is designed to limit a perceptually weighted error signal. The coder operates at an 8 kHz sampling rate with a frame size of 240 samples (30 ms).

3.2 Linear Interpolation

In G723.1, the analysis frame designates the speech frame (of 30 ms), which is weighted by a Hamming window, and from which the QLSP coefficients are calculated. Note that the center of the analysis frame is aligned with the center of the first sub-frame ($\hat{f}^{(0)}$) of the frame to be encoded, which implies that the QLSP coefficients calculated from this analysis frame are those of the first sub-frame. The coefficients of the other sub-frames ($\hat{f}^{(1)}, \hat{f}^{(2)}$ and $\hat{f}^{(3)}$) are obtained by linear interpolation between the QLSP coefficients calculated from the current and the next analysis frame [22]. Fig. 2 shows the first two frames to be coded.

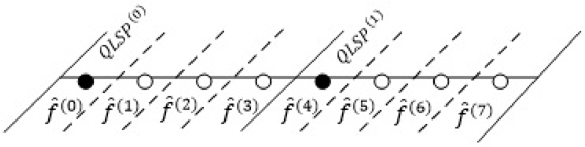


Fig. 2 Examples of interpolated QLSP vectors corresponding to eight successive sub-frames.

In Fig. 2, the black spots represent the original QLSP coefficients and the white spots represent the interpolated QLSP coefficients. Each frame is divided into four sub-frames and each sub-frame is represented by its own set of QLSP coefficients calculated according to Equation (1):

$$\hat{f}^{(j)} = \begin{cases} 0.75\hat{f}^{(0)} + 0.25\hat{f}^{(4)}, & \text{sub-frame 0.} \\ 0.50\hat{f}^{(0)} + 0.50\hat{f}^{(4)}, & \text{sub-frame 1.} \\ 0.25\hat{f}^{(0)} + 0.75\hat{f}^{(4)}, & \text{sub-frame 2.} \\ \hat{f}^{(4)}, & \text{sub-frame 3.} \end{cases} \tag{1}$$

3.3 Characteristics of PPI

The multi-pulse positions index (PPI) for each sub-frame is represented with a combinatorial code. For sub-frames one and three, there are 6 pulses in 30 positions. For sub-frames two and four, there are 5 pulses in 30 positions. The total number of combinations of positions for the individual sub-frames is then
$\binom{30}{6}$ or $\binom{30}{5}$, where $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ is a binomial coefficient. Hence

$$\mathrm{PPI}(1) = \mathrm{PPI}(3) = \binom{30}{6} = 593,775, \tag{2}$$

$$\mathrm{PPI}(2) = \mathrm{PPI}(4) = \binom{30}{5} = 142,506. \tag{3}$$

Coding the PPI for each sub-frame separately would require $20+18+20+18=76$ bits. The number of combinations for all $i=0,\ldots,3$ sub-frames is, however, $2^{72} < \prod_{i=0}^{3} \mathrm{PPI}(i) < 2^{73}$. This can be coded with 73 bits, representing a saving of three bits with respect to separate coding.

The strategy used in G.723.1 is to express the code values for each sub-frame in modulo notation, as follows:

$$c(k) = p(k)M(k) + q(k), \tag{4}$$

for

$$q(k) = c(k) \bmod M(k), \tag{5}$$

and

$$p(k) = \lfloor c(k)/M(k) \rfloor, \tag{6}$$

where $\lfloor . \rfloor$ denotes the nearest integer towards negative infinity. For $k=1$ and $k=3$, $M(k)=2^{16}$, therefore, $q(k)$ can be coded with 16 bits, and $p(k)$ takes values in the interval $[0,8]$. For $k=2$ and $k=4$, $M(k)=2^{14}$ is used, therefore, $q(k)$ can be coded with 14 bits, and $p(k)$ takes values in the interval $[0,5]$.

The total number of the combinations of the 4 $p(k)$ is $9\cdot 6\cdot 9\cdot 6 = 5,184$, which can be coded with 13 bits. This approach achieves a total of 73 bits. The procedure in G.723.1 allocates 90 values (more than the minimum 72) to code $p(1)$ and $p(2)$, and the remainder of the 13 bits to code $p(3)$ and $p(4)$ (which can represent up to 91 values). The coding procedure is as follows [23]:

1) Obtain $q(k)$ for each sub-frame.
2) Obtain $p(k)$ for each sub-frame.
3) The first code word (13 bits) is calculated from a combined value as follows: $p(4)+9p(3)+90p(2)+810p(1)$.
4) The next four code words are $q(k)$, which requires 16, 14, 16, and 14 bits, respectively.

4 The Proposed Scheme

The proposed scheme for covert communication of a speech signal, using the SIAE hiding module to conceal speech content (confidential) within another speech content (non-confidential), is illustrated in Fig. 3. Similarly, a recovery module for the extraction of the secret speech is shown in Fig. 5. Both modules are described below. The meaning of the terms used in the SIAE steganographic scheme are provided in Table 2.

4.1 Embedding Process

The embedding module (Fig. 3) carries out the following three fundamental operations:

Table 2 Notation

Term	Meaning
$x_{\text{ind}(j),j}$	Secret speech samples to be transmitted secretly (circled values in Fig.4)
$\boldsymbol{G_i}$	Group of four samples of secret speech
$f_k^{(j)}$	LSPs of a sub-frame (not quantized)
$\hat{f}_k^{(j)}$	QLSPs of a quantized sub-frame
$\hat{f}_{\text{ind}(j)}^{(j)}$ or $\hat{f}_{k_0}^{(j)}$	Original QLSP that have minimum distance to a group $\boldsymbol{G_i}$ of secret speech samples (black bold values in Fig. 4)
$\boldsymbol{D_j}$	Minimum distance between each element of $\boldsymbol{G_i}$ with each $\hat{f}_k^{(j)}$
$\boldsymbol{E_j}$	Error (array of size 1×4) between $\hat{f}_{\text{ind}(j)}^{(j)}$ and $x_{\text{ind}(j),j}$
$\boldsymbol{P_n}(x)$	Polynomial of the $\boldsymbol{n}$-the degree generated with the Lagrange algorithm

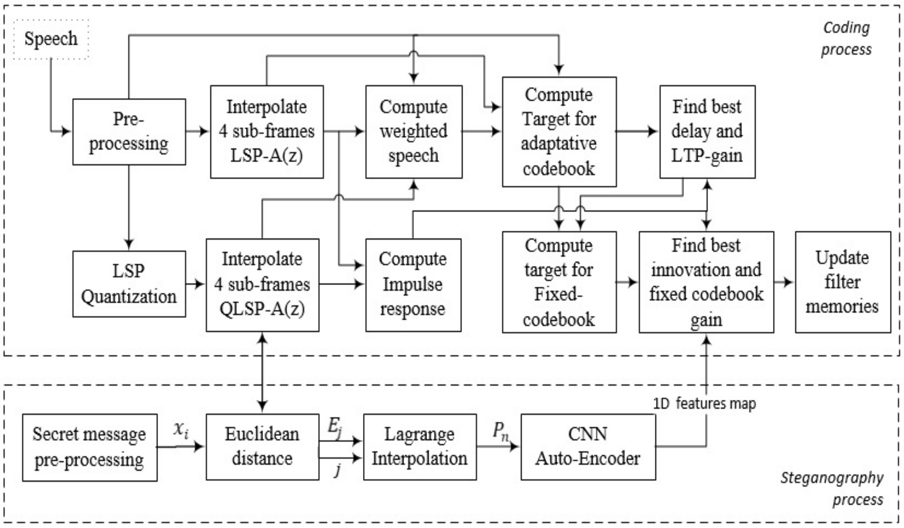

Speech
Coding process
Pre-processing
Interpolate 4 sub-frames LSP-A(z)
Compute weighted speech
Compute Target for adaptative codebook
Find best delay and LTP-gain
LSP Quantization
Interpolate 4 sub-frames QLSP-A(z)
Compute Impulse response
Compute target for Fixed-codebook
Find best innovation and fixed codebook gain
Update filter memories
1D features map
Secret message pre-processing
x_i
Euclidean distance
E_j
j
Lagrange Interpolation
P_n
CNN Auto-Encoder
Steganography process

Fig. 3 Steganography in G723.1 coding process.

4.1.1 Secret Speech Pre-processing and Lagrange Interpolation

The secret speech is re-sampled to 8 kHz and then split into groups $\boldsymbol{G_i}$ of four samples each;

$$G_i = x_{i,j}, j = 0, \ldots, 3. \tag{7}$$

The linear interpolation of the QLSP results in four vectors, $\hat{f}^{(0)}, \hat{f}^{(1)}$, $\hat{f}^{(2)}$ and, $\hat{f}^{(3)}$, with ten elements each, which are converted to line spectral frequencies to predict filter coefficients using the lsf2poly function to output 11 elements ($k =$

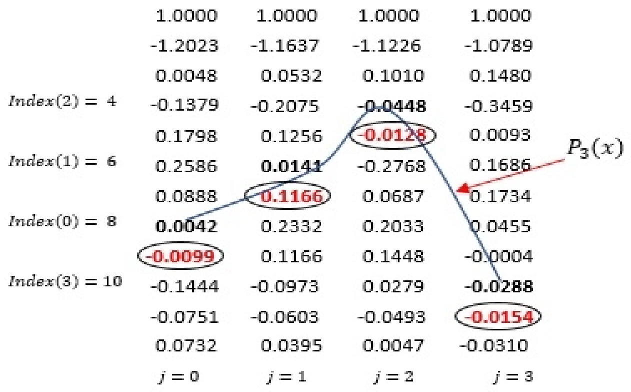
$1, \ldots, 11$). The role of *minimum distance function* is to find the minimum distance between each element of G_i with each $\hat{f}_k^{(j)}$, one by one.

$$D_j = \arg\min_{\hat{f}_k^{(j)}} \left(G_i - \hat{f}_k^{(j)} \right)^2, j = 0, \ldots, 3. \tag{8}$$

We extract the *index* of the positions of our group of secret samples (G_i) from the QLSP matrix as follows:

$$\text{ind}(j) = k_0 : D_j = \hat{f}_{k_0}^{(j)}. \tag{9}$$

Fig. 4 recapitulates the embedding process. The proposed method neither replaces the $\hat{f}_{\text{ind}(j)}^{(j)}$ with $x_{\text{ind}(j),j}$ samples when the minimal distance is reached, nor adds extra values to the original $\hat{f}_k^{(j)}$. It only superimposes them to extract their positions, (indexes of $\hat{f}_k^{(j)}$).



	1.0000	1.0000	1.0000	1.0000
	-1.2023	-1.1637	-1.1226	-1.0789
	0.0048	0.0532	0.1010	0.1480
$Index(2) = 4$	-0.1379	-0.2075	**-0.0448**	-0.3459
	0.1798	0.1256	-0.0128	0.0093
$Index(1) = 6$	0.2586	**0.0141**	-0.2768	0.1686
	0.0888	0.1166	0.0687	0.1734
$Index(0) = 8$	**0.0042**	0.2332	0.2033	0.0455
	-0.0099	0.1166	0.1448	-0.0004
$Index(3) = 10$	-0.1444	-0.0973	0.0279	**-0.0288**
	-0.0751	-0.0603	-0.0493	-0.0154
	0.0732	0.0395	0.0047	-0.0310
	$j = 0$	$j = 1$	$j = 2$	$j = 3$

Fig. 4 Embedding process and Lagrange interpolation.

Then, we calculate the error E, between the original $\hat{f}_{\text{ind}(j)}^{(j)}$ and the homologous $x_{\text{ind}(j),j}$, as follows:

$$E_j = x_{\text{ind}(j),j} - \hat{f}_{\text{ind}(j)}^{(j)}, j = 0, \ldots, 3. \tag{10}$$

The obtained E_j are polynomial interpolated, using $\hat{f}_{\text{ind}(j)}^{(j)}$ as inputs and E_j as output ($P : x = \hat{f}_{\text{ind}(j)}^{(j)} \rightarrow E_j$), to obtain the polynomial $P_n(x)$. There is one and only one polynomial P_n of degree less than or equal to n verifying:

$$P_n(x) = E_j, \quad \forall i = 0, \ldots, n. \tag{11}$$

The Lagrange polynomial is written as follows:

$$P_n(x) = \sum_{j=0}^{n} E_j L_j(x), \tag{12}$$

where,

$$L_j(x) = \prod_{j=0, j \neq l}^{n} \frac{x - \hat{f}^{(l)}_{\mathrm{ind}(l)}}{\hat{f}^{(j)}_{\mathrm{ind}(j)} - \hat{f}^{(l)}_{\mathrm{ind}(l)}}.$$

Hence, for $n = 3$, Equation (12) becomes:

$$\begin{aligned} P_3(x) &= E_0.L_0(x) + E_1.L_1(x) + E_2.L_2(x) + E_3.L_3(x) \\ = E_0.&\frac{(x - \hat{f}^{(1)}_{\mathrm{ind}(1)})(x - \hat{f}^{(2)}_{\mathrm{ind}(2)})(x - \hat{f}^{(3)}_{\mathrm{ind}(3)})}{(\hat{f}^{(0)}_{\mathrm{ind}(0)} - \hat{f}^{(1)}_{\mathrm{ind}(1)})(\hat{f}^{(0)}_{\mathrm{ind}(0)} - \hat{f}^{(2)}_{\mathrm{ind}(2)})(\hat{f}^{(0)}_{\mathrm{ind}(0)} - \hat{f}^{(3)}_{\mathrm{ind}(3)})} + \\ E_1.&\frac{(x - \hat{f}^{(0)}_{\mathrm{ind}(0)})(x - \hat{f}^{(2)}_{\mathrm{ind}(2)})(x - \hat{f}^{(3)}_{\mathrm{ind}(3)})}{(\hat{f}^{(1)}_{\mathrm{ind}(1)} - \hat{f}^{(0)}_{\mathrm{ind}(0)})(\hat{f}^{(1)}_{\mathrm{ind}(1)} - \hat{f}^{(2)}_{\mathrm{ind}(2)})(\hat{f}^{(1)}_{\mathrm{ind}(1)} - \hat{f}^{(3)}_{\mathrm{ind}(3)})} + \\ E_2.&\frac{(x - \hat{f}^{(0)}_{\mathrm{ind}(0)})(x - \hat{f}^{(1)}_{\mathrm{ind}(1)})(x - \hat{f}^{(3)}_{\mathrm{ind}(3)})}{(\hat{f}^{(2)}_{\mathrm{ind}(2)} - \hat{f}^{(0)}_{\mathrm{ind}(0)})(\hat{f}^{(2)}_{\mathrm{ind}(2)} - \hat{f}^{(1)}_{\mathrm{ind}(1)})(\hat{f}^{(2)}_{\mathrm{ind}(2)} - \hat{f}^{(3)}_{\mathrm{ind}(3)})} + \\ E_3.&\frac{(x - \hat{f}^{(0)}_{\mathrm{ind}(0)})(x - \hat{f}^{(1)}_{\mathrm{ind}(1)})(x - \hat{f}^{(2)}_{\mathrm{ind}(2)})}{(\hat{f}^{(3)}_{\mathrm{ind}(3)} - \hat{f}^{(0)}_{\mathrm{ind}(0)})(\hat{f}^{(3)}_{\mathrm{ind}(3)} - \hat{f}^{(1)}_{\mathrm{ind}(1)})(\hat{f}^{(3)}_{\mathrm{ind}(3)} - \hat{f}^{(2)}_{\mathrm{ind}(2)})} \end{aligned} \tag{13}$$

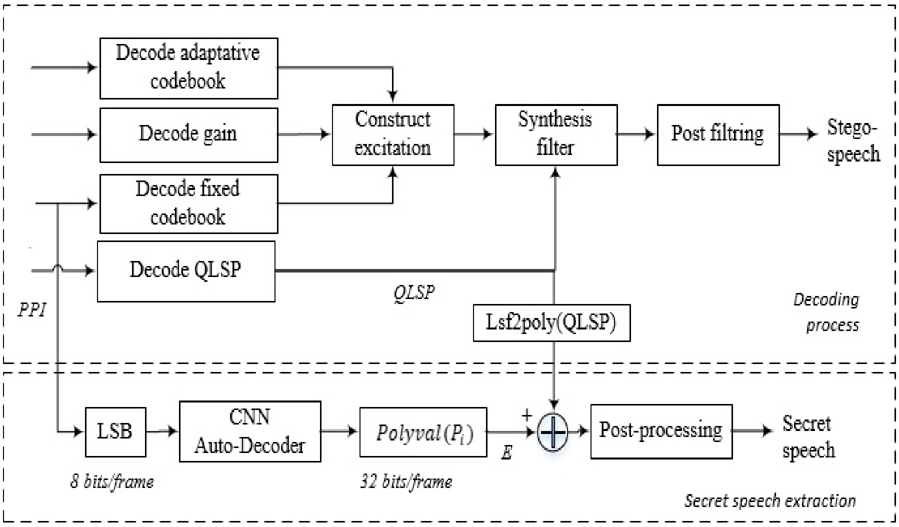


Fig. 5 Steganography in G723.1 decoding process.

4.1.2 CNN Auto-Encoder

The output data from the polynomial interpolation block are four parameters (polynomial degree equals three) for each of the four secret speech samples. These parameters will be transmitted in the bitstream, precisely, in the PPI that is coded with 73 bits, divided into five values of 13-bit words. Experimental tests show that applying LSB in all values of PPI would lead to either high radical changes in statistical proprieties of the whole speech signal or an inability to decode all the speech frames.

Mathematically, the hiding procedure can be depicted as a mapping $E : C \times M \rightarrow C$, where C is the set of prospective covers and M is the set of prospective messages. Clearly, it is necessary that the length of M be smaller than the length of C:

$$|C| \geq |M|. \tag{14}$$

For that reason, we compress these polynomial parameters using a 1D CNN auto-encoder (a similar idea is described in [26]) to reduce the size of all polynomial parameters. The final bottleneck layer of the CNN auto-encoder provides an abstract and compact representation of the $P_n(x)$ signal named **1D feature-map** (equivalent to **bitstream** in traditional speech coding). Note that AE is the Auto-Encoding process,

$$M' = \mathrm{AE}(M). \tag{15}$$

The obtained feature-map M' not only satisfies the condition of equation (14), but it makes relatively $|C| \gg |M'|$. In that way, we can embed the feature-map in only **one value** (the highest PPI value) among five PPI.

The details about the embedding procedures are summarized in Algorithm 1. An example of auto-encoder architecture that satisfies our conditions can be found in the Github platform[1], and the performance of the proposed AE is detailed in Section (5.3).

4.2 Extraction Process

The extraction module (Fig. 5) carries the following three fundamental operations:

4.2.1 Feature-map Extraction and CNN Auto-Decoding

By applying LSB extraction to the PPI parameter. This step needs to be performed just before the fixed codebook decoder, otherwise the feature-map and PPI would be converted together, via the fixed codebook, to parameters that contribute only to constructing excitation, which would not be useful for our scheme. In other words, a steganalyser would not possibly find the exact hidden data that is embedded in the PPI after the speech synthesis process (in the decoded speech).

The CNN auto-decoder expands the feature-map to its original format (polynomial coefficients) with negligible changes that are analysed in the next section. The design of the decoder structure can be very similar (symmetric) or dissimilar to the encoder architecture.

[1] https://github.com/usthbstar/autoEncoder

Algorithm 1: Secret speech embedding process

```
EMBEDDING(Secret.wav,Cover.wav)
$x_i \leftarrow$ read(Secret.wav)
$n \leftarrow |x_i|$                                  ▷ $|\cdot|$ is the length of a vector
$G_i \leftarrow$ reshape($x_i$, 4, $|x_i|/4$)            ▷ Arrange $x_i$ in four vectors of equal length
$c_i \leftarrow$ read(Cover.wav)
$nn \leftarrow |c_i|$
$QLSP \leftarrow$ AMR_coder($c_i$)
$\mathrm{P_n(x)} \leftarrow$ []
for $i \leftarrow 1$ to number_frames do
    $P \leftarrow 0$
    for $j \leftarrow 0$ to 3 do
        $\hat{f}_k^{(j)} \leftarrow$ Convert($QLSP[i, j]$)
                                                       ▷ Convert LSP to polynomial
        $D_j \leftarrow \arg\min_{\hat{f}_k^{(j)}} \left(G_i - \hat{f}_k^{(j)}\right)^2$
        ind($j$) $\leftarrow k_0 : D_j = \hat{f}_{k_0}^{(j)}$
        $E_j \leftarrow x_{\mathrm{ind}(j),j} - \hat{f}_{\mathrm{ind}(j)}^{(j)}$
        $P \leftarrow$ LagrangeInterpolation($\hat{f}_{ind(l)}^{(l)}, E_j$)
    end
    $P_n(x) \leftarrow P$                              ▷ Buffering Lagrange coefficients
end
FM $\leftarrow$ CNNAutoEncoder($P_n(x)$)                  ▷ Build 1D freatureMap (FM)
$k \leftarrow 1$                                       ▷ or $k > 1$ for ON-OFF mode
PPI $\leftarrow$ AMR_coder(Cover.wav)
for $i \leftarrow 1$ to number_frames increase by $k$ do
    $\mathrm{PPI_{max}} \leftarrow \max_{j=1,\ldots,5}(\mathrm{PPI}[j])$
    Position $\leftarrow k_0 : \mathrm{PPI}[k_0] = \mathrm{PPI_{max}}$
    $P \leftarrow \mathrm{LSB_{embed}}$(Frame[i].$\mathrm{PPI_{max}}$, FM)     ▷ Embed the 8 bits of FM in $\mathrm{PPI_{max}}$
    UPDATE_PPI($P$, Position)
end
return Stego_speech
```

4.2.2 Extraction of the Secret Message

The values obtained from the auto-encoder are first arranged in groups of four values and then evaluated using the polyval($P_n(x)$) function. The indexes vary from 1 to 11 (length of $\hat{f}^{(j)}$) during evaluation; however, we take only the samples whose indexes have interpolation errors $E_{\mathrm{interp}} = 0$. The variable x (in equation 16) verifies the latter condition if $x = \hat{f}_{\mathrm{ind}(j)}^{(j)}$.

$$E_{\mathrm{interp}} = |\mathrm{QLSP}_{\mathrm{int}}(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \left| \prod_{j=0}^{n} \left(x - \hat{f}_{\mathrm{ind}(j)}^{(j)}\right) \right|, \tag{16}$$

where, $M_{n+1} = \max_{x \in I} \left| P_n^{(n+1)}(x) \right|$, and $P_n^{(n+1)}(x)$ denotes here the $(n+1)$-th derivative of $P_n(x)$.

Notice that, for $n = 3$, Equation (16) becomes:

$$E_{\text{interp}} \quad \leq \quad \frac{M_4}{4!}\left|\left((x - \hat{f}^{(0)}_{\text{ind}(0)}\right)\left(x - \hat{f}^{(1)}_{\text{ind}(1)}\right)\left(x - \hat{f}^{(2)}_{\text{ind}(2)}\right)\left(x - \hat{f}^{(3)}_{\text{ind}(3)}\right)\right|. \quad (17)$$

The exact values of E_j previously interpolated during the embedding process will be retrieved correctly using equation 18.

$$E_j = \text{polyval}\left(P_3\left(\hat{f}^{(j)}_{\text{ind}(j)}\right)\right), j = 0, \ldots, 3. \quad (18)$$

The secret message S_j, which consists of four samples in each frame, is then the sum of both E_j and the $\hat{f}^{(j)}_{ind(j)}$ values output from the decoder. Formally: 1.06

$$S_j = E_j + \hat{f}^{(j)}_{ind(j)}, j = 0, \ldots, 3, \quad (19)$$

The post-processing consists of gathering all the secret samples in a 1D array and then storing it in a wave file. The details about extraction procedures are summarised in Algorithm 2.

Algorithm 2: Secret speech extraction process

```
EXTRACTION(BitStream)
$k \leftarrow 1$
$F \leftarrow 0$
FM ← []
Secret_speech← []
$k \leftarrow 1$                                          ▷ or $k > 1$ for ON-OFF mode
for $i \leftarrow 1$ to number_frames increase by $k$ do
    $F \leftarrow \text{LSB}_{\text{extract}}(\text{Frame[i].PPI}_{\text{max}})$
    FM ← FM + $F$
end
$P_n(x) \leftarrow \text{CNNAutoDecoder}(FM)$
$E_j \leftarrow \text{polyval}(P_n(x))$                   ▷ Evaluate the polynomial
QLSP ← AMR_decoder(BitStream)
$k \leftarrow 1$                                          ▷ or $k > 1$ for ON-OFF mode
for $i \leftarrow 1$ to number_frames increase by $k$ do
    $S \leftarrow 0$
    for $j \leftarrow 0$ to 3 do
        $\hat{f}^{(j)}_{\text{ind}(j)} \leftarrow \text{Convert}(\text{QLSP}[i, j])$   ▷ Convert LSP to polynomial
        $S \leftarrow \hat{f}^{(j)}_{\text{ind}(j)} + E_j$
    end
    Secret_speech[i] ← S                              ▷ Buffering secret speech
end
return Secret_speech
```

4.3 SIAE's steganographic key

A secret key steganography is analogous to a symmetric ciphering key, which is required by both the sender and the receiver to hide and recover the secret message. Only if the recipient knows the secret key used during the embedding process, he/she can reverse the process and retrieve the secret message. The secret steganographic key for the SIAE method consists of the following parameters: the interpolation degree n, the polynomial coefficients $P_n(x)$ calculated from Equation (12), and the ind(j) obtained from Equation (9).

5 Experimental Results

In order to evaluate the performance of the proposed method, it was applied on a G.723.1 codec operated at 6.3 kbps. The perceptual quality of the stego speech with a secret message embedded using our SIAE method is computed and compared to that of the original G723.1 speech (without steganography). We refer to this comparison as the ***steganographic quality loss (SQ-Loss)*** metric. Moreover, the experiments evaluate the performance of the proposed solution in two additional metrics, including ***capacity*** and ***security***. All experiments are compared with other works in the literature.

Typically, imperceptibility is not considered a requirement in steganography [4] as long as a perceptible change does not lead to the detection of the secret message. For example, if the sentence “My daughter is pretty” was replaced by “My son is handsome” without any audible artifacts in the stego speech, such a change would possibly stay unnoticed by anyone who is sniffing the network traffic. In our case, however, the type of distortions introduced in the embedding speech must remain below the perceptibility threshold if we want the secret message to stay undetectable, and that is why we analyze imperceptibility too.

In case that VoIP transmissions are being eavesdropped, audible distortions could raise suspicions about the existence of a covert communication. Hence, maintaining a convenient level of (cover) speech quality is a typical requirement of VoIP steganography.

5.1 Speech Database

In all experiments, the speech data used consists of 2,000 speech sentences randomly chosen from the TIMIT speech database [10] for English speakers with a sampling rate of 8 kHz. The minimum and maximum lengths of the chosen speech samples are 2.57 s and 4.49 s, respectively.

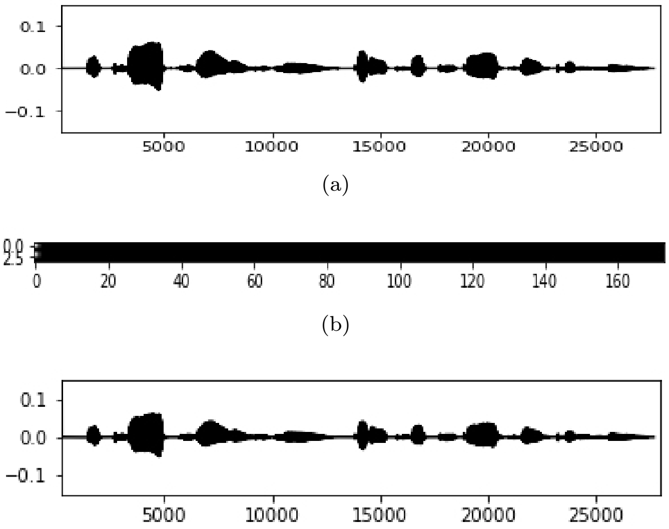
5.2 Speech Quality Assessment Method

The perceptual evaluation of speech quality (PESQ) described in the ITU-T P.862 recommendation [24] is employed to assess the speech quality. The Matlab software developed by the Center for Robust Speech Systems of the University of Texas at Dallas [17] has been used to compute the PESQ MOS results. For each encoder, there is a maximum PESQ score. This standard gives an assessment of speech quality from -0.5 to 4.5, yet the result is frequently confined to the range $[1.0, 4.5]$, similar to a mean opinion score (MOS) scale from 1 (worst) to 5 (best) [16,27,17]. In such a scale, the PESQ score equal to 2 is selected as the intelligibility threshold. The PESQ MOS of the stego speech must be greater than this threshold to avoid raising suspicions if the channel is being eavesdropped. Lower values of PESQ MOS could lead to the application of steganalysis by the eavesdropper, with the risk of the secret communication being detected.

It must be taken into account that the PESQ MOS values obtained with different high-quality coders can be greater than 4 (G.711), somewhat lower than 4 (G.723.1 and Speex), around 3 (MELP 2.4 kbps) and even somewhat lower than 3 (MELP 1.2 kbps). Since some high-quality coders can lead to values lower than 3, we have set the threshold for avoiding suspicions to 2, although it could also be somewhat larger (around 2.5). Needless to say, the higher the PESQ MOS score achieved after data embedding, the better for avoiding such suspicions.

5.3 Performance Tests of the CNN Auto-Encoder

In order to evaluate the proposed CNN auto-encoder, we have fixed the size of the input frame signal to 160 samples, i.e. 20 ms per frame. In every epoch, 85% of the secret speeches are used for training (1,700 speech files) and 15% (300 speech files) of database are used for validation. The CNN auto-encoder is run on a central processing unit (CPU), since all traditional speech coding algorithms, including our cover's coder G723.1, run on CPU as well.



(a)

(b)

(c)

Fig. 6 Performance of proposed CNN auto-encoder, (a) original speech, (b) feature-map, (c) CNN auto-encoded speech.

An example of a single CNN speech predicted from our auto-encoded model is shown in Fig.6. In this example, an input signal of 27,680 samples is compressed to a feature-map of 346 samples and reshaped into two rows and 173 columns to fit the input of the CNN decoder $2 \times x$. This means that the original signal has been compressed 80 times less than its original size, equivalent to maintaining 1.25% of its original size.

The experimental results show that, when the proposed auto-encoder learns the training dataset (accuracy of training is 100%), the mean squared error (MSE) of the proposed CNN auto-encoder is about 0.007 when we do not use a batch-normalisation layer, and it can reach $3.5 \cdot 10^{-3}$ when we include it. Adding more batch-normalisation layers does not lead to any other improvement in terms of MSE. The last MSE is the minimal modification of the secret signal caused by the proposed CNN auto-encoder.

5.4 Performance Testes of Embedding A Feature-Map in PPI

The PPI consists of ony five values (with 13 bits each) transmitted in one speech packet of 240 samples. Several tests have to be carried out in order to determine which size of the feature-map can be transmitted in one packet at a time. For example, consider the input signal as cover speech , which has 27,680 samples. The latter signal would be divided into 115 packets ($27,680/240 = 115$ packets). Each packet produces an array E of four samples each, which makes it possible to embed $115 \cdot 4 = 460$ samples. The experimental results show that, after CNN auto-encoding, 460 samples are compressed to only 8 feature-map samples of 32 bits each. It is also shown that PPI can support a change of only 8 bits in each PPI parameter (73 bits each) in a packet, otherwise, the quality of the stego speech would decrease significantly. For that reason, we divided each feature-map sample into 8 bits each. This operation results in a feature-map array of 32 samples of 8 bits each.

5.4.1 Analysis of Imperceptibility and SQ-Loss

From the discussion above, it is proven that $|M'| \ll |C|$. For this reason, we have embedded the feature-map into the PPI coefficients in two different ways:

(a) Embedding the feature-map in a continuous manner (single block). In the previous example, 32 feature-map samples were embedded in the first 32 packets and the remaining 84 packets were left in their original state without any steganograms.

(b) Embedding the feature-map in an ON-OFF way, embedding in one packet, then skip k packets, and embed again. In this way, spread the feature-map among the packets of the cover speech.

Fig. 7 depicts the performance of the SIAE scheme applied to 20 speech files using the two mentioned transmission ways, with $k = 2$ in ON-OFF mode. G723.1 codes these 20 files with an average PESQ equal to 3.665 MOS, a maximum equal to 3.783 and a minimum equal to 3.601 MOS. Embedding the feature-map using a single-block and ON-OFF way yields an average PESQ equal to 3.540 and 3.583 MOS, respectively, with a maximum PESQ of 3.663 and 3.712 MOS, respectively, and a minimum PESQ of 3.372 MOS and 3.474 MOS, respectively. This indicates that the proposed method provides a high imperceptibility, with a very low average **SQ-Loss**, equal to 0.125 MOS and 0.082 MOS, respectively, for single-block and ON-OFF transmission modes. Hence, embedding the feature-map using ON-OFF

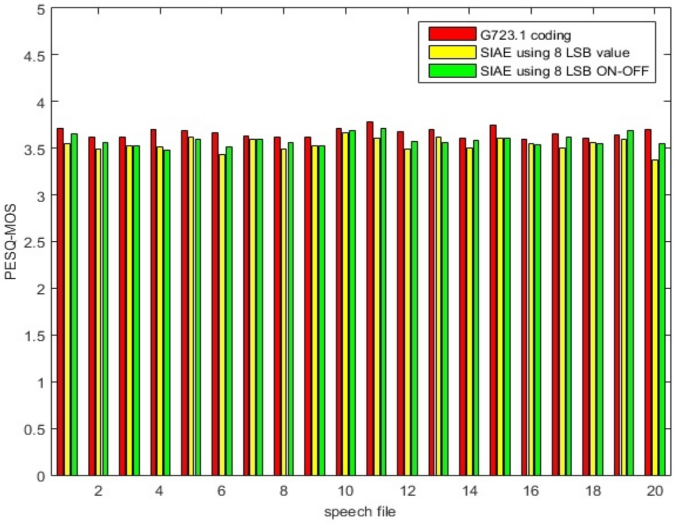


Fig. 7 PESQ of the stego speech after embedding the feature-map in the PPI parameter.

provides, most of the time, a slight enhancement in imperceptibility compared to the single-block mode. The variation of PESQ for both modes shows that the suggested approach, along with speech coding, has a reduced influence on the synthesised speech quality.

The proposed SIAE method and the AFA [42], PDM-AFS [43], Miao et al. [35], and Geiser and Vary [11] schemes create a stego speech with average PESQ scores of 3.54 and 3.58 (for single-block and ON-OFF modes), 3.79, 3.69, 3.66, and 3.66 MOS, respectively. In terms of quality, the PESQ scores of the stego audio produced by SIAE, PDM-AFS, Miao et al., and Geiser and Vary are practically identical, indicating that there is no substantial variation in imperceptibility between these four steganographic schemes. Furthermore, the AFA-generated stego speech has a slightly higher average PESQ score than the other methods. However, the SIAE's average PESQ scores are 3.54 and 3.58, indicating that the scheme's perceptual quality is sufficient and the perceptual loss cannot be perceived by the human auditory. In addition, as shown in the following sections, the AFA scheme provides worse capacity and detectability results as compared with the proposed SIAE scheme. Table 3 summarizes a comparison between the proposed approach and several other schemes in the literature.

For different SIAE steganography modes, the average and variance of PESQ, of both cover and stego speech, are calculated and compared with different works in literature. The average and variance are two metrics that can evaluate the stability of a scheme. In Table 3, the results show that the PESQ of the SIAE scheme is higher than the other schemes, and the result is statistically stable (the variance is smaller).

For the whole speech database, the SQ-Loss (ΔPESQ between the original and the stego speeches) are summarised in Table 4 and compared with other works that consider SQ-Loss as a metric.

Table 3 PESQ for different steganography schemes.

Metric	Steganographic scheme	Coding rate (kbps)	PESQ-MOS
Average	Cover	6.3	3.533
	SIAE (Block)	**6.3**	**3.399**
	SIAE (ON-OFF)	**6.3**	**3.437**
	Huang et al. [19]	6.3	3.311
	PDU-AAS [41]	6.7	3.256
	Yan et al. [50]	6.7	3.176
	AFA [42]	12.2	3.790
Variance	Cover	6.3	0.019
	SIAE (Block)	**6.3**	**0.025**
	SIAE (ON-OFF)	**6.3**	**0.018**
	Huang et al. [19]	6.3	0.069
	PDU-AAS [41]	6.7	0.068
	Yan et al. [50]	6.7	0.061
	AFA [42]	12.2	0.025

Table 4 Comparison of the maximum, minimum and average SQ-Loss (ΔPESQ) between SIAE and other works in literature

Scheme		ΔPESQ Max	ΔPESQ Min	ΔPESQ Mean
SIAE	**Block**	**0.6156**	**0.0031**	**0.1902**
	ON-OFF	**0.5200**	**0.0026**	**0.1666**
Janicki [25]		1.29	0.10	0.5 to 0.7
Liu et al. [29]		1.04	0.11	0.590
Peng et al. [37]		0.62	0.61	0.615
Peng et al. [38]		≈ 1.00	≈ 0.25	≈ 0.20
PDM-AFS [43]		≈ 0.60	≈ 0.20	≈ 0.33

It can be noticed that the proposed method is better than other works in the literature in term of SQ-Loss; hence, better in terms of imperceptibility.

5.4.2 Hiding Capacity Analysis

The embedding capacity of the proposed method is evaluated according to the following equation:

$$C = \frac{4n_b R}{N} \text{ [bps]}, \tag{20}$$

where:

- n_b is the number of embedded bits per sub-frame (in our case, $n_b = 8$ bits for any coding rate of G723.1).
- R is the coder rate (in our case, $R = 6{,}300$ bps).
- N is the number of bits per frame (in our case, $N = 189$ bits).

The embedding rate is the amount of bits that steganograms occupy in one packet, mathematically:

$$\text{Emb\_rate}(\%) = \frac{N_{\text{steg}}}{N} \cdot 100, \tag{21}$$

where N_{steg} stands for the number of hidden bits in one frame.

- **Hiding capacity without the auto-encoder:** In the proposed method, the embedded data is only 8 bits per frame, in other words, we embed 8 bits per 4 sub-frames. Using Equation (20), the embedding capacity is $(4{\cdot}8/4){\cdot}6300/189 = 266.66$ bps. In that case, embedding an array E_j, requires four consecutive frames, meaning that the cover should be four times greater than the embedded data. The embedding rate, according to Equation (21), is $(8/189{\cdot}100) = 4.23\%$.
- **Hiding capacity with auto-encoder:** For schemes that auto-encode their steganogram, the proposed method can offer a very high capacity steganograhic channel, in a G723.1 bitstream, with capacity, according to Equation (20), equal to $(4 \cdot 8 \cdot 6300/189) = 1.06$ kbps. The embedding rate can be obtained as follows:

$$\text{Emb\_rate}(\%) = \frac{N_{\text{steg}}}{N} \cdot \mu \cdot 100, \tag{22}$$

  where μ is the compression coefficient that varies from one auto-encoder architecture to another. The experiments show that the proposed auto-encoder yields $\mu \approx 0.28$. Since we use the auto-encoder, we need only to embed $N_{\text{steg}} = 8$ bits/frame of feature-map rather than secret data, and the embedding rate is computed using Equation (22), which yields $(8/189) \cdot 100 \cdot 0.28) = 1.185\%$. The apparent contradiction between a large hiding capacity of 1.06 kbps and an embedding rate of only 1.185% is justified by the use of the auto-encoder in the proposed scheme.

Table 5 summarizes the characteristics of the proposed method, and compares it with other methods in literature in terms of coding rate, capacity in bits per frame, capacity in bits per second and embedding rate.

Table 5 Maximum embedding capacity for various steganography methods with different encoders.

Scheme		Coding rate (kbps)	Capacity (bits/frame)	Capacity (kbps)	Embedding rate (%)
SIAE		**6.3**	**32**	**1.06**	**1.18**
Huang et al. [19]		6.3–12.2	4	0.20	1.64–2.11
Peng et al. [37]		64	15–240	0.50–8.00	0.78–12.50
Peng et al. [38]		64	24	0.80	1.25
Methods compared in [43]	PDM-AFS [43]	12.2	42	2.06	17.21
	Geiser and Vary [11]	12.2	40	2.00	16.40
	Miao et al. [35]	12.2	40	2.00	16.40
	AFA [42]	12.2	20	1.00	8.19

From Table 5, it is obvious that the proposed method is the best among all the cited methods [21], [37], and [38] in capacity and, at the same time, it alters only 1.18% of the frame bits, which is the benefit of using the auto-encoder. The methods compared in [43] use the PPI parameter to embed secret data and are the only schemes comparable to the proposed method in terms of capacity. However, this capacity is achieved when the embedding rate is high (up to 17.21%), which may considerably affect the statistical proprieties of the signal if they are implemented on a lower coder rate. Note that the latter methods use a 12.2 kbps coder rate (244 bits per frame), which may allow more embedded bits than our coder that uses half that rate (6.3 kbps, 189 bits/frame). Both the proposed SIAE scheme and the AFA method [42] provide a capacity of 1 kbps. The SIAE method has been implemented in a coder with rate equal to 6.3 kbps and, thus, it consumes half the transmission bandwidth compared to the AFA scheme. Moreover, the proposed method modifies 1.18% of the cover (frame of 189 bits) to transmit 1 kbps, whereas the AFA method modifies 8.19% of the cover (frame of 244 bits) to transmit the same amount of hidden data. The AFA scheme may considerably affect the statistical proprieties of the signal if it is implemented on a 6.3 kbps coder rate.

The only method that can be compared to SIAE in terms of coding rate is the one proposed by Huang et al. in [21]. This method provides a very low steganographic capacity with a competitive embedding rate. The common characteristic in SIAE and Huang et al.'s steganographic schemes [21] is the capacity in each frame (32 and 4 bits per frame, respectively), which is fixed whatever the coding rate. The method described in [38] is the only scheme whose results are comparable to the proposed method in terms of embedding rate (1.25% in a frame length equal to 1920 bits). However, the embedding capacity of [38] is lower compared to the proposed SIAE method, although [38] uses G.711, whose coding rate is 64 kbps (ten times that of G723.1). Similarly, the method [37] also employs a high coder rate G.711, whose coding rate is 64 kbps. The proposed method can compete with [37] in terms of capacity and embedding rate.

5.4.3 Statistical Security Analysis

The SIAE scheme produces a stego speech that is considered as a distorted version of cover speech. Since SIAE embeds only in the PPI parameters, the G723.1 decoder selects a different row of the fixed codebook (FCB) from the row that would be used without applying SIAE method and, hence, the coder produces distorted version of the speech. Let us denote the cover sequence at the input of the encoder by $x = (x_1, x_2, \ldots, x_i, \ldots, x_L)$, where the sample x_i is an integer and L is the length of the cover and stego signals. We assume that the cover x to be fixed and the embedding operations on x_i are mutually independent, and thus the distortion is introduced by changing x to $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_i, \ldots, \hat{y}_L)$ instead of producing $y = (y_1, y_2, \ldots, y_i, \ldots, y_L)$. The latter fact is caused by applying a bit-mask function ($F(\cdot)$) that clears the LSB of a PPI value ($\mathrm{PPI}_{\max}$) among five. Note that $\mathrm{PPI} = [\mathrm{PPI}_1, \mathrm{PPI}_2, \ldots, \mathrm{PPI}_5]$, applying the LSB method on the cover, yielding to: $\mathrm{PPI}'_{\max} = F(\mathrm{PPI}_{\max}) + \mathrm{FM}$, where FM are 8 bits taken from the feature map to be embedded. The average of the non-linear distortion $D(y, \hat{y})$ at signal level can be simply denoted as follows:

$$D(y, \hat{y}) = \frac{1}{n} \min_{y} \sum_{i=1}^{n} |\hat{y}_i - y_i| \,. \tag{23}$$

In this paper, we consider two categories of steganalytic methods to estimate the impact of distortion caused by our scheme to the hole speech signal database, which are the following:

- **Necessary steganalytic methods**: They cover all basic methods that rely on speech recognition features such as DMFCC. If the proposed steganographic scheme is successfully detected using these features, that would mean the method is very weak, causing strong changes in the statistical proprieties of the signal and making a deeper analysis unnecessary. If the proposed steganographic schemes prevent being detected by this category, we can consider the second category of steganalysis.

- **Necessary and sufficient steganalytic methods**: This category includes all the methods that make a deep analysis to detect steganograms, such as convolutional neural networks (CNN), which obtain a group of features (chaotic features) in an automatic manner. Besides, this category covers all steganalytic methods that are targeted to specific steganographic schemes and successfully uncover and detect the steganograms.

Statistical security represents the ability of the steganographic scheme to avoid the detection by means of statistical analysis and/or steganalysis. The test error rate (TER) is widely used as a metric, such as in [43,41,42,13] to evaluate statistical security, which is calculated using the following equation:

$$\mathrm{TER} = \frac{1}{2}(P_{\mathrm{MA}} + P_{\mathrm{FA}}), \tag{24}$$

where P_{MA} and P_{FA} are the probabilities that the cover speech has been misclassified as stego speech and the stego speech has been misclassified as cover speech, respectively. A hither TER means that the undetectability is higher for the steganographic scheme, as long as the value does not become larger than 0.5. Note that, for a totally detectable scheme, we would have $P_{\mathrm{MA}} = 0$ and $P_{\mathrm{FA}} = 0$, since all the tested objects will be correctly classified as cover or stego. On the other hand, for maximum undetectability, the steganalytic classifier would perform (almost) randomly both for cover and stego objects, thus yielding $P_{\mathrm{MA}} \approx 0.5$ and $P_{\mathrm{FA}} \approx 0.5$ and, hence, TER ≈ 0.5. Besides the TER metric, we employed the accuracy (Acc) metric. Formally, the accuracy can be calculated using equation 25.

$$\mathrm{Acc}(\%) = \frac{\mathrm{TPR} + \mathrm{TNR}}{\mathrm{TPR} + \mathrm{TNR} + \mathrm{FPR} + \mathrm{FNR}} \cdot 100, \tag{25}$$

where, TPR is the rate of true positives out of all positives, FPR is the rate of false positives out of all negatives, TNR is the rate of true negatives out of all negatives and FNR is the rate of false negatives out of all negatives.

The ideal condition for statistical undetectability is considered to achieve detection accuracy around 50%, which is equivalent to random guessing. We have used the derivative mel-frequency cepstral coefficients (DMFCC) as a feature and the binary support vector machine (SVM) classifier to detect speech steganography.

Table 6 summarises the results obtained for both accuracy and TER metrics with different frame sizes. 200 files have been use for training with a 20% for testing. The testing database has been cross-validated ten times to prevent the over-fitting phenomenon.

It can be observed that the proposed scheme is extremely secure, since the average accuracy is around 50% for both modes, block and ON-OFF. The maximum obtained accuracy results are 0.54% and 0.50%, and the minimum are 0.45% and 0.49%, for block and ON-OFF modes, respectively. The obtained TER is around 0.49% and 0.50% for block and ON-OFF modes, respectively. These high TER values indicate that the proposed method evades detection by steganalytic schemes using DMFCC as a feature.

Table 7 presents an outline of the comparison of the proposed scheme with the methods in [19,50,46]. Those methods also use DMFCC as a feature, SVM as a

Table 6 Steganalysis results of the SIAE scheme through the use of DMFCC at different analysis windows sizes.

Frame	Window (ms)	SIAE (Block)		SIAE (ON-OFF)	
		Acc	TER	Acc	TER
0.5	15	0.4881	0.5119	0.4943	0.5057
1	30 (standard)	0.4529	0.5471	0.4966	0.5034
2	60	0.4573	0.5427	0.5034	0.4966
3	90	0.5049	0.4951	0.5011	0.4989
5	150	0.5043	0.4957	0.4966	0.5034
7	210	0.5449	0.4551	0.5034	0.4966
10	300	0.5265	0.4735	0.4960	0.5040
20	600	0.5119	0.4881	0.5028	0.4972
30	900	0.5195	0.4805	0.4966	0.5034
	Max	0.5449	0.5471	0.5034	0.5057
	Min	0.4529	0.4551	0.4943	0.4966
	Average	0.5055	0.4944	0.4989	0.5010
	Standard deviation	0.0258	0.0258	0.0037	0.0037

classifier, accuracy and TER as statistical metrics. It can be observed that both SIAE methods provide accuracy results closer to 0.5 compared to [19, 50, 46]. The same remark can be noticed regarding TER metric, since the proposed scheme provide the highest TER compared to the latter cited methods.

The steganalysis framework based on the probability of same pulse position (SPP) in the same track is proposed in [40]. This approach can be considered as the necessary and sufficient steganalytic method as long as it is targeted for AMR steganography-based PPI methods. For that reason, we have extracted and classified, using SVM, the SPP feature of both clean speech and stego speech, and the results are summarized in Table 7, where compared with other existing work in the literature. Besides extracting SPP features from the TIMIT database, the SPP features has been extracted from the CMU database also (4000 recorded speech), to be in the same conditions with the methods cited in [43] so as to establish a fair comparison. It can be noticed that the SPP features fails to detect the SIAE method. We can achieve best security level (TER values closer to 0.5), especially for the on-off mode, compared to the methods cited in [43].

In fact, for almost all steganographic methods, there is a trade-off between the coder rate (number of bits per frame), the embedding rate and security (in terms of undetectability). The higher the coder rate with a lower embedding rate, the higher the security of the steganographic scheme. This trade-off is illustrated in Tables 5 and 7. It can be observed that the proposed method, for both modes, has a better security level (TER values closer to 0.5) compared to the methods summarised in both two tables. The compression provided by the auto-encoder reduces the embedding rate and the proposed method is designed in such a way that it does not leave statistically significant traces in the stego speech or, in other words, it causes a reduced impact and evades steganalysis.

Besides the above explanation, the steganalysis of the LSB method had precomputed results (for low embedding rate) using an appropriate method in [51], which is based on a high order histogram moments in frequency domain (HMFD) to extract a single specific vector feature using wavelet packet decomposition (WPD). The latter method produces an accuracy equal to 56.7% 58.3%, and 60.8% when

Table 7 Comparison based on SIAE's accuracy and TER results with other works in the literature that use DMFCC or SPP as features.

Feature	Database	Scheme		Accuracy	TER
DMFCC	TIMIT	**SIAE**	**Block**	**0.547**	**0.453**
			On-Off	**0.503**	**0.479**
		Huang et al. [19]	CSW	0.537	0.462
		Yan et al. [50]	sample-2	0.521	0.478
		Tang et al. [46]	CW	0.574	0.425
SPP [40]	TIMIT	**SIAE**	**Block**	**0.613**	**0.387**
			On-Off	**0.571**	**0.429**
	CMU [43]	**SIAE**	**Block**	**0.540**	**0.460**
			On-Off	**0.556**	**0.444**
		Methods compared in [43]	PDM-AFS [43]	0.581	0.419
			Geiser and Vary [11]	0.925	0.075
			Miao et al. [35]	0.908	0.092
			AFA [42]	0.800	0.200

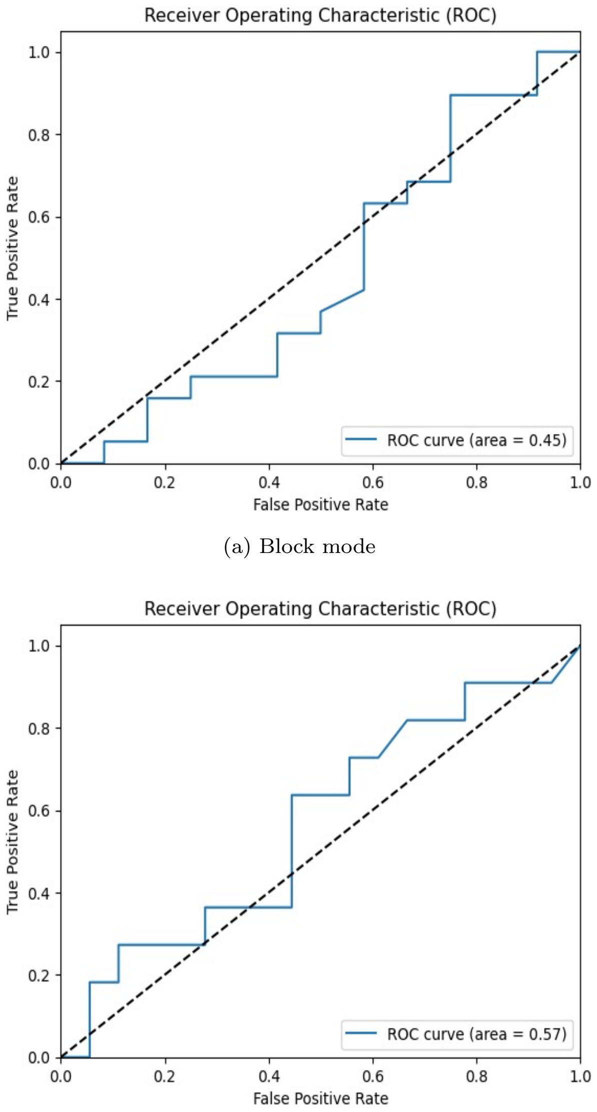
embedding rate equals 1%, 2%, and 3%, respectively. Since our scheme embeds steganograms in only 1.18% and the accuracy is around 56.7%, this means that the mentioned method is not effective to detect our scheme. In speech steganalytic, a scheme that produces an accuracy lower than 80% is considered ineffective and can not detect the steganograms.

A single feature may provide an indication of the presence of steganography, but changing the feature or using several features may lead to conflicting classification results. For this reason, combining multiple features in a single classifier is typically the preferred solution. In order to test this scenario, we have used a CNN binary classifier by adopting the architecture of the powerful CNN steganalysis scheme in [49] that shows an excellent accuracy. We are motivated by the latter method because it employs, in various stages, 1008 different filters with size 3×3, for long-term analysis and 1920 different filters with size 1×1 for short-term analysis. That indicates that the method extracts 1008 features from 9 samples and, then, extracts 1920 features from each sample, yielding to high precision in steganalysis that can achieve an accuracy of 90.39%. Fig. 8 (a) and (b) shows the receiver operating characteristic (ROC) curves of the steganalysis for the two SIAE modes.

The ROC curve is a plot that illustrates the performance of a binary classifier system. It is created by plotting the TPR versus the FPR. TPR is also known as sensitivity, and FPR = 1 − TNR, also known as false alarm probability. The obtained ROC curves show that the steganalysis of the SIAE output has an area under the roc curve (AUC) of 0.45 and 0.57 for block and ON-OFF modes, respectively, that are close to 0.5 (no discrimination). These results support the accuracy results obtained with the SVM classifier summarised in Table 6 and the results obtained in [51]. Thus, the proposed method is secure enough to transmit the secret speech and evades steganalysis.

6 Conclusion

In this paper, we propose a novel steganographic scheme, called Steganography-based Interpolation and Auto-Encoding (SIAE), that is based on calculating, in each frame, the error E between four QLSP and four secret speech samples. The



(a) Block mode

(b) ON-OFF mode

Fig. 8 ROC curves for the CNN classifier with SIAE for block and ON-OFF modes.

error E is then interpolated and compressed to a very short bit string using a CNN auto-encoder. The proposed CNN auto-encoder's architecture provides an excellent performance with a low MSE. Only one value of PPI among five is selected, in each frame, to carry the hidden feature-map, which leads to reduced distortion for the transmitted stego speech. Many statistical metrics have been evaluated to prove the high level of indetectability of the proposed scheme. The results show that the proposed approach is secure enough to transmit secret information. The secrecy is shown using different statistical metrics such as SQ-Loss, variance, TER, accuracy, and ROC curves. In addition, the proposed method provides a higher capacity compared to other methods in the literature, while keeping statistical detectability at the level of random guessing.

For further research, we will explore the possibility of applying the suggested scheme to other lower bit rate AMR modes, such as the 5.3 kbps mode. This would require transmitting the compressed data into a covert channel established in the transmission protocols, due to the low length of the PPI parameters in that mode. Besides that, a steganalysis method will be developed in the near future to deal with this kind of embedding scheme that alters a very limited number of bits.

Acknowledgements The first author would like to acknowledges the funding by the directorate general for scientific research and technological development (DGRSTD). The second author acknowledges the funding of the Spanish Government and the EIG CONCERT-Japan through grants RTI2018-095094-B-C22 “CONSENT” and PCI2020-120689-2 “DISSIMILAR”.

References

1. Ballesteros, D.M., Renza, D.: Secure speech content based on scrambling and adaptive hiding. Symmetry **10**(12), 694 (2018)
2. Berk, V., Giani, A., Cybenko, G., Hanover, N.: Detection of covert channel encoding in network packet delays. Rapport technique TR536, de lUniversité de Dartmouth **19** (2005)
3. Bobade, S., Goudar, R.: Secure data communication using protocol steganography in ipv6. In: Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on, pp. 275–279. IEEE (2015)
4. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital watermarking and steganography. Morgan Kaufmann (2007)
5. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE transactions on image processing **6**(12), 1673–1687 (1997)
6. Cvejic, N., Seppanen, T.: A wavelet domain lsb insertion algorithm for high capacity audio steganography. In: Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th, pp. 53–55. IEEE (2002)
7. Delforouzi, A., Pooyan, M.: Adaptive digital audio steganography based on integer wavelet transform. Circuits, Systems & Signal Processing **27**(2), 247–259 (2008)
8. Elsadig, M.A., Fadlalla, Y.A.: Packet length covert channels crashed. Journal of Computer Science & Computational Mathematics (JCSCM) **8**(4), 55–62 (2018)
9. Fraczek, W., Mazurczyk, W., Szczypiorski, K.: Stream control transmission protocol steganography. In: Multimedia Information Networking and Security (MINES), 2010 International Conference on, pp. 829–834. IEEE (2010)
10. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S.: Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. NASA STI/Recon technical report n **93** (1993)
11. Geiser, B., Vary, P.: High rate data hiding in acelp speech codecs. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4005–4008 (2008)
12. Ghasemzadeh, H., Kayvanrad, M.H.: Toward a robust and secure echo steganography method based on parameters hopping. In: 2015 Signal Processing and Intelligent Systems Conference (SPIS), pp. 143–147. IEEE (2015)
13. Gong, C., Yi, X., Zhao, X.: Pitch delay based adaptive steganography for amr speech stream. In: International Workshop on Digital Watermarking, pp. 275–289. Springer (2018)
14. Gopalan, K., Wenndt, S., Noga, A., Haddad, D., Adams, S.: Covert speech communication via cover speech by tone insertion. In: Proc. 2003 IEEE Aerospace Conference, vol. 4, pp. 4_1647–4_1653 (2003)
15. Hamdaqa, M., Tahvildari, L.: Relack: a reliable voip steganography approach. In: Secure Software Integration and Reliability Improvement (SSIRI), 2011 Fifth International Conference on, pp. 189–197. IEEE (2011)
16. He, J., Chen, J., Xiao, S., Huang, X., Tang, S.: A novel amr-wb speech steganography based on diameter-neighbor codebook partition. Security and Communication Networks **2018** (2018)

17. Hu, Y., Loizou, P.C.: Evaluation of objective measures for speech enhancement. In: Ninth international conference on spoken language processing (2006)
18. Huang, T., Zhang, L., Hu, X., Lei, X.: A data validation method based on ip covert channel packet ordering. In: 2018 14th International Conference on Computational Intelligence and Security (CIS), pp. 223–227. IEEE (2018)
19. Huang, Y., Liu, C., Tang, S., Bai, S.: Steganography integration into a low-bit rate speech codec. IEEE transactions on information forensics and security **7**(6), 1865–1875 (2012)
20. Huang, Y., Xiao, B., Xiao, H.: Implementation of covert communication based on steganography. In: Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on, pp. 1512–1515. IEEE (2008)
21. Huang, Y.F., Tang, S., Yuan, J.: Steganography in inactive frames of voip streams encoded by source codec. IEEE Transactions on information forensics and security **6**(2), 296–306 (2011)
22. ITU, I.: 723.1: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. Telecommunication Standardization Sector of ITU (1996)
23. ITU-T, D.R.S.C.: for multimedia communications transmitting at 5.3 and 6.3 kbit/s. ITU-T Recommendation G **723** (2006)
24. ITU-T Recommendation: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs (2001). Rec. ITU-T P. 862
25. Janicki, A.: Pitch-based steganography for speex voice codec. Security and Communication Networks **9**(15), 2923–2933 (2016)
26. Keles, H.Y., Rozhon, J., Ilk, H.G., Voznak, M.: Deepvocoder: A cnn model for compression and coding of narrow band speech. IEEE Access **7**, 75081–75089 (2019)
27. Kheddar, H., Bouzid, M., Megías, D.: Pitch and fourier magnitude based steganography for hiding 2.4 kbps melp bitstream. IET Signal Processing **13**(3), 396–407 (2019)
28. Liu, J., Zhou, K., Tian, H.: Least-significant-digit steganography in low bitrate speech. In: Communications (ICC), 2012 IEEE International Conference on, pp. 1133–1137. IEEE (2012)
29. Liu, X., Tian, H., Huang, Y., Lu, J.: A novel steganographic method for algebraic-code-excited-linear-prediction speech streams based on fractional pitch delay search. Multimedia Tools and Applications **78**(7), 8447–8461 (2019)
30. Mazurczyk, W.: Voip steganography and its detection—a survey. ACM Computing Surveys (CSUR) **46**(2), 20 (2013)
31. Mazurczyk, W., Lubacz, J.: Lack—a voip steganographic method. Telecommunication Systems **45**(2-3), 153–163 (2010)
32. Mazurczyk, W., Szaga, P., Szczypiorski, K.: Using transcoding for hidden communication in ip telephony. Multimedia Tools and Applications **70**(3), 2139–2165 (2014)
33. Mazurczyk, W., Szczypiorski, K.: Covert channels in sip for voip signalling. In: Global e-security, pp. 65–72. Springer (2008)
34. Mazurczyk, W., Szczypiorski, K.: Steganography of voip streams. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 1001–1018. Springer (2008)
35. Miao, H., Huang, L., Chen, Z., Yang, W., Al-Hawbani, A.: A new scheme for covert communication via 3G encoded speech. Computers & Electrical Engineering **38**(6), 1490–1501 (2012)
36. Miao, R., Huang, Y.: An approach of covert communication based on the adaptive steganography scheme on voice over ip. In: Communications (ICC), 2011 IEEE International Conference on, pp. 1–5. IEEE (2011)
37. Peng, J., Jiang, Y., Tang, S., Meziane, F.: Security of streaming media communications with logistic map and self-adaptive detection-based steganography. IEEE Transactions on Dependable and Secure Computing (2019)
38. Peng, J., Tang, S.: Covert communication over voip streaming media with dynamic key distribution and authentication. IEEE Transactions on Industrial Electronics (2020)
39. Qi, Q., Peng, D., Sharif, H.: Dst approach to enhance audio quality on lost audio packet steganography. EURASIP Journal on Information Security **2016**(1), 1–10 (2016)
40. Ren, Y., Cai, T., Tang, M., Wang, L.: AMR steganalysis based on the probability of same pulse position. IEEE Transactions on Information Forensics and security **10**(9), 1801–1811 (2015)
41. Ren, Y., Liu, D., Yang, J., Wang, L.: An AMR adaptive steganographic scheme based on the pitch delay of unvoiced speech. Multimedia Tools and Applications **78**(7), 8091–8111 (2019)

42. Ren, Y., Wu, H., Wang, L.: An AMR adaptive steganography algorithm based on minimizing distortion. Multimedia Tools and Applications **77**(10), 12095–12110 (2018)
43. Ren, Y., Yang, H., Wu, H., Tu, W., Wang, L.: A secure AMR fixed codebook steganographic scheme based on pulse distribution model. IEEE Transactions on Information Forensics and Security **14**(10), 2649–2661 (2019)
44. Schmidt, S., Mazurczyk, W., Kulesza, R., Keller, J., Caviglione, L.: Exploiting ip telephony with silence suppression for hidden data transfers. Computers & Security **79**, 17–32 (2018)
45. Su, Z., Li, W., Zhang, G., Hu, D., Zhou, X.: A steganographic method based on gain quantization for ilbc speech streams. Multimedia Systems pp. 1–11 (2019)
46. Tang, S., Chen, Q., Zhang, W., Huang, Y.: Universal steganography model for low bit-rate speech codec. Security and Communication Networks **9**(8), 747–754 (2016)
47. Tian, H., Jiang, H., Zhou, K., Feng, D.: Adaptive partial-matching steganography for voice over ip using triple m sequences. Computer Communications **34**(18), 2236–2247 (2011)
48. Tian, H., Liu, J., Li, S.: Improving security of quantization-index-modulation steganography in low bit-rate speech streams. Multimedia systems **20**(2), 143–154 (2014)
49. Wang, Y., Yang, K., Yi, X., Zhao, X., Xu, Z.: Cnn-based steganalysis of mp3 steganography in the entropy code domain. In: Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, pp. 55–65 (2018)
50. Yan, S., Tang, G., Sun, Y., Gao, Z., Shen, L.: A triple-layer steganography scheme for low bit-rate speech streams. Multimedia Tools and Applications **74**(24), 11763–11782 (2015)
51. Yang, W., Tang, S., Li, M., Cheng, Y., Zhou, Z.: Steganalysis of low embedding rates lsb speech based on histogram moments in frequency domain. Chinese Journal of Electronics **26**(6), 1254–1260 (2017)
52. Yargıçoğlu, A., İlk, H.G.: Hidden data transmission in mixed excitation linear prediction coded speech using quantisation index modulation. IET Information Security **4**(3), 158–166 (2010)
53. Zhang, L., Huang, T., Rasheed, W., Hu, X., Zhao, C.: An enlarging-the-capacity packet sorting covert channel. IEEE Access **7**, 145634–145640 (2019)