Cryptanalysis of a data security protection scheme for VoIP[*]

Chengqing Li[1], Shujun Li[2][†], Dan Zhang[3] and Guanrong Chen[4]

[1] Department of Mathematics, Zhejiang University, Hangzhou 310027, P.R. China
[2] Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, P.R. China
[3] College of Computer Science, Zhejiang University, Hangzhou, Zhejiang 310027, P.R. China
[4] Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong SAR, P.R. China

Abstract

Recently, a VoIP (voice over Internet protocol) technique with a new hierarchical data security protection (HDSP) scheme was proposed by using a secret chaotic bit sequence. This paper points out some insecure properties of the HDSP scheme, and then uses them to develop known/chosen-plaintext attacks. The following main findings are: 1) given n known plaintexts, about $(100 - \frac{50}{2^n})$ percent of secret chaotic bits can be uniquely determined; 2) given only one specially-chosen plaintext, all secret chaotic bits can be uniquely derived; 3) the secret key can be derived with practically small computational complexity when only one plaintext is known (or chosen). These facts reveal that HDSP is very weak against known/chosen-plaintext attacks. Experiments are given to show the feasibility of the proposed attacks. Furthermore, it is also found that the security of HDSP against the brute-force attack is not practically strong. Finally, some countermeasures are discussed for enhancing the security of HDSP, and several basic principles are suggested for the design of a secure encryption scheme.

1 Introduction

With the rapid development of the Internet and digital communication technologies, it becomes possible to make a telephone call with a PC connected to the Internet, or to connect the link between two telephones partially via the Internet. This technique is called voice over Internet protocol (VoIP in short) [1]. In fact, VoIP can be extended to provide many other services, such as fax over Internet protocol (FoIP), video teleconferencing, and so on. Due to the obvious benefits and potential applications of the VoIP technology, it attracts more and more interests from both vendors and consumers.

Differing from the traditional telephony service, VoIP faces new risks in the networked world: since all data are transmitted over the Internet, any algorithm to attack digital computers can be used to break a VoIP system. Thus, it is very important to provide sufficient security for VoIP services with a reasonable cost. In [2, 3], a new VoIP protocol with a hierarchical data security protection (HDSP) scheme was proposed as a possible solution to the following two problems: reducing continuous packet loss to avoid large voice corruption, and encrypting the transmitted voice to provide a high-level security. HDSP works under the control of a chaotic bit sequence, which is generated by iterating a discrete-time chaotic map whose initial condition and control parameter serve as the secret key. In [3], it was claimed that HDSP can resist the known-plaintext attack.

The present paper analyzes the security of HDSP in detail. Due to some insecure properties of the HDSP scheme, the following problems are found: 1) given n known plaintexts, about $(100 - \frac{50}{2^n})$ percent of secret chaotic bits can be uniquely determined; 2) given only one specially chosen plaintext, all secret chaotic bits can be uniquely derived; 3) the secret key can be derived with a practically small complexity when only one plaintext is known (or chosen). As a result, HDSP is very weak against known/chosen-plaintext attacks. In addition, it is found that the

[*]This paper has been published in *IEE Proceedings -Vision, Image & Signal Processing*, vol. 153, no. 1, pp. 1-10, 2006.
[†]Shujun Li is the corresponding author, personal web site: `http://www.hooklee.com`.

security of HDSP against the brute-force attack (i.e., the attack of exhaustively searching the key) is not practically strong.

The organization of this paper is as follows. Firstly, some preliminaries on the modern cryptology are given in Sec. 2, to facilitate the following discussions and analyses. Next, a brief introduction to HDSP is given in Sec. 3. Section 4 is the main portion of this paper, which focuses on the cryptanalysis of the HDSP scheme. Some experimental results are shown in Sec. 5 to support the theoretical results about the proposed attacks. Section 6 briefly discusses how to improve the security of HDSP, and suggests some basic principles for designing a good cipher. The last section concludes the paper.

2 Preliminaries of the Modern Cryptology

In this section, to facilitate the following discussions on HDSP and the proposed attacks, a brief introduction is given to the basic theory of the modern cryptology [4]. Generally speaking, cryptology is the technology of encryption, composing of the following two parts: 1) *cryptography*, which studies how to design good encryption algorithms; 2) *cryptanalysis*, which tries to find security weaknesses of proposed encryption algorithms and studies whether or not they are vulnerable to various attacks.

An encryption/decryption system is also called a *cipher*, or a *cryptosystem*. Accordingly, the encryption machine is called an *encipher*, and the decryption machine is called a *decipher*. The message for encryption is called the *plaintext*, and the encrypted message is called the *ciphertext*. Assuming that the plaintext and the ciphertext are denoted by P and C, respectively, the encryption procedure in a cipher can be described as $C = E_{K_e}(P)$, where K_e is the encryption key and $E(\cdot)$ is the encryption function. Similarly, the decryption procedure is $P = D_{K_d}(C)$, where K_d is the decryption key and $D(\cdot)$ is the decryption function. When $K_e = K_d$, the cipher is called a *private-key* cipher or a *symmetric* cipher. For private-key ciphers, the encryption-decryption key must be transmitted from the sender to the receiver via a separate secret channel. When $K_e \neq K_d$, the cipher is called a *public-key* cipher or an *asymmetric* cipher. For public-key ciphers, the encryption key K_e is published, and the decryption key K_d is kept private, for which no additional secret channel is needed for key transfer.

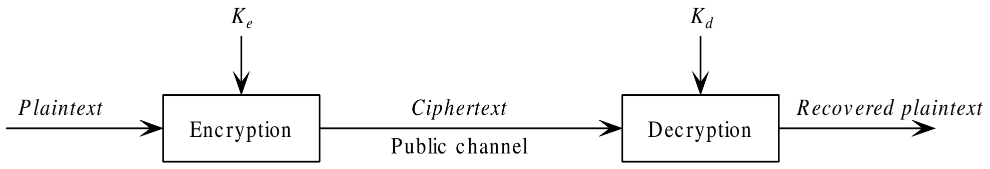

Figure 1: The encryption and decryption of a cipher.

Following the widely-acknowledged Kerckhoffs' principle in cryptology [4], the security of a cipher should rely only on the decryption key K_d, and it is assumed that all details of the encryption/decryption algorithms are known to the attackers. Thus, the main task of cryptanalysis is to reconstruct the key, or its equivalent form that can be used to successfully decrypt all or part of the plaintexts.

From a cryptographical point of view, a secure cipher should have the capability against all kinds of attacks. In most cases, the designed cipher should be secure against the brute-force attack (i.e., the attack of exhaustively searching all possible keys), and the following four typical attacks:

- *ciphertext-only attack*: attackers can only observe some ciphertexts;
- *known-plaintext attack*: attackers can get some plaintexts and the corresponding ciphertexts;
- *chosen-plaintext attack*: attackers can choose some plaintexts and get the corresponding ciphertexts;
- *chosen-ciphertext attack*: attackers can choose some ciphertexts and get the corresponding plaintexts;

In the four kinds of attacks, ciphertext-only attack is the easiest and the most common attack, due to the fact that the communication channel is generally accessible to attackers. Known/chosen-plaintext attacks are possible when an attacker can temporarily access the encryption machine, or he can successfully guess the plaintexts. Chosen-ciphertext attack is possible when an attacker can have a temporary access to the decryption machine. The last three kinds of attacks, which seemed to seldom be used in practice, are feasible in some real applications [4, Sec. 1.1] and have become more and more common in the digital world today. For example, when the VoIP software is configured to be run with a previously-stored password[1], an attacker can successfully carry out known/chosen-plaintext or chosen-ciphertext attack by hacking the computer, and he can also do so easily when the authenticated user leaves the computer without logging it out (e.g., the attacker is a colleague of the user in the same office). Generally speaking, if a cipher cannot resist the known/chosen-plaintext or chosen-ciphertext attack, it can hardly be accepted by the computer and network security community.

3 The HDSP encryption scheme

In the HDSP-based VoIP system, the encryption part is placed after the speech encoder, and the decryption part is placed before the speech decoder. So, the plaintext of HDSP is the bit-stream encoded by the speech CODEC, not the raw voice signal. The ciphertext is obtained by performing the following two steps on the plaintext [2,3]:

1. *the inter-frame interleaving (frame swapping)* – divide the plaintext into S_f-byte frames, and pseudo-randomly permute the orders of S_g continuous frames (i.e., every S_g frames compose an interleaving group);

2. *the intra-frame encryption (bit swapping and masking)* – for each input byte, pseudo-randomly permute some bits, and mask the 4 odd bits with XOR operations.

The frame/bit swapping and the bit masking operations are all controlled by a secret chaotic bit sequence $\{b(i)\}$, which is generated by iterating the chaotic Logistic map [5]: $f(x) = \mu x(1-x)$. Although the *inter-frame interleaving* was only intended for avoiding possible loss of continuous packets [2,3], the use of secret chaotic bits has effectively made it one part of the whole encryption scheme.

For a plaintext $g = \{g(i)\}_{i=0}^{N-1}$, where $g(i)$ denotes the i-th byte of g, the HDSP encryption scheme can be described as follows (to better describe the original algorithm, some notations and definitions used in [2,3] have been intentionally changed):

- *The secret key* is the control parameter μ and the initial condition $x(0)$ of the chaotic Logistic map, which are both represented in the 16-bit binary form.

- *The initialization procedure*[2]: run the Logistic map from $x(0)$ to generate a chaotic sequence $\{x(i)\}_{i=0}^{\lceil L_b/16\rceil-1}$, where L_b denotes the number of bits required during the following encryption procedure, and then extract the 16-bit representation of each chaotic state $x(i)$ to obtain a pseudo-random bit sequence (PRBS) $\{b(i)\}_{i=0}^{L_b-1}$.

- *The encryption procedure* is composed of the following stages:

  1. *The inter-frame interleaving*:

     – divide g into S_f-byte frames: $\{frame(i)\}_{i=0}^{N_f-1}$, where $N_f = \lfloor N/S_f \rfloor$;

     – further divide g into S_g-frame groups: $\{group(i)\}_{i=0}^{N_g-1}$, where $N_g = \lfloor N_f/S_g \rfloor$;

     – set $L = \lfloor \log_2 S_g \rfloor$ and $\Delta_L = S_g - 2^L$;

     – all S_g frames in each group is permuted with S_g pseudo-random swapping operations: for $i = 0 \sim (N_g - 1)$ and for $j = 0 \sim (S_g - 1)$, swap $frame(i \cdot S_g + j)$ and $frame(i \cdot S_g + j')$, where

$$j' = \sum_{k=0}^{L-1} 2^k \times b(s+k) + \sum_{k=0}^{\Delta_L - 1} b(s+L+k), \tag{1}$$

     and $s = (i \cdot S_g + j) \cdot L$.

     * Note: in this stage, totally $(N_g \cdot S_g \cdot L + \Delta_L)$ chaotic bits are required: $b(0) \sim b(N_g \cdot S_g \cdot L + \Delta_L - 1)$.

[1]As is known, to bring convenience to the end users, many softwares support the auto-saving function of the password for quick startup. Although the risk of such a function has been strongly criticized by many security experts, it is still widely used in today's password-based softwares, such as Microsoft IE, MSN, etc.

[2]Such an initialization procedure has been widely used in other chaos-based ciphers proposed by Yen (and Guo) et al., as a common method for generating pseudo-random bits. See Sec. 4.4.3 of [6] for a survey on other chaotic ciphers.

2. *The intra-frame encryption* – assuming $g^* = \{g^*(i)\}_{i=0}^{N-1}$ is the output plaintext of the inter-frame interleaving stage, the ciphertext $g' = \{g'(i)\}_{i=0}^{N-1}$ is determined in the following two steps:
   - *pseudo-randomly swap the 4 most significant bits (MSB-s) and the 4 least significant bits (LSB-s) of each byte $g^*(i) = \sum_{k=0}^{7} d_k^*(i) \cdot 2^k$ to get an intermediate byte $g^{**}(i) = \sum_{k=0}^{7} d_k^{**}(i) \cdot 2^k$*: $\forall\ k = 0 \sim 3$,
   $$\left(d_k^{**}(i), d_{k+4}^{**}(i)\right) = Swap_{b(4i+k)}\left(d_k^*(i), d_{k+4}^*(i)\right), \tag{2}$$
   where $Swap_w(a,b) = \begin{cases}(a,b), & w=0,\\ (b,a), & w=1;\end{cases}$
   - *mask the 4 odd bits of $g^{**}(i)$ to get the cipher-byte $g'(i) = \sum_{k=0}^{7} d_k'(i) \cdot 2^k$*: $\forall\ k = 1,3,5,7$,
   $$d_k'(i) = d_k^{**}(i) \oplus b(4i+k), \tag{3}$$
   where $\oplus$ denotes the XOR (exclusive OR) operation.
   - * Note: in this stage, totally $(4N+2)$ chaotic bits are required: $b(0) \sim b(4N-1), b(4N+1), b(4N+3)$.

   * Note: in the whole encryption procedure, totally $\max(N_g \cdot S_g \cdot L + \Delta_L, 4N+2)$ chaotic bits are required. Since the index of the last required chaotic bit is $\max(N_g \cdot S_g \cdot L + \Delta_L - 1, 4N+3)$, one has $L_b = \max(N_g \cdot S_g \cdot L + \Delta_L, 4N+4)$.

- *The decryption procedure* is the reversion of the encryption procedure, and can be briefly described as follows:
  1. *The inverse intra-frame decryption (bit swapping and masking operations)*:
     - *mask the 4 odd bits of the cipher-byte $g'(i) = \sum_{k=0}^{7} d_k'(i) \cdot 2^k$ to restore the intermediate byte $g^{**}(i)$*;
     - *pseudo-randomly swap the 4 MSB-s and the 4 LSB-s of each byte $g^{**}(i) = \sum_{k=0}^{7} d_k^{**}(i) \cdot 2^k$ to restore another intermediate byte $g^*(i) = \sum_{k=0}^{7} d_k^*(i) \cdot 2^k$.*
  2. *The same inter-frame interleaving* is inversely exerted on $g^* = \{g^*(i)\}_{i=0}^{N-1}$ to restore the plaintext $g = \{g(i)\}_{i=0}^{N-1}$, where the term "inversely" means that $i = 0 \sim (N_g - 1)$ and $j = (S_g - 1) \sim 0$.

4 Cryptanalysis of HDSP

4.1 The brute-force attack

The brute-force attack is the attack of exhaustively searching the secret key from the set of all possible keys [4], which can be used in different attacking scenarios including ciphertext-only and known-plaintext attacks. Apparently, the computational complexity of the brute-force attack is determined by the size of the key space and the complexity of verifying each key. The secret key in HDSP is $(\mu, x(0))$, which is represented by $2 \cdot 16 = 32$ secret bits. Thus, the size of the key space is 2^{32}. Since the complexity of verifying each key is equal to the one in the encryption procedure of HDSP – $O(N)$ [3, Sec. 4.1], the total complexity of the brute-force attack is $O\left(N \times 2^{32}\right)$. However, because not all values of μ can ensure the chaoticity of the Logistic map [5], actually the size of the key space is smaller than 2^{32} and the attack complexity is smaller than $O\left(N \times 2^{32}\right)$.

Here, the following facts should be noted: 1) in the known-plaintext attack scenario, the key can be validated by simply comparing the decrypted signal with the known plaintext; 2) in the ciphertext-only attack scenario, the key has to be validated by some inherent features of the plaintexts, which may increase the attack complexity to some extent, but generally the complexity will not be larger than $O\left(N^2 \times 2^{32}\right)$.

To guarantee a high-level security in today's digital world, a complexity of order $O\left(2^{100}\right)$ is required [4]. Apparently, $O\left(N \cdot 2^{32}\right)$ is too small to reach such a goal. An attacker can easily find the secret key within a few hours (or even minutes) using a PC with a 1G CPU, if N is not large enough (when N is too large, it is easy for one to check only a small segment of the plaintext). To ensure a high-level security, 64-bit binary representations of μ and $x(0)$ are suggested to provide a complexity of order $O\left(2^{128}\right)$ against the brute-force attack.

4.2 The known-plaintext attack

In [3, Sec. 4.2], it was claimed that HDSP can efficiently resist the known-plaintext attack[3]. Here, this claim is re-evaluated, concluding that HDSP is not sufficiently secure against the known-plaintext attack. Basically, with n known plaintexts, from the probabilistic point of view, $\left(100-\frac{50}{2^n}\right)$ percent of chaotic bits can be correctly restored and only $\frac{16}{2^n}$ bits need to be exhaustively guessed to break the secret key[4]. Even with $n=1$, the complexity of breaking the secret key is sufficiently small: only $O\left(2^8\right)$. Once the secret key is uncovered, the HDSP scheme is completely broken. Furthermore, because HDSP works like a stream cipher, even without deriving the secret key, one can still use the partially-reconstructed chaotic bit sequence to partially (almost completely when n is relatively large) recover the unknown plaintexts.

Similar to the encryption procedure of HDSP, the known-plaintext attack also works in two steps: 1) break the inter-frame interleaving; 2) break the intra-frame encryption. The two steps provide a partial (or total) reconstruction of the secret chaotic bit sequence, which can then be used to restore the secret key. Generally, if N is sufficiently large, only one known plaintext is enough for an attacker to restore the sub-key μ and an equivalent of another sub-key $x(0)$. In addition, if two or more plaintexts of size N are known, it is possible to correctly restore most chaotic bits, which can also be directly used as a replacement of the secret key to decrypt the ciphertexts encrypted with the same key (if their sizes are not larger than N).

Next, some important properties of the intra-frame encryption step of HDSP are discussed, which are the essential reasons for the insecurity of HDSP against known/chosen-plaintext attacks. These insecure properties are structural defects of the encryption procedure used in HDSP, independent of any specifications of the speech data and the speech CODEC. The theoretical analysis of the proposed attacks will be based on the assumption that the plaintext (i.e., the output of the speech CODEC) is a uniformly-distributed random source. When the plaintext does not obey the uniform distribution, our experiments show that the performance of the proposed attacks may be somewhat better or worse (see Sec. 5).

4.2.1 Some insecure properties of the HDSP encryption scheme

In the intra-frame encryption step of HDSP, the i-th byte of the input signal, $g^*(i)=\sum_{k=0}^{7} d_k^*(i)\cdot 2^k$, and the i-th byte of the output signal (i.e., the ciphertext), $g'(i)=\sum_{k=0}^{7} d'_k(i)\cdot 2^k$, satisfy the following properties.

Property 1 *For $k=0,2$: a) $d_k^*(i)+d_{k+4}^*(i)\equiv d'_k(i)+d'_{k+4}(i)$; b) when $d_k^*(i)\neq d_{k+4}^*(i)$, one has $b(4i+k)=d_k^*(i)\oplus d'_k(i)=d_{k+4}^*(i)\oplus d'_{k+4}(i)$.*

Proof: From Eqs. (2) and (3), one can see that the 4 even bits are swapped without being masked. Thus, for $k=0,2$, $d_k^*(i)+d_{k+4}^*(i)$ remains unchanged after the intra-frame encryption, i.e., $d_k^*(i)+d_{k+4}^*(i)\equiv d'_k(i)+d'_{k+4}(i)$.

When $d_k^*(i)\neq d_{k+4}^*(i)$, which means that $d_k^*(i)=\overline{d_{k+4}^*(i)}$ and $d_{k+4}^*(i)=\overline{d_k^*(i)}$, the bit swapping operation $Swap_w(a,b)$ becomes (in this case, $a\neq b$)

$$Swap_w(a,b)=\begin{cases}(a,b)=(a\oplus 0,b\oplus 0)=(a\oplus w,b\oplus w), & w=0,\\ (b,a)=(\bar{a},\bar{b})=(a\oplus 1,b\oplus 1)=(a\oplus w,b\oplus w), & w=1.\end{cases} \tag{4}$$

That is, $Swap_w(a,b)\equiv(a\oplus w,b\oplus w)$. Then, one has $d'_k(i)=d_k^*(i)\oplus b(4i+k)$ and $d'_{k+4}(i)=d_{k+4}^*(i)\oplus b(4i+k)$, which immediately leads to $b(4i+k)=d_k^*(i)\oplus d'_k(i)=d_{k+4}^*(i)\oplus d'_{k+4}(i)$. Thus, the proof is completed. ■

Property 2 *For $k=1,3$: a) when $d_k^*(i)=d_{k+4}^*(i)$, one has $b(4i+k)=d_k^*(i)\oplus d'_k(i)$ and $b(4(i+1)+k)=d_{k+4}^*(i)\oplus d'_{k+4}(i)$; b) when $d_k^*(i)\neq d_{k+4}^*(i)$, one has $d'_k(i)\equiv d_k^*(i)$ and $b(4i+k)\oplus b(4(i+1)+k)=d_{k+4}^*(i)\oplus d'_{k+4}(i)$.*

Proof: The two conditions are proved separately.

a) When $d_k^*(i)=d_{k+4}^*(i)$, the bit swapping operation disappears and the cipher-bit is determined by the masking operation only: $d'_k(i)=d_k^*(i)\oplus b(4i+k)$ and $d'_{k+4}(i)=d_{k+4}^*(i)\oplus b(4i+k+4)$. That is, $b(4i+k)=d_k^*(i)\oplus d'_k(i)$ and $b(4(i+1)+k)=d_{k+4}^*(i)\oplus d'_{k+4}(i)$.

[3]As introduced in Sec. 2, the known-plaintext attack becomes possible when an attacker can temporarily access the encryption machine. Under such a condition, it is reasonable to assume that the attacker can get all plaintext outputs from the speech CODEC, since details of the CODEC are not kept secret in the HDSP-based VoIP system.

[4]To break the secret key, one need to get 32 consecutive chaotic bits to recover two consecutive chaotic states (see Sec. 4.2.4). Since only $\frac{50}{2^n}$ percent of chaotic bits cannot be uniquely derived, one has to exhaustively search $32\cdot\frac{50}{2^n}\%=\frac{16}{2^n}$ bits. Note that this figure is meaningful only in a probabilistic sense. As discussed later in Sec. 4.2.4, in real attacks, it is possible to exhaustively search less bits.

b) When $d_k^*(i) \neq d_{k+4}^*(i)$, from Eq. (4), $Swap_w(a,b) \equiv (a \oplus w, b \oplus w)$, so one can get $d_k^{**}(i) = d_k^*(i) \oplus b(4i+k)$ and $d_{k+4}^{**}(i) = d_{k+4}^*(i) \oplus b(4i+k)$. Substituting the two results into Eq. (3), one has the following results:

- $d_k'(i) = d_k^{**}(i) \oplus b(4i+k) = (d_k^*(i) \oplus b(4i+k)) \oplus b(4i+k) \equiv d_k^*(i)$;
- $d_{k+4}'(i) = d_{k+4}^{**}(i) \oplus b(4i+k+4) = \left(d_{k+4}^*(i) \oplus b(4i+k)\right) \oplus b(4i+k+4) = d_{k+4}^*(i) \oplus (b(4i+k) \oplus b(4i+k+4))$, which is equivalent to $b(4i+k) \oplus b(4(i+1)+k) = d_{k+4}^*(i) \oplus d_{k+4}'(i)$.

From the above two conditions, the property is proved. ■

Property 3 *For* $k = 1, 3$, $\forall\ i = 0 \sim N-2$, *if* $d_k^*(i) = d_{k+4}^*(i)$ *and* $d_k^*(i+1) = d_{k+4}^*(i+1)$, *then* $b(4(i+1)+k) = d_{k+4}^*(i) \oplus d_{k+4}'(i) = d_k^*(i+1) \oplus d_k'(i+1)$.

Proof: This property is a direct corollary of the above Property 2a. ■

In the following, it will be shown that the above properties make the known-plaintext attack feasible in practice.

4.2.2 Breaking the inter-frame interleaving

The frame swapping operations in the inter-frame interleaving step actually correspond to a pseudo-random and secret frame-permutation in each group. One can represent the permutation of $group(i)$ by a permutation vector $\boldsymbol{v}(i) = [v(i,0), \cdots, v(i, S_g - 1)]$, where $\forall\ j_1 \neq j_2$, $v(i, j_1) \neq v(i, j_2)$. With the permutation vector, the inter-frame interleaving of $group(i)$ can be described as follows: $\forall\ j = 0 \sim (S_g - 1)$, the j-th frame in $group(i)$ is permuted to be the $v(i,j)$-th frame, i.e., $frame(i \cdot S_g + j)$ is permuted to be $frame(i \cdot S_g + v(i,j))$. Basically, in this restoration stage, the goal is to restore the permutation vectors of all groups: $\boldsymbol{v}(0) \sim \boldsymbol{v}(N_g - 1)$.

To restore the permutation vectors, at least an input signal (i.e., the plaintext g) and the corresponding output signal (i.e., g^*) should be known. However, in the known-plaintext attack, generally the intermediate signal g^* is not known. Fortunately, due to Property 1a proved above, some information of g^* can be obtained from the ciphertext g'. This generally is enough to restore the permutation vectors.

Next, define three sequences, $\widehat{g} = \{\widehat{g}(i)\}_{i=0}^{N-1}$, $\widehat{g}^* = \{\widehat{g}^*(i)\}_{i=0}^{N-1}$ and $\widehat{g}' = \{\widehat{g}'(i)\}_{i=0}^{N-1}$, where

$$\begin{aligned} \widehat{g}(i) &= (d_0(i) + d_4(i)) + (d_2(i) + d_6(i)) \times 3 \in \{0, \cdots, 8\}, && (5)\\ \widehat{g}^*(i) &= (d_0^*(i) + d_4^*(i)) + (d_2^*(i) + d_6^*(i)) \times 3 \in \{0, \cdots, 8\}, && (6)\\ \widehat{g}'(i) &= (d_0'(i) + d_4'(i)) + (d_2'(i) + d_6'(i)) \times 3 \in \{0, \cdots, 8\}. && (7) \end{aligned}$$

From Property 1a, one can see that $\widehat{g}^* = \widehat{g}'$. Considering that the inter-frame interleaving stage does not change the values of all frames but their positions, one can use $\widehat{g}$ and $\widehat{g}'$ to restore the permutation vectors. To do so, one has to divide both $\widehat{g}$ and $\widehat{g}'$ into N_f frames, $\left\{\widehat{frame}(i)\right\}_{i=0}^{N_f-1}$, $\left\{\widehat{frame}'(i)\right\}_{i=0}^{N_f-1}$, and N_g groups, $\left\{\widehat{group}(i)\right\}_{i=0}^{N_g-1}$, $\left\{\widehat{group}'(i)\right\}_{i=0}^{N_g-1}$, respectively, in the same way as the encryption procedure of HDSP. Now, $\forall\ i = 0 \sim (N_g - 1)$, the permutation vector of $group(i)$ can be estimated as follows[5]:

- *Step 1*: For $j = 0 \sim (S_g - 1)$, calculate $R_f(i,j) = \sum_{k=0}^{S_f-1} \widehat{g}_f(i,j,k) \times 9^k$ and $R_f'(i,j) = \sum_{k=0}^{S_f-1} \widehat{g}_f'(i,j,k) \times 9^k$, where $\widehat{g}_f(i,j,k)$ and $\widehat{g}_f'(i,j,k)$ denote the k-th byte of the j-th frame in $\widehat{group}(i)$ and in $\widehat{group}'(i)$, respectively.
- *Step 2*: Compare the values of $R_f(i,0) \sim R_f(i, S_g - 1)$ and $R_f'(i,0) \sim R_f'(i, S_g - 1)$ to get two partitions of the index-set $\mathbb{S}_g = \{0, \cdots, S_g - 1\}$: $\{\Lambda(k)\}_{k=0}^{K-1}$ and $\{\Lambda'(k)\}_{k=0}^{K-1}$, where K is the number of different values in the set $\{R_f(i,0), \cdots, R_f(i, S_g - 1)\} = \{R_f'(i,0), \cdots, R_f'(i, S_g - 1)\}$ and $\forall\ a, b \in \Lambda(k), \forall\ a', b' \in \Lambda'(k)$, $R_f(i,a) = R_f(i,b) = R_f'(i,a') = R_f'(i,b')$.

  * Note 1: $\{\Lambda(k)\}_{k=0}^{K-1}$ and $\{\Lambda'(k)\}_{k=0}^{K-1}$ are partitions of $\mathbb{S}_g$, which means that $\bigcup_{k=0}^{K-1} \Lambda(k) = \bigcup_{k=0}^{K-1} \Lambda'(k) = \mathbb{S}_g$ and $\forall\ k_1 \neq k_2$, $\Lambda(k_1) \cap \Lambda(k_2) = \Lambda'(k_1) \cap \Lambda'(k_2) = \varnothing$.

[5]A more general description of this algorithm for breaking secret permutations with a number of known/chosen plaintexts can be found in [7], where some common permutation-only image ciphers are analyzed.

  * Note 2: because the frame swapping operations do not change the value of $R_f(i,j)$, it is obvious that $\forall\, k = 0 \sim (K-1)$, the cardinality (size) of $\Lambda(k)$ is equal to that of $\Lambda'(k)$, i.e., $\#(\Lambda(k)) = \#(\Lambda'(k))$.

- *Step 3*: Derive an estimation of the permutation vector $\boldsymbol{v}(i)$ of *group*(i) from $\{\Lambda(k)\}_{k=0}^{K-1}$ and $\{\Lambda'(k)\}_{k=0}^{K-1}$, under the following two conditions:
  - *Condition 1*: if $\forall\, k \in \{0,\cdots,K-1\}$, $\Lambda(k)$ contains only one element, i.e., $\#(\Lambda(k)) = \#(\Lambda'(k)) = 1$ (and $K = S_g$), then the permutation vector $\boldsymbol{v}(i)$ of *group*(i) can be *uniquely* derived: $\forall\, k = 0 \sim (K-1 = S_g - 1)$, and $v(i,\Lambda(k))$ is set to be the only element in $\Lambda'(k)$.
  - *Condition 2*: if $\exists k \in \{0,\cdots,K-1\}$, $\Lambda(k)$ contains more than one elements, i.e., $\#(\Lambda(k)) = \#(\Lambda'(k)) \geq 2$ (and $K < S_g$), then the permutation vector $\boldsymbol{v}(i)$ cannot be uniquely derived from $\{\Lambda(k)\}_{k=0}^{K-1}$ and $\{\Lambda'(k)\}_{k=0}^{K-1}$. But one can get an estimated permutation vector $\widetilde{\boldsymbol{v}}(i) = [\widetilde{v}(i,0),\cdots,\widetilde{v}\,(i,S_g-1)]$ as follows: for $k = 0 \sim (K-1)$, determine a one-to-one mapping $f_{\Lambda(k)} : \Lambda(k) \to \Lambda'(k)$, and then $\forall\, a \in \Lambda(k)$, set $\widetilde{v}(i,a) = f_{\Lambda(k)}(a)$.

Under *Condition 1* in Step 3, the permutation vector can be correctly derived without any error. However, under *Condition 2*, the estimated permutation vector $\widetilde{\boldsymbol{v}}$ may be wrong with a non-negligible probability. This is due to the following fact: $\forall\, k = 0 \sim (K-1)$, there are $(\#(\Lambda(k)))!$ possible mappings of $f_{\Lambda(k)}$, so there are $\prod_{k=0}^{K-1}(\#(\Lambda(k)))!$ possible estimations of $\boldsymbol{v}(i)$, among which only one is the correct permutation vector $\boldsymbol{v}$.

Now, let us study the occurrence probability of *Condition 2*. Assuming the number of all possible values of $R_f(i,j)$ is N_{R_f}, this probability can be easily calculated as follows:

$$Prob[\text{Condition 2 occurs}] = 1 - \left(1 - \frac{0}{N_{R_f}}\right)\cdot\left(1 - \frac{1}{N_{R_f}}\right)\cdots\left(1 - \frac{S_g - 1}{N_{R_f}}\right). \tag{8}$$

From the definition of R_f, one has $N_{R_f} = 9^{S_f}$. In most cases, 9^{S_f} is much larger than S_g, so the occurrence probability of *Condition 2* is so small that it can be simply ignored in practice. When *Condition 2* cannot be ignored (i.e., when 9^{S_f} is not much larger than S_g), the following constraints in the intra-frame encryption stage can be used to detect wrong estimations:

- *Constraint 1*: $k = 1,3$, $\forall\, i = 0 \sim (N-1)$, if $d_k^*(i) \neq d_{k+4}^*(i)$, $d_k'(i) = d_k^*(i)$.
- *Constraint 2*: $k = 1,3$, $\forall\, i = 0 \sim (N-2)$, if $d_k^*(i) = d_{k+4}^*(i)$ and $d_k^*(i+1) = d_{k+4}^*(i+1)$, then $d_{k+4}^*(i) \oplus d_{k+4}'(i) = d_k^*(i+1) \oplus d_k'(i+1) = b(4(i+1)+k)$.
- *Constraint 3*: As shown later in the next sub-subsection, the two chaotic bit sequences $\{b(4i+1)\}_{i=0}^{N}$ and $\{b(4i+3)\}_{i=0}^{N}$ are completely correlated, i.e., they satisfy Eq. (9) below. It will be precisely explained there as how to use this constraint to detect wrong permutation vectors.

The above three constraints can be deduced from Properties 2 and 3. Once any of these constraints is violated, it can be immediately asserted that the current permutation vector is wrong.

Finally, the following fact should be noticed: in *Condition 2*, the larger the number of the known plaintexts is, the less the probability of $\widetilde{\boldsymbol{v}}(i) \neq \boldsymbol{v}(i)$ will be. Given n known plaintexts $g_1 \sim g_n$, the number of all possible combinations of $R_{f,1}(i,j) \sim R_{f,n}(i,j)$ becomes $N_{R_f}^n = 9^{nS_f}$, which means that the probability that *Condition 2* is satisfied exponentially decrease as n increases. That is, the probability of getting a wrong permutation vector will be exponentially decrease, so it is negligible when n is sufficiently large[6].

4.2.3 Breaking the intra-frame encryption

Once the inter-frame interleaving is correctly broken, one can successfully get the intermediate signal $g^* = \{g^*(i)\}_{i=0}^{N-1}$ from the plaintext g. With g^* and the ciphertext g', the intra-frame encryption can also be partially (or even totally) broken. During this breaking stage, one can partially (or even totally) reconstruct the secret chaotic bit sequence, which can be used to further derive the secret key. The following properties, summarized above, are now available for an attacker to break the intra-frame encryption:

- *Property 1b*: $k = 0,2$, when $d_k^*(i) \neq d_{k+4}^*(i)$, $b(4i+k) = d_k^*(i) \oplus d_k'(i) = d_{k+4}^*(i) \oplus d_{k+4}'(i)$;

[6]Generally speaking, "n is sufficiently large" if $n \gg (\log_9 S_g)/S_f$, which ensures that $9^{nS_f} \gg S_g$.

- *Property 2a*: $k = 1, 3$, when $d_k^*(i) = d_{k+4}^*(i)$, $b(4i + k) = d_k^*(i) \oplus d_k'(i)$ and $b(4(i + 1) + k) = d_{k+4}^*(i) \oplus d_{k+4}'(i)$;
- *Property 2b*: $k = 1, 3$, when $d_k^*(i) \neq d_{k+4}^*(i)$, $b(4i + k) \oplus b(4(i + 1) + k) = d_{k+4}^*(i) \oplus d_{k+4}'(i)$.

Based on the above three properties, one can estimate how many chaotic bits may be obtained in a known-plaintext attack. Without loss of generality, assume that each bit in $g^*(i)$ and $g'(i)$ distributes uniformly over $\{0, 1\}$ and any two bits are independent of each other. Under this assumption, one can deduce the following results.

- *Even bits ($k = 0, 2$, from Property 1b)*:
  - *with only one known plaintext*: the probability of $d_k^*(i) \neq d_{k+4}^*(i)$ is $\frac{1}{2}$, so averagely 50% of all even chaotic bits can be correctly restored;
  - *with $n > 1$ known plaintexts*: when $d_k^*(i) \neq d_{k+4}^*(i)$ holds for at least one plaintext, the bit $b(4i + k)$ can be correctly restored, and then it can be deduced that the probability of the above event is $1 - \left(\frac{1}{2}\right)^n$.
- *Odd bits ($k = 1, 3$, from Property 2a/b)*:
  - *with only one known plaintext*: $\forall\ i = 0 \sim (N - 1)$, the value of $b(4i + k) \oplus b(4(i + 1) + k)$ can be correctly determined, i.e., one can get a new sequence $\left\{b_k^{\oplus}(i) = b(4i + k) \oplus b(4(i + 1) + k)\right\}_{i=0}^{N-1}$. Apparently, if only one bit $b(4i^* + k)$ is known, the whole bit sequence $\{b(4i + k)\}_{i=0}^{N}$ can be correctly restored with the deterministic sequence $\{b_k^{\oplus}(i)\}_{i=0}^{N-1}$ as follows:

$$b(4i^* + k) \begin{cases} \xrightarrow{\oplus b_k^{\oplus}(i^*-1)} b(4(i^* - 1) + k) \xrightarrow{\oplus b_k^{\oplus}(i^*-2)} \cdots \xrightarrow{\oplus b_k^{\oplus}(0)} b(4 \cdot 0 + k), \\ \xrightarrow{\oplus b_k^{\oplus}(i^*)} b(4(i^* + 1) + k) \xrightarrow{\oplus b_k^{\oplus}(i^*+1)} \cdots \xrightarrow{\oplus b_k^{\oplus}(N-1)} b(4N + k). \end{cases} \tag{9}$$

    From Property 2a above, two bits $b(4i + k)$ and $b(4(i + 1) + k)$ can be correctly restored when $d_k^*(i) = d_{k+4}^*(i)$. Thus, it can be deduced that the probability that at least two bits in $\{b(4i+k)\}_{i=0}^{N}$ are correctly restored is

$$1 - \left(Prob[d_k^*(i) \neq d_{k+4}^*(i)]\right)^N = 1 - \frac{1}{2^N}.$$

    Since N is generally sufficiently large, it is probabilistically true in almost all cases that all odd bits can be correctly restored.

    * Note 1: Even under an extreme condition where no intermediate byte $g^*(i)$ satisfying $d_k^*(i) = d_{k+4}^*(i)$, one can randomly guess the value of any bit $b(4i + k)$ and then get the whole bit sequence $\{b(4i + k)\}_{i=0}^{N}$. In this case, at most four guesses (two for $k = 1$ and the other two for $k = 3$) are needed to correctly restore all odd bits. In this sense, all odd bits can always be correctly reconstructed.
    * Note 2 (a precise explanation of *Constraint 3*): Assume that two bytes $g^*(i_1)$ and $g^*(i_2)$ satisfy $d_k^*(i_1) = d_{k+4}^*(i_1)$ and $d_k^*(i_2) = d_{k+4}^*(i_2)$ and all in-between bytes $g^*(i_1 + 1) \sim g^*(i_2 - 1)$ do not satisfy this condition, where $i_2 \geq i_1 + 2$. Then, the sub-sequence $\{b(4i + k\}_{i=i_1+1}^{i_2-1}$ can be uniquely derived by using Eq. (9) twice:

$$\begin{aligned} b(4i_1 + k) \Rightarrow \quad & b(4(i_1 + 1) + k) \rightarrow \cdots \rightarrow b(4(i_2 - 1) + k), \\ & b(4(i_1 + 1) + k) \leftarrow \cdots \leftarrow b(4(i_2 - 1) + k) \quad \Leftarrow b(4i_2 + k). \end{aligned}$$

      If the two derived sub-sequences are not identical, it can be asserted that at least one permutation of the $frame(s)$ between $g(i_1)$ and $g(i_2)$ is wrong, i.e., at least one permutation vector of the *group*(*s*) between $g(i_1)$ and $g(i_2)$ is wrong, and, when there is only one group between $g(i_1)$ and $g(i_2)$, the permutation vector of this group must be wrong.
  - *with $n > 1$ known plaintexts*: the probability that at least two bits in $\{b(4i + k)\}_{i=0}^{N}$ can be correctly restored is $\left(1 - \dfrac{1}{2^{n \cdot N}}\right)$.

As a summary, when only one plaintext is known, averagely 50% of even bits and all odd bits, i.e., 75% of all bits in $\{b(i)\}_{i=0}^{4N-1}$ and the last two bits $b(4N+1), b(4N+3)$, can be correctly restored; furthermore, when $n \geq 1$ plaintexts are known, the percentage of correctly restored bits in $\{b(i)\}_{i=0}^{4N-1}$ becomes

$$\left(50 + 25 + \frac{25}{2} + \cdots + \frac{25}{2^{n-1}}\right)\% = \left(100 - \frac{50}{2^n}\right)\%,$$

which exponentially approaches 100% as n increases. For the rest unrecovered chaotic bits, one can randomly guess their values, and averagely half of the bits can be correctly matched to the true values. That is, the correctly restored bits in $\{b(i)\}_{i=0}^{4N-1}$ will reach $\left(100 - \frac{50}{2^{n+1}}\right)\%$, which is 87.5% for $n = 1$.

In addition, if some permutation vectors have been uniquely determined in the stage of breaking the inter-frame interleaving, the corresponding guessed bits can be checked with Eq. (1), and the correctly restored bits can be even more. In addition, since the wrong bits distribute randomly within the whole bit sequence and since human ears have a high capability to bear large audio noises, their negative influence on the quality of the voice data may not be so much, as verified by our experiments to be discussed later.

Finally, let us study the success probability of the decryption with the partially reconstructed chaotic bit sequence $b(0) \sim b(4N-1), b(4N+1), b(4N+3)$, when only one plaintext is known (i.e., $n = 1$). Since the two even bits $b(4i), b(4i+2)$, corresponding to a plain-byte $g(i)$, are correctly restored with probability $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$, it seems that the probability of correctly decrypting an unknown plain-byte $\widetilde{g}(i) = \sum_{k=0}^{7} \widetilde{d}_k(i) \cdot 2^k$ should also be $\frac{1}{4}$. However, such an "intuition" is not true, because this probability is actually the addition of the following four probabilities:

- the probability that $b(4i), b(4i+2)$ are both correct, which is $\frac{1}{4}$;
- the probability that $b(4i)$ is correct, but $b(4i+2)$ is incorrect, and $\widetilde{d}_2^*(i) = \widetilde{d}_6^*(i)$, which is $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$;
- the probability that $b(4i)$ is incorrect, but $b(4i+2)$ is correct, and $\widetilde{d}_0^*(i) = \widetilde{d}_4^*(i)$, which is also $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$;
- the probability that $b(4i), b(4i+2)$ are both incorrect, $\widetilde{d}_0^*(i) = \widetilde{d}_4^*(i)$ and $\widetilde{d}_2^*(i) = \widetilde{d}_6^*(i)$, which is $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{16}$.

So, the final probability of correctly decrypting a plain-byte is $\frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{16} = \frac{9}{16}$. In a similar way, this probability with $n (\geq 1)$ known plaintexts can be calculated as

$$\begin{aligned} P(n) &= \left(1 - \frac{1}{2^n}\right)^2 + \left(1 - \frac{1}{2^n}\right) \cdot \frac{1}{2^n} \cdot \frac{1}{2} + \frac{1}{2^n} \cdot \left(1 - \frac{1}{2^n}\right) \cdot \frac{1}{2} + \frac{1}{2^n} \cdot \frac{1}{2^n} \cdot \frac{1}{2} \cdot \frac{1}{2} \\ &= 1 - \frac{1}{2^n} + \frac{1}{2^{2n+2}} = 1 - \frac{2}{2^{n+1}} + \left(\frac{1}{2^{n+1}}\right)^2 = \left(1 - \frac{1}{2^{n+1}}\right)^2. \end{aligned}$$

Since each undetermined chaotic bit is identical with the original bit with probability $\frac{1}{2}$, one has

- $Prob[b(4i) \text{ is incorrect}] = Prob[b(4i+2) \text{ is incorrect}] = \frac{1}{2} \cdot \frac{1}{2^n} = \frac{1}{2^{n+1}}$;
- $Prob[b(4i) \text{ is correct}] = Prob[b(4i+2) \text{ is correct}] = 1 - \frac{1}{2^{n+1}}$.

So, the probability of correctly decrypting a plain-byte in real attacks will become $P'(n) = P(n+1) = \left(1 - \frac{1}{2^{n+2}}\right)^2$. This result well agrees with our experiments (see Table 2 below).

4.2.4 Completely breaking the encryption scheme (breaking the secret key)

With the reconstructed permutation vectors $\boldsymbol{v}(0) \sim \boldsymbol{v}(N_g - 1)$ and the partially restored chaotic bit sequence $b(0) \sim b(4N-1), b(4N+1), b(4N+3)$, a ciphertext g' can be decrypted to get an estimated plaintext $\widetilde{g} = \{\widetilde{g}(i)\}_{i=0}^{N-1}$ as follows:

- use the partially reconstructed chaotic bits to cancel the intra-frame encryption;
- derive the inverse permutation vector of each interleaving group and use it to cancel the inter-frame interleaving.

In the previous sub-section, it was shown that the breaking performance is not satisfactory because about 25% of plain-bytes cannot be correctly decrypted when only one plaintext is known. Although the performance can be exponentially enhanced with more known plaintexts, the above procedure cannot decrypt any plain-byte beyond the position N. Therefore, to thoroughly break the HDSP encryption scheme, one has to break the secret key itself. In the following, we show how to break the secret key from some consequent partially-reconstructed chaotic bits.

a) Breaking the initial condition $x(0)$ or an equivalent $x(i)$

Recalling the generation of chaotic bits in the initialization procedure of HDSP, one can see that each chaotic state $x(i)$ can be represented in the binary form as $0.b(16i+0)\cdots b(16i+15)$. Thus, if the first 16 chaotic bits are all correctly restored, one can directly get the initial condition $x(0) = 0.b(0)\cdots b(15)$. However, following the discussion in the previous sub-subsection, generally not all even bits in $b(0) \sim b(15)$ can be uniquely determined. To restore these bits, one has to exhaustively guess their values. Assuming the number of undetermined bits is $m \in \{0, \cdots, 8\}$, the guessing complexity is $O(2^m)$. Averagely, $m = \frac{1}{4} \cdot 16 = 4$, and the searching complexity is $O(2^4)$, which is practically small for PC-s. Even under the worst condition, $m = 8$, the guessing complexity is only $O(2^8)$.

When $x(0)$ is enhanced to be represented with $B > 16$ bits and all the $B' \leq B$ bits are extracted to generate the chaotic bit sequence $\{b(i)\}$, the guessing complexity under the worst condition will be $O\left(2^{\frac{B'}{2}} \cdot 2^{B-B'}\right) = O\left(2^{B-\frac{B'}{2}}\right)$. If $\left(B - \frac{B'}{2}\right)$ is sufficiently large[7], the guessing complexity will be too large for the exhaustive guess of $x(0)$ on PC-s. In this case, instead of guessing $x(0)$, one can try to find another chaotic state $x(i) = 0.b(i \cdot B') \cdots b(i \cdot B' + (B'-1))$ as an equivalent of the sub-key $x(0)$, where the number of undetermined bits is less than $m < \frac{B'}{4}$, to get a reduced guessing complexity not greater than $O\left(2^{B-B'+m}\right) < O\left(2^{B-\frac{3B'}{4}}\right)$. With $x(i)$ and the derived μ (see below), one can exactly reconstruct all chaotic bits beyond the position $(i \cdot B')$ and then restore all plain-bytes starting from the first group located after the plain-byte that is encrypted by the $(i \cdot B')$-th chaotic bit. Apparently, for the original HDSP with $B = B' = 16$, such an idea of reducing the complexity is also feasible (but somewhat meaningless). Of course, using this method, the guessing complexity has a lower bound $O\left(2^{B-B'}\right)$, which is the complexity of exhaustively guessing the $(B - B')$ bits that do not occur in the chaotic bit sequence.

b) Breaking the control parameter μ

Given 32 consequent chaotic bits $b(16i+0) \sim b(16i+31)$, two consecutive chaotic states can be determined: $x(i) = 0.b(16i)\cdots b(16i+15)$ and $x(i+1) = 0.b(16i+16)\cdots b(16i+31)$, and then an estimated value of the secret sub-key μ can be derived as $\widetilde{\mu} = \frac{x(i+1)}{x(i)\cdot(1-x(i))}$. Due to the quantization errors existing in the finite-precision environment, generally $\widetilde{\mu} \neq \mu$. Fortunately, following the error analysis of $\widetilde{\mu}$ given in Sec. 3.2 of [8], it has been shown that when $x(i+1) \geq 2^{-n}$ ($n = 1 \sim 16$), one has $|\widetilde{\mu} - \mu| \leq 2^{n+3} \cdot 2^{-16}$. For example, when $x(i+1) \geq 2^{-1} = 0.5$, one can exhaustively search $2^4 = 16$ values in the neighborhood of $\widetilde{\mu}$ to find the right value of μ. All correctly-restored chaotic bits after $b(16i+31)$ can be used to verify whether or not a searched value of μ is right.

As discussed above, generally there are undetermined bits in $b(16i+0) \sim b(16i+31)$. The average number of such bits is $\frac{1}{4} \cdot 32 = 8$, and the maximal number is 16. This means that one has to exhaustively search all values of the undetermined bits to calculate a number of different values of $\widetilde{\mu}$. One can see that the complexity will be about $O(2^8)$ in average and be $O(2^{16})$ in the worst condition. To further reduce the searching complexity, one can try to find two consecutive chaotic states that contain less than $m < 8$ undetermined bits[8]. The occurrence probability of such an event is $p(m) = \sum_{i=0}^{m} \binom{16}{i} \left(\frac{1}{2}\right)^{16}$, and the average position of its first occurrence is $1/p(m) = \frac{2^{16}}{\sum_{i=0}^{m} \binom{16}{i}}$. In Table 1, the values of $p(m)$ and $1/p(m)$ for $m = 0 \sim 8$ are shown. If the plaintext does not have a uniform distribution, the probability of $d_k^*(i) \neq d_{k+4}^*(i)$ may not be $\frac{1}{2}$, and so the probability $p(m)$ may be different from the theoretical value (compare Tables 1 and 3).

[7]For example, when $B = 80$, $B' = 16$, one has $B - \frac{B'}{2} = 72$, which can be considered to be sufficiently large.

[8]Similar to the condition of $x(0)$, when $B > 16$ bits are used to represent chaotic states, this will become very useful to reduce the total complexity.

Table 1: The occurrence probability of 32 consequent bits with less than $m \leq 8$ undetermined bits, and the average position of the first occurrence of such bits in the chaotic bit sequence.

m	0	1	2	3	4	5	6	7	8
$p(m) \approx$	0.0000153	0.000259	0.00209	0.0106	0.0384	0.105	0.227	0.402	0.598
$\lceil 1/p(m) \rceil$	65536	3856	479	95	27	10	5	3	2

4.3 The chosen-plaintext attack

In the above-mentioned known-plaintext attack, the existence of the undetermined even bits in the chaotic sequence are due to the fact that for 25% of plain-bytes, $d_0(i) = d_4(i)$ or $d_2(i) = d_6(i)$, i.e., $d_0^*(i) = d_4^*(i)$ or $d_2^*(i) = d_6^*(i)$ holds for the corresponding intermediate bytes. In the chosen-plaintext attack, one can create a plaintext $g = \{g(i)\}_{i=0}^{N-1}$ as follows: $\forall\ i = 0 \sim (N-1)$, $d_0(i) \neq d_4(i)$ and $d_2(i) \neq d_6(i)$. With such a plaintext and its ciphertext g', all chaotic bits can be uniquely determined. Thus, the secret sub-key $x(0) = 0.b(0) \cdots b(15)$ will be accurate, and another sub-key μ can be exactly derived from any two consecutive chaotic states $x(i)$ and $x(i+1)$. The complexity of deriving μ is the complexity of searching over the neighborhood of $\widetilde{\mu} = \frac{x(i+1)}{x(i)\cdot(1-x(i))}$. By selecting $x(i+1)$ as the first chaotic state that is not less than 0.5, the search complexity will be minimized. As a result, one can see that HDSP is not secure at all against the chosen-plaintext attack.

5 Experiments

In this section, some experiments are shown to support the theoretical analysis on the known-plaintext attack[9]. The parameters used in the experiments are $N = 65536$, $S_f = 32$, $S_g = 16$, and the secret key is $x(0) = \frac{16326}{2^{16}} \approx 0.249$, $\mu = \frac{259752}{2^{16}} \approx 3.96$. Note that the values of $x(0)$ and μ are both randomly generated via the standard `rand()` function, not specially chosen to optimize the proposed attacks. The eight involved plaintexts are shown in Fig. 2, from top to bottom, denoted by $g_0 \sim g_7$, respectively, where the first seven ones are candidates of known plaintexts and the last one is used to show the breaking performance. The corresponding ciphertexts of the eight plaintexts are respectively denoted by $g_0' \sim g_7'$, which are not shown here since all of them are like meaningless noisy signals.

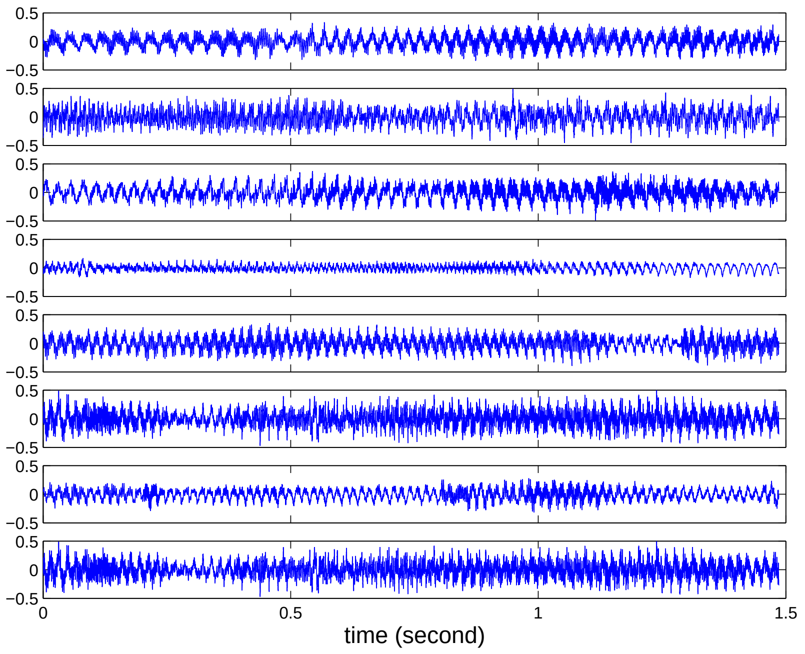

Figure 2: The eight plaintexts used in the experiments: $g_0 \sim g_7$ (from top to bottom).

[9]The chosen-plaintext attack is omitted here, since it is just a special case of the known-plaintext attack.

To simplify the experiments, we remove the speech CODEC in the whole HDSP-based VoIP system and directly use the uncompressed raw data as the plaintext. Such a simplification does not have any influence on the breaking performance of the secret key from the recovered chaotic bit sequence. For the known-plaintext attack, when the partially-recovered chaotic bit sequence is directly used to decrypt a ciphertext, the existence of speech CODEC may enlarge the recovery errors in the decoded voice signal. If such an enlargement is so serious that the intelligibility of the recovered voice signal is damaged, one can turn to derive the secret key, which always works as a perfect tool to break HDSP.

5.1 Partially reconstructing the chaotic bit sequence

Following the breaking procedure discussed in the last section, when plaintexts $g_0, \cdots, g_{n-1}$ and their ciphertexts $g'_0, \cdots, g'_{n-1}$ are known (for $n = 1 \sim 7$), we test the partial reconstruction of the chaotic bit sequence and the breaking performance when it is directly used to decrypt the ciphertext g'_7. The percentage of undetermined chaotic bits for different values of n is shown in Table 2, from which one can see that the percentage of the undetermined bits and the percentage of the undetermined bytes are both close to the theoretical predictions: $\frac{1}{2^{n+1}}$, and $1 - P'(n) = 1 - \left(1 - \frac{1}{2^{n+2}}\right)^2$, respectively.

Table 2: The percentage of the undetermined bits in the partially-reconstructed chaotic bit sequence, Per_1, and the percentage of plain-bytes of g_7 that are not correctly decrypted, Per_2, when $n = 1 \sim 7$ plaintexts are known.

n	1	2	3	4	5	6	7
Per_1	26.4%	13.8%	7.23%	4.23%	2.28%	1.17%	0.640%
$\frac{1}{2^{n+1}} \approx$	25.0%	12.5%	6.25%	3.13%	1.56%	0.781%	0.391%
Per_2	24.5%	13.1%	6.90%	3.92%	2.05%	1.06%	0.591%
$1 - P'(n) = 1 - \left(1 - \frac{1}{2^{n+2}}\right)^2 \approx$	23.4%	12.1%	6.15%	3.10%	1.55%	0.780%	0.390%

By randomly assigning values to all the undetermined bits, the partially-reconstructed bit sequence is used to decrypt the ciphertext g'_7 so as to get an estimation of the plaintext g_7. The decrypted results with respect to different values of n are given in Fig. 3. The decryption errors between the recovered plaintexts and the original plaintext g_7 are shown in Fig. 4, and the percentage of the decryption errors are listed in the last row of Table 2. Although the recovery error when $n = 1$ looks rather large, the recovered plaintext is still recognizable by human ears. The reason is that almost all frequency information remains in the recovered plaintext. For a comparison of the power energy spectrum of the original plaintext g_7 and those of the seven recovered plaintexts when $n = 1 \sim 7$, see Fig. 5. It is obvious that all important frequency peaks remain in the spectra of all the seven recovered plaintexts (but with larger amplitudes). As a result, even under the condition that the secret key is not derived, one known plaintext is still enough for recovering an intelligible version of the secret voice information. In addition, with a good noise reduction algorithm, the audio quality of the decrypted signal can be further enhanced.

5.2 Breaking the secret key

As analyzed above, generally it is possible to derive the secret key $x(0)$ (or its equivalent $x(i)$) and μ from the partially-reconstructed chaotic bit sequence. To do so, one needs to find 32 consequent chaotic bits in which less than m bits are undetermined. When only the plaintext g_0 is known, for different values of m, the numbers of all 32-bit groups satisfying the above requirement are listed in Table 3. One can see that even for $m = 0$ there are enough positions to derive the secret key. By taking the first occurrence of the 32 consequent bits to successfully derive a chaotic state $x(i)$ and the value of μ, one can decrypt any ciphertext from *the first group after* $x(i)$, which begins at the position $\left\lceil \frac{4 \cdot i}{S_g \cdot S_f} \right\rceil \cdot (S_g \cdot S_f)$. When g_0 is known, it was found that the first 32 consequent bits satisfying $m = 0$ occurs at $x(163)$, which is used to derive μ and then to decrypt g'_7. The decrypted plaintext and the recovery error are shown in Fig. 6. It can be seen that all plain-bytes from $g\left(\left\lceil \frac{4 \cdot 163}{16 \cdot 32} \right\rceil \cdot (16 \cdot 32)\right) = g(1024)$ are exactly recovered.

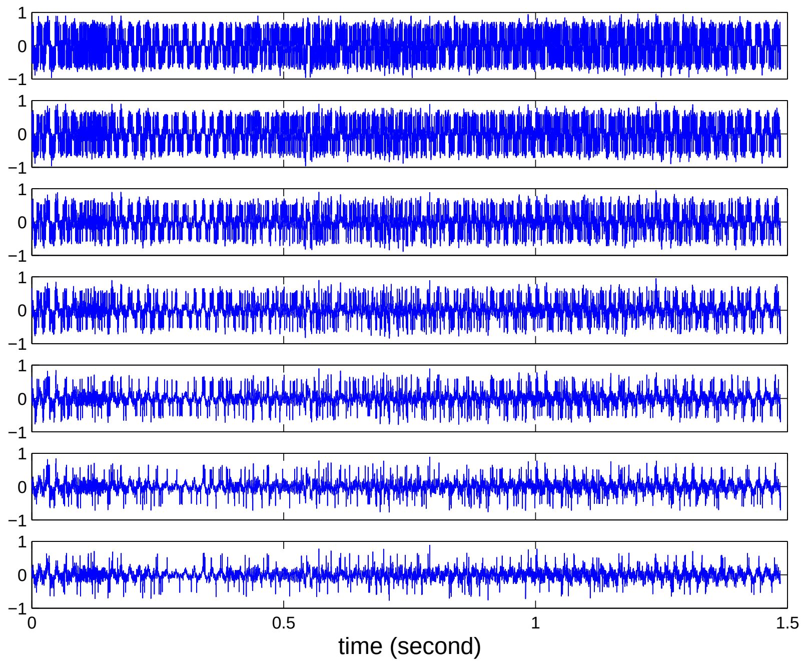

Figure 3: The decrypted plaintexts of g_7' with the partially-reconstructed chaotic bit sequence when $n = 1 \sim 7$ (from top to bottom) plaintexts are known.

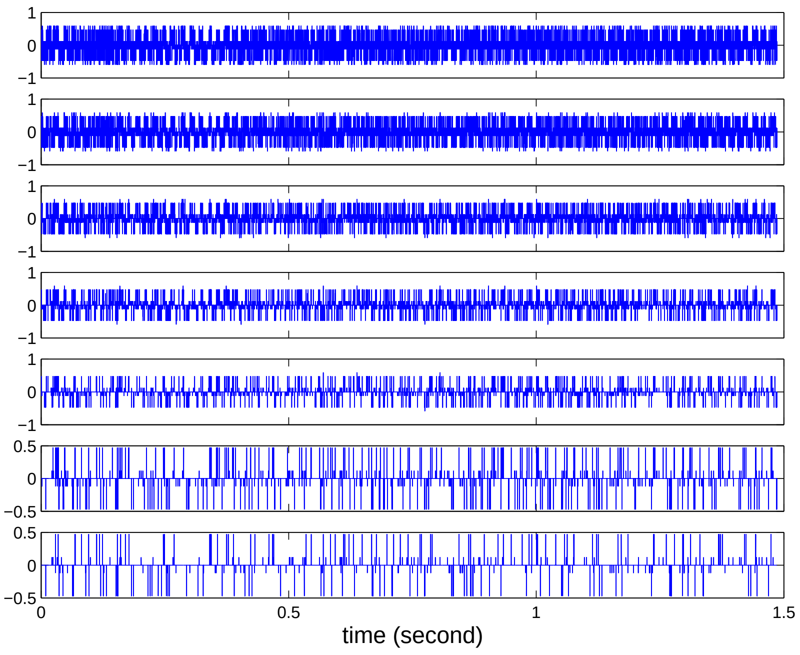

Figure 4: The decryption errors between the recovered plaintexts and the original plaintext g_7 when $n = 1 \sim 7$ (from top to bottom) plaintexts are known.

6 Improving HDSP

As shown in the last section, the insecurity of HDSP against known/chosen-plaintext attacks is attributed to the properties proved in Sub-section 4.2.1. In the first stage of breaking the inter-frame interleaving, Properties 1a,

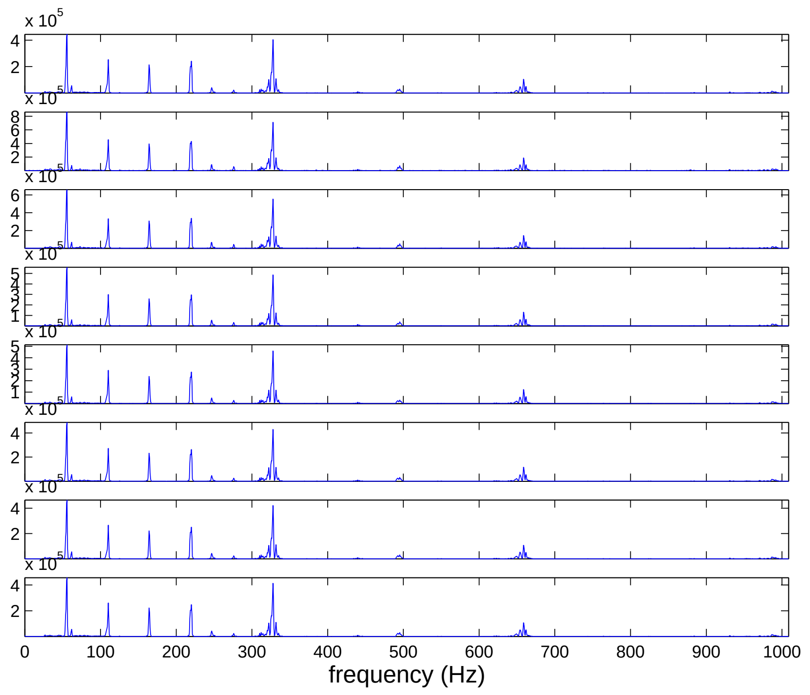

Figure 5: The power energy spectrum of the original plaintext g_7 (the 1st line) and the spectra of the recovered plaintexts when $n = 1 \sim 7$ (from the 2nd line to the last) plaintexts are known.

Table 3: The number of 32 continuous chaotic bits that have $m \leq 8$ undetermined bits, and the occurrence frequency.

m	0	1	2	3	4	5	6	7	8
$N(m)$	72	185	464	888	1574	2537	4106	6113	8715
$Freq[N(m)] \approx$	0.0045	0.0113	0.0283	0.0542	0.0961	0.155	0.251	0.373	0.532

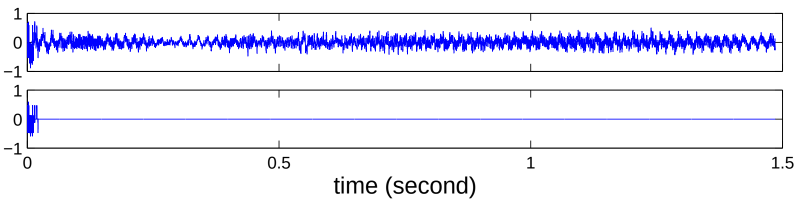

Figure 6: The decrypted plaintext of g_7' and the recovery error with the derived key.

2a, 2b and 3 are involved; in the second stage of breaking the intra-frame encryption, Properties 1b, 2a and 2b are involved. Also, the second breaking stage relies on the first one, since the intermediate signal g^* will not be available if the first stage does not work. This implies that the insecure properties play different roles in the known/chosen-plaintext attacks, which can be shown as follows:

$$g \xrightarrow[\text{(Property 2a, 2b and 3)}]{\text{Property 1a}} \left.\begin{matrix} g^* \\ \\ g' \end{matrix}\right\} \xrightarrow{\text{Property 1b, 2a, and 2b}} \{b(i)\} \to \left\{\begin{array}{l} x(0) \\ x(i), x(i+1) \to \mu \end{array}\right. \tag{10}$$

One can see that Property 1a is the basis of the whole attack, since Property 2a, 2b and 3 are just used to detect wrong permutation vectors. As a hint, if HDSP is modified to eliminate Property 1a, then the security against known/chosen-plaintext attack will be enhanced. However, if S_g is too small, it may be possible for an attacker to exhaustively search all $(S_g!)$ possible permutation vectors so as to pass the first breaking stage. Therefore, from the cryptographical point of view, all insecure properties of HDSP should be avoided to provide a high level of security. In addition, to resist other potential attacks, all known security defects should also be removed. In the following, we discuss how to amend the original HDSP scheme for better encryption.

- *Property 1a* is caused by the fact that no even bits are masked pseudo-randomly by the secret chaotic bits. Also, *Property 1b* is related to this defect. To fix this defect, we suggest *masking all bits, including odd bits and even bits.*
- *Properties 1b, 2a and 3* are caused by the incapability of swapping operations for encryption purpose, and *Properties 2a and 3* are also partially caused by the invertibility of the XOR operation. These properties can be destroyed by *changing the bit swapping and masking operations to other more complicated ones*, for example, inserting an extra masking operation before the two bits are swapped, or changing bit swapping operation to a different bit function.
- *Property 2b* is caused by two flaws: a) the equality of the masking operation and the swapping operation $Swap_w(a,b)$ when $a \neq b$ (see Eq. (4)); b) the reuse of all odd bits: for $k = 1, 3$, $\forall\ i = 1 \sim N-1$, each bit $b(4i+k)$ are used thrice – once for the swapping operation of $d_k^*(i)$, and twice for the masking operations of $d_{k+4}^*(i-1)$ and $d_k^*(i)$; similarly, when $i = 0$, the bit $b(k)$ is used twice for the swapping and masking operations of $d_k^*(0)$. The above two flaws disable the encryption for some bits, and make the two chaotic bit sequences $\{b(4i+1)\}_{i=0}^N$ and $\{b(4i+3)\}_{i=0}^N$ totally correlated. The first flaw can be fixed with the same method for eliminating Property 2a, and the second one can be removed by *avoiding any reuse of chaotic bits*, which means that at least $4+8=12$ chaotic bits are required for the encryption of each plain-byte (if swapping and masking operations are not replaced by other functions).

From the above discussions, one arrives at some principles for the design of a good encryption algorithm:

- Never repeatedly use any (secret) intermediate variables generated in the encryption procedure, such as the chaotic bits in the HDSP encryption scheme.
- Carefully avoid the occurrence of strong correlation between the plaintexts (or some intermediate variables *observable* for attackers) and the ciphertexts, and strong correlation between different secret parameters.
- Avoid the possibility of deriving the secret key under known/chosen-plaintext attacks. For HDSP, it is to avoid the possibility of deriving any chaotic states.
- Do not combine too-simple invertible functions to realize the encryption function. Instead, use good encryption functions defined in different groups to make their product irreversible, such as those used in IDEA [4].

7 Conclusion

In this paper, the security of a recently-proposed encryption technique for VoIP, called HDSP [2, 3], has been analyzed in detail. It has been found that HDSP cannot resist known/chosen-plaintext attacks, and that only one known/chosen plaintext is enough to break the secret key. It has also been found that the security of HDSP against the brute-force attack is very weak even for PC-s. Both theoretical and experimental analyses have been given to support the feasibility of the proposed attacks. In conclusion, HDSP is not suggested for security-sensitive applications, particularly, if the secret key may be reused to encrypt more than one plaintext (which is the scenario where known/chosen-plaintext attacks work very well [4]). Finally, some remedies have been suggested to improve the security of HDSP, along with some general guidelines for designing a secure encryption algorithm.

8 Acknowledgement

This research was partially supported by the National Natural Science Foundation, China under grant no. 60202002, by the Applied R&D Centers of the City University of Hong Kong under grants no. 9410011 and no. 9620004, and by the Hong Kong Polytechnic University's Postdoctoral Fellowship Scheme under grant no. G-YX63.

References

[1] V. Kulathumani, "Voice over IP: Products, services and issues," available online at http://www.cse.ohio-state.edu/~jain/cis788-99/voip_products/index.html, November 23, 1999.

[2] J.-I. Guo, C.-C. Lin, M.-C. Tsai, and S.-W. Lin, "An efficient voice over Internet protocol technique combining the speech data encryption and G.729 error recovery," in *Proc. Int. Computer Symposium (ICS'2002)*, 2002.

[3] J.-I. Guo, J.-C. Yen, and H.-F. Pai, "New voice over Internet protocol technique with hierarchical data security protection," *IEE Proc. – Vis. Image Signal Process.*, vol. 149, no. 4, pp. 237–243, 2002.

[4] B. Schneier, *Applied Cryptography – Protocols, Algorithms, and Souce Code in C*, 2nd ed. New York: John Wiley & Sons, Inc., 1996.

[5] Hao Bai-Lin, *Starting with Parabolas: An Introduction to Chaotic Dynamics.* Shanghai, China: Shanghai Scientific and Technological Education Publishing House, 1993, (In Chinese).

[6] S. Li, G. Chen, and X. Zheng, "Chaos-based encryption for digital images and videos," in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds. CRC Press, LLC, 2004, ch. 4, pp. 133–167, with preprint available at http://www.hooklee.com/pub.html.

[7] S. Li, C. Li, G. Chen, D. Zhang, and N. G. Bourbakis, "A general cryptanalysis of permutation-only multimedia encryption algorithms," Cryptology ePrint Archive: Report 2004/374, available online at http://eprint.iacr.org/2004/374, 2004.

[8] C. Li, S. Li, D. Zhang, and G. Chen, "Cryptanalysis of a chaotic neural network based multimedia encryption scheme," in *Advances in Multimedia Information Processing - PCM 2004 Proceedings, Part III*, ser. Lecture Notes in Computer Science vol. 3333. Springer-Verlag, 2004, pp. 418–425.