Multimedia Tools and Applications (2019) 78:33985–33997
https://doi.org/10.1007/s11042-019-08151-6

Securing audio data using K-shuffle technique

Salamudeen Alhassan [1] · Mohammed Muniru Iddrisu [2] · Mohammed Ibrahim Daabo [3]

Received: 23 October 2018 / Revised: 7 July 2019 / Accepted: 2 September 2019
Published online: 10 October 2019

Abstract
This paper presents a k-shuffle based audio scrambling technique which yields cipher audio with varying audibility that is useful in perceptual video encryption algorithms. In this technique the encryption and decryption algorithms group the input audio signals into m-piles of n-values each. The encryption procedure retains the first signal value (of the first pile) and the last signal value (of the last pile) in their respective positions while re-arranging the rest by taking in turns, the 1st values of each of the m-piles, then the 2nd values, up to the n^{th} values. At the receivers end, the decryption algorithm re-orders the scrambled audio signals back in to their respective original positions using the reverse of the k-shuffle technique. It is shown that, the audibility of the cipher audio is controlled by varying the number of piles. The performance of the proposed technique is demonstrated by simulations. The results of the simulations show that the proposed technique offers some level of resistance to chosen/known plaintext attacks.

Keywords Audio encryption · Audio decryption · Perfect shuffle · K-shuffle · Sound data · Faro shuffle

✉ Salamudeen Alhassan
salamprog@yahoo.com

Mohammed Muniru Iddrisu
immuniru@gmail.com

Mohammed Ibrahim Daabo
daabo2005@yahoo.com

[1] Department of Mathematics and Information Technology, Bagabaga College of Education, Tamale, Ghana

[2] Department of Mathematics, University for Development Studies, Navrongo-Campus, Navrongo, Ghana

[3] Department of Computer Science, University for Development Studies, Navrongo-Campus, Navrongo, Ghana

1 Introduction

In recent times various data protection techniques have been proposed by researchers to address the financial, security or privacy interest of multimedia data. One technique that is widely employed by researchers to address these needs is cryptography. In cryptography, the content of multimedia data is protected using an encryption algorithm. Authorized users can view the original content by using the decryption algorithm. Depending on the sensitivity level, multimedia data can either be totally, partially or perceptually encrypted.

Total encryption algorithms encrypt every bitstream of the input data. They are recommended for highly sensitive multimedia data. However, total encryption algorithms area not ideal for real-time applications due to their computational overheads. In partial encryption algorithms, selected parts of multimedia data are encrypted in order to reduce computational cost. They may be applied to protect the privacy of people in public video surveillance systems. Perceptual encryption algorithms secure the content of multimedia data (video) which have some financial interest. After applying these techniques, some perceptible parts of the cipher video are still visible / audible to the user. Cipher videos produced through this technique are mostly for advertisement purposes (try-before-purchase). For this reason, perceptual encryption algorithms are not ideal for protecting highly sensitive multimedia content [8]. Over the years, several video encryption algorithms have been proposed to address different security challenges [1, 14]. However, most of these algorithms have failed to address the security needs of the audio content embedded in source video. On the other hand, existing audio encryption algorithms lack the flexibility to control audibility. They are geared towards the total concealment of the original audio signals from adversaries [2–6, 11, 12, 15–17].

This paper proposes a k-shuffle based audio scrambling technique which can be integrated in perceptual video encryption algorithms such as the work of [1] in order to secure audio data embedded in video. The proposed scrambling scheme creates a variety of degraded audio qualities that are commensurate with the degraded visual qualities of the cipher video. Audio signals are extracted from video and the values arranged in m-piles of n-values each. The scrambling procedure involves maintain the first and last values while arrangement the rest by taking the 1st values of each pile, then the 2nd values of each pile, up to the n^{th} values of each pile. At the receivers' end, the scrambled audio signals are extracted from the cipher video and the reverse of the k-shuffle is applied to transform them back into their original form.

The rest of the paper is organized as follows. The k-shuffle technique use for the encryption and decryption algorithms is introduced in Section 2. In Section 3, a brief review of some related works on audio cryptosystems are outlined. The proposed cryptosystem is presented in Section 4. In Section 5, extensive analyses of the proposed cryptosystem are outlined. Finally, some conclusions are drawn on the work in Section 6.

2 Perfect k-shuffle

A deck of 52 cards when broken into two piles of 26 cards each can be reordered to get back into its original arrangement by alternatively taking cards from each pile in turns in 8 different repetitions. This is known as a perfect 2-shuffle and often referred to as a faro shuffle. Two types of faro or perfect 2-shuffles exist; the first shuffle also called an out

shuffle leaves the top card at the top and the second shuffle (in shuffle) is where the top card becomes the second card [9].

Let $n, m \in \mathbb{Z}$ and $n, m > 1$. Given $l = nm$ cards are orderly numbered from 1 to l. Place the cards in m piles of n cards each in order as follows; the first pile have cards from 1 through n, the second pile containing cards $n+1$ through $2n$, the third pile containing cards $2n+1$ through $3n$, etc., and the last pile containing cards $(ml)n + l$ up to nm. A perfect k-shuffle as described by Packard & Packard in [9] re-arranges the cards as follows; take the first cards in each pile, then the second, then the third, etc., and ending with the last cards in each pile. After the re-arrangement, the first and the last cards remain at their respective positions. Packard & Packard described the order of the k-shuffle, $dk(n)$, to be the minimum number of times the k-shuffle needs to be repeated to return the cards to their original configuration [9, 10, 15]. The flowchart diagram of a perfect k-shuffle is shown in Fig. 1.

3 Related work

Gnanajeyaraman et al. [3] explored the properties of sensitivity to initial conditions in higher dimensional chaotic maps to encrypt audio signals. In their method, variables are used as cipher keys and because chaotic maps are sensitive to initial conditions and also chaotic trajectory is unpredictable, their method achieved higher security, high key space and could withstand chosen/known-plaintext attacks. However, the processing time and error rate are not measured to ascertain the applicability to real-time processing of audio signals.

The performances of various audio encryption techniques are compared against basic symmetric encryption standards by Manpreet & Sukhpreet [7]. In their work, known cryptographic techniques such as DES, 3DES, Rc2, RC4, RC6, BLOWFISH are compared and their strengths and weaknesses in terms of the key size, block size, cipher type, attacks and network security are outlined. They concluded that making modifications to those techniques will lead to more secured audio data.

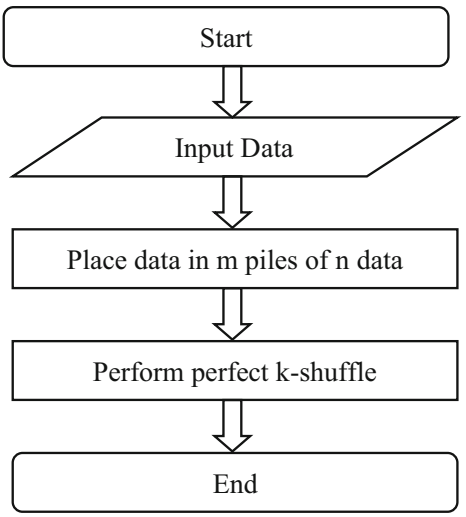

Fig. 1 Flow diagram of perfect k-shuffle technique

In [4] Khalil proposed a symmetric key encryption algorithm to encrypt real-time audio signal using logarithm operation. His method uses two real numbers (a, b) as cipher keys to encrypt audio signals p into cipher signals s by using the formula $s = \log_b(ap) = \log_b(a) + \log_b(p)$. The decryption function $p = b^s/a$ uses the same cipher keys to recover the plain audio signal p. Khalil's concluded that his method had a better error rate (0.00017) and yields audio signal exactly as the original than the RSA method which yielded low quality audio signal (error rate = 0.9737). However, no explanation is offered on how the method threats negative values in the encryption process given that logarithm of negative numbers yield complex values and also, audio signals contain real value.

Also, Khalil in [5] proposed a novel encrypting/decrypting technique for audio signal by using digital images as cipher keys and cover for audio signals. In his technique, each sample of the audio signal is combined with values of three color components of a pixel fetched from the cover image to produce a quaternion number. The absolute value of the quaternion number is transmitted and when received it is decrypted into the original audio signal using quaternion mathematics. Simulation results showed that the proposed method is robust and more secured against common signal processing (e.g. Gaussian noise) attacks without affecting the used bandwidth of the communication channel.

Sharma et al. in [13] proposed a partial encryption technique to secure wav audio signal by selective encryption and decryption of important audio information. In their technique, time domain audio signal is transformed to frequency domain audio signal by the use of Discrete Fourier Transform (DFT). RSA technique is then applied to encrypt and decrypt the lower frequency bands, since all the frequency regions do not participate equally in the communication. They observed that, encryption on the lower frequency band (phase values) is more effective than the higher one.

4 The proposed algorithms

In this Section, a different scrambling technique for encrypting/decrypting audio data using k-shuffle technique is introduced. We first present the cryptographic functions follow by the encryption and decryption algorithms.

4.1 Cryptographic functions

The cryptographic functions scramble original/cipher audio into cipher/plain audio using a modification of the k-shuffle technique. These functions compute the index of the x^{th} value to include in the next re-arrangement. The encryption function for the proposed algorithm is given as:

$$E(x) = \begin{cases} x & \text{if } x = 1 \text{ or } x = l \\ l & \text{if } \left|\left(n(x-1)+1\right|_{l-1} = 0 \\ |(n(x-1)+1)|_l & \text{if otherwise} \end{cases}$$

where n, m, l, $x \in \mathbb{Z}$, $l = nm$, $x = 1, 2, 3, \ldots, l$ and $\|$ is the modulus operator.

Thus, for the first condition, the index of the x^{th} value remains the same if it is the first or the last value. For all x^{th} values which meet condition two, the index is l (*i. e. nm*). Otherwise, the index of any other x^{th} value is computed as $|(n(x-1)+1)|_l$.

Also, the decryption function for the proposed algorithm is given as;

$$D(x) = \begin{cases} x & if\ x = 1\ or\ x = l \\ l & if\ \left|\left(n(x-1)+1\right|_l = 0 \\ |(m(x-1)+1)|_{l-1} & if\ otherwise \end{cases}$$

where $n, m, l, x \in \mathbb{Z}$, $l = nm$, $x = 1, 2, 3, \ldots, l$

4.2 The proposed encryption and decryption algorithms

The encryption algorithm employs the encryption function to scramble plain audio (p) into cipher audio (s). The cipher keys are *n, m, and i*. *n* is the number of rows, *m* the number of columns (piles) and *i* is the number of re-shuffling to perform. List 1 details the encryption algorithm.

List 1: Proposed encryption algorithm

1. Input p, n, m, i
2. Compute

```
l = nm
for k = 1 to i
    for x = 1 to length(p)
        if (x = 1 or x = l or x = length(p))
            s(x) = p(x)
        else
            temp = |(m(x − 1) + 1)|l−1
            if (temp == 0)
                temp = l − 1
            end
            s(x) = p(temp)
        end
    next x
p = s
next k
```

3. Transmit s

The decryption algorithm uses the decryption function to recover plain audio (p) from cipher audio (s).

The cipher keys are *n, m and i*. List 2 details the decryption process.

List 2: Proposed decryption algorithm

1. Input s, n, m, i
2. Compute

$l = nm$
for $k = 1$ to i
  for $x = 1$ to length(p)
    if ($x = 1$ or $x = l$ or $x = length(p)$)
      $p(x) = s(x)$
    else
      $temp = |(m(x-1)+1)|_{l-1}$
      if ($temp == 0$)
        $temp = l - 1$
      end
      $p(x) = s(temp)$
    end
next x
$s = p$
next k

3. Transmit p

Figure 2 shows the block diagram of the proposed audio cryptosystem. The original audio data is acquired and transformed (encrypted) into cipher audio using the secret key in the encryption sub-block. The cipher audio is then transmitted through a noise-free communication channel into the receiver. At the decryption sub-block, the received cipher audio is deciphered (decrypted) using the same cipher (secret) key.

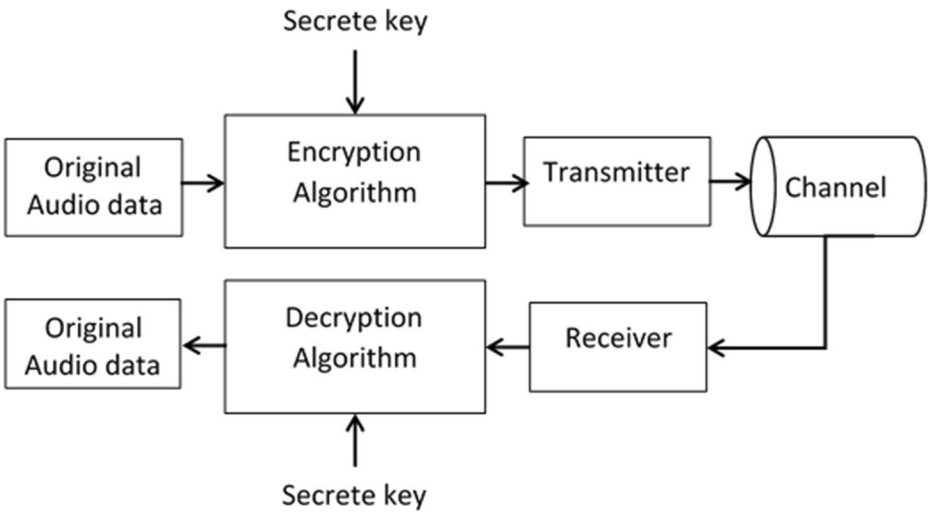

Fig. 2 Block diagram of the proposed encryption and decryption process

4.3 Key management

The cipher key for the proposed method is (n, m, $i \in Z$). n and m are chosen such that $l = nm$ is very close to the length of audio data to be encrypted/decrypted. Also, since k-shuffle is periodic for a given l, i should be chosen such that it does not compromise the technique.

4.4 Experimental results

The proposed algorithms are implemented using MATLAB Simulink tools to encrypt/decrypt real-time audio data from either an audio file or the microphone. Figure 3 shows the block diagram of the MATLAB Simulink simulation processes.

In Fig. 3, the encryption function (kShuffleEncoder) takes the input audio data p (a real mono audio file with sampling rate of 22,050 Hz, 16 bit) together with the cipher key (n, m, i) and encrypts it into the cipher audio signal s using List 1. The decryption function (kShuffleDecoder) receives the cipher audio s and the same cipher key (n, m, i) to recover the original audio signal p using List 2. The outputs of both encryption (cipher audio signals) and decryption (recovered audio signals) functions are stored to multimedia files for analysis. As an example, the figure shows the cipher key (n, m, i) as (2, 512, 1) and the input audio signal sample rate of 1024. Figure 4 shows the graphs of the audio signals obtained from the simulation process. Different k- (i.e. 50-, 1024-, 5000-, 10,000-) shuffles of audio signals are simulated for the encryption/decryption algorithms.

Table 1 summarizes the parts of the graph in Fig. 4. As seen in the graph, each cipher audio signal is different from the original audio signal and by implication every cipher key yields a different cipher audio signal.

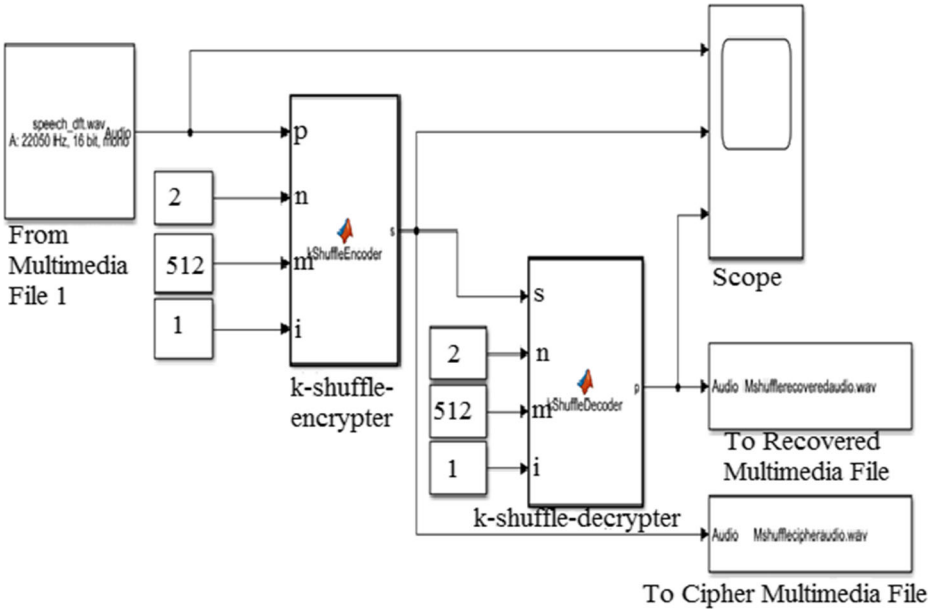

Fig. 3 MATLAB Simulink system for the proposed encryption and decryption of real-time audio data

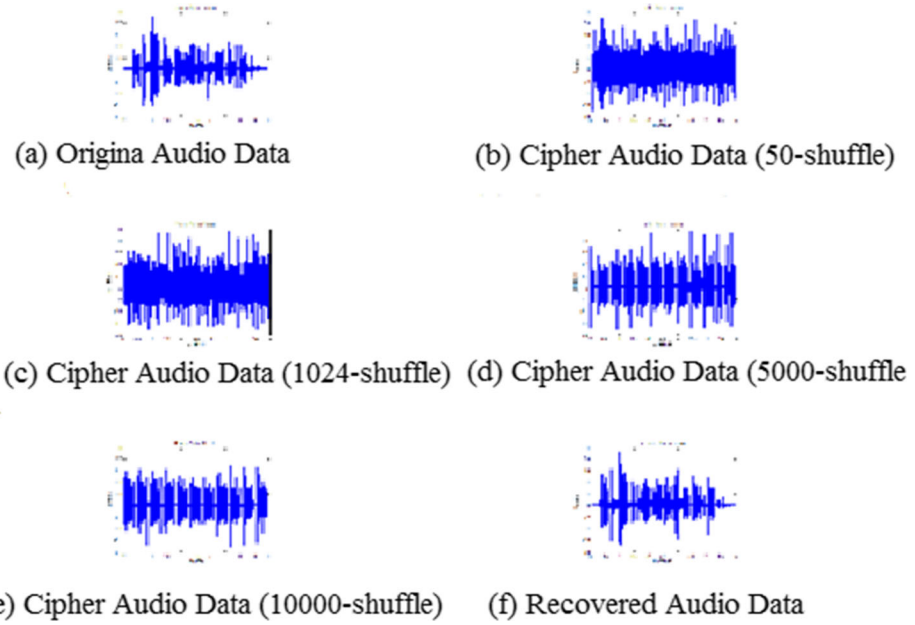

Fig. 4 Graph of MATLAB simulation results of Real-Time audio data using proposed method

5 Analysis of results

In this section, performance and security analysis of simulated results are presented. Issues pertaining to error rate, key space, processing time and some cryptographic attacks are tackled.

5.1 Error rate

We measure the difference between the recovered audio signals and the original input signals. Figure 5 shows the MATLAB Simulink system for measuring the error rate. As an example, the cipher key (*n, m, i*) for the simulation is (32, 32, 1) and the error rate computed between the original audio signal and the recovered one is 0.

Table 2 summarizes the error rates of the proposed method, RSA method and LOG method proposed by [4]. As seen from the table, the proposed method in terms of error rate is better than the RSA and LOG methods. Obviously, a few signals might have been lost during the recovery processes in [4] due to the way encoding processes are done in the encryption algorithms. In particular, the LOG method uses the absolute values of signals to avoid complex numbers.

Table 1 Summary of simulation properties of Fig. 4

Part	Cipher key (*n, m, i*)
a) Original audio signals	None
b) Cipher audio signals	(2200, 50, 1)
c) Cipher audio signals	(107, 1024, 1)
d) Cipher audio signals	(22, 5000, 1)
e) Cipher audio signals	(11, 10,000, 1)
f) Recovered audio signals	(2200, 50, 1)

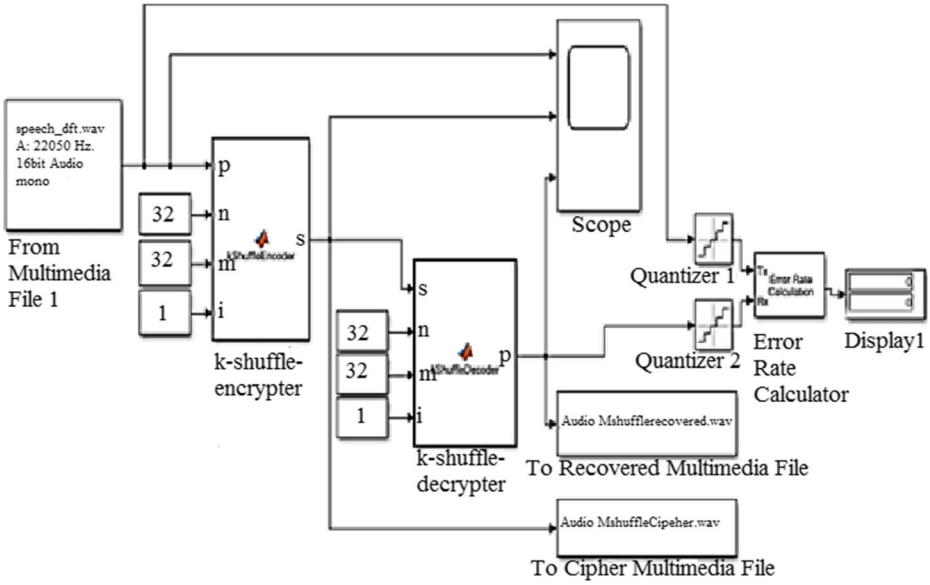

Fig. 5 MATLAB Simulink system to compute error rate of original and recovered audio signals

5.2 Key space

The proposed method uses three (3) parameters (n, m, i) for the cipher key. n and m increases as audio signal length increases and i does not exceed l. Thus each parameter has $n-$bits for every $n-$bits audio sample. For a 16-bits audio sample, the number of combinations for n and m is $2^{16} \times 2^{16}$. Since the parameter i does not exceed l, the number of repetitions for any possible combinations of n and m is $l-1$. Obviously, the key space is good enough for audio signal emanating from video data whose financial interest is only for a short time. However, the LOG and RSA methods have much larger key spaces than our proposed method.

5.3 Chosen and known-plaintext attack

Chosen and known-plaintext attacks are attacks in which the adversary besides knowing the encryption algorithm, can access or choose a set of plain messages, modify them and observe the corresponding cipher message. These attacks are considered the most prevalent and most threatened in the point of view of data protectors. One of the requirements of a good cryptosystem is its security against these attacks. Part c) in Fig. 6 is obtained by XOR-ing original audio sample a) and its corresponding cipher audio sample b). Part e) is a XOR-ing of the audio sample in c) and an unknown cipher audio sample d). Since part e) failed in recovering any meaningful (plain) audio sample, we conclude that the proposed method is safe against chosen/known-plaintext attack.

Table 2 Error rate comparison of three audio encryption methods

Method	Cipher key (n, m, i)	Error rate
RSA Method	$(e, n) = (5, 35)$	0.97370
LOG Method	$(a, b) = (0.0003, 300{,}000)$	0.00017
Our Proposed Method	$(n, m, i) = (32, 32, 1)$	0.00000

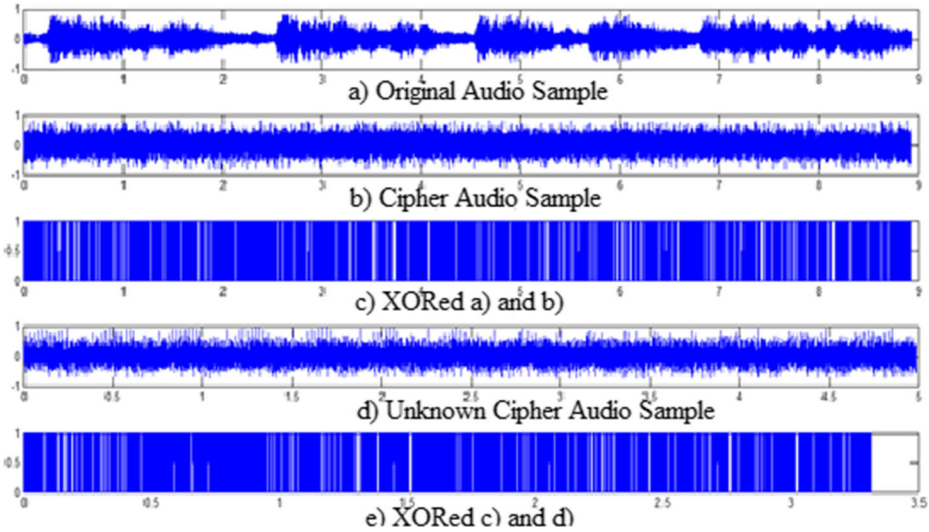

Fig. 6 Effect chosen/known-plaintext attack on proposed method

5.4 Processing time

We measure the processing time of the encryption and decryption algorithms to ascertain their feasibility in real-time operations. Table 3 shows the processing time for the simulation of "speech_dft.wav" (4 s) audio file. The results obtained show that the processing time for the encryption algorithm decreases as the number of piles (k-shuffles) increases for one-iteration as against the increase in processing time for the decryption algorithm. It is however observed that the processing time is directly proportional to the number of repetition. Higher repetitions lead to better degradation in audio quality. The average processing time taken by our proposed technique to complete an encryption and decryption processes for the 4 s audio file is ~2.50432 s. Also, the average processing times for both encryption and decryption processes of the LOG and RSA methods are respectively ~7.6125 s and ~9.96000 s.

6 Conclusion

This paper proposes a k-shuffle based audio scrambling technique which produces cipher audio whose audibility can be controlled. Contrary to the reviewed works, the proposed method provides a mechanism for the gradual degradation of audible information in source

Table 3 Processing time of proposed algorithms

Cipher key (n, m, i)	Encryption time (sec)	Decryption time (sec)
(2200, 50, 1)	3.07333	2.74167
(220, 500, 1)	0.32366	0.33366
(107, 1024, 1)	0.17133	0.34333
(22, 5000, 1)	0.03666	1.17433
(11, 10,000, 1)	0.02400	4.21500
(220, 500, 2)	1.97566	0.61333

audio. Thus, depending on the security needs, audibility ranges from intelligible to unintelligible.

Performance analyses of the method are carried out using simulations. In terms of recovery efficiency and processing speed, it is shown that, the proposed method has an advantage over the work of [4]. Obviously, the use of the absolute function in the LOG method to avoid complex numbers and the round function to truncate signal values in the RSA method might account for the error rates recorded.

It is also established that, the proposed method offers some reasonable levels of resistance against Bruce-force and chosen/known-plain attacks.

However, in terms of key space and security the work of [4] out-performs the proposed method.

As the proposed method is suitable for perceptual video encryption schemes, future work will look at its actual integration with existing works such as that of [1].

References

1. Alhassan S, Iddrisu MM, Daabo MI (2018) Perceptual Video Encryption via Unit Anti-diagonal Matrix. Appl Math Inf Sci 12(5):923–929
2. Farkash S, Raz S, Malah D (1991) Analog speech scrambling via the gabor representation. *17th Convention of Electrical and Electronics Engineers in Israel,* IEEE, 365–368
3. Gnanajeyaraman R, Prasadh K, Ramar D (2009) Audio encryption using higher dimensional chaotic map. Int J Recent Trends Eng 1(2):103–107
4. Khalil MI (2016) Real-time encryption/decryption of audio signal. Int J Comput Netw Inf Secur (IJCNIS) 8(2):25–31
5. Khalil MI (2017) Quaternion-based EncryptionDecryption of audio signal using digital image as a variable key. Int J Commun Netw Inf Secur 9(2):216–221
6. Makwana V, Parmar N (2014) Encrypt an audio file using combine approach of transformation and cryptography. Int J Comput Sci Inf Technol 5(3):4473–4476
7. Manpreet K, Sukhpreet KM (2014) Survey of various encryption techniques for audio data. Int J Adv Res Comput Sci Softw Eng 4(5):1314–1317
8. Menezes A, van Oorschot P, Vanstone S (1996) Handbook of applied cryptography, 3rd edition. CRC Press
9. Packard RW, Packard ES (1994) The order of a perfect k-shuffle. Off J Fibonacci Assoc, 136–44
10. Persi D, Graham RL, Kantor WM (1983) The mathematics of perfect shuffles. Adv Appl Math 4:175–196
11. Pitale P, Pateria A, Singh P, Golchha N (2015) Audio based secure encryption and decryption. Int J Comput Appl 0975 8887:1–4
12. Sadkhana SB, Mohammed RS (2015) Proposed random unified chaotic map as PRBG for voice encryption in wireless communication. Int Conf Commun Manag Inf Technol 65:314–323
13. Sharma S, Kumar L, Sharma H (2013) Encryption of an audio file on lower frequency band for secure communication. Int J Adv Res Comput Sci Softw Eng 3(7):79–84
14. Sultana SF, Shubhangi DC (2017) Video encryption algorithm and key management using perfect shuffle. Int J Eng Res Appl 7(7):1–5
15. Sultana SF, Shubhangi DC (2017) Video encryption algorithm and key management using perfect shuffle. Int J Eng Res Appl 7(7):01–05
16. Tamimi AA, Abdalla AM (2014) An Audio Shuffle-encryption algorithm. Proc World Congress Eng Comput Sci 1
17. Zhu Z, Burke J, Zhang L, Gasti P, Lu Y, Jacobson V (2011) A new approach to securing audio conference tools. *Proceedings of the 7th Asian Internet Engineering Conference*. ACM

Salamudeen Alhassan is a PhD student in Computational Mathematics at the University for Development Studies, Tamale, Ghana. He obtained his MSc degree in Computational Mathematics and BSc degree in Computer Science from the same University. His research interests are in the areas of cryptography, algorithms, image/video processing and software engineering.

Mohammed Muniru Iddrisu is a Senior Lecturer, a former Head of the Department of Mathematics and the current Vice-Dean of the Faculty of Mathematical Sciences, University for Development Studies, Ghana. He received his BSc, MSc, and PhD degrees in Mathematics from the University of Cape coast in Ghana, Norwegian University of Science and Technology, Trondheim, Norway and, University for Development Studies, Ghana respectively. His research interests are in Mathematical Analysis, Coding Theory, Cryptography, Mathematical Statistics and Applications. He has published several research articles in refereed journals and he is also a reviewer to many international journals of pure and applied mathematics. He is a member of the Ghana Mathematics Society, a member of Ghana Science Association and a member of the Management Board of the National Institute for Mathematical Sciences, Ghana.

Mohammed Ibrahim Daabo is a senior lecturer in Computer Science at the Department of Computer Science, Faculty of Mathematical Sciences, UDS, Navrongo campus. He holds PhD and MSc degrees in Computational Mathematics and BSc degree in Computer Science from the University for Development Studies, Tamale, Ghana. Dr. Daabo has over fifteen (15) years teaching experience in tertiary education and has conducted Academic Research in the fields of Overflow Detection, Reverse conversion, Error Detection and Epidemiological Modeling. His research areas include Computer Arithmetic, Residue Number System, Digital Logic Design and Mathematical Modeling.