Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol.11 No.1, 2022, pp.1-4
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajeat-2022.11.1.3200

Twin K-Shuffle Based Audio Steganography

Salamudeen Alhassan[1], Mohammed Ibrahim Daabo[2] and Gabriel Kofi Armah[3]
[1]Department of Computer Science, University for Development Studies, Tamale, Ghana
[2]Department of Computer Science, [3]Department of Business Computing
[2&3]C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
E-mail: salamudeen.alhassan@uds.edu.gh, idaabo@cktutas.edu.gh, garmah@cktutas.edu.gh

(Received 2 December 2021; Revised 20 December 2021; Accepted 7 January 2022; Available online 14 January 2022)

***Abstract* - Secure communication is most effective when it is covert. In the realm of covert communication, steganography conceals secret message within a cover medium. This ensures that adversaries who have access to this carrier medium are unaware of the existence of the secret message. This paper proposes a novel twin K-Shuffling and embedding technique that scrambles and hides secret message inside audio samples. The scrambling phase of the proposed technique consists of bit and character shuffling. The bit-shuffling scrambles the bit-string of each character in the secret message into cipher-text via K-Shuffle. The characters of the resulting cipher-text are then shuffled by another K-Shuffle technique to yield chaotic cipher-text. At the embedding phase, the scrambled cipher-text is randomly planted into the carrier audio samples. The novelty in this proposed technique is the provision of a three-layer protection for secret messages; bit, character, and encoding layers. Results and analyses show that this technique satisfied both embedding and encryption requirements of steganographic systems.**

***Keywords*: Steganography, K-Shuffle, Bit-Shuffling, Secret Message, Cover Audio, Stego Audio, Embedding Function, Extraction Function**

I. INTRODUCTION

A secret message can be concealed in another message (i.e. carrier medium) in such a way that its existence is unnoticed. The goal here is to communicate securely in an entirely unperceivable manner and to modify the carrier in such a way that nothing is revealed (neither the concealment of the message nor the secret message itself). This process is referred to as Steganography. The concept of data hiding using steganographic techniques has gain usage in areas such as covert communication, digital rights management, annotation, access control, etc. [1].

A typical steganographic system consists of two functions; the embedding and the extraction functions (see Fig. 1). The embedding function generates a steganogram (stego audio) by use of the secret message and the carrier medium (cover audio). A number of options such as Image, Audio, Video, Text File, etc. exist as carrier media. This work adopts an audio file as the carrier medium due to its excellent noisy spaces which is ideal for data masking. The extraction function at the receiver's end reproduces the secret message in its original form from the stego audio [1].

The use of a steganographic algorithm is undetected if the statistical characteristics of a stego audio and its cover audio are the same. Thus, secured steganographic systems ensure that messages read from a stego audio are statistically similar to potential messages read from a cover audio [1].

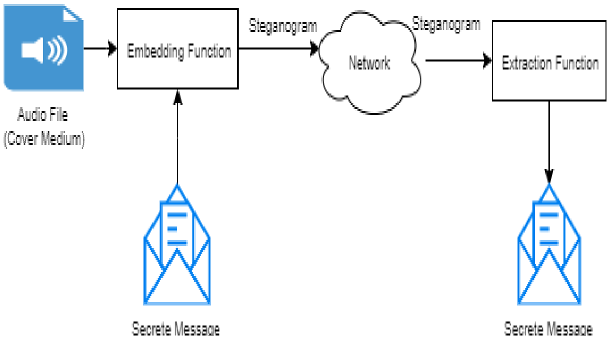


Fig. 1 Steganographic System

A steganographic algorithm may use secret keys for both encryption and embedding. Here, the encryption process uses a cryptographic key to encrypt the secret message while the embedding process employs a steganographic key to hide the cipher message over a cover audio. The most common steganographic technique is the Least Significant Bit (LSB) insertion method which implants the secret message in the LSB of the cover audio. However, increasing embedding capacity of the LSB method generates stego audio that is susceptible to statistical attacks [2] and could lead to easy recovery and destruction of secret message [3]. Other audio steganographic techniques involve the use of Genetic Algorithms (GA) [1] [4]-[9], Echo Hiding and Spread Spectrum [3] to conceal secret message in cover audio.

This work proposes a novel audio steganographic system via twin k-shuffle and substitution techniques. The first phase consists of encryption. Here, the secret message is double encrypted via scrambling of both its bit-strings and characters using k-shuffle. Thus, for each character of the secret message, its bit-string is obtained and scrambled. This is followed by a second scrambling and normalization of the resulting cipher-text. At the embedding phase, the normalized cipher-text are carefully scattered into the carrier audio signals.

In the rest of the paper, section II discusses the K-Shuffle technique while the proposed steganographic system is presented section III. Experimental results and analysis are examined in sections IV and conclusions to the work are presented in section V.

II. THE K-SHUFFLE TECHNIQUE

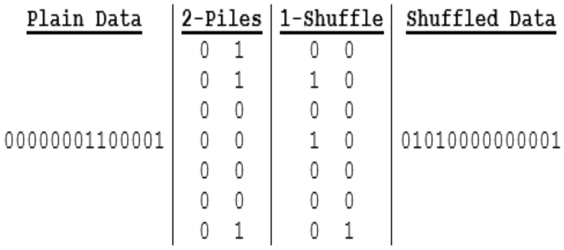
K-Shuffle is a card shuffling technique that has been used over the years to shuffle cards in piles. This technique ensures that the original arrangement of the cards recur after a number of rearrangement of the piles. One such K-Shuffling technique is the Faro-Shuffle (perfect 2-Shuffle). Given a deck of 52 cards, a Faro-Shuffle can be achieved by rearranging the cards broken in 2-piles of 26 cards each. The rearrangement is done by alternatively taking cards from each of the piles. After 8 different repetitions, the original arrangement of the cards is obtained. A perfect 2-shuffle can either be out-shuffle (i.e. the top card is not shuffled) or in-shuffle (i.e. the top card becomes the second card) [10], [11].

Let n, m ∈ ℤ and n, m > 1. Given l = nm bit-strings that are orderly numbered from 1 to l. Place the bits in m-piles of n bits each in order as follows; the first pile have bits from 1 through n, the second pile containing bits n + 1 through 2n, the third pile having bits 2n+1 to 3n, etc., and the last pile containing bits (m-l)n + l up to nm [10]. Thus, the position of a bit (b_p) in a K-Shuffling process is presented in Equation (1).

$$s(b_p) = \begin{cases} b_p, & for\ b_p = 1\ or\ b_p = l \\ l, & for\ \left|n(b_p-1)+1\right|_{l-1} = 0 \\ \left|n(b_p-1)+1\right|_{l}, & otherwise \end{cases} \quad (1)$$

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

Packard & Packard [11] described a perfect K-Shuffle as taking the first card in each pile, then the second, then the third, etc., and ending with the last card in each pile. This rearrangement does not affect the positions of the first and last cards [9], [11], [12]. Fig. 2 illustrates a perfect 2-shuffle of 14 bit-string.



Plain Data	2-Piles		1-Shuffle		Shuffled Data
	0	1	0	0	
	0	1	1	0	
	0	0	0	0	
00000001100001	0	0	1	0	01010000000001
	0	0	0	0	
	0	0	0	0	
	0	1	0	1	

Fig. 2 1-Shuffle of 2-piles Data

Aside card shuffling, K-Shuffle have found usage in encryption [10] – [13] to protect data.

III. PROPOSED ALGORITHMS

A flow diagram of the proposed steganographic system is presented in Fig. 3. In order to create a stego audio, the sender passes the secret message through an encryption process (K-Shuffling functions) for scrambling of both bit-strings and characters. The resulting cipher-text together with the cover file (audio file) are passed to the Embedding function for processing (see subsection *A*). At the receiver's end, the cipher-text is extracted from the stego audio and sent through a decryption process (reversed K-Shuffling functions) to recover the original secret message (see subsection *B*).

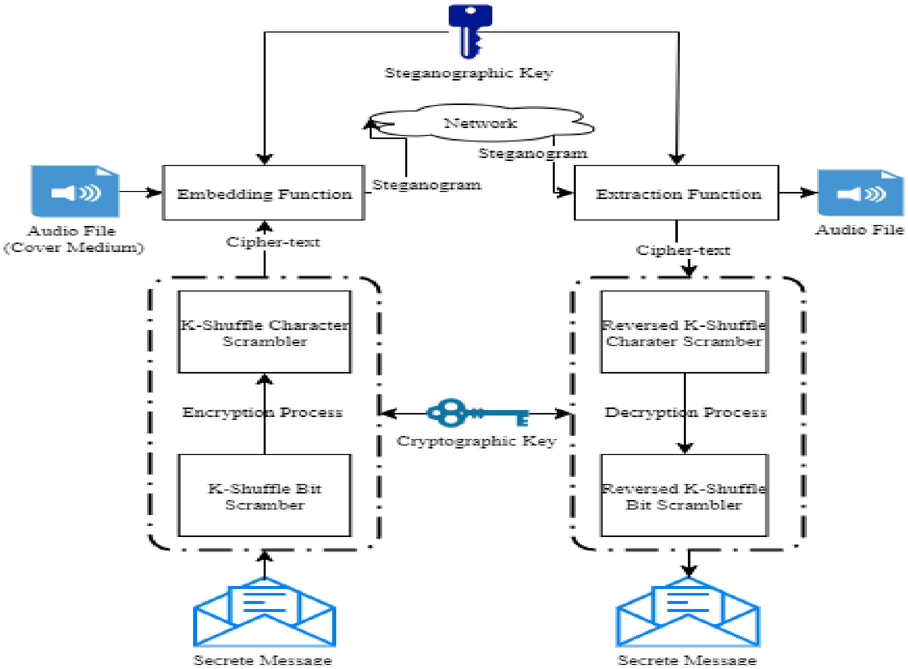


Fig. 3 Proposed Steganographic System

A. *Proposed Encryption and Embedding Algorithms*

Input: Cover file (cf), Secret message (sm), Bit-piles (bp), Character-piles (cp), steganographic key (sk)

Output: stego audio (stego), length of secret message (lsm)

```
1. Read c f, sm, bp, cp and sk.
2. Convert each character of sm to ASCII code.
3. for each ASCII code.
   a. Convert to 12+ (pb - (12 % pb)) bit-string.
   b. Scramble bit-string using K-Shuffle bit scrambler.
   c. Convert scrambled bit-string to integer value.
4. In the K-Shuffle character scrambler.
   a. Shuffle values using cp into cipher-text.
   b. Normalize cipher-text.
5. Using the Embedding Function.
   a. Obtain lsm and cf.
   b. Determine signals points in cf to embed cipher-text.
   c. Embed cipher-text cf.
   d. Obtain the steganogram (stego).
6. Transmit stego.
```

B. *Proposed Extraction and Decryption Algorithms*

Input: Steganogram (stego), Length of Secret message (lsm), Bit-piles (bp), Character-piles (cp), steganographic key (sk)

Output: Original secret message

```
1. Read stego, lsm, bp, cp and sk.
2. In the Extraction Function
   a. Compute the embedded points in stego using lsm and sk.
   b. Extract cipher-text from stego.
3. Perform reversed character K-Shuffle using cp.
4. Perform reversed bit K-Shuffle.
5. Normalize recovered values.
6. Convert values to Original secret message.
```

TABLE I APPLYING K-SHUFFLING TECHNIQUE ON “ENCRYPTION”

Procedure	**Plain**									
	e	**n**	**c**	**R**	**y**	**P**	**t**	**i**	**o**	**n**
ASCII Code	101	110	99	114	121	112	116	105	111	110
Bit Shuffle	5137	5204	5125	5380	5441	5376	5392	5185	5205	5204
Character Shuffle	5137	5392	5204	5185	5125	5205	5380	5204	5441	5376

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Results

The proposed Audio Steganography technique was simulated on stereo and mono audio sample formats (.aif, .flac, .ogg, .mp3, and .wav) and sample rates (8 kHz, 22.05 kHz, and 44.1 kHz). The duration of samples ranged between 4 to 20 seconds. All coding, simulations and analyses were done using MATLAB.

Table I shows the application of Twin K-Shuffle on the message “encryption”. Both bit and character shuffles are done using a single 2-Shuffle process. After shuffling, the positions of the first and last bits and characters are unaffected.

Fig. 4 shows the plots of a 393586 samples “ogg” and a 31488 samples “wav” files and their respective stego audios. The stegos were obtained by embedding a message containing 1107 characters.

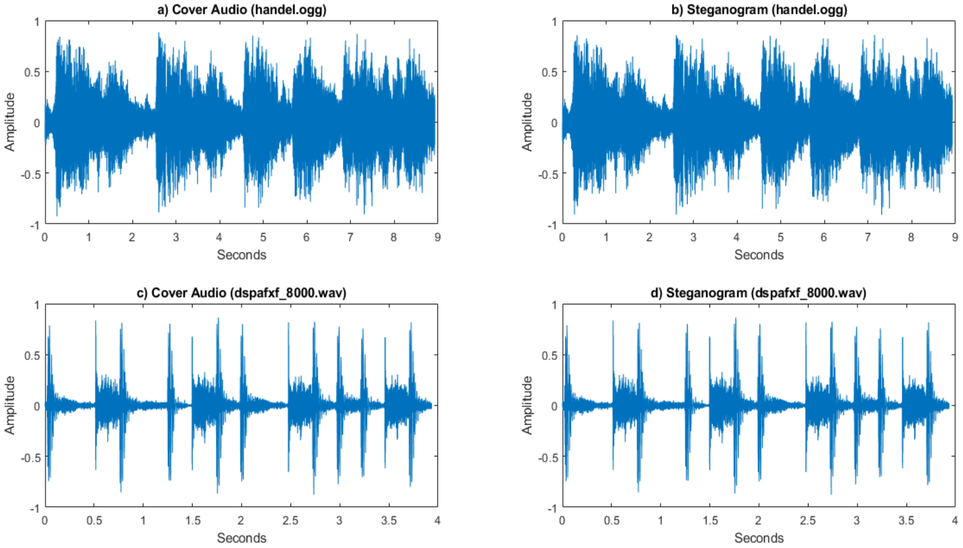


Fig. 4 Graph of cover audios and their respective stego audios

B. Analysis

The efficiency of a steganographic algorithm is assessed according standards such as; complexity, transparency, embedding capacity and robustness. The Mean Square Error (MSE) and the Signal to Noise Ration (SNR) are two important measures of robustness. While MSE measures the difference between cover and stego audios, SNR measures the level of distortion between them. Smaller values of MSE depict close similarity with the two file. Thus zero (0) MSE implies that the two file are the same [3], [14] – [15]. MSE is computed from Equation (2).

$$MSE(p,s) = \frac{1}{N}\sum_{i=1}^{N}\left(c(i)-s(i)\right)^2 \qquad (2)$$

Where c(i) and s(i) are cover and stego audios respectively.

The MSE obtained are 0.000405 and 0.000171 respectively for the cover and stego audios of “handle.ogg” and “speech_dft.wav”. The similarities between audios are demonstrated further in Fig. 4 and Fig. 5 as no noticeable differences exist between them. SNR is measured in decibels (db) and computed as in Equation (3). It measures the quality of audio signal by comparing the cover and stego audios. Higher SNR values imply more useful signals than noise.

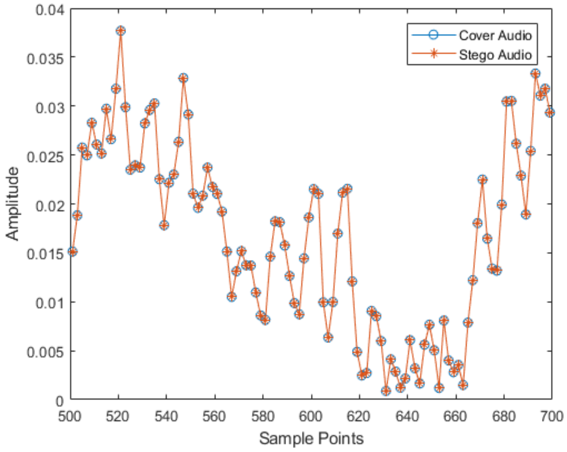


Fig. 5 First 100 samples of “speech_dft.wav” against its stego audio

$$SNR = 10\log_{10}\frac{\sum_{i=1}^{N} p(i)^2}{\sum_{i=1}^{N}\left(c(i)-s(i)\right)^2} \qquad (3)$$

Where c(i) and s(i) are cover and stego audios respectively.

The SNR computed are 33.8548db and 36.6027db respectively for the cover and stego audios of “handle.ogg” and “speech_dft.wav”. These depict that the quality of stego audios are similar to their corresponding cover audios.

V. CONCLUSION

This paper presented a new audio steganography technique that integrates cryptography through K-shuffling. The first stage encrypts, encodes and embeds the secret message in to cover audio to produce stego audio. This offers a 3-layer protection for secret message. The second stage occurs at the receiver’s end which extracts, decodes and decrypts the stego audio in to the secret message. Experimental results revealed that the proposed technique is robustness and satisfy both embedding and encryption requirements of steganographic systems. Notwithstanding these advantages, the quality of stego audio is greatly affected as the length of secret message increases relative to cover audio. Further experiment should that this limitation is pronounced when the length of secret message is greater than a fifth of cover audio. Future research will aim at overcoming this limitation.

REFERENCES

[1] R. Manisha, and T. Manisha, “Genetic algorithm in audio steganography,” *ArXiv: abs/1407.2729*, Vol. 13, No. 1, pp. 29-34, 2014.

[2] N. Cvejic, and T. Seppanen, “Increasing the capacity of LSB-based audio steganography,” *2002 IEEE Workshop on Multimedia Signal Processing*, St. Thomas, VI, USA, 2002. [Online]. Available: https://doi.org/10.1109/MMSP.2002.1203314

[3] A. Koyun, and H. B. Macit, “Generating a stego-audio data using LSB technique and robustness test,” *Journal of Engineering Sciences and Design*, Vol. 6, No. 1, pp. 87-92, 2018.

[4] K. Bhowal, D. Bhattacharyya, A. J. Pal, and T. H. Kim, “A GA based audio steganography with enhanced security,” *Telecommunication Systems*, Vol. 52, No. 4, pp. 2197-2204, 2011.

[5] K. Bhowal, A. J. Pal, G. S. Tomar, and P. P. Sarkar, “Audio steganography using GA,” *2010 International Conference on Computational Intelligence and Communication Networks*, 2010. [Online]. Available: https://doi.org/10.1109/CICN.2010.91

[6] K. M. Christine, “Genetic algorithm based model in text steganography,” *The African Journal of Information Systems*, Vol. 5, No. 4, pp. 131-144, 2013.

[7] B. A. Mitras, and N. F. H. al-Alusi, “Using hybird genetic algorithm in audio steganography,” *Iraqi Journal of Statistical Sciences*, Vol. 13, No. 25, pp. 150-164, 2013.

[8] C. C. Sobin, and V. M. Manikandan “A Secure Audio Steganography Scheme using Genetic Algorithm,” *5th Proceedings of the IEEE International Conference Image Information Processing, ICIIP 2019*, pp. 403-407, 2019.

[9] S. F. Sultana, and D. C. Shubhangi, “Video encryption algorithm and key management using perfect shuffle,” *Int. Journal of Engineering Research and Application*, Vol. 7, No. 7, pp. 01-05, 2017.

[10] S. Alhassan, M. M. Iddrisu, and M. I. Daabo, “Securing audio data using K-shuffle technique,” *Multimedia Tools and Applications*, Vol. 78, No. 23, pp. 33985-33997, 2019.

[11] R. W. Packard, and E. S. Packard, “The Order of a perfect k-shuffle,” *The Official Journal of the Fibonacci Association*, pp. 136-44, 1994.

[12] D. Persi, R. L. Graham, and W. M. Kantor, “The Mathematics of Perfect Shuffles,” *Advance in Applied Mathematics*, Vol. 4, pp. 175-196, 1983.

[13] I. Z. Alhassan, E. D. Ansong, G. Abdul-Salaam, and S. Alhassan, “Enhancing image security during transmission using residue number system and K-shuffle,” *Earthline Journal of Mathematical Sciences*, Vol. 4, No. 2, pp. 399-424, 2020.

[14] R. Kaur, J. Bhatia, H. S. Saini, and R. Kumar, “Multilevel Technique to Improve PSNR and MSE in Audio Steganography,” *International Journal of Computer Applications*, Vol. 103, pp. 1-4, 2014.

[15] Z. Wang and A. C. Bovik, “Mean Squared Error: Love It or Leave It?”, *IEEE Signal Processing Magazine[98]*, 2009. [Online]. Available: https://doi.org/10.1109/MSP.2008.930649.