AUDIO CRYPTOGRAPHY VIA ENHANCED GENETIC ALGORITHM

Salamudeen Alhassan[1], Gabriel Kofi Armah[2] and Issah Zabsonre Alhassan[3]

[1,2]Department of Business Computing, School of Computing and Information Sciences, C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
[3]Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
University for Development Studies, Tamale, Ghana

ABSTRACT

As communication technologies surged recently, the secrecy of shared information between communication parts has gained tremendous attention. Many Cryptographic techniques have been proposed/implemented to secure multimedia data and to allay public fears during communication. This paper expands the scope of audio data security via an enhanced genetic algorithm. Here, each individual (audio sample) is genetically engineered to produce new individuals. The enciphering process of the proposed technology acquires, conditions, and transforms each audio sample into bit strings. Bits fission, switching, mutation, fusion, and deconditioning operations are then applied to yield cipher audio signals. The original audio sample is recovered at the receiver's end through a deciphering process without the loss of any inherent message. The novelty of the proposed technique resides in the integration of fission and fusion into the traditional genetic algorithm operators and the use of a single (rather than two) individual(s) for reproduction. The effectiveness of the proposed cryptosystem is demonstrated through simulations and performance analyses.

KEYWORDS

Cryptography, Bit fission, Fusion, Genetic algorithm, Cipher audio

1. INTRODUCTION

The growing demand for secured data transmission between communication parties over networks has led to the design of varied techniques to curb unauthorized access. The Topmost among these techniques is Cryptography (the science of concealing and revealing data). In cryptography, data is encrypted (to conceal the intended information from unauthorized access) and decrypted (to reveal the intended information to authorized users) using a key. Two types of cryptographic systems exist; public- and private-key cryptography. In public-key (asymmetric) cryptography, two keys are used (one for enciphering and the other for deciphering) while in private-key (symmetric) cryptography one key is used for both the enciphering and deciphering processes [1, 2, 3].

One type of multimedia data cryptography is perceptual video encryption [2, 3]. Here, some traits of the original video are still visible after enciphering. This technique is suitable for a video which has financial value and yet needs to be advertised to attract more potential customers [2, 3]. Of the much research on perceptual encryption, little has been done on the security of the embedded audio samples. The work of [4] proposed an enhanced audio scrambling technique that is ideal for integration with perceptual video encryption algorithms. This work expands [4] by applying an enhanced Genetic Algorithm (GA) to secure audio samples in the embedded video.

Here successive audio samples are computationally engineered to produce degraded audio files of varied audibility.

The enciphering algorithm of this proposed technique acquires and conditions audio samples into 24-bit strings. This is followed by bit fission (into n-blocks), crossover, mutation, fusion, and deconditioning processes to produce the cipher audio samples which are transmitted. The receiver's side applies the reverse of the enciphering process (i.e. deciphering process) to recover the original audio samples.

Opposed to most GA algorithms, the proposed algorithm utilises individual signals to reproduce offspring. This preserves size and speeds up data processing. Also, the introduction of fission and fusion into the GA operation enhances the security of the technique.

The rest of the paper discusses GA and related works in section 2, the proposed cryptosystem in section 3, experimental results and performance analysis in section 4 and section 5 respectively. In section 6, conclusions are drawn.

2. REVIEW OF RELATED WORKS

Genetic Algorithms (GA) have received tremendous attention in the field of data protection [5, 6, 7]. Having its lineage in biological evolutionary theories, GA is commonly used to find solutions to optimization problems. In GA, successive populations are replaced by biologically engineered populations of chromosomes [11]. Normally represented in binary format, a chromosome denotes a candidate solution to a problem. A chromosome is assumed to be a point in the search space of candidate solutions [11, 14, 16].

GA methods typically consist of a population of chromosomes, a selection via a fitness function, a crossover to produce new individuals, and a random mutation of new individuals. The starting point for GA is the initialization of a population of individuals via guess. These individuals grow through iterations commonly referred to as generation. Using the fitness function, each individual in each generation is evaluated. To generate the next generation of individuals, genetic operators are used to manipulate individuals in the population. This process is continued until some form of criterion is met. The three types of operators commonly used in genetic algorithms are [11, 16];
Selection: The reproduction process requires this operator to select suitable chromosomes in the population. The fitter the chromosome, the more times it is likely to be nominated for reproduction.

Crossover: The crossover operator randomly elects a point and swap the subsequences before and after that point between two chromosomes to produce two offspring. This operator typically resembles biological recombination between two single-chromosome organisms. Given two bit strings 10101001 and 10010001, a single-point crossover after point four (4) will yield the two offspring 1010**0001** and 1001**1001**. Crossover can be a single point, two-point or uniform crossover.

Mutation: Mutation arbitrarily flips some of the bits in a chromosome. This operator may be applied at every bit position in a string with a very small probability. Thus mutating the bit string 10101001 at positions three (3) and five (5) will produce 10**0**0**1**001.

Genetic algorithms have found usage in different areas of data protection. In the field of steganography, GAs are used to conceal information into carrier signals before transmission

through the network. Here, the existence of the message is normally unknown to adversaries who have access to the carrier signal [5, 6, 14, 15, 16].

In the field of cryptography, GAs have also been applied to secure data from adversaries via enciphering algorithms. Authorize users obtain the original information by use of deciphering algorithms [7, 8, 10, 13].

Jhingran et al. [9] surveyed works on image security via GAs. The work summaries the techniques, strengths and weaknesses of the algorithms surveyed.

Nazeer et al. [12] proposed multiple-step encryption and decryption algorithm using a random number generator and GA for text security. By applying genetic operation on randomly generated numbers and prime numbers a key of about 80 to 128 length is generated. Crossover and mutation are applied to the data and the resulting diffused form XORed with the encryption key. The fitness of each chromosome was measured by use of Shannon Entropy. The authors concluded that their scheme has an advantage over DES and AES in terms of key strength and lack behind in terms of processing time.

A three-layered encryption and decryption scheme was proposed in [13] by Agbedemnab et al. to secure textual data using a blind of GA and Residue Number Systems. Simulated results produced chaotic cipher-text and they revealed that their scheme was robust with good keyspace.

In [7] Cheng et al. proposed a new image encryption technique by combining quantum genetic algorithm and compressive sensing. The authors stated that their proposed scheme reduces data storage, speeds up enciphering and deciphering processes, and resists both statistical and plaintext attacks.

It is instructive to note that the reviewed works use the traditional GA operators and generate offspring from two individuals in the population. However, our work augments the GA operators by two (fission and fusion) and depends on a single individual for reproduction. These enhance security and processing speed.

3. PROPOSED AUDIO CRYPTOSYSTEM

The proposed encryption and decryption processes for audio samples operate at the bit level which is ideal for the on-the-fly encryption/decryption of multimedia data. Each process starts with the selection and conditioning of audio samples (from the population) into n-bits string (e.g. 24-bits). The conditioning process involves adding (a small positive integer, e.g. 1) and multiplying (by a big positive integer, e.g. 10000000). The resulting bit string is fission into n-blocks. Crossover, mutation, fusion, and deconditioning processes are then applied to yield cipher/plain audio samples.

3.1. The Enciphering Algorithm

Figure 1 shows the flow diagram of the enciphering process. The inputs to the encryption algorithm include the plain audio samples and the encryption key parameters. The Enciphering Process is presented as follows;

1. Acquire audio samples and encryption keys.
2. Condition and fission audio sample into n-blocks bit strings.
3. Perform crossover operations between adjacent blocks.

4. Perform mutation operations on the bit strings.
5. Fusion chaotic n-blocks bit strings.
6. Decondition cipher bit string into cipher audio sample.
7. Transmit cipher audio sample.

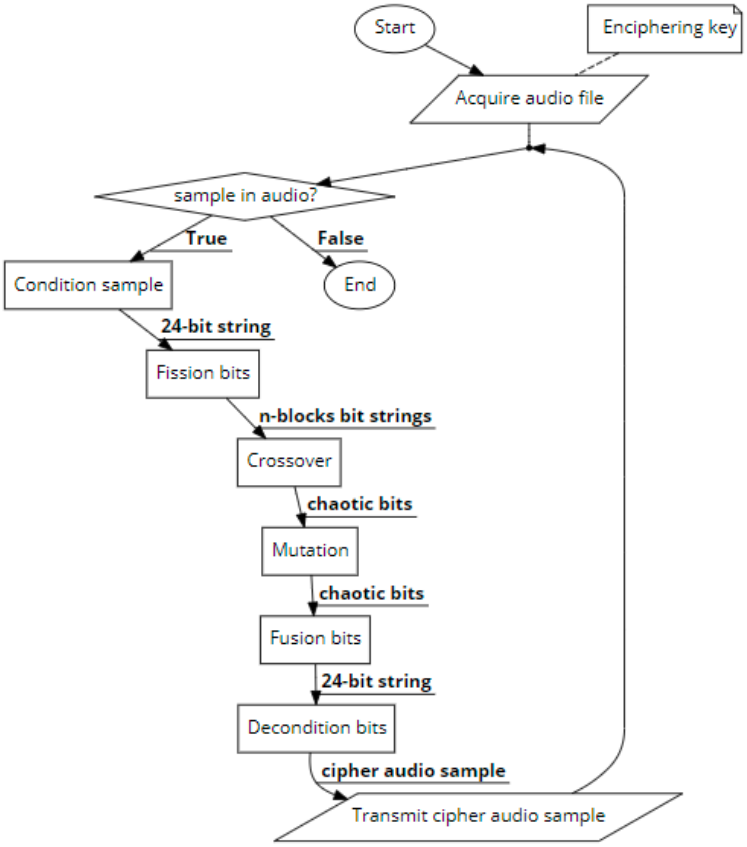


Figure 1. Flow diagram of enciphering process

3.2. The Deciphering Algorithm

The deciphering algorithm is a reverse of the enciphering algorithm that is used to recover the original audio sample. Figure 2 illustrates this process. The inputs to the deciphering algorithm are the cipher audio samples and deciphering key parameters (same as enciphering key parameters). The Deciphering Process is illustrated as follows;

1. Acquire cipher audio sample and deciphering key parameters.
2. Condition and fission audio sample into n-blocks of bit strings.
3. Perform reverse crossover between adjacent blocks.
4. Perform reverse mutation operations on the bit strings.
5. Fusion ordered n-blocks bit strings.
6. Decondition recovered bit strings into the original audio sample.

4. EXPERIMENTAL RESULTS

We present an illustration of the enciphering and deciphering of the audio sample -0.000092. The encryption/decryption key parameters are; c => crossover, f => fission, m => mutation, cp1 =>

first conditioning parameter and cp2 => second first conditioning parameter fo => order of fusion.

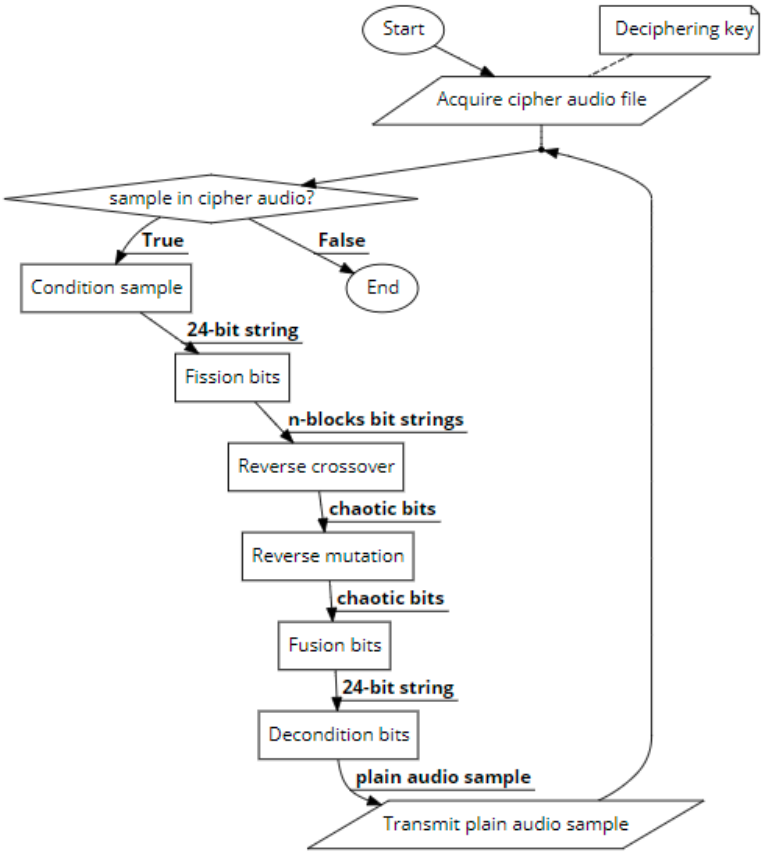


Figure 2. Flow diagram of deciphering process

4.1. Enciphering Process

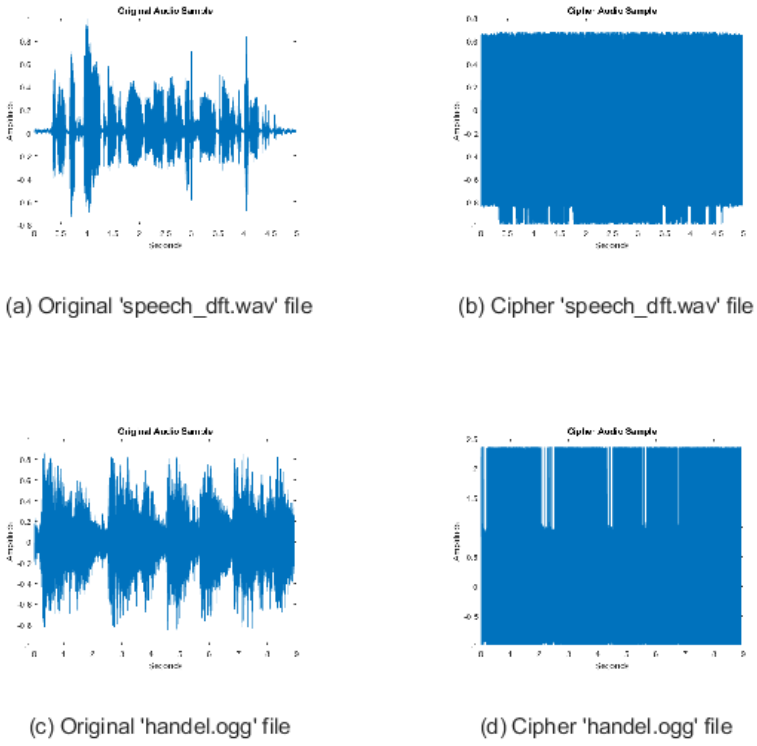
Input: Original audio sample = -0.000092, encryption keys (c = 7 (single-point crossover), f = 2, m = 5, cp1 = 1, cp2 = 10000000, fo = (2, 1)).
Output: Cipher audio sample.
1. Conditioning process
   xp = (-0.000092 + cp1) × cp2 = 100110001001001011101000
2. Fission audio bit string into 2-blocks
   sp1 = 100110001001
   sp2 = 001011101000
3. After a single-point crossover operation from bit 7
   sp1 = 100110**101000**
   sp2 = 001011**001001**
4. After mutation of bit 5
   sp1 = 10010**0**101000
   sp2 = 00100**0**1001001
5. After fusion chaotic 2-blocks bit strings using fo(2, 1)
   xc = 001001001001100100101000
6. Decondition of xc into cipher audio sample
   xc = (2398504 / cp2) - cp1 = -0.760100

4.2. Deciphering Process

Input: Cipher audio sample = -0.760100, encryption keys (c = 7 (single-point crossover), f = 2, m = 5, cp1 = 1, cp2 = 10000000, fo = (2, 1)).
Output: Original audio sample
1. Conditioning process
   xc = (-0.891200 + cp1) × cp2 = 001001001001100100101000
2. Fission cipher audio bit string into 2-blocks
   sp1 = 001001001001
   sp2 = 100100101000
3. After reverse single-point crossover operation from bit 7
   sp1 = 001001**101000**
   sp2 = 100100**001001**
4. After reverse mutation of bit 5
   sp1 = 0010**1**1101000
   sp2 = 1001**1**0001001
5. After fusion of reordered 2-blocks bit strings using fo(2, 1)
   xc = 100110001001001011101000
6. Decondition of xc into original audio sample
   xc = (9999080 / cp2) - cp1 = -0.000092

4.3. Application

To test the working of the proposed algorithms, various audio files were simulated using MATLAB. Figure 3 shows the graphs of original and corresponding cipher audio samples. The illustration clearly shows that each original audio sample has been completely altered.



(a) Original 'speech_dft.wav' file (b) Cipher 'speech_dft.wav' file

(c) Original 'handel.ogg' file (d) Cipher 'handel.ogg' file

Figure 3. Graphs of plain and cipher audio samples

5. PERFORMANCE ANALYSIS

Data obtained from simulation processes were analysed to measure the strength and viability of the proposed algorithms. This section details the analyses that were conducted.

5.1. Key Sensitivity Analysis

Enciphering and deciphering algorithms should produce different results for every key in the given keyspace. Thus, no two should produce the same plain or cipher audio. To this end, any slight variables in decryption key parameters should lead to different cipher audio rather than the recovery of original audio samples. The results in Figure 4 show that the proposed method yields different results for every key in the keyspace. In this simulation, only m is slightly changed from 2 (for encryption) to 3 (for decryption). This confirms that the scheme is sensitive to slight variations in the cipher key parameters. Hence, the cipher keys are unique.

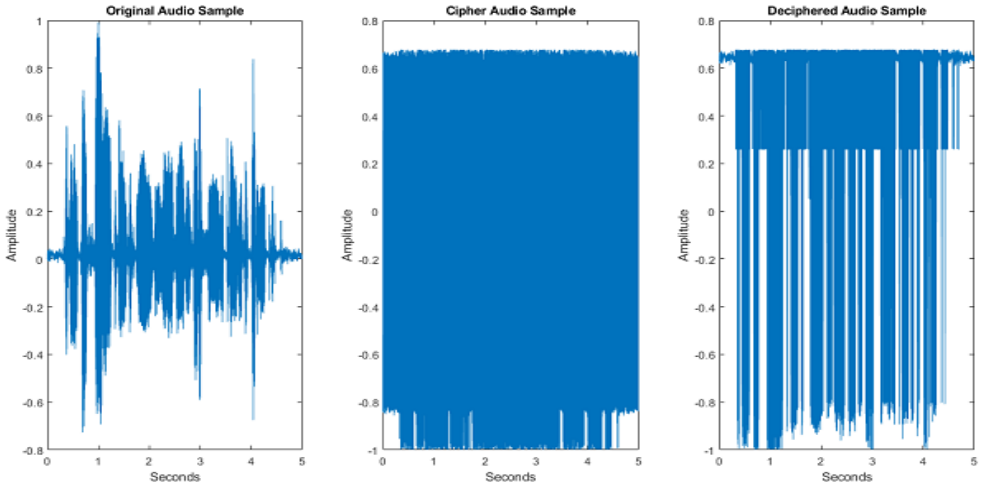


Figure 4. Graph of the failed recovery process

5.2. Keyspace Analysis

Cryptographic schemes are robust against brute-force attacks when the keyspace is sufficiently large. The proposed algorithm uses five (5) – key parameters which together give a keyspace of 2^{128}. This is an improvement as compared to the keyspace of [4].

5.3. Error Rate

Good decryption algorithms should recover the original audio sample at the authorized receiver's end with very little or no variable from the plain (unencrypted) form. The error rate obtained between the plain and recovered audio sample was calculated as zero (0). However, the error rate computed between plain and cipher audio samples was -0.128965.

5.4. Audio Degradation

Since perceptual video encryption leaves some visible aspects of the original video after encryption [2, 3], the audibility of embedded audio samples should also be controlled during encryption. It is revealed that audio degradation increases as the crossover point between blocks increases. Table 1 shows the standard deviations of different cipher samples of the

"speech_dft.wav" file. This indicates the cipher audio defer greatly from the plain form when the crossover point between blocks is bigger.

Table 1. Standard deviations of cipher audio samples.

Crossover point	Standard deviation
2	0.444348075
4	0.484717425
6	0.484561465
8	0.484642411
10	0.484669622

5.5. Runtime

The average runtimes for enciphering and deciphering an audio sample were computed to be 2.550540×10^{-4}secs and 2.281230×10^{-4}secs respectively. For a 4secs *"speech_dft.wav"* file with 110033 audio samples, the average runtimes were 28.064357secs and 25.101058secs respectively for enciphering and deciphering processes of all the samples. These times are considered good for the audio samples under consideration and bit-level security. However, the times realized are bigger than the work of Alhassan et al. [4].

6. CONCLUSIONS

The rising demands for secure communication call for drastic measures to allay fears of communication parties. This paper proposes novel audio enciphering and deciphering techniques that use enhanced GA to secure embedded audio signals in digital video. The technique successfully blinds conditioning, fission, fusion, and GA operators (selection, crossover, and mutation) to encipher and decipher audio files.

Apart from providing a GA audio encryption, the proposed technique also preserves data size and enhances security. Data analyses have shown that the proposed technique has a good keyspace and sensitive to slight changes in key parameters.

Notwithstanding these successes, the proposed technique was found to be ineffective when;

- the crossover point between blocks is 1, and
- the mutated bits are same (i.e. when both are "1"s or "0"s)

Future work will investigate the application of the proposed technique in steganography.

REFERENCES

[1] Al-kateeb, Zeena N. & Mohammed, Saja J., (2020) "A novel approach for audio file encryption using hand geometry", *Multimed Tools Appl*, Vol. 79, pp19615–19628.

[2] Chloe, Albin, Dhruv, Narayan, Ritika, Varu & V., Thanikaiselvan, (2018) "DWT Based Audio Encryption Scheme", *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp920-924.

[3] Naskar, Prabir Kuma, Paul, Soumya, Nandy, Dipta & Chaudhuri, Atal, (2019) "DNA Encoding and Channel Shuffling for Secured Encryption of Audio Data", *Multimed Tools Appl*, Vol. 78, pp25019–25042.

[4] Salamudeen, Alhassan, Mohammed Muniru, Iddrisu & Mohammed Ibrahim, Daabo, (2019) "Securing audio data using k-shuffle technique", *Multimed Tools and Appl*, Vol. 78, No. 23, pp33985-33997.

[5] Krishna, Bhowal, Debnath, Bhattacharyya, Anindya, Jyoti Pal & Tai-Hoon Kim, (2011) "A ga based audio steganography with enhanced security", *Telecommunication Systems*, Vol. 52, No.4, pp2197-2204.

[6] Ayan, Chatterjee & Nikhilesh Barik, (2018) *A new data hiding scheme combining genetic algorithm and artificial neural network*, Advances in Computational Intelligence and Robotics Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired Metaheuristic Algorithms, pp94-103.

[7] Guangfeng, Cheng, Chunhua, Wang, & Cong Xu, (2020) "A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing", *Multimed Tools and Appl*, Vol. 79, No. 39, pp29243-29263.

[8] Mohammed, A.f. Al Husainy, (2006) "Image encryption using genetic algorithm", *Information Technology Journal*, Vol. 5, No. 3, pp516-519.

[9] Rajat, Jhingran, Vikas, Thada & Shivali Dhaka, (2015) "A study on cryptography using genetic algorithm", *International Journal of Computer Applications*, Vol. 118, No. 20, pp10-14.

[10] Ankit, Kumar & Kakali, Chatterjee, (2016) "An efficient stream cipher using genetic algorithm" *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp2322-2326.

[11] Melanie, Mitchell, (1998) *An introduction to genetic algorithms*, MIT.

[12] Muhammad Irshad, Nazeer, Ghulam, Ali, Noor, Ahmed, Rakhi, Bhatra, Raheel, Ahmed, & Muhammad, Ismail, (2018) "Implication of genetic algorithm in cryptography to enhance security", *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 6, pp375-379.

[13] Peter, Awon-natemi Agbedemnab, Edward, Yellakuor Baagyere & Mohammed Ibrahim, Daabo, (2019) "A Novel Text Encryption and Decryption Scheme using the Genetic Algorithm and Residual Numbers", *Proceedings of 4th International Conference on the Internet Cyber Security and Information Systems 2019, Kalpa Publications in Computing*, Vol. 12, pp20-31.

[14] Manisha, Rana & Rohit Tanwar, (2014) "Genetic algorithm in audio steganography" *International Journal of Engineering Trends and Technology*, Vol. 13, No. 1, pp29-34.

[15] Sobin, C. C. & Manikandan, V. M., (2019) "A secure audio steganography scheme using genetic algorithm", *2019 Fifth Proceedings of the IEEE International Conference on Image Information Processing (ICIIP)*, pp403-407.

[16] Mazdak, Zamani, Hamed, Taherdoost, Azizah A. Manaf, Rabiah, B. Ahmad & Akram M. Zeki, (2009) "Robust audio steganography via genetic algorithm", *2009 International Conference on Information and Communication Technologies*, pp149-153.

AUTHORS

Salamudeen Alhassan obtained his B.Sc. degree in Computer Science in 2009 from the University for Development Studies, Tamale, Ghana. He received his M.Sc. and Ph.D. degrees in Computational Mathematics in 2013 and 2019 respectively from the same university. His research interests are in the areas of cryptography, algorithms, image/video processing and software engineering.

Gabriel Kofi Armah obtained his B.Sc. in Computer Science in 1997 from the Kwame Nkrumah University of Science and Technology, Ghana. He received an MBA degree in Management Information Systems from the University of Ghana, Ghana in 2003 and a Ph.D. degree in Computer Science and Technology in 2015 from the University of Electronic Science and Technology, China in 2015. His research interests are in the areas of Machine Learning, Artificial Intelligence and Software

Issah Zabsonre Alhassan is a Ph.D. student in Computer Science at the Kwame Nkrumah University of Science and Technology, Kumasi, Ghana, where he also obtained his M.Sc degree in Information Technology in 2019. He obtained his B.Sc degree in Computer Science in 2012 from the University for Development Studies, Tamale, Ghana. His research interests are in the areas of system engineering, algorithms, cryptography and image/video processing