iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4774

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

Digital Audio Encryption using Media Type Conversion and Genetic Algorithm based Image Encryption

Deveki Nandan Shukla[1], Jitendra Kurmi[2], Deena Nath[3]
[1, 2, 3]*Department of Computer Science, Babasaheb Bhimrao Ambedkar University.*

Abstract: ***Recently, security of audio data has pulled much attention as any other security systems. Security becomes obvious, especially when these audios are being sent through a communication network. An audio encryption technique transforms an audio to another audio, which is very difficult to interpret. In this issue, we are proposing an encryption system methodology for WAV audios, which is the most widely used audio format. This Algorithm is based on the characteristics of genetic algorithms. Also in this, we introduce an ambiguity factor to our audio encryption system where we can send or store our audio in form of an image or embedded in another image by steganography, which makes it very hard to guess about the kind and nature of data stored in the file. Performance analysis of proposed algorithm confirms that it has good key sensitivity, statistical character and can also is resistant to brute-force attacks, differential attacks, plaintext attacks and entropy attacks quite efficiently.***
Keywords: ***Genetic Algorithms, Secret Key, Digital Audio protection, Encryption, WAV, audio to image conversion***

I. INTRODUCTION

Audios are at the heart of the music industry as well as in many other fields. In the last years, the development of multimedia security techniques has attracted the attention of researchers from several fields of knowledge. This is mainly due to the increasing ease to share a digital image, video, and audio through communication networks [1]. Now, it is necessary to provide high security to digital audio to propagate through communication medium without losing any data, its confidentiality, its integrity, and its authenticity. Encryption methodologies, which are used in existing encryption systems are usually, based on the common methods like RSA, AES, DES etc., These old encryption methodologies are not useful for implementing practical digital audio encryption. It is so due to some of the unique characteristics of audio such as negative floating-point values. Only a few audio encryption systems, particularly designed for WAV audio, are seen in the research. Digital audio such as music recorded audio, recorded confessions and secret messages are widely used. While some people are not knowing about the threat of leaking of audio data, that could create moments with intense embarrassment, loss of work and money. Therefore, there is a necessity to protect important audio from falling into the wrong hand. Existing audio encryption schemes are not preferable due to following reasons.

A. Low encryption rate
B. Lossy in nature,
C. Statistical attacks are successful
D. Brute force attack is easy on audio files.

Thus we need to devise a secure, lossless and efficient encryption mechanism, especially for audio archival or for sending copyrighted music over the internet, where large number digital audio is transmitted across public/open networks.
In [7] Narendra K. Pareek told much about the genetic algorithm. It is biologically motivated computing particularly genetic algorithms have dragged much consideration of researchers to develop a strong and robust encryption algorithm. They fall into the category of evolutionary algorithms. These algorithms are used to search solutions of various optimization problems. They utilize methodologies based on biological evolution such as mutation, crossover, selection, and inheritance [2][3]. the aim of all this is just to replicate the working of nature and to achieve randomness of nature, in which the population readjusts itself according to its environment using natural selection process and by the behaviour of the natural system. It also meant that the chromosome propagates by the removal of undesired characteristics. Genetics has shown its ability in optimization problems [4] and thus is getting popular in the field of encryption and optimizing algorithms.

Only a handful of research papers for audio-based encryption is available. [6] proposed an audio shuffle-encryption algorithm to encrypt an audio using a key. The security of this algorithm comes from the complexity of the shuffle operations where as they are prone to brute force attacks.

Further, Section II explains the methods and detail of proposed algorithm. In section III, we have mainly focused on analysing security metrics and performance metrics of the algorithm.

II. PROPOSED AUDIO ENCRYPTION ALGORITHM

The sound is a pressure wave having pressure variance in an elastic medium. The variance propagates in the medium in the form of compression and rarefaction. Compression occurs when pressure is higher than the atmospheric pressure and rarefaction occurs when the pressure of the propagating wave is less than the atmospheric pressure. Just like that, a WAVE file only represents the sampled sound waves, which happen to be above or below the equilibrium or here atmospheric pressure. In genetic algorithms, chromosomes can be encoded in the form of binary strings, e.g.,10010011 [3]. In a Genetic Algorithms based encryption system, a binary string generally represents values that correspond to the chromosomes.

The proposed algorithm to encrypt an audio is derived from a genetic algorithm based grayscale image encryption algorithm []. Originally, this algorithm is for only grayscale images, which have been modified to encrypt RGB images. Audio contains positive and negative floating-point values of amplitude, which are needed to be mapped into the image, which accepts an only non-negative integer in range 0-255. Further changing audio into the image and repeating this process may cause loss of data, which is needed to be addressed.

Thus we proposed an idea, to the used encrypted image as representative of encrypted audio. This encrypted image contains encrypted audio, and this image can be sent over a network or used in archival of data. Original audio can only be extracted from this encrypted image by decrypting it using right key. Converting a audio into image also increase our capability to encrypt data in a more complicated manner as audio is a one-dimensional data, whereas the image is two dimensional (grayscale) and three dimensional (RGB image)

From here onwards, we will be discussing the mechanism and processes, which are used, in our noble audio encryption system.

A. Media type Conversion

As already mentioned a wave file consists of series of positive and negative floating-point values over its entire range of samples usually between -1 to 1. Two different matrixes M_1 and M_2 of size HxW (such that no. of boxes in any particular matrix should be greater than no. of values in WAV audio) is created to store positive and negative values respectively, as follow,

```
k=1
for i=1: H
    for j=i:W
        if(audio(k,1)>=0)
                M1(I,j)=audio(k,1)
                M2(I,j)=0
        else
                M2(I,j)= (-1) * audio(k,1)
                M1(I,j)=0
        end
        k=k+1
end
end
```

An RGB image consists of three different 2-dimensional matrices for each red green and blue color component. Each box in matrix store the value of the color intensity of respective color at the respective pixel position. This intensity value is stored in form of positive integer value ranging between 0 to 255. Values in matrix M_1 and M_2 are floating point no. between 0 and 1 which are converted into their equivalent positive integer no. in range 0 to 255. M_1 serves as RED whereas M_2 serve as Green layer in RGB image, an identity matrix of size same as M_1 and M_2 serves as a blue layer for this algorithm. On both red and green layer, the encryption is performed separately. Blue layer is left untouched in order to avoid any unwanted computation.

B. Recombination

Now as our audio is transformed into an image we can now work on this image as image encryption algorithm. In image processing, pixels are considered chromosomes. A valid population consists of large no. of chromosomes. In recombination process, existing population is used to generate a new population of the same size as the original population. Correlation between pairs of adjacent pixels of the image is being used a fitness function. Each chromosome is replaced with the new chromosome by mixing the properties of adjacent chromosomes. Here Mixing is done using exclusive-OR operation as follows.

for i=1:h
  for j=1:w
    $X_{i,j} = xor(X_{i,j},X_{i,j-1})$
  end
end

where $X_{i,0} = \begin{cases} 0 & j = 1 \\ X_{i-1,w} & j > 1 \end{cases}$, h and w are height and width of the image.

C. Population Generation

In this process, using some repeated procedure on the existing population we generate a new population which is known as generation [2]. To generate the initial population from the plain image, the image is divided into several non-overlapping, square sub-images of sizes N × N. Columns of square blocks are rearranged on the order of the encryption key. After that, values in each square block is transposed and the resultant square block is converted into its equivalent sub-images of size N × M (Fig.1).

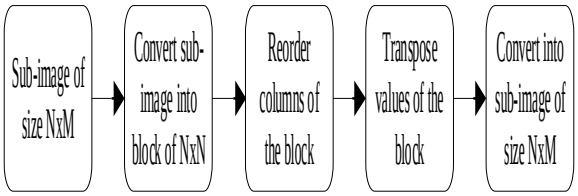


Figure 1: Initial population generation process

D. Crossover process

Crossover approximately imitates biological recombination process between two chromosomes of an organism. In genetic algorithms, the crossover is a genetic operator that helps in combining two chromosomes to create a new pair of chromosomes also known as offspring. These offspring's take place of their parents. This also implies that now there is no need for the selection operator as well as the fitness function. Crossover can be classified into following categories—single point, two points, multiple points, uniform, arithmetic, etc., [5]. In [7] The basic method is a one-point crossover, in which a random crossover point or locus is first chosen, and then the second part of the first parent is recombined with the first part of the second parent to create the first child and the first part of the first parent is recombined with the second part of the second parent to create the second child as shown in Table 1.

Chromosome 1	0 1 0 | 1 1 0 0 1
Chromosome 2	1 1 0 | 0 1 0 1 1
Offspring 1	0 1 0 | 0 1 0 1 1
Offspring 2	1 1 0 | 1 1 0 0 1

Table 1: One point crossover technique. Symbol |

indicates the locus point from where the chromosome will be dissected. To find the pairs on which the crossover operation to be done, we choose the pair of chromosomes in sequence from the zigzag path as shown in Figure 2.

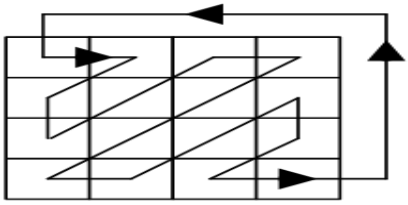


Figure 2: Crossover zigzag path to find a pair of chromosomes

Existing population is divided into various non-overlapping groups of the size N ×N. The starting location for selecting chromosomes in a group is made key dependent completely. For this purpose pairs of adjacent sub-keys are formed, i.e., $(k_1,k_2),(k_3,k_4),...,(k_{31},k_{32})$. In the first generation, for the first, second, third ... groups, sub-key pairs $(k_1,k_2),(k_3,k_4),(k_5,k_6),...$, are used respectively, as the location of the starting chromosome. When all sub-key pairs are consumed, it is initiated again from the first sub-key pair (k_1,k_2). In this process, we have used different types of crossover operations—single point, two points, and multiple points. Selection of crossover operation depends on the session-key (K_i) which is used as an encryption key in the algorithm. For first, second, third ... crossover, session-key K_1, K_2, K_3,..., respectively, are used. When all session-keys are consumed, it is initiated again from the first key K_1. Table 2. shows two chromosomes with a session-key along with the generated offspring.

Chromosome 1	0 1 0 1 1 0 0 1
Chromosome 2	0 1 1 1 0 0 1 1
Session key	**1 1** 0 0 **1** 0 **1 1**
Offspring 1	0 1 0 1 0 0 1 1
Offspring 2	0 1 1 1 1 0 0 1

Table 2: Bold digit indicates positions used for crossover using session key.

E. Mutation

Mutation is performed on the population after completion of crossover operation. This guarantees to prevent original data to remain in the population. Mutation randomly changes an offspring from what it was after the crossover. It is a genetic operator, which converts one or more bits in a chromosome. In [5] various types of mutation operations are discussed which are as follows,

1) Flipping of bits,
2) Boundary mutation,
3) Non-uniform mutation
4) Uniform mutation,
5) Gaussian mutation,

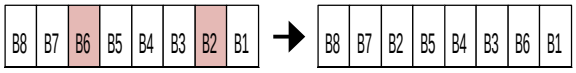


Figure 3: Mutation Operation

In this algorithm, [3] we have used swapping mutation operation, which falls into the category of boundary mutation, shown in Fig. 3. The mutated chromosome then replaces the original chromosome. In mutation operation, the gene of the chromosome that should be mutated depends on the sub-key (k_i) used in the algorithm. The genes, to be swapped, corresponding to a particular sub-key differs in each generation. For example, in the first generation, the gene at b2 is swapped with gene at a b6 and this is corresponding to sub-key value 4. In other generation, for sub-key value 4, a gene at b1 is swapped with gene at b8. Table 3 gives the swapping operations for each generation.

Swap Position		$B_1 \leftrightarrow B_{i+5}$				$B_2 \leftrightarrow B_{i+5}$				$B_3 \leftrightarrow B_{i+5}$				$B_3 \leftrightarrow B_{i+5}$			
Key (K_i)		0	4	8	C	1	5	9	D	2	6	A	E	3	7	B	F
Value of I in generation	1	3	0	1	0	3	0	3	0	1	0	1	0	3	0	3	0
	2	0	1	2	1	0	1	0	1	2	1	2	1	0	1	0	1
	3	1	2	3	2	1	2	1	2	3	2	3	2	1	2	1	2
	4	2	3	0	3	2	3	2	3	0	3	0	3	2	3	2	3

Table 3: Mutation swapping operation table

F. Complete encryption algorithm

1) New audio is recorded or existing audio is selected, and then converted into an RGB image.
2) Every second-pixel value in the red component is swapped with every second-pixel value in a green component of RGB image,
3) A secret random key of 128-bits size is generated. This key is referred as sub keys. This subkey is divided further into blocks of 8-bits each referred as session keys.
Sub-key = $k_1k_2k_3$ .....k_{32} (Sub-key in hexadecimal),
Session-key = $K_1K_2K_3$ .....K_{16} (session Key in ASCII),
4) Step 4,5,6,7,8 are performed on both red and green components of the RGB image separately,
5) Pass image through a recombination process

6) Set U=3 and V=8
7) For i = 1:4 do
a) Seed = $\sum_{j=0}^{3} K_{(4*i-3)+j}$
b) Set N = U×V
c) Divide the resultant image into square non-over lapping sub-images of the same sizes of N ×N.
d) Each sub-image passes through the population process as discussed in Sect. 2.3.
e) Set U = U+1
Endfor
8) Set N =16.
For i = 1:4 do
a) Divide the resultant image into squared non-overlapping sub-images i.e., S_1, S_2,..., S_n of the same sizes of N × N. Total number of blocks (n) depends on the size of the image to be encrypted.
b) Set x =1 and y = 1
c) For p =1:n do
d) In this step, sub-image (Sp) passes through the crossover process as discussed in Sect. 2.3 with a sub-key pair (k_x, k_y).
e) the next step is the mutation operation. Resultant sub-image passed through the mutation process as discussed in Sect. 2.4
f) Set x = x+1and y = y+1 in order to select next pair of sub-key (k_x,k_y)
g) Set N = N +8 Endfor
9) Resulting block is written as a file.
10) In the end, new red green blue components are mixed to make an encrypted RGB image.
11) This red component and green component are used to reconstruct WAV format audio.

III. PERFORMANCE AND SECURITY ANALYSIS

Encryption scheme must resist all kinds of known attacks such as statistical, cryptanalytic, and brute-force attacks. In this context, we will discuss the security analysis of this novel algorithm using key-space, statistical, and key sensitivity analysis, etc., in order to prove the robustness and effectiveness of this algorithm. For this purpose, we have implemented the recommended audio encryption methodology in MATLAB and analysis of audio have been done using DSP application tools of MATLAB.

A. Statistical Analysis

Statistical attacks are employed to find flaws in any encryption algorithms. Here we have performed various statistical analyses such as waveform analysis, correlation analysis, PSNR analysis, to explain the effectiveness of the algorithm.

1) *Waveform analysis:* Waveform analysis represents amplitude distribution in an audio by describing their value relative to each pressure level. We have studied the waveform of many audio pairs with different contents. Some of those waveforms are shown in Fig. 4. In Figure 4(a), we have shown an audio and its encrypted audio produced using the key '78C84BE299C9916C08694A7846504C5A' which is shown in Figure 4(b). Similarly, the waveform of another audio and its encrypted audio are shown in Figure 4(c)-(d), and Figure 4(e)-(f). From both waveforms, it is clear that most of the amplitude is in the range -1 to 1 (Fig. 4a) in the original audio whereas amplitude is almost uniformly distributed (Fig. 4d) in the encrypted audio. The encrypted audio waveform (Fig.4d), approximated by a uniform distribution, is significantly different from the Original audio waveform (Fig.4c). We obtain the similar results for all other audio samples the with various combinations of the secret keys. A uniform distribution of amplitude is the good quality of the encryption scheme. This also means that this encrypted audio will not provide any hint to the attacker to operate any statistical attack on this audio encryption scheme [7]

2) *Correlation Analysis:* A lower correlation value between the plain audio and its cipher indicates less resemblance between them, which provides more resistance to attacks. An extensive study of the correlation between original and its encrypted audio is done by calculating their correlation coefficient. The key used in encrypting audio for all test cases is '78C84BE299C9916C08694A7846504C5A'.

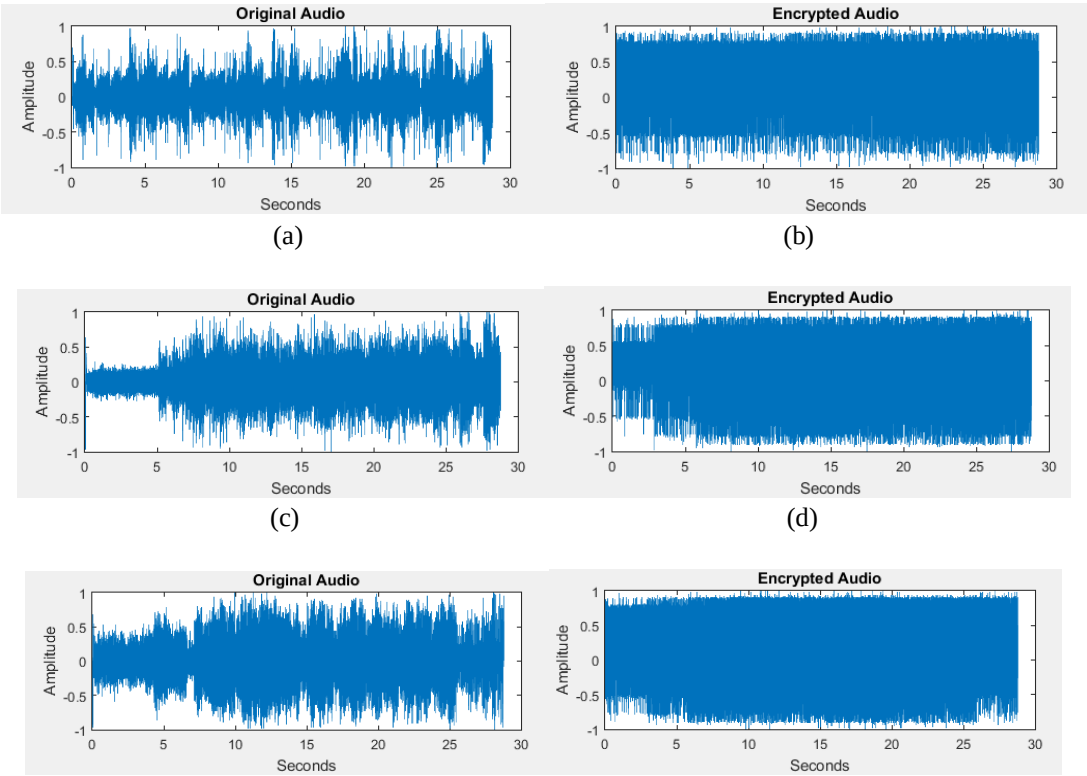


Figure 4: (a) Original audio sample 1, (b) Encrypted audio of (a), (c) Original audio sample 2 , (d) Encrypted audio of (c), (e) Original audio sample 3, (f) Encrypted audio of (d)

Results of some audios of various contents are shown in Table 3. The average value of the correlation coefficients is almost equal to zero which means that that the original audio is almost independent of the encrypted audio.

Audio	Cipher Audio	Correlation Coefficient
A	B	0.0167
C	D	0.0276
E	F	0.0280
Average		0.02095

Table 3: Correlation coefficient between audio and their corresponding cipher audio

3) Peak signal-to-noise ratio (PSNR) : Low PSNR values indicate a high level of noise in the cipher audio, which indicates more resistant to attacks. PSNR is computed by Eq.

$$PSNR=10\log_{10}(peakval^2/MSE) \quad \text{..... (1)}$$

Table 4 shows PSNR values calculated for all encrypted audio files,

Audio	Cipher audio	PSNR	SNR
A	B	9.1893	-5.6614
C	D	8.5988	-5.6950
E	F	6.9280	-4.8565

Table 4: PSNR value of audio and cipher audio pair

B. Key Sensitivity Analysis

A perfect encryption algorithm should be extremely sensitive to the key used. Even a change in a single bit should produce an extremely distinctive encrypted audio. We have tested key sensitivity for few digital audios decrypted with the slightly modified key. We calculated correlation coefficient when encrypted audio is compared to audio decrypted using a slightly different key. One such example is discussed below.

The encrypted audio (Fig. 4a) is decrypted with a slightly modified original key '88C84BE299C9916C08694A7846504C5A'. The resultant decrypted audio waveform is shown in Figure. 5a.

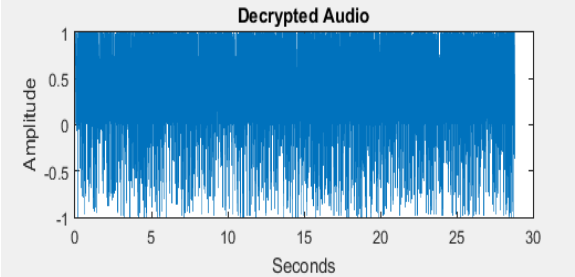


Figure 5: (a) First decrypted audio

The encrypted audio (Fig. 4a) is decrypted with a slightly modified original key '78C84BE299C9916**2**08694A7846504C5A'. The resultant decrypted audio waveform is shown in Fig. 5b.

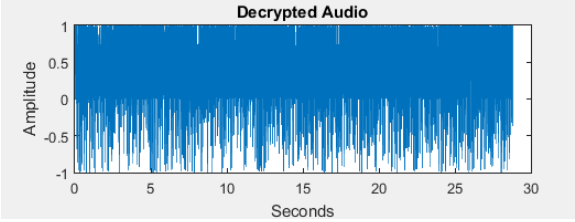


Figure 5: (b) Second decrypted audio

The encrypted audio (Fig. 4a) is decrypted by slightly modified original key '78C84BE299C9916C08694A7846504C5**B**'.The resultant decrypted audio is shown in Figure 5c.

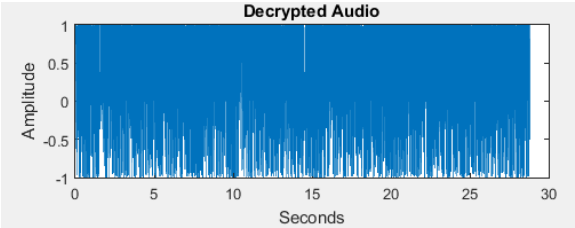


Figure 5: (c) Third decrypted audio

With a very small change in the key, the attacker is unable to find any hint about the original audio from the decrypted audio. To compare the decrypted audio, we have calculated correlation coefficient among the decrypted audio and the results are shown in Table 5, which is close to zero.

Cipher audio with slightly changed key	Comparision Cipher-audio with slightly changed key	Correlation coefficient
a	b	0.0751
a	c	0.0536
b	c	0.0650

Table 5: Correlation coefficient of the decrypted audio

We can now come to the conclusion that no one can find any clue about the original audio even if there is a little change in the key. Above study has shown that the decryption of the encrypted audio with the wrong key will not disclose any information about the original audio.

C. Differential and known/chosen Plaintext Attack

As per the well-known Kerckhoffs's principle, for assuring complete security of an encryption algorithm, every detail of algorithm must be known to the public, except the key used. Due to this, an attacker may use cryptanalysis. He may choose an audio and encrypt it, and then search for a relationship between the key and decrypted image, to observe the changes in decrypted audio when he changes the key by a bit or two.

The net amplitude change rate (NACR) is the index which is used to test the capability of algorithm to stay robust against any chosen plain text attack. The NACR calculates the percentage of sites where amplitude have changed between two encrypted audios. We took two encrypted audios, X and Y, which are similar. In Y we make change in absolutely one site of amplitude. We changed the amplitude of audio Y at the first position only. The $D_{(i,j)}$ is calculated from $X_{(i,j)}$ and $Y_{(i,j)}$. if $X_{(i,j)} = Y_{(i,j)}$ then $D_{(i,j)} = 1$, otherwise $D_{(i,j)} = 0$. The NACR is defined by the following equation.

$$NACR = ((\sum_{i,j} D_{(i,j)}) x100)/l \% \quad \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

where, l is the length of the cipher audio. We have calculated NACR for a considerable amount of audios using our encryption scheme. Resulting average NACR was 99 % which thereby proves that the encryption scheme is very sensitive to the plaintext/input. For further analysis, we chose an audio (Fig. 4c) and encrypted it with all zeros keys, i.e., 00000000000000000000000000000000'. Then the same audio (Fig.4c) was further encrypted with the slightly different keys,
i.e.,'10000000000000000000000000000000','00000000000000010000000000000000',and'00000000000000000000000000000001'. We then analyzed these encrypted audios to see any relationship among them. For this, we compared each audio to check the no. of amplitude values that remained unaltered after encryption in each case. It turns out to be hardly 1 % of amplitude remained unaltered and no patterns were seen in encrypted audio. he algorithm that we have proposed in this paper, uses feedback mechanism, which resists all known/chosen plaintext attacks. The result of the recombination process is an audio which is encrypted. After population generation operation, the crossover and mutation operations are executed which adds further complexity to the encryption method.

D. Audio Quality Criterion

A very practical test to check the real capability of the algorithm by using mean square error (MSE). A smaller value of MSE denotes the high audio quality of decrypted audio. In contrast, a higher value of MSE means the corruption of data. The mean square error between a audio, X, and Y, is the squared norm of the difference divided by the number of elements in the audio.

$$MSE = (||X-Y||^2)/N \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \ldots\ldots\ldots\ldots\ldots\ldots...(3)$$

We have calculated MSE for few WAV audios to be very close to zero (4.40400e-07). Hence, this algorithm comes under the category of a lossless audio cipher.

E. Encryption Speed

This algorithm is the totally computation based algorithm and thus totally depends on the computing power of the CPU or GPU (in case of GPU computation). Time analysis had been performed on a laptop with Intel(R) Core(TM) i5-3230M CPU at 2.60 GHz. the average rate of encryption of the submitted algorithm is about 60 KB/s. For large datasets algorithm can be easily made for GPU enabled computation platforms, which can boost this algorithm, by the average of 700 KB/s of encryption rates depending on computation power of GPU.

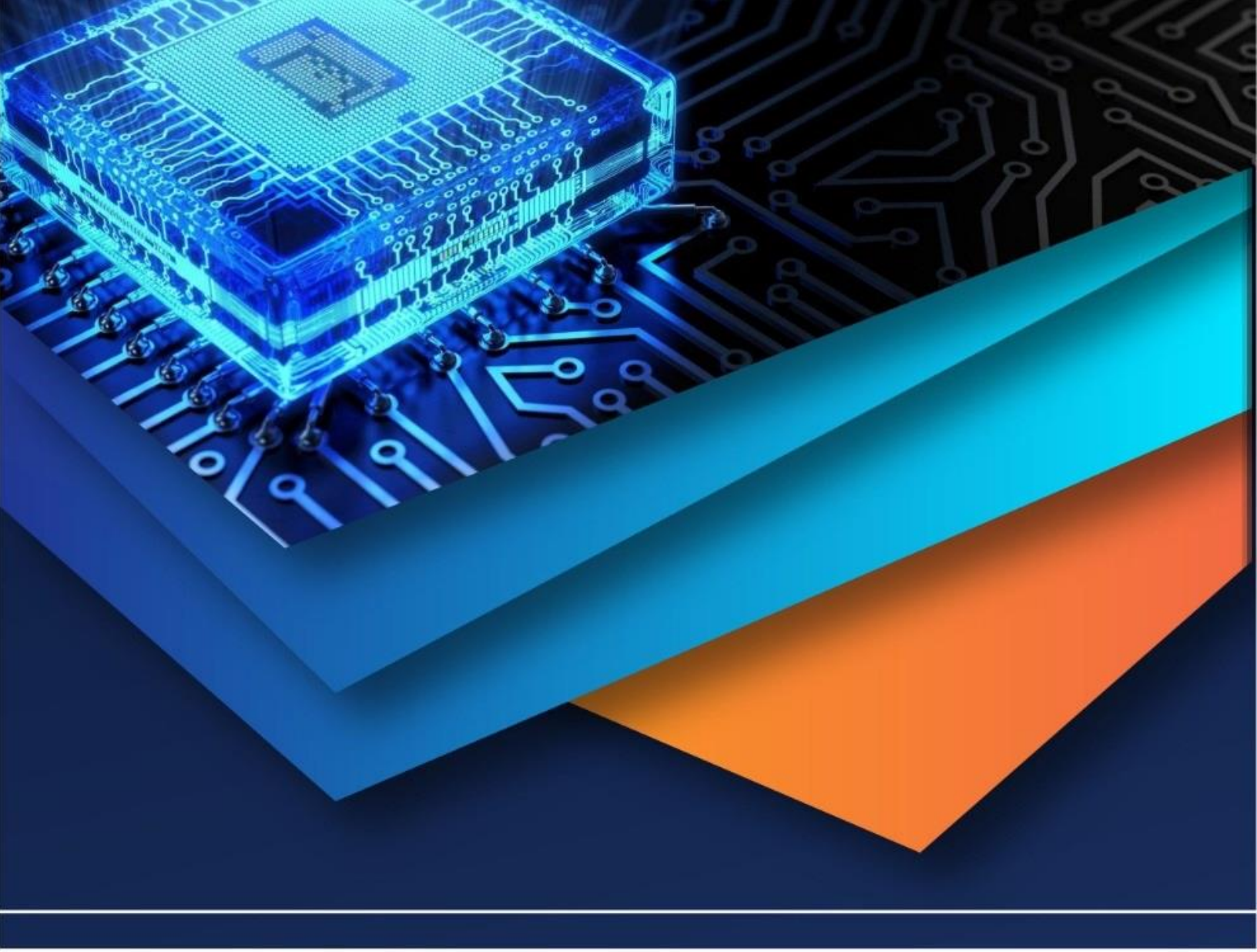
IV. CONCLUSION AND FUTURE SCOPE

In this paper, a new approach based on characteristics of Genetic Algorithms for Digital WAV audio protection is proposed. Different processes included in the algorithm are completely key-dependent and the repetition of crossover and mutation processes makes the robustness of the algorithm. Conversion of audio into an image opens the scope for audio data to be safeguarded with more applications like steganography. Using this method audio can also be sent in image formats making attacker harder to track exact audio file thus adding ambiguity as another level of security. Results obtained from the specific security analysis (correlation coefficient, key sensitivity, NACR, MSE, etc.) are evidence of the robustness, and a high degree of security and thus can be used for encryption of digital WAV audio for data archival as well as sending data through the communication network. In future this

algorithm may be boosted using GPU computing techniques which will make these complex computations to finish way faster than now.

REFERENCES

[1] Shih FY (2012) Multimedia security: watermarking, steganography and forensics. CRC Pres
[2] Haupt RL, Haupt SE (2004) Practical genetic algorithms. Wiley, New Jerse
[3] Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridg
[4] Tsang PWM, Au ATS (1996) A genetic algorithm for projective invariant object recognition. In: IEEE TENCOM Conference proceedings on Digital Signal Processing Applications, pp 58–6
[5] Mishra S, Bali S (2013) Public key cryptography using genetic algorithm. International Journal for Recent Technology Engineering, Volume 2, pp 150–15
[6] Abdelfatah A. Tamimi and Ayman M. Abdalla, "An audio shuffle-encryption algorithm," World Congress on Engineering and Computer Science Volume 1, October 201
[7] Narendra K. Pareek and Vinod Patidar, "Medical image protection using genetic algorithm operations," Soft Computing (2016), volume 20 pp 763–772



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY