A Protected Data Transfer through Audio Signals by Quantization combined with Blowfish Encryption: The Genetic Algorithm Approach

Rashmi P Shirole[1], Shivamurthy G[2]

[1] Department of Information Science & Engineering, NMAM Institute of Technology, Nitte, Udupi, India
[2]Department of Computer Science & Engineering, Visvesvaraya Technological University, Muddenahalli, India

[1]rashmin@nitte.edu.in, [2]kgshivam@gmail.com

Abstract: Information is actually very potent. It is very common for data to be transferred over the internet, and everyone is responsible for ensuring its security. Data loss, data manipulation, and theft of confidential information are all effects of security events. Information security is a set of practices and protocols that help to secure this information. Such sensitive data can be secured using a variety of methods. Information security includes two important subfields: cryptography and steganography. With the help of cryptography and steganography, information is altered into an unintelligible state and made secret respectively. The purpose of this chapter is to preserve impenetrability and improve invulnerability. The objectives of this chapter are to be recognized by enhancement of Blowfish which is believed as highly secure algorithm; the implementation of chaotic sequence-quantization method for audio samples. The proposed work's performance is contrasted with that of the existing blowfish method and standard audio LSB algorithm. The following criteria shows the demonstration of analysis of the work done – Entropy values, Avalanche effect, Attack scenario, Execution time, PSNR value, Embedding capacity, Structural similarity index etc. The suggested system is the most effective method for intensifying protection and preserving the high caliber of the original entity.

Keywords: Data Hiding; Symmetric Encryption; Vernam Cipher; Chaotic Sequence; Quantization; Least Significant Bit; Avalanche Effect; Entropy Value; Attack Scenario; PSNR.

1. Introduction

Numerous internet-based applications, such as accommodations, tickets and other services online, bill payment via email, e-banking and social networking; have emerged in the last few years. A major discipline is cryptography, while another is steganography, which provides confidentiality, Integrity, Authenticity to these applications.

The use of cryptography involves disassembling the information into a form that can only be read and processed by the intended user. Messages or information are encoded with encryption so that only authorized parties can read them, and cipher text is decoded to plaintext with decryption. Depending on the mode of operation, keys and processing of text, there are many cryptography algorithms are present.

From the early cavemen's grunts, which made living simple, media and communication techniques have advanced significantly. This involves concealment of both - the actuality that a secret information has been exchanged and its contents. Imperceptibility, Robustness and quality of stego object are the main concerns to be fulfilled by the steganography algorithm.

Among global risks, cyber risks continue to rank high in the World Economic Forum's Global Risks Report 2021. In addition to accelerating technology adoption, an outbreak of COVID-19 revealed cyber vulnerabilities and a lack of preparedness, while exacerbated technology inequalities within and between societies. As a result of digitalization, all aspects of our lives and businesses are being affected. Efforts to create confidentiality and information safeguarding rules are necessary, however, this can also lead to fragmentation and conflicting priorities among companies, weakening their defenses. It is possible that disruptive innovations may radically change the outlook of information security in the near future. It is therefore necessary to develop and follow effective procedures for information security. Integration of cryptography and steganography is one of the effective approaches to deal with eavesdroppers. As encrypted confidential information is embedded in the cover object, it is very challenging task for an attacker to ascertain about it.

This work aims to accomplish the following main objectives-

a. Reduce the ability of an attacker to perform cryptanalysis

b. Equivalence between the properties of cover audio and the stego audio based on statistics

c. Complex to analyze the stego object.

2. BACKGROUND

In order to maintain the confidentiality and integrity of data, cryptography is used. With cryptography, information is protected from intruders as well as an unauthorized access to the information is prevented. Depending on operations like substitution, transformations, permutations; number of keys used and processing of text, cryptography algorithms are categorized. Heart of cryptography is encryption and decryption. Cipher text is obtained from original text by applying one of the above algorithms. This process is called as encryption. Converting ciphertext back to the original text is accomplished with the process of decryption. W. Stallings (2016) discussed the types of cryptography algorithms. Referring the number of keys used for encryption and decryption, Asymmetric and symmetric cryptography are two types of algorithms available. The encryption and decryption keys are the same with symmetric cryptography. DES, 3DES, Blowfish, AES, Twofish etc. are some of the symmetric cryptography algorithms. Asymmetric cryptography uses different keys. Cryptography that uses private keys is also called symmetric cryptography, and cryptography using public keys is also called asymmetric cryptography. RSA, Diffie-Helman, Elliptic curve cryptography etc. are some asymmetric cryptography algorithms. H. Dibas and K. E. Sabri (2021) conducted a performance evaluation between four of the symmetric algorithms AES, 3DES, Blowfish and Twofish. The results show that AES has the quickest encryption process and decryption process, while Twofish has the slowest. Compared to blowfish and twofish, AES and 3DES consume less memory during the encryption process, and their usage amounts are very close. In contrast, the decryption process of AES utilized less memory. The Length of the generated ciphertext is more for Blowfish and Twofish algorithms. M. Bishop (2018) demonstrated the different cryptography algorithms with examples. In its latest version, DES has been upgraded to 3DES. For the parity check, 8 bits of the 64 bits in the key are used. DES is vulnerable to a variety of attacks like brute-force attack due to the availability of high-power computation and it is no longer secure due to cryptoanalysis. To address the security flaws in the DES algorithm, 3DES was designed. DES is repeated three times with three different keys to implement 3DES which is analyzed by N. Aleisa (2015). A symmetric block cipher algorithm designed by Bruce Schneier in 1993, Blowfish employs a Feistel network structure to encrypt data. A Blowfish block has a size of 64 bits, while a key has a size varying from 32 bits to 448 bits. During the encryption process, 16 rounds are executed. As a symmetric block cipher that applies the feistel structure, Twofish is considered one of the strongest algorithms. The efficiency of this technique makes it ideal for use in smart cards. There are three key sizes available: 128,192, and 256 bits. During the encryption and decryption process, 16 rounds are executed. Where the block size is 128 bits is shown in the work by M. Albahar et al. (2018).

Prior to sending an encrypted message or archive over an uncertain correspondence channel from the message source to the message beneficiary. It becomes crucial to appoint a secret key between the sender and receiver in order to initiate communication. Both of these may be completely foreign to one another and at considerable distances from one another. An algebraic formulation of the process of generating a public key and creating a secret encryption key is described by O. Ahmedova (2020) in order to generate and distribute secret encryption keys. This leads to the strong keys in terms of secrecy. Using a genetic algorithm, S. R. P. Rao and J. K (2022) generates the keys with the help of real-time clock values. An analysis of the given security model is conducted using the existing blowfish encryption method. In addition to providing added security, the proposed algorithm enables efficient execution. In order to produce the key from real-time clocks, a genetic algorithm is used. System clock values are subjected to crossover and mutation functions. Main intention of a real time clock values is that different values are displayed at different instants, which increases security, and reinforcement of key is achieved as a result of use of genetic algorithm.

Information can be exchanged in a concealed manner through steganography and studying invisible communication refers to finding out how to hide the existence of the communicated message. This prevents eavesdroppers and attackers from snooping on the message if it is successfully achieved. Different embedding media, or carriers, can be used to conceal information with steganography. Image carriers, audio carriers, video carriers, and text carriers can all be used to transport confidential data. Different image steganography techniques are discussed by Hamid (2012). Besides analyzing and discussing how much information can be hidden in image files, these techniques are also compared in terms of how well they hide information, and whether or not they are robust to different attacks on image processing. A steganographic technique is intended to achieve good stego-image quality, a high masking capacity, low computational intricacy, visual indiscernibility, and higher security. Using least significant bit (LSB) substitution and enhanced modified signed digit (EMSD) algorithms, a hybrid image steganography technique is presented by S Solak (2020). With EMSD algorithm, the secret data is hidden within 'n' adjacent pixels of the cover image, and with LSB substitution algorithm, the secret data is hidden within the least significant k-bits. Due to its broad embedding capacity and its exploitation of modification direction (EMD)-based algorithms, it is capable of embedding more information than the EMSD algorithm. A. E. Altinbaş and Y. Yalman implemented a bit reduction-based approach of audio steganography. Bit-reduced audio files usually differ insignificantly from their original counterparts. In order to obtain the audio file containing the hidden image (stego-audio), the small changes made to the reduced audio file must be added together and saved while preserving the original bit depth. The first step is to perform bit reduction on the stego-audio in order to extract the data. An image hidden within stego-audio and reduced stego-audio contributes to the difference. Through this study, relatively higher bits are able to be embedded within an audio file without causing excessive distortion. SNR and PSNR results demonstrate the effectiveness of the developed algorithm.

The combination of cryptography and steganography can enhance data security. Idea behind this design is to encapsulate an encrypted text in mask media, such as an image, audio or video. Since the original text is encrypted, even if the steganalysis succeeds, it is difficult to acquire. Information can be leaked more easily over the internet, a variety of intelligent terminals and portable storage devices due to the rapid development of information technology. There may be a considerable impact on security of people as well as enterprises and the state when a leak occurs. The way these challenges can be addressed becomes a significant concern due to the information security issues involved. Shirole Rashmi PrakashRao and K. Jyothi (2021) demonstrated variable sample selection method for audio samples and use of secure algorithm blowfish for encrypting and decrypting the text file which contains the confidential information. This technique enhances the audio quality and improves the security. Data is protected against eavesdropping using this technology. Rashmi N and Jyothi K (2018) have proposed the technique in which the AES algorithm is used for encryption and decryption of private data. It uses same key to encrypt and decrypt the text. By using the traditional RDH algorithm, the resultant cipher text will be injected into the cover object to obtain the stego object. The image chosen by the user is used as a cover object. The Stego object will be sent over the network. In order to obtain the cipher text, stego object is recovered. And decrypting extracted cipher text with AES key will yield the original text.

The mentioned method collaborates the work of mentioned cryptography and steganography algorithms which provides an added covering of security to the information.

3. ISSUES, CONTROVERSIES, PROBLEMS

A. Key Generation

In symmetric cryptography, both sender and receiver encrypt and decrypt messages using the same keys. In contrast, for asymmetric cryptography two keys are used for encryption and decryption. The major problem with the symmetric as well as asymmetric cryptography is the creation and issuing of keys. As discussed by L. Hu (2018), to distribute group keys, traditional methods use advance distribution methods or use trusted third parties. When using advance distribution method, updating keys can be a challenge. Slow key updating frequency is carried out to save on key distribution and management costs. Military operations may be vulnerable to serious security threats if the pre-shared key has remained unchanged for a long time. Asymmetric encryption algorithms are usually used for group key distribution via trusted third parties, such as public key infrastructures (PKIs). A key management and distribution center, however, is difficult to establish because wireless networks have a dynamic topology and terminal mobility. It is possible to resolve this problem through the use of various key generation and distribution techniques. The pitfall of the approach of Diffie and Hellman as discussed in D. Hellman (1980) is that to achieve an improved performance, large primes with a length of 512 bits or more are needed, as a result, computational costs and time expenditures increases. As per the Y. Wu (2017), the collection of runtime statistics of some critical blocks of key generation in RSA shows that the time of large prime number generation occupies the vast majority time of the key generation.

For effective secure communication, Key Management is one of the most important issues. As a novelty, this work proposes the optimized key generation and distribution algorithm.

B. Encryption

Symmetric encryption has a speed advantage of nearly 1000-fold over asymmetric encryption, as documented in various papers on cryptography and the key size must be relatively small. As a result, symmetric key algorithms can be studied by their performance metrics. Depending on the operation used on the text, algorithms are divided into substitution ciphers and transposition ciphers. Some examples of accepted private key algorithms are: Playfair cipher, Caesar cipher, Vigenere cipher, Vernam cipher, Serpent, Twofish, AES (Rijndael), Blowfish, CAST5, RC4, RC6, DES, 3DES, IDEA etc. There are only 26 possible combinations in the cipher. Even without computer intervention, it is easy to break. The weakness of Playfair cipher is the fact that there will be corresponding plaintexts (UR and RU) for a digraph in the ciphertext (AB) and it's reverse (BA) i.e., Inverse substitutions are possible. Frequency analysis can easily make sense of that, if it is known what language the plaintext uses. In Advanced Encryption Standard (AES), every block uses the same encryption method and the analytical structure is too simple. Ciphertext is generated by using precalculated, key-dependent substitutions in Twofish's encryption algorithm. Side channel attacks are made more possible as a consequence of precompiling this value.

Among mentioned algorithms, blowfish is the most secure algorithm. Furthermore, Blowfish takes fewer operations to complete than other encryption algorithms. The proposed work combines Vernam ciphers with updated blowfish algorithm to provide extra security to the information.

C. *Embedding*

With the advent of digital technology, by covering up a secret message inside another message, steganography conceals the existence of a communication. The main limitation of image steganography is based on the total data size, the embedded data can be as large as the total data size. The embedding of more data may not be possible if a piece of data is already highly compressed. There is an abundance of information, large file, therefore someone can suspect about it. By reducing the noise-inducing nature of audio Steganography, phase coding reduces the disadvantages associated with these techniques. The major disadvantage of LSB steganography is that, it hides only few bits of information and can be easily detected by intruder.

In the view of above problems, this work designs a novel system which integrate the cryptography and steganography techniques to overcome with their respective cons with the updated key generation scenario.

4. Solutions and Recommendations

The proposed work introduces the novel technique to provide extra security to the confidential information. In this technique, Integration of cryptography and steganography is analyzed with the better results. This concept is shown in illustrative form in Fig.1. Encryption scenario is experimented with enhanced blowfish algorithm and embedding is performed with different audio files using chao-quant technique. Key generation is the most challenging problem in the cryptography. In the proposed work, keys are generated using chaotic maps and genetic algorithm. Fig. 1 depicts the overall design of the proposed system.

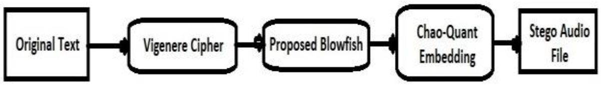


Fig. 1 Schematic of Proposed Work

A. *Encryption – Embedding Process*

Encryption-Embedding process is carried out at the sender side. Stego audio is obtained as a result of this process. In the proposed method, for the encryption and decryption, vigenere cipher and updated blowfish algorithm is used.

The following steps are executed in the Encryption – Embedding process.

Step 1: Generate the key using chaotic hash value and genetic algorithm.

Step 2: Obtain the intermediate cipher text using Vigenere cipher method.

Step 3: Using key generated in step 1 and intermediate cipher generated in step 2, execute proposed blowfish algorithm to get the crypt text at the end.

Step 4: Embed the crypt text obtained in step 3 in a cover audio using proposed chao-quant method.

These steps are described in detail below.

1) Key Generation

Key management and distribution problem in symmetric cryptography algorithm has been resolved with the proposed key generation algorithm using real time clock value and chaotic hash values. Fig. 2 shows the schematic of generation of secret key for the proposed blowfish algorithm. Random 64-bit s-box sequence is used for the exor operation. Real time clock value is divided into n blocks. First key block is derived by using equation (1). Next successive key blocks are obtained using equation (2).

$$T_1 = M_1 \oplus sbox \tag{1}$$

$$T_n = T_{n-1} \oplus M_n \tag{2}$$

Genetic algorithm is performed on the obtained key blocks by following below steps.

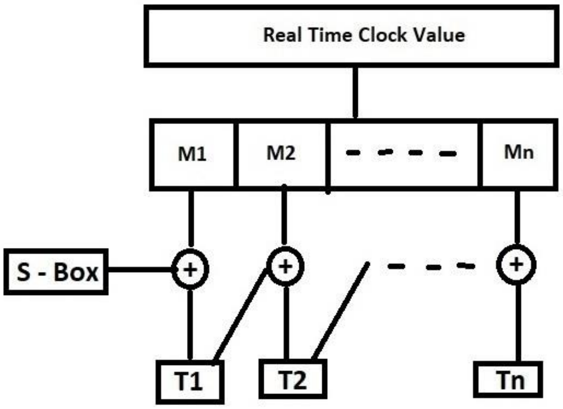


Fig. 2 Generation of Secret Key

a) Crossover

First step in the genetic algorithm is Crossover in which two chromosomes are merged to form a new chromosome. It is divided into three types - one point, two point and uniform crossover. Crossover points are selected randomly. The proposed system uses two-point crossover in which two crossover points are selected randomly. After performing crossover, mutation is performed on the resultant chromosome.

b) Mutation

In the final mutation process, new offspring is obtained by complimenting resultant chromosome obtained as a result of crossover function. This new offspring is used as a key in the proposed encryption algorithm.

Following steps are executed to generate the key using genetic algorithm. Generated $H_{offspring}$ is used as a key in proposed blowfish algorithm.

Steps	Input: $T_1 \ldots . T_n$ blocks obtained from chaotic hash value
	Output: New offspring used as a key
1	Divide input into two halves
	$H_1 = \frac{T_1}{\frac{T_n}{2}}$ and $H_2 = \frac{\frac{T_n}{2}}{T_n}$
2	Select p and q randomly
3	$H_{1new} = interchange\ \sum_p^q (H_1, H_2)$ $H_{2new} = interchange\ \sum_p^q (H_2, H_1)$
	Where, p and q are two cross-points
	$H_{new} = concat(H_{1new}, H_{2new})$
4	$H_{offspring} = mut(H_{new})$

2) Encryption

Generated key in the above process is used in the proposed symmetric blowfish encryption. To increase the complexity and security, vigenere cipher is generated before the message is processed by the proposed blowfish algorithm. Major disadvantage of Vigenere cipher is can easily be broken individually because of repeating nature of its keys. But as vigenere cipher undergoes to the blowfish encryption it made it difficult to crack it.

a) Vigenere Cipher

Vigenere Cipher is a method of encrypting alphabetic text. Vigenere cipher is a kind of substitution cipher that employs polyalphabetic substitutions. It is stronger than Caesar cipher as it uses series of interwoven Caesar ciphers. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenere Square or Vigenere Table. The given keyword is repeated in a circular manner until it matches the length of the plain text. Vigenere cipher is very simple method of encryption. To encrypt, a table of alphabets is used, termed a tabula rectum. It has the alphabet written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers. At different points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword. The letter of the key is chosen, and that row is gone along to find the column heading that matches the message character. The letter at the intersection of [key-row, message-col] is the enciphered letter. Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in that row and then using the column's label as the plaintext. Cipher text generated V_{Ci} by vigenere cipher in equation (3) is the input to the proposed blowfish algorithm.

$$V_{Ci} = Vig(\,P_i\,, K_i) \qquad (3)$$

b) Proposed Blowfish Encryption

Proposed blowfish algorithm uses 64-bit block size with 16 rounds. 64-bit result obtained in the proposed key generation algorithm is used as a key. V_{Ci} is the text input and $H_{offspring}$ is the key input to the given algorithm. Fig. 3 depicts the idea of proposed blowfish encryption.

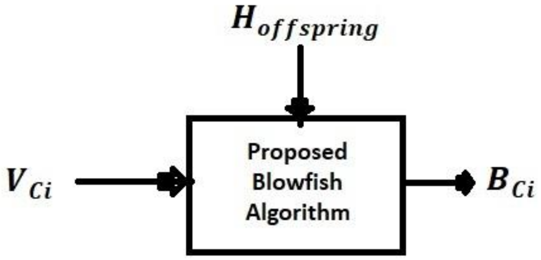
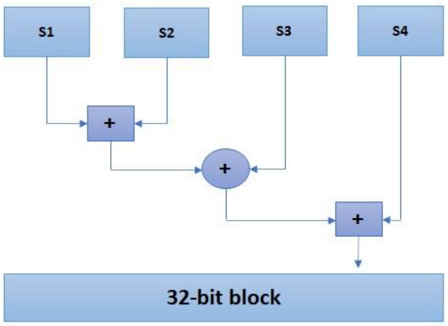


Fig. 3 Proposed Blowfish Encryption

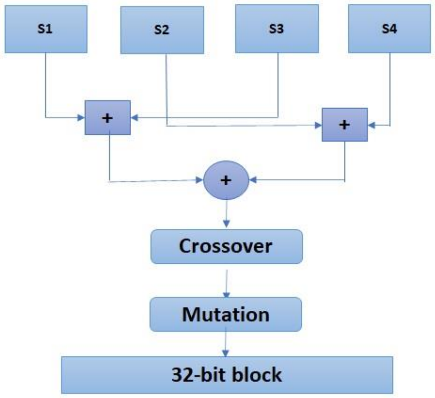
Blowfish algorithm uses substitution-permutation network. V_{Ci} is divided into two halves $Left$ and $Right$. 16 rounds are executed on these halves where each round uses different key P_i . Operations in one round are described below where F is the round function. B_{Ci} is the final cipher text generated by the proposed encryption algorithm.

1. $Left_{i+1} = Left_i \oplus P_i$
2. $Right_{i+1} = Right_i \oplus F$
3. Swap $Left_{i+1}$ and $Right_{i+1}$
4. Repeat steps 1 - 3 for 16 times
5. $Right_{i+1} = Right_i \oplus P_{17}$
6. $Left_{i+1} = Left_i \oplus P_{18}$
7. $B_{Ci} = concat(Left_{i+1}, Right_{i+1})$

Proposed blowfish algorithm updates the round function. Fig. 4 illustrates a) the round function in an original algorithm and b) the updated round function.



(a)



(b)

Fig. 4 Blowfish round function (a) Original algorithm (b) Proposed algorithm

c) Embedding

An audio Steganography system embeds secret messages into digital sound. It is done by slightly altering the binary sequence of a sound file to encode the secret message. Currently, audio Steganography software is capable of hiding messages within WAV files, AU files, and even MP3 files. There is generally a greater challenge in embedding secret messages in digital sound than in other media, such as picture data; as Sound perturbations are very sensitive to the human ear, which can detect them as low as one part in ten million. There is a wide range of methods that can be used to hide information, from relatively simple algorithms that rely on signal noise to more powerful methods that depend on signal processing techniques.

In the proposed chao-quant embedding technique an audio cover sample is chosen with the help of chaotic map and the cipher text generated in the above encryption process is camouflaged in the selected samples using suggested quantization technique.

Chaotic sequence is generated and is stored in chaotic matrix. For the generated chaotic sequence, proposed system uses quantization method. Equation (4) is used to calculate the chaotic sequence

$$X_n + 1 \ = X_n * \ \mu * (1 + X_n) \qquad (4)$$

Where, μ=3.6 and X_n as a current audio channel value

After generation of the chaotic sequence of audio samples, quantization operation is performed on it to embed the cipher text. Coefficients are calculated using equation (5) for an audio sample with m X n dimensions.

$$coeff = (C_i)(C_j)(total) \qquad (5)$$

Where,

$$C_i = \frac{\sqrt{2}}{\sqrt{m}} \qquad (6)$$

$$C_j = \frac{\sqrt{2}}{\sqrt{n}} \qquad (7)$$

$$total = \sum\nolimits_{k=0}^{m} \sum\nolimits_{l=0}^{n} [((\mathrm{pixel}[k][l]) \left(\frac{\cos(2*k+1)(k)(\pi)}{2*m}\right)][\left(\frac{\cos(2*k+1)(k)(\pi)}{2*n}\right)] \qquad (8)$$

The proposed algorithm to embed cipher text in cover audio is as follows.

Algorithm for incorporating text into audio covers

	Input: Cover Audio A and Cipher Text B_{Ci} Output: Stego audio A_s
Step 1	Split the audio A into multiple 16-bit samples S_i, resulting in two channels per sample C_0 and C_1 of 8-bit each.
Step 2	Calculate chaotic sequence $X_n + 1$ using equation (4).
Step 3	Calculate the quantized blocks $coeff$, using equation (5) for the generated chaotic sequence $X_n + 1$ and separate dequantized blocks $dcoeff$.
Step 4	Embed the bits of secret data into audio samples starting from $coeff$ followed by $dcoeff$ to obtain stego audio A_s .
Step 5	Generated Stego audio A_s and the final sequence $dSeq = coeff + dcoeff$ is sent to the receiver.

B. Extraction – Decryption Process

For the Extraction and Decryption process, the process is reversed on the received stego audio to obtain the original text. The following steps are executed in the Extraction – Decryption process.

Step 1: Generate the key using chaotic hash value and genetic algorithm.

Step 2: Extract the cipher text from received stego audio using reverse chao-quant audio steganography method.

Step 3: Using key generated in step 1 and cipher obtained in step 2, execute proposed blowfish decryption algorithm to get the ciphertext used in intermediate encryption.

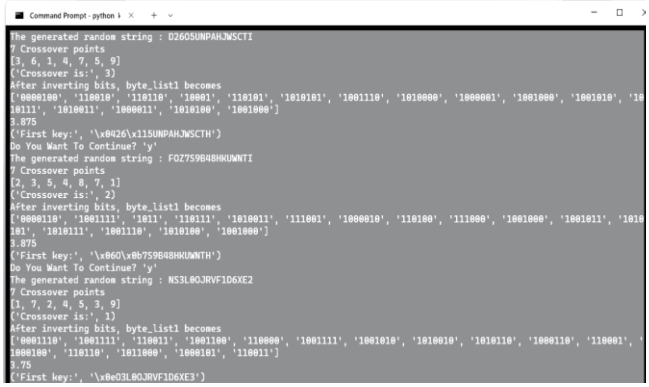
Step 4: Decrypt the crypt text obtained in step 3 using Vigenere decryption method to get the original text.

C. Observational Perspectives

The compliance of the proposed algorithms is observed by conducting different tests.

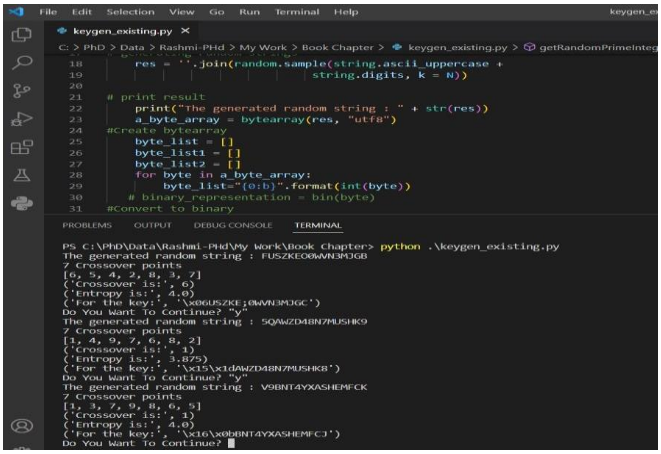
1) Key Generation

Keys are generated using proposed key generation algorithm for the mentioned system. Entropy values are considered to select the best key among the generated keys. Fig. 5a shows the obtained entropy values for the generated keys using existing key generation algorithm by Turcanik Michal and Javurek Martin (2021). Fig. 5b shows the keys generated using proposed key generation algorithm.



```
The generated random string : D26O5UNPAHJWSCTI
7 Crossover points
[3, 6, 1, 4, 7, 5, 9]
('Crossover is:', 3)
After inverting bits, byte_list1 becomes
['0000100', '110010', '110110', '10001', '110101', '1010101', '1001110', '1010000', '1000001', '1001000', '1001010', '10
10111', '1010011', '1000011', '1010100', '1001000']
3.875
('First key:', '\x0426\x115UNPAHJWSCTH')
Do You Want To Continue? 'y'
The generated random string : FOZ7S9B48HKUWNTI
7 Crossover points
[2, 3, 5, 4, 8, 7, 1]
('Crossover is:', 2)
After inverting bits, byte_list1 becomes
['0000110', '1001111', '1011', '110111', '1010011', '111001', '1000010', '110100', '111000', '1001000', '1001011', '1010
101', '1010111', '1001110', '1010100', '1001000']
3.875
('First key:', '\x06O\x0b7S9B48HKUWNTH')
Do You Want To Continue? 'y'
The generated random string : NS3L0OJRVF1D6XE2
7 Crossover points
[1, 7, 2, 4, 5, 3, 9]
('Crossover is:', 1)
After inverting bits, byte_list1 becomes
['0001110', '1001111', '110011', '1001100', '110000', '1001111', '1001010', '1010010', '1010110', '1000110', '110001', '
1000100', '110110', '1011000', '1000101', '110011']
3.75
('First key:', '\x0eO3L0OJRVF1D6XE3')
```

Fig. 5a Keys generated using existing algorithm



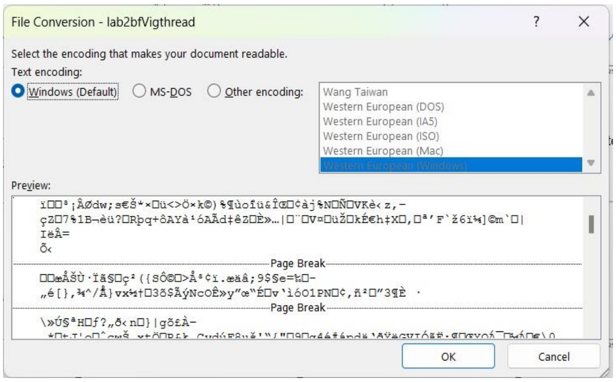
```python
        res = ''.join(random.sample(string.ascii_uppercase +
                                    string.digits, k = N))

    # print result
        print("The generated random string : " + str(res))
        a_byte_array = bytearray(res, "utf8")
    #Create bytearray
        byte_list = []
        byte_list1 = []
        byte_list2 = []
        for byte in a_byte_array:
            byte_list="{0:b}".format(int(byte))
      # binary_representation = bin(byte)
    #Convert to binary
```

```
PS C:\PhD\Data\Rashmi-PHd\My Work\Book Chapter> python .\keygen_existing.py
The generated random string : FUSZKEO0WVN3MJGB
7 Crossover points
[6, 5, 4, 2, 8, 3, 7]
('Crossover is:', 6)
('Entropy is:', 4.0)
('For the key:', '\x06USZKE;0WVN3MJGC')
Do You Want To Continue? "y"
The generated random string : 5QAWZD48N7MUSHK9
7 Crossover points
[1, 4, 9, 7, 6, 8, 2]
('Crossover is:', 1)
('Entropy is:', 3.875)
('For the key:', '\x15\x1dAWZD48N7MUSHK8')
Do You Want To Continue? "y"
The generated random string : V9BNT4YXASHEMFCK
7 Crossover points
[1, 3, 7, 9, 8, 6, 5]
('Crossover is:', 1)
('Entropy is:', 4.0)
('For the key:', '\x16\x0bBNT4YXASHEMFCJ')
Do You Want To Continue?
```

Fig. 5b Keys generated using proposed key generation algorithm

2) Cryptography

Cryptography implementation is performed using Python. Encryption-decryption experiments are conducted on text files with different sizes between 20 kb and 80 kb. Fig. 6 shows the final cipher text file after performing vigenere encryption and proposed blowfish encryption.



Select the encoding that makes your document readable.

Text encoding:

● Windows (Default) ○ MS-DOS ○ Other encoding:

Wang Taiwan
Western European (DOS)
Western European (IA5)
Western European (ISO)
Western European (Mac)
Western European (Windows)

Preview:

```
ï□□³¡ÂØdw;s€Š*×□ü<>Ö×k©)%¶ùoîü&ÎŒ□¢àj%N□Ñ□VKè‹z,–
çZ□7%1B¬èü?□Rþq+ôAYà¹óAÃd‡êZ□È»…|□¨□V¤□üŽ□kÉ€h‡X□,□ª’F`ž6ï¼]©m`□|
IëÂ=
Õ‹
Page Break
□□æÅŠÙ·Ïã§□ç²({SÔ©□>Å³¢ï.æäâ;9$§e=‰□–
„é[},¾^/Å}vx½†□3õ$ÃýNcOÊ»y”œ“É□v‘ìó01PN□¢,ñ²□”3¶È ·
Page Break
\»Ú§ªH□f?„ð‹n□}|gõ£À–
```

Fig. 6 Final cipher text generated using proposed encryption algorithm

3) Audio Steganography

Execution of existing and proposed steganography algorithms is accomplished using MATLAB R2015a from Mathworks. A diverse set of audio samples from Pathmind - including music and speech - make up the cover audio dataset. Various frequencies are used to analyze speech samples as well as music samples. Sample datasets are depicted in the form of a histogram as shown in Fig. 7 and Fig. 8.

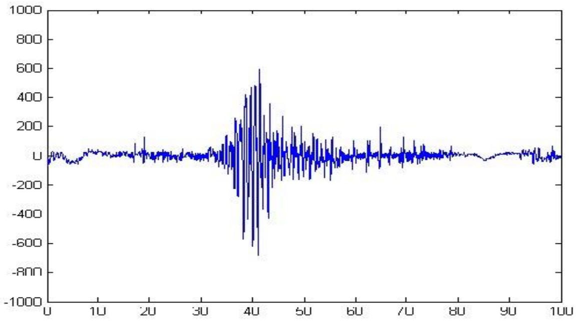


Figure 7a Histogram of experimental test files - Original music sample (950KB)

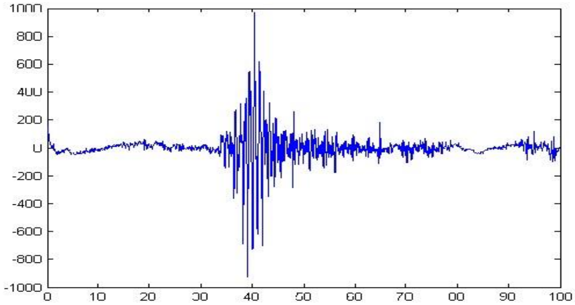


Figure 7b Histogram of experimental test files - Stego music sample after embedding 20 KB

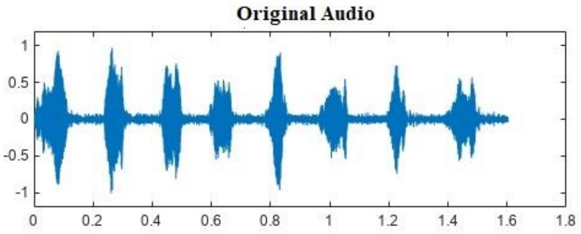


Figure 8a Histogram of experimental test files - Original speech sample (620KB)

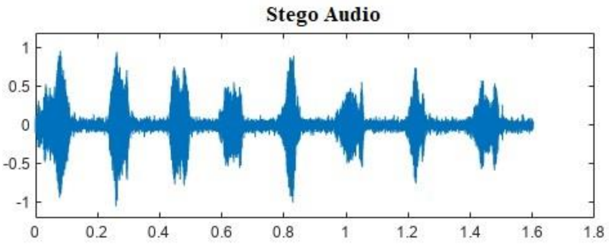


Figure 8b Histogram of experimental test files - Stego speech sample after embedding 20 KB

D. *Performance Analysis*

This section discusses the details of performance of the proposed system for generated keys, security of cryptography algorithm and quality of stego audio. Different criteria are used to demonstrate the performance of proposed system.

1) *Key Generation*

Generated keys are selected based on entropy values. Key with best entropy value is given as an input to the proposed blowfish algorithm. Security of the selected is inspected with the bruteforce attack scenario. The result of bruteforce attack scenario shows that the proposed key generation method takes more time to break the key.

a) *Entropy*

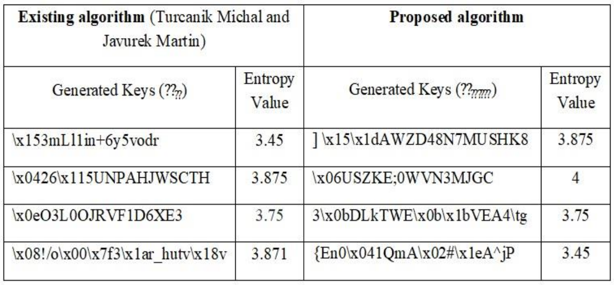
Despite years of research, cryptographic keys still pose a persistent challenge in both academic settings and in the real world. We can assess the randomness of a data-generating function using entropy. Utilizing entropy, cryptographic functions are evaluated for strength and effectiveness. Data with full entropy does not show any discernible patterns since it is completely random. An analysis of low-entropy data can help predict future values. Table I shows the generated keys using existing and proposed algorithms. Entropy is calculated using equation (9).

$$H = (-1/\log N)\ \sum_{i=0}^{n} (P_i)(\log(P_i)) \qquad (9)$$

where,

N – total number of observed even
P_i – probability of i^th^ event

Table I Generated keys with entropy values

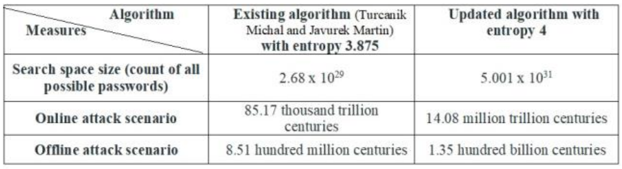


Existing algorithm (Turcanik Michal and Javurek Martin)		**Proposed algorithm**	
Generated Keys ($??_{n}$)	Entropy Value	Generated Keys ($??_{nnn}$)	Entropy Value
\x153mL11in+6y5vodr	3.45	] \x15\x1dAWZD48N7MUSHK8	3.875
\x0426\x115UNPAHJWSCTH	3.875	\x06USZKE;0WVN3MJGC	4
\x0eO3L0OJRVF1D6XE3	3.75	3\x0bDLkTWE\x0b\x1bVEA4\tg	3.75
\x08!/o\x00\x7f3\x1ar_hutv\x18v	3.871	{En0\x041QmA\x02#\x1eA^jP	3.45

b) *Bruteforce Attack Scenario*

In order to analyze the keys generated, Gibson research corporation employs the Haystack software. Using this software analysis of generated keys has been done. It is clear from the analysis that the proposed algorithm takes a greater amount of time to search a password selected by the user, and there is a huge amount of search space both within an online attack scenario and an offline attack scenario. Table II shows the bruteforce attack scenario for the selected keys.

Table II Bruteforce Attack Scenario for Generated Keys



Algorithm Measures	Existing algorithm (Turcanik Michal and Javurek Martin) with entropy 3.875	Updated algorithm with entropy 4
Search space size (count of all possible passwords)	2.68 x 10^{29}	5.001 x 10^{31}
Online attack scenario	85.17 thousand trillion centuries	14.08 million trillion centuries
Offline attack scenario	8.51 hundred million centuries	1.35 hundred billion centuries

2) *Encryption-Decryption*

The proposed work implements crossover and mutation techniques on a pair of S-boxes separately after changing the order of OR and EXOR operation; in order to obtain the complex intermediate cipher text and make it difficult for an intruder to modify the final cipher text. In the following subsections, the avalanche effect, execution time, and throughput of the proposed encryption algorithm is examined. Two main characteristics distinguish one encryption method from another are its ability to defend data from attacks and its rapid and effective response time. As a result of the proposed method, it is shown that altering the plaintext very slightly results in a substantial alteration to the ciphertext. Considering the throughput of the suggested technique, it is clear that the suggested technique achieves greater throughput than the existing algorithm. Also executing the proposed technique takes a very short amount of time. Following criteria shows the performance of the proposed cryptography algorithm.

a) *Avalanche Effect*

It is desirable for cryptographic algorithms, especially block ciphers and cryptographic hash functions, to exhibit the avalanche effect, which shows the small change in input results in a dramatical change in output. High-quality block ciphers should dramatically change ciphertext if either the key or plaintext changes. The lack of avalanche effect in a block cipher or cryptographic hash function, indicates that it has insufficient organization, and as a result, From the output alone, a cryptologist can identify the input. Depending on how severe the problem is, it may cause the algorithm to malfunction in part or completely. As a result, the designer of the cryptographic algorithm views the avalanche effect as desirable property of cryptographic algorithm.

Avalanche effect for n-bit change in plain text is calculated using following equation.

$$Aval\ eff_n = \frac{Cipher_C}{Cipher_T} \qquad (10)$$

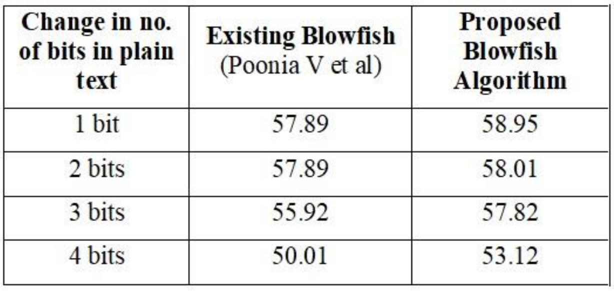
Where,

$Cipher_C$ are the changed cipher bits when n-bits change in plain text.

$Cipher_T$ are the total number of cipher text bits.

Experiments on different text files is carried out and Table III shows the average avalanche effect on files ranging from 10 KB to 47 KB.

Table III Avalanche effect (in %)

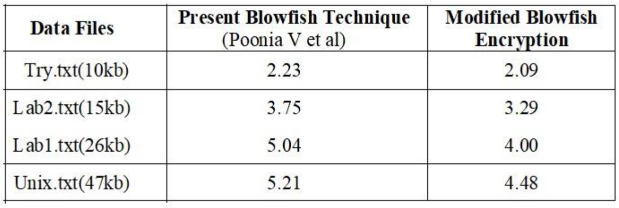


Change in no. of bits in plain text	Existing Blowfish (Poonia V et al)	Proposed Blowfish Algorithm
1 bit	57.89	58.95
2 bits	57.89	58.01
3 bits	55.92	57.82
4 bits	50.01	53.12

b) *Execution Time*

An execution time E_t is the amount of time needed to perform encryption and decryption. In this study, both the existing blowfish algorithm and the proposed blowfish algorithm are measured in terms of execution time. Table IV provides a comparison of the existing algorithm's speed and the proposed algorithm's speed (in seconds).

Table IV Algorithm Speed (in seconds)



Data Files	Present Blowfish Technique (Poonia V et al)	Modified Blowfish Encryption
Try.txt(10kb)	2.23	2.09
Lab2.txt(15kb)	3.75	3.29
Lab1.txt(26kb)	5.04	4.00
Unix.txt(47kb)	5.21	4.48

c) *Throughput*

Using encryption time, we are able to determine the throughput of the encryption scheme. Basically, it tells how fast the encryption is. Using the formula: plaintext divided by encryption time equals throughput, which is calculated based on the amount of plaintext encrypted per second, which is illustrated in equation (11).

$$T = \frac{\sum Plain_T}{E_t} \qquad (11)$$

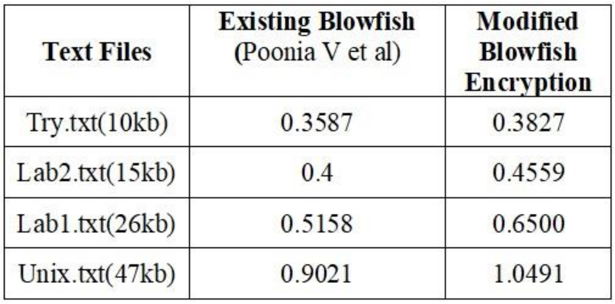
Where,

$Plain_T$ are the total number of plain text bits.

E_t is the encryption time required to encrypt $Plain_T$.

Table V provides a comparison of the existing algorithm's throughput and the proposed algorithm's throughput (in bits/sec).

Table V Throughput (in bits/sec)



Text Files	Existing Blowfish (Poonia V et al)	Modified Blowfish Encryption
Try.txt(10kb)	0.3587	0.3827
Lab2.txt(15kb)	0.4	0.4559
Lab1.txt(26kb)	0.5158	0.6500
Unix.txt(47kb)	0.9021	1.0491

3) *Audio Steganography*

In the proposed chao-quant embedding technique, a selection of samples is done from the audio cover with the help of chaotic map and the cipher text generated in the above encryption process is embedded in the selected samples through the proposed quantization technique.

Performance of the audio steganography algorithms is measured with respect to the quality of audio, embedding capacity, similarity between original and stego audio etc. In this section, an algorithm's performance is calculated with the help of metrics- peak signal to noise ratio, embedding capacity, structural similarity index and correlation coefficient.

a) *Peak Signal to Noise Ratio (PSNR)*

As an estimate of how accurately the signal is represented, Peak signal to noise ratio (PSNR) demonstrates the introduction of compression noise into the signal. It is possible to evaluate the standard of stego audio by comparing its characteristics with those of the original audio.

Original audio and noisy audio are examined when calculating PSNR values. According to the mentioned method, the PSNR value is higher than that of existing LSB audio steganography methods. Better audio quality is achieved with a higher PSNR value.

MSE is defined as: Given $m\ X\ n$ original audio matrix A and its stego audio K. The following formula measures the given quantifiers

$$mse = \frac{1}{m\ X\ n}\sum_{0}^{m-1}\sum_{0}^{n-1}\ (A\ (i,j) - K\ (i,j)) \tag{12}$$

For n-bits in the quantization process, PSNR value is calculated as follows.

$$PSNR = 20\ \ log10\left(\frac{2n}{\sqrt{mse}}\right) \tag{13}$$

Table VI shows the PSNR, MSE and SSIM measures for the proposed algorithm and existing algorithm.

Several samples from a speech audio file with different frequencies are also tested using the proposed chao-quant audio steganography algorithm. In the first part, a segment of the selected audio file is cut into smaller pieces from the beginning. To carry out the further research, from the end of the selected audio file, a portion of it is cut off. The two scenarios are assessed based on readings. Using this analysis, it can be confirmed that cutting a sample from the beginning decreases the quality of stego audio. In contrast to the previous readings, due to echo, the stego audio quality is excellent when cut from the end. Table VII summarizes the observations made during trimming process.

b) *Structural Similarity Index Matrix (SSIM)*

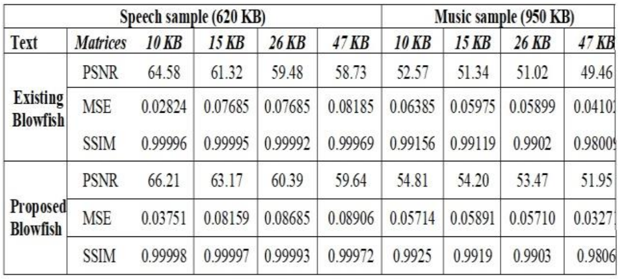
As a measure for predicting how well digital television and film images and other kinds of digital media will be received by viewers, the structural similarity index measure (SSIM) can be used. Two audio files are compared using SSIM to determine their similarity. Considering the structural information, it is demonstrated how neighboring samples seem to have a close relationship and thus are affected by each other. Especially with samples, which are spatially proximal, the same applies. For multiple cameos, SSIM index for pixel x and y is calculated using equation (14).

$$SSIM(x,y) = \frac{(2\mu x\mu y + C_1)(2\delta x\ \delta y + C_2)}{(\mu x^2 + \mu y^2 + C_1)(\delta x^2 + \delta y^2 + C_2)} \tag{14}$$

Where,

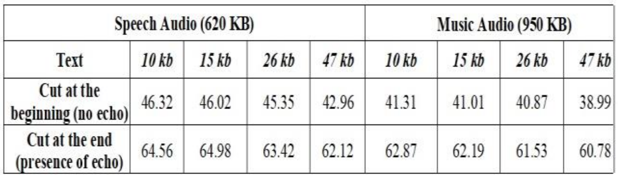
$$\mu x, \mu y\ are\ sample\ mean\ of\ x\ and\ y$$
$$\delta x, \delta y\ are\ variance\ of\ x\ and\ y$$
$$C_1\ and\ C_2 - to\ stabilize\ the\ division$$

Table VI PSNR (in db), MSE and SSIM (between -1 and 1) for proposed chao-quant audio steganography



		Speech sample (620 KB)				Music sample (950 KB)			
Text	*Matrices*	*10 KB*	*15 KB*	*26 KB*	*47 KB*	*10 KB*	*15 KB*	*26 KB*	*47 KB*
Existing Blowfish	PSNR	64.58	61.32	59.48	58.73	52.57	51.34	51.02	49.46
	MSE	0.02824	0.07685	0.07685	0.08185	0.06385	0.05975	0.05899	0.0410
	SSIM	0.99996	0.99995	0.99992	0.99969	0.99156	0.99119	0.9902	0.9800
Proposed Blowfish	PSNR	66.21	63.17	60.39	59.64	54.81	54.20	53.47	51.95
	MSE	0.03751	0.08159	0.08685	0.08906	0.05714	0.05891	0.05710	0.0327
	SSIM	0.99998	0.99997	0.99993	0.99972	0.9925	0.9919	0.9903	0.9806

Table VII Analysis of proposed chao-quant audio steganography for trimmed audio signal



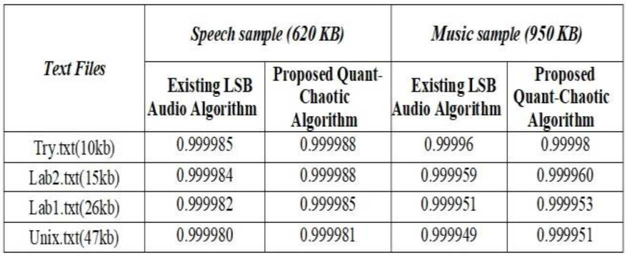
	Speech Audio (620 KB)				Music Audio (950 KB)			
Text	*10 kb*	*15 kb*	*26 kb*	*47 kb*	*10 kb*	*15 kb*	*26 kb*	*47 kb*
Cut at the beginning (no echo)	46.32	46.02	45.35	42.96	41.31	41.01	40.87	38.99
Cut at the end (presence of echo)	64.56	64.98	63.42	62.12	62.87	62.19	61.53	60.78

c) *Correlation Coefficient* (cc)

It is possible to determine the pitch of a discrete speech signal using the function of autocorrelation. In an interval defined between two frequencies with amplitudes greater than 30% of the original energy, the sample with the greatest amplitude represents the pitch frequency, which correlates with the sampling frequency. Between -1 and 1, an autocorrelation coefficient represents the relationship between the two samples. Correlation coefficient value 1 indicate Perfect positive correlation, 0 indicate zero correlation and -1 indicate Perfect negative correlation. To measure autocorrelation, correlate the reference window and lagged window of the signal by stepping through the signal sample-by-sample. Pearson's r test is used to calculate the correlation coefficient cc between two images. Equation (15) shows the formula to perform Pearson's r test where x_m is the m^th sample in first audio and y_m is the m^th sample in stego audio. Correlation coefficients of original audio and stego audio are depicted in Table VII. The tests are done for existing method and proposed method for varying text sizes.

$$cc = \frac{[(\sum_{m=0}^{255} x_m y_m) - (\sum_{m=0}^{255} x_m)(\sum_{m=0}^{255} y_m)]}{\left[\sqrt{\sum_{m=0}^{255} {x_m}^2 - (\sum_{m=0}^{255} x_m)^2}\right]\left[\sqrt{\sum_{m=0}^{255} {y_m}^2 - (\sum_{m=0}^{255} y_m)^2}\right]} \tag{15}$$

Table VIII Correlation coefficient



Text Files	*Speech sample (620 KB)*		*Music sample (950 KB)*	
	Existing LSB Audio Algorithm	Proposed Quant-Chaotic Algorithm	Existing LSB Audio Algorithm	Proposed Quant-Chaotic Algorithm
Try.txt(10kb)	0.999985	0.999988	0.99996	0.99998
Lab2.txt(15kb)	0.999984	0.999988	0.999959	0.999960
Lab1.txt(26kb)	0.999982	0.999985	0.999951	0.999953
Unix.txt(47kb)	0.999980	0.999981	0.999949	0.999951

CONCLUSION

The intendment of this study is to present an innovative technique that integrates cryptography and steganography to improve information security. In addition to ensuring highly secure audio, the proposed method would also be effective for high-frequency audio files.

Using genetic algorithms, random, unpredictable key values are generated in the proposed work. A novel blowfish algorithm is implemented using a key generated with a high entropy value. An attack scenario is used to observe the security of the key and shows that cracking the generated key takes trillions of centuries. Various aspects of the performance of the new blowfish algorithm are assessed, such as the avalanche effect, speed, and throughput. In all aspects mentioned, the proposed algorithm provides the best results. Proposed chao quant audio steganography algorithm produces higher quality audios after embedding. In order to determine the quality of an audio, PSNR, MSE, SSIM, and correlation coefficient are used, which shows better sequel than existing audio steganography algorithm. Robustness, reliability, and security criteria are satisfied by this combined approach. People will be able to communicate without being eavesdropped on by the proposed mechanism.

A variety of applications can benefit from the use of this technique. Examples include applications related to online business, automation systems for security, system for Dissemination of digital information, etc. There are further possibilities for extending the proposed work to integrate neural network techniques, Artificial intelligence with computer vision and linguistics processing scenarios.

REFERENCES

[1] W. Stallings. (2016). "Cryptography and Network Security Principles and Practices", 7th Edition, William Stallings.

[2] H. Dibas and K. E. Sabri. (2021). "A comprehensive performance empirical study of the symmetric algorithms: AES, 3DES, Blowfish and Twofish," 2021 International Conference on Information Technology (ICIT), Amman, Jordan, pp. 344-349, doi: 10.1109/ICIT52682.2021.9491644.

[3] M. Bishop. (2018). "Computer Security: Art and Science", 2nd Edition, Addison-Wesley Professional.

[4] N. Aleisa. (2015). "A comparison of the 3DES and AES encryption standards", International Journal of Security and Its Applications, 9. 241- 246.

[5] M. Albahar, O. Olawumi, K. Haataja and P. Toivanen. (April 2018). "Novel Hybrid Encryption Algorithm Based on Aes, RSA, and Twofish for Bluetooth Encryption", Journal of Information Security, Volume 9, Issue 2, PP. 168-176.

[6] O. Ahmedova, U. Mardiyev and O. Tursunov. (2020). "Generation and Distribution Secret Encryption Keys with Parameter,*"* 2020 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, pp. 1-4, doi: 10.1109/ICISCT50599.2020.9351446.

[7] S. R. P. Rao and J. K. (2022). "Secret Key Generation using Genetic Algorithm for the Hybrid Blowfish Encryption and Substitution Ciphers," 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1-5, doi: 10.1109/ICCSEA54677.2022.9936357

[8] Hamid, Nagham & Yahya, Abid & Ahmad, R.Badlishah & Al-qershi, Osamah. (2012). "Image Steganography Techniques: An Overview", International Journal of Computer Science and Security, 6. 168-187.

[9] S. Solak. (2020). "High Embedding Capacity Data Hiding Technique Based on EMSD and LSB Substitution Algorithms," *in* IEEE Access, vol.8, pp.166513-166524, doi: 10.1109/ACCESS.2020.3023197

[10] A. E. Altinbaş and Y. Yalman (2021). "Bit Reduction based Audio Steganography Algorithm," 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, pp. 703-706, doi: 10.1109/UBMK52708.2021.9558943.

[11] Shirole Rashmi PrakashRao, K. Jyothi. (September 24, 2021). "A Novel Audio Steganography Technique Integrated with a Symmetric Cryptography: A Protection Mechanism for Secure Data Outsourcing", International Journal of Computational Science and Engineering, Inderscience Journal, Vol. 24, No. 5, pp 530–537•, https://doi.org/10.1504/IJCSE.2021.118102

[12] N. Rashmi and K. Jyothi. (2018). "An Improved Method for Reversible Data Hiding Steganography Combined with Cryptography," 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, pp. 81-84, doi: 10.1109/ICISC.2018.8398946.

[13] L. Hu, A. Hu, G. Li and C. Cao. (*2018*). "Lightweight Group Key Distribution Method Based on High Similar Wireless Channel Characteristics," IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 2018, pp. 140-144, doi: 10.1109/ICCT.2018.8600139.

[14] D. Hellman. (April 1980). “Cryptographic Apparatus and Method,” US Patent 4.200.770.

[15] Y. Wu and X. Wu. (*2017*). "Implementation of efficient method of RSA key-pair generation algorithm," IEEE International Symposium on Consumer Electronics (ISCE), Kuala Lumpur, Malaysia, pp. 72-73, doi: 10.1109/ISCE.2017.8355552.

[16] Turcanik, Michal and Javurek, Martin. (2021). “The Use of Genetic Algorithms for Cryptographic Keys Generation”, Digital Transformation, Cyber Security and Resilience of Modern Societies, pp.315-324, 10.1007/978-3-030-65722-2_19.

[17] Poonia V. and Yadav N. S. (2015). "Analysis of Modified Blowfish Algorithm in Different Cases with Various Parameters," International Conference on Advanced Computing and Communication Systems, Coimbatore, India, pp. 1-5, doi: 10.1109/ICACCS.2015.7324114.