Proposed Algorithm CRB for Audio Encryption with High Security and Robust

Chetan Rathod[1], Prof. (Dr.) Atul M Gonsai[2]
Vivekanand College for Advanced Computer and Information Science, VNSGU, Surat[1]
Dept. of Computer Science, Saurashtra University, Rajkot[2]

Abstract:
Ensuring the security of data transmission and exchange is a crucial issue in today's world. Cryptographic algorithms play a significant role in securing data by rendering it unreadable, with only authorized users able to decrypt and access the data using keys[1,2,3]. This research aims to enhance the efficiency of audio encryption by designing and developing a more robust algorithm[4]. A high-security encryption algorithm has been created to achieve this objective, employing robust encryption standards. The proposed approach improves encryption quality while minimizing resource consumption compared to traditional methods, offering substantial advantages. The paper contributes to network security by safeguarding the confidentiality of data, particularly audio data stored or transmitted over networks. It is valuable for professionals and researchers in fields such as system analysis, database administration, software development, information security analysis, data science, and network security.

Keywords: 3DES, AES, Blowfish, Cryptography, Cypher Encryption, Decryption, DES, Feistel network.

Introduction

The Feistel network is widely recognized as one of the most secure networks in the field of cryptography [6]. Its advantage lies in the similarity between encryption and decryption processes, which often require only a reversal of the key schedule to be effective [5]. However, the Feistel network's operation is well-known, and intercepted ciphertexts can be decrypted, posing a security risk [4,8]. To enhance the complexity of the Feistel network, a modified approach integrating genetic algorithms and mutation concepts has been introduced [9,10]. This modified Feistel network, when combined with the genetic algorithm and mutation ideas, enhances encryption quality without compromising the overall level of security [7]. The research findings demonstrate that this approach consumes fewer resources compared to traditional methods while producing higher-quality ciphertexts. This paper is relevant to network security, particularly for protecting the confidentiality of audio data stored or transmitted over networks. It is beneficial to professionals and researchers in fields related to network security.

The Blowfish Block Cipher

Blowfish, a block cipher developed by Bruce Schneier in 1994, achieves its security through the iteration of a basic encryption algorithm 16 times. It operates on 64-bit blocks and supports key lengths up to 448 bits, except for zero. Although a setup process is required before encryption, the actual encryption of data is straightforward. Blowfish consists of 16 rounds, divided into a key-dependent permutation and a key- and data-dependent replacement. All operations involve XORs and additions on 32-bit words, with only four indexed array data lookups performed per round. The method comprises two parts: key expansion and data encryption. Key expansion transforms a variable-length key into subkey arrays, while data encryption encrypts the actual data [11-12].

Modified F Function

The suggested algorithm incorporates a modified Feistel network, inspired by the genetic algorithm and mutation concepts. Unlike the original Feistel network, the modified approach introduces genetic crossover and flip bit mutation to enhance encryption quality. These modifications ensure that the values in the S-box and the XOR operation of the S-box values are distinct. The genetic crossover and mutation ideas, along with the modified Feistel network, are depicted in Figure 1, providing a detailed illustration of the concepts involved [13].

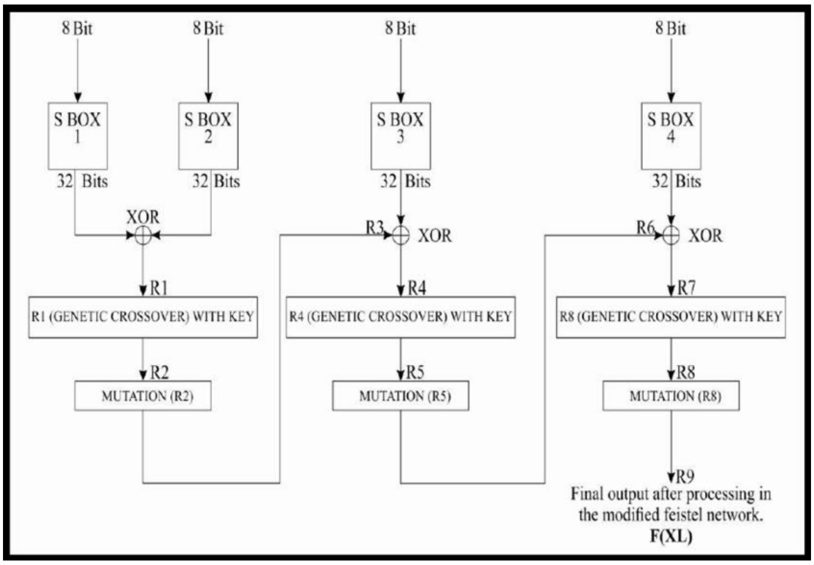
8 Bit
8 Bit
8 Bit
8 Bit
S BOX 1
S BOX 2
S BOX 3
S BOX 4
32 Bits
32 Bits
32 Bits
32 Bits
XOR
R3
XOR
R6
XOR
R1
R4
R7
R1 (GENETIC CROSSOVER) WITH KEY
R4 (GENETIC CROSSOVER) WITH KEY
R8 (GENETIC CROSSOVER) WITH KEY
R2
R5
R8
MUTATION (R2)
MUTATION (R5)
MUTATION (R8)
R9
Final output after processing in the modified feistel network.
F(XL)

Figure 1 Modified Feistel Network

Encryption Process

This section explains the process of encrypting information securely. Initially, the user-provided key is transformed into a machine-produced internal key, which is then used to encrypt the data. The resulting cipher text can be transmitted in two formats: ASCII and binary values. This dual format confuses potential interceptors, making the data useless to them if they do not have access to the binary form. Since the machine-produced internal key remains unknown to anyone, including the sender and recipient, the encryption process is unique and secure [12-14].

Decryption Process

The decryption procedure starts with the binary form of the cipher text as the input. The user's key is sent to the key creation module, which generates a new internal key for decrypting the data. Decryption, similar to the Blowfish method, involves reversing the encryption process. It is worth noting that decryption heavily relies on the machine-produced internal key. If the interceptor lacks knowledge of the lookup table values and does not have the binary representation of the cipher text, the data cannot be decrypted. This approach ensures superior performance compared to previous methods without compromising the quality of the cipher text [12-14].

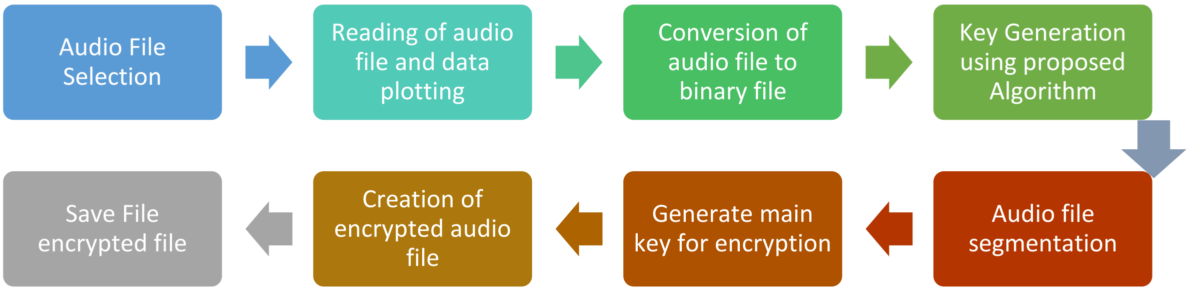
Audio File Selection
Reading of audio file and data plotting
Conversion of audio file to binary file
Key Generation using proposed Algorithm
Audio file segmentation
Generate main key for encryption
Creation of encrypted audio file
Save File encrypted file

Figure 2 Overall flow of working of proposed algorithm

Proposed Algorithm

The proposed algorithm aims to address two objectives. Firstly, it focuses on enhancing the speed of the encryption process, particularly the initialization time required by the standard Blowfish algorithm. Secondly, it aims to improve security and encryption standards by utilizing different parameters to enhance algorithm performance. The proposed algorithm offers increased speed, improved performance, and robust security for audio encryption.

Specifications Of Algorithm

The below mentioned specification are of proposed algorithm CRB for robust audio encryption technique.

Block Size: N/A

Key Size: 255 to 448-bits variable size (256 for main key)

Number of sub keys: 21 [CRB-array]

The proposed audio encryption algorithm consists of the following steps:

1. Encryption, 2. Encoding, 3. Decoding, 4. Decryption

Encryption Technique

Step 1: Selection of Audio File to encrypt

Step 2: Conversion of audio file to binary format using default methods

Step 3: Generation of Sub keys:

Sub keys will be generated using the proposed equation

Step 4: 1 main key generated with the 256 bits and other 21 sub keys {CRB[0]….CRB[20]} with size of 128 bits each are needed for both encryption and decryption process.

Step 5: One main key and 21 subkeys are stored in a CRB-array with each element being a 128 bit entry which illustrated as follow :

CRB_MAIN_KEY=”14QYQ0EQ8LQP634HOM6E8MS
N2E9XC5JQ9G6JAJEL1S1A3YOB56B82IJHG6G4IIZD
XLIW8BR2P6CBYWN6JYAXFLJZD6XZGHY8N7RQ4ED
4MKO7NS597FKNHYXSM3Z9FXYWHGB58CBSF95O
8SDQV06ZONHPXQTCI3PIOIXVN6BHK1PWO590KH
OUDFE0Y9JU5IMNZ5YR1UZRKBYNQQ10324DV6D
WF4S3RJK9LQS84FY946R6AF5FN1YHEQETF7A9AQ
Z”

CRB[0]=”M7HL0K72V5XWYM3LP092BJH4MKV1TH
7N77HCE71EIXXKTPUU3B081G2VI0X2P0FTB4XF6B
DLT0ZTOEV0AW93TIHU25OJ6D338G5YT4D7D2JSL
E3GG4MQ2QSDU22SKJUU”

CRB[1]=”FNC1M77LQJJ7QIZ1HQ6I7PQUJIWHJHYN
B7FINWSR7NS85T3C3ALP717NB5B6LZ4FU96EHJ31
9DZGTI9FHIEK1VZMVN0ZPEQETBHHXAKOH990NE
2CVAK930V0MS6WHCQV”

.

.

CRB[21]=”NC1M77LQJJ7QIZ1HQ6I7PQUJIWHJHYN
B7FINWSR7NS85T3C3ALP717NB5B6LZ4FU96EHJ31
9DZGTI9FHIEK1VZMVN0ZPEQETBHHXAKOH990NE
2CVAK930V0MS6WHCQV”

Step 6: Now on the first phase the main key will used for the encryption the file followed by all 22 subkeys.

CHIPER_TEXT=PLAIN_TEXT xor MAIN_KEY
CHIPER_TEXT=CHIPER_TEXT xor SUB_KEY[0]
CHIPER_TEXT=CHIPER_TEXT xor SUB_KEY[1]
.
.
.
CHIPER_TEXT=CHIPER_TEXT xor SUB_KEY[20]

The resultant CRB-array holds 21 sub keys that is used during the entire encryption process

Step 7: The encryption in proposed technique consist of two parts Rounds and Final Processing. The final created file can be send on the network which is fully protected.

Decryption Technique

The decryption process is quiet similar to that of encryption and the sub keys are used in reverse {CRB [20] – CRB[0]}. The entire decryption process can be elaborated as:

Step 1: Selection of Audio File to encrypt

Step 2: Conversion of audio file to binary format using default methods

Step 3: 21 sub keys {CRB [0]…CRB[20]} and main key are needed in decryption process.

These 21 sub keys are stored in a P-array with each array element being a 128-bit entry along with the main key with 256-bit.

Step 5: Now each of the sub key and main key are used in the decryption one by one as shown below.

DESC_TEXT=CHIPER_TEXT xor SUB_KEY[20]
DESC_TEXT=CHIPER_TEXT xor SUB_KEY[19]
DESC_TEXT=CHIPER_TEXT xor SUB_KEY[18]
.
.
.
DESC_TEXT=CHIPER_TEXT xor SUB_KEY[0]
DESC_TEXT=CHIPER_TEXT xor MAIN_KEY

(The resultant CRB-array holds 21 sub keys that is used during the entire decryption process)

Step 6: The decryption in proposed technique consists of two parts Rounds and Final Processing. The resultant CRB-array holds 21 sub keys that is used during the entire decryption process.

Step 7: At the end of all the process original file will be generated at DESC_TEXT.

Comparative Study And Result Analysis Of Proposed Work Development Of Crb Algorithm

Here in this section we have represented the detail output of the proposed work and algorithm. We have done the experiment on the five different audio (.mp3) files with different size namely Child, Cat, Bird, Bell and Cinematic. The main purpose of our research is to encrypt the audio file and proposed algorithm of the same. As a base of our algorithm we have used blowfish and in that we have done the f function modification and increased the bit level security. Here displayed the result analysis of each file along with status of wave signals. We have done the implementation using open source technology that is python. Result of five different sized file of audio:

TABLE-1 RESULT ANALYSIS: BELL.MP3 (1008.51 KB)

Algo Name	Key size	Block size	Cipher Type	Encryption Time (seconds)	Decryption Time (seconds)	Memory Utilization Encryption	Memory Utilization Decryption	Loss	Throughput (MB/ second)	Encry ption Ratio
AES	256 bits	128 bits	Block	0.032056	0.03203	880640	770048	No	31.46113	1
DES	64 bits	64 bits	Block	0.151982	0.04	311296	872448	No	6.635691	1
Triple DES	192 bits	64 bits	Block	0.207975	0.215944	839680	1486848	No	4.849166	1
Blowfish	256 bits	64 bits	Block	0.11205	0.055956	806912	786432	No	9.000534	1
Twofish	256 bits	64 bits	Stream	0.040017	0.023994	1478656	847872	No	25.20216	1
ARC 4	256 bits	64 bits	Stream	0.056029	0.06723	77824	385024	No	17.99976	1
CRB	256 bits	N/A	Stream	15.62194	14.15813	37638144	67043328	No	0.064557	1

TABLE-2 RESULT ANALYSIS: BIRD.MP3 (139.44 KB)

Algo Name	Key size	Block size	Cipher Type	Encryption Time (seconds)	Decryption Time (seconds)	Memory Utilization Encryption	Memory Utilization Decryption	Loss	Throughput (MB/ second)	Encry ption Ratio
AES	256 bits	128 bits	Block	0.0480292	0.024056	45056	0	No	2.9032765	1
DES	64 bits	64 bits	Block	0.0206919	0.0100114	217088	0	No	6.7389747	1
Triple DES	192 bits	64 bits	Block	0.0100188	0.0200129	0	0	No	13.917999	1
Blowfish	256 bits	64 bits	Block	0.0100284	0.5	0	0	No	13.904763	1
Twofish	256 bits	64 bits	Stream	0.5	0.0100114	49152	585728	No	0.278884	1
ARC 4	256 bits	64 bits	Stream	0.0100152	0.5	0	0	No	13.922969	1
CRB	256 bits	N/A	Stream	2.1508648	1.9277451	29777920	4014080	No	0.0648307	1

TABLE-3 RESULT ANALYSIS: BOMB.MP3 (216.31 KB)

Algo Name	Key size	Block size	Cipher Type	Encryption Time (seconds)	Decryption Time (seconds)	Memory Utilization Encryption	Memory Utilization Decryption	Loss	Throughput (MB/ second)	Encry ption Ratio
AES	256 bits	128 bits	Block	0.0100369	0.0099945	0	0	No	21.551776	1
DES	64 bits	64 bits	Block	0.0199685	0.0199966	57344	0	No	10.832756	1
Triple DES	192 bits	64 bits	Block	0.0400276	0.0500298	454656	348160	No	5.4041187	1

Blowfish	256 bits	64 bits	Block	0.0200338	0.0300026	40960	0	No	10.797433	1
Twofish	256 bits	64 bits	Stream	0.0099709	0.0200369	348160	0	No	21.694524	1
ARC 4	256 bits	64 bits	Stream	0.0099995	0.0099714	0	0	No	21.632452	1
CRB	256 bits	N/A	Stream	3.2233102	3.2477894	1396736	30855168	No	0.0671093	1

TABLE-4 RESULT ANALYSIS: CAT.MP3 (426.71 KB)

Algo Name	Key size	Block size	Cipher Type	Encryption Time (seconds)	Decryption Time (seconds)	Memory Utilization Encryption	Memory Utilization Decryption	Loss	Throughput (MB/ second)	Encryption Ratio
AES	256 bits	128 bits	Block	0.0300584	0.0199981	774144	303104	No	14.196073	1
DES	64 bits	64 bits	Block	0.040035	0.04003	442368	565248	No	10.658446	1
Triple DES	192 bits	64 bits	Block	0.0499699	0.0733678	606208	925696	No	8.5393587	1
Blowfish	256 bits	64 bits	Block	0.0199606	0.0600183	749568	413696	No	21.377619	1
Twofish	256 bits	64 bits	Stream	0.0300245	0.009995	667648	323584	No	14.21208	1
ARC 4	256 bits	64 bits	Stream	0.0400219	0.0199988	163840	323584	No	10.661939	1
CRB	256 bits	N/A	Stream	5.6276307	5.7909899	48439296	15978496	No	0.0758243	1

TABLE-5 RESULT ANALYSIS: TEST2.WAV (94.23 KB)

Algo Name	Key size	Block size	Cipher Type	Encryption Time (seconds)	Decryption Time (seconds)	Memory Utilization Encryption	Memory Utilization Decryption	Loss	Throughput (MB/ second)	Encryption Ratio
AES	256 bits	128 bits	Block	0.5	0.5	0	0	No	0.188456	1
DES	64 bits	64 bits	Block	0.0684352	0.0100322	0	0	No	1.3768939	1
Triple DES	192 bits	64 bits	Block	0.0200157	0.0299695	0	0	No	4.7077006	1
Blowfish	256 bits	64 bits	Block	0.0099998	0.5	53248	0	No	9.4230336	1
Twofish	256 bits	64 bits	Stream	0.5	0.0100067	0	0	No	0.188456	1
ARC 4	256 bits	64 bits	Stream	0.0099993	0.5	0	0	No	9.423483	1
CRB	256 bits	N/A	Stream	1.4658806	1.4877794	4603904	2162688	No	0.0642808	1

Conclusion

Based on the analysis conducted, the CRB algorithm was compared with AES, DES, Triple DES, and Blowfish, considering various factors such as key size, block size, cipher type, encryption time, decryption time, memory utilization, throughput time, losses, and encryption ratio. The findings indicate that the CRB algorithm outperforms the mentioned algorithms in terms of security and provides better protection against different attacks. It is a lightweight, lossless cryptographic algorithm with an exceptionally higher throughput ratio compared to others.

References

[1] Y. Gyawali, "ENCRYPTION ALGORITHM Advanced Encryption Standard," no. November, 2020.

[2] C. H. Lin, G. H. Hu, J. S. Chen, J. J. Yan, and K. H. Tang, "Novel design of cryptosystems for video/audio streaming via dynamic synchronized chaos-based random keys," Multimed. Syst., vol. 28, no. 5, pp. 1793–1808, 2022, doi: 10.1007/s00530-022-00950-6.

[3] Z. Chang and M. Woźniak, "Encryption technology of voice transmission in mobile network based on 3DES-ECC algorithm," Mob. Networks Appl., vol. 25, no. 6, pp. 2398–2408, 2020, doi: 10.1007/s11036-020-01617-0.

[4] S. Adhikari and S. Karforma, "A novel audio encryption method using Henon–Tent chaotic pseudo random number sequence," Int. J. Inf. Technol., vol. 13, no. 4, pp. 1463–1471, 2021, doi: 10.1007/s41870-021-00714-x.

[5] Y. Zhou, C. Li, W. Li, H. Li, W. Feng, and K. Qian, "Image encryption algorithm with circle index table scrambling and partition diffusion," Nonlinear Dyn., vol. 103, no. 2, pp. 2043–2061, 2021, doi: 10.1007/s11071-021-06206-8.

[6] Y. Guo, J. Yang, and B. Liu, "Application of chaotic encryption algorithm based on variable parameters in RFID security," Eurasip J. Wirel. Commun. Netw., vol. 2021, no. 1, 2021, doi: 10.1186/s13638-021-02023-0.

[7] A. Sharma and A. Mishra, "Cryptographic Algorithm For Enhancing Data Security : A Theoretical Approach," vol. 10, no. March, pp. 2–6, 2021, DOI : 10.17577/IJERTV10IS030158

[8] Valea, E., Da Silva, M., Flottes, M., Di Natale, G. and Rouzeyre, B. (2019) Stream vs block ciphers for scan encryption,. Microelectronics J., vol. 86, no. February, pp. 65–76, doi: 10.1016/j.mejo.2019.02.019.

[9] Chunguang, H. and Hai, C. (2015) Permutation of Image Encryption System Based block cipher and stream cipher encryption Algorithm IEEE, vol. 1, pp. 1–4, DOI: 10.1109/RVSP.2015.46.

[10] Sharif, S.O. and Mansoor, S.P. (2010) "Performance analysis of Stream and Block cipher algorithms", IEEE, pp. 522–525, DOI: 10.1109/ICACTE.2010.5578961.

[11] Nalawade, S.B. (2017) "Design and Implementation of Blowfish Algorithm using Reconfigurable Platform", IEEE, pp. 27–29, DOI: 10.1109/RISE.2017.8378204.

[12] Gowda, S.N. (2016) "Using Blowfish encryption to enhance security feature of an image Proc." 6th Int. Conf. Inf. Commun. Manag. ICICM 2016-IEEE, no. 200, pp. 126–129, DOI: 10.1109/INFOCOMAN.2016.7784228.

[13] Poonia, V. and Yadav, N.S. (2015) "Analysis of modified Blowfish algorithm in different cases with various parameters" ICACCS 2015 - Proc. 2nd Int. Conf. Adv. Comput. Commun. Syst. IEEE, pp. 5–9, DOI: 10.1109/ICACCS.2015.7324114.

[14] Panda, M. (2017) "Performance analysis of encryption algorithms for security", Int. Conf. Signal Process. Commun. Power Embed. Syst, SCOPES 2016 - Proceedings-IEEE, pp. 278–284, DOI: 10.1109/SCOPES.2016.7955835.