Journal of Advanced Engineering Research
ISSN: 2393-8447
Volume 9, Issue 1, 2022, pp. 21-28

Comparison of Cryptographic Algorithms for Wireless Underground Sensor Networks Security

Tanveer Kaur[1],*, Simarpreet Kaur[2]

[1]Student of Department of Electronics and Communication Engineering, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib-140404, India

[2]Assistant Professor (HOD) of Department of Electronics and Communication Engineering, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib-140404, India

*Corresponding author email: tanveerkaur238@gmail.com, Tel.: +91 9876418378

ABSTRACT

As computer networks, telecommunications, and the Internet become more interconnected and cyber-attacks more sophisticated, it is becoming increasingly critical to ensure privacy, security, authenticity, integrity, availability, and identity of data users. Cryptography is one approach to assure the privacy and security, endorsement, veracity, obtainability, in addition to identity of statistics users, as well as the data protection of data given to the user, as computer networks grow increasingly linked and cyber-attacks become more sophisticated. Cryptographic techniques are employed for data encryption and decryption. Encryption is the conversion of ordinary text into cypher text, which is unreadable by people and machines alike. A comparison of encryption methods for wireless subterranean sensor networks is carried out in this research to examine how well they function when applied to the underground network. Five algorithms were chosen to be employed on the wireless underground sensor network for security analysis after. Data Encryption Standard (DES), Triple DES (3DES), Triple Data Encryption Algorithm (TDEA), Advanced Encryption Standard (AES), Blowfish, in addition to RC4 are the five algorithms judged on their capability to safeguard records, the time it takes to scramble and decrypt data, how each technique manages keys, and how much energy each algorithm expends for encryption and decryption. The performance of the various algorithms changes depending on the inputs.

Keywords – Asymmetric, Cryptography, Decryption, Encryption, Security, Symmetric

1. INTRODUCTION

The manner of transforming original documents into cypher script in demand to obfuscate its connotation plus preclude the original data as of being recovered by an unauthorized receiver is known as encryption. As a result, encryption's fundamental objective is to maintain confidentiality. Data is typically encrypted before being sent over the internet to guarantee that it remains secure while in transit [1]. The encrypted data is sent via a public network, where it is unscrambled by the envisioned receiver. Various scrambling set of rules have been advanced and are extensively utilized in the arena of information security. It's possible to distinguish between asymmetric (private) and symmetric (public) keys [2].

2. ENCRYPTION ALGORITHMS

2.1 Categories of Scrambling Algorithm

Asymmetric (also named public-key algorithms) and symmetric (also known as secret-key) encryption algorithms use different types of keys to encrypt data [3].

2.2 Dissimilarities Amongst Symmetric and Asymmetric Encoding

2.2.1 Symmetric Encryption

Single-key encryption, also known as consistent encoding, practices only one key to scramble and descramble data or statistics. In homogeneous scrambling, an encrypting and decrypting key is utilized. The practice of scrambling and rearranging a zip file with the same key is known as homogeneous scrambling. Because the key must stay hidden from third parties, homogenous scrambling is also referred to as "secret key" scrambling [4]. The method's fundamental flaw is that it necessitates the key being securely shared between two parties before secure communication can take place. Hurry and strong point per key bit are two examples of fortes. This cryptographic primitive is the foundation of all cryptography [5].

2.2.2 Asymmetric Encryption

Irregular encryption, also acknowledged as public key cryptography, is a newer technique of encrypting and decoding data with dualistic keys: a secretive key and a unrestricted key In contradiction to symmetric encryption, typically requires one key to scramble and another to decipher data, asymmetric scrambling employs twofold keys to scramble and twofold keys to decipher data, it employs two keys to encode and decode data. The public key can be disseminated widely; however solitary the possessor has access to the reserved key [6]. The term "public key" refers to the fact that it can only be used to encrypt a communication, not decode it. It's termed a public key since it may be shared widely. The data is scrambled using the transmitter's public key, and the receiver's private key is used to decode it [7].

3. ENCRYPTION ALGORITHMS

This part contains an overview and cryptographic algorithms of five well-known methodologies: AES, DES, Blowfish, 3DES, and RC4 [8].

3.1 Data Encryption Standard (DES)

The Data Encryption Standard, or DES, is a block cypher by means of a symmetric key that converts all plain 64-bit text data into 48-bit keyed ciphertext blocks. This is the current NIST (National Institute of Standards and Technology) accepted standard. The term symmetric-key suggests that the technique decrypts and encrypts data using the same 48-bit key. Asymmetrical algorithms often require dual keys: one for encoding and the other for decoding [9]. The ciphertext of a 64-bit basic script is converted to 64-bit ciphertext. Because the techniques are asymmetric, the same key is employed for together encoding and decoding the text.
The stages of the algorithm procedure are as follows:

1. After being transmitted to the 64-bit basic script block, the 64-bit text block undergoes preliminary rearrangement. The 64-bit pure text n function block is used in the initial permutations function block [9].
2. A permuted block is split into two portions using IP: Leftward Plain Text and Rightward Plain Text.
3. Each LPT and RPT is encrypted 16 times.
4. On the newly combined LPT and RPT, an Ultimate Arrangement is performed.
5. The 64-bit cypher text required for encryption is generated in this step [10].

3.1.1 DES Modes of Operation

- Connoisseurs that utilize DES can choose from five dissimilar approaches of maneuver.
- A computerized codebook (ECB). Each 64-bit block is scrambled and unscrambled self-reliantly.
- Cipher Block Chaining (CBC) - Each 64-bit block uses an Initialization Vector and is conditional on the one preceding it [11].
- Comments on Ciphers (CFB) - The previous ciphertext unit is used as contribution for the encoding process, which produces The randomly generated result is then XORed with the original text to generate the following ciphertext unit (OFB). The encoding technique contribution is the result of the preceding DES; therefore it's similar to CFB.
- Opposite (CTR) - Each original text item is XORed with an encoded counter. The counter is subsequently increased for each subsequent block. After we have a better knowledge of what DES is, we will look at DES implementation [12].

3.2 Triple DES

The Triple DES Data Encryption Standard is a version of the Data Encryption Standard (DES). It employs a 64-bit key that contains 56 functional key bits and 8 parity bits. Triple-DES has an 8-byte block size. Eight-byte chunks of data are encrypted using Triple-DES. Triple DES ensures that DES is more secure by encrypting it three times with three different keys. Triple DES is extremely secure (large institutions employ it to protect critical transactions), but it is also quite slow [12]. Using Triple-DES encryption, keys are 112 - 168 bits in size, which encrypts data three times with different keys. If you do this, you should have 112 bits of strength, is more than enough to withstand brute force attacks [13]. It is faster than certain contemporary block ciphers, but more powerful than (single) DES. Since cryptographers identify triple DES as an insufficient long-term solution, NIST sent out a call for ideas regarding encryption standards in 1997 that would completely swap DES, the Advanced Encryption Standard (AES) (AES) [14].

3.2.1 Triple DES Modes

The Electronic Code Book (Electronic Code Book) is the triple list.

- The ECB mode of Triple DES works similarly to this variant.
- As the most common mode, it is also the most common.

Cipher block chaining (CBC) with triple encryption

- There is an excessive agreement of correspondence amongst this technique and the customary DES-CBC mode [15].
- The basic functions of CBC mode are also used. The effective key length of Triple ECB is 168 bits, and the keys are used as described above, but they are concatenated.
- Initialization vector is the first 64-bit key

Despite not being as widely used as Triple ECB, this method of protecting Triple DES allows it to be more secure than Triple DES [16].

Encryption-decryption follows the following steps:

- DES key K1 should be used to encrypt plaintext blocks.
- The result of step 1 should be encrypted using single DES encryption and key K2.
- Step 2's output should then be encrypted using single DES with key K3.
- Upon completing step 3, you will have the ciphertext [17].

3.3 *Advanced Encoding Standard (AES) Algorithm*

AES is an iterative secret message, in contrast to Feistel. This is accomplished through substitution-permutation networks. Permutations and substitutions are interconnected processes which require the movement of bits to accomplish other actions (permutations) [18].

In AES, all calculations are carried out with bytes relatively than bits. AES contemplates original text contained in 128 bits as 16 bytes as a result. Four columns and four rows make up these 16 bytes for matrix processing.

Like DES, AES allows you to adjust the number of rounds, based on the distance. AES uses 10 rounds for keys of 128 bits, 12 rounds for keys of 192 bits, and 14 rounds for keys of 256 bits. Each of these rounds' 128-bit round keys is compared with the original AES key [19].

The number of rounds (R) will increase as the cypher key size increases. For example, if the cypher key size is 128 rounds will be 10, 12, and 14 when the cypher key size is 192 and 14 when the cypher key size is 256 [20].

3.3.1 Progression of Encryption

The process is divided into three steps as follows.

1. Replacement of Bytes (Sub Bytes)

The 16 input bytes are substituted by searching up a pre-determined table (S-box) in the strategy. A four-row, four-column matrix is the end product. The rows have been rearranged. All four rows of the matrix have been moved to the left [21]. Any entries that are 'dropping off' are re-inserted on the row's right side. The process for completing the shift is as follows:

- The leading row has remained unchanged.
- The next row has been pushed one byte to the leftward, whereas the third row has been shifted two positions.
- The quarter row has been shifted to the left three spaces.
- With the same 16 bytes, but in a different order, a new matrix is generated [22].

2. Combine Sections

This approach substitutes four bytes from one area with four bytes from a different area. As an outcome, a new condition is produced with 16 more bytes. It should be noted that in the final round, this level is skipped. Include a key that is round [23]. The matrices' the round key's 128 bits are XORed with the 16 bytes, which are now considered as 128 bits. If this is the closing round, the scrambled content will be the output. Following that, the 128 bits are transformed to 16 bytes, and the procedure is repeated [24].

3. Progression of Decoding

Backwards decryption of an AES cypher text is equivalent to encryption in the first place. The four procedures in each cycle are conceded out in inverse mandate.

- Make a corpulent key.
- Arrange the columns differently.
- Move the rows around.
- Change the bytes

In a Feistel Cipher, in contrast, the sub-processes are switched in every round. This requires each cryptographic algorithm to be implemented individually while remaining closely linked [25].

3.4 Blowfish

Counterpane Systems president Bruce Schneier is credited with creating one of the most widely used public-domain encryption algorithms. Since 1993, blowfish has been used publicly. [26].

3.4.1 Blowfish Operation

Encrypting a 64-bit block cipher, blowfish uses a variable-length key. It consists of two parts.

- Generating sub keys: This step divides the key into sub keys of up to 448 bits each, totaling 4168 bits.

A basic routine that encrypts data involves 16 iterations in this method. The permutations and replacements in each cycle are keys and data based. The best use for Blowfish is to encrypt communications links where the key stays the same for a long time, but not packet switching, where the key will often change [28].

3.4.2 Sub Keys and Key Expansion

Key expansion divides 448-bit keys into 4,168-byte sub key arrays. Without sub keys, the Blowfish strategy, which employs a huge number of them, would be incomplete. Before any encoding or decryption can take place, these sub keys must be identified [29].

Eighteen 32-bit sub keys and four 256-entry 32-bit S-boxes make up Blowfish's P-array.

The following is a list of the sub keys:

1. A predetermined string of pi hexadecimal digits is used to make ready the P-array and S-boxes.
2. For P1, the early 32 pieces of the key have been XORed with the piece P2, the second 32 pieces, and so on until all pieces of the P-exhibit have been XORed.
3. As recently expressed, the methodology is used to encode every one of the zero strings [30].
4. The output from step 3 is replaced in the P1 and P2 arrays.
5. Blowfish is used to encrypt this output with changed sub keys.
6. Step 5's production changes P3 and P4 in the P-array.
7. This method is repeated up until all of the P-arrays and four S-boxes have been updated [31].

To generate all of the sub keys and processes, Blowfish runs 521 times in total, resulting in roughly 4 kilobytes (KB) of data.

3.5 RC 4

Data streams are encrypted using the RC4 algorithm. At a time Processes an input or unit of data. Byte data is made up of a few bits. The variable's length can be encrypted and decrypted in this approach. This solution eliminates the need to wait for a certain amount of data to be processed or to encrypt further bytes [32]. A block cypher, for example, analyses a given quantity of information at the similar period (typically a 64-bit or 128-bit block). Blowfish, DES, Idea, RC5, Safer, Square, Twofish, RC6, Loki97, and others are examples. RC4 is a licensed symmetric encryption stream developed by RSA Data Security, Inc...

3.5.1 RC4 Varieties

In addition to RC4, Spritz, RC4A, and VMPC are also available.

- AEAD is a key encryption mechanism, which is a cryptographic algorithm, dynamic random bit generator, and an algorithm of automatically generated random bits [33].
- A more powerful version of RC4 was shown by Souraduyti Paul and Bart Preneel, RC4A.
- In an acronym for Variably Improved Possible Combination Structure, VMPC represents Irregularly Reformed Rearrangement Configuration.
- There are two main differences between RC4A+ and RC4: the latter has a more complicated key schedule that requires about three times the processing time and the former has an expanded ability to operate autonomously, which requires four extra extractions in the S array for each byte output, which takes about 1.3 times the processing time. [34].

3.5.2 Technique for Encryption

1. The user enters an unencrypted file and a secret key.
2. The encryption engine generates the key stream using the KSA and PRGA techniques.
3. To generate the encrypted content, the key stream is now byte by byte XORed with the original message.

4. The encrypted content is then transferred to the designated recipient, who decentralizes the memo and retrieves the original plain text [35].

3.5.3 System for Decryption

1. The same byte-wise X-OR method is used to decrypt the Cipher text.

2. Assume that A is plain text and B is the key stream (A xor B) B xor B =A [36].

4. COMPARITIVE ANALYSIS OF DIFFERENT CRYPTOGRAPHIC ALGORITHS

Table 1 compares different cryptography algorithms based on key type, key size, block size, speed, security level, flexibility, and structure type.

Table 1. Comparison of different algorithms

Algorithm	**Key-Type**	**Key-Size**	**Block Size**	**Speed**	**Security Level**	**Flexible**	**Structure**
AES	Symmetric	128 bit	128, 192, 256 bit	fast	excellent	yes	Substitution, permutation
DES	Symmetric	56 bit	64 bit	moderate	adequate	no	feistel
3-DES	Symmetric	112 or 118 bit	64 bit	Very slow	adequate	yes	feistel
Blowfish	Symmetric	64 bit	32-448 bit	fast	excellent	yes	feistel
RC4	Symmetric	variable	40-2048 bit	slow	adequate	yes	feistel

In fig 1 comparison all five algorithms were taken for the calculations energy consumption in mJ where the results out to be that the Blowfish consumes the least amount of energy meanwhile the maximum amount is consumed by the 3-DES, while other three algorithms consumes adequate amount of power as shown in table 2.

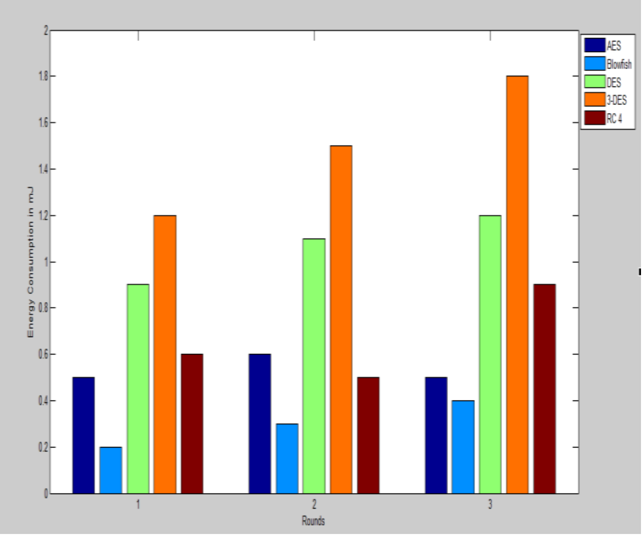

Figure 1. Representation of comparison of Energy Consumption of all five algorithms

Table 2. Comparison of power consumption in mJ by all five algorithms

Algorithms	**Power consumption in mJ**		
	Round 1	**Round 2**	**Round 3**
AES	**0.5**	**0.6**	**0.5**
Blowfish	**0.2**	**0.3**	**0.4**
DES	**0.9**	**1.1**	**1.2**
3-DES	**1.2**	**1.5**	**1.8**
RC 4	**0.6**	**0.5**	**0.9**

In fig 2 all the five algorithms are compared for encryption time taken individually by every method. It can be seen in the graph below and also in the table 3 that blowfish here again takes least encryption time while RC4 took maximum time for encryption

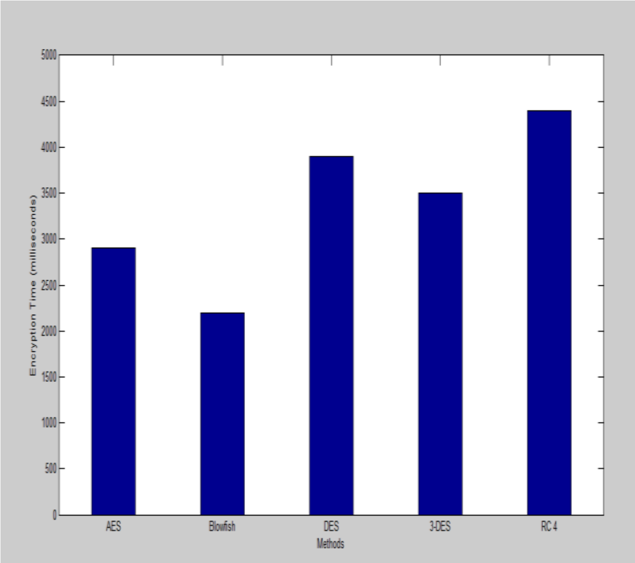

Figure 2. Representation of comparison encryption time taken by all five algorithms

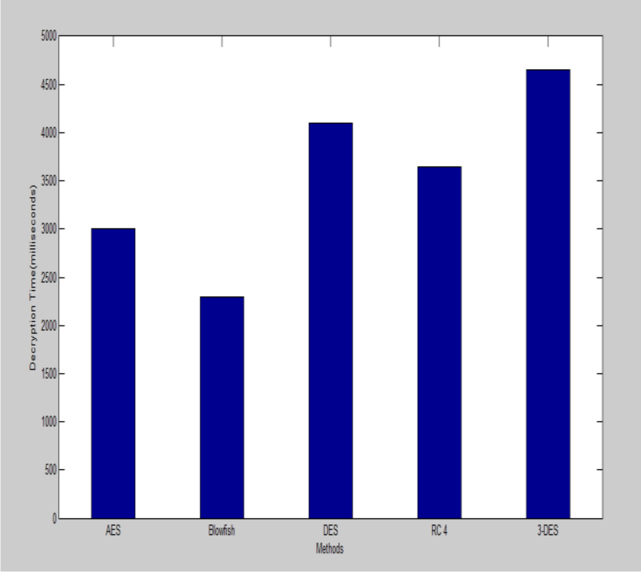

Figure 3. Representation of comparison decryption time taken by all five algorithms

Table 3. Evaluation of encipher time occupied by all five algorithms

Algorithm	**Encryption Time (in milliseconds)**
AES	**2900**
Blowfish	**2200**
DES	**3900**
RC 4	**3500**
3DES	**4400**

In figure 3 all the five algorithms are compared for decryption time taken individually by every method. It can be seen in the graph and table 4 below that blowfish here again takes least decryption time while 3DES took maximum time for decryption

Table 4. Assessment of decipher time occupied by all five algorithms

Algorithm	**Decryption Time (in milliseconds)**
AES	**3000**
Blowfish	**2300**
DES	**4100**
RC4	**3650**
3DES	**4650**

In the final study accompanied on all the five algorithms key storage is observed, as shown in the figure 4 and all the values given in the table 5, that means how much keys are stored by each algorithm for security, according to the outcomes blowfish stored maximum keys meanwhile DES and 3DES store almost same and least keys as compared to other protocols, AES stored second highest amount of keys after blowfish and RC4 is on adequate stage of key storage.

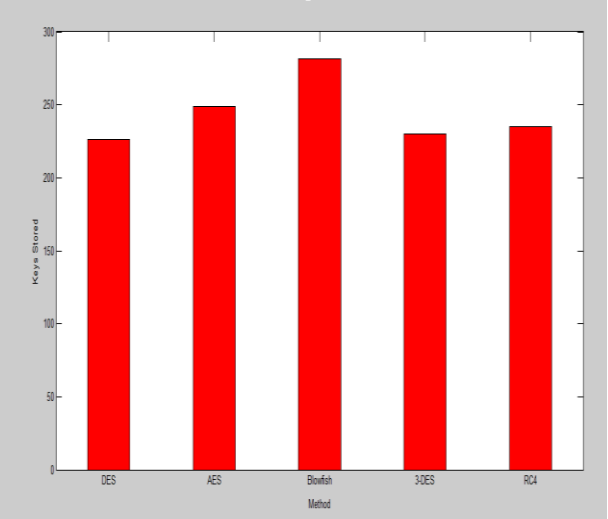

Figure 4. Representation of comparison of key stored by all five algorithms

Table 5. Comparison of key storage by all five algorithms

Method	**Key stored**
DES	**225**

AES	**250**
Blowfish	**280**
3-DES	**230**
RC4	**240**

5. CONCLUSION

In this paper most famous five algorithms such as: AES, DES, Blowfish, 3-DES, RC4 is equated by means of dissimilar constraints: key type, key size, encryption time and decryption time and also key stored by all algorithms individually. According to the result outcomes Blowfish algorithms is best suited for security in wireless underground sensor networks for secure and reliable communication which is followed by AES. 3-DES is least suited for security follower by RC4. In future works, eventually, these algorithms will be compared to see how they perform, and to overcome the flaws and speed the process of algorithms when used in wireless underground sensor networks.

REFERENCE

[1] T. S. Vamsi, P. Ramya, A survey on Underground Distributed Wireless Sensor Networks: Design & Research Challenges, *Journal of Innovation in Electronics and Communication Engineering, 8(1),* 2018, 35-45.

[2] S. Vyakaranal , S. Kengond, Performance Analysis Of Symmetric Key Cryptographic Algorithms, *International Conference On Communication And Signal Processing (ICCSP) IEEE,* 2018, 0411-0415.

[3] B. E. Hamouda, H. Hamouda , Comparative Study of Different Cryptographic Algorithms, *Journal of Information Security, 11(1),* 2020, 138-148.

[4] Dr. K. L. Bharti, Dr. V. Tiwari , A Brief Survey of Cryptography Techniques, *International Journal of Computer Science Trends and Technology (IJCST), 6(2),* 2018, 8-1.

[5] T. N. Lakshmi, S. Jyothi, Cryptography Algorithms - Issues On Recent Trends, *International Journal of Innovative Research and Advanced Studies (IJIRAS), 5(7),* 2018, 45-58.

[6] P. Singh, R.K. Chauhan, A Survey on Comparisons of Cryptographic Algorithms Using Certain Parameters in WSN, *International Journal of Electrical and Computer Engineering (IJECE), 7(4),* 2017, 2232-2240.

[7] NC Zynab, M. Jasim , Image Encryption Using Modification Blowfish Algorithm , *International Journal Of Advances In Scientific Research And Engineering (Ijasre), 6(3),* 2020, 125-136.

[8] Rashid, Muhammad, Flexible Architectures for Cryptographic Algorithms—A Systematic Literature Review, *Journal of Circuits, Systems and Computers, 28(3),*2019, 193-203.

[9] Barker, Elaine, and A. Roginsky, Transitioning the use of Cryptographic Algorithms and Key Lengths, *No. NIST Special Publication (SP) Journal Of National Institute of Standards and Technology,* 2018, 193-223.

[10] Sallam, Suzan, and B. D. Beheshti, A Survey on Lightweight Cryptographic Algorithms, *TENCON IEEE Region 10 Conference. IEEE,* 2018, 500-508.

[11] Chen, Yu-Chi, A New Reversible Data Hiding in Encrypted Image Based on Multi-Secret Sharing and Lightweight Cryptographic Algorithms, *IEEE Transactions on Information Forensics and Security, 14(12),* 2019, 3332-3343.

[12] Haque, M. Enamul, Performance Analysis of Cryptographic Algorithms for Selecting Better Utilization on Resource Constraint Devices, *2018 21st International Conference of Computer and Information Technology (ICCIT), IEEE,* 2018, 506-512.

[13] A.P. Parkar, M.N. Gedam, N. Ansari, S. Therese, Performance Level Evaluation of Cryptographic Algorithms, *InIntelligent Computing and Networking, Springer, Singapore, 14(6),* 2021, 157-167.

[14] C. Rathod, A. Gonsai, Performance Analysis of AES, Blowfish and Rijndael: Cryptographic Algorithms for Audio, *InRising Threats in Expert Applications and Solutions, Springer, Singapore, 11(87),* 2021, 203-209.

[15] H. Dibas, K.E. Sabri, A Comprehensive Performance Empirical Study of the Symmetric Algorithms: AES, 3DES, Blowfish and Twofish, *International Conference on Information Technology (ICIT), IEEE,* 2021, 344-349.

[16] E.A. Al-Kareem, R.S. Mohammed, A Review of the Most Effective Cryptography Techniques Based on Conventional Block Cipher and Lightweight, *1st Babylon International Conference on Information Technology and Science (BICITS), IEEE,* 2021, 257-262.

[17] A.P. Parkar, M.N. Gedam, N. Ansari, S. Therese, Performance Level Evaluation of Cryptographic Algorithms, *InIntelligent Computing and Networking, Springer, Singapore, 14(6),* 2021, 157-167.

[18] R.R. Kureshi, B.K. Mishra, A Comparative Study of Data Encryption Techniques for Data Security in the IoT Device, *InInternet of Things and Its Applications, 82(5),* 2022, 451-460.

[19] B. Rahul, K. Kuppusamy, Efficiency Analysis of Cryptographic Algorithms for Image Data Security at Cloud Environment, *IETE Journal of Research, 6(4),* 2021, 1-12.

[20] Hassan, B. Mohammed, Comparative Study of Encryption Algorithms for Data Security in WoT and IoT, *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12),* 2021, 2722-2727.

[21] A.K. Bermani, T.A. Murshedi, Z.A. Abod, A Hybrid Cryptography Technique for Data Storage on Cloud Computing, *Journal of Discrete Mathematical Sciences and Cryptography, 24(6),* 2021, 13-24.

[22] M.N. Ul Haq, N. Kumar, A Novel Data Classification-Based Scheme for Cloud Data Security Using Various Cryptographic Algorithms, *International Review of Applied Sciences and Engineering, 12(10),* 2021, 5-12.

[23] J.D. Gaur, A.K. Singh, N.P. Singh, Comparative Study on Different Encryption and Decryption Algorithm, *International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE,* 2021, 903-908.

[24] A. Dutta, A. Bhattacharyya, C. Misra, S.S. Patra, Analysis of Encryption Algorithm for Data Security in Cloud Computing, *InSmart Computing Techniques and Applications, Springer, Singapore, 36(6),* 2021, 637-644.

[25] J. Agarwal, M. Kumar, A.K. Srivastava, Estimation of Various Parameters for AES, DES, and RSA, *InEmerging Technologies in Data Mining and Information Security, Springer, Singapore, 17(6),* 2021, 275-283.

[26] M.N. Alenezi, H. Alabdulrazzaq, N.Q. Mohammad, Symmetric Encryption Algorithms: Review and Evaluation Study, *International Journal of Communication Networks and Information Security, 12(2)*, 2020, 256-72.

[27] J.D. Gaur, A.K. Singh, N.P. Singh, Comparative Study on Different Encryption and Decryption Algorithm, *International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE*, 2021, 903-908.

[28] A. Dutta, A. Bhattacharyya, C. Misra, S.S. Patra, Analysis of Encryption Algorithm for Data Security in Cloud Computing, *InSmart Computing Techniques and Applications, Springer, Singapore, 36(6),* 2021, 637-644.

[29] J. Agarwal, M. Kumar, A.K. Srivastava, Estimation of Various Parameters for AES, DES, and RSA, *InEmerging Technologies in Data Mining and Information Security, Springer, Singapore, 17(6),* 2021, 275-283.

[30] M.N. Alenezi, H. Alabdulrazzaq, N.Q. Mohammad, Symmetric Encryption Algorithms: Review and Evaluation Study, *International Journal of Communication Networks and Information Security, 12(2)*, 2020, 256-72.

[31] M.N. Anwar, M. Hasan, M.M. Hasan, J.Z. Loren, S.T. Hossain, Comparative Study of Cryptography Algorithms and Its' Applications, *International Journal of Computer Networks and Communications Security, 7(5)*, 2019, 96-103.

[32] S. Pamidiparthi, S. Velampalli, Cryptographic Algorithm Identification Using Deep Learning Techniques, *InEvolution in Computational Intelligence, Springer, Singapore, 117(6),*2021, 785-793.

[33] Hanchinamani, Gururaj, and R. Savakknavar, Design of S-Box Based on Chao Initialized RC4, In *2021 International Conference on Computer Communication and Informatics (ICCCI), IEEE*, 2021, 1-4.

[34] S. K. Mousavi, A. Ghaffari, S. Besharat, & H. Afshari, Improving the Security of Internet of Things Using Cryptographic Algorithms: A Case of Smart Irrigation Systems, *Journal of Ambient Intelligence and Humanized Computing, 12(2)*, 2021, 2033-2051.

[35] K. Elavarasi, & J. Deepa, (2021), Self-Powered Cardiac Pacemaker Using RC4 Algorithm, In *Journal of Physics: Conference Series, IOP Publishing, 1717(1)*, 2021, 012-061.

[36] Salih, M. Huda, and R. Salam, A. Mahdawi, The Security of RC4 Algorithm Using Keys Generation Depending on user's Retina, *Indonesian Journal of Electrical Engineering and Computer Science, 24(1)*, 2021, 452-463.