International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
***Article Received**: 10 April 2022 **Revised**: 18 May 2022 **Accepted**: 02 June 2022 **Publication**: 30 June 2022*

AL-TEA: Alternative Tea Algorithm for Healthcare Image in IoT

Choudhary Leeladevi Babulal*
Department of CSE,
JJTU, Jhunjhunu, Churu Rd, Vidyanagari,
Churela, Rajasthan 333001,
India

Anand Sharma**
School of Engineering and Technology,
Mody University, Lakshmangarh,
Sikar, Rajasthan 332311,
India

**E-mail: leela928choudhary@gmail.com*
***E-mail: anand_glee@yahoo.co.in*

Abstract
Millions of devices are predicted to be connected via the Internet of Things (IoT), which is a promising technology for the future. In numerous industries, interest in leveraging the Internet of Things is predicted to expand. Various IoT applications in the healthcare industry are being studied, and the potential for IoT to improve healthcare will be huge. The rise in communications is likely to result in mountains of data, posing a danger to data security. The architecture's gadgets are substantially smaller and less powerful. Due to their complexity, traditional encryption algorithms are computationally demanding, requiring a significant number of rounds for encryption, and draining the limited power of devices. A less sophisticated method, on the other hand, may jeopardise the desired result. Many encryption techniques have recently been suggested to guarantee the security of data transmission across the Internet of Things. Because it requires less memory and is simple to implement in both hardware and software, of all the algorithms, the Tiny Encryption Algorithm (TEA) seems to be the most appealing. TEA has a number of flaws, particularly when it comes to equivalent keys and matching key assaults. As a result, in this study, we present "AL-TEA: An Alternative TEA Technique for Healthcare Images in the IoT," a lightweight encryption algorithm.

Keywords: IOT, TEA, Lightweight cryptography, LSB, Image Steganography

Subject Classification: *G16Y 10/60*

1. Introduction

The Internet of Things (IoT) brings the virtual and physical worlds together by creating an integrated communication environment of networked devices and platforms [1]. The sharing of medical data has become a regular habit with the introduction of IoT devices based on digital health remotely. As a result, an effective methodology for ensuring It is necessary to develop the confidentiality and reliability of diagnostic information compiled and obtained from the IoT environment. This is accomplished by hiding digital information in the image using a combination of steganography techniques and system encryption algorithms. The confidentiality and reliability of diagnostic information has become a key problem for healthcare applications as the Internet of Things (IoT) has grown in importance in the medical industry [2]. This approach can also be used in the healthcare industry, in situations where a patient's medical report, record, and history must be sent to a new institution for treatment or other reasons. The secrecy of the report between the doctor and the patient is essential in medical science. The main aim of this study is to improve the secureness of medical data transmission using a lightweight algorithm in order to create a highly dependable healthcare system that consumes less power.

2. Steganography in Healthcare

Because hiding communications using cryptography appears to be "simple" cryptanalysis (although this is a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
***Article Received**: 10 April 2022 **Revised**: 18 May 2022 **Accepted**: 02 June 2022 **Publication**: 30 June 2022*

subjective term), it's critical that we utilise a more secure approach (in this instance steganography) to hide texts on media like photos, audio, and video. This research focuses on leveraging the least significant bit (LSB) approach to develop picture steganography for safer data transport, particularly over the Internet [3]. Picture steganography is a technique for hiding data in images so that the original image remains unchanged.

2.1. To carry out the work, image steganography necessitates the following elements:

- **Cover medium:**

This is an image with a message concealed inside it.

- **The Secret message:**

This is the message that has to be sent out. Text, pictures, or other data can be plain or encrypted.

- **The Stego-key:**

This is the code that was used to encrypt the message (it may or may not be used).

EXAMPLE:

We can make use of picture files to conceal data in steganography by replacing the final bit of each byte of color with a bit from the message. Here the message:

Message: 11110110

An image with three-pixel

Pixel 1 : 10101010 11010101 11101111
Pixel 2 : 11111000 11001001 11101111
Pixel 3 : 11111000 11001001 11101111

Now we hide the message in this image.

Message: 11110110

Pixel 1 : 1010101**1** 1101010**1** 1110111**1**
Pixel 2 : 1010101**1** 1101010**0** 1110111**1**
Pixel 3 : 1010101**1** 1101010**0** 11101111

New image:

3. Related Work

The most essential encryption techniques for data transmission in IoT networks are discussed here. We go into the design and implementation of these algorithms in detail, as well as their flaws and issues. We also go into the concept, implementation, and shortcomings of TEA and its newest versions in great depth. cryptography algorithms are just a few examples: Tiny Encryption Algorithm (TEA) and TEA Extension (XTEA) [4]. Tiny Encryption Algorithms are the subject of a paper. The TEA is a cryptographic algorithm that is meant to save space with time. This is a Feistel-type encryption which employs mixed (orthogonal) algebraic groups to operate. Also offered was a modest encryption method (TEA) for various random number generator test criteria [5]. The TEA is amongst the most simple hardware implementation techniques. Where there is a time restriction, it can be mentioned. The trade-off between the required security level in terms of the number of cycles and the time to encrypt or decrypt data, for example, can be closed. Validating random numbers is critical in all cryptographic applications. Because there are no widely agreed criteria for measuring and clarifying cryptography's capabilities, it's best to do a series of tests on various numbers to identify their strengths and shortcomings. In order to do this, we conducted a set of randomness and security experiments, which are detailed in this article [6]. The use of the same key in all stages of encryption is one of the fundamental drawbacks of TEA and its numerous modified variations, which leads to a reduction in security, as evidenced by the avalanche effect of the algorithm [7]. Furthermore, text encoding and decoding take a long time, reducing the embedded IoT network's efficiency. By providing extra key confusions for each encryption round, these offers a NTSA (novel tiny symmetric encryption algorithm) that gives improved security mechanism for text file transmission over IoT [8]. In comparison to the complex current encryption techniques, the suggested NTSA algorithm is safer and more efficient. The flow diagrams of the encryption and decryption processes are shown in Figures 1 and 2.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
***Article Received**: 10 April 2022 **Revised**: 18 May 2022 **Accepted**: 02 June 2022 **Publication**: 30 June 2022*

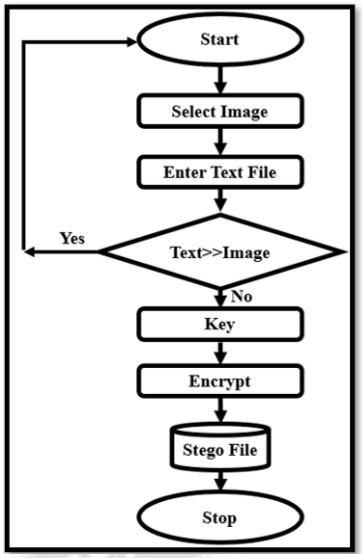

Figure 1: Image Encryption Flow Chart

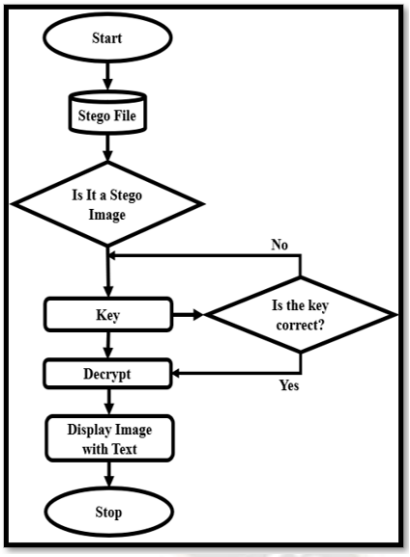

Figure 2: Image Decryption Flow Chart

The symmetrical key block cypher RC5 was one of the earliest efforts in this field. Many flaws exist in TEA, primarily as a result of assaults using similar and related keys. As a result, a Modified TEA Algorithm (MTEA) is presented, which employs a Linear Feedback Shift Register (LFSR) to mitigate the TEA algorithm's weaknesses against assaults. The MTEA method is implemented in this paper, along with a comparison of the regular TEA algorithm in terms of area and energy usage [9]. To achieve security, plain text is encrypted and decrypted in TEA using orthogonal algebraic group operations with lots of rounds. A total of 2883 gates are utilised throughout the TEA encoding procedure along with a 16.73ns delay, and 2805 gates are utilised throughout the decoding procedure with a 14.78 ns latency in a round of 64 feistel cycles. The architecture may be adapted to mobile devices that demand a high level of security based on these findings [10]. Using a 128-bit key and a 64-bit block size, there are 32 rounds. However, the algorithm in question is vulnerable to saturation assaults. The suggested SIT is a lightweight encryption technique that improves data transfer security among Internet of Things (IoT) devices SIT employs a Feistel structure and a network with similar substitution-permutation properties [11].

4. Proposed Work

When in comparison to other cryptographic algorithms like IDEA and AES, the TEA is a lightning-fastest, simplest, and Feistel-based structure that is one of the fastest and most efficient. The TEA, which was established by David J. Wheeler and Roger M. Needham in 1994, has a number of flaws, mostly owing to comparable key attacks and key attacks.

4.1. Notation

Here are some notations which are used in TEA.

- **Exclusive-OR:**
  The x⊕y the operation Exclusive-or.
- **Bitwise Shifts:**
  The logical left shift of x by y bits is represented by x<<y.
  The logical left shift of x by y bits is represented by x>>y.
- **Integer Addition:**
  It is indicated by x 2n y (where x, y, ∈). The context should reveal the value of n. 2nZ
- **Integer Subtraction:**
  2n denotes the integer subtraction modulo operation. x y (where x, y ∈)

4.2. Feistel Cipher:

Using the substitution and permutation option, a Feistel cipher encryption structure is presented to disguise the numerical properties of plain text. Claude Shannon was the first to propose the development of block codes to replace diffusion and confusion. Shannon was worried about cryptanalysis, which assumes that some communications in many languages have a frequency distribution, such as letters, words, or phrases, which may be transmitted to the ciphertext with the same frequency distribution. If an attacker has a basic understanding of these statistics, they can get the encryption key and encrypt the plain text.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
***Article Received**: 10 April 2022 **Revised**: 18 May 2022 **Accepted**: 02 June 2022 **Publication**: 30 June 2022*

Shannon proposes diffusion and confusion as two ways for masking the numerical properties of plain text [12].

- Diffusion:

The purpose is to make statistical analysis of the link between plain text and ciphertext as difficult as possible to prevent an attacker from obtaining a key.

- Confusion:

Its purpose is to make the statistical analysis of the relationship between the ciphertext and the encryption key as difficult as possible so that an attacker cannot uncover the key.

- Substitution:

After being applied to the right half of the plaintext, function F is performed in the left half, and its result is XORed with the left half. Each round, the structure of function F is the same.

- Permutation:

Each round is completed by substituting two bits of plain text for one another. The decryption method is identical to the encryption process, with the exception that the function F sub key is utilised in reverse order. Many symmetric block ciphers, such as Data Encryption Standard (DES) and Tiny Encryption Algorithm, employ the Feistel cypher architecture (TEA).

4.3. The Encryption of TEA :

At start, the 64-bit image is divided into two 32-bit inputs (which are X and Y). The user can choose from four passkeys to use as their passcode. These passkeys are assigned in the TEA architecture as inputs. The whole TEA encryption technique is depicted in Figure 3. The first four bits of the Y input value are left-shifted, and the output is further added with K[0], the first passkey, and the output is taken in the Y_1 memory. The initial Y input value is multiplied by a Golden Ratio constant (GRC) with a decimal value of 2654435769, and the output is recorded in the Y_2 memory. The following step is to three-bit right-shift the initial Y input value, then add the result with the second passkey, K[2], and record the result in the Y_3 memory. The final output is then saved in the Y_4 memory. Furthermore, the Y_4 value is added to the X, and the result is stored in memory X_0.

By right-shifting the value of X_0 by 5-bits and adding the result to the third passkey, K[1], this encryption process is forwarded for next round. Then the result is saved in a memory called X_1. The method is repeated, this time addition to the X_0 value to the GRC and recording the result in the X_2 memory. After that, the value of X_0 is used to make a 2-bit left shift, which is then summed with K[3], the fourth passkey. Then the result is recorded in X_3.

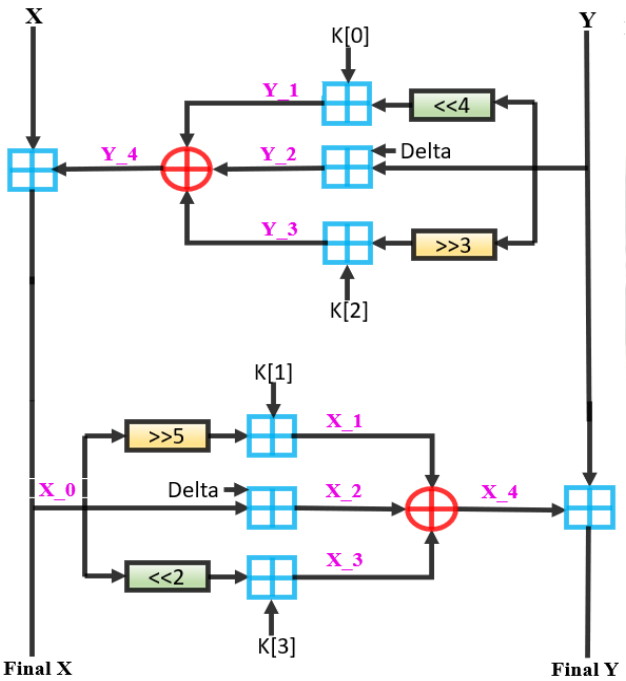

Figure 3: AL-TEA Encryption Process

For all three values in X_1, X_2, and X_3, the XOR procedure is used once more. This means that X_1 is XORed with X_2, with the result being XORed again with X_3. After that, The X_4 memory is used to save the final result. In addition, the X_4 value is to be added to the Y input value, and then the result kept in Y_0. Final_X and final_Y are the names of the output (encrypted image). The second TEA round is now complete

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
Article Received: 10 April 2022 Revised: 18 May 2022 Accepted: 02 June 2022 Publication: 30 June 2022

Above TEA encryption complete cycle is now repeated 32 times to meet the criterion.

4.4. Operation of XTEA in iteration

Delta = The magic constant 2654435769
Right-Shift = >>
Left-Shift = <<
XOR = ⊕
X_0 = ((Y<<4 + K[0] ⊕ (Y + delta) ⊕ (Y>>3 + [2])) ⊕ X
Y_0 = ((X_0>>5 + K[1] ⊕ (X_0 + delta) ⊕ (X_0<<2 + [3])) ⊕ Y

4.5. The Decryption of TEA

TEA decryption is nearly identical to TEA encryption, with the exception that the function is reversed. Figure 4 shows how the TEA decryption mechanism works. The algorithm's input is now the decryption process, which began with the encrypted image, with final_X referred to as input_X and final_Y referred to as input_Y. The output is further added to the second passkey, K[1], and the result is recorded in memory as X_1. After that, the GRC is multiplied by the input_X, and the resultant is saved in X_2. The input_X value is then left-shifted by two bits, and added with K[3], the fourth passkey and then it is saved in the X_3 memory. The X_1 was XORed with X 2, and resultant is XORed with the X_3 once again to ample the decryption. After that, the final result is recorded in memory as X_4. Utilizing the one-round TEA decryption at this point, it lower the value of X_4 by the input_Y value and save in memory X_0. By left-shifting the X_0 by 4-bits and adding the result to the first passkey, K[0], the decryption process is forwarded to the next round. After that, the output is saved in the Y_1 memory. Multiplying the X_0 value by the GRC and recording the resultant in Y_2 memory is how the operation is carried out. The X_0 value is then used to perform a 3-bit right-shift, with the output being added to the value of the third passkey, which is K[2].

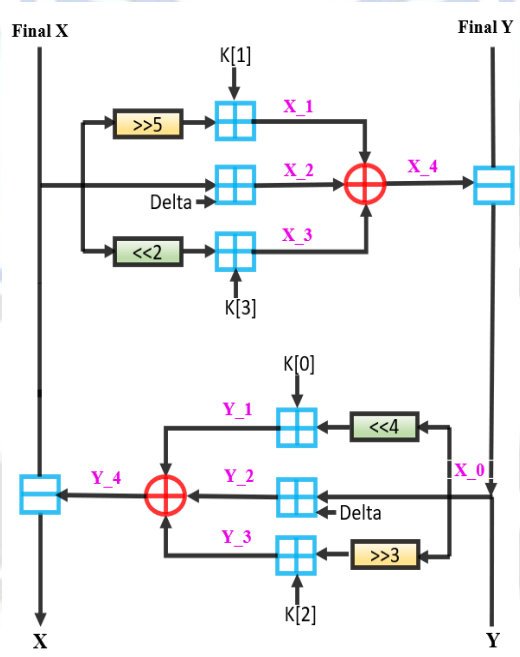

Figure 4: AL-TEA Decryption Process

After that, the final resultant is saved in the Y_3. XORing the Y_1 with the Y_2 and then XORing the result with the Y_3 completes the decryption process. The name Y 4 is subsequently assigned to this final outcome in the memory. Furthermore, the input_X must be used to reduce the value of Y_4, and the resultant is saved in the Y_0. To achieve the criterion of a full TEA decryption, The TEA cycle has now been completed thirty-two times. To take the same value or message, the output (decrypted image) is compared to the encryption process's input (original image).

4.6. Implementation and result

MATLAB is used to develop TEA. To begin, a new file is generated in the MATLAB programme, and the TEA encryption code is built in MATLAB. After that, the TEA encryption result is simulated. Following the successful modelling of the TEA encoding, the TEA decoding code is constructed. The TEA decryption result has been

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
Article Received: 10 April 2022 Revised: 18 May 2022 Accepted: 02 June 2022 Publication: 30 June 2022

simulated. By putting an image into TEA encoding process and getting the encoded image, both the encryption and decryption outputs may be compared. To acquire the decrypted image, insert the encrypted image into the TEA decryption procedure. Finally, the decrypted picture is compared to the original image, and if the two images are identical, the encryption and decryption processes have been effective. This proposed work was completed using MATLAB R2020b and a MacBook Air.

4.7. Performance Comparison of proposed AL-TEA with TEA, XTEA, XXTEA, NTSA

Tables 1 and 2 compare the encryption and decryption timings of AL-TEA, TEA, XTEA, XXTEA, and NTSA using a key of 128 bits with file sizes ranging from 0.37 kB to 26.7 kB. It's worth noting that, compared to all other IoT security algorithms, AL-TEA achieved a quicker encryption time with a larger key size, as shown in Table 1. The AL-TEA technique takes 0.4ms to encrypt a 0.37 kB image file with a 128-bit key, which is substantially quicker than the TEA, XTEA, NTSA, and XXTEA algorithms. Because to the AL-design TEA's simplicity, the approach was able to achieve quicker encryption speeds with a variety of file sizes and key sizes.

Table 1: Encryption Time for key size of 128 bits

FILE SIZE (in kilo bytes)	ENCRYPTION TIME (in milliseconds)				
	TEA	XTEA	BLOCK TEA (XXTEA)	NTSA	AL-TEA
0.37	0.059	0.125	0.068	0.51	0.4
0.95	0.126	0.264	0.158	0.109	0.106
1.6	0.198	0.423	0.232	0.189	0.172
2.6	0.332	0.686	0.384	0.221	0.22
6.8	0.696	1.584	0.743	0.548	0.573
8.6	0.948	1.669	1.171	0.899	0.872
12.2	1.277	2.807	1.535	1.02	1.02
16.2	1.12	3.263	1.864	1.1	1.1
26.7	2.209	5.207	2.224	1.983	1.942

Table 2 reveals that the decryption timings followed the same trend. When it comes to decryption, AL-TEA surpassed its competitors with a key size of 128-bit and a variety of file-sizes. AL-TEA took 0.44 milliseconds to decode a 0.37 kB picture file, compared to 0.058 milliseconds, 0.143 milliseconds, 0.068 milliseconds, and 0.49 milliseconds for TEA, XTEA, XXTEA, and NTSA for similar file sizes. In 1.02, 1.09, and 1.952 milliseconds, AL-TEA encrypted files with size of 12.2 kB, 16.2 kB, and 26.7 kB, respectively. Even with bigger text files, AL-TEA was able to achieve faster decryption speeds, similar to the results observed during encryption. Because of its quicker encryption and decryption speeds with 128-bit keys, AL-TEA might be the appropriate security mechanism for mobile devices and other IoT devices with different processing capabilities for successfully sending information.

Table 2: Decryption Time for key size of 128 bits

FILE SIZE (in kilo bytes)	DECRYPTION TIME (in milliseconds)				
	TEA	XTEA	BLOCK TEA (XXTEA)	NTSA	AL-TEA
0.37	0.058	0.143	0.068	0.49	0.44
0.95	0.125	0.276	0.179	0.101	0.102
1.6	0.195	0.43	0.233	0.174	0.171
2.6	0.324	0.673	0.388	0.311	0.232
6.8	0.673	1.57	0.75	0.556	0.557
8.6	0.936	1.641	1.21	0.889	0.863
12.2	1.241	2.764	1.538	1.03	1.02
16.2	1.111	3.184	1.956	1.1	1.09
26.7	2.179	5.178	2.193	1.989	1.952

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v10i6.5620
Article Received: 10 April 2022 Revised: 18 May 2022 Accepted: 02 June 2022 Publication: 30 June 2022

5. Conclusion

The Internet of Things will become a fundamental part of our daily lives in the future. For this aim, AL-TEA: ALTERNATIVE TEA ALGORITHM FOR HEALTHCARE IMAGE IN IOT is a lightweight security algorithm detailed in this paper. The purpose of this project is to employ image steganography with the Least Significant Bit (LSB) technology to make data transmissions more secure, especially over the internet. There are a few weaknesses in TEA, the most famous of which are attacks that use identical keys and related-key attacks. As a result, we've developed a strategy for overcoming the issue of equivalent keys using the AL-TEA described algorithm.

6. Future Scope

One of the easiest algorithms to implement in hardware is TEA (Tiny encryption algorithm). We want to use our algorithm in the defence and military sectors in the future to create systems that fulfil throughput, power consumption, and security requirements for sensitive data transmitted over public networks.

References

[1] Ploennigs, J.; Cohn, J.; Stanford-Clark, A.”The Future of IoT”. IEEE Internet Things Mag. 2018, 1, 28–33.

[2] CL Babulal, A Sharma (2021) ”Modified DES Cryptosystem With Steganography For Healthcare Systems in IoT” Design Engineering, Pages 5530-5538

[3] Mukesh Kumar Gupta, Pankaj Dadheech, Ankit Kumar, Sanwta Ram Dogiwal, Ramesh Chandra Poonia, Linesh Raja & Devershi Pallavi Bhatt (2022) Detection and localization for watermarking technique using LSB encryption for DICOM Image, Journal of Discrete Mathematical Sciences and Cryptography, DOI: 10.1080/09720529.2021.2009193)

[4] Paul E.A. Adriaanse, Miray Ayşen, Zekeriya Erkin. “A Comparative Study of the TEA, XTEA, PRESENT and Simon lightweight cryptographic schemes”. Cyber Security Group Department of Intelligent Systems Delft University of Technology. 14-7-2021

[5] Vishal Sharma, Anand Sharma (2021) “IoT Security Architecture with TEA for DoS Attacks Prevention” Advances in Information Communication Technology and Computing, Springer, Singapore, 215-226

[6] Hun, Stephanie & Md Naziri, Siti Zarina & Idris, Norina. (2012). “The Development of Tiny Encryption Algorithm (TEA) Crypto-Core for Mobile Systems”. International Conference on Electronic Devices, Systems, and Applications. 10.1109/ICEDSA.2012.6507813.

[7] Virmani, Dr. Deepali & Beniwal, Nidhi & Mandal, Gargi & Talwar, Saloni. (2013). “Enhanced Tiny Encryption Algorithm with Embedding (ETEA)”. INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY. 7. 10.24297/ijct.v7i1.3479.

[8] Deepak Mathur, N. K. V. . (2022). Analysis & Prediction of Road Accident Data for NH-19/44. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(2), 13–33. https://doi.org/10.17762/ijrmee.v9i2.366

[9] R. M. De Leon, A. M. Sison and R. P. Medina, "A Modified Tiny Encryption Algorithm Using Key Rotation to Enhance Data Security for Internet of Things," 2019 International Conference on Information and Communications Technology (ICOIACT), 2019, pp. 56-60, doi: 10.1109/ICOIACT46704.2019.8938456.

[10] M. R. Adiwiganda, E. Ariyanto, R. Yasirandi, N. A. Suwastika and Y. A. Setyoko, "Adopting Tiny Encryption Algorithm for Patient Healthcare Record on Smart Card," 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), 2019, pp. 1-5, doi: 10.1109/ICoSNIKOM48755.2019.9111534.

[11] Rajesh, Sreeja & Paul, Varghese & Menon, Varun & Khosravi, Mohammad. (2019). “A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices”. Symmetry. 11. 293. 10.3390/sym11020293.

[12] Abdelhalim, mohamed b & El-Mahallawy, Mohamed & Ayyad, Mohammad. (2013). Design and Implementation of an Encryption Algorithm for use in RFID System. International Journal of RFID Security and Cryptography. 2. 51-57. 10.20533/ijrfidsc.2046.3715.2013.0007.

[13] Ljubomir M. Vračar, Milan D. Stojanović, Aleksandar S. Stanimirović, Zoran D. Prijić, "Influence of Encryption Algorithms on Power Consumption in Energy Harvesting Systems", Journal of Sensors, vol. 2019, Article ID 8520562, 9 pages, 2019.

[14] Gill, D. R. . (2022). A Study of Framework of Behavioural Driven Development: Methodologies, Advantages, and Challenges. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(2), 09–12. https://doi.org/10.17762/ijfrcsce.v8i2.2068