ETSET: Enhanced Tiny Symmetric Encryption Techniques to Secure Data Transmission among IoT Devices

S. Alexander Suresh SDB[a] and Dr. Jemima Priyadarsini[b]
[a]
Research Scholar, Department of Computer Science, Bishop Heber College, Trichy
[b]Research Supervisor & Associate Professor, Department of Computer Science, Bishop Heber College, Trichy

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: The Internet of Things (IoT) is an assembly of physical devices or objects that link through the Internet to deliver interaction and communication among each other. In recent years, the growth of IoT is rapidly tremendous, and it starts to occupy people's lives in many areas such as hospitals, homes, roads and much more. IoT devices and data transfer create many new threats to the data transfer between the connected IoT devices. Securing the data in the IoT environment is the most important concern. Due to the IoT devices' computation complexity, a strong, lightweight security technique is proposed in this paper. This proposed technique is an Enhanced Tiny Symmetric Encryption Technique (ETSET). It is enhanced from the existing TEA (Tiny Encryption Algorithm). The major issue in TEA is, it uses the same key in all cycles of encryption and decryption. It can be broken using a related-key attack. The proposed ETSET changes the key generation in a dynamic approach for each cycle. ETSET also reduces the number of rounds and enhance dynamic bit shifting in each round. The proposed ETSET is implemented using JAVA and compared with TEA, XTEA. The results show that the proposed ETSET produces a better solution with respect to encryption and decryption time.

Keywords: TEA, IoT, IoT Security, XTEA, encryption

1. Introduction

It is prophesied that the quantity of attached Internet of Things (IoT) devices will rise to 38.6 billion by 2025 and an expected 50 billion by 2030 [1]. The enlarged deployment of IoT devices into assorted areas of people's life has provided substantial assistance, such as enhanced quality of life and job automation. Conversely, each time a new IoT device is connected, new and matchless security dangers emerge or are hosted into the environment on which the device is operated [2]. Instantaneous detection and mitigation of every security threat introduced by different IoT devices deployed can be very challenging. Security is a restraint process, protection against fraud, damage, theft, invasion of privacy, illegal entry and other events caused by deliberate action. Threats to break security are constantly evolving and increasingly sophisticated. Hence, security is a critical aspect of the Internet of Things data. The confidential and sensitive data is stored and transferred between IoT devices; it should be secured to maintain the data's safety. The conventional safety techniques are not adapted to the IoT environment due to their computational complexity and difficulty implementing the IoT environment. IoT requires an appropriate security mechanism for IoT data being transferred between devices.

Generally, cryptography techniques are used to secure data in the network. The cryptographic approaches of IoT networks can be classified as symmetrical and asymmetric techniques. Symmetrical ciphers use the same key for encryption and decryption. The strength of symmetrical algorithms really depends on how the key is exchanged securely between the sender and the receiver. Asymmetrical algorithms have two different keys, one of which is public and private. The private key is never transmitted on the network, which makes it safe. The public key is transmitted via the network to the addressee. During encryption, the sender encrypts the plain text using the recipient's public key and sends the resulting encrypted message to the recipient using the network. Even though the public key is known to the attackers, they cannot read the scrambled (or hashed) message because the secret key is unknown. During decryption, the recipient uses the private key to decipher the text. Asymmetrical algorithms are more complicated to implement and use than symmetrical algorithms. As a result, most modern IoT networks use symmetrical algorithms to provide security to transmitted information.

The symmetric cryptosystem is easy to implement, uses fewer resources with little overhead, and is safe while the key is kept secret. With many IoT devices with different computer capabilities, IoT networks 'efficient security mechanism must be light. Security mechanism turnaround time should be kept to a minimum to reduce delays and improve efficiency. Furthermore, the security mechanism must be less complex with minimum overhead costs.

Considered the constraints in the IoT environment. This paper proposes an enhanced tiny symmetric encryption technique. This is an improved or enhanced technique from the existing Tiny Encryption Algorithm. TEA is a lightweight encryption technique that can be easily implemented in an IoT environment [3]. The traditional TEA has suffered from many vulnerabilities. The TEA uses the same keys throughout all encryption

cycles, resulting in reduced security, apparent from the algorithm's avalanche effect. Executes the data for 64 rounds. A fixed amount of bit changes leads to data analysis by cryptanalysis.

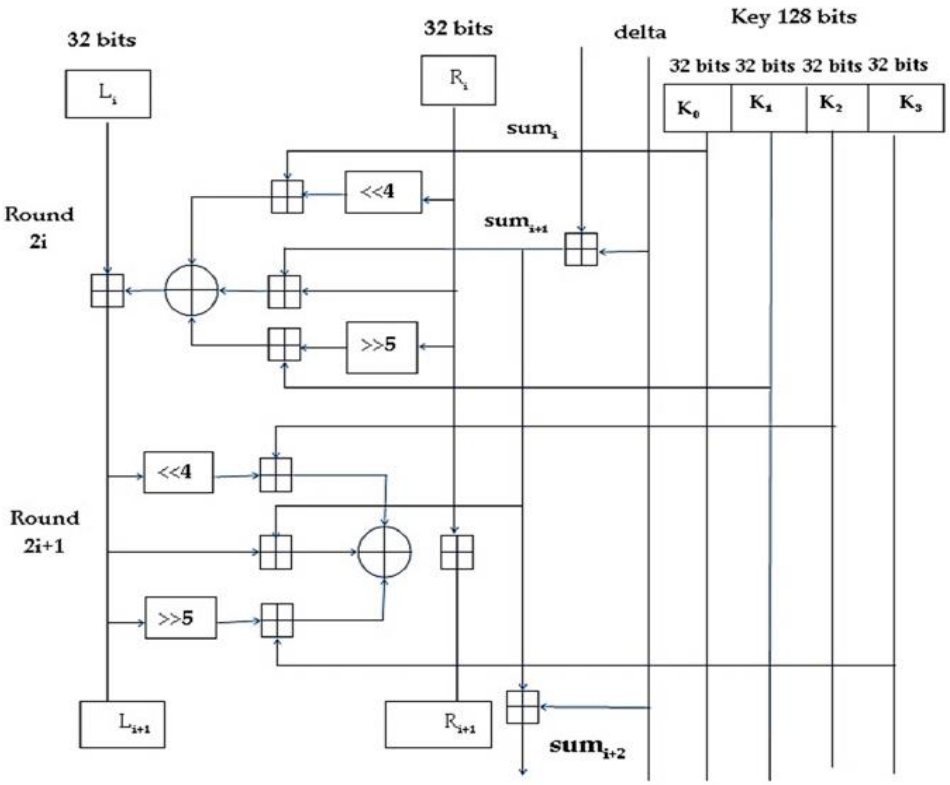

Fig. 1 Block diagram of TEA [4]

Additionally, text encryption and decryption time is high, which reduces the effectiveness of IoT networks with embedded devices. Over the years, many encryption algorithms have been offered to ensure data transmission via the IoT network. TEA is the most attractive, with its low memory usage and easy implementation of hardware and software ladders. However, one of the main problems of TEA and its many developed releases is the use of the same key. It uses the same keys for all encryption rounds, which reduces the obvious safety of the algorithm's avalanche effect. Figure 1 shows the block diagram of TEA.

The proposed work is an enhanced version of TEA. This work's main objective is to develop an Enhanced Tiny Symmetric Encryption Techniques (ETSET) to Secure Data Transfer Between IoT Devices. The data considered in this work is text and numerical based data.

2. Related Work

IoT is a growing technology in the field of information technology. This provides the user with much sophistication. The darkest part of the IoT has to do with security vulnerabilities. Data protection is more important in the IoT environment. This section outlines the existing data security techniques adapted to the IoT environment. Sreeja Rajesh et al. [4] proposed an algorithm called NTSA (Novel Tiny Symmetric encryption Algorithm), which enhances TEA's security functionality by introducing more key confusions. Most of the work with TEA and its variants concentrated solely on reducing the delay in delivery. Very few research has been done on modifying the key to improve the security of the transmission algorithm. This method dynamically introduces multiple key modifications and secures the key to the intruders. Since the key is dynamically computed, the key values are modified during the run time and cannot be pre-compiled. Besides, NTSA takes less time to encrypt and decrypt than TEA, thereby providing greater security and efficiency for all modern IoT networks. Experiments are conducted to analyze the avalanche effect, encryption and decryption time of the NTSA within an IoT network, including on-board devices. NTSA modified only in generating key is the following cycle. Among the four 32-bit keys, only two 32-bit keys are altered; the other two keys are identical for all cycles. It also uses a static bit switch and executes 64 revolutions.

Deepali Virmani et al. [5] proposed an enhanced tiny encryption algorithm with integration (ETEA), a data masking technique called steganography, as well as encryption (cryptography). ETEA has the advantage of integrating cryptography and steganography. The Enhanced Tiny Encryption algorithm is Feistel-like encryption that utilizes operations from mixed algebraic groups. A second change involves repeatedly mixing all data bits with the key. The proposed algorithm has the benefit of hiding messages. ETEA suffers from equivalent keys and can be broken using a key-linked attack requiring 223 selected applicants and 232-time complexity.

Several variants of the TEA algorithm have recently been proposed. Dian Rachmawati et al. proposed an algorithm [6] that uses combined asymmetrical and symmetric encryption for secure file transfer. File security is

provided by the symmetrical TEA algorithm and key security by the asymmetrical LUC algorithm based on the Lucas function. However, the method used the same key for all encryption towers, resulting in decreased security.

Novelan et al. proposed an SMS security system for IoT mobile devices utilizing the Tiny Encryption algorithm [7]. This system ensures that confidential messages are encrypted when there is a key to obtain the encrypted SMS message sent to the mobile destination number. The cipher on the receiving side can be decrypted with the same key to obtain the SMS. The SMS security method's downside is that the text message's size determines the encrypted and decrypted time taken for the process. A further variation of TEA is XTEA proposed in [8]. XTEA is a block-symmetrical cryptographic algorithm that uses the Feistel structure. This algorithm employs 64-bit clear blocks, 128-bit keys and 64 encryption towers. This algorithm uses a more complicated key program than TEA with rearrangements of changes, XOR and additions [9]. XXTEA, also referred to as the TEA block, uses the same round function as XTEA but applies it cyclically on an entire message for several iterations [10].

Anuj Kumar et al. [11] proposed a hybrid cryptography system to provide better security on the data stored on cloud storage through IoT devices. The proposed approach uses the RSA algorithm and DES algorithm and provides a hybrid of the two algorithms to provide more security on the data before storing it on the cloud. This approach may provide data security, but it is not an efficient method to integrate RSA and DES. Because RSA is an asymmetric cryptosystem, it takes more time to encrypt a small amount of data. DES also takes more time for encryption. Both algorithms are more complex; they can not be run on IoT devices. Hence, this hybrid approach is not suitable for IoT device data security.

Muhammad Usman et al. [12] came up with a light encryption algorithm called Secure IoT (SIT). It is 64-bit block encryption and requires a 64-bit key for data encryption. The architecture of the algorithm is a blend of Feistel and a uniform substitution switch network. The findings show that the proposed method offers substantial security in just five encryption towers. The proposed algorithm is limited to only five turns. To further improve energy efficiency, each encryption tower includes mathematical operations which run on 4 bits of data. To create sufficient confusion and data dissemination to deal with attacks, the algorithm uses the Feistel network of substitution diffusion functions.

Abdelhalim et al. [13] proposed a modified TEA algorithm (MTEA) that improves TEA security and energy consumption. A change to the standard TEA (MTEA) is proposed to increase the TEA's security against attacks. It uses a pseudorandom number generator (PRNG) such as the Linear Feedback Shift Register (LFSR). Here, the MTEA key is frequently changed in every turn rather than using a single key for all rounds in the standard TEA. TEA and MTEA are a network of Feistel structures that depend on the confusion and diffusion properties as substitution and permutation properties. The pseudorandom number generator frequently modifies the MTEA key in each round. Zhdanov et al. [14] suggested an algorithm based on multi-valued logic and variable block length principles. The encrypting process is carried out iteratively with five rounds. The number of rounds may vary, with the former consisting of gamma and permutation procedures; the remaining rounds include substitution and gamma procedures.

Various present research work related to the security of data in IoT devices are studied. There are still security is an open issue. The following section describes the proposed ETSET in detail.

3. Problem Definition

IoT is made up of millions of connected devices that can detect, calculate and communicate data. Each second, a great deal of data is transferred between these devices. Given IoT network applications 'sensitivity, such as connected vehicles or mobile health devices, transmitted information security has remained a major issue. The increase in intrusions and hackers has made this task very difficult. Maintaining the security of data transfer between devices is an important concern in the IoT environment. Mitigate data attacks by providing improved encryption techniques. IoT devices are small in size, and it has low computation energy. The design of the security technique is based on the environment of the data. Complex computations are not able to run by IoT devices. Hence, it is necessary to concentrate on lightweight security approaches to improve data security in the IoT environment.

4. Methodology

The proposed ETSET is a symmetric nature of the cryptosystem. The ETSET is a 64bit block cipher. It uses the size of the primary key is 128 bits. This key is divided into four sub-keys of 32 bits. The total number of rounds is 16 and consists of 8 cycles. Each cycle consists of 2 rounds. The 64bit plaintext is divided into two 32bits. The two halves use other 32bits in every round. For each round, two 32 bits keys are used. Every cycle, the key is dynamically changed. Introduce extra key confusion dynamically for each encryption cycle. As a result, provides both greater security and efficiency for all modern applications in IoT networks.

5. ETSET System Operation

The proposed ETSET follows the TEA process but has some modifications in the round function and key generation. The ESET has eight cycles and 16 rounds. The key is generated along with the encryption to prevent any unwanted delay in the encryption time. Dynamic bit movement is performed at each round. This creates confusion in the outcome of the encrypted information. If all rounds are complete, both 32 bits are exchanged and generate 64-bit encrypted data. Dynamical sub-keys are generated for each round. Thirty-two sub-keys are generated, and four sub-keys are used by each cycle in the encryption process. The ETSET block diagram is shown in the figure. The round function is as follows:

Encryption Round i (i is odd):

L0⊕= ((R⊕LSHIFTCNT) k1) XOR (R0 RSHIFTCNT) ⊕k2)

Round i (i is even):

R0⊕= ((L0 LSHIFTCNT) ⊕ k3) XOR (L0 RSHIFTCNT) ⊕k4)

Key is generated for each cycle as follows, Main Key2 = Merge(64 bits swap((KL0⊕Merge(L1,R1)), (KR⊕Merge(L1,R1))))

Figure 2 depicts the block diagram of the proposed ETSET. The procedure of the round function and key generation is as follows.

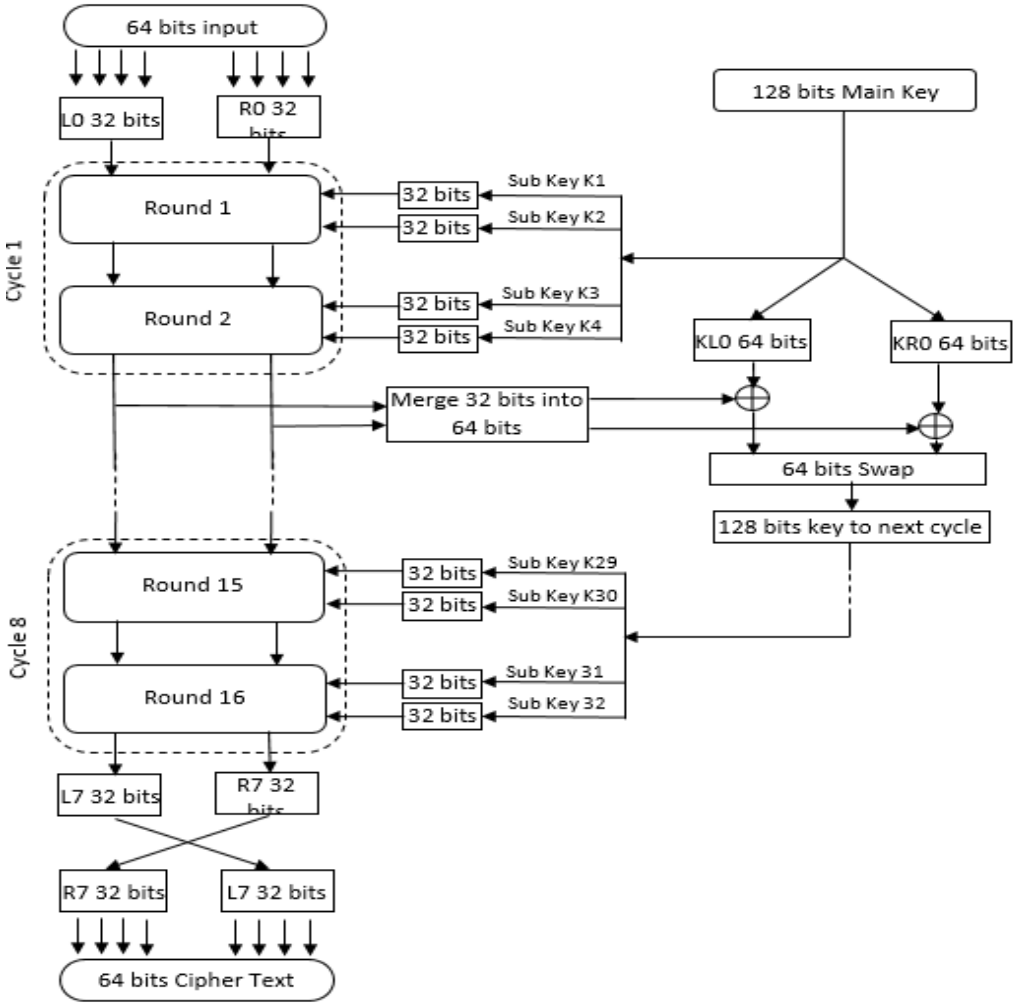

Fig.2 Block diagram ETSET

Round Function

The following steps involved in the Proposed ETSET procedures,

Step 1: Considered the Plaintext as 64 bits.
Step 2: Split the 64 bits into two 32 bits(L0_32 bits and R0_32 bits).
Step 3: Round 1 is started. Count number of 1 's in R0_32 bits.
Step 4: Shift left the 32 bits at several time the count value.
Step 5: Find XoR with Sub Key k1
Step 6: Count the number of 0 's in R0_32 bits.
Step 7: Shift right the 32 bits at the number of time the count value.
Step 8: Find XoR with Sub Key k2.
Step 9: Find XoR with the result of step 5 and step 8 and get 32 bits.
Step 10: Find XoR of this 32bits with L0_32bits. L0_32bits is taken as input to round 2
Step 11: Round 1 Completed
Step 12: Round 2 is started. Count number of 1 's in L0_32 bits.
Step 13: Shift left the 32 bits at the number of time the count value.
Step 14: Find XoR with Sub Key k3
Step 15: Count the number of 0 's in L0_32 bits.
Step 16: Shift right the 32 bits at the number of time the count value.

Step 17: Find XoR with Sub Key k4.
Step 18: Find XoR with the result of step 14 and step 17 and get 32 bits.
Step 19: Find XoR of these 32bits with R0_32bits. R0_32bits is taken as input to round 3
Step 20: Round 2 Completed.
Step 21: Repeat the steps from 3 to 19 to complete all the rounds.
Step 22: After completion of 16 rounds, a 32 bits swap is done
Step 23: Merge two 32 bits and get the 64 bits ciphertext.

Key generation

The ETSET encrypt the data based on the procedure described in the previous section. The key for the encryption is generated using random number generation. The key generated during each cycle of encryption. The main key size is 128 bits. This 128 bit is processed to generate 32 number 32 bits keys. Group of four 32 bits keys are used in each cycle. There are 8 cycles, and in each cycle, two encryption rounds are carryout. Totally 16 rounds, each round uses pair of two keys. The procedure of key generation is as follows.

Step 1: The main key is 128 bits. It is given as a key to the first cycle of encryption.
Step 2: The 128 bits key is divided into four 32 bits, K1, K2, K3 and K4.
Step 3: The first round of cycle 1 uses the Key K1 and K2 and K3 and K4 are used by the second round of cycle 1.
Step 4: The main key 128 bits is divided into two 64 bits keys.
Step 5: Both 64 bits are XORed with the result of 64 bits from the cycle.
Step 6: Both 64 bits keys are swapped.
Step 7: Merge the two 64 bits into single 128 bits.
Step 8: The derived 128 bits key is given to the input key for the next cycle.
Step 9: Steps from 4 to 8 is repeated until all cycles are completed.

6. Results and Discussion

The work is implemented using JAVA. The message sends from one device to another are encrypted based on the proposed techniques. The comparison encryption time and decryption time is given in the below tables. The experiment is conducted for different size of the data. Table 1 shows the encryption time taken for the proposed and existing encryption technique. The table shows the time taken for processing in milliseconds. ETSET has taken 1 milliseconds for encrypting 10 KB of data transmitted between IoT devices from other technique, but TEA and XTEA taken 1.94 MS and 2.74 MS, respectively.

I. COMPARISON OF ENCRYPTION TIME

Data Size (KB)	**TEA**	**XTEA**	**ETSET**
	Milliseconds (MS)		
10	1.94	2.74	1.00
20	3.56	5.12	1.98
30	5.34	7.32	2.88
40	7.16	9.99	3.64
50	8.98	12.49	4.43
100	17.23	24.35	8.65

Table 2 shows the comparison of decryption time taken for the proposed and existing decryption technique. For decrypting the data, ETSET has taken 0.99 MS, and other techniques TEA and XTEA has taken 1.92 and 2.71, respectively. The experiment results show that the proposed ETSET produces better results compared to the existing technique.

II. COMPARISON OF DECRYPTION TIME

Data Size (KB)	**TEA**	**XTEA**	**ETSET**
	Milliseconds (MS)		
10	1.92	2.71	0.99
20	3.53	5.09	1.94
30	5.31	7.28	2.83
40	7.12	9.95	3.59
50	8.93	12.44	4.38
100	17.19	24.31	8.59

7. Conclusion

IoT environment is a fast-growing Technology in today world. Security is the most important concern to address in the IoT environment. The symmetric cryptosystem is more suitable for the IoT environment. A symmetric Encryption technique is proposed to secure the data transmitted between IoT devices. It is lightweight encryption because it does not use any permutation and substitution tables. It uses dynamic shifting and key generation to strengthen the security of the data in the IoT environment. The changes in ETSET from TEA is, keys for each cycle is changed entirely. The number of cycles and rounds is reduced to 8 and 16 numbers, respectively. The right and left shift of bits is done dynamically. The work is implemented and compared with existing techniques. The results showed that the ETSET produced better performance than other techniques.

References

1. Nickson M. Karie, Nor Masri Sahri and Paul Haskell-Dowland, "IoT Threat Detection Advances, Challenges and Future Directions", 2020 IEEE Workshop on Emerging Technologies for Security in IoT (ETSecIoT), DOI 10.1109/ETSecIoT50046.2020.00009, pp. 22-29.
2. Ali Hameed and Alauddin alomary, "Security Issues in IoT: A Survey", 2019 IEEE International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 2019, pp. 1-5.
3. Wheeler, D.; Needham, R. TEA, a tiny encryption algorithm. In Proceedings of the 1995 Fast Software En-CryptionWorkshop, Leuven, Belgium, 14–16 December 1995; Springer: Berlin/Heidelberg, Germany, 1995; pp. 97–110.
4. Sreeja Rajesh, Varghese Paul, Varun G. Menon, and Mohammad R. Khosrav, A Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices, Symmetry 2019, 11, 293, pp, 1-21.
5. Dr. Deepali Virmani, Nidhi Beniwal, Gargi Mandal, Saloni Talwar, Enhanced Tiny Encryption Algorithm with Embedding (ETEA), INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY, vol 7, issue 1, June 2013, pp.1-9.
6. Rachmawati, D.; Sharif, A.; Jaysilen; Budiman, M.A. Hybrid Cryptosystem Using Tiny Encryption Algorithm and LUC Algorithm. IOP Conf. Ser. Mater. Sci. Eng. 2018, 300, 012042. [CrossRef.
7. Novelan, M.S.; Husein, A.M.; Harahap, M.; Aisyah, S. SMS Security System on Mobile Devices Using Tiny Encryption Algorithm. IOP Conf. Ser. J. Phys. Conf. Ser. 2018, 1007, 012037. [CrossRef]
8. Needham, R.M.; Wheeler, D.J. TEA Extensions; Technical Report; Computer Laboratory, University of Cambridge: Cambridge, MA, USA, 1997.
9. Kaps, J.-P. Chai-tea, Cryptographic Hardware Implementations of XTEA. In Proceedings of the INDOCRYPT 08 Proceedings of the 9th International Conference on Cryptology in India: Progress in Cryptology, Kharagpur, India, 14–17 December 2008.
10. Wheeler, D.; Needham, R. XXTEA: Correction to XTEA; Technical report; Computer Laboratory, University of Cambridge: Cambridge, MA, USA, 1998.
11. Anuj Kumar, Vinod Jain and Anupam Yadav, "A New Approach for Security in Cloud Data Storage for IoT Applications Using Hybrid Cryptography Technique", IEEE, 2020, pp.514-517.
12. Muhammad Usman, Irfan Ahmedy, M. Imran Aslamy, Shujaat Khan and Usman Ali Shah, SIT: A Lightweight Encryption Algorithm for Secure Internet of Things, International Journal of Advanced Computer Science and Applications, Vol. 8, No. 1, 2017, pp.1-10.
13. Abdelhalim, M.B.; El-Mahallawy, M.; Elhennawy, M.A.A. Design and Implementation of an Encryption Algorithm for use in RFID System. Int. J. RFID Secure. Cryptogr. 2013, 2, pp. 51-57.
14. Zhdanov, O.N.; Sokolov, A.V. Block Symmetric Cryptographic Algorithm based on Principles of variable block length and many-valued logic. Far East J. Electron. Commun. 2016, 16, pp. 573–589.