International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 3, June, 2024, pp. 1075-1083
Journal homepage: http://iieta.org/journals/isi

Block of Data Encryption Using the Modified XTEA Algorithm

Ahmed Abd Ali Abdulkadhim*, Ali Shakir Mahmood, Mohanad Ridha Ghanim

College of Education, Computer Science Department, Mustansiriyah University, Baghdad 964, Iraq

Corresponding Author Email: ahmed_198@uomustansiriyah.edu.iq

https://doi.org/10.18280/isi.290325

Received: 6 November 2023
Revised: 7 March 2024
Accepted: 25 March 2024
Available online: 20 June 2024

Keywords:
data encryption, XTEA algorithm, rossler system

ABSTRACT

The Internet of Things (IoT) employs various devices with varying hardware capabilities, including those with restricted resources like wireless sensor networks and those with ample resources like satellites. One of the primary hurdles is developing a streamlined encryption algorithm suitable for IoT devices with limited hardware capabilities. This paper introduces an enhanced lightweight algorithm that not only addresses side-channel vulnerabilities but also guards against nonce misuse attacks In this work, we present a design that generates encryption keys using chaotic systems, thereby increasing their unpredictability and randomness. The primary objective of this research is to fortify security measures against a range of novel attack techniques, guaranteeing comprehensive defense, unpredictability, and resilience. The aim of putting strategic defenses and tactics into place is to shield valuable assets from possible threats.

1. INTRODUCTION

The Internet of Things (IoT) is often described as a system of devices that connect with cloud nodes to share data. Unlike computers these devices have high capacities when it comes to RAM, memory and processing power [1-3].

Many IoT devices have security vulnerabilities that make them vulnerable to cyber attacks and unauthorized access. This expose data security and personal privacy. The absence of communication protocols among devices leads to compatibility issues posing a major obstacle. This slow down their ability to interact and exchange information across platforms and devices. Additionally several IoT devices are reserved by power, storage capacity and energy resources impacting directly on their performance, in handling tasks and processing substantial data volumes. Their overall functionalities and capabilities are typically restricted by these limitations [4, 5].

These restrictions are mostly motivated by security concerns. Because many encryption techniques are complex, implementing encryption on Internet of Things devices can be difficult. Nonetheless, a number of encryption algorithms are available that are especially made for these limited settings. These algorithms are renowned for their straightforward design, rapid speed, and effective memory management [6].

Encryption algorithms are completely immune to attacks, even algorithms like XTEA have vulnerabilities that can be exploited under certain conditions. It's important to note that XTEA is not considered a highly secure encryption algorithm by today's standards due to its vulnerability to certain attacks because of these vulnerabilities and other weaknesses, called the need to make a combination with chaotic systems. Chaotic signals exhibit several features that align with the prerequisites of cryptography, including their pseudo-randomness, extreme sensitivity to initial conditions and system parameters, and high degree of nonlinearity-all essential qualities in the field of cryptography.

Consequently, chaotic systems have gained extensive utilization in the construction of cryptographic systems in recent times. In this paper, we propose an architecture based on the chaotic system of generating keys and obtaining unpredictable randomness. The remaining structure of this paper is as follows: An overview of Related Works is presented in Section 2. Section 3 describes Rossler System. Section 4 comprises XTEA algorithm. Section 5 contains Modified XTEA Algorithm. Evaluation parameters are discussed in Section 6. Conclusions are presented in Section 7.

2. RELATED WORK

Many works in the literature focus on improving performance of XTEA algorithm for better productivity. We now present some recent, closely related work.

Kotel et al. [7] offered a block cypher algorithm that is lightweight. This update modifies the key obstetrics function present in the XTEA1 algorithm by use of a linear feedback shift register. The objective evaluation of lightweight cryptographic systems metric, which computes metrics like RAM usage, ROM usage, and execution time to measure performance across three commonly used embedded devices: 32-bit ARM, 8-bit MSP, and 32-bit AVR microcontrollers, was used to perform a code analysis of this XTEA variant using the same benchmarks. The implementation results are shown that this modified version imposes fewer software demands in comparison to the original XTEA. Furthermore, the Software performance and security have been notably improved.

Acholli and Ningappa [8] enhance the security and privacy of information. It focuses on communication systems. The researchers focus on encryption and decryption techniques. They stress the importance of key generation for encryption security. Keys are vulnerable to modern computers. So, a hybrid crypto algorithm is proposed. It combines XTEA with IDEA. The researchers use a method called ID-XT-EA-LFSR. It is based on the International Data - Extended Tiny Encryption Algorithm. They use it to boost the security of real-time apps, especially for online streaming data. They recognize that a large key size adds randomness. But it also makes the network busier and more complex. To fix this, generate random numbers using an LFSR scheme for the key. It improves FPGA performance by up to 50.43% compared to other algorithms. These include QTL, DROM-CSLA-QTL, and XTEA. The research highlights the usefulness of the proposed hybrid cryptographic algorithm. It improves security for real-time applications.

Anusha et al. [9] introduce a pipelined architecture that utilizes parallel computation to enhance throughput while simultaneously bolstering security measures. Our proposed XTEA cipher is designed with configurability, allowing for the adjustment of its mode to either encryption or decryption as needed. Simulations for TEA and XTEA on an FPGA platform, specifically Artix-7, taking into account resource limitations such as area, time, and power, and have presented the results in tabular form.

Mishra and Acharya [10] introduced a hybrid model featuring a pipelined architecture that emulates all three lightweight ciphers: TEA, XTEA, and XXTEA, within a single framework tailored for high-speed IoT and RFID applications. The pipelined implementation of these ciphers was undertaken to enhance the computational throughput of the encryption This was accomplished by parallelizing the computation, which led to a significant increase in throughput at the expense of increased resource consumption. Additionally, they combined the TEA architecture to incorporate a hybrid approach. These specially created encoders were used with both FPGA and ASIC platforms.

A simplified sequential design for the XXTEA-192 block cipher specifically intended for use in RFID applications was proposed by Kella et al. [11]. RFID tags usually have very strict limitations on the amount of power and space they can use. Among the TEA family of ciphers, XXTEA is the strongest member. It is notable for its ability to handle messages of varying lengths; messages must be at least 62 bits long and multiples of 32-bit word sizes. This special characteristic enables the cipher to encrypt larger data blocks more effectively. Every clock cycle, the XXTEA-192's proposed serial architecture performs 32-bit operations like addition, XOR, and shifting.

Nagarajan et al. [12] introduced an upgraded architecture known as RXTEA (Renovated XTEA), along with a secure mutual authentication protocol termed RXMAP (RXTEA-based Mutual Authentication Protocol). In the revised XTEA design, various modifications have been implemented to enhance computational efficiency and bolster resistance against related key attacks and meet-in-the-middle attacks. A novel delta value is computed for each successful authentication using random numbers, increasing security. Additionally, only half of the ciphered text is employed for server and tag authentication, rendering key value prediction nearly impossible. Furthermore, key values are dynamically updated at both the server and tag sides through internal computations, enhancing security by making the prediction of the new key highly improbable.

Table 1. Summary of previous work

Year	Authors	Methodology Key	Features	Measurements and Results
2017	[7]	A Linear Feedback Shift Register is used to replace the key generation function in the XTEA1 Algorithm.	Using the same evaluation conditions, analyzed the software implementation of the modified XTEA using FELICS (Fair Evaluation of Lightweight Cryptographic Systems) a benchmarking framework which calculates RAM footprint, ROM occupation and execution time on three largely used embedded devices: 8-bit AVR microcontroller, 16-bit MSP microcontroller and 32-bit ARM microcontroller.	Implementation results show that it provides less software requirements compared to original XTEA. Enhanced the security level and the software performance.
2019	[8]	The research's main method is to enhance the security and privacy of information. It focuses on communication systems. The researchers focus on encryption and decryption techniques. They stress the importance of key generation for encryption security. Keys are vulnerable to modern computers. So, a hybrid crypto algorithm is proposed. It combines XTEA with IDEA.	The researchers use a method called ID-XT-EA-LFSR. It is based on the International Data - Extended Tiny Encryption Algorithm. They use it to boost the security of real-time apps, especially for online streaming data. They recognize that a large key size adds randomness. But it also makes the network busier and more complex. To fix this, generate random numbers using an LFSR scheme for the key.	It improves FPGA performance by up to 50.43% compared to other algorithms. These include QTL, DROM-CSLA-QTL, and XTEA. The research highlights the usefulness of the proposed hybrid cryptographic algorithm. It improves security for real-time applications.
2019	[9]	This article introduces a Lightweight Block cipher method. The hardware layout of Tiny Encryption Algorithm (TEA) has been	The suggested XTEA is reconfigurable, allowing it to conduct encryption or decryption	Similar previous research efforts, such as AES-8bit, TinyXTEA1, and Tiny XTEA-3, with improvements in area,

		developed to be straightforward, adaptable requiring computations and with a key scheduling process. To address security vulnerabilities, in the scheduling of TEA, an Enhanced TEA (XTEA) has been created with a structure that enables parallel processing to enhance efficiency and enhance security measures.	by switching modes. The Xilinx ISE tool on the Model Sim 6.5f simulator yielded the TEA and XTEA simulation results, which were then implemented on the FPGA Platform-Artix-7 with tabulated resource limitations, including area, time, and power.	throughput, and efficiency on the same FPGA platform, are compared with the suggested XTEA. With a high throughput of 81 Mbps and an efficiency of 0.34 Mbps/slice, the suggested XTEA operates.
2021	[10]	TEA, XTEA and XXTEA encryption methods were utilized to achieve the specified goal. They were developed, put into practice and enhanced using hardware technologies such, as platforms for Application Specific Integrated Circuits (ASIC) and Field Programmable Gate Arrays (FPGA). Block sizes, implementation rounds, and important scheduling techniques were among the attributes that the designs were put into practice to evaluate.	Utilizing T1, T2 and T3 through a sequential approach to boost performance measures like occurrence and size. Introducing a blended model (T4) that merges TEA family codes into one blueprint employing a sequential structure.	This research paper introduces four different hardware architectures named TEA (T1), XTEA (T2), XXTEA (T3) and a hybrid model (T4). T1, T2 and T3 were developed using a pipelined approach, resulting in significant frequency improvements of 75.9%, 162% and 89% respectively. Additionally, there were notable area enhancements of 85.43%, 57.08% and 90.79% for each architecture. The hybrid model T4 combines the TEA family designs into a single pipelined architecture, showing a gate equivalent improvement of 47.50% for T2 specifically. Despite having similar throughput to T1, T2 and T3, the hybrid model (T4) requires fewer gates compared to the combined total of the three individual architectures. The efficiency enhancement achieved by these new designs surpasses that of existing literature by more than eighteen times.
2021	[11]	This paper introduces a design for the XXTEA 192 block cipher suitable for RFID applications. RFID tags face limitations in terms of space and power. XXTEA is considered the cipher, within the TEA family. It operates on messages of varying lengths with a message size of 62 bits, which must be multiples of 32-bit words.	This feature enables the cipher to encrypt larger block sizes. Serial architecture of XXTEA-192 has been proposed that performs 32-bit operations like addition, XOR and shifting in each clock cycle.	The suggested plan has been put into action using verilog. Evaluation of performance measures such as size, speed, effectiveness, power, and energy has been conducted on types of FPGA devices like Spartan 6 Virtex 6 and Virtex 7. The proposed layout has attained 45 slices and a data transfer rate of 498.44 Mbps on Spartan 6 making the design compact and fast, for RFID uses.
2021	[15]	This article presents an updated and more secure iteration of the XXTEA algorithm by incorporating an S-box to bolster security against various attacks. Additionally, the M XXTEA algorithm is combined with a generation system providing an additional layer of security that aligns with the concept of a onetime pad. The encryption keys change for each block of plaintext during the encryption process.	As a result, it offers a stronger security system than both the AES and the original XXTEA. In addition to improving data security, the M-XXTEA can work with various text block and key widths. To compare the M-XXTEA's performance with that of the AES and the original XXTEA, several experiments are conducted.	The encryption and decryption time efficiencies of M-XXTEA surpass AES by 60%, according to the results. Additionally, the M-XXTEA outperforms the AES in speed efficiency by 57%. It can be effectively applied to safeguard e-health systems, smart cities, and industrial IoT smart device security.
2022	[12]	The research's main method is to address security challenges in wireless sensor networks. It focuses on Radio Frequency Identification (RFID) and green solutions. Chose the Extended Tiny Encryption Algorithm (XTEA) as a lightweight encryption scheme. But fixed its vulnerabilities with domain-specific customization. Employ random numbers and secret key renewal.	The researchers propose two Renovated XTEA Mutual Authentication Protocol (RXMAP) encoder architectures. They are RXMAP-1 and RXMAP-2. They replace precise blocks with rough ones to boost security. The evaluation process includes assessing two things: computational and storage overhead. It also includes verifying against various security threats. Verification uses BAN logic formal and informal methods.	For functional verification, Simulate the proposed encoders, also make an Application-Specific Integrated Circuit (ASIC). It has a 132 nm process node. The results show that RXMAP-1 and RXMAP-2 designs take up much less space. They take up 53.11% to 53.31% less space than XTEA I and XTEA II implementations. Also, RXMAP-1 and RXMAP-2 use much less power (68.76% to 71.64% less) than XTEA II. But they have the same throughput.

Shailaja [13] proposed a hybrid encryption approach that combines the Extended Encryption Algorithm (XTEA) and the IDEA algorithm to increase security in real-time applications. While a larger key size ensures randomness, it also increases network load and complexity proportionally. To tackle this issue, generate keys by generating random numbers using the Linear Feedback Shift Register system. This algorithm proved suitable for encrypting and decrypting online streaming data, leading to the method's designation as the International Data-XTEA-LFSR (ID-XT-EA-LFSR). Notably, the ID-XT-EA-LFSR method significantly improved FPGA performance, demonstrating a 50.43% enhancement compared to QTL, DROM-CSLA-QTL, and XTEA algorithms.

Ragab et al. [14] to improved its defences against different types of attacks, added an improved S-box to the original XXTEA cypher, making it a more robust version. Furthermore, a chaotic key generating system is integrated with the M-XXTEA, adding an extra degree of security consistent with the idea of a one-time pad. In contrast to the original XXTEA, the cypher keys are dynamically changed for every plaintext block during the encryption process, resulting in a more adaptable security architecture. M-XXTEA is flexible enough to work with a variety of text block sizes and key sizes, enhancing data security.

Researchers stated in Table 1 describe their previous works were studied ways to improve XTEA's productivity. They have showcased recent progress. Nagarajan and colleagues introduced RXTEA, a new XTEA structure. They also introduced RXMAP, a secure mutual authentication protocol. It offers better speed and more resilience against attacks. Shailaja proposed ID XT EA LFSR. It is a hybrid encryption technique that combines XTEA and IDEA. It shows better FPGA performance for real time apps. Mishra and Acharya presented a hybrid model. It uses pipelined architecture for TEA, XTEA, and XXTEA. This speeds up computation in high speed IoT and RFID applications. Kella et al. I devised a sequential XXTEA 192 block cipher for RFID systems. It handles variable length messages well and boosts security. Kotel et al. made a lightweight version of XTEA1. It had reduced software needs and better performance on embedded devices. Anusha and her team introduced a configurable pipelined XTEA cipher. It has better throughput and security due to parallel computation techniques. Ahmed Ab M. Ragaband et al. Strengthened XXTEA. That was done by adding an improved S box and by adding dynamic key changes. These changes provide adaptable security against diverse attacks. They make M XXTEA a versatile solution for various data security needs.

3. ROSSLER SYSTEM

The Rossler system was discovered by Otto Rossler when he was conducting research in the field of chemical kinetics [16, 17]. Mathematically, the Rossler system is characterized by three interconnected nonlinear differential equations, which are explained by the following Eq. (1):

$$dx/dt = -y - z,\ dy/dt = x + a \times y \\ dz/dt = b + z \times (x - c) \tag{1}$$

where, *a*=0.2, *b*= 0.2, *c*=5.7.

By studying the Rossler system, it is often used as a simple example to illustrate chaotic behavior and explore the properties of chaotic systems and the system's butterfly-shaped attractor, using techniques such as numerical simulations. The bifurcation diagrams and Lyapunov exponents can analyze its dynamics, shown in Figure 1.

The Rossler system serves as a foundational model for understanding the concepts of chaos, attractors, and complex behavior in dynamical systems. Because of its visually appealing patterns, it has uses in physics, engineering, biology, and even the arts and aesthetics [18].

The chaotic character of the Rössler systemits and great sensitivity to initial conditions, which offers a great degree of complexity and randomness, is one of the reasons it was chosen for encryption. Because of this, even if an attacker knows the equations and parameters of the system, it will be challenging for them to forecast the key stream. Furthermore, the Rössler system is computationally efficient for encryption applications because to its comparatively basic structure when compared to other chaotic systems.

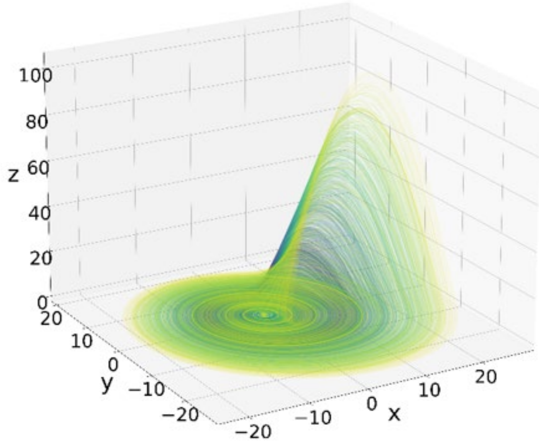

Figure 1. Rossler system bifurcation diagrams when *a*=0.2, *b*=0.2, *c*=5.7

The Rössler system is superior to other encryption techniques in several ways. Firstly, it offers a high level of security due to its chaotic behavior and resilience to attacks. Second, because of its simplicity of implementation and high computational efficiency, it is suitable for real-time encryption applications. In the end, a range of important stream sequences can be generated by merely altering the Rössler system's parameters. which makes it a strong option for encryption algorithms that require a high level of unpredictability and security [17, 19].

4. EXTENDED TINY ENCRYPTION ALGORITHM (XTEA)

The cryptographic algorithm is simple, lightweight, and has a tiny code size. It is therefore suitable for small applications. The block size of XTEA is 64 bits, with a key size of 128 bits. It also made use of the Feistel network for N rounds. Where N is usually 32. XTEA uses the following operations (Exclusive OR, Additions & Shifts). With a 128-bit key size, 64 rounds of operations, and 64 bits of block data, the XTEA is the most potent and quickest light-weight cryptographic block cipher (LCBC) with a basic Feistel structure. Most real-time cryptography applications make advantage of it [20].

For microcontrollers with limited memory, the clear key schedule and dynamic scheduling of the rounds' keys optimize memory consumption [21].

XTEA it was described as a valuable and advanced alternative for enhancing security by the same scholars who developed it. Although XTEA is widely regarded as one of the most significant lightweight algorithms, it suffers from a low-round security vulnerability, and high-security applications should be able to handle 32 rounds [22].

As depicted in Figure 2, The XTEA encrypts data by dividing it into two halves, Li and Ri, which are subsequently applied to a regular function for N rounds (usually 32). First, the Ri half is shifted four times, then five times. The outcomes of these two processes are XORed together. Subsequently, the Li portion is added to the result. All that is needed for encryption in each round are additions, shifts, and exclusive-or operations (modulo 2^{32}).

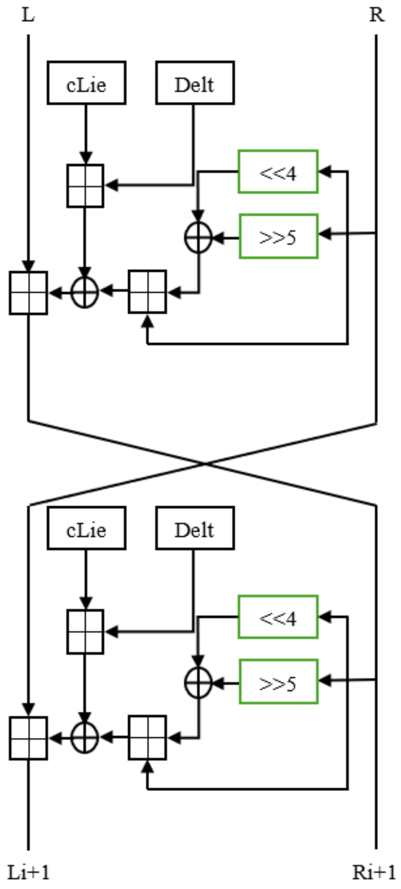

Figure 2. XTEA algorithm block diagram

The permutation functions are written as below.

$$f(x) = (x << 4) \oplus (x >> 5) + x \tag{2}$$

The operation of XTEA's key scheduling is very simple function: the 128-bit master key K is divided into four 32-bit sub key blocks (K [0], K [1], K [2], and K [3]), and the subkey generation functions are expressed as

$$\begin{gathered} Kr = K(r - 1/2.\delta \gg 11)\&3, if\ r\ is\ odd \\ Kr = K(r/2.\delta \gg 11)\&3, if\ r\ is\ even \end{gathered} \tag{3}$$

In above equation:

- kr round key, r round
- ≪ represents the bitwise left shift operation.
- ≫ represents the bitwise right shift operation.
- & represents the bitwise AND operation.
- δ and K assumed to be constants specific to your context.

In each round, sum is the incrementation value of the constant number DELTA=0x9e3779b9. DELTA is derived This constant number is used to guarantee that the sub keys are different.

$$DELTA = (\sqrt{5} - 1) * 2^{31} = 9E3779B9_h$$
$$DELTA[i] = (i + 1) * \ DELTA, i = 0,1,2 \dots 31$$

5. MODIFIED XTEA ALGORITHM

In this section, the modifications are discussed that are undertaken to elevate the algorithm's security against code analysis. One of the most notable drawbacks of the XTEA algorithm is the fixed key size of 128 bits. Although this provides a reasonable level of security, it may be considered limited compared to algorithms such as AES, which support a range of key sizes (128, 192, and 256 bits). There is also no built-in mechanism to extend or rotate the key. This means that if the key is compromised, all data encrypted with that key is potentially compromised. Related key attacks: XTEA is vulnerable to related key attacks, where an attacker can exploit the relationship between keys to decrypt data. Cryptanalysis: Although XTEA is considered secure against many types of attacks it is not immune to cryptanalysis. This vulnerability can be mitigated by ensuring that keys are generated using a secure random number generator such as a chaotic system.

Therefore, we propose to use the Rossler system (Mentioned in Section 3) to generate keys, where the parameters of the system a, b, and c are initialized in the first stage to generate 64 random keys with a size of 32 bits. Where each round has two different keys. The keys generated for each round are different from the next, giving the system a high level of security. As shown in the Eq. (4).

$$\begin{gathered} dx/dt = -y - z,\ dy/dt = x + a \times y \\ dz/dt = b + z \times (x - c) \\ K_{xi} = x_i,\ K_{yi} = y_i \end{gathered} \tag{4}$$

The stage of modifying the algorithm architecture. The algorithm is expanded to operate on a 128-bit block instead of a 64-bit block, using 64 different random keys generated from the chaotic system (each key is 32 bits), where each round has two different keys. Each input block is then divided into four partitions (L_{1n},R_{1n} ,$L_{2n}$and R_{2n}), which are then applied to a routine like the Feistel network for N rounds, where N is usually 32. The partitions are swapped between them, as explained in Figure 3. The proposed architecture in the process of exchanging four partitions (32 bits) created from a 128-bit block increases the strength of the encryption and reduces the time taken for encryption and decryption by half.

L_{1n+1}=R_{2n} , $L_{2n+1} = R_{1n}$ and R_{1n+1} ,R_{2n+1} is computed using following:

$$\begin{aligned} R_{1n+1} &= L_{2n} \boxplus (((R_{2n} \ll 4 \oplus R_{2n} \gg 5) \boxplus R_{2n}) \oplus (det \\ &\qquad \boxplus K_{xi})) \\ R_{2n+1} &= L_{1n} \boxplus (((R_{1n} \ll 4 \oplus R_{1n} \gg 5) \boxplus R_{1n}) \oplus (det \\ &\qquad \boxplus K_{yi})) \end{aligned} \tag{5}$$

The steps of the proposed XTEA algorithm are explained in the following.

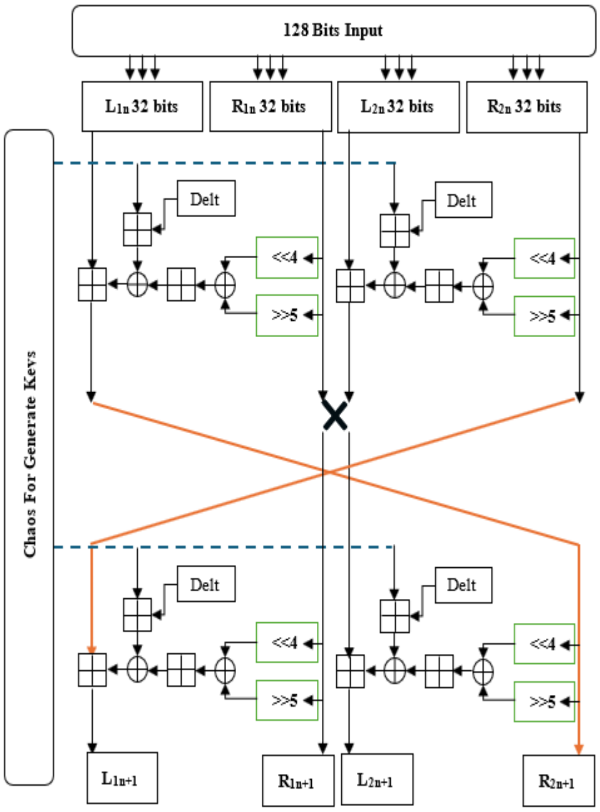

Figure 3. Round of modified XTEA algorithm

Modified algorithm

```
Step 1: Initialize parameters.
        a = 0.2, b = 0.2, and c = 5.7          // Parameters
Step 2: Initialize variables x, y, z// Initial.
Step 3: Set nr = 64                            // Number of time steps.
Step 4: Loop for i from 1 to nr     // Loops
        dx/dt = - (y + z)           // Equation to generate chaos
        dy/dt = x + a * y
        dz/dt = b + z * (x - c)
        End
Step 5: Save x, y in arrays $K_{xi}$ , $K_{yi}$ ,       // Save the output as a key.
Step 6: Split the plaintext ($L_{1n}$,$R_{1n}$ ,$L_{2n}$,$R_{2n}$)
Step 7: For i from 1 to 23 do      // Number of rounds
Step 8:   Set $L_{1n+1}$=$R_{2n}$
Step 9:   Set $L_{2n+1} = R_{1n}$
Step 10:   Set $R_{1n+1} = L_{2n} \boxplus (((R_{2n} \ll 4 \oplus R_{2n} \gg 5) \boxplus R_{2n}) \oplus (det \boxplus K_{xi}))$
Step 11:   Set $R_{2n+1} = L_{1n} \boxplus (((R_{1n} \ll 4 \oplus R_{1n} \gg 5) \boxplus R_{1n}) \oplus (det \boxplus K_{yi}))$
Step 12: Cipher text = $L_{1n+1}$,$R_{1n+1}$ ,$L_{2n+1}$,$R_{2n+1}$
Step 13: End
```

6. RESULTS AND DISCUSSIONS

The outcomes are derived from the operationalization of the prescribed methodology, facilitated via the execution of two distinct evaluations: the "Avalanche Effect" examination and the analysis of key sensitivity. Concurrently, due consideration is accorded to the temporal aspects of encryption and decryption processes. A judiciously efficacious encryption algorithm is requisite to manifest resilience against a gamut of established incursions. To this end, the magnitude of the key space should surpass a threshold adequate for forestalling brute-force attacks, and furthermore, the algorithm should evince discernible reactivity to the confidential key substrate. The criteria for evaluating the security posture of block encryption algorithms encompass pivotal metrics, among which the Avalanche Effect and key sensitivity analysis are salient. These assessments were meticulously conducted to empirically validate the commendable security efficacy inherent in our proposed XTEA algorithm, compared against the benchmark performance of the conventional XTEA instantiation.

6.1 Avalanche effect test evaluation

The effect of avalanche holds substantial significance within the domain of cryptographic algorithms. A block cipher of notable caliber is expected to promptly and profoundly alter the resulting cipher text upon even minor perturbations to either the plaintext or the encryption key. This quality ranks among the foundational design imperatives, representing a coveted criterion within the purview of cryptographic algorithm development. To assess the avalanche phenomenon's impact upon both the modified XTEA and the conventional XTEA, as have recourse to the subsequent mathematical expression.

$$\text{Avalanche Ratio} = \frac{\text{No. of flipped bits in ciphertext}}{\text{No. of bits in ciphertext}} * 100\% \text{ [22]} \quad (6)$$

The outcomes are displayed in Table 2. To assess the avalanche effect, were conducted a test where ten plaintexts were encrypted, producing ten corresponding cipher texts. Following this, were repeated the encryption process using the same ten plaintexts, but with a single-bit variation in the key that was used in the previous test. This led to pairs of cipher texts, each pair resulting from the same plaintext but with the key and its one-bit different version. Ultimately, were calculated the average percentage across all ten results to evaluate the extent of the avalanche effect in the algorithm, which measures how a slight change in input or key leads to a significant alteration in the output.

Table 2. Avalanche Effect Test

Block	XTEA	Our Proposal
1	46.87	47.65
2	48.43	49.21
3	50	50.78
4	45.31	61.7
5	51.56	52.34
6	53.12	53.29
7	50	54.68
8	54.68	55.46
9	56.25	58.59
10	53.12	61.71
Average	50.93	54.54

As depicted in Table 2, the altered XTEA demonstrates a superior avalanche effect (with an average of 54.54%) compared to the original XTEA (with an average of 50.93%).This difference highlights that modified XTEA is sensitive to minor modifications in plain text.

6.2 Key sensitivity test

The procedure was described involves testing the sensitivity of an encryption algorithm to changes in the encryption key. This is an important aspect of cryptographic security, as a "good" encryption algorithm should produce significantly different cipher texts when even a small change is made to the encryption key. The results of this experiment, as stated in Table 3, can offer perceptions into the key sensitivity between both of original and modified XTEA algorithm.

Fifteen blocks of unencrypted text were encrypted using a particular key, which was subsequently modified the last block of the same key. Subsequently, fifteen blocks were encrypted with the new key, and the difference between the cipher text blocks was computed. This test has been performed for both the original XTEA and modified XTEA, with the results being displayed in Table 3, and the test you described helps evaluate this important security characteristic.

Table 3. Key sensitivity test

Standard XTEA %	Block	Proposed %
43.4	1	50.00
43.5	2	50.10
44.11	3	47.80
44.32	4	46.90
46.55	5	50.02
47.78	6	50.10
46.91	7	60.10
55.12	8	55.81
36.98	9	46.88
39.85	10	49.81
48.11	11	53.20
48.75	12	51.11
45.7	13	58.01
46.99	14	55.11
49.19	15	60.33
45.81	Average	52.35

6.3 Randomness test

To test the keys used in XTEA modified algorithm, it must be subjected to a diverse of statistical tests designed to disclose the randomness of the sequence, were use a NIST statistical test as shown in Table 4.

Table 4. NIST test results for modified XTEA

NIST Test	P Value	Status Passed
Frequency (Monobit)	0.232330	Ok
Block frequency	0.127801	Ok
Cumulative sum	0.494146	Ok
Linear complexity	0.018712	Ok
Runs	0.067882	Ok
Non-overlapping template matchings	0.550106	Ok
Discrete fourier transform	0.121115	Ok
Serial	0.110325	Ok
Longest run	0.018812	Ok

6.4 Execution time

Calculating the execution time for both the original algorithm and the modified algorithm across varying block lengths, taking into account that the algorithm is implemented in google Colab with GPU acceleration, Table 5 states the obtain result of time across the different block size for both original and modified XTEA in two cases encryption and decryption.

Table 6 shows how long it takes to encrypt a modified XTEA TEA, XTEA, and XXTEA file with files of different sizes. As clear results show that modified XTEA encryption time is shorter than that of all the security algorithms currently in use in IoT networks.

Table 5. Encryption and decryption execution time for original and modified XTEA algorithms

Block Size (Byte)	XTEA Encryption Time (ms)		XTEA Decryption Time (ms)	
	Original	Modified	Original	Modified
128	1.9801	0.9791	2.4569	1.9801
256	1.9897	1.0097	3.0000	2.0290
512	2.1084	1.6084	2.6709	3.1003
1024	2.9017	2.1011	4.9084	2.9098
2048	3.8901	3.0201	5.9651	4.9876
Average	2.574	1.74368	3.80026	3.00136

Table 6. Encryption time of TEA, XTEA, XXTEA and modified XTEA algorithm

Block Size (Byte)	TEA	XTEA	XXTEA	Modified XTEA
128	2.009	1.9801	0.9801	0.9791
256	1.999	1.9897	1.0278	1.0097
512	3.0347	2.1084	1.6080	1.6084
1024	3.9081	2.9017	2.1009	2.1011
2048	4.1590	3.8901	3.0392	3.0201
Average	3.02196	2.574	1.7512	1.74368

6.5 Entropy analysis

Entropy is a term used in cryptography to describe how random or unpredictable the data being encrypted is. The optimal value of the data is near (8) when the entropy is high, indicating that the data is more random and hence harder to predict. The following formula can be used to determine the entropy of a cipher text.

$$H(s) = \sum_{i=0}^{2N-1} P(s_i) \log_2 \frac{1}{P(s_i)} \qquad (7)$$

Table 7. Comparison of entropy analysis between XTEA and modified XTEA algorithm

Blocks	XTEA	Modified XTEA
1	7.89981	7.90091
2	7.90121	7.99995
3	7.89999	7.89992
4	7.92908	7.94994
5	7.99886	7.99697
6	7.95941	7.99905
Average	7.93139	7.95779

The entropy value before and after changing the algorithm is displayed in Table 7. We observe that the modified algorithm's entropy value has increased. Thus, based on the data above, we can conclude that optimizing the algorithm can raise the cipher text's entropy value. Naturally, this is regarded as the outcome of a modification that is better equipped to fend off an entropy attack; the value of 7.95779(~8) indicates that it is safe from an entropy attack.

6.6 Computational complexity and resource consumption

Designing encryption algorithms involves considering complexity and resource utilization as crucial factors. Complexity theory illustrates the inherent challenges of computational problems, while resource consumption analysis examines the practical resource usage during program execution. Both concepts play a vital role in enhancing our comprehension and enhancement of algorithms and systems across various computing environments.

6.6.1 Resource consumption

The memory requirements of a standard XTEA algorithm can be described as follows. The key is 128 bits (16 bytes), block has 64 bits (8 bytes), key schedule generates 64 subkeys, each of which is 64 bits (8 bytes), for a total of 64 * 8 = 512 bits (64 bytes).

Therefore, the total memory requirement for XTEA can be calculated as 16 bytes (key) + 8 bytes (block) + 64 bytes (subkeys) = 88 bytes.

To calculate the memory requirements for the modified XTEA algorithm as follows. The block has 128 bits (16 bytes), each of the 64 keys produced by the Rossler system has 128 bits (16 bytes), totaling 64 * 16 = 1024 bits (128 bytes), the total memory requirement for XTEA can be calculated as:

$$\text{Total Memory} = (\text{Total Keys}+\text{Block Size})$$
$$T_m=128+16=144 \text{ bytes}$$

6.6.2 Computational complexity

The computational complexity of the standard XTEA algorithm can be calculated by dissecting it into its constituent parts:

1. Number of rounds: 64 rounds are used by XTEA for encryption and decryption.

2. Operations per round: Four fundamental operations are performed in each round of XTEA: two additions, one XOR operation, and one bitwise shift.

$$\text{Computational Complexity} = \text{Total Number of Rounds} * \text{Number of Operations in Each Round}$$
$$C_X=64*4=256$$

Modified XTEA algorithm operate on a 128-bit block size and each round includes eight basic operations (four additions, two XOR operations, and two bitwise shifts) and 64 rounds for both encryption and decryption, we can calculate the complexity as follows:

$$C_X=64*8=512$$

6.6.3 Time

To calculate the encryption time for (800 blocks) and We assume Time per Block is (0.08 ms), can use the given time per block and multiply it by the number of blocks:

$$\text{Encryption Time} = \text{Number of Blocks} * \text{Time per Block}$$
$$\text{Standard XTEA Et}=(800/64)*0.08=(1 \text{ ms})$$
$$\text{Modified XTEA Et}= (800/128)*0.08 = (0.5 \text{ ms})$$

7. CONCLUSION

The architecture of the Modified XTEA is ideal for devices where cost-effectiveness and minimal power usage are paramount. The suggested modefication has demonstrated excellent performance compared with standard XTEA. This algorithm design stems from a meticulous analysis of every element within the Modified XTEA algorithm. Our implemented approach has proven to be resilient against various types of attacks, as verified through security and statistical tests conducted on the system's outcomes. Its encryption speed, operational efficiency, and cost-effectiveness make it an excellent choice for resource-constrained applications, including passive Raspberry Pi setups and wireless sensor networks. Therefore, the algorithm can be applied in various uses, including email encryption, embedded systems, file encryption, network security, and unique network protocols for safe data transfers. It can also be used to encrypt data kept in databases or on storage devices.

ACKNOWLEDGMENT

The authors would be grateful to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) in Baghdad, Iraq for collaboration and support in the present work.

REFERENCES

[1] Collaguazo, A., Villavicencio, M., Abran, A. (2023). An activity-based approach for the early identification and resolution of problems in the development of IoT systems in academic projects. Internet of Things, 24: 100929. https://doi.org/10.1016/j.iot.2023.100929

[2] Mohamed, S., Hassan, A.M., Aslan, H.K. (2021). IoT modes of operations with different security key management techniques: A survey. International Journal of Safety and Security Engineering, 11(6): 641-651. https://doi.org/10.18280/ijsse.110604

[3] Hasan, M.Z., Mohd Hanapi, Z. (2023). Efficient and secured mechanisms for data link in IoT WSNs: A literature review. Electronics, 12(2): 458. https://doi.org/10.3390/electronics12020458

[4] Aziz Al Kabir, M., Elmedany, W., Sharif, M.S. (2023). Securing IoT devices against emerging security threats: Challenges and mitigation techniques. Journal of Cyber Security Technology, 7(4): 199-223. https://doi.org/10.1080/23742917.2023.2228053

[5] Ko, Y., Hong, S., Lee, W., Lee, S., Kang, J.S. (2004). Related key differential attacks on 27 rounds of XTEA and full-round GOST. In Fast Software Encryption: 11th International Workshop, FSE 2004, Delhi, India, pp. 299-316. https://doi.org/10.1007/978-3-540-25937-4_19

[6] Adriaanse, P.E. (2021). A comparative study of the TEA, XTEA, PRESENT and simon lightweight cryptographic schemes. Bachelor thesis, Delft University of Technology.

[7] Kotel, S., Zeghid, M., Machhout, M., Tourki, R. (2017). Lightweight encryption algorithm based on modified XTEA for low-resource embedded devices. In Proceedings of the 21st International Database Engineering & Applications Symposium, Bristol, United Kingdom, pp. 192-199. https://doi.org/10.1145/3105831.3105853

[8] Acholli, S., Ningappa, K.G. (2019). VLSI implementation of hybrid cryptography algorithm using LFSR key. International Journal of Intelligent Engineering and Systems, 12(4): 10-19.

https://doi.org/10.22266/ijies2019.0831.02

[9] Anusha, R., Veena Devi Shastrimath, V. (2019). LCBC-XTEA: High throughput lightweight cryptographic block cipher model for low-cost RFID systems. In Cybernetics and Automation Control Theory Methods in Intelligent Algorithms: Proceedings of 8th Computer Science On-line Conference 2019, Zlin, Czech Republic, pp. 185-196. https://doi.org/10.1007/978-3-030-19813-8_20

[10] Mishra, Z., Acharya, B. (2021). High throughput novel architectures of TEA family for high speed IoT and RFID applications. Journal of Information Security and Applications, 61: 102906. https://doi.org/10.1016/j.jisa.2021.102906

[11] Kella, C., Mishra, Z., Acharya, B. (2021). A compact & low power architecture of XXTEA192 lightweight block cipher. In 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, pp. 972-976. https://doi.org/10.1109/ICCES51350.2021.9489097

[12] Nagarajan, M., Rajappa, M., Teekaraman, Y., Kuppusamy, R., Thelkar, A.R. (2022). Renovated XTEA encoder architecture-based lightweight mutual authentication protocol for RFID and green wireless sensor network applications. Wireless Communications and Mobile Computing, 2022(1): 8876096. https://doi.org/10.1155/2022/8876096

[13] Shailaja, A. (2019). Design and Analysis of Lightweight Block Ciphers. Shodhganga: A Reservoir of Indian Theses.

[14] Ragab, A.A.M., Madani, A., Wahdan, A.M., Selim, G.M. (2023). Design, analysis, and implementation of a new lightweight block cipher for protecting IoT smart devices. Journal of Ambient Intelligence and Humanized Computing, 14: 6077-6094. https://doi.org/10.1007/s12652-020-02782-6

[15] Jinwala, D., Patel, D., Dasgupta, K. (2012). FlexiSec: A configurable link layer security architecture for wireless sensor networks. arXiv preprint arXiv:1203.4697. https://doi.org/10.48550/arXiv.1203.4697

[16] Rysak, A., Sedlmayr, M., Gregorczyk, M. (2023). Revealing fractionality in the Rössler system by recurrence quantification analysis. The European Physical Journal Special Topics, 232(1): 83-98. https://doi.org/10.1140/epjs/s11734-022-00740-1

[17] Zou, C.Y., Wang, L. (2023). A visual DNA compilation of Rössler system and its application in color image encryption. Chaos, Solitons & Fractals, 174: 113886. https://doi.org/10.1016/j.chaos.2023.113886

[18] Kontorovich, V., Beltrán, L.A., Aguilar, J., Lovtchikova, Z., Tinsley, K.R. (2009). Cumulant analysis of Rössler attractor and its applications. The Open Cybernetics & Systemics Journal, 3(1): 29-39. https://doi.org/10.2174/1874110X00903010029

[19] Albahrani, E.A., Alshekly, T.K., Lafta, S.H. (2023). New secure and efficient substitution and permutation method for audio encryption algorithm. The Journal of Supercomputing, 79(15): 16616-16646. https://doi.org/10.1007/s11227-023-05249-5

[20] Kaps, J.P. (2008). Chai-tea, cryptographic hardware implementations of xtea. In Progress in Cryptology-INDOCRYPT 2008: 9th International Conference on Cryptology in India, Kharagpur, India, pp. 363-375. https://doi.org/10.1007/978-3-540-89754-5_28

[21] Abed, S.E., Jaffal, R., Mohd, B.J., Alshayeji, M. (2019). FPGA modeling and optimization of a SIMON lightweight block cipher. Sensors, 19(4): 913. https://doi.org/10.3390/s19040913

[22] Hussein, S.N., Al-Alak, S.M. (2021). Secret keys extraction using light weight schemes for data ciphering. Journal of Physics: Conference Series, 1999(1): 012114. https://doi.org/10.1088/1742-6596/1999/1/012114