Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECURING E-VOTING DATA USING PROPOSED LIGHTWEIGHT ARX BLOCK CIPHER ALGORITHM

SAFA A. AHMED [1,2*], ALI M. SAGHEER [3]

[1] Informatics Institute for Postgraduate Studies, Information Technology & Communications University, Baghdad, Iraq
[2] Department of Chemical Engineering and Petroleum Pollution, College of Chemical Engineering, University of Technology, Baghdad, Iraq
[3]Department of Computer Networks Systems, College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq
Email: safa.a.ahmed@uotechnology.edu.iq[1,2], ali_makki@uoanbar.edu.iq[3]

ABSTRACT

The Internet of Things applications have developed fast, leading to the need to implement lightweight encryption schemes that can balance security with resources efficiency in resource-constrained environments. This paper suggests a lightweight ARX-based block cipher that combines dynamically generated substitution boxes based on a six-dimensional chaotic system with an adjusted key expansion algorithm based on AES. The plaintext is separated into two parallel streams, and the work of each round consists of key addition, nonlinear substitution by a dynamic S-Box, and ARX transformation in a symmetric Feistel-like network. The suggested key scheduling mechanism minimizes inter-key correlation and increases sensitivity to small key variations. The security evaluation methods used are entropy, Hamming distance, correlation analysis, NIST randomness tests, SAC and BIC tests. Experiments indicate that the proposed framework is highly diffusive, random, and efficient in encryption and decryption, thereby making it appropriate in protecting confidential database in a constricted environment.
Keywords: *Lightweight Cryptography, ARX Block Cipher, Key Expansion, Chaotic system, S-Box*

1. INTRODUCTION

With the fast development of cloud computing, intelligent systems and Internet of Things (IoT) applications in the modern world, the issue of data safety and secure communications begins to gain importance in the environment with limited memory and computing capabilities. Recent studies have come up with lightweight encryption algorithms that can fit in the processors that have low capacity to ensure that they become effective, simple and are not vulnerable to attacks. Among these trends, ARX (Addition-Rotation-XOR) algorithms have become prominent with high propagation efficiency, non-linearity, and speed, and exemplified with modern-day algorithms such as the Dynamic ARX-Based Cipher (DABC) which provide strong diffusion properties [1].

The Lightweight ARX White-Box Cipher (LWARX) is one of the White-Box methods that have proven to be popular in satellite communication contexts [2]. Others have worked on enhancing the encryption tools by creating the S-Boxes with the help of two-dimensional chaotic maps [3] or using meta-heuristic algorithms to create more random substitution boxes, such as BH-Box [4]. Even improved versions of Feistel architectures specifically created to work in an IoT environment have been developed with a trade-off between performance and security [5]. Studies in the field of data security offered detailed reviews of the possible vulnerabilities of lightweight encryption algorithms (linear and differential attacks) [6][7], while and other general surveys were devoted to the issues of the NIST lightweight encryption standard, either in relation to its implementation, resistance to attacks, or applicability to a number of industrial needs [8][9][10][11].

Also, symmetric and asymmetric encryption can be used in cloud and smart systems and their role.[12][13], and the significance of adding chaotic maps and sponge forms to the design of contemporary algorithms [14][15]. Modern hashing functions have become strong tools of key generation thanks to their high speediness and flexibility. Along with a series of studies that considered the work of hashing algorithms in

conventional devices and IoT-focused systems [16][17].

The design of new lightweight encryption algorithms, relying on new designs or new mathematical properties [18][9]. With most of the literature targeting ARX architectures, as they are relatively simple and secure. In 2023, Chen et al. introduced the Dynamic ARX-Based Cipher (DABC) algorithm, which has high propagation abilities and is resistant to the differential attack because the algorithm effectively uses addition, rotation, and XOR operations [1]. Another recent innovation in the same area is the Lightweight ARX White-Box Cipher (LWARX), an ARX algorithm designed to work with satellite communications, which is a good balance between speed and security [2].

In the area of developing encryption infrastructure, in 2023, Mohammad Shah et al. presented an improved Feistel network model suitable for IoT environments, where test results showed improvements in key sensitivity and propagation values, and reduced bit correlation [5].

In 2024, Hameed and Bahaa Abdulwahid have focused on developing S-Boxes using chaotic maps [3]. In 2023, Issa et al. refined algorithms in the Black Hole-based S-Box (BH-Sbox) to enhance linear and different resistance to the attack [4]. In terms of creating new technologies, in 2024, Kapalova et al. suggested a new lightweight encryption algorithm, which does not directly derive off an existing algorithm, but rather is designed using the principles of lightweight encryption design, with the focus being on simplification of computations and propagation as well as key sensitivity, making it an appropriate algorithm in resource-limited systems [18]. In 2023, Mohammed et al. introduced a lightweight encryption algorithm relying on Stream Cipher with the help of chaotic maps and a Sponge structure to improve the security of applications in the IoT. The suggested algorithm was differentiated together with the fact that it attains a good rate of randomness and propagation using low amount of resources, as well as high key sensitivity, making it resistant to differential attacks[14].

In 2023, Noura et al. introduced a new lightweight encryption algorithm named LESCA specifically targeting emerging systems with limited resources. This is an algorithm designed around a much more efficient Stream Cipher design, and propagation and randomness enhancements to improve resistance to linear and differential attacks[15]. In 2023, Ahmed et al. introduced an encryption system, which is the simplified implementation of the AES algorithm and requires less number of rounds to calculate and less computation complexity and better performance. The researchers concentrated on homomorphic AES features to permit the manipulation of encrypted data without their decryption to improve the privacy in sensitive systems [12].

In 2025, Khan et al. introduced a new lightweight encryption algorithm specifically applicable in resource-constrained systems in the IoT environment, the basis of which is a simple, highly efficient architecture with improved propagation and nonlinearity properties to enhance its resistance to differential and linear attacks [9]. In 2025, Tian and Li et al. introduce a new model of secure data sharing in the IoT context based on attribute-based encryption and blockchain technologies that can increase data integrity and avoid manipulation. The model is expected to resolve the issue of accurate access control and at the same time safeguard the privacy of the data when either two or more nodes exchange data s[13]. Research in the hashing functions area has made available study of the performance and applicability of the hash functions to key generations and enhancement of the security of distributed systems.

In 2025, Kwala et al. introduced a novel key exchange protocol, a Boolean-semiring architecture with the BLAKE3 hash function, called BKEX-B3. It is very flexible a protocol good example includes the “eXtendable Output Function (XOF)” of BLAKE3 which means that the key size and output may be changed on-demand, dynamically adjusting the security level to meet new quantum computing attacks [19]. Despite this development, there is still an unmet need of high security and the performance and resource limitations of interconnected systems. Hence, the study will seek to design and analyze an Attribute-Based Lightweight ARX Block Cipher algorithm that has been optimized to support key extension. This algorithm allows considering the user attributes into the generation process, reaching a new degree of security and personalization without reducing the efficiency that the systems with very limited capabilities need.

Although the current state of developing lightweight algorithms and substitution boxes development through chaotic systems or optimization algorithms has reached significant advances, the main shortcoming of the previous works is still evident: none of the studies has offered a comprehensive lightweight encryption algorithm, which is sensitive of the strength of the round function, the dynamics of the substitution boxes, and the efficiency of the key expansion cycle. Most work has focused either on improving the ARX architecture without enhancing key scheduling, or on designing independent chaotic S-Boxes without integrating them into a complete block cryptanalytic, or on relying on traditional key tables that may increase the correlation between keys in successive rounds.

This research presents a novel lightweight block cryptanalysis based on a balanced ARX architecture with a Feistel-inspired substitution mechanism. It integrates a dynamic set of substitution boxes generated using a six-dimensional chaotic system, along with an improved key expansion mechanism inspired by AES but based on a chaotic S-Box and a custom transformation function. This integration achieves a practical balance between agility and security, enhances diffusion and non-linearity, and reduces correlation between round keys, thereby increasing the algorithm's resistance to differential and linear attacks, while maintaining its suitability for application in IoT and resource-limited systems environments. Compared with conventional lightweight ARX ciphers, the proposed algorithm additionally integrates dynamic chaotic S-Boxes and an enhanced key expansion mechanism to improve nonlinearity, diffusion, and resistance against statistical, differential, and linear attacks. In the remainder of this paper, Section 2 shows the material and methods, Section 3 reports on experimental results and discussion, finally, the conclusion is provided in Section 4.

2. PROPOSED METHOD

The suggested lightweight block cipher algorithm incorporates three essential elements a round function, which uses the ARX architecture, modified key scheduling scheme, and a dynamically generated S-Box. The design will provide a safe and computationally efficient encryption framework that is suitable for resource-constrained environments (i.e. Internet of Things (IoT) systems, embedded systems, and wireless sensor platforms). The encryption and the decryption process take place in the suggested procedure. The description of the process of encryption and its detailed stages are clarified firstly, and the process of decryption is described briefly as the direct opposite of encryption to demonstrate how simple and convenient the structure is.

2.1 Encryption Process

Encryption is a very important step in the proposed algorithm, in which plaintext is transformed to ciphertext after a sequence of organized mathematical functions. figure 1 shows the overall architecture and the functional blocks in which the encryption process is composed.

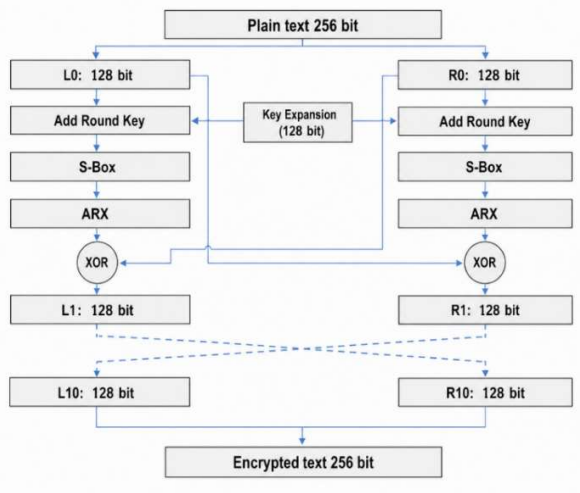


Figure 1: Proposed Lightweight Block Cypher Encryption Algorithm

The suggested algorithm consists of a combination of three cryptographic properties into an efficient lightweight encryption design, i.e. a six-dimensional chaotic system generating high-speed and unpredictable S-Boxes, an ARX-based key-expansion design that exploits the XOR operations, rotating transformations, and round constants to produce strong subkeys and an internal round function based on Addition-Rotation-XOR to bring about successful diffusion and nonlinearity. The encryption algorithm begins with the division of the 256-bit plaintext in two independent parts, the right and left halves that are 128 bits as illustrated in figure 1. The dynamic key scheduling module produces a round key and sends the key to the two branches. Each half-round then follows three fundamental processes in each round: Add Round Key, nonlinear replacement with the dynamically generated S-Box and an ARX transformation

comprising of modular addition, rotation, and XOR. Both data streams are processed in parallel and that is why the processes are performed efficiently, and the diffusion of the data is cryptographically secure. The symmetric design reduces the computational cost in addition to offering a simple methodology of ensuring that the security effectiveness is equally distributed on both sides of the cipher make it applicable to situations with resource restrictions, such as IoT devices, and embedded systems.

2.1.1 Proposed dynamic S-box

The 6D chaotic system is used in generating nonlinear and highly unpredictable values to create substitution boxes (S-boxes). The 6D chaotic system has six equations that is, Eq. (1), Eq. (2), e Eq. (3), Eq. (4), Eq. (5), Eq. (6) that have state variables (x_1 , x_2 , x_3 , x_4 , x_5 , x_6) and system parameters (α, β, a, b, c, d, e, f, r, and k) [20]. Firstly, the 6D system is used to generate chaotic sequences based on the provided initial conditions and control parameters. These sequences are then normalized, rounded, and converted to integers.

$$\dot{x}_1 = \alpha(1 - \beta|x_6|)x_2 - ax_1 \quad (1)$$

$$\dot{x}_2 = cx_1 + dx_2 - x_1x_3 + x_5 \quad (2)$$
$$\dot{x}_3 = -bx_1 + x_1^2 \quad (3)$$
$$\dot{x}_4 = ex_2 + fx_4 \quad (4)$$
$$\dot{x}_5 = -rx_1 - kx_5 \quad (5)$$
$$\dot{x}_6 = -x_2 \quad (6)$$

The values of the six variables are stacked together in a haphazard array with dimensions 257x6, which is used to create the S-Boxes. The chaotic string is acquired and then six S-Boxes are created each time in a process that is carried out on each of the six columns of the array. To overcome the initial instability, the first value in each column is removed, and further values in the chart are set in ascending order. Rank indices are used as the basis in the generation of a scale of integer numbers in the scale of [0-255] by subtracting one number in the index. The indices are reorganized into a 16X16 matrix to represent one single S-Box as shown in figure 2. The process is carried out on six dimensions of the chaotic system and 43 complete cycles to create 256 S-boxes, each one sized in 16X16. Algorithm 1 describe how the proposed S-Box is dynamically generated.

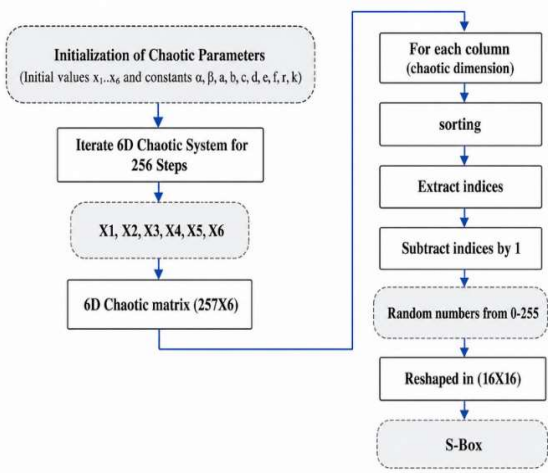


Figure 2: Chaotic-Based S-Box Generation

Algorithm 1. Proposed S-Box generation

***Step 1**: Determine the length of 256.*
***Step 2**: Generate a chaotic sequence using one dimension of the 6D chaotic system*
***Step 3**: Sort the generated chaotic sequence in ascending order.*
***Step 4**: Extract the ranking indices of the sorted sequence*
***Step 5**: Rearrange the master key bits according to the extracted indices.*
***Step 6**: Generate the permuted master key for the key expansion stage.*

2.1.2 Proposed modified key expansion

The Key Expansion mechanism in the proposed algorithm relies on a tightly controlled sequence of operations that combines chaotic dynamics and binary key processing with AES key expansion to produce highly random, nonlinear key expansions, as shown in figure 3.

This key expansion process is proposed by adding a permutation stage, which is based on chaotic indices generated by the six-dimensional chaotic system. A chaotic sequence having the length of 256 is first generated and sorted in ascending order. The sorted sequence is then scored with ranking indices, and the original half of the secret key bits are rearranged to form a permuted structure of key bits which has less internal correlation and greater randomness prior to the round key generation process which is describe in algorithm 2.

Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 2. Permutation process in the proposed key expansion

Step 1: *Initialize the 6D chaotic system parameters and initial conditions.*
Step 2: *Iterate the chaotic system for 257 iterations to generate chaotic sequences.*
Step 3: *Store the generated chaotic values in a 257×6 matrix.*
Step 4: *Remove the first row of each chaotic sequence to eliminate initial instability.*
Step 5: *Sort each column independently in ascending order.*
Step 6: *Extract the ranking indices of the sorted values.*
Step 7: *Subtract one from each index to generate integer values within the range [0–255].*
Step 8: *Reshape the resulting sequence into a 16×16 matrix to construct the dynamic S-Box.*
Step 9: *Repeat the process for all six chaotic dimensions and multiple iterations to generate 256 dynamic S-Boxes.*

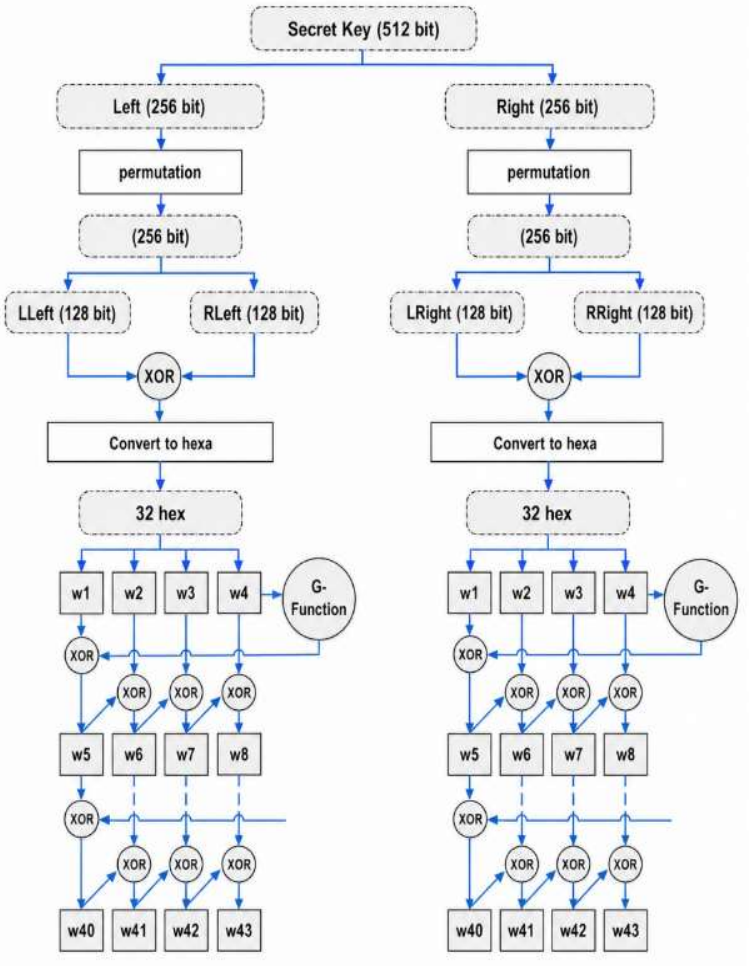


Figure 3: Proposed Key Expansion

Initially, the secret key is converted from a hexadecimal format to a binary representation. Next, the binary vector is divided into two independent halves of 256 bits each. Then, the bits for each half are internally rearranged based on a random order generated by a real number generator to minimize internal correlations. An XOR operation is applied between two equal segments of each half to enhance randomness and reduce linear correlation. The resulting bits are reformulated as (32×4) arrays, which are then transformed into a hexadecimal representation to generate two final key segments: the left key and the right key. These segments represent the starting point for both the left and right paths in the expansion phase. After the initial key is prepared, a specific row is selected from the resulting six-dimensional chaotic array. Two numerical values are then extracted and

transformed into the range [0–255] using division by 256. These values serve as pointers for selecting two dynamic replacement boxes (S-Box and Inverse S-Box) from among 256 pre-existing boxes for each path. Each input key fragment in each trajectory is split into four prime words w_1, w_2, , w_3 and w_4. These prime words are then used with a dynamic substitution S-Box and a round constant RC_i, which it has ten values("01", "02", "04", "08", "10", "20", "40", "80", "1b", "36") each used in 10 round for in a G-Function to generate a transition word.

The G-Function function as shown in figure 4. is used in the proposed algorithm as a key-generation element in the add round key phase. It performs a role like the key-expansion function in AES but incorporates a dynamically selected replacement box derived from the chaotic system. The process begins by receiving a four-byte key word in hexadecimal format B_0, B_1, B_2, and B_3, along with the dynamically selected replacement box and the round constant RC_i. A byte-level rotation is first performed to rearrange the key components, thus enhancing diffusion.

The resulting keyword is divided into hexadecimal pairs, then passed through a nonlinear substitution process, where each pair is processed using the Proposed dynamic S-Box, which performs a nonlinear substitution process at the byte level, converting each hexadecimal pair into a row and column index within the proposed dynamic S-Box and then substituting it for the corresponding value. The values are then converted back to form of hexadecimal to be used in further expansions. Lastly, XOR is used to combine the replacement word with the round constant RC_i, given a time difference within each round, and eliminating the symmetry between the successive round key.

These words are then operated in a sequence of XORs to form four new words in each round which are then joined together to form an independent round key. This procedure is repeated ten times in succession, producing a series of unique round keys on the left and the right paths. The combination of the proposed dynamic S-Box, XOR operations and rotation make the keys highly dispersed and nonlinear. In this manner, the multi-level architecture finds a useful compromise between run time performance and security level which heightens the resistance of the algorithm to key analysis, differential and linear attacks, at the same time ensuring its compatibility with resource-constrained environments.

The key expansion mechanism proposed was designed to reduce the amount of correlation between subsequent round keys and make it sensitive to small changes in the input. The permutation operations, the XOR transformations, the rotation shift, and the dynamically selected chaotic S-Boxes provide an improved nonlinearity and diffusion in the round key generation. Furthermore, left and right expansion paths are generated independently, making adjacent-round key predictions more difficult and preventing related-key attacks and/or statistical attacks. Thus, the proposed key scheduling process not only has lightweight computational complexity for resource constrained environments but also has good randomness and security.

2.1.3 Add round key

The Add Round Key step is one of the most important parts of encryption process in each round so that the encrypted message is directly connected with round keys that are generated during the initial stage. The round key of that path is chosen among the arrays of expansion keys and then each segment is combined with the current data block in the right and left paths using XOR. The output is divided into 16-bit blocks (four hexadecimal digits), converts each block to a decimal value, and binary XORs the resultant values, and then converts the final value into a hexadecimal one again and joins it to the previous block until the whole ciphertext is encrypted. The mechanism provides a strong, non-linear fusing of the data and the round key, upholds the ambiguity principle of the algorithm and avoidance of the separation of key and plaintext effect. This further enhances resistance of the system to key resolving and differential attacks and at the same time makes it simple to implement and applicable to systems with limited resources.

2.1.4 Diffusion process (S-Box substitution)

Once the Add Round Key phase has been applied, a nonlinear substitution on data blocks in the left and right paths is performed by dividing the resultant hexadecimal block into diverse sets of two sequential bits that are reassembled back to a two-dimensional array of independent bytes. The two pairs are then transformed into two number pointers that are used to ascertain the row and column number in the chosen substitution S-Box which is already predetermined. Every single byte is substituted by its own value in the suggested

dynamic S-Box, and this has a solid nonlinear transformation that enhances the concept of confusion in the algorithm.

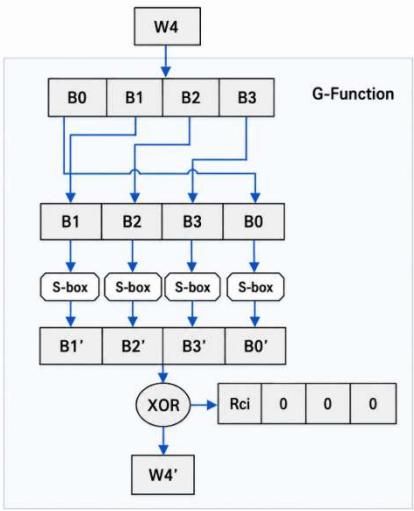


Figure 4: The G-Function

Once the replacement of all the bytes is done, the resulting numbers are coded to a hexadecimal format and re-packaged into a continuous string that they use in the following steps of the round function. This methodology not only increased the structural complexity of the proposed system, but also restricts the applicability of differential and linear attacks and conserves computational efficiency to the resource-constrained applications.

2.1.5 Addition-Rotation-XOR (ARX)

After the nonlinear substitution phase comprising of dynamic substitution boxes, the Addition-Rotation-XOR (ARX) round operation is used to maximize diffusion and nonlinearity of the proposed cipher architecture. At this phase, the input data block is split into four words of equal length and denoted in hexadecimal form. Each word is then rotated using different offset values, which makes sure that the effect of each fragment of data is different. It is then followed by modular addition on the selected words followed by XOR which causes nonlinear mixing of the block components thereby facilitating diffusion. This operation is based on simple and inexpensive operations like addition, rotation, and XOR, and can therefore be implemented in environments with resource constraints with an appropriate degree of security. The use of ARX architecture with the aim of resisting the differential and linear attacks makes the mathematical relationships between the plaintext and ciphertext complex and the effect of any slight change in the input is spread over the whole output block therefore augmenting the robustness and efficiency of the proposed algorithm as indicated in figure 5.

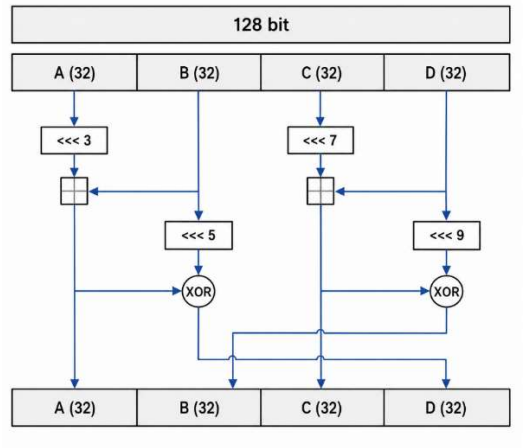


Figure 5: ARX

Following each round of encryption, a structural change of the left and right paths is carried out to provide diffusion within the architecture of the algorithm. The data of that path is combined with the ARX transform data of the path by XOR. The new right path is obtained by adding the last left path and the right transform output and the new left path is obtained by adding the last right path with the left transform output. The paths are then swapped, a method inspired by Feistel architectures, to make sure that the impact of nonlinear operations is transferred between the paths on the successive rounds. This step is repeated ten times with each round making the effect of any minor variation in the input spread throughout the data structure. The two paths that have been, at the end of the last round, merged into a single block, are used to create the final ciphertext. This supports the values of ambiguity and diffusion, making the algorithm resistant to the forms of differential and linear attack, and is still computationally efficient enough to operate in resource-constrained systems.

2.2 Decryption Process

The Algorithm that will be proposed is based on the systematic reversal of the encryption steps to ensure the decryption is performed with the help of an ARX based symmetric architecture and a Feistel inspired switching network. This enables the sharing of the round keys without the need to have a different decryption structure. This is initiated by

the receipt of the ciphertext and breaking it down into two 128-bit paths (left and right). The process of decryption of ten rounds is then reversed on the round keys in the same order. The reverse switching between the two paths is then done in each round, then the output of the ARX transformation is combined with the respective path using XOR. A second ARX transformation is then executed in which the rotation and modular addition operations are inverted. Substitution is then undertaken in a nonlinear way with an inverse S-Box that is associated with the substitution box employed in encryption step. Lastly, the Add Round Key operation is carried out with the help of a suitable round key. These cycles are repeated in this order until all the rounds have been used, at which point the two resulting paths are joined up, and the original plain text is restored. This symmetrical structure guarantees high implementation efficiency and easy application and preserves the same security properties obtained during the encryption process making the algorithm fit in resource-constrained resources like the Internet of Things and embedded systems.

3. EXPERIMENTAL RESULTS AND DISCUSSION

The experiment was accomplished using a processor of type Intel Core i7, a RAM of 16GB and MATLAB R2024a environment. It was evaluated to determine the degree of security; speed of computation and scalability as compared to the conventional cryptographic designs. The efficiency and effectiveness of the suggested lightweight block cypher encryption model are tested with the help of the database of the Iraqi voter registration during the parliamentary elections. This database will have 128 MB of 1 million records i.e. one record per voter. (PER_ID: voter ID, PER_FAMNO: voter family number, PER_FIRST: voter first name, PER_FATHER: voter middle name, PER_GRAND: voter last name, PER_FULLNAME: voter full name, PER_DOB: voter date of birth, PER_GENDER: voter gender, PER_BRNO: voter electoral branch number, PER_PGNO: voter electoral page number, PER_PCID: voter electoral centre ID, PC_VRCID: ID of the district area to which the polling station belongs, PER_VRTYPE_2013: type of voter registration according to the 2013 register). Figure 6 shows a sample of Iraqi voter registration in the parliamentary elections database.

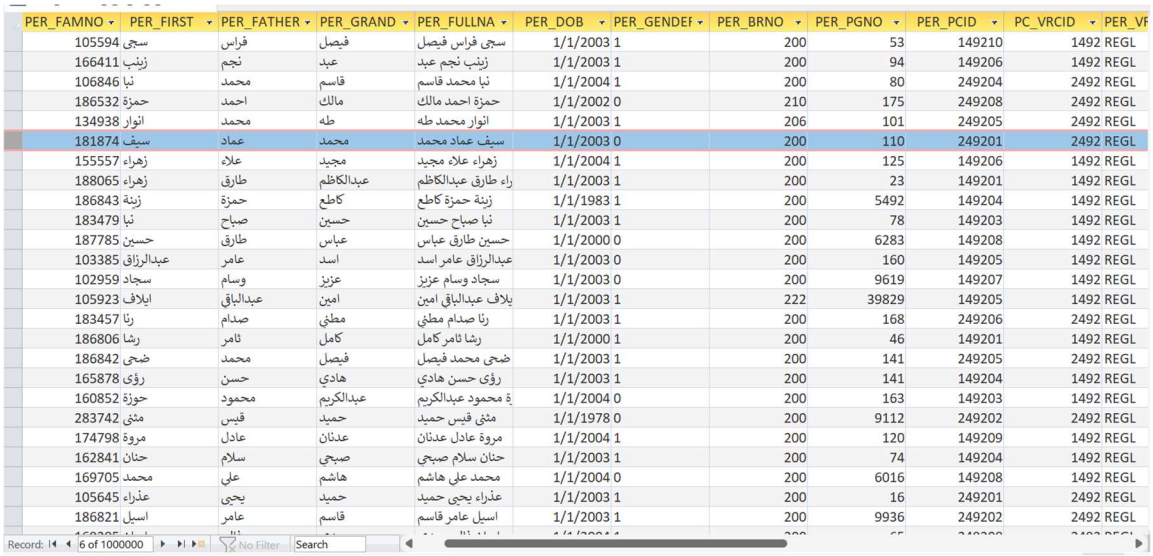
Each record in the election database is individually processed, with all the cells of a single record concatenated into a single cell to ensure uniformity and data integrity during encryption, as shown in table 1. Specifically, every row is retrieved and converted into its corresponding hexadecimal representation, after which appropriate padding is applied to achieve a standardized length of 512 hexadecimal digits. Every row of the database stores data regarding a single voter, due to space limitations, only some of the hexadecimal representations are included. As shown in table 2, and this preprocessing step is done to all the records of the database in a systematic manner that ensures employment of a uniform and safe encryption process. To ensure the effectiveness and strength of the presented lightweight block cypher encryption model, some experimental trials were performed with the focus on the plaintext and ciphertext statistical analysis, key security analysis, S-Box cryptographic properties.

Table 1: Sample of Iraqi Voter Registration in The Parliamentary Elections Database After Record Concatenation

Record No.	Concatenated Data
1	10033921,105594,"سجى فراس","فيصل","فراس","سجى" 2003/1/1,"فيصل" 53,1,149210,1492,"00168",200,"1",00:00:00,"REGL"
2	10033923,166411,"زينب نجم","عبد","نجم","زينب" 2003/1/1,"عبد" 94,1,149206,1492,"م0627",200,"1",00:00:00,"REGL"
3	10033926,106846,"نبا محمد","قاسم","محمد","نبا" 2004/1/1,"قاسم" 80,1,249204,2492,"م0278",200,"1",00:00:00,"REGL"

Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



PER_FAMNO	PER_FIRST	PER_FATHER	PER_GRAND	PER_FULLNA	PER_DOB	PER_GENDEI	PER_BRNO	PER_PGNO	PER_PCID	PC_VRCID	PER_VF
105594	سجى	فراس	فيصل	سجى فراس فيصل	1/1/2003	1	200	53	149210	1492	REGL
166411	زينب	نجم	عبد	زينب نجم عبد	1/1/2003	1	200	94	149206	1492	REGL
106846	نبا	محمد	قاسم	نبا محمد قاسم	1/1/2004	1	200	80	249204	2492	REGL
186532	حمزة	احمد	مالك	حمزة احمد مالك	1/1/2002	0	210	175	249208	2492	REGL
134938	انوار	محمد	طه	انوار محمد طه	1/1/2003	1	206	101	249205	2492	REGL
181874	سيف	عماد	محمد	سيف عماد محمد	1/1/2003	0	200	110	249201	2492	REGL
155557	زهراء	علاء	مجيد	زهراء علاء مجيد	1/1/2004	1	200	125	149206	1492	REGL
188065	زهراء	طارق	عبدالكاظم	راء طارق عبدالكاظم	1/1/2003	1	200	23	149201	1492	REGL
186843	زينة	حمزة	كاطع	زينة حمزة كاطع	1/1/1983	1	200	5492	149204	1492	REGL
183479	نبا	صباح	حسين	نبا صباح حسين	1/1/2003	1	200	78	149203	1492	REGL
187785	حسين	طارق	عباس	حسين طارق عباس	1/1/2000	0	200	6283	149208	1492	REGL
103385	عبدالرزاق	عامر	اسد	عبدالرزاق عامر اسد	1/1/2003	0	200	160	149205	1492	REGL
102959	سجاد	وسام	عزيز	سجاد وسام عزيز	1/1/2003	0	200	9619	149207	1492	REGL
105923	ايلاف	عبدالباقي	امين	يلاف عبدالباقي امين	1/1/2003	1	222	39829	149205	1492	REGL
183457	رنا	صدام	مطلق	رنا صدام مطلق	1/1/2003	1	200	168	249206	2492	REGL
186806	رشا	ثامر	كامل	رشا ثامر كامل	1/1/2000	1	200	46	149201	1492	REGL
186842	ضحى	محمد	فيصل	ضحى محمد فيصل	1/1/2003	1	200	141	249205	2492	REGL
165878	رؤى	حسن	هادي	رؤى حسن هادي	1/1/2003	1	200	141	249204	2492	REGL
160852	حوزة	محمود	عبدالكريم	ة محمود عبدالكريم	1/1/2004	0	200	163	149203	1492	REGL
283742	مثنى	قيس	حميد	مثنى قيس حميد	1/1/1978	0	200	9112	249202	2492	REGL
174798	مروة	عادل	عدنان	مروة عادل عدنان	1/1/2004	1	200	120	149209	1492	REGL
162841	حنان	سلام	صبحي	حنان سلام صبحي	1/1/2003	1	200	74	149204	1492	REGL
169705	محمد	علي	هاشم	محمد علي هاشم	1/1/2004	0	200	6016	149208	1492	REGL
105645	عذراء	يحيى	حميد	عذراء يحيى حميد	1/1/2003	1	200	16	249201	2492	REGL
186821	اسيل	عامر	قاسم	اسيل عامر قاسم	1/1/2003	1	200	9936	249202	2492	REGL

Record: 6 of 1000000 No Filter Search

Figure 6: Sample of Iraqi Voter Registration in the Parliamentary Elections Database

Table 2: Partial Sample of Iraqi Voter Registration in the Parliamentary Elections Database After Processing

Record No.	Data in Hexadecimal
1	0310300300330330390……..504704C022000000000000000000000000000000000000000
2	2C02203102202C03203……4504704C022000000000000000000000000000000000000000
3	03003402C0320340390……..202C02205204504704C022000000000000000000000000000

3.1 Statistical Analysis of Plain Text and Ciphertext

This section aimed to evaluate the statistical properties of the plaintext and ciphertext generated by the proposed algorithm, specifically to verify its ability to conceal the original statistical patterns of the data. To compare the amount of noise and diffusion produced by the encryption process, as well as the resistance of the algorithm to statistical and differential attacks, a set of simple statistical tests, such as Hamming distance, entropy, and the correlation coefficient, was performed.

3.1.1 Hamming distance analysis

The purpose of Hamming distance analysis is to quantify the number of bits that differ between the plaintext and the encrypted message that comes out of the proposed algorithm, in order to test the magnitude of diffusion that the encryption procedure has managed to accomplish. Each record was taken separately and the Hamming distance was determined by comparing the bits of the plaintext and ciphertext. This distance is the count of bits whose value has been altered due to the process of encryption. table 3 presents the Hamming distance values of a group of records of different lengths in bits (1628 to 1996 bits) and the Hamming distance values were between 845 and 1026 bits. The findings reveal that the mean number of changing bits is about half of the record length, almost the optimal value in a secure encryption system, and this indicates a very strong avalanche effect.

Table 3: Hamming Distance of Records

Record No.	Record length in bits	Hamming distance
1	1872	969
2	1908	979
3	1760	894
4	1628	845
5	1752	902
6	1996	1026
7	1632	848
8	1748	895
9	1744	910
10	1884	978

11	1768	923
12	1792	920

These values indicate the capability of the proposed algorithm to spread the effect of any alteration in the plaintext over a lot of ciphertext bits, thereby lowering the statistical correlation between the two texts and increasing the resistance of the algorithm to the differential and statistical attacks. The findings also affirm the fact that the ARX architecture, which is further complemented by key shuffling and dynamic substitution boxes, is effective in contributing to the high, as well as, balanced propagation in different data length, which is an effective interpretation of the suitability of the algorithm in encrypting different databases and applications with both large and small data size.

3.1.2 Entropy analysis

One of the most crucial statistical signs applied to determine the extent of encrypted data randomness and unpredictability is entropy. The use of high entropy values implies that there is almost a regular distribution of symbols, and exploitable statistical patterns are significantly reduced. table 4 demonstrates the value of entropy of the plaintext prior to encryption, and the encrypted message after the application of the proposed algorithm to a sample of various records.

Table 4: Entropy of Text Before and After The Encryption Process

Record N0.	Entropy before encryption	Entropy after encryption
1	4.98367	7.92809
2	4.80894	7.86781
3	6.42104	7.90258
4	5.65408	7.88624
5	6.11901	7.89000
6	4.62762	7.89887
7	4.80665	7.89525
8	5.10750	7.86657
9	4.93659	7.94789
10	4.54127	7.94034
11	4.99137	7.89432
12	4.93074	7.85197

These outcomes indicate that the entropy of the unencrypted samples was between 4.54 and 6.42 bits, indicating a definite statistical pattern in the original texts. Contrastingly, the entropy levels post-encryption marked 7.85 and 7.95 bits which were quite near the optimal entropy level (8 bits) in binary data, and this was a high level of randomness. This high entropy post-encryption value proves that the proposed algorithm is efficient in eliminating the statistical characteristics of the plaintext and minimizes the chance of obtaining any valuable information out of the encrypted text. The findings also reveal that the performance of the algorithm does not change significantly in connection to various records, and this makes it more resistant to statistical attacks and proves that the algorithm is well designed to deliver strong encryption when resources are limited to support the environment.

3.1.3 Correlation analysis

The measurement of the level of statistical correlation in plaintext and ciphertext is determined through correlation analysis. A value near to zero means that the relationships between the values are weak or non-existent, which is a considerable sign of the strength of encryption and resistance to attacks on statistical analysis. table 5 presents the correlation coefficient between the plaintext and ciphertext obtained after the application of the proposed algorithm to a collection of various records.

Table 5: The Correlation Between Plain and Cipher Text

Record No.	Correlation between plain and cipher text
1	-0.01492
2	0.07593
3	0.02521
4	-0.08644
5	0.02629
6	0.07945
7	-0.06831
8	0.01576
9	0.04804
10	0.09794
11	0.01689
12	-0.05604

The results indicate that the correlation coefficients of all records are within a very minimal range and near to zero as they are between -0.08644 to 0.09794. These negative or positive low values validate the fact that there is no statistically significant correlation between plaintext and

ciphertext and this is confirmation that the proposed algorithm would work in breaking the structural correlation between the original data and the encrypted output. These findings also show the stability of the algorithm behaviour under varying records and the absence of a consistent correlation pattern which increases the ability to resist statistical analysis attacks and known-plaintext attacks. Thus, the insignificant values of correlation indicate a high degree of the confusion property realized by the suggested algorithm and proves its appropriateness to use in data security applications in resource-constrained settings.

3.2 Key Security Analysis

The purpose of this section is to assess the level of security of the keys produced by the offered algorithm with the help of a series of standard statistical tests. The analysis involves NIST tests to confirm the importance of randomness, entropy analysis to test uncertainty and Hamming distance analysis to gauge the sensitivity of the key to even small modification. The tests reveal the robustness of the keys and their immunity against a range of splicing attacks thereby validating their aptitude in confidentiality encryption in the setting of resources contentiousness.

3.2.1 NIST statistical test

Randomness of keys produced in the proposed algorithm was tested by a series of standard test suites of the NIST Statistical Test suite based on the key length of 1024 bits, 2048 bits, and 3072 bits. table 6 contains the averages values of the p-values of each of the tests; a test is a success when the p-value exceeds the set level of significance (0.01).

Table 6: NIST Test of Generated Sequences

	NIST Test	Key length			Average	Status
		1024	2048	3072		
1	Frequency test	0.079516	0.014831	0.022852	0.039066	pass
2	Frequency test with block	0.000155	0.010676	0.007371	0.006067	pass
3	Run test	0.018619	0.033854	0.006416	0.01963	pass
4	Longest run ones in a block	0.006541	0.086899	0.003773	0.032404	pass
5	Binary matrix rank	0.044596	0.000274	0.002713	0.015861	pass
6	Discrete Fourier transform	0.005714	0.007408	0.011936	0.008353	pass
7	Non-overlapping template matching	0.011128	0.008287	0.002223	0.007213	pass
8	Overlapping template matching test	0.026081	0.062475	0.004295	0.03095	pass
9	Approximate entropy test	0.010298	0.025959	0.048263	0.028173	pass
10	Cumulative sum test	0.0707	0.002628	0.003756	0.025695	pass

The findings reveal that all the keys produced were able to complete all the listed NIST tests that are the frequency test, the frequency within blocks test, the run test, the longest run of one's in the block test, the binary matrix rank test, the discrete Fourier analysis test, the nested and non-nested template test, the approximate entropy test, and the cumulative sum test. Moreover, the mean p-value of the various key lengths was found to be in the statistically acceptable range which showed that the randomness properties were not dependent on the key length. The above findings indicate that the proposed algorithm can produce keys with good statistical characteristics and high randomness, and thus is more resistant to statistical analysis attacks,

and is valid in the application of secure encryption systems, especially in a set up with limited computing ability.

3.2.2 Entropy analysis

Entropy analysis is one of the indicators that is important in determining the degree of randomness and uncertainty of generated keys. High values of entropy indicate a higher level of uniform distribution of bits making it more challenging to predict the key or even the statistical patterns of the

key. Table 7 indicates the entropy values of the keys produced by the proposed algorithm in various key lengths 1024, 2048 and 4096 bits.

Table 7: The Entropy Value of the Generated Key Sequences

Entropy of key	1024-bit	2048-bit	4096-bit
1	5.937472	6.410065	6.018378
2	6.336767	5.725153	5.770985
3	6.056237	6.185506	6.064291
4	5.788857	5.705554	6.371497
5	5.896295	6.466295	6.099058
6	6.278642	5.685717	5.961928
Average	6.049045	6.029715	6.04769

The experiment results indicate that the mean entropies were 6.049045 with 1024-bit keys, 6.029715 with 2048-bit keys, and 6.04769 with 4096-bit keys. These similar values imply that the degree of randomness is constant despite the key length and imply that the distribution of bits in generated keys is almost equal. These findings indicate the effectiveness of the key generation algorithm in the proposed algorithm and how it can generate keys that have strong statistical properties and thereby minimizes the probability of success of statistical analysis attacks. The fact that high levels of entropy are achieved at various key lengths also assures us of the appropriateness of the algorithm being proposed to be used in secure encryption applications, and this is particularly in resource-constrained applications that need to have a trade-off between security and efficiency.

3.2.3 Hamming distance analysis

Key sensitivity is measured by Hamming distance analysis. It quantifies the amount of bit discrepancies between two keys because of minor alterations in their starting values. Any value that is near to half of the key length is a good indication that it is highly randomized and resistant to key replication attacks. The results of Hamming distance analysis between various key pairs as generated using the proposed algorithm in the key length of 1024 bits, 2048 bits and 4096 bits are presented in table 8. The experiment demonstrates that the mean Hamming distance of 1024-bit keys, 2048-bit keys and 4096-bit keys are 522.2 bits, 1054.5 bits and 2114.4 bits respectively, which are quite close to the ideal percentage (50) of the key length.

Table 8: Hamming Distance Between Generated Key Sequences

First key No.	Second key No.	Hamming distance	Hamming distance	Hamming distance
		1024-bit	2048-bit	4096-bit
1	2	507	1020	2088
1	3	534	1098	2180
1	4	525	1058	2142
1	5	500	1094	2018
2	3	541	1022	2105
2	4	522	1031	2116
2	5	523	1010	2185
3	4	511	1076	2038
3	5	523	1041	2184
4	5	536	1095	2088
Average		522.2	1054.5	2114.4

These outcomes indicate that a slight alteration in the key generation input results in a broad range of variation in the resulting bits, which implies that the sensitivity of the key is high and it is thus much harder to predict or re-form the key. Thus, the suggested algorithm provides sufficiently strong security against key sniping attacks, and it has the features, which can be applied in secure encryption systems, especially in resource-capped conditions.

3.3 S-Box Cryptographic Properties

S-Box is an essential element of symmetric algorithms that is a key factor in realization of nonlinearity and promotion of the principle of confusion. This part is an analysis of the cryptographic characteristics of a proposed dynamic S-Box by applying a number of standard tests, such as the Strict Avalanche Criterion (SAC) and Bites Independence Criterion (BIC), to determine the functionality of the algorithm to propagate variations and minimize statistical correlations, making the algorithm more resistant to differential and linear attack strategies.

The results of obtained SAC and BIC indicate that the proposed dynamic S-Boxes are both good nonlinear and balanced diffusion properties. The constructed S-Boxes are dynamically generated from chaotic sequences, which will vary during the operation: it is difficult to construct stable statistical approximations. These properties make them

resistant against differential and linear cryptanalysis and have low computational complexity that is appropriate for lightweight cryptographic applications.

3.3.1 Strict avalanche criterion (SAC)

The Strict Collapse Criterion (SAC) is one of the most important metrics used to evaluate the quality of S-Boxes. It measures the impact of a single input bit change on the output bits. A good S-Box is assumed to result in a random, independent change in approximately half of the output bits. table 9 shows the correlation values between the collapse groups generated by S-Boxes using the proposed chaotic system.

Table 9: Correlation Between Avalanche Variable Sets of Generated S-Boxes

S-Box No.	1	2	3	4	5	6
1	0	0.04554	0.19249	0.006272	0.076902	0.13179
2	0.04554	0	0.039965	-0.01665	0.043294	0.104323
3	0.19249	0.039965	0	0.025463	0.044466	-0.0172
4	0.006272	-0.01665	0.025463	0	0.043335	-0.0123
5	0.076902	0.043294	0.044466	0.043335	0	-0.0312
6	0.13179	0.104323	-0.0172	-0.0123	-0.0312	0

Since the SAC is typically assessed by analysing the correlation between collapse variables, values close to zero are a direct indicator of achieving this criterion. The values ranged from −0.0312 to 0.19249. These low values indicate weak statistical correlation between the outputs of the S-Boxes when a single input bit is changed, demonstrating the achievement of strong and balanced collapse characteristics. Furthermore, the absence of any high or regular correlation pattern between the different S-Boxes confirms that the generated S-Boxes exhibit highly efficient nonlinear behaviour and can spread the impact of small changes broadly and unexpectedly. Therefore, the proposed replacement box achieves the SAC standard to a high degree, which enhances the resistance of the proposed algorithm to differential attacks and contributes to raising the overall security level of the cryptographic system.

3.3.2 Bit independence criterion (BIC)

Bit Independence (BIC) is a key metric for evaluating the quality of swap boxes. It measures the degree to which output bits change independently when a single input bit changes. A good swap box is assumed to have a statistically independent probability of each output bit changing, with BIC values approaching the optimal value of 0.5. Table10 shows the BIC test results for swap boxes generated using the proposed chaotic system. Each row represents the S-Box generated. Conversely, the columns are the measurements of the independence of bits between pairs of output bits. The values demonstrate that the values of the BIC fell within the range of 0.474459 and 0.536481, most of the values were evident around the optimal value of 0.5. This even distribution is an indication of a high degree of statistical freedom between the various output bits.

Table 10: Bit Independent Criteria (BIC) of Generated S-Boxes

———	0.509019	0.48561	0.515917	0.515967	0.50661	0.498156	0.502846
0.498901	———	0.528387	0.510285	0.485781	0.506739	0.49805	0.499986
0.503392	0.536481	———	0.491161	0.51643	0.511764	0.512075	0.498289
0.495564	0.496441	0.491319	———	0.502392	0.504088	0.499221	0.507185
0.507202	0.531751	0.474459	0.497291	———	0.494438	0.52263	0.498281
0.491297	0.512251	0.511851	0.517949	0.526399	———	0.499795	0.520658
0.503969	0.480297	0.505696	0.496225	0.501349	0.509484	———	0.521339
0.509004	0.515435	0.517005	0.524443	0.519914	0.529445	0.510041	———

These results indicate that the proposed swap boxes possess strong characteristics for propagating changes and minimizing interdependence between bits, thus enhancing their nonlinearity and increasing the algorithm's resistance to differential and linear attacks. Thus, S-Boxes produced display a high level of compliance with the BIC standard, which justifies their efficiency in the offered encryption architecture and their applicability to security tasks in the resource-constrained environments.

3.3.3 Resistance against differential and linear attacks

The proposed lightweight ARX block cipher is resistant to differential and linear cryptanalysis, using ARX operations, dynamic chaotic S-Boxes and Feistel-inspired diffusion mechanism. Modular addition, rotation and XOR operations in combination add a high level of nonlinearity between plaintext, ciphertext and round keys, reducing the possibility of statistical analysis.

The S-Boxes are dynamically generated by a chaotic system of six dimensions, so that they are used to provide the substitution patterns, which are continuously changing and will not lead to the construction of effective S-box differential or linear approximations. Furthermore, the flipping operation between the two paths increases the diffusion by quickly cascading small changes at the input through the entire structure of the ciphertext.

These properties are corroborated by the experimental results, where the Hamming distance values were close to 50% of the data length, which means that the avalanche effect was strong. Moreover, the values of the SACs were distributed close to zero and the values of the BICs were close to the best value 0.5, indicating that the outputs are very non-linear and statistically independent. The entropy and correlation analyses also indicated that the proposed algorithm is effective to remove statistical relationships between plaintext and ciphertext. As a result, the algorithm proposed in this paper can be resistant to both differential and linear attacks due the use of dynamic chaotic S-Boxes, ARX operations, and Feistel inspired diffusion as well as it has efficient computation complexity, which is suitable for resource-limited environments.

3.4 Performance Evaluation

Encryption and decryption time of a database records set was used as a measure of the computational performance of the proposed algorithm. This was to determine the efficiency of implementation and the applicability of the algorithm to resource constrained applications. Table 11 indicates the encryption and decryption time of different sizes of 12 records.

Table 11: Time Consuming of Encryption and Decryption of Records in Database

Record No.	Encryption time (ms)	Decryption time(ms)
1	0.03775	0.02705
2	0.03884	0.02755
3	0.03606	0.02535
4	0.03338	0.02379
5	0.03575	0.02544
6	0.04030	0.02834
7	0.03352	0.02396
8	0.03525	0.02547
9	0.03513	0.02533
10	0.03842	0.02696
11	0.03616	0.02543
12	0.03658	0.02562

The results show that the encryption time ranged from 0.03775 ms to 0.04030 ms, while the decryption time ranged from 0.02705 ms to 0.02834 ms. It is interesting to know that the decryption time was never the same or more than the encryption time, which is expected in a symmetric algorithm whose structure is lightweight and indicates how well the reverse retrieval operations can be performed compared to the forward encryption operations. The findings also show that the difference in the times of execution of various records is relatively slight, which proves the consistency of the performance and a small effect of the difference in the data contents. This stability indicates the simplicity of the operations used in the proposed algorithm which mainly uses low-computational-cost Addition Rotation XOR (ARX) operations. Overall, the obtained performance outcomes verify that the proposed algorithm can be successfully used in resource-constrained systems including the IoT systems, embedded systems, and databases that need fast encryption and cannot affect the response time. Table 12 is an evaluation of the algorithm in terms of computational performance and the results indicated that the time taken was measured as a function of the total data size. It was therefore determined that the encryption and decryption of data is currently constant and

potentially quasi-linear with increasing data volume, which is a constant and scaled value. As an example, the encryption time on 10 large records of the encryption length of 17,940 bits was about 0.54 ms with the data encryption time of 0.45 ms. But the encryption and decryption time were 3.16 ms and 2.51 ms respectively on 60 records and total data size of 107,607 bits. It is also interesting that the decryption time was always less than the encryption time, which interferes with symmetric illumination encryption algorithms.

Table 12: Encryption and Decryption Time of the Proposed Lightweight ARX Block Cipher for Different Database Sizes

No. of records	Data size (bit)	Encryption time (ms)	Decryption time(ms)
10	17940	0.543067906	0.448048464
20	35868	1.089475311	0.906901651
30	53776	1.549997769	1.27039235
40	71739	2.163233948	1.66781176
50	89647	2.636193055	2.154193682
60	107607	3.156897672	2.510485623

The results indicate that the execution time increases nonlinearly with an increase in the data size as indicated in figure 7. It demonstrates rationality in the algorithm and the possibility to use it to manage the medium and large-sized databases without the need to introduce a new increase in the execution time, which proves its suitability in the resources-limited setting like the Internet of Things (IoT) systems and sensitive databases, where the performance aspect is a significant concern in addition to the degree of security.

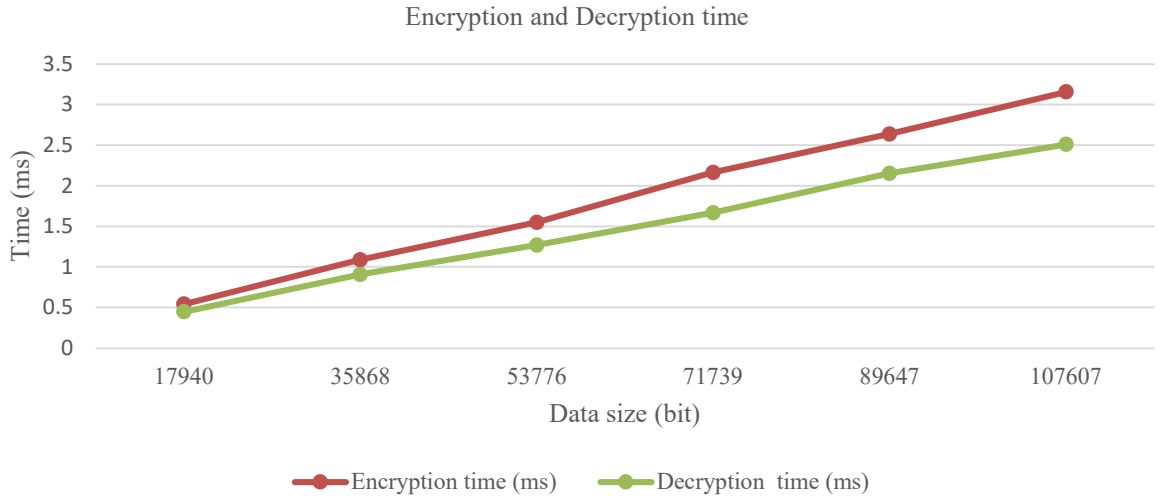


Figure 7: Encryption and Decryption Time with Different Data Sizes

The given algorithm based on the AES algorithm and PRESENT algorithm was run in an identical programming environment, on a computer having an Intel Core i7 processor and 16GB of RAM, on the MATLAB R2024a platform. Table 13 compares the computational speed of the proposed algorithm with AES and PRESNT algorithms.

The block and key size of the algorithms were different, but the comparison was made based on the efficiency in the execution. The findings indicate that the suggested method has the shortest encryption and decryption time with a bigger block and key size and is highly efficient with not lowering the security. Moreover, the critical start up time was not exceeded by competitive standards, which proved the appropriateness of the given algorithm to resource-limited conditions. All algorithms presented in table 13 were implemented and evaluated under the same MATLAB R2024a environment using identical hardware specifications to ensure a fair computational performance comparison.

Table 13: Performance Evaluation of the Proposed Algorithm with AES and PRESNT Algorithms.

Encryption algorithm	Key size in bit	Plain text block size in bit	Encryption time (ms)	Decryption time (ms)	Key schedule (setting) time (ms)
AES	128 bits	128 bits	0.012231	0.00978978	0.00074073
present	80 bits	64 bits	0.009651	0.00772473	0.00068013
Proposed method	256 bits	256 bits	0.007375	0.005903	0.00067339

3.5 Comparative Analysis with Existing Lightweight Cipher Algorithm

To further investigate the proposed lightweight ARX block cipher, a qualitative comparison with several existing lightweight encryption algorithms, such as AES, PRESENT, SPECK and SIMON is done. The comparison is based on the following factors: block size, key size, structural design, dynamic substitution capability, and suitability for resource-constrained environments as shown in table 14.

Table 14. Comparative analysis of the proposed algorithm with other lightweight algorithms.

Algorithm	Structure	Block size	Key size	Dynamic S-Box	ARX operations	Resource-constrained suitability
AES	SPN	128-bit	128/192/256-bit	No	Partial	Moderate
PRESENT	SPN	64-bit	80/128-bit	No	No	High
SIMON	Feistel	64/128-bit	96/128-bit	No	Yes	High
SPECK	ARX	64/128-bit	96/128-bit	No	Yes	High
Proposed Algorithm	Feistel-like ARX	256-bit	256-bit	Yes	Yes	High

The comparison shows that the proposed algorithm is a lightweight algorithm that integrates the benefits of the dynamically generated chaotic S-Boxes which are not frequently used in traditional lightweight ciphers like SPECK and SIMON. Furthermore, the suggested algorithm can be implemented with larger block and key sizes with low computational complexity and efficient execution time applicable to IoT and embedded systems.

4. CONCLUSION

The proposed algorithm can be used to create an efficient trade-off between high security needs and computational efficiency, and it is therefore applicable to the resource-constrained IoT, embedded systems, and sensitive databases. The lightweight block encryption algorithm proposed is designed on the ARX architecture, dynamic replacement box construction using 6D chaotic system, and an extended round key generation mechanism, which is based on the AES architecture and simplified and efficient computation processes. The outcomes of the statistical and security analysis

revealed that the suggested algorithm attains high rates of randomness and dissemination. Entropy values of the cipher-text were almost optimum, and the Hamming distance between the plain-text and cipher-text was close to the theoretical value (around 50%), indicating that it is highly sensitive to even minor variations in input. Moreover, the correlation coefficients between the plaintext and the ciphertext were near to zero proving weak statistical correlation between them. The generated keys were found to be in full adherence with all the available NIST statistical tests at different key lengths at the key level, thus showing great randomness as well as strong resistance against key spoofing attacks. It was also found that the proposed substitution boxes had good cryptographic properties through high SAC and BIC values, which increased their vulnerability to differential and linear attacks. Regarding the performance, the results proved that the proposed algorithm has low encryption and decryption time, high throughput with references to a range of popular lightweight algorithms and has a larger block and key size which contributes to better security. According to these findings, the suggested algorithm is a desirable scientific addition to the sphere of lightweight cryptography, which unites computational simplicity and high-security level. Future studies might extend this paper by running the algorithm on low-power hardware systems, exploring its resistance to side-channel attacks, and incorporating it into bigger security systems to secure information in smart and distributed systems.

CONFLICTS OF INTEREST

The conflict of interest concerning this research is non-existent.

AUTHORS CONTRIBUTIONS

The authors collaboratively worked on the idea of the research, research was conducted, results were analysed, manuscript was written and revised. The authors consented to the final research version.

ACKNOWLEDGMENTS

This work was supported by the Informatics Institute for Postgraduate Studies of Information Technology & Communications University, University of Technology, and Computer Science and Information Technology of the University of Anbar,

REFERENCES

[1] W. Chen, L. Li, and Y. Guo, “DABC: A dynamic ARX-based lightweight block cipher with high diffusion”, *KSII Transactions on Internet and Information System*, Vol. 17, No. 1, 2023, pp. 165-184.

[2] Y. Yang, H. Dong, L. Chen, Z. Li, and C. Xia, “LWARX: Lightweight ARX white-box cipher for satellite communications”, *J. King Saud Univ. Inf. Sci.*, Vol. 36, No. 4, 2024, pp. 102032-102044, /doi.org/10.1016/j.jksuci.2024.102032

[3] B. A. Hameed, “New S-Box Generation Based on Hybrid Two-Dimensional Chaotic Map for Color Image Encryption”, *Int. J. Intell. Eng. Syst.*, Vol. 17, No. 6, 2024, pp.702-716, doi.org/10.22266/ijies2024.1231.54.

[4] S. S. Issa, A. H. Ali, S. Q. Salih, and F. H. Taha, “BH-Sbox: A Substitution Box Generating Algorithm Based on Chaotic Black Hole Algorithm”, *Int. J. Intell. Eng. Syst.*, Vol. 16,No. 6, 2023, pp. 249–260, doi.org/ 10.22266/ijies2023.1231.21.

[5] I. N. Mohammad Shah, E. S. Ismail, F. Samat, and N. Nek Abd Rahman, “Modified generalized feistel network block cipher for the internet of things”, *Symmetry (Basel).*, Vol. 15,No. 4, 2023, pp. 900-916, doi.org/10.3390/sym15040900.

[6] P. K. Gundaram, “Cryptanalysis of Selected ARX-Based Block Ciphers”, *Vietnam J. Comput. Sci.*, Vol. 11, No. 04, 2024, pp. 553–568, doi.org/10.1142/S2196888824500143.

[7] H. Madushan, I. Salam, and J. Alawatugoda, “A review of the nist lightweight cryptography finalists and their fault analyses”, *Electronics*, Nol. 11, no. 24, 2022, pp. 4199-4220, doi.org/10.3390/electronics11244199.

[8] J. Kaur, A. C. Canto, M. M. Kermani, and R. Azarderakhsh, “A comprehensive survey on the implementations, attacks, and countermeasures of the current NIST lightweight cryptography standard”, *arXiv Prepr. arXiv2304.06222*, 2023, doi.org/10.48550/arXiv.2304.06222.

[9] S. Khan, P. A. Ferreira Lopes martins, B. Sousa, and V. Pereira, “A Comprehensive Review on Lightweight Cryptographic Mechanisms for Industrial Internet of Things

Systems", *ACM Comput. Surv.*, Vol. 58, No. 1, 2025, pp. 1–37, doi.org/10.1145/375773.
[10] E. Konstantopoulou, G. Athanasiou, and N. Sklavos, "Review and Analysis of FPGA and ASIC Implementations of NIST Lightweight Cryptography Finalists", *ACM Comput. Surv.*, Vol. 57, No. 10, 2025, pp. 1–35, doi.org/10.1145/372112
[11] F. Thabit, O. Can, S. Alhomdy, G. H. Al-Gaphari, and S. Jagtap, "A novel effective lightweight homomorphic cryptographic algorithm for data security in cloud computing", *Int. J. Intell. networks*, Vol. 3, 2022, pp. 16–30, doi.org/10.1016/j.ijin.2022.04.001
[12] A. Ahmed, M. M. Madboly, and S. K. Guirguis, "Securing signal encryption based on reduced round homomorphic AES", *Int. J. Intell. Eng. Syst*,Vol. 16, No. 3, 2023, pp. 440-454, doi.org/10.22266/ijies2023.0630.35
[13] H. Tian and M. Li, "A Lightweight IoT Data Security Sharing Scheme Based on Attribute-Based Encryption and Blockchain", *Comput. Mater. Contin.*, Vol. 83, No. 3, 2025, pp. 5539-5559, doi.org/10.32604/cmc.2025.060297.
[14] R. S. Mohammed, "Design a Lightweight Authentication Encryption Based on Stream Cipher and Chaotic Maps with Sponge Structure for Internet of Things Applications", *Int. J. Intell. Eng. Syst.*, Vol. 16, No. 1, 2023, pp. 532-547, doi.org/10.22266/ijies2023.0228.46
[15] H. Noura, O. Salman, R. Couturier, and A. Chehab, "Lesca: Lightweight stream cipher algorithm for emerging systems", *Ad Hoc Networks*, Vol. 138, 2023, pp. 102999-103015, doi.org/10.1016/j.adhoc.2022.102999.
[16] M. Pandya, "Performance Evaluation of Hashing Algorithms on Commodity Hardware", *arXiv Prepr. arXiv2407.08284*, 2024, doi.org/10.48550/arXiv.2407.08284
[17] A. Sevin, "Implementation of a Data-Parallel Approach on a Lightweight Hash Function for IoT Devices", *Mathematics*, Vol. 13, No. 5, 2025, pp. 734-759, doi.org/10.3390/math13050734.
[18] N. Kapalova, K. Algazy, and A. Haumen, "Development of a New Lightweight Encryption Algorithm", *Eastern-European J. Enterp. Technol.*, Vol. 123, No. 9, 2023, pp. 6-19, doi.org/10.15587/1729-4061.2023.280055
[19] A. K. Kwala, A. Mishra, and S. Kant, "Boolean semiring key-exchange with BLAKE3 security (BKEX-B3)", *Discov. Comput.*, Vol. 28, No. 1, 2025, pp. 158-181, doi.org/10.1007/s10791-025-09650-x.
[20] F. Yu *et al.*, "Chaos-Based Engineering Applications with a 6D Memristive Multistable Hyperchaotic System and a 2D SF-SIMM Hyperchaotic Map", *Complexity*, Vol. 2021, No. 1, 2021, pp. 6683284-6683305, doi.org/10.1155/2021/6683284.