JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, XXX-XXX (2022)
DOI: 10.6688/JISE.20220X_38(X).00XX

A Robust Lightweight Authenticated Encryption Scheme with Provable Security in IoT Environment

KOUSALYA R[+,1] AND SATHISH KUMAR G A[2]
[1,2]*Department of Electronics and Communication Engineering*
[1,2]*Sri Venkateswara College of Engineering*
Sriperumbudur, Tamilnadu, India
E-mail: [1,+]*kousi@svce.ac.in,* [2]*sathish@svce.ac.in*

The significant development of the Internet of Things (IoT) has allowed businesses and consumers to utilize various resource-constrained devices, including smartphones, connected vehicles, intelligent systems, and services. However, security, interoperability, power/processing capabilities, and availability are the primary challenges of resource-constrained devices that can affect the implementation of an IoT system. This paper proposes an efficient, Lightweight Authenticated Encryption Protocol (LAEP) that uses a one-dimensional (1-D) logistic Chaotic map for secret key generation and a key-dependent S-box for generating confidential and authenticated data. A two-point Diffie-Hellman key exchange algorithm and one-way hash function facilitated the secret key sharing. Furthermore, a novel method of key-dependent S-box is imposed on the existing PRESENT algorithm, which addresses security and authenticity. It achieves 50% of the Strict Avalanche Criterion (SAC) and 85% of non-linearity with 1730 Gate Equivalents (GEs). The computational analysis proved that the proposed scheme consumes less power and one-fourth of computation time, which is better than the other encryption scheme. Furthermore, the results of the AVISPA simulation demonstrate that the LAEP effectively resists the attacks. Additionally, a real-world testbed environment was implemented using the Raspberry Pi 4 Model B, and the experimental findings confirm the robustness of the proposed protocol. As a result, the proposed protocol is ideal for resource-constrained devices.

Keywords: Chaotic map, Data security, Internet of Things (IoT), Lightweight authenticated cipher, PRESENT cipher

1. INTRODUCTION

IoT is a massive network connecting various electronic devices that can exchange data. The IoT industry is assessed to exceed USD 19 trillion in the upcoming future. By 2025, it is predicted that about 100 billion smart components will be in practice across the universe, with an estimated economic worth of more than USD 11000 billion. Most devices that will constitute the IoT environment are resource-constrained, as depicted in Fig. 1 [1].

IoT-connected devices possess limited resources and have no insights into security to handle additional functionalities and protocols. Since data is sent over an insecure channel, a third party can intercept and use it to carry out attacks [2], like data alteration, delivery delays, information impersonation, and data exchange disruption, it is also highly mandatory to offer data authenticity [3]. Consequently, a reliable and secure network is required to safeguard the data flow. Therefore, the communication platform of the IoT environment is unsafe, and this issue must be addressed with an appropriate protocol/algorithm that

ensures the data's confidentiality and authenticity. The security challenges [4, 5] encountered in IoT have been described in Fig. 2.

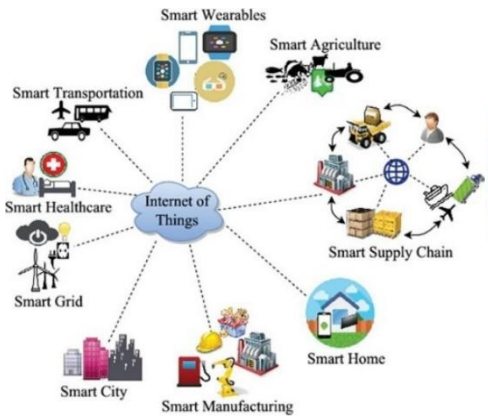

Fig. 1. Application Domain of IoT

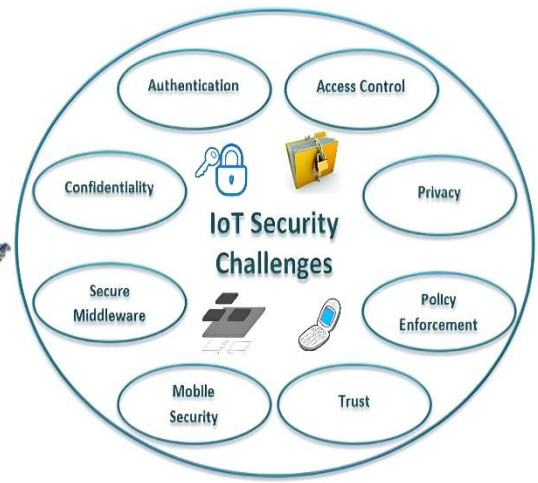

Fig. 2. IoT Security Challenges

Cryptography is an advanced data/ information protection method that converts intelligible data into non-comprehensible information that can be decoded and processed by those for whom it is intended. Cryptographic algorithms are classified into Symmetric and Asymmetric Key Cryptography. The symmetric key cryptographic algorithm uses a similar secret key for both enciphering and deciphering. In contrast, the asymmetric cryptographic algorithm uses a public key for encipherment and a secret key for decipherment. In symmetric key cryptography, the secret key is shared between entities using various key exchange algorithms such as Diffie-Hellman [6], Elliptic Curve key exchange, etc. Unfortunately, traditional cryptographic algorithms are unsuitable for devices with limited resources. A new branch of cryptography is Lightweight cryptography, specifically designed to enhance security in ubiquitous computing applications. Since smart devices are resource-limited, Lightweight cipher algorithms secure data flow. Similarly, the receiver must validate the authentication of the sender. The hashing function is a simple solution to this problem. The authentication tag is generated using a Lightweight hashing algorithm, and a Lightweight authenticated scheme [7] that will be verified by the receiver. An efficient Lightweight authenticated cipher scheme based on the encrypt-then-MAC paradigm was presented to address the abovementioned issues.

The chaos-based key generation technique was highly recommended as Chaotic systems [8, 9] are sensitive to initial conditions and chaotic parameters. Enormous ways for achieving SPN properties in any cipher techniques created using chaos combined with cryptography. Various Chaotic map theories were used to produce the private keys for the enciphering and authentication process.

1.1 Motivation and Contribution

Researchers have proposed Lightweight, authenticated cipher schemes for transferring data over an unsecured Internet environment. However, existing approaches are

exposed to numerous security concerns, including impersonation, alteration, replay, and other potential threats. Due to the complex structure of cryptographic procedures, traditional authentication systems demand enormous hardware components. As a result, ensuring improved security and performance in protocol design remains significant for addressing IoT concerns.

A novel technique has been proposed to address the earlier issues for secure data transmission over a public channel. Thus, the primary focus of this paper outlines the key contributions as follows:

•Proposed chaotic-based secret key generation and two-point mutual key agreement mechanism for data exchanges. The primary function of a chaotic map is to improve the randomness of the secret key.

•Proposed a novel, secure, Lightweight, authenticated algorithm to generate ciphertext and authentication data. A new modified PRESENT algorithm is robust and utilizes minimum hardware.

•Performed a formal and informal security study to ensure resistance of the proposed method to critical security issues such as alteration, spoofing, replay, Man-in-the-Middle (MIMA), related key, and known-key security.

•Various metrics such as Strict Avalanche Criterion (SAC), Non-linearity, Hardware cost, Balanced output, Computation time, and Power consumption are measured to analyze the performance of the suggested method.

The summary of this work structure follows: Section 2 explains literature surveys on Lightweight authenticated schemes. Section 3 defines the preliminaries of the security scheme, chaos map, and Lightweight hashing function. Section 4 explains the new Lightweight authenticated cipher scheme for the IoT environment. Section 5 provides a brief security study of the proposed method and investigates its performance metrics. Finally, the conclusion and future scope are explained in Section 6.

2. RELATED WORK

In the following section, a comprehensive overview provides a review on the existing approaches related to security threats and different cryptographic primitives.

Ahmed Aziz et al. [10] suggested Lightweight Secure Scheme (LSS) to resist against a Chosen-Plaintext Attack (CPA) by generating secret compressed samples. By reducing network complexity, this approach results in a prolonged network lifetime. However, it is liable to impersonation, modification, and replay attacks.

Manish Gupta et al. [11] suggested a hybrid chaotic map to generate the random session key for each image encryption. Furthermore, a crossover operator is utilized to increase confusion and diffusion. The proposed picture cryptographic technique's viability is evaluated based on its resistance to differential attacks, statistical attacks, and susceptibility to secret keys. Moreover, this approach could not address the authenticity between two entities during the key exchange phase.

Qiming Zheng et al. [12] have presented a Lightweight authenticated cipher approach with correlated data to provide secrecy and integrity. The keystream generator in this approach is the chaos-based S-box coupled map framework. The performance outcome suggests the Lightweight authenticated cipher approach will guarantee secrecy and integrity

with appropriate efficiency in the railway IoT cloud environment.

Tabassum Ara et al. [13] presented a dynamic key-dependent S-box algorithm for resource-constrained devices. It generates 16 distinct S-boxes with strong security attributes, balancing, avalanche effect, etc. Furthermore, sixteen S-boxes consume the excess memory in resource-limited devices. Abdulrazzaq H et al. [14] proposed a detailed examination of Lightweight symmetric-key cryptography (block ciphering and hashing algorithm). This classification shows the contrast and difference between ciphers in favor of several characteristics. This accurately defines symmetric-key cryptography (block ciphering and hashing algorithm) for limited resource devices through cipher classification and hardware implementations.

Bogdanov et al. [15] proposed a new block cipher Lightweight algorithm: PRESENT. This algorithm mainly addresses resource-limited devices like RFID tags, smart devices, etc. Furthermore, they have concluded that the algorithm above requires 1570 GEs for implementation. However, this algorithm could not provide a performance analysis of the S-box. Chaudhary, N et al. [16] presented a chaos-based image encryption and block cipher techniques for image encryption. The performance metrics such as PSNR, NPCR, histogram and computation time are measured.

Bhaskar et al. [17] presented a lightweight encryption technique for images using chaotic maps and diffusion circuits. This scheme utilizes simple bit-wise operation that reduces computation overhead. Yasmin N et al. [18] presented a modified lightweight cryptography scheme utilizing bit-slice substitution to enhance security features. Furthermore, this scheme takes high computation time. Abutaha et al. [19] proposed a secure lightweight cryptosystem for IoT devices. The robustness of the encryption scheme is affected by one-bit change.

Zhiying Tang et al. [20] introduced an improved S-box that solves the issues in the original PRESENT S-box. The authors have proved that the suggested method is extremely resistant to differential and linear attacks. They also came to the conclusion of various aspects for improvement, like optimizing the algorithm and searching for a stronger dynamic S-box. Prathiba et al. [21] proposed a lightweight, secure S-box architecture for IoT applications. The S-box architecture enables sub-pipelining and reduces gate count. The proposed S-box was resistant to linear and differential cryptanalysis and utilized less hardware complexity compared to GF (2^4).

Verma. S et al. [22] recommended the application "NCRYPT," which aims to preserve data on Android so that unauthorized users cannot access it. The proposed application uses the Lightweight scheme: Hummingbird-2 and provides safe data storage. Furthermore, it necessitates a greater number of Gate Equivalents (GEs).

Majid Khan et al. [23] proposed a chaotic-based Lightweight S-box to enhance the characteristics of cryptographic primitives. Further, they claimed that their scheme offers remarkable randomness behavior and is resistant to differential and linear attacks. However, the hardware cost is increased because it has 24 distinct S-boxes.

Jebri. S et al. [24] proposed a lightweight secure IoT architecture to provide the authentication and anonymity for IoT devices. This scheme addresses vulnerabilities in sensitive IoT applications. However, the computation time and power consumption is higher than the proposed scheme.

According to the existing literature, some approaches have offered a remarkable level of security at the expense of hardware costs. On the contrary, Bogdanov et al. utilized a

low hardware cost with a limited level of security [15]. Henceforth, there is a trade-off between security and hardware cost. To overcome the current constraints, a new Lightweight authenticated cipher scheme has been proposed to offer better security with low hardware costs.

3. PRELIMINARIES

3.1 Chaotic Map

Chaos theory [24] involves the study of dynamical systems that depend on preliminary conditions. It demonstrates the efficacy of the sensitive design can be advantageous in cryptographic processes. Dynamical systems are simple equations that vary according to time.

Logistic map

A degree 2 polynomial mapping is a classic example of how complicated, chaotic behavior can emerge from simple dynamical equations. The logistic map's [25] simplicity makes it a very attractive starting point for investigating the concept of chaos. It is a one-dimensional and iterative map stated mathematically as:

$$y_{n+1} = r(1 - y_n) \tag{1}$$

where y_n represents a variable ranging from 0 to 1, representing the proportion of the current to the most significant population feasible. The constant parameter r defines the interval [0,4].

Tent Map

The tent map [26] is defined as

$$f_\mu(x_n) = \begin{cases} \mu(x_n), & x < 0.5 \\ \mu(1 - x_n), & x \geq 0.5 \end{cases} \tag{2}$$

where μ represents real and positive value and f_μ maps the interval [0,1] defining a discrete-time dynamic system on it.

The significance of folding the unit interval in half, and then stretching the resulting interval [0,0.5] back to [0,1] is interpreted as the efficacy of the function f_μ when the parameter $\mu = 2$. As the operation is iterated, starting from point x_0 in the interval, the process described above causes it to assume successive positions, generating a sequence x_n in [0,1].

3.2 PRESENT Algorithm

The PRESENT algorithm [15] is composed of Substitution and Permutation blocks. An SPN-based algorithm takes the same secret key at transmission and reception. This algorithm utilizes small bus sizes in devices with minimal-sized hardware, resulting in adequate resource utilization. S-box plays a pivotal role in enhancing security in an SPN network. The keys that vary every round can modify the S-box, defined as a key-dependent S-box [27 - 30].

3.3 DAVIES-MEYER construction

Merkle and Damgard have given the theoretical foundation of the compression function. The compression function 'H' processes the fixed-length input and involves a chaining variable and a message extract.

The function 'H' produces fixed-length output. For each message (M), the Davies–Meyer compression function [26] generates a key for the block cipher. It also feeds ciphertext to the past value of a hashing function (H_i). Furthermore, the hashing value (H_i) is produced by XORing the ciphertext with the past hashing value (H_{i-1}).

$$H_i = E(H_{i-1}, M) \oplus H_{i-1} \tag{3}$$

where E represents the encryption process using either PRESENT-80 or PRESENT-128 [15], both of which ensure a 64-bit security level. In each iteration, the compression function processes a 64-bit chaining variable along with an 80-bit message-oriented input. Fig. 3 depicts the layout of the Davies-Meyer construction.

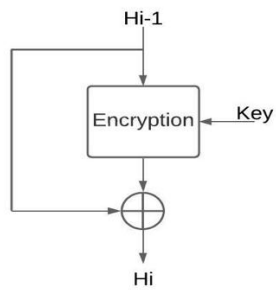

Fig. 3. Davies-Meyer Construction

Table 1. Notations used in the Scheme

Parameter	Description
a, b, c, d	Random point chosen from Chaotic map
i	Iteration
s, r	Sender's and Receiver's Private key
G (g_1, g_2)	Points calculated at the sender side
F (f_1, f_2)	Points calculated at the receiving end
Pub_s	Sender's Public key
Pub_r	Receiver's Public key
K_{ss}	Secret key derived at the sender side
K_{sr}	Secret key derived at the reception end
K_s	Secret key shared between two entities
T_1, T_2, T_3, T_4	Time stamp
Δ_T	Threshold time
Enc_{Ks}	Encryption using K_s
Dec_{Ks}	Decryption using K_s
C_1	Cipher text
C_t	Authentication tag
$\|\|$	Concatenation
$\oplus$	XOR operation

4. PROPOSED LAEP SCHEME

The following section presents a lightweight authenticated cipher technique for secure transmission over an IoT environment prone to security threats. Randomization of the chosen key improved data secrecy, highlighting how the choice of keys is vital in ensuring reliable data communication. A chaos cryptographic component is implemented iteratively to generate the private keys stream, while the public keys are created using the double point Diffie-Hellman Key Exchange algorithm and shared between the two entities over the insecure channel. When two entities receive the public keys, they compute the shared secret keys for the enciphering and deciphering processes. Table 1 explains the notations employed in the overall structure.

A novel lightweight cipher approach is realized to generate ciphertext and authentication code that might have been communicated over a public channel. The suggested system is validated using various cryptographic strength primitives. This approach has three different phases as follows: (i) Key Generation and Key Exchange Phase, (ii) Encryption and Authentication Phase, and (iii) Decryption and Verification Phase. Fig. 4 depicts the overview of the proposed technique.

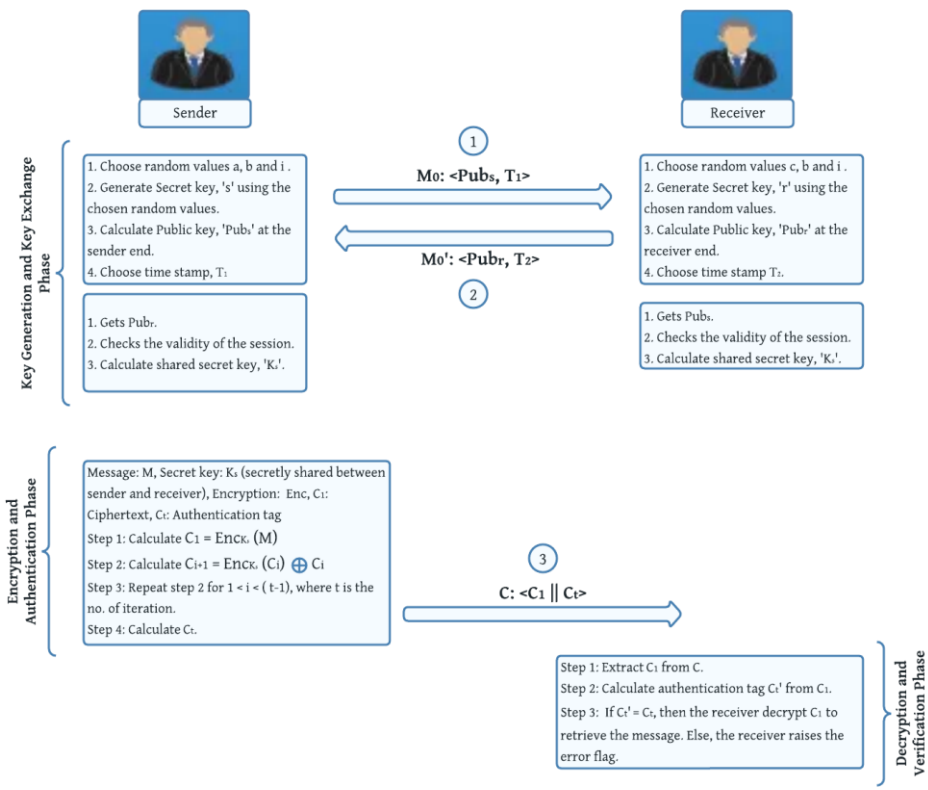

Fig. 4. Overview of LAEP

4.1 Key Generation and Key Exchange Phase

Based on the logistic and tent maps, a random private key is produced in this methodology. Initial random values 'a' and 'b' are selected from the map at the sender end with iteration count 'i' of the logistic map using a tent map. Consequently, the iteration count 'i' is exploited to produce random secret keys 's'. The same process is followed at the receiver end with random values 'c' and 'd' to produce secret value 'r'. The 1-D logistic chaotic map generates the private key with a good randomness level.

The Public keys Pub_s (at the sender site) and Pub_r (at the reception site) are derived by realizing the Diffie-Hellman Key exchange algorithm [10] on the random points chosen from the chaos map. The secret key 'K_s' shared between the two entities is explained below:

(a) Computation of private and public keys at two entities are explained in Algorithm 1 and 2:

Algorithm 1: **Sender end**

Step 1: Choose *'a'* and *'b'* parameters from chaos map.
Step 2: Choose a random prime number *'P', G_1 and G_2* within the bounds *(1, i).*
Step 3: Generate secret key *'s'*.
Step 4: Compute g_1 and g_2 as follows:

$$g_1 = G_1{}^s \ mod\ P \quad (4)$$

$$g_2 = G_2{}^s \bmod P \quad (5)$$

$$Pub_s = (g_1, g_2) \quad (6)$$

Step 5: Choose time stamp T_1.

Algorithm 2: **Receiver end**

Step 1: Choose *'c'* and *'d'* parameters from chaos map.
Step 2: Choose a random prime number *'P', G_1 and G_2* within the bounds *(1, i).*
Step 3: Generate secret key *'r'*.
Step 4: Compute f_1 and f_2 as follows:

$$f_1 = G_1{}^r \ mod\ P \quad (7)$$

$$f_2 = G_2{}^r \bmod P \quad (8)$$

$$Pub_r = (f_1, f_2) \quad (9)$$

Step 5: Choose time stamp T_2.

(b) Once the public key is calculated, the sender chooses the Time stamp T_1 and shares M_0: <Pub_s, T_1> through the public channel. Similarly, the receiver chooses the T_2 and shares M_0': <Pub_r, T_2> with the sender.

(c) Once the session is valid, the sender computes $A_1 = h(a||b)$, $B_1 = h(g_1 \,||\, f_1) \oplus A_1$ and sends M_1: <B_1, T_3>. Meanwhile, the receiver verifies the validity of the timestamp and computes $A_2 = h(c||d)$, $B_2 = h(g_2 \,||\, f_2) \oplus A_2$ and sends M_1': <B_2, T_4>.

(d) On receiving M_1 and M_1', the sender and receiver computes the shared secret key as follows:
Step 1: The sender checks the validity of the timestamp T_3 by ensuring that $T_{s3} - T_3 < \Delta T$. If this condition is satisfied, the sender computes $A_2' = h(g_2 \,||\, f_2) \oplus B_2$ and the shared secret key is determined as follows:

$$K_{ss} = h(f_1{}^{s}) \oplus h(f_2{}^{s}) \oplus A_1 \oplus A_2') \quad (10)$$

$$K_{ss} = h((G_1{}^{r} \bmod P)^{s}) \oplus h((G_2{}^{r} \bmod P)^{s}) \oplus A_1 \oplus A_2')$$

$$K_{ss} = h(G_1{}^{rs} \bmod P) \oplus h(G_2{}^{rs} \bmod P) \oplus A_1 \oplus A_2') \quad (11)$$

Step 2: The receiver checks the validity of the timestamp T_4 by ensuring that $T_{s4} - T_4 < \Delta T$. If this condition is satisfied, the receiver computes $A_1' = h(g_1 \,||\, f_1) \oplus B_1$ and the shared secret key is as follows:

$$K_{ss} = h(g_1{}^{r}) \oplus h(g_2{}^{r}) \oplus A_1' \oplus A_2) \quad (12)$$

$$K_{ss} = h((G_1{}^{s} \bmod P)^{r}) \oplus h((G_2{}^{s} \bmod P)^{r}) \oplus A_1' \oplus A_2) \quad (13)$$

Since K_{ss} and K_{sr} are equal, it is concluded that the shared key, K_s, is shared between two users across an unsafe channel.

4.2 Encryption and Authentication Phase

A novel encryption mechanism has been developed to strengthen the secrecy and authenticity of data transported in an IoT environment. The procedure of changing plaintext to ciphertext is termed encryption. Since an IoT context is a cluster of resource-limited devices, the objective mentioned above is addressed by a modified PRESENT Lightweight algorithm. S-box is the pivotal component of cryptographic algorithms. In conventional algorithms, S-box is a static object unchanged by input or key. In this proposed cipher scheme, the key derived in every round is utilized to modify the S-box. The modified S-box is explained in Algorithm 3:

Algorithm 3: **Proposed Key-dependent S-box**

Input: Secret key, K_s is 80-bit

Step 1: Compute 64-bit AddRoundKey,

$k = k_{63}k_{62}k_{61}k_{60}k_{59}k_{58} \ldots\ldots\ldots k_3k_2k_1k_0$ using a key scheduling method.

Step 2: Compute the 4-bit key,

$K = k_{63}k_{62}k_{61}k_{60} \oplus k_{59}k_{58}k_{57}k_{56} \oplus \ldots\ldots\ldots \oplus k_3k_2k_1k_0$

Step 3: The 4-bit key, *K*, derived from step 2, is taken for constructing a unique S-box by circularly shifting each nibble towards the left and right sides.

Step 4: If the count of 0s and 1s in *'K'* is the same, the existing S-box is exploited and represented as an S_1 box. If the count of 0s exceeds the count of 1s, then the conventional S-box is circularly shifted by 3 levels towards the left and represented as *'S_2'*. If the count of 1s exceeds the count of 0s, then the conventional S-box is circularly shifted by 3 levels right and represented as *'S_3'*. For all 0s and 1s in *'K'*, the conventional S-box is circularly shifted by 4 levels towards left and right, respectively. It is represented as *'S_4'* and *'S_5'*. The recommended S-box was deployed in the PRESENT Lightweight algorithm, and the ciphertext C_1 was created.

Algorithm 4: **Generation of Encryption data and authentication tag**

Input: Message (M), Shared secret key (K_s) and iteration count (t)

Step 1: Compute the ciphertext C_1 as follows:

$C_1 = \text{Enc}(M, K_s)$

Where, Enc – Encryption algorithm based on Key dependent S-box.

Step 2: Compute the authentication tag C_t as follows:

$C_{i+1} = \text{Hash}(C_i, K_s)$

Where $1 \le i \le (t\text{-}1)$, t = 80 (number of iterations).

Step 3: $C_1 \,||\, C_t$ is transmitted over an unsecured channel.

Algorithm 4 explains the generation of cipher text and authentication tag. In this phase, pairs of 64-bit data named C_1 and C_t are created. The primary data C_1 is the ciphertext and the other, C_t is the authentication value generated using a Lightweight Hash function. Here, DM-PRESENT is preferred for hashing operations on ciphertext. Fig. 5 shows the structure of the encryption and authentication phase. Another advantage of the suggested methodology is that the S-box chosen in every round is decided by the 4-bit key, 'K' derived from the 64-bits around the key. Now, ciphertext C_1 and authentication data C_t are concatenated and transferred through an unsecured channel. Table 2 elucidates the structure of the S-box based on the 4-bit key derived from 64-bit AddRoundKey.

Table 2. Modified S-Box

S-box	K	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S_1	0011,0101,0011,0110, 1001, 1010, 1111	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2
S_2	0001,0010,0100,1000	6	B	9	0	A	D	3	E	F	8	4	7	1	2	C	5
S_3	0111,1110	7	1	2	C	5	6	B	9	0	A	D	3	E	F	8	4
S_4	0000	B	9	0	A	D	3	E	F	8	4	7	1	2	C	5	6
S_5	1111	4	7	1	2	C	5	6	B	9	0	A	D	3	E	F	8

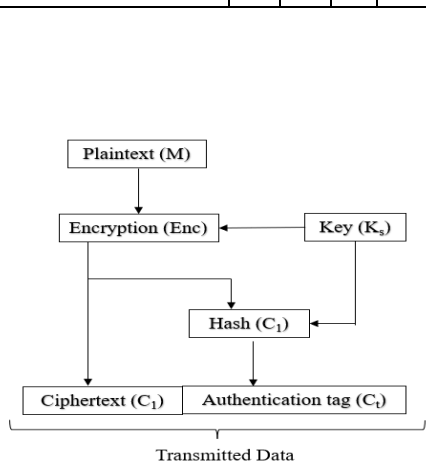

Fig. 5. Generation of ciphertext and authentication tag

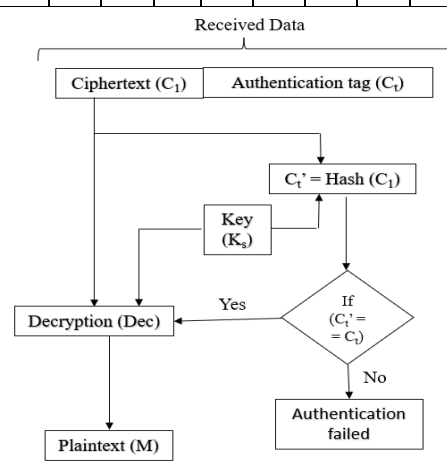

Fig. 6. Decryption and Verification Phase

Algorithm 5: **Verification Phase**

Input: $C_1 \,||\, C_t$, *Shared secret key* (K_s) *and iteration count (t)*

Step 1: Extract C_1 from $C_1 \,||\, C_t$ and Compute the authentication tag as follows:

$$C'_{i+1} = \text{Hash}(C_i, K_s)$$

Where, $1 \leq i \leq (t-1)$, $t = 80$ (number of iterations).

Step 2: Check the authentication tag C_t' and C_t are equal.

Step 3: If step 2 is true, compute the plaintext from C_1 as follows:

$$M = \text{Dec}(C_1, K_s)$$

$$M = \text{Dec}((\text{Enc}(M, K_s), K_s)$$

4.2 Decryption and Verification Phase

The authentication data was extricated from the data stream during this phase. To authenticate the sender's identity, the authentication tag C_t' was generated at the receiver end by applying a Lightweight hashing function to ciphertext C_1 and comparing C_t' to C_t.

If equal, the receiver would decipher the ciphertext C_1 using K_s; Otherwise, the receiver returns an invalid sign $\perp$ and discards the received content. Algorithm 5 elucidates the steps of decipherment phase. Fig. 6 briefs the structure of decryption and verification phase.

5. RESULTS AND DISCUSSION

This section evaluates the hardware cost and performance of the proposed scheme by conducting simulations and comparing its performance with several established schemes.

5.1 Informal Security Analysis

In the context of IoT, ensuring security is a fundamental necessity for the smooth flow of data. Strength is analyzed against numerous security threats. The simulation experiment was implemented in a Python environment. Table 3 illustrates that the two entities securely exchanged the two-point secret key (4553, 11225). Subsequently, an 80-bit shared secret key, denoted as K_s, was derived using the MD5 hashing algorithm

Table 3. Public and Shared Secret Keys using Two-Point Diffie-Hellman Algorithm

Parameter	At the sender's end	At the receiver's end
Random prime number, P, G_1, G_2	13903, 21859, 21221	13903, 21859, 21221
Derived secret key	7902741264313042902 55630	181677149734999388 720191
Public key, *Pub*	(351, 6961)	(8553, 7841)
80-bit Shared secret key, K_s	2a8a36ae719e54844aac	2a8a36ae719e54844aac

An 80-bit key enciphered the 64-bit plaintext shared between two entities. The proposed S-boxes were deployed to generate 64-bit ciphertext and 64-bit authentication tags. The ciphertext and authentication data were concatenated and communicated over uncertain channels. Assuming the data's authenticity was valid at reception, it implies that the ciphertext was deciphered using a shared secret key, K_s. Otherwise, the error flag was raised. Computation overhead was avoided at the receiver end since the deciphering process was carried out for the authenticated data. The sample input-output of the authentication and verification phase is shown below:

At the sender end:
Plain text, M (64-bit): abcd1234ecdc2345 *Shared secret key, K_s (80-bit):* 2a8a36ae719e54844aac *Enc (M, K_s), C_1 (64-bit):* 180e6943a609dabe *Hash (C_1), C_t (64-bit):* a5b40fc1ea75841f *Transmitted Data, C_1||C_t (128-bit):* 180e6943a609dabea5b40fc1ea75841f
At the receiver end:
Received Data, C_1||C_t: 180e6943a609dabea5b40fc1ea75841f *Extract C_1 (64-bit):* 180e6943a609dabe *Extract C_t (64-bit):* a5b40fc1ea75841f *Shared secret key, K_s (80-bit):* 2a8a36ae719e54844aac *Hash (C_1), C_t' (64-bit):* a5b40fc1ea75841f If C_t' and C_t are equal, then plaintext is computed from ciphertext C_1. *Dec (C_1, K_s), M:* abcd1234ecdc2345

The proposed design evaluates the stability of security against several attacks, as explained below:

5.1.1 Impersonation attack

An attacker may attempt to impersonate a trustworthy sender in an impersonation attack. In this event, the authentication data C_t is exceptional for each data and is validated at the reception end before decryption. Also, the authentication data is generated using K_s, which is securely exchanged between two parties. Therefore, it is tough for an unauthorized party to make them a legitimate user.

5.1.2 Man-in-the-Middle attack

In the proposed protocol, the participating entities authenticate each other using shared secret information. For an adversary to execute a Man-in-the-Middle attack (MIMA) between these entities, they would need access to the secret values a, b, c, and d to compute A_1 and A_2. However, without knowledge of these secret values, the adversary cannot successfully execute MIMA. Therefore, the proposed protocol is resistant to such attacks.

5.1.3 Replay attack

In a replay attack, an adversary resends the ciphertext in a public network. Based on the suggested scheme, the session is validated using the timestamp. If the message is received beyond the time limit ΔT, then the receiver discards the received information immediately. Therefore, our approach does not facilitate a replay.

5.1.4 Known-key security

An advantage of using known-key security is that in the event of session key leakage, it does not compromise past or future session keys. In this approach, the chaotic theory is employed for generating random private keys ‘s’ and ‘r’. Each session generates a new private key, which is secretly exchanged between two entities. Furthermore, the session keys in each session are created using a one-way hash function. Thus, the proposed design attains the property of known-key secrecy.

5.1.5 Related-key attack

In a related-key attack, an adversary utilizes the ciphertext to gather the mathematical connection between the keys. The private key is created using a random point in a chaotic map. Since a different random private key is generated in each session, an unauthorized party cannot decipher the ciphertext.

5.1.6 Modification attack

An adversary may try to revise the transmitted information in a modification attack. Here, plaintext is enciphered using a derived secret key unknown to the attackers. Meanwhile, the authentication data appended with the ciphertext data may improve the original message's integrity. Thus, an unauthorized party will have a difficult time launching this attack.

5.1.7 Password Guessing attack

Suppose an intruder intercepts s and r exchanged between the sender and receiver.

Additionally, the adversary gathers all the messages M_0, M_0', M_1 and M_1'. Despite having access to these messages, the adversary cannot extract any information about a, b, c or d . Consequently, the proposed protocol demonstrates resilience against this type of attack.

5.1.8 Forward Secrecy

In this approach, the secret key $K_{ss} = h({G_1}^{rs} \, mod \, P) \oplus h({G_2}^{rs} \, mod \, P) \oplus A_1' \oplus A_2)$ is established between the sender and receiver. Even if an adversary manages to obtain the values of s and r from the sender and receiver, the session key from a previous session remains secure due to the computational complexity of solving chaotic map problems. Hence, the proposed protocol ensures perfect forward security.

5.2 Performance Analysis

In this section, the proposed design's efficiency was validated using various parameters like non-linearity, SAC, implementation cost, and balanced output. Considering computational complexity, computation time, and power consumption for devices with restricted resources is necessary. Each primitive is discussed as follows:

5.2.1 Non-linearity analysis

In cryptographic technique, the S-box is the only non-linear component. The capacity of S-box to evade differential and linear cryptanalysis is validated by its non-linearity value. The higher the non-linearity, the more secure the data. The non-linearity of an n-bit Boolean function b is calculated as follows:

$$NL(b) = \frac{1}{2}[2^n - (|WS_b(f)|)] \quad (14)$$

Where $WS_b(f)$ Walsh transform of Boolean function b and calculated as follows

$$WS_b(f) = \sum_{k\in(0,1)^n}(-1)^{b(k)\oplus k\cdot f} \quad (15)$$

Where $k.f = (k_1 \oplus f_1) + \cdots\cdots + (k_n \oplus f_n)$ is a bitwise of XOR operation.

It is noticed that the average report of non-linearity for [21] is higher than the suggested method, but the hardware requirement is too high as it takes more memory in resource-limited devices. This result provides better security with minimum hardware cost. The minimum, maximum, and average values of the confusion property for various S-boxes are shown in Table 4. Based on observation, the proposed scheme provides the minimum non-linearity value of 2.3, which is higher than previous studies [20 - 23].

Table 4. Evaluation of Non-Linearity

Non-Linearity	[20]	[21]	[22]	[23]	LAEP
Minimum	2	4	2	4	2.3
Maximum	4	4	4	4	4
Average	3	4	3	4	3.4

5.2.2 SAC analysis

Tavares and Webster presented the metric to authorize the strength of good S-boxes. To achieve a good avalanche value, altering one bit in input leads to 50% of the change in output bits. Thus, 50% avalanche value is expected to reduce any relationship between inputs and outputs and certify the privacy of information. Any value close to 0.5 is

consistently regarded as deserving. Table 5 briefs the minimum, maximum and average values of SAC for various algorithms. In particular, the proposed scheme provides an average SAC value of 0.5 which is higher than that found in other studies [20 - 23]. Table V concludes that the proposed methodology provides good security concerns.

Table 5. Validation of SAC

SAC	[20]	[21]	[22]	[23]	LAEP
Minimum	0.125	0.25	0.325	0.372	0.375
Maximum	0.875	0.75	0.75	0.625	0.688
Average	0.468	0.461	0.461	0.492	0.5

5.2.3 Balanced Output

The balanced output is a pivotal metric for validating the strength of the S-box. To safeguard against unauthorized decryption attempts and ensure the security of the encrypted data, it is imperative to maintain a balanced distribution of 0s and 1s in the ciphertext. The proposed S-box produces a balanced output by comparing this metric with existing methods.

5.2.4 Hardware Cost

The methodology generates the key dependent S-box using the XORing of AddRoundKey. The existing S-box is modified based on the 4-bit key derived from AddRoundKey in every round. Since the single S-box is modified and utilized in every round, the proposed scheme takes the additional cost required for XOR operation alone. Furthermore, GEs, equivalent to 2-input NAND gates, will validate the hardware/implementation costs, and a similar cipher technique is utilized to create ciphertext and authentication data. Hence, this scheme offers good security with minimum hardware cost.

Table 6. Definition and Hardware Cost of Lightweight Encryption Scheme

Parameter	Definition	Area (μm^2)
A_{KS}	Key schedule	492.307
A_{XOR}	XOR operation	106.56
A_{sbox}	S-box operation	3885.232
A_{ARK}	Addround Key	1703.12
A_{MUX4x1}	4 x 1 Multiplexer operation	66.53
A_{MUX8x1}	8 x 1 Multiplexer operation	153.014
$A_{MUX16x1}$	16 x 1 Multiplexer operation	352.598
$A_{counter}$	32-bit counter	3442.824

Table 7. Calculation of Hardware utilization of various Lightweight Encryption Scheme

Parameters	Total area occupied in Lightweight encryption scheme	Number of GEs	Area (μm^2)
[20]	$A_{KS} + 8A_{sbox} + 4A_{XOR} + A_{MUX8x1} + A_{ARK} + A_{counter}$	4709	40820.16
[21]	$A_{KS} + 16A_{sbox} + 4A_{XOR} + A_{MUX16x1} + A_{ARK} + A_{counter}$	1486	72101.6
[22]	$A_{KS} + 4A_{sbox} + A_{XOR} + A_{MUX4x1} + A_{ARK} + A_{counter}$	3220	25192.75
[23]	$A_{KS} + 24A_{sbox} + A_{XOR} + A_{MUX16x1} + A_{ARK} + A_{counter}$	11884	102830.86
LAEP	$A_{KS} + A_{sbox} + 4A_{XOR} + A_{ARK} + A_{counter}$	1730	13470.52

Table 6 depicts the definition and hardware cost of various blocks in the Lightweight encryption scheme. Table 7 evidences that [20 - 23] utilizes additional S – boxes and multiplexer blocks for enhancing the level of security.

5.2.5 Computation time

Computation time is a significant metric for evaluating the time required to perform encryption/decryption in any IoT-connected devices. The computation time is likely to be less without compromising security aspects. As the system complexity increases, the time taken to compute the algorithm increases.

Table 8. Definition and Computation time of Lightweight Encryption Scheme

Parameter	Definition	Computation time (ns)
T_{Enc}	Encryption Phase	500.2
T_{XOR}	XOR operation	0.581
T_{MUX4x1}	4 x 1 Multiplexer operation	843.2
T_{MUX8x1}	8 x 1 Multiplexer operation	1232.1
$T_{MUX16x1}$	16 x 1 Multiplexer operation	2087.2

Table 8 describes the computation time of each block used in the Lightweight encryption scheme. Table 8 shows that the time taken to compute S-box is higher than the other components. The S-box component appreciates the level of security. The number of S-boxes used in [20 - 23] are higher and a Multiplexer block selects the S-box in every round. Moreover, the proposed system generates the new S-boxes from the existing S-box at the cost of 4-bit XOR operations. Table 9 shows that the suggested scheme takes less computation time than [20 - 24]. Henceforth, the proposed scheme provides good security with minimum computation time.

Table 9. Calculation of Computation time for various Lightweight Encryption Schemes

Parameters	Total computation time of encryption scheme	Computation time (ns)
[20]	$T_{Enc} + T_{XOR} + T_{MUX8x1}$	1732.85
[21]	$T_{Enc} + T_{XOR} + T_{MUX16x1}$	2587.96
[22]	$T_{Enc} + T_{XOR} + T_{MUX4x1}$	1343.99
[23]	$T_{Enc} + T_{XOR} + T_{MUX16x1}$	2587.96
[24]	$2T_{Enc}$	1000.4
LAEP	$T_{Enc} + T_{XOR}$	500.785

5.2.6 Power consumption

Power consumption is an important metric for analyzing the power consumed by any resource-constrained devices connected in IoT environments. Mostly power consumption is dependent on computation time. Table 10 depicts the power consumption of encryption, 4-bit XOR and Multiplexer. The simulation results are from the Cadence Virtuoso EDA tool with 90nm technology.

Table 10. Definition and Power consumption of Lightweight Encryption Scheme

Parameter	Definition	Power consumption (µW)
P_{Enc}	Encryption Phase	8515.32
P_{XOR}	XOR operation	144.86
P_{MUX4x1}	4 x 1 Multiplexer operation	3546.86
P_{MUX8x1}	8 x 1 Multiplexer operation	11369.38
$P_{MUX16x1}$	16 x 1 Multiplexer operation	24121.39

Table 11. Calculation of Power consumption for various Schemes

Parameters	Total Power consumption of encryption scheme	Power consumption (uW)
[20]	$P_{Enc} + P_{XOR} + P_{MUX8x1}$	20029.6
[21]	$P_{Enc} + P_{XOR} + P_{MUX16x1}$	32781.6
[22]	$P_{Enc} + P_{XOR} + P_{MUX4x1}$	12207.1
[23]	$P_{Enc} + P_{XOR} + P_{MUX16x1}$	32781.6
[24]	$2P_{Enc}$	17030.64
LAEP	$P_{Enc} + P_{XOR}$	8660.182

5.3 Formal Security Analysis

The AVISPA (Automated Validation of Internet Security Protocols and Applications) tool is employed to analyze the security of the proposed approach [32]. Serving as a simulator, it leverages various back-end models to conduct automated analyses and represents security protocols using the formal language HLPSL (High-Level Protocol Specification Language) [32]. For transmission channels, the Dolev-Yao attack model is adopted [33].

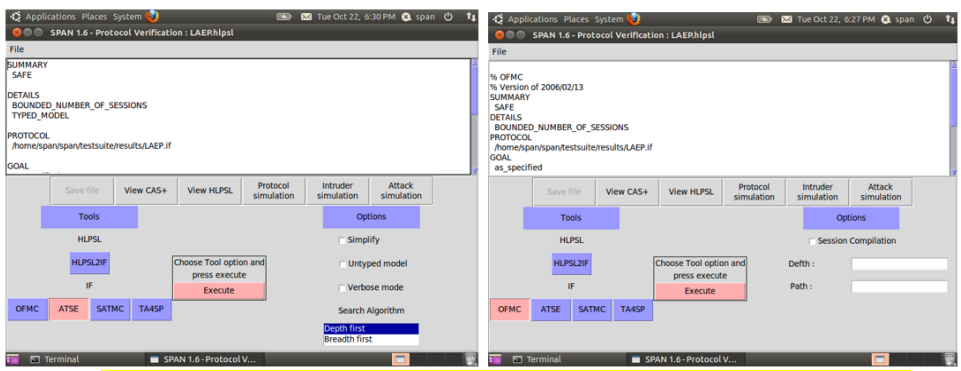

Fig. 7 (a). Simulation results using CL-AtSe and OFMC backends of LAEP Scheme

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS

BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/span/span/testsuite/results/LAEP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 4 nodes
  depth: 2 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/LAEP.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.02 seconds
  Computation: 0.00 seconds
```

Fig. 7 (b). Summary report of LAEP scheme using OFMC and CL-AtSe backends

The sender and receiver are the two communicating parties. A symmetric channel for

secure registration between the two users is represented as SKspms. H denotes a one-way hash function, XOR refers to bitwise addition under modulo 2, and Exp represents exponentiation. The two channels used for transmitting and receiving messages are denoted as SND() and RCV().

Figure 7 (a) and (b) illustrate the back-end simulation results of the proposed technique. The Security Protocol ANimator for AVISPA (SPAN) integrates the On-the-Fly Model-Checker (OFMC) and the Constraint-Logic-based Attack Searcher (CL-AtSe). Based on the simulation results, the proposed system is deemed "SAFE," demonstrating its resilience against both passive and active attacks.

5.4 Experimental testbed

For the experimental study, a real-world testbed is set up consisting of a Lenovo laptop running Windows 11, powered by an 11th-generation Intel i5 core with 32 GB of RAM. Figure 8 illustrates the testbed setup, where the Raspberry Pi 4 Model B boards serve as the sender and the laptop acts as the receiver. The LAEP scheme is implemented in a Python environment and deployed on the Raspberry Pi boards.

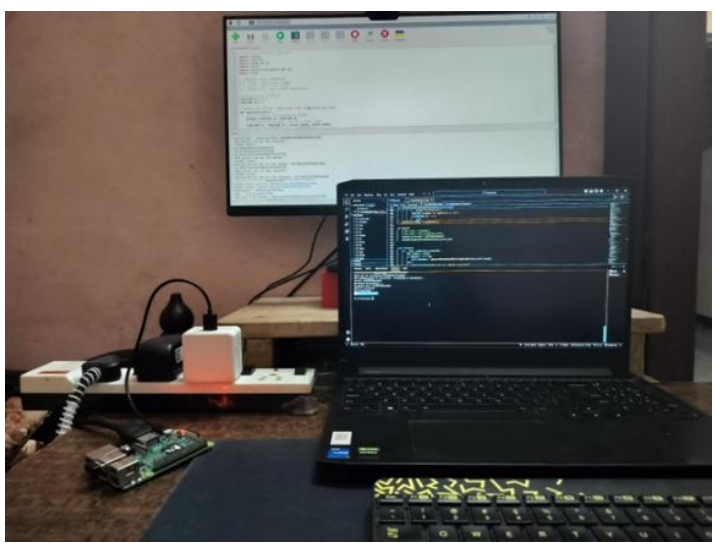

Fig. 8. Experimental testbed of LAEP

6. CONCLUSIONS AND FUTURE SCOPE

This paper presented a novel lightweight authenticated cipher algorithm using a Key-dependent S-box. The shared secret key 'K_s' was created by combining two modular operations with a one-way hashing function, making it challenging for an unauthorized party to deduce the key. The security analysis of the LAEP scheme has proven that the proposed cipher technique is impervious to quite a few well-known attacks, like impersonation, Man-in-the-Middle, related key, known-key security, modification, and replay, and offers an unprecedented scale of security with limited hardware cost.

The performance analysis of LAEP indicates that the proposed scheme achieved 50% of SAC, 85% of non-linearity, and 1730 GEs. Additionally, simulation results conducted with the Cadence Virtuoso EDA tool using 90nm technology demonstrate that the proposed cipher scheme requires one-fourth of the computation time and reduces power consumption, rendering it suitable for IoT-connected devices. Simultaneously, formal security assessments conducted using the AVISPA simulator have shown that the LAEP approach effectively withstands security threats. Furthermore, the experiment was performed in a real-world testbed environment, demonstrating that the message was successfully shared among all entities. Thus, the proposed system surpasses various performance metrics while securely transmitting data over unsecured channels.

The proposed method has addressed secure data transmission between two entities. Future work may explore the integration of post-quantum cryptographic primitives to ensure resilience against attacks from quantum adversaries in the Internet of Everything (IoE) or Internet of Vehicle (IoV) environment.

REFERENCES

1. M. G. Samaila, M. Neto, D. A. Fernandes, M. M. Freire, P. R. Inácio, "Challenges of securing Internet of Things devices: A survey," Security and Privacy, vol. 1, no. 2, 2018. pp. e20.
2. M. J. Covington, R. Carskadden, "Threat implications of the internet of things," 5th International Conference on Cyber Conflict (CyCon), 2013. pp. 1-12.
3. M. Soni, D. K. Singh, "LAKA: Lightweight Authentication and Key Agreement Protocol for Internet of Things Based Wireless Body Area Network," Wireless Personal Communications, vol. 127, no. 2, pp. 1067-1084, 2022. doi:10.1007/s11277-021-08565-2
4. F. Jindal, R. Jamar, P. Churi, "Future and challenges of Internet of things," International Journal of Computer Science Information Technology, vol. 10, no. 2, 2018. pp. 13–25.
5. C. C. Sobin, "A survey on architecture, protocols and challenges in IoT," Wireless Personal Communications, vol. 112, no. 3, 2020. pp. 1383-1429.
6. W. Diffie, M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, vol. 22, 1976. pp. 644–654.
7. S. Jebri, A. Ben Amor, M. Abid, A. Bouallegue, "Enhanced Lightweight Algorithm to Secure Data Transmission in IoT Systems," Wireless Personal Communications, vol. 116, pp. 2321–2344, 2021. doi:10.1007/s11277-020-07792-3
8. N. Nesa, T. Ghosh, I. Banerjee, "Design of a chaos-based encryption scheme for sensor data using a novel logarithmic chaotic map," Journal of Information Security and Applications, vol. 47, pp. 320-328, 2019. doi:10.1016/j.jisa.2019.05.017
9. M. Luo, Y. Zhang, M. K. Khan, D. He, "An efficient chaos-based 2-party key agreement protocol with provable security," International Journal of Communication Systems, vol. 30, no. 14, pp. e3288, 2017. doi:10.1002/dac.3288
10. A. Aziz, K. Singh, "Lightweight Security Scheme for Internet of Things," Wireless Personal Communications, vol. 104, pp. 577–593, 2019. doi:10.1007/s11277-018-6035-4

11. M. Gupta, K. K. Gupta, M. R. Khosravi, P. K. Shukla, S. Kautish, A. Shankar, "An Intelligent Session Key-Based Hybrid Lightweight Image Encryption Algorithm Using Logistic-Tent Map and Crossover Operator for Internet of Multimedia Things," Wireless Personal Communications, vol. 121, no. 3, pp. 1857-1878, 2021. doi:10.1007/s11277-021-08742-3
12. Q. Zheng, X. Wang, M. K. Khan, W. Zhang, B. B. Gupta, W. Guo, "A Lightweight Authenticated Encryption Scheme Based on Chaotic SCML for Railway Cloud Service," Special Section on Recent Advances in Computational Intelligence Paradigms for Security and Privacy for Fog and Mobile Edge Computing, IEEE Access, vol. 17, no. 6, pp. 711-722, 2017. doi:10.1109/ACCESS.2017.2775038
13. T. Ara, P. G. Shah, M. Prabhakar, "Dynamic key Dependent S-Box for Symmetric Encryption for IoT Devices." Second International Conference on Advances in Electronics, Computer and Communications (ICAECC-2018), 2018. pp. 1-5.
14. A. Alahdal, N. K. Deshmukh, "A Systematic Technical Survey of Lightweight Cryptography on IoT Environment," International Journal of Scientific & Technology Research, vol. 9, no. 3, 2020.
15. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, C. Vikkelsoe, "PRESENT: an ultra-lightweight block cipher," Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, pp. 450-466, 2007.
16. Chaudhary, N., Shahi, T.B., & Neupane, A, "Secure Image Encryption Using Chaotic, Hybrid Chaotic and Block Cipher Approach", Journal of Imaging, vol. 8, 2022, doi:10.3390/jimaging8060167
17. Bhaskar, M, Singh, J.P, "A lightweight image encryption scheme based on chaos and diffusion circuit", Multimedia Tools and Applications, vol. 81, pp. 34547 – 34571, 2022. doi:10.1007/s11042-021-11657-7
18. Yasmin, N., Gupta, R, "Modified lightweight cryptography scheme and its applications in IoT environment", International Journal of Information Technology, vol. 15, pp. 4403–4414, 2023. doi:10.1007/s41870-023-01486-2
19. Abutaha, M., Atawneh, B., Hammouri, L. "Secure lightweight cryptosystem for IoT and pervasive computing" Scientific Report vol. 12, 2022. doi:10.1038/s41598-022-20373-7
20. Z. Tang, J. Cui, H. Zhong, M. Yu, "A random PRESENT encryption algorithm based on dynamic S-box," International Journal of Security and its Applications, vol. 10, no. 3, pp. 383–392, 2016.
21. A. Prathiba, V. S. K Bhaaskaran, "Lightweight S-box architecture for secure internet of things," Information, Vol. 9, No. 1, pp. 1–14, 2018.
22. S. Verma, S. K. Pal, S. K. Muttoo, "A new tool for lightweight encryption on android," IEEE International Advance Computing Conference (IACC), pp. 306–311, 2014.
23. M. Khan, S. S. Jamal, "Lightweight Chaos-Based Non-linear Component of Block Ciphers," Wireless Personal Communications, vol. 120, no. 4, pp. 3017–3034, 2021. doi:10.1007/s11277-021-08597-8
24. Sarra Jebri, Arij Ben Amor, Mohamed Abid, Ammar Bouallegue, "Enhanced Lightweight Algorithm to Secure Data Transmission in IoT Systems," Wireless Personal Communication, Vol. 116, pp. 2321–2344, 2021, doi:10.1007/s11277-020-07792-3.

25. L. You, E. Yang, G. Wang, “A novel parallel image encryption algorithm based on hybrid chaotic maps with OpenCL implementation,” Soft Computing, Vol. 24, pp. 12413-12427, 2020. doi:10. 1007/s00500- 020- 04683-4
26. G. V. Kaushik, S. Giridaran, G. N. L. Reddy, G. A. Sathish Kumar, T. J. Jeyaprabha, R. Kousalya, “A Secret Key Generation using Double Chaotic Maps and Hash Algorithms for Image Encryption,” IETE CHENCON International Conference on Power of Digital Technologies in Societal Empowerment, 2021. pp. 132-135.
27. B. T. Hammad, N. Jamil, M. E. Rusli, M. R. Z'aba, “A survey of Lightweight Cryptographic Hash Function,” International Journal of Scientific & Engineering Research, vol. 8, 2017. pp. 806-814.
28. M. Khan, T. Shah, M. A Gondal, “An efficient technique for the construction of substitution box with chaotic partial differential equation,” Non-linear Dynamics, vol. 73, pp. 1795–1801, 2013. doi:10.1007/s11071-013-0904-x
29. R. Kousalya, G. A. Sathish Kumar, “Security Analysis against Differential Cryptanalysis using Active S-boxes,” Turkish Journal of Computer and Mathematics Education, vol. 12, no. 13, 2021. pp. 701-709.
30. R. Mishra, M. Okade, and K. Mahapatra, “Optimized s-box architectures of present cipher for resource constrained applications,” in Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (ISES), 2020. pp. 304–307.
31. H. Kim, Y. Jeon, G. Kim, J. Kim, B. Y. Sim, D. G. Han, H. Seo, S. Kim, S. Hong, J. Sung, D. Hong, ”PIPO: A lightweight block cipher with efficient higher-order masking software implementations,” Information Security and Cryptology–ICISC 2020: 23rd International Conference, Proceedings 23, 2021. pp. 99-122, Springer International Publishing.
32. A. Armando, D. Basin, J. Cuellar, M. Rusinowitch, L. Vigano, “The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications”, ERCIM News 64, 2006.
33. D. Dolev, D, A. D. Yao, “On the security of public key protocols”, IEEE Transaction Information Theory, 29:2 1983, pp. 198–202.

Kousalya, R. received her B.E. degree in Electronics and Communication Engineering from the University of Madras and M.E degree in Applied Electronics from Anna University, Chennai. She is currently working as an Assistant Professor at Sri Venkateswara College of Engineering, Sriperumbudur. Her research interest includes VLSI Design, Computer Networks, Cryptography, and Network Security.

Sathish Kumar, G.A. obtained his M.E degree from PSG College of Technology, Coimbatore. He has completed his Ph.D. at Anna University, Chennai. Currently, he is working as a professor in Sri Venkateswara College of Engineering, Sriperumbudur. His research interest is Network Security, Wireless Networks, and Cryptography.