

Optimized Round key generation in Lightweight Secure Encryption for Next-Generation IoT Networks

Archana S Nadhan^{1*}, Jeena Jacob I²

^{1,2}Department of Computer Science & Engineering GITAM School of Technology,
GITAM Deemed to be University, Bengaluru, India

KEYWORDS:

Internet of Things (IoT),
Lightweight Encryption,
Secure IoT (SIT),
Round Key Modification,
Data Security,
Entropy, Correlation Analysis, 5G
Communication,
Edge Computing,
FPGA

ARTICLE HISTORY:

Received 04-01-2025
Revised 02-02-2025
Accepted 23-03-2025

DOI:

<https://doi.org/10.31838/NJAP/07.01.14>

ABSTRACT

The rise of the Internet of Things (IoT) has significantly transformed modern communication systems by linking billions of devices and producing large volumes of data. However, conventional encryption methods are often too heavy for the limited processing capabilities and energy constraints typical of IoT devices. To address this, our study focuses on improving the Secure IoT (SIT) encryption algorithm—a lightweight, 64-bit block cipher that combines elements of Feistel structures with substitution-permutation networks, achieving encryption in just five rounds. Although efficient, the original SIT algorithm shows a security gap: a minor change in a single bit of the key can lead to the unintended exposure of nearly half the encrypted data. To overcome this limitation, we propose a refined key generation approach that modifies the structure of the fourth round key—identified as the most structurally unique among the five. This reconfiguration enhances the sensitivity of the key schedule, making the encryption more robust against attacks that exploit minimal key differences. We implemented the modified algorithm in MATLAB and assessed its performance through statistical metrics such as entropy and correlation. The results reveal that the improved version offers stronger image encryption, greater randomness, and reduced similarity between the original and encrypted data. To demonstrate its practical use, we evaluated the algorithm's potential in secure wireless communication environments, particularly within 5G, edge computing, and smart IoT systems. Additionally, its lightweight nature makes it suitable for hardware-level implementation on FPGA or ASIC platforms, supporting secure, real-time communication in embedded systems. Overall, the proposed enhancement strengthens the SIT encryption algorithm and makes it a promising solution for next-generation IoT applications requiring efficient and secure data transmission.

Author's e-mail: asnathan@gitam.edu, ijacob@gitam.edu

Author's Orcid id: 0000-0002-8035-4903, 0000-0001-6706-1017

How to cite this article: Archana S Nadhan, Jeena Jacob I, Optimized Round key generation in Lightweight Secure Encryption for Next-Generation IoT Networks, National Journal of Antennas and Propagation, Vol. 7, No.1, 2025 (pp. 93-107).

INTRODUCTION

In recent years, networks have grown in importance as a communication tool. Consequently, the need for internet security has grown to be essential for safe information sharing. Passwords may be sent safely across extensive networks by using cryptography. In a cryptographic system, communications are encrypted and decrypted using sequences of operations known as cryptographic algorithms. The Advanced Encryption Standard is one

of them; it is a requirement for data encryption in both software and hardware that is used to conceal important and sensitive information.^[1] The one-time pad is an application-layer encryption method that provides information-theoretic security. The physical-layer secret key generation process is a good contender for supplying the random key for this method.^[2, 31] The protection of patient data is now one of the most difficult issues in telemedicine as well as e-health because of the

integration of technological advancements with the medical industry. Adequate procedures are needed for the distribution of the encrypted medical picture in order to ensure patient privacy.^[3] Sensitive data is produced and sent by IoT devices in large quantities. IoT security threats are a big worry, particularly when using cutting-edge methods like Non-Orthogonal Multiple view (NOMA) approach, which allows users to view the data of other users. Surprisingly, NOMA-based IoT systems face threats to their communication from both outside listeners and dubious inside users. This highlights the critical need of using an encryption mechanism to safeguard communications inside IoT systems.^[4, 32] The highly linked network of diverse gadgets known as the “IoT,” makes it seem as if any kind of communication—even illegal communication—is feasible. Due in large part to the fact that some classes of IoT devices have limited resources, traditional Internet security protocols have been shown to be useless in these kinds of networks, making security requirements for these networks more important. If we want to make sure that data communicated between a bunch of IoT devices is safe, reliable, and easy to retrieve, then we need secured group communication. SGC is a challenging undertaking since IoT devices was often resource-constrained, having limited memory, computation, energy, and power.^[5]

With the exponential growth of the Internet of Things (IoT), the demand for lightweight and secure communication protocols has become more pressing than ever. Devices deployed in smart environments—

ranging from healthcare systems to industrial monitoring platforms—operate under severe constraints in power, memory, and processing capacity. Traditional cryptographic approaches are often unsuitable for these contexts. The improved key generation mechanism proposed in this study strengthens the Secure IoT (SIT) encryption algorithm, making it more adaptable for use in real-time wireless communication systems. This includes deployment in secure WLANs, cognitive radio networks, and low-latency sensor-based infrastructures, where encryption must be both fast and resilient to ensure uninterrupted and protected data exchange.^[6]

New methods of internet-based communication between machines and people have been made possible by the Internet of Things. Even while sensor-driven technology has generally made life easier, most Internet of Things infrastructures have security issues. Lightweight encryption algorithms have been a more viable choice for intelligent along with sensor-based applications with the advent of the Internet of Things. As long as public-key infrastructure predominates globally, symmetric key encryption, such Advanced Encryption Standard, has a lot of potential to coexist with IoT devices in smart homes.^[33] In telemedicine, patient medical pictures must be sent securely and kept confidential. To increase the safety of medical pictures to be encrypted, integrated chaotic key generators is suggested to improve key creation and sensitivity. For both the permutation as well as diffusion processes, respectively, the initial seed for the fractional order chaos system along with

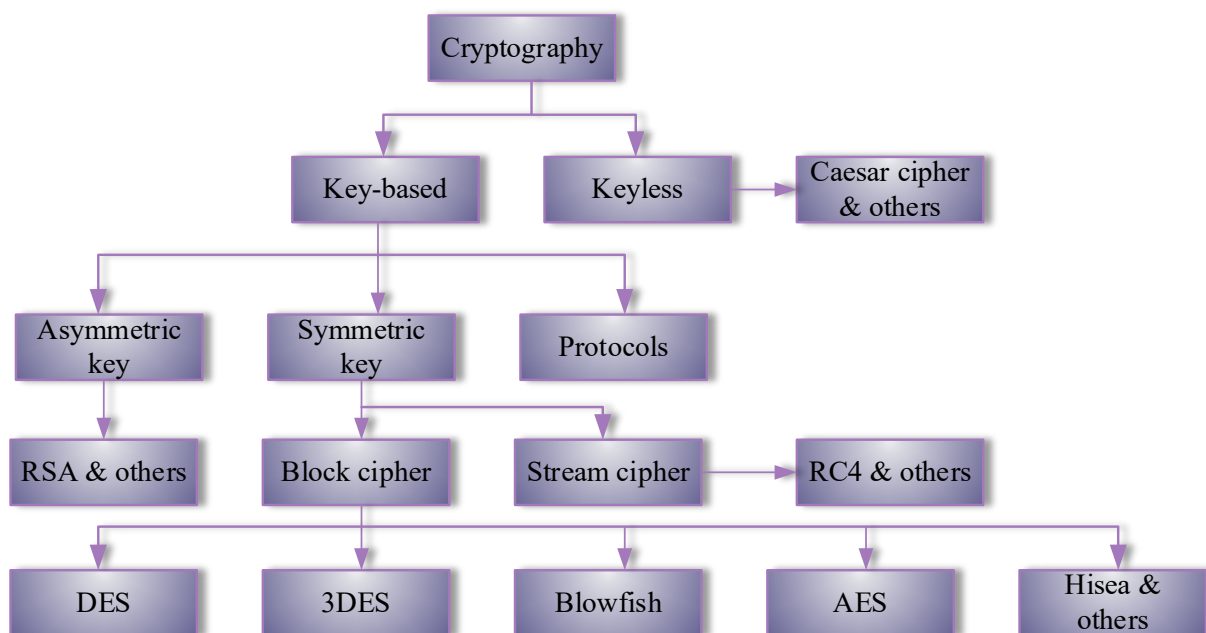


Fig. 1. Cryptography Mechanisms

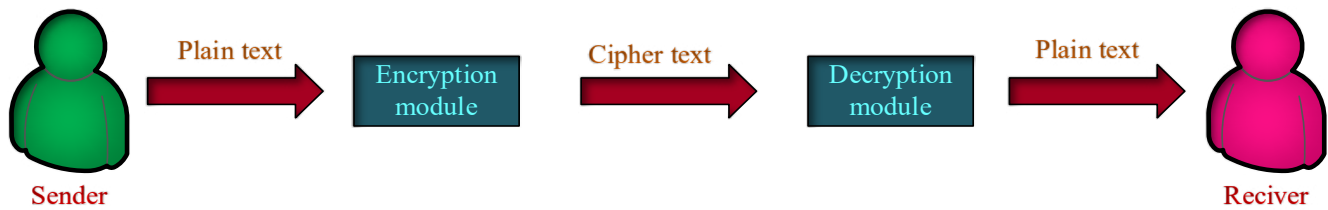


Fig.2. Process of Encryption and Decryption

Lorenz system are generated via CCKG.^[7] Ciphertext-policy attributes-based encryption might enable the data owner to exchange his private information via the cloud with self-defined access structures in the context of the internet-assisted industrial network of Things. The length of the ciphertext, the cost of generating keys, and the decoding overhead of generic CP-ABE systems all rise with the variety of characteristics involved.

Furthermore, weighted qualities are not taken into account; only ordinary attributes are.^[8, 34] Because IoT devices often have limited computational and energy capabilities more complicated encryption techniques are not appropriate for ensuring secure connection. The physical layer's key generation offers a viable solution to this issue by enhancing wireless communications security by generating shared secret keys from noisy wireless channels measurements.^[9] With the recent developments in networking and IoT technology, the areas of security and privacy in IoT environments have gotten the most attention. IoT devices are being used to gather real-world person datasets; however security and privacy present significant challenges.^[10] Cryptographic keys, which are lengthy random bit strings produced by specialized algorithms, help safeguard data by rendering it predictable to any potential attacker. Cryptographic keys find use in a wide range of cryptographic techniques across several domains, including cloud computing, fog computing, IoT, and others. Cryptographic data decryption and encryption methods depend on key generation algorithms.^[11]

A plethora of low-powered Wide Area Network methods have been put out recently to secure IoT gadgets with little power consumption but resource constraints. Using two-session preshared private keys, the long-range wide-area network (LoRaWAN) is an inexpensive power network protocol that enables message integrity, authentication, and encryption. Nevertheless, while supporting some security features, the LoRaWAN has issues with key updates and session key creation.^[12] The IoT with 5G capabilities is amazing in terms of its growth, impact, and potential. Some security issues arise from the amount of data delivered and processed by IoT devices that rely on coverage and connection.

As they utilize IoT technology more often in our everyday lives, the risks associated with today's internet might become more widespread. For IoT-based 5G network devices to be secure there must be increased system life, coverage, and connection. These mistakes provide gaps that allow security breaches to occur.^[13] New advances in quantum technology have shown the flaws in the traditional public cryptosystem. Shor's technique suggests that the use of asymmetric keys will not be feasible or safe in the near future, even if it can't currently be used on quantum computers.^[14] The IoT with cloud assistance is becoming more and more common in many industries, including healthcare. Sensitive information, such as private computerized medical records, might be readily exposed in such a situation, raising security risks. Because Symmetric Searchable Encrypted may facilitate effective search on encrypted data, it has been investigated extensively.^[15]

Among the numerous cryptographic methods at your disposal, ECC stands out as a formidable tool. Users are required to authenticate before they may access any stored data or make any requests. An authorized user must encrypt or decrypt data inside the database after obtaining a key via a key generator. You can get whatever key you need by storing it in a key generator. All three levels of database encryption—rows, columns, and elements—use a total of 256 bits of AES encryption. For row-level encryption, the following method employs a 521-bit ECC signature; for column-level encryption, it employs a secured AES 256-bit random key. Elements in databases may be securely encrypted at the element level using AES and ECC signatures, ensuring their privacy and integrity. This method is detailed last in the article. To safeguard against database signature security, it is crucial to encrypt data both at rest and while in transit across our network. It is difficult to counter the advantages of the components level when the attacker obtains a key that reduces their element count by one. It will take hundreds, if not millions, of keys to deal with the downsides. The remaining sections of this document are organized as follows. In Section 2, we will go over a few publications that are relevant to the strategy that has been suggested. Section 3 will then include some early research. The recommended strategy is laid out in

Section 5, while Section 4 provides an overview. In Section 6, we go over the security analysis. The experimental setting, findings, and discussion are all part of Section 7's evaluation of the performance analysis. By Section 8, the whole document comes to a close, leaving room for some future studies.

RELATED WORK

The primary goal of^[16] is to create a modified AES system that is more effective and resistant to multiple assaults by including basic operations. Key lengths of 128 bytes, 192 bits, along with 256 bits are available with AES. The initial dissemination rate of AES is rather low. By including additional processes into the cipher cylindrical and key formation process, the diffusion characteristics of the basic AES may be enhanced. The avalanche effect is used to compare the diffusion properties of the suggested approach with the traditional AES. The enhanced avalanche effect of the suggested AES algorithm demonstrates its superior security over the traditional AES. The Zybo-Zynq Z-7010 AP SoC research board is intended to receive the results of the suggested algorithm, which is run on the Vivado 2016.2 ISE Development Suite. That study also suggests an enhanced AES algorithm, which was made possible by changing the sub-bytes operation. That modification was designed to increase its reliance on round keys. In order to strengthen the algorithm's defense against assaults, the key length was increased to 256 bits. In order to increase the secret key rate, researchers suggested in^[17] a combined SKG and OTP encryption technique using reconfigurable intelligent surfaces. To improve the efficiency of secure communication, they separated the secure transmission process into two steps: SKG as well as encrypted packet transfer. In the meanwhile, they create the best possible algorithm to assign slots of time for SKG in a way that maximizes its effectiveness while posing no security risk. In addition, they develop an OTP encryption key update mechanism that utilizes our SKG scheme. The safe and efficient generation of keys by our technique is confirmed by simulation findings, which also show a considerable improvement in the secure communication efficiency of an intelligent Internet of Things system.

The goal of the^[18] research is to protect and sensitively cipher the COVID-19 pictures from a CT chest scan for the patient with the infection. Protein key generation has been suggested as a means of achieving a high level of safety in the encryption process. The goal of that work is to encrypt photos using a protein key and two rounds of AES. The suggested method's level of security has been estimated using the histogram. To assess the

level of security for the suggested technique, four criteria have been chosen: Standardized Mean Changing Intensity, Entropy, Number of Pixels Changing Rate, and Correlation coefficient. This results in an entropy close to eight, a UACI more than 30%, a correlation coefficient almost zero, and an NPCR of 99.5% and higher for the proposed method. When compared to earlier studies, the findings validate that the suggested strategy delivers a high degree of security and sensitivity. As a result, the suggested technique may be regarded as an algorithm that has been successfully implemented to meet the security criteria for sending CT scans to COVID-19 patients. A comparative research of secret key creation is presented in^[19] article, where several PLS key generation techniques are used to analyze the performance of a secure NOMA-enabled network of Internet of Things in an environment of untrusted users.

A Fully homogeneous encryption using Optimal Key Generating Secure Group Communications approach is designed for the IoT environment in the [20] article. The primary goal of the FHEOKG-SGC approach that is being discussed is to securely encrypt and route data in an Internet of Things context using group communication. In order to do it, the FHEOKG-SGC technique creates an encryption method based on FHE to protect data in an Internet of Things environment. Then, the method known as sine-cosine is used to pick the keys of the FHE technique appropriately. Concurrently, the plum tree technique is used to identify the paths inside the Internet of Things network. Lastly, a trust model is employed by the FHEOKG-SGC approach to enhance the safe communication process, as well as the management of keys center is used for the best possible key handling. A series of tests are used to evaluate the simulation-based evaluation of the FHEOKG-SGC approach, and the results are examined under different conditions. A thorough comparison analysis demonstrated how the FHEOKG-SGC algorithm outperformed other contemporary methods. Specifically, in^[21] paper, they combined PUF with conventional Key-based Hardware Obfuscation method to exploit PUF features by producing distinct and unclonable secret keys. they examined the PUF's consistency, distinctiveness, and dependability. They also consider PUFs as a possible seeding for pseudo-random numbers generators, which would further improve obfuscation while reducing power and space costs. To further improve authentication and data encryption in IoT devices, the PUF-based obfuscation method is also integrated with the AES (Advanced Encryption Standard) IP core. The suggested model is protected against convincing side-channel assaults, SAT attacks, and reverse-engineering, according to experimental findings and NIST analysis. The suggested model is put

into practice using Basys3 FPGAs while using the fewest resources possible in comparison to the existing methods. For cloud-assisted IIoT applications, authors introduced the secure, efficient, as well as weighted access control system in.^[22] The DO may create whatever fine-grained access architecture over weighted characteristics thanks to SEWAC without having to make it more difficult. Moreover, the ciphertext's length would not be increased by such weighted features. In order to reduce the authority's computing burden from processing many key request in the online phase—the majority of which are performed offline—SEWAC additionally facilitates online/offline key creation. The cloud bears the brunt of the decryption above. They have designed an effective batch verification approach that requires just three bilinear pairing procedures from the user to verify the accuracy of batch results, therefore guaranteeing the cloud's integrity throughout the outsourced decryption process. Additionally, they provide the formal safety proof for the suggested system. Thorough comparisons and implementation outcomes show that SEWAC is more effective at achieving the following: decreased length of ciphertext, efficient creation of keys, guarantee of the result of the outsourcing decryption, with weighted access control.

Physical layer key creation on Internet of Things nodes is made possible by the exceptionally low implementation complexity of the key generation strategy that researchers suggested in^[23] paper. To increase channel reciprocity, they first preprocess the channel data using a straightforward moving average filtering method before applying quantization. In order to accomplish reliable quantization of channel measurements, a bidirectional variance quantization approach is then proposed. By not depending on quantization threshold during quantization, that astute technique prevents crucial bits from being mismatched due to measurements being close to quantization thresholds. Then, they propose an improved Cascade technique to achieve lightweight and effective information reconciliation. While generating keys, our system manages to strike a good balance between efficiency and reliability, and it performs admirably in terms of implementation difficulties and key randomness, according to the simulation results. For the aim of safeguarding privacy within the framework of the Internet of Things, a public-key cryptosystem called ElGamal was developed.^[24] It makes use of an EGPKC-CERDA key generation process, which combines a red deer optimization algorithm with cross entropy. Based on the IEEE 802.15.4 MAC standard, which includes the security feature in the MAC header, the EGPKC-CERDA method is provided. For further peace of mind, the MAC header employs the EGPKC technique, which uses

optimum authentication key construction. To further enhance both the local and global optimum, the cross entropy technique has been incorporated into the RDA of the CERDA approach; this leads to the best key selection procedure for EGPKC. In order to adequately assess the EGPKC-CERDA algorithm's performance, a plethora of experiments were carried out and their results were thoroughly analyzed from many perspectives. The results of the experiments proved that the EGPKC-CERDA approach was more effective. A cryptographic key generating approach for generating randomised keys based on Salp Swarm algorithm and Shannon entropy was developed in^[25] study. The suggested technique for Cryptographic Key Generation makes use of salps' dynamic mobility to generate strong, randomized, and high-quality keys that are resistant to assaults. A salp is transformed into an encryption key using the quantization technique and transfer function. The suggested method for Cryptographic Key Generation has been assessed using four transfer functions in comparison to three cutting-edge swarm intelligence met heuristics: particle swarm optimization, Bayesian Adversarial Technology, and grey wolf optimization. Eight distinct bit length keys (512, 256, 192, 128, 96, 80, and 64) were created and assessed in light of their use in the various encryption algorithms (AES, DES, SIMON, PRESENT, SPECK, as well as 3DES). The suggested key generation technique successfully generates safe cryptographic keys, as shown by the simulation research. The development of a secure, lightweight authorization component for 5G intelligent lamp the poles which includes information abstracts and advanced encryption algorithms is the main focus of the research presented in the [26] paper, which focuses on trusted methods of authentication for Internet of Things terminals. An encrypted application programming interface, or API, is created by combining the AES symmetric encryption method with a changeable key in the trusted authentication algorithm. After the suggested trustworthy verification technique is integrated into the computer module, a test platform is constructed to show how well it prevents typical network assaults.

The team's key agreement method was proposed by scholars in^[27] to avoid the exposure of sensitive information and is used for encryption and decoding data sent between IoT devices. The multipolicy access, ciphertext storage, and hidden attribute authentication are all part of the key agreement mechanism. Key agreement is created via a collaborative network architecture that is edge-cloud based. The terminal has to use the key algorithm to create its own both private and public keys before it can connect to the cloud server and check its legitimacy. Secondly, Internet of Things (IoT)

terminals may get the rights linked with each property by encrypting them and then validating them to the cloud. With these permissions, the terminal encrypts the FL algorithm's parameters and sends them to the edge servers as secret variables. These ciphertext decryption variables are sent with the different types of FL terminals to the stored on the edge server. Ultimately, the shared model settings are downloaded and decrypted for FL purposes in order to train further terminal models. When compared to the referenced literature, the performance study demonstrates that model performs better in terms of computational difficulty and computational time. To circumvent the [28] problem, the authors proposed a blockchain-based secure information search mechanism (BSSMeta) in MEC. BSSMeta maintains the safety of the metadata finding tasks by using an autonomous key generation technique on the blockchain in conjunction with a minimal proxy re-encryption scheme. To improve the speed of the searching with uploading procedures, a buffer uploading approach and a main/secondary smart contracts mechanism are suggested. As a result, BSSMeta allows users to regulate metadata access and provides metadata searches in a multiuser context. Finally, they implement a functional prototype on the the Hyperledger project Fabric's architecture and provide a security assessment of BSSMeta. Experiments conducted on different workloads have shown the feasibility and flexibility of BSSMeta in terms of efficiency and security. Security at the physical layer is often cited in the [29] framework as an efficient way to compel secrecy in Internet of Things systems. It uses wireless channel properties, either with or without encryption, to provide a secure communication method. One interesting method minimizes the possibility of eavesdropping by taking employ of the geographic decorrelation properties of the wireless connection and allowing registered users to reach an agreement on a secret key using the channel's unpredictability. The connection among the channel samples collected by users and the noise that constantly interferes with wireless communications is one of the many factors that determine how reliable the channel-based generation of keys process is. The secrecy key rate, or the greatest amount of secret bits obtained from each channel observation, may be used to illustrate how sensitive the key generation process is. In that study, the influence of primary channel characteristics on the SKR values is investigated by computing the secrecy key rates value via simulations run under various operating settings. Unlike earlier research, the secrecy key rate is calculated while taking into account various correlation values between the eavesdropper and genuine users across a line-of-sight wireless channel. Establishing a standard an asymmetric cryptography which should be

immune to quantum computers is the main emphasis of .^[30] In recent years, that has grown in significance. At now, the task of standardizing asymmetric encryption is almost complete. Two post-quantum encryption techniques that were both chosen as NIST fourth-round finalists were assessed for performance in that research. The research provided information on the efficacy and practicality of the decapsulation, encapsulation, as well as key generation procedures by evaluating them. It will need more study and standardization work to make post-quantum encryption safe and effective. It is important to consider several criteria including safety levels, requirements for performance, key dimensions, and platform compatibility when choosing suitable post-quantum encryption techniques for particular applications. In the era of quantum computing, that work offers practitioners and researchers in post-quantum cryptography useful information to help them make decisions about which algorithms to use to safeguard sensitive data.

Research Gaps

Database security is in place to keep databases safe from harm. This kind of security control encompasses data, applications that access or use stored functionalities, databases, database servers, and associated network connections. Data protection techniques employ subject qualifications and traits, data contents, and other pertinent contextual information (like time) to impose access control restrictions. Among the most significant threats to database security are inadvertent or malicious actions taken by authorized customers of databases, administrators, or network/system managers; database applications, structures, or security settings being altered or accessed by unauthorized individuals; hackers obtaining access to sensitive data, metadata, or database functionality. In addition, malware infections can cause a host of problems in databases, such as unapproved disclosure of sensitive information, loss or corruption of data or applications, suspension or denial of authorized access, assaults on other computer systems, and unanticipated databases service outages.

PROPOSED WORK

A. System Model

There are four critical steps to ensure the safety of data transfer in our suggested architecture. Using a Public Key Servers distribution strategy and combining parts of symmetric and asymmetric algorithms, our model incorporates all of these properties. When using symmetric cryptography, the encryption and decryption processes share a single key. Since there is no need to

generate key pairs when using a symmetric approach, the whole procedure is often quicker due to the shared key. Public Key Cryptography and Asymmetric Cryptography are two sides of the same coin; the former uses a single key for both encryption and decryption. The suggested model implemented subsequent cryptographic procedures and set up a secure channel. The primary objective is the efficient and safe conveyance of data and instructions between the IoT and a user, with the utmost attention to protecting confidentiality, integrity, and authentication. At the outset, all entities (Internet of Things devices, user devices, etc.) must register with the Public Key Server by providing its Public Key—a function of Elliptic Curve Cryptography—and its Mac Address—a parameter that is unique for each individual—as inputs. We skip the signup step if you are already registered. After that, in order to prevent eavesdropping and attacks like MITM, the user and the IoT device will exchange public keys securely after appropriate

authentication. After the Public Key exchange goes through, the two parties will settle on a single secret key that was calculated using the Elliptic Curve Diffie-Hellman algorithm. The Advanced Encryption Standard (AES) encryption method, which encrypts and decrypts instructions and data delivered over the network, will use the Shared Private Key as an input.

B. ECDH

Lipizzare curve Earle Conway Hellman (ECDH) Lipizzare curve Key agreement protocols like Elliptic Curve Cryptography (ECDH) enable two parties to create a shared key using the public-private key combinations that were established using Elliptic Curve Cryptography (ECC).

By locating an area on the curve and applying the format, Alice and Bob are able to produce key Paris using ECC: with a prime number 'p' which is

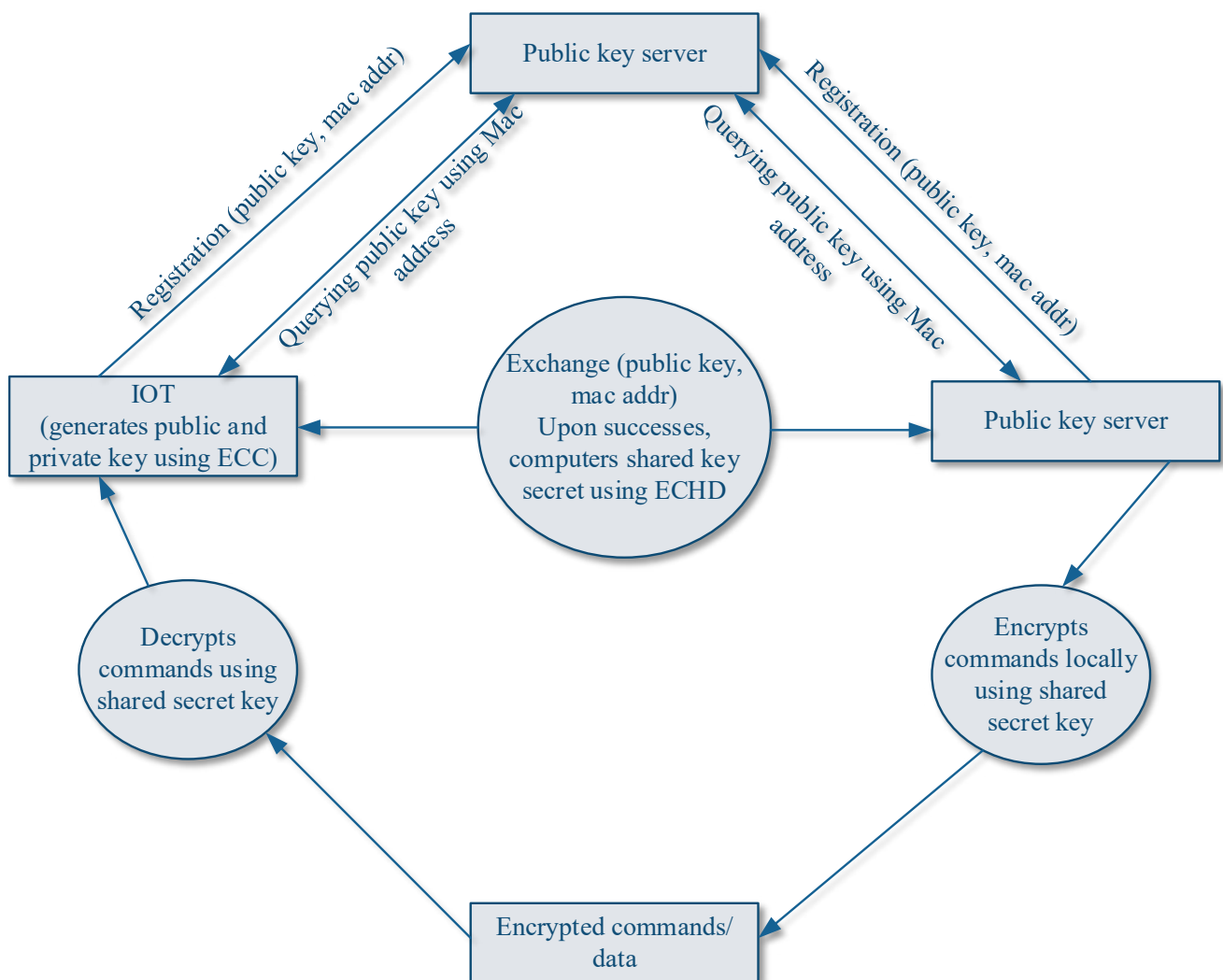


Fig.3: Working Flow of Hybrid Encryption Model

- a. Modp will be used for all operations.
- b. Each user’s private key is a string of seemingly random integers. The private key d_A belongs to Alice while the secret key d_s belongs to Bob.
- c. As an elliptic curve, the largest point is “G.”
- d. Q_A (Alice’s Pubic Key) is the public key that both Alice and Bob will use. $X G$ Public Key)

Subsequently, Bob and Alice will trade public keys. To find the shared secret key, they’ll each utilize the other’s public key & their own private key. Alice’s Shared Key SharedKey

C. Modified AES

Advanced Encryption Standard (AES) is a symmetric-key block cypher technique that is both highly secure and notoriously difficult to decipher. Figure 6 indicates that AES employs three block ciphers with a constant amount of 128 bits, keys having sizes of 192 bits and 256 bits, and more. There is no restriction to the size of a key, however there is a limitation of 256 bits for a block. The Data Encryption Standard, or DES, Feistel networks are not used in the AES architecture; instead, it relies on substitution-permutation networks (SPNs). This paper suggests an enhanced AES algorithm that uses round keys more heavily in the sub-bytes operation. This ensures that even minor changes in key are detected in the cipher.

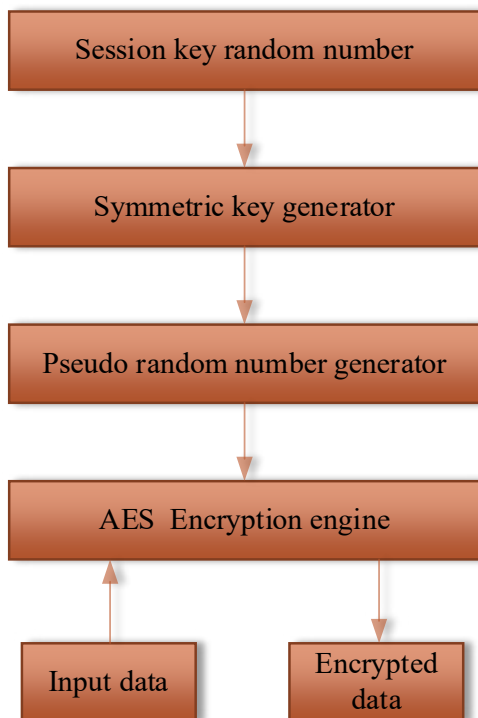


Fig.4: AES for Encryption and Decryption

In order to offer various keys with various cascading levels, the suggested approach utilizes the diffusion of the AES algorithm with a cascading technique. It employs a 200-bit plaintext and a 400-bit key, which is split into two equal halves. Instead of ten rounds, the new method uses a cascaded format with just five. This will reduce the AES’s time complexity by removing the mix column twice.

Modifications to the suggested algorithm include:

Cascading technique: While the original AES did not incorporate any cascading concepts, the proposed algorithm incorporates them for two rounds, each utilizing a different key. This increases the algorithm’s security, and every one of the two transmitted layers only requires five rounds. This means that the algorithm’s time complexity may drop from 2^{48} to 2^{16} if MixColumn were to be removed twice.

	C_1	C_2	C_3	C_4	C_5	C_n
R_1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{1n}
R_2	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{2n}
R_3	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{3n}
R_4	d_{41}	d_{42}	d_{43}	d_{44}	d_{45}	d_{in}
R_5	d_{51}	d_{52}	d_{53}	d_{54}	d_{55}	d_{5n}
R_m	d_{m1}	d_{m2}	d_{ml}	d_{met}	d_{m5}	d_m

Structure of the AES (Fig. 3)

- Here, we’ve started with a 200-bit plaintext and a 400-bit key.
- Half of the 400-bit key is 200 bits, while the other half is 200 bits. Making two separate keys for each iteration of the cascaded encryption approach would render the encryption process more resistant against Brute force attacks.
- To handle all 200 bits of plain text concurrently, the size of the state array is adjusted from 4x4 to 5x5.
- The suggested method encrypts the identical plaintext twice, once using the AES technique and once with a different key.
- Every encryption cycle now only requires 5 rounds instead of 10 before.
- As a result, the method is able to reduce the time it takes to encrypt data since it skips over MixColumn twice.

Rows in Figure 3 are labeled as

- $R_i=(d_{i1} \ \square \ d_{i2} \ \dots \ d_{in}), i=1,2,\dots,m$
- $C_j=(d_{1j} \ \square \ d_{2j} \ \dots \ d_{nj}), j=1,2,\dots,n$

In order to complete the encryption process, the matrix model is crucial. A 4x4 matrix was created and used on the raw data. In order to implement encryption, each input stream is split into blocks of 16 bits. There are two distinct iterations of key creation for every 16-bit input block. Data mapping makes use of a generator for the hexadecimal number system.

KEY GENERATION

Both phases of the key building model must process the input blocks for key creation to be complete.

Producing safe secrets has been greatly aided by key construction. Encryption strength is dependent on the production of strong keys. Without the key, the whole security system would collapse, and outsiders would be able to access sensitive information. We are generating keys in a block-wise fashion. Each data block has its own unique key.

Data entry: Data secrecy, data integrity, authorization, and non-repudiation are at the heart of contemporary cryptography, which is a subfield of network security that focuses on building and analyzing protocols to prevent third parties or the public from studying private messages. We can look at an example of the suggested technique in action and see that it works for input data of any length. There are 16-bit blocks in the input data that are used for processing. Create a 4x4 matrix and place the cells in the correct spots. Numbers ranging from zero to fifteen are filled into each cell of the matrix.

Round 1

The process of creating a key began at this point; the values chosen as a key are from the 4x4 matrix's left diagonal. The value at the location (0,0)th is crucial for the X₀ row. Also, every four bits of data requires a new key.

In order to generate a key for the first round, we'll use 16-bit block X. Out of the 16 bits of input, the key is set as,

$$X_0 = X_{(0,0)}, X_1 = X_{(1,1)}, X_2 = X_{(2,2)}, X_3 = X_{(3,3)}$$

For R1, the values located on the left diagonal are crucial. X₀, X₁, X₂, and X₃ were all computed in round 1 using the following formulas:

$$X_0 = \{X_{(0,0)} + X_{(0,0)}\} \{X_{(0,1)} + X_{(0,0)}\} \{X_{(0,2)} + X_{(0,0)}\} \{X_{(0,3)} + X_{(0,0)}\}$$

$$X_1 = \{X_{(1,0)} + X_{(1,1)}\} \{X_{(1,1)} + X_{(1,1)}\} \{X_{(1,2)} + X_{(1,1)}\} \{X_{(1,3)} + X_{(1,1)}\}$$

$$X_2 = \{X_{(2,0)} + X_{(2,2)}\} \{X_{(2,1)} + X_{(2,2)}\} \{X_{(2,2)} + X_{(2,2)}\} \{X_{(2,3)} + X_{(2,2)}\}$$

$$X_3 = \{X_{(3,0)} + X_{(3,3)}\} \{X_{(3,1)} + X_{(3,3)}\} \{X_{(3,2)} + X_{(3,3)}\} \{X_{(3,3)} + X_{(3,3)}\}$$

Appropriate values have been examined and applied to the aforementioned formula. You may expect,

$$R1 = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} F2 & E9 & BC & EB \\ D9 & E4 & E6 & E1 \\ C9 & DA & C2 & D1 \\ B1 & BB & AE & 9A \end{pmatrix}$$

R1 stands for the first round. To illustrate the key generating process using round 1, Equation 5 was used.

Round 2

The values on the right side of the 4x4 matrix are crucial for the second round. An important element of the X₀ row is the (0,3)th matrix's positioned values. A key is used for the X₁ (1,2)th "positioned value. Keys have also been produced for every four bits in the same manner.

For the second round, we'll use 16-bit block X to generate a key, which contains the following 16 bits of data,

$$X_0 \rightarrow X_{(0,3)}, X_1 \rightarrow X_{(1,2)}, X_2 \rightarrow X_{(2,1)}, X_3 \rightarrow X_{(3,0)}$$

Key R2 values are those located on the right diagonal. The values of X₀, X₁, X₂, and X₃ have been determined in round 2 using the key, and the following formulas are

$$X_0 = \{X_{(0,0)} + X_{(0,3)}\} \{X_{(0,1)} + X_{(0,3)}\} \{X_{(0,2)} + X_{(0,3)}\} \{X_{(0,3)} + X_{(0,3)}\}$$

$$X_1 = \{X_{(1,0)} + X_{(1,2)}\} \{X_{(1,1)} + X_{(1,2)}\} \{X_{(1,2)} + X_{(1,2)}\} \{X_{(1,3)} + X_{(1,2)}\}$$

$$X_2 = \{X_{(2,0)} + X_{(2,1)}\} \{X_{(2,1)} + X_{(2,1)}\} \{X_{(2,2)} + X_{(2,1)}\} \{X_{(2,3)} + X_{(2,1)}\}$$

$$X_3 = \{X_{(3,0)} + X_{(3,0)}\} \{X_{(3,1)} + X_{(3,0)}\} \{X_{(3,2)} + X_{(3,0)}\} \{X_{(3,3)} + X_{(3,0)}\}$$

After plugging the matrix X values into the previous calculation, the output is,

$$R2 = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 9A & E2 & B5 & E4 \\ DB & E6 & E8 & E3 \\ E1 & F2 & DA & E9 \\ C8 & D2 & C5 & B1 \end{pmatrix}$$

Matrix R2 depicts the second round of key creation. When building keys, the subtraction operation is crucial. It is proper to deduct the results of round 2 from those of round 1.

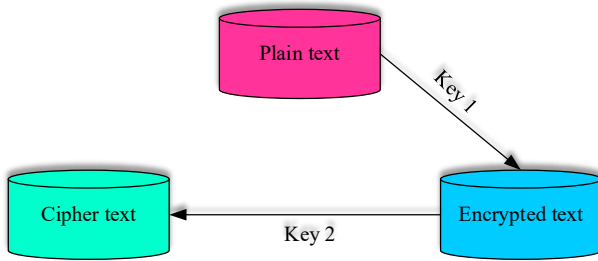


Fig.5. Two Round Keys in AES

Plaintext

KEY=FECDBA1357924680ABCDEF1234567890FEDCBA0

The initial round of cascade uses key to encrypt the plaintext, as shown by KEY1=FECDBA1357924567890ABCDEF123.

The bottom half of a key is used in yet another cascading cycle as

KEY2 FEDCBA0987654321AB

The second key is used to encrypt the plaintext once again.

2. Our proposed method takes use of a 200-bit block size rather than the original 128-bit block. To include exactly 200 bits in one state array, for instance, the dimension of a state array is increased from 4*4 of 5*5, which causes changes to the stages of each round.

A. SubByte: Modification: This modification is unaffected by changes in the plaintext's size.

B. ShiftRow: The suggested technique does a 5 ShiftRow operation rather than a 4 shift operation since the plaintext has changed.

SR _{0,0}	SR _{0,1}	SR _{0,2}	SR _{0,3}	SR _{0,4}
SR _{1,0}	SR _{1,1}	SR _{0,0}	SR _{1,3}	SR _{1,4}
SR _{2,0}	SR _{2,1}	SR _{2,2}	SR _{2,3}	SR _{2,4}
SR _{3,0}	SR _{3,1}	SR _{3,2}	SR _{3,3}	SR _{3,4}
SR _{4,0}	SR _{4,1}	SR _{4,2}	SR _{4,3}	SR _{4,4}

Table1: Before ShiftRow Operation:

SR _{0,0}	SR _{0,1}	SR _{0,2}	SR _{0,3}	SR _{0,4}
SR _{1,1}	SR _{1,2}	SR _{1,3}	SR _{1,1}	SR _{1,0}
SR _{2,2}	SR _{2,3}	SR _{2,4}	SR _{2,0}	SR _{2,1}
SR _{3,3}	SR _{3,4}	SR _{3,0}	SR _{3,1}	SR _{3,2}
SR _{4,4}	SR _{4,0}	SR _{4,1}	SR _{4,2}	SR _{4,3}

Table2: After ShiftRow Operation

C. The MixColumn polynomials equation is transformed into $2X^4+4X^3+3X^2+X+1$. Also, instead of 4*4, the polynomial matrix becomes 5*5. This.

$$\begin{pmatrix} 02 & 01 & 01 & 03 \\ 04 & 02 & 01 & 01 \\ 03 & 04 & 02 & 01 \\ 01 & 03 & 04 & 02 \\ 01 & 01 & 03 & 04 \end{pmatrix} 5 \times 4$$

Fig.6: Fixed Polynomial matrix

$$\begin{pmatrix} E0 & 4C & 09 & 8A & 4C \\ 4C & E0 & 7D & 09 & 8A \\ 8A & 4C & E0 & 7D & 09 \\ 01 & 8A & 4C & E0 & 7D \\ 7D & 01 & 8A & 4C & E0 \end{pmatrix} 5 \times 5$$

Fig 7: A matrix of inverse polynomials

D. AddRoundKey: The rise from 44 to 55 words is due to the increase in the amount of round key operations.

3. Omissions of MixColumn: The key reason for the decrease in time complexity in our suggested approach is the omission of MixColumn twice from the overall algorithm. ($2^{48} - 2^{16}$).

4. Increase in key size: The algorithm's security is enhanced by increasing the key's size from 128 bits to 400 bits. Here, a 400-bit key is divided into two 200-bit keys.

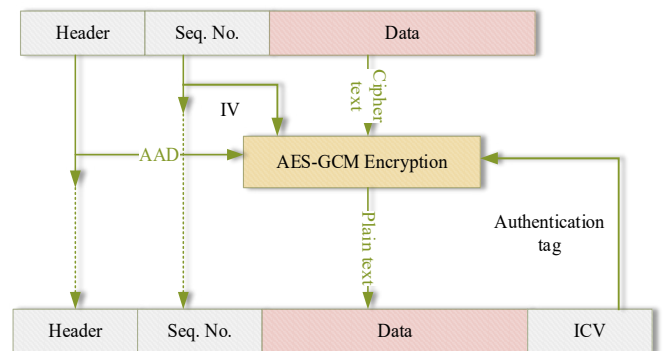


Fig.8. AES GCM Encryption Mode

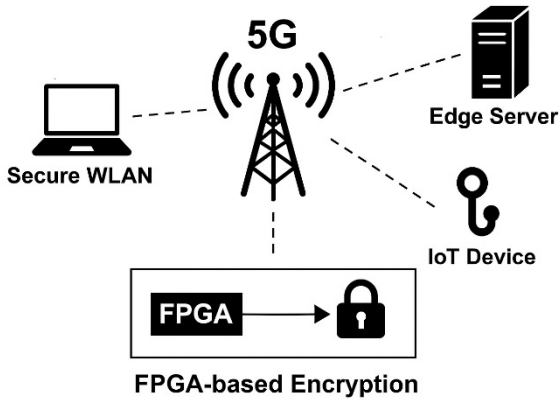


Fig 9. The schematic diagram illustrates the enhanced round key generation process within the improved SIT encryption algorithm.

The above diagram illustrates how the enhanced Secure IoT (SIT) encryption algorithm with improved round key generation can be integrated within various wireless communication environments. It shows data flow from IoT edge devices through secure encryption modules before transmission over networks such as secure WLAN, 5G/6G, and cognitive radios. The figure emphasizes real-time encryption efficiency and compatibility with constrained-resource nodes, making it suitable for modern smart communication ecosystems.

RESULTS & DISCUSSION

With the use of AES, Serpent, and ECC, this study proposes a new method for healthcare IoT systems. Healthcare data is better protected by the suggested system. When encrypting data using the suggested approach, the symmetric keys are encrypted using the current timestamp. This study provides evidence of the effectiveness of the suggested strategy via security analysis and performance comparisons. The results prove without a doubt that the suggested hybrid encryption technique outperforms the current gold standard.

The algorithm's performance is evaluated by applying the idea to the MATLAB simulator. This yields the results of the technique that is most suitable for the encryption protocol. In the following experiments, we tested this method in action, evaluating its efficacy in terms of processing time, data size, and the various encryption and decryption techniques we used. Results from a basic encryption technique with the key method for evaluating computation results were used in the Matlab Simulation test of utilizing the IoT features of implementing sky mote. The relationship between encryption time and data packet size is clearly positive. Both encryption

techniques take about the same amount of time to decrypt a packet with a 2M size. The suggested approach in AES and ECC slows down encryption performance as packet size rises, however.

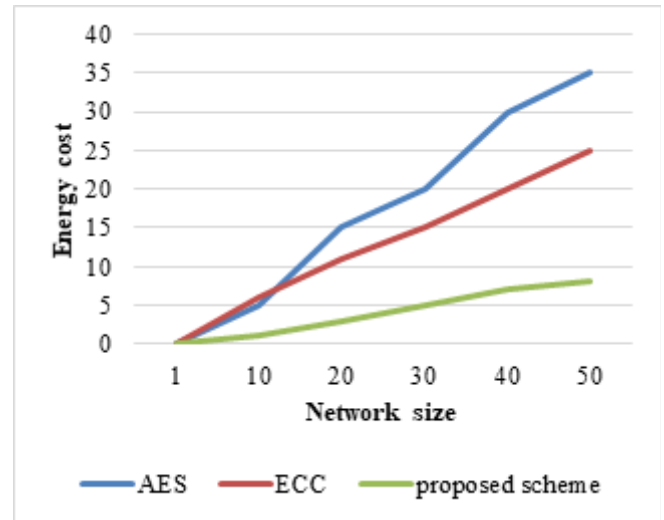


Fig.10. Energy Cost Modeling

In order to obtain a good degree of security, the available literature uses the ECC encryption technique for all plaintext. This study discusses the simultaneous use of ECC and AES for message encryption. Even though AES has much worse security than the ECC technique, this work utilizes AES to encrypt part of the plaintext communications and the keys to AES with a lower degree of security and ECC to encrypt part of the plaintext messaging with AES keys with a greater level of security. An adversary may decipher an AES-encrypted communication, but they will only be able to decipher about half of the information. As a result, the suggested algorithm's security is quite similar to the proposed algorithm. By combining the findings of the simulations with the method improvements presented in this research, it is possible to accomplish quicker calculation while maintaining the same level of security as the approach in reference.

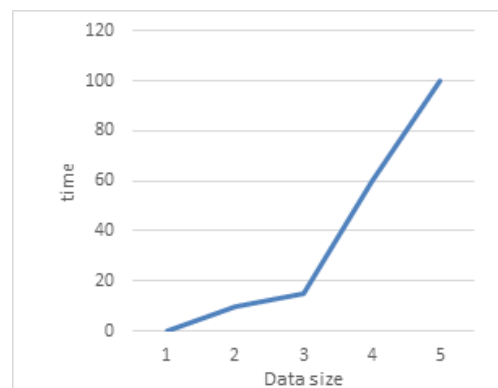


Fig.11: Key Generation Time

where devices need to communicate securely without draining their limited power or processing capabilities. On top of that, this encryption scheme is highly suitable for implementation in hardware, such as FPGAs or ASICs. That means it can be built directly into devices for faster, real-time encryption with lower energy use—perfect for applications in remote or energy-constrained environments. Looking ahead, building a prototype using an FPGA could help showcase just how scalable and responsive this method is for real-time use. In short, the proposed enhancement not only strengthens data protection but also aligns well with the needs of future IoT ecosystems, offering a practical and secure solution for the next generation of connected technologies.

REFERENCES

1. Kapalova, N., Algazy, K., Haumen, A., & Sakan, K. (2023). Statistical analysis of the key scheduling of the new lightweight block cipher. *International Journal of Electrical and Computer Engineering (IJECE)*.
2. Sindhu, N., & Vijaykumar, P. M. (2015). High Capacity Image Steganography Technique Based on Four Band Wavelet Transform. *International Journal of Advances in Engineering and Emerging Technology*, 6(1), 36-48.
3. Tong, Q., Miao, Y., Liu, X., Choo, K.R., Deng, R.H., & Li, H. (2022). VPSL: Verifiable Privacy-Preserving Data Search for Cloud-Assisted Internet of Things. *IEEE Transactions on Cloud Computing*, 10, 2964-2976.
4. Fakiha, B. (2023). The Role of Raspberry Pi in Forensic Computer Crimes. *Journal of Internet Services and Information Security*, 13(4), 76-87. <https://doi.org/10.58346/JISIS.2023.14.005>
5. Chanal, P.M., & Kakkasageri, M.S. (2022). Secured Data Integrity Scheme for Internet of Things. *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, 1-5.
6. Dayi, A.K., Rodoplu, V., Nakip, M., Pehlivan, B., & Güzelis, C. (2022). Multi-Channel Subset Iteration with Minimal Loss in Available Capacity (MC-SIMLAC) Algorithm for Joint Forecasting-Scheduling in the Internet of Things. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 13(2), 68-95. <https://doi.org/10.22667/JOWUA.2022.06.30.068>.
7. Chanal, P.M., & Kakkasageri, M.S. (2021). Preserving Data Confidentiality in Internet of Things. *SN Computer Science*, 2.
8. Turan, C., Ayas, D., Doğdu, S. A., & Ergenler, A. (2022). Extension of the striped eel catfish *Plotosus lineatus* (Thunberg, 1787) from the eastern Mediterranean coast to the Mersin Bay on the western Mediterranean coast of Turkey. *Natural and Engineering Sciences*, 7(3), 240-247.
9. <http://doi.org/10.28978/nesciences.1183740>
10. Okey, O.D., Maidin, S.S., Lopes Rosa, R., Toor, W.T., Carrillo Melgarejo, D., Wuttisittikulij, L., Saadi, M., & Zegarra Rodríguez, D. (2022). Quantum Key Distribution Protocol Selector Based on Machine Learning for Next-Generation Networks. *Sustainability*.
11. Pundir, M., Kumar, A., & Choudhary, S. (2023). Efficient Diffie Hellman Two Round Secret Key Agreement Protocol. *2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP)*, 7-10.
12. Xing, Y., & Li, S. (2021). A Compact Hardware Implementation of CCA-Secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021, 328-356.
13. Rajawat, A.S., Bedi, P., Goyal, S.B., Shukla, P.K., Jamal, S.S., Alharbi, A.R., & Aljaedi, A. (2021). Securing 5G-IoT Device Connectivity and Coverage Using Boltzmann Machine Keys Generation. *Mathematical Problems in Engineering*.
14. Rahman, Z., Yi, X., Khalil, I., & Sumi, M.A. (2021). Chaos and Logistic Map based Key Generation Technique for AES-driven IoT Security. *IACR Cryptology ePrint Archive*.
15. Devi, P., Sathyalakshmi, S., & Subramanian, D.V. (2021). An optimal metaheuristic optimization based ElGamal public key cryptosystem for privacy in IoT environment. *International Journal of System Assurance Engineering and Management*.
16. Khandani, A.K. (2023). Looping for Encryption Key Generation Over the Internet: A New Frontier in Physical Layer Security. *2023 Biennial Symposium on Communications (BSC)*, 59-64.
17. M, S., K, M.L., & G M, M.S. (2023). Enhanced AES-256 cipher round algorithm for IoT applications. *The Scientific Temper*.
18. Chen, L., Cao, K., Lu, T., Lu, Y., & Hu, A. (2022). A one-time pad encryption scheme based on efficient physical-layer secret key generation for intelligent IoT system. *China Communications*, 19, 185-196.
19. Shehab, S. (2022). Protein Key Generation for Secure CT-Chest Images Encryption. *International Journal of Intelligent Computing and Information Sciences*.
20. Chamkhia, H., Al-Ali, A.K., Mohamed, A., Guizani, M., Erbad, A., & Refaey, A. (2021). Performance Analysis of PLS key generation-based Secure NOMA-enabled IoT Networks in the presence of Untrusted Users. *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 633-638.
21. Albakri, A., Alshahrani, R., Alharbi, F., & Ahamed, S. (2023). Fully Homomorphic Encryption with Optimal Key Generation Secure Group Communication in Internet of Things Environment. *Applied Sciences*.
22. Chhabra, S., & Lata, K. (2022). Hardware Obfuscation of AES IP Core Using PUFs and PRNG: A Secure Cryptographic Key Generation Solution for Internet-of-Things Applications. *SN Computer Science*, 3.
23. Li, Q., Zhang, Q., Huang, H., Zhang, W., Chen, W., & Wang, H. (2022). Secure, Efficient, and Weighted Access

- Control for Cloud-Assisted Industrial IoT. *IEEE Internet of Things Journal*, 9, 16917-16927.
24. Guo, D., Cao, K., Xiong, J., Ma, D., & Zhao, H. (2021). A Lightweight Key Generation Scheme for the Internet of Things. *IEEE Internet of Things Journal*, 8, 12137-12149.
 25. Devi, P., Sathyalakshmi, S., & Subramanian, D.V. (2021). An optimal metaheuristic optimization based ElGamal public key cryptosystem for privacy in IoT environment. *International Journal of System Assurance Engineering and Management*.
 26. Kaleem, W., Sajid, M., & Rajak, R. (2023). Salp Swarm Algorithm to solve Cryptographic Key Generation problem for Cloud computing. *International Journal of Experimental Research and Review*.
 27. Huang, J., Chen, G., Guo, X., Lin, Z., Zeng, J., Shao, W., & Lin, D. (2023). Terminal Trusted Authentication Technology for 5G Intelligent Lamp Poles Based on an Improved AES Algorithm Combining Random Key Generation and Clock Synchronization. 2023 42nd Chinese Control Conference (CCC), 6244-6249.
 28. Song, W., Liu, M., Baker, T., Zhang, Q., & Tan, Y. (2023). A group key exchange and secure data sharing based on privacy protection for federated learning in edge-cloud collaborative computing environment. *International Journal of Network Management*, 33.
 29. Tian, W., Liu, Q., Li, R., Xu, Z., Zhang, Y., & Huang, Y. (2023). A Blockchain-Based Secure Searching Strategy for Metadata in Mobile Edge Computing. *IEEE Internet of Things Journal*, 10, 19795-19809.
 30. Del Prete, S., Fuschini, F., & Barbiroli, M. (2022). A Study on Secret Key Rate in Wideband Rice Channel. *Electronics*.
 31. Farooq, S., Altaf, A., Iqbal, F., Thompson, E.B., Vargas, D.L., Díez, I.D., & Ashraf, I. (2023). Resilience Optimization of Post-Quantum Cryptography Key Encapsulation Algorithms. *Sensors (Basel, Switzerland)*, 23.
 32. Mahlake, N., Mathonsi, T.E., Plessis, D.D., & Muchenje, T. (2023). A Lightweight Encryption Algorithm to Enhance Wireless Sensor Network Security on the Internet of Things. *J. Commun.*, 18, 47-57.
 33. Du, H., Wang, J., Niyato, D.T., Kang, J., Xiong, Z., Guizani, M., & Kim, D.I. (2022). Rethinking Wireless Communication Security in Semantic Internet of Things. *IEEE Wireless Communications*, 30, 36-43.
 34. Jiang, L., & Cui, H. (2023). Private and Mutual Authentication Protocols for Internet of Things. *Mathematics*
 35. Chanal, P.M., & Kakkasageri, M.S. (2021). Preserving Data Confidentiality in Internet of Things. *SN Computer Science*, 2, 1-12.
 36. Kavitha, M. (2023). Beamforming techniques for optimizing massive MIMO and spatial multiplexing. *National Journal of RF Engineering and Wireless Communication*, 1(1), 30-38. <https://doi.org/10.31838/RFMW/01.01.04>
 37. Dorofte, M., & Krein, K. (2024). Novel approaches in AI processing systems for their better reliability and function. *International Journal of Communication and Computer Technologies*, 12(2), 21-30. <https://doi.org/10.31838/IJCCTS/12.02.03>
 38. Udhayakumar, A., Ramya, K. C., Vijayakumar, P., Sheeba Rani, S., Balamanikandan, A., & Saranya, K. (2024). Reversible Vedic Direct Flag Divider in Key Generation of RSA Cryptography. *Journal of VLSI Circuits and Systems*, 6(2), 75-83. <https://doi.org/10.31838/jvcs/06.02.08>
 39. Barhoumi, E. M., Charabi, Y., & Farhani, S. (2024). Detailed guide to machine learning techniques in signal processing. *Progress in Electronics and Communication Engineering*, 2(1), 39-47. <https://doi.org/10.31838/PECE/02.01.04>
 40. Choset, K., & Bindal, J. (2025). Using FPGA-based embedded systems for accelerated data processing analysis. *SCCTS Journal of Embedded Systems Design and Applications*, 2(1), 79-85.