JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Hybrid Cryptographic Framework for Secure IoT Communication Integrating AES-ECDH and SHA-512

[1]Libi Kurian *, [2]Dr. Sumitra P.

***Email:** libi.kurian@yahoo.com

[1,2] PG & Research Department of Computer Science and Applications,

[1,2] Vivekanandha College of Arts and Sciences for Women (Autonomous), Tamil Nadu, India

Abstract

Wireless Sensor Networks (WSNs) play a crucial role in enabling IoT applications by facilitating data collection, monitoring and communication. However, ensuring security remains a major challenge in WSN, as in all types of networks. The resource -constrained nature of sensor nodes and their deployment in open and often unattended environments, WSNs are highly vulnerable to security threats. Attackers may physically tamper with sensor nodes, eavesdrop on transmitted data, inject false information or modify legitimate communications, thereby threatening data confidentiality, integrity and availability. To protect from such harms, it is required to provide security to sensitive data and check integrity in order to verify if the data has been modified. To improve energy efficiency, data security, and network lifetime, we propose a hybrid cryptographic framework in this article that combines Diffie-Hellman Key Exchange (DHKE)-based key establishment with Advanced Encryption Standard (AES)-based data encryption. SHA-512 is used for integrity purposes along with clustering through the standard LEACH protocol. AES's key exchange problem has been resolved by the suggested algorithm. The framework minimises energy usage by confining asymmetric cryptographic operations to the startup phase while ensuring confidentiality, integrity and forward secrecy.

Keywords: Diffie-Hellman Key Exchange (DHKE), Clustering, cryptography, Encryption, Decryption, Advanced Encryption Standard (AES), Computation Time, Energy Consumption, Internet of Things (IoT), LEACH, Packet Delivery Ratio (PDR), Secure Hash Algorithm (SHA), Security, Wireless Sensor Networks (WSNs).

1. INTRODUCTION

The Internet of Things (IoT) has become known as a disruptive technology, permitting a network of millions of devices that share and exchange information, thereby supporting applications in healthcare, smart cities, industrial automation, as well as environmental monitoring. [1]. WSNs are essential to IoT applications, that feature distributed sensing, collection of data, and communication capabilities.

Furthermore, the sensor nodes within the network possess constrained capabilities, including reduced memory, reduced computing power, and limited battery life. Traditional security methods, when applied to resource-constrained environments that include IoT devices, embedded systems, as well as WSNs, often encounter substantial challenges due to their significant computational power requirements, rendering them ineffective and vulnerable to security threats.[2].

Researchers have investigated numerous approaches to secure data transmission; however, data encryption remains the most widely adopted and effective technique. Encryption works by scrambling data into an unreadable format, ensuring that only authorised recipients can access the original information. In essence, encryption transforms plaintext into ciphertext, which can be interpreted only by users possessing the appropriate decryption key.

As cryptanalysis techniques have evolved over time, encryption mechanisms have also become increasingly sophisticated to preserve data confidentiality. The main encryption methods are symmetric-key and asymmetric-key. Data is encrypted and decrypted with the same secret key in symmetric-key cryptography. Conversely, symmetric-key cryptography utilizes two separate yet mathematically interconnected keys: a Public key for encryption and a Private key for decryption. Every method possesses distinct advantages and limits.

Symmetric-key encryption is recognized for its rapidity, simplicity, and minimal computational expense, rendering it appropriate for resource-limited settings. Nevertheless, due to its dependence on a singular shared key, it is relatively less secure, especially in key distribution contexts. Asymmetric encryption methods, on the other hand, are more computationally intensive and complex, but they offer a higher level of security by eliminating the need for a shared secret key.

Two prevalent symmetric encryption algorithms are the “Data Encryption Standard (DES)” and the AES. Asymmetric key cryptography, often called as public-key cryptography, employs distinct key pairs for secure communication. Prevalent asymmetric encryption methods encompass RSA and algorithms based on “Elliptic Curve Cryptography (ECC)”, including the “Elliptic Curve Digital Signature Algorithm (ECDSA)” and the “Elliptic Curve Diffie-Hellman (ECDH)” algorithm. ECC is esteemed for its robust security assurances with reduced key dimensions and is often employed in Diffie-Hellman key exchange protocols [5-8].

Symmetric key cryptography is vulnerable to attacks primarily due to challenges in key security and management. To address these key management issues, asymmetric -key cryptography algorithms were introduced that offer a higher level of security by using separate encryption/decryption keys; however, it suffers from slow execution speed and higher computational overhead. To overcome the limitations of both approaches, many research efforts have focussed on combining the strengths of symmetric and asymmetric cryptography. This integrated method, termed Hybrid Cryptography, employs the efficacy of symmetric encryption for data transmission while utilizing asymmetric encryption for safe key exchange, thus attaining both strong security and improved efficiency.

Hence, this study proposes a hybrid cryptographic framework that integrates the AES (Advanced Encryption Standard) based encryption mode along with the (Secure Hash Algorithm) SHA-512 for ensuring confidentiality, integrity and authentication and Diffie-Hellman Key Exchange (DHKE)-based key establishment for secure key transmission. The proposed framework has also used the LEACH-based cluster protocol in this study [9], [10]. This protocol ensures that the sensor nodes in the network send the collected data to the fewer sensor nodes (i.e., cluster heads) to handle the communication. This reduces the use of the energy and improves the data transfer capability.

The Key contribution of this article is summarised as follows:

- An Advanced Encryption Standard (AES) encryption and decryption technique is utilized to ensure secure and confidential information is transmitted throughout the network.
- Diffie–Hellman Key Exchange (DHKE) facilitates secure key generation and exchange, allowing for the development of a shared secret key for encryption without the necessity of prior key distribution.
- SHA-512, a cryptographic one-way hash function, is employed to produce a message digest that validates the integrity and validity of transmitted data.
- An energy-efficient clustering approach is implemented for WSNs. Choosing suitable cluster heads reduces power consumption and extends network longevity. The “LEACH (Low Energy Adaptive Clustering Hierarchy)” protocol is utilized to structure sensor nodes into clusters and enable effective routing.

This paper is organized: Section 2 reviews existing research literature. The proposed methodology is detailed in Section 3. Section 4 presents results and discussion, and Section 5 concludes this paper.

2. LITERATURE REVIEW

Many researchers are primarily focusing on making the communication more secure and efficient in WSNs particularly in the evolving domains like Internet of Things (IoTs). Because these kinds of networks are made up of small devices that have the ability to send or receive the data safely. Some of the traditional methods like

RSA [11], Elliptic Curve Cryptography (ECC) [12], [13]. The RSA and ECC provide robust public-key operations. The Advanced Encryption Standard (AES) [14] have been able to provide strong security to the regular computer systems. These methods are good at keeping the data safe from unauthorized access and hackers. Moreover, these traditional methods have required more computing power, memory and more energy usage in order to provide security to the computer systems.

The study [15] conducts a comparative, software-only empirical evaluation of SHA-256 and the SHA-3 Kangaroo Twelve (K12) variation across three generations of IoT hardware: Sky Mote, Z1 Mote, and Wismote. The SHA-3 Kangaroo Twelve variation achieves the requirements for post-quantum durability and adaptability in resource-limited settings, primarily suitable for high-capacity implementations, although it fails to optimize the security-efficiency balance in operating IoT networks.

[16] emphasis that strong public-key cryptography (specifically ECC) is viable on resource-constrained 8-bit devices without the need for hardware acceleration. By optimizing software-level multiplication and leveraging specific curve properties, the researchers demonstrated that Elliptic Curve Cryptography (ECC) significantly outperforms RSA on small-word-size processors. Despite the performance gains, several limitations and trade-offs persist for implementing public-key cryptography on 8-bit microcontrollers.

Elliptic Curve Cryptography (ECC) is significantly more efficient as well as suitable for resource-constrained IoT environments than RSA. While ECC offers superior performance in terms of resource management, certain limitations and specific trade-offs still remain [17]. Selecting the suitable elliptical curve is not simple. ECC performance is highly dependent on the specific platform and the curve chosen.

This work [18] presents an asymmetric encryption algorithm utilizing Elliptic Curve Cryptography (ECC), that simplifies key management and transmission relative to symmetric systems. The technology integrates Elliptic Curve Cryptography (ECC) with a “Piece-Wise Linear Chaotic Map (PWLCM)” and SHA-256. To mitigate the sluggishness of asymmetric encryption, the technique aggregates several pixel values into large integers. The approach is based on the “Elliptic Curve Discrete Logarithm Problem (ECDLP)”, offering superior security with reduced key sizes compared to RSA. Despite the "big integer" grouping method, the total number of cryptographic operations remains quite high in comparison to just symmetric chaotic encryption.

A new lightweight encryption strategy has been proposed [19] that focuses on securing Wireless Sensor Networks (WSNs) using a lightweight approach that combines chaotic maps and genetic operations. Integrating chaotic maps with genetic operations (mutation and crossover) provides a robust, low-complexity security solution specifically tailored for resource-constrained WSN nodes. While symmetric systems are faster, they rely on secret key agreements that remains a critical challenge in WSN security. Chaotic maps exhibit heightened sensitivity to initial settings; if these parameters (keys) are violated or lack true randomness, the predictability of the sequence increases.

The authors employed “Chaos-based Secured Communication in Energy Efficient Wireless Sensor Networks” by integrating chaotic encryption with the “LEACH” routing system. Combining chaotic encryption with a clustered routing protocol (LEACH) allows WSNs to achieve high-level data security without significantly compromising the network's energy efficiency or lifetime. The practical challenge of securely distributing and updating the chaotic "keys" (initial parameters) across a large, remote network remains a complex issue.

The finding [21] is that all chaotic encryption algorithms using constant keys are highly vulnerable to attack and argues that chaotic systems must move away from fixed parameters to dynamic, non-linear functions.

3. PROPOSED METHODOLOGY

The proposed framework employs a hybrid cryptographic approach where ECDH is used for secure session key establishment, and AES is employed for lightweight data encryption. The asymmetric operations are restricted to initialize and rekeying phases, thereby significantly reducing energy overhead while ensuring confidentiality, integrity and forward secrecy in clustered WSNs.

3.1 LEACH clustering

The suggested technique uses clustering to improve network management. This technique groups sensor nodes and uses the “LEACH (Low-Energy Adaptive clustering Hierarchy)” protocol to dynamically select a cluster-head (CH) for each cluster (Figure1). Within LEACH, nodes from local clusters where each CH is responsible for collecting data from its member nodes, performing necessary processing, as well as transmitting the aggregated information to the remote base station.

Consequently, cluster-head nodes use far more energy than non-clusters. If cluster-hea sensor nodes were stationary as in static clustering techniques, the assigned nodes would quickly exhaust their finite energy resources, resulting in network failure. To resolve this issue, LEACH utilizes a rotational cluster-head selection mechanism, in which the cluster-head role is regularly redistributed among the cluster members. This technique uniformly distributes energy load across the network, hence prolonging overall network longevity and enhancing reliability [22]. By integrating LEACH-based clustering with the proposed Hybrid Cryptography approach, the scheme effectively ensures data security and user authentication while maintain energy efficiency in the network.

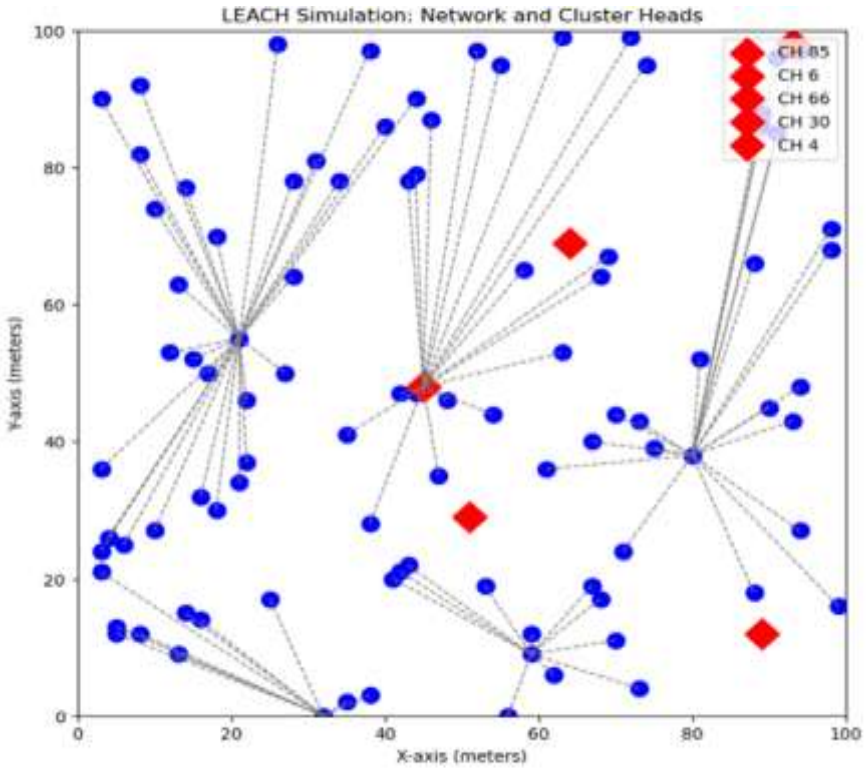

Fig.1. Simulation setup for 100 nodes of LEACH based clustering

3.2 Advanced Encryption Standard (AES)

The AES is a symmetric encryption technique utilizing a single key, providing robust protection against various assaults, including brute-force attacks that necessitate the examination of all possible key combinations until the correct one is identified.Compared to DES and 3DES, AES provides significantly higher security and efficiency because of its larger key sizes and faster execution speed—being approximately six times faster than 3DES. The "substitution–permutation network (SPN)" architecture of AES offers key lengths of 128, 192, and 256 bits and processes data through several rounds of substitution, permutation, as well as mixing to provide strong encryption and superior performance. The strength of AES encryption relies on the combination of these operations over multiple rounds, confirming strong diffusion and confusion of data.[23]

Process of AES

AES block cipher encryption uses cryptographic keys to encrypt and decrypt 128-bit blocks. The key sizes have been considered appropriate for protecting classified information to a satisfactory degree of confidentiality [23].

Steps for AES Encryption Process are as follows:

1. **Input Plaintext Data Block** (128 bit) If the data exceeds 128 bits, it is segmented into numerous blocks for independent processing)
2. Derive the set of round keys from the cipher key (AES Key expansion)
3. **Initial Round**- Incorporate Round Key: Before any subsequent modifications, the plaintext block undergoes an XOR operation with the initial round key, which is produced from the primary encryption key).
4. **Main Rounds** (Repeating Steps): Each main round consists of four transformations:
    a. **SubBytes:** S-box-based non-linear substitution to replace each byte with a value.
    b. **ShiftRows:** A permutation step where rows of the block are cyclically shifted to the left.

c. **MixColumns:** A linear transformation that mixes the bytes inside every row for diffusion (not performed in the last round).
d. **Add RoundKey:** XOR operation between the current block and the round key for encryption. The quantity of principal rounds is dependent upon the AES key size: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key.

5. **Final Round:** It includes the same steps as the main rounds, except for the MixColumns operation. This ensures that the ciphertext remains 128 bits in size.
6. **Output Ciphertext Block**: After all rounds are complete, the final block is the encrypted ciphertext.

AES Decryption Process

Inverse operations are used in the decryption process, which follows the same structure as encryption: Inverse of SubBytes, Inverse of ShiftRows, Inverse of MixColumns, Add RoundKey: Same as in encryption.

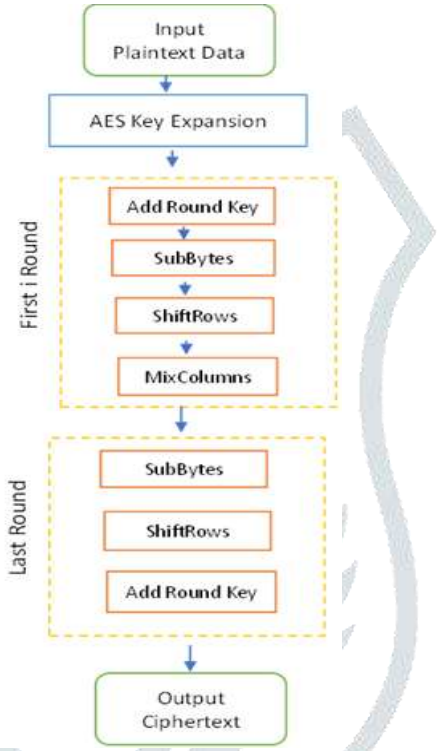

Fig.2. Block diagram of AES Encryption process

3.3 Elliptic Curve Diffie–Hellman (ECDH)

Deterministic key generation techniques produce cryptographic keys from a fixed-length secret and public parameters, guaranteeing consistent reproduction of the same key across many devices. By obviating the necessity for persistent high-entropy randomization, these methods markedly diminish computational overhead, rendering them optimal for resource-limited systems. The ECDH technique is a notable deterministic key generation method. ECDH enables the efficient generation of shared secrets for secure communication, providing a strong equilibrium between elevated security and low processing demands [24].

Elliptic curve Diffie-Hellman lets two users safely generate a shared secret key through an unsafe channel. With the exception of using ECC point multiplication rather than modular exponentiations, ECDH is fairly similar to the traditional "DHKE (Diffie–Hellman Key Exchange)" algorithm. The following characteristic of EC points serves as the foundation for ECDH:

$$(a * G) * b = (b * G) * a \qquad \text{Eq. (1)}$$

Where, two secret numbers a and b (private keys) belong to sender and receiver and G is the ECC elliptic curve generator point. Selecting appropriate elliptic curves for ECC often give a reasonable compromise between security and performance. Compared to typical ECC curves, the key generation method on these curves needs fewer calculations, which reduces processing time and memory use [25].

The public key values (a*G) and (b*G) of sender and receiver can be exchanged over an insecure channel.

Then we can derive a Shared Secret Key = (a * G) * b = (b * G) * a.

Secret Shared=Sender_PubKey * Receiver_PrivKey = Receiver_PubKey * Sender_PrivKey

The shared Secret key is computed on either side of the two parties and it is compared to verify whether it is same.

The "ECDH algorithm (Elliptic Curve Diffie–Hellman Key Exchange)" works as follows:

1. Sender generates a random ECC key pair: {S_PrivKey, S_PubKey = S_PrivKey * G}
2. Receiver generates a random ECC key pair: {R_PrivKey, R_PubKey = R_PrivKey * G }
3. Sender and Receiver exchange their public keys through the insecure channel (e.g. WSN network)
4. Sender calculates sharedKey = R_PubKey * S_PrivKey
5. Receiver computes sharedKey = S_PubKey * R_PrivKey
6. Now both Sender and Receiver have the same SharedKey == R_PubKey * S_PrivKey == S_PubKey * R_PrivKey.

3.4 Hybrid ECDH–AES Algorithm

The proposed Hybrid ECDH + AES secure framework for WSN is divided into two phases Key generation and management, and Secure Encryption/Decryption.

Phase I: Key Generation and Secure Key Establishment (ECDH)

The key is the most crucial part of encoding or decoding. An adversary discovering this key threatens data security. Key secures data overall.

The features of the key generation and management process are explained below:

1. An ECDH Key Pair is generated by the two communicating parties (Sender and Receiver) {PrivKey, PubKey = PrivKey * G}, where G is an elliptic curve base point.
2. Sender and Receiver exchange their public keys over the insecure channel. Private keys are never transmitted.
3. Both Sender and Receiver compute the same shared secret key

   Shared Secret (SS) = S_PubKey * R_PrivKey = R_PubKey * S_PrivKey

   Any attacker without the private key cannot compute Shared Secret key (SS).
4. AES Session Key Derivation is done using SHA-512. The raw shared secret is not employed directly as an encryption key. Instead, SHA-512 is applied as a Key Derivation Function (KDF):

   AES_Key = SHA-512 (SS || Node ID || Session ID)

   Where, SS is the ECDH shared secret key, Node ID is the unique sensor identity and Session ID is the LEACH round / timestamp. AES_Key is a uniform, unpredictable 128-bit key. This session key prevents key reuse and correlation attacks.

Phase II: Secure Data Encryption at Sensor Node (AES + SHA-256) and Decryption at Receiver.

After the key generation and management process, the key-encryption process begins as follows:

1. Data Sensing and Nonce Generation: The sensor node collects data and a Nonce or counter is generated to ensure freshness. This prevents reply attacks.
2. AES-128 encryption is carried out for data encryption and produce a scrambled data known as cipher text.

   Ciphertext=AES_Encrypt (AES_Key, Data, Nonce)

   AES provides confidentiality with minimal energy consumption
3. Message Authentication Using SHA-512: To ensure data integrity and authenticity a MAC is calculated which helps to detect packet tampering and protects against forgery.

   MAC = SHA-512 (AES_Key || Ciphertext || Nonce)
4. Transmitted packet contains Ciphertext, Nonce and MAC. (AES_Key is never transmitted.)
5. A secure encrypted packet is received at the Receiver side for Decryption.
6. Freshness is verified by checking the Nonce or counter. Old or duplicate nonces are rejected to prevents replay attacks.
7. Integrity Verification (SHA-512): The receiver recomputes MAC '

   MAC'= SHA-512 (AES_Key || Ciphertext || Nonce)

   If MAC' is not equal to MAC, packet is discarded. This guarantees integrity and authentication.

8. If verification succeeds AES Decryption is performed.
   Data = AES_Decrypt (AES_Key, Ciphertext, Nonce)
   Plaintext is recovered.
9. Rekeying in WSN: Key update (rekeying) means periodically generating a new AES session key instead of using the same key for a long time. To generate a new shared secret, ECDH key exchange is repeated and SHA-512 is employed to generate an AES key. Old keys are discarded.
   - Rekeying is initiated when LEACH round changes or Cluster Head changes or Session timeout occurs or when Intrusion is detected.
   - The purpose of key update and rekeying is to periodically refresh cryptographic keys to limit the impact of node compromise, provide forward secrecy, and maintain secure communication under dynamic WSN topology changes.

Figure 3 illustrates the use of the proposed hybrid ECDH–AES cryptography method to data for the goal of ensuring data security. The steps undertaken in the procedure are illustrated in the figure.

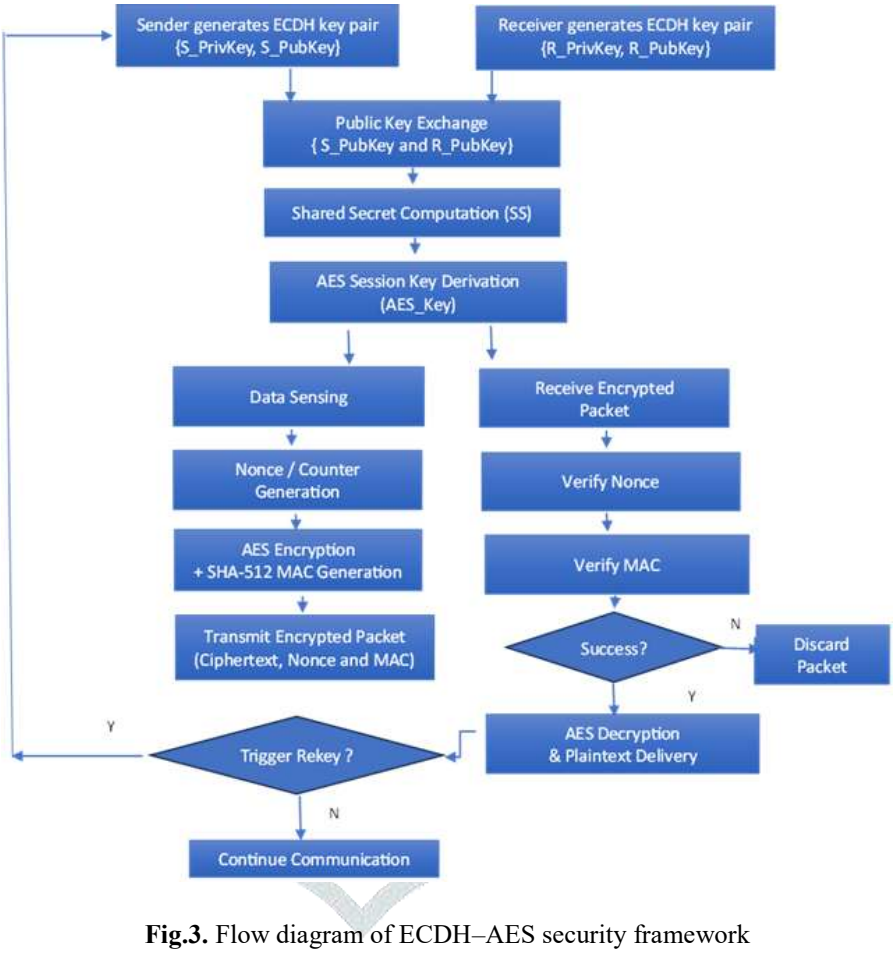

Fig.3. Flow diagram of ECDH–AES security framework

The security of this developed model has been analysed by how well it performs against common cyber-attacks and the data integrity. Table 1 gives the security properties achieved by the Hybrid ECDH–AES Algorithm.

Table1. Security properties of Hybrid ECDH–AES

Confidentiality	AES
Secure key exchange	ECDH
Integrity	SHA-512
Authentication	SHA-512 MAC
Forward secrecy	ECDH rekeying
Replay protection	Nonce
Scalability	Cluster-based design

4. RESULTS AND DISCUSSION

This section discusses the performance evaluation of the proposed Hybrid ECDH–AES algorithm. The simulation was carried out for the 10 simulation rounds in the created environment. PDR, Energy Consumption, and Computation Time show that the technique is acceptable for secure communication in resource-constrained IoT contexts.

Symmetric block cryptography has been employed to compare the proposed method to SPN and Feistel. Each approach generates a unique safe key to encrypt data.

The proposed model ran for 10 simulation rounds and it achieved significant results. Each of the key metric results is explained below:

1. **Packet Delivery Ratio (PDR):** The PDR has achieved an average score of 99.47%. This shows how reliable and efficient the messages are transferred from the sender/input to the receiver/output.
2. **Energy Consumption:** The energy consumption reminds of the average score of 24.32 uJ for every round even with the dynamic cluster head rotation.
3. **Key expansion Time:** The average computation time taken by the proposed method is 0.0029922 seconds. This shows it takes less time, introduces minimal delay and makes it suitable for the low-power IoT devices having limited resources capability.

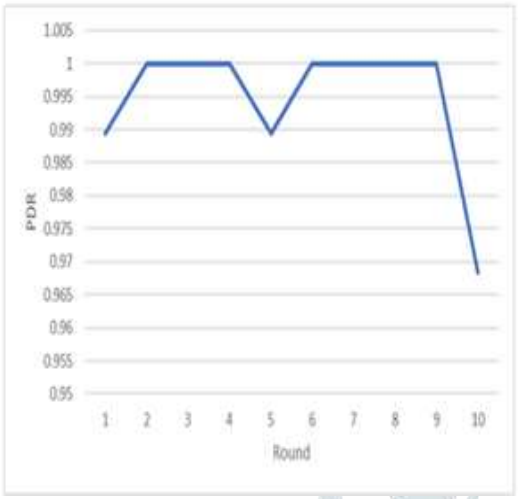

Fig.4. Proposed PDR

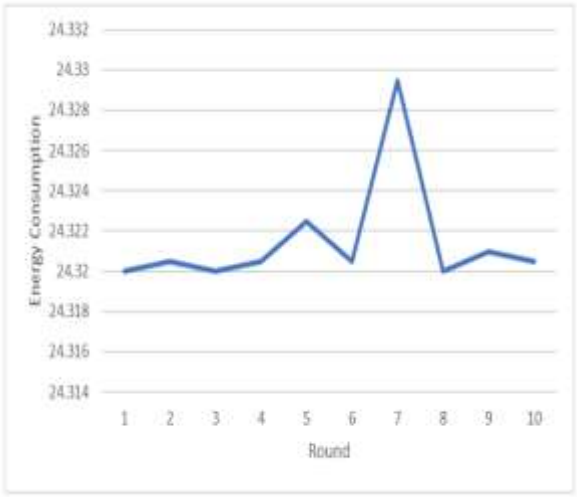

Fig.5. Proposed Energy Consumption

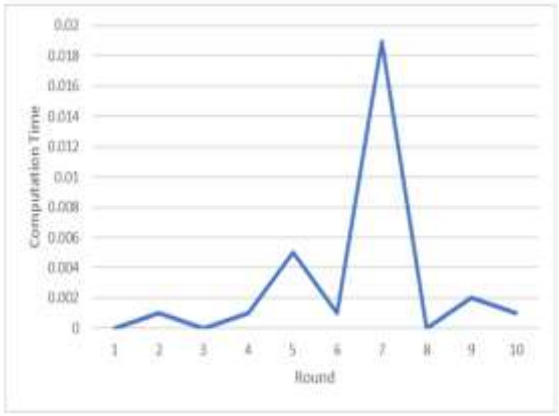

Fig.6. Proposed Computation time

Comparison Results

The proposed model in this study shown the effective performance and outperforms them in terms of key metrics like PDR, Energy consumption, Key Expansion Time when compared with the existing methods such as LSA, SPN and Feistel and comparison results are shown below:

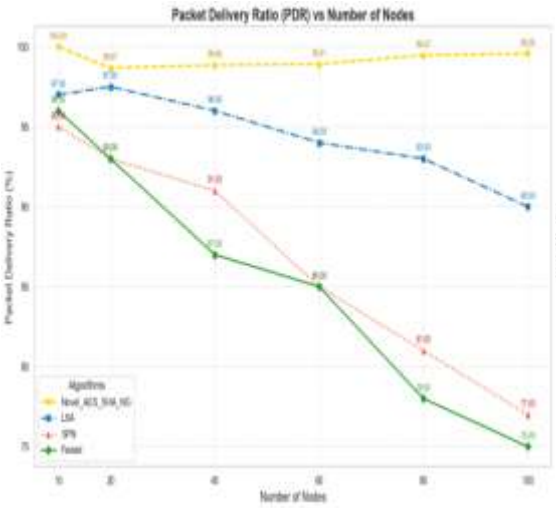

Fig.7. Packet Delivery Ratio

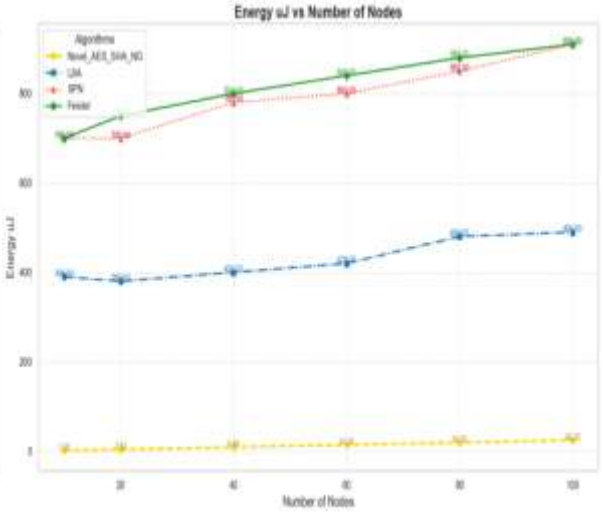

Fig.8. Energy uj

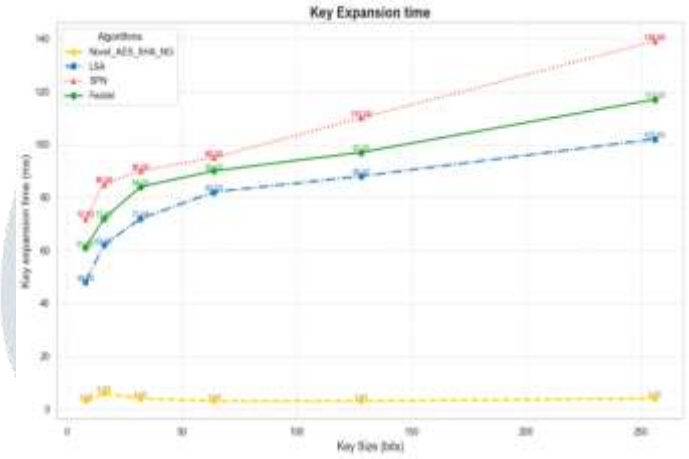

Fig.9. Key Expansion Time (ms)

5. CONCLUSION

The secure communication in the resource constrained IoT enabled WSNs is still a significant issue because of the limited energy, memory and computational capability. In order to address the issues, this study proposes a, novel framework that integrates the AES-ECDH encryption with the SHA-512 hashing. This Combination ensures the data confidentiality, integrity and authentication while achieving the highest delivery reliability, computational overhead, better security and reduced energy consumption. The performance of this proposed model was conducted in the simulated environment and analysed by using the key performance metrics like PDR, Energy Consumption, Computation Time. These results are compared with the other baseline schemes such as LSA, SPN and Feistel. By analysing the results, the proposed framework has achieved the PDR the average score of 99.47% while LSA of 93%, SPN of 81% and Feistel of 78%, the proposed framework is able to transmit more data with the minimal data loss. The average consumption of energy is recorded as 24.32 units per round while LSA, SPN and Feistel has achieved the highest values in energy consumption when compared to the proposed framework. The average computation time of 0.0029922 indicates that key generation and encryption/decryption need minimal time. Based on these findings the proposed framework has shown strong performance while maintaining the data integrity as well as confidentiality and also consuming the least amount of energy, quick in performing the actions. This makes it more suitable for IoT devices particularly for resource-constrained environments by providing secure communication.

REFERENCES

[1] T. Jabeen, I. Jabeen, H. Ashraf, N. Z. Jhanjhi, A. Yassine, and M. S. Hossain, "An intelligent healthcare system using IoT in wireless sensor network," Sensors, vol. 23, no. 11, Art. no. 5055, 2023.

[2] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Imran, and M. Ur-Rehman, "Challenges, applications, and future of wireless sensors in Internet of Things: A review," IEEE Sensors Journal, vol. 22, no. 6, pp. 5482-5494, 2022.

[3] G. Singh, Supriya, A study of encryption algorithms (RSA,DES, 3DES and AES) for information security, Int. J. Comput.Appl. 67 (19) (2013) 33–38.

[4] W. Burr, Selecting the advanced encryption standard, IEEE Secur. Priv. 1 (2) (2003) 43–52.)

[5] D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA), Int. J. Inf. Secur. 1 (1) (2001) 36-63.

[6] M. Balitanas, WiFi protected access-pre-shared key hybrid algorithm. Int. J. Adv. Sci. Technol., 2009, 12.

[7] M. Frunza, Gh. Asachi, Improved RSA encryption algorithm for increased security of wireless networks, in: ISSCS International Symposium, Vol. 2, 2007.

[8] R. Kodali, N. Sarma, Energy efficient ECC encryption using ECDH, Lecture Notes in Electrical Engineering, vol. 248, Springer, 2013, pp. 471–478).

[9] S. Bhatia, A. Kumar, A. Kumar, and A. S. Alshuhail, "Performance analysis of energy efficient improved LEACH protocol in IoT networks," IET Communications, vol. 19, no. 1, Art. no. e12500, 2025.

[10] B. Suresh and G. S. C. Prasad, "An energy efficient secure routing scheme using LEACH protocol in WSN for IoT networks," Measurement: Sensors, vol. 30, Art. no. 100883, 2023.

[11] M. M. Elamir, M. S. Mabrouk, and S. Y. Marzouk, "Secure framework for IoT technology based on RSA and DNA cryptography," Egyptian Journal of Medical Human Genetics, vol. 23, no. 1, pp. 116-, 2022.

[12] J. R. Arunkumar, S. Velmurugan, B. Chinnaiah, G. Charulatha, M. R. Prabhu, and A. P. Chakkaravarthy, "Logistic Regression with Elliptical Curve Cryptography to Establish Secure IoT," Computer Systems Science & Engineering, vol. 46, no. 1, 2023.

[13] A. E. Adeniyi, R. G. Jimoh, and J. Awotunde, "A review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security," 2024.

[14] R. S. Salman, A. K. Farhan, and A. Shakir, "Lightweight modifications in the Advanced Encryption Standard (AES) for IoT applications: a comparative survey," in 2022 International Conference on Computer Science and Software Engineering (CSASE), Mar. 2022, pp. 325-330.

[15] I. Baird, I. Wadhaj, B. Ghaleb, C. Thomson, and G. Russell, "Evaluating the Energy Costs of SHA-256 and SHA-3 (Kangaroo Twelve) in Resource-Constrained IoT Devices," IoT, vol. 6, no. 3, Art. no. 40, 2025.

[16] Gura, N.; Patel, A.; Wander, A.; Eberle, H. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. CHES 2004, 8, 119–132.

[17] Z. Vahdati, Sharifah M.Y., A. Ghasempour, M. Salehi, Comparison of ECC and RSA Algorithms in IoT Devices", Journal of Theoretical and Applied Information Technology, Vol.97. No 16 ,2019

[18] X. Zhang, X. Wang, Digital image encryption algorithm based on elliptic curve public cryptosystem, IEEE Access 6 (2018) 70025–70034.

[19] A. Dutta, K. Naveen Kumar, N. Sai, R.R. Chintala, An efficient light weight cryptography algorithm scheme for WSN devices using chaotic map and GE, Int. J. Pure Appl. Math. 118 (20) (2018) 861–875

[20] M.P. Nidarsh, M.G.P. Devi, Chaos based secured communication in energy efficient wireless sensor networks, Chaos 5 (6) (2018) 742–747

[21] M.I. Sobhy, A.-E. Shehata, Methods of attacking chaotic encryption and countermeasures, in: 2001 IEEE international conference on acoustics, speech, and signal processing, Salt Lake City, UT, 7–11 May 2001, pp.1001–1004. New York: IEEE.

[22] S. Urooja, S. Lata, S. Ahmad ,S. Mehfuz , S Kalathila, Cryptographic Data Security for Reliable Wireless Sensor Network, Alexandria Engineering Journal 2023,Vol. 72,pp. 37–50

[23] Madhumita Panda, Data Security in Wireless Sensor Networks via AES Algorithm, IEEE 9th International Conference on Intelligent Systems and Control (ISCO) 2015

[24] Madden, M.; Batina, L.; Gurkaynak, F.K.; Guajardo, J.; Sadeghi, A.R.; Tuyls, P. Diffie-Hellman over Minimal Extension Fields for IoT Devices. In Workshop on Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec); Springer: Berlin/Heidelberg, Germany, 2016; pp. 128–146.

[25] Gürkaynak, F.K.; Güneysu, T.; Paar, C. LECC: Lightweight Elliptic Curve Cryptography for RFID Tags and Sensor Nodes. In Workshop on Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec); Springer: Berlin/Heidelberg, Germany, 2014; pp. 113–127.