The Scientific Temper (2024) Vol. 15 (3): 2663-2671
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.34
E-ISSN: 2231-6396, ISSN: 0976-8653
https://scientifictemper.com/

RESEARCH ARTICLE

An asymmetric key encryption and decryption model incorporating optimization techniques for enhanced security and efficiency

Annalakshmi D.*, **C. Jayanthi**

Abstract

In wireless sensor networks (WSN), ensuring data security is crucial for maintaining the confidentiality and integrity of transmitted information. Asymmetric key encryption methods serve as fundamental tools in securing communication within WSNs. This paper introduces an innovative asymmetric key encryption and decryption model, integrating optimization techniques to enhance security and efficiency in data transmission within WSNs. By incorporating optimization algorithms into key generation and encryption processes, the proposed model strengthens cryptographic key robustness and reinforces encryption mechanisms against potential threats. Leveraging advanced optimization methodologies like genetic algorithms, simulated annealing, or particle swarm optimization, the model optimizes key parameters to mitigate vulnerabilities and bolster resistance against brute force and cryptanalysis attacks. Additionally, the model streamlines encryption and decryption procedures, optimizing computational resources and reducing associated overheads. Through experimental validation and performance analysis, the effectiveness of the proposed model is demonstrated by achieving improved security, reduced computational complexity, and enhanced data transmission efficiency. This research contributes to advancing WSN security by offering a sophisticated and efficient solution for safeguarding sensitive information in digital communication networks.

Keywords: Security, Cryptography, Encryption, Decryption, WSN, Optimization, Asymmetric key, Security threat.

Introduction

Group communication services, such as medical diagnosis systems through telemedicine, traffic management systems through video conferences and home security systems, are the most common applications of multicast communication where the messages sensed by the sensors are sent to the base station through a group of member nodes in wireless sensor networks. In this scenario, the group can be formed either by the local members of a cluster formed by sensor nodes with a cluster head for coordination among them or it can be formed with multiple clusters, where each cluster will act as a subgroup. Moreover, the sending option can be either local or global. If the sending option is local, then the sensor node can send the data only to the members of the same cluster. However, the cluster heads will be provided with global data communication privileges and hence, the cluster head nodes can perform intra-cluster communication as well as inter-cluster communication. Therefore, the cluster heads will be selected from the member nodes with high energy, minimum distance from member nodes and high security-based reputation based on high trust values Qazi, R., (2021); Tropea, M., (2022).

The members of a group can be either static or dynamic in nature. In static groups, the member nodes are given permanent membership to the group and it will not change for the entire duration of communication. In dynamic member based groups, the membership will be changing during the data communication process. Since the energy levels of nodes change, the cluster heads will change. If the new cluster head is far away from the member nodes, re-clustering will be performed. Moreover, the group consisting of the cluster heads will change often due to

PG & PG and Research, Department of Computer Science, Government Arts College (Autonomous) Affiliated to Bharathidasan University, Tiruchirappalli), Tamil Nadu, India.

***Corresponding Author:** Annalakshmi D., PG & PG and Research, Department of Computer Science, Government Arts College (Autonomous) Affiliated to Bharathidasan University,Tiruchirappalli), Tamil Nadu, India., E-Mail: poorna23.priya@gmail.com

How to cite this article: Annalakshmi, D., Jayanthi, C. (2024). An asymmetric key encryption and decryption model incorporating optimization techniques for enhanced security and efficiency. The Scientific Temper, **15**(3):2663-2671.

Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.34

Source of support: Nil

Conflict of interest: None.

Received: 05/07/2024 **Accepted:** 13/08/2024 **Published** : 10/09/2024

cluster head rotation. Therefore in a WSN, the groups are dynamic in nature since member join and member leave operations will be occurring continuously. The member joins or member leaves can be either a single-member operation where only a single member will join the group of cluster heads, or it can be multiple members who are joining the group or are leaving the group Mallick, B. B.,(2021), Mohindru, V., (2020), Gupta, S. C., (2021, March).

The security issues will become complex when there are changes in members who are making the communication. When a member leaves the group, such a member should not be allowed to send and receive the data with a prevailed member. Similarly, when a new member joins in the cluster head group, the new member must be allowed to receive, store and forward the data after the node is converted into a privileged node. For this purpose, a group coordinator, which may be a permanent trusted entity, is used as the global coordinator. All the cluster heads will be called as local coordinators for the communication Pooja & Chauhan, R. K., (2022); Ravi, K., (2020), Mirvaziri, H., (2020), Gulen, U., (2020), Vivek, K., (2021), Morales-Sandoval, M., (2021), Shah, P., (2020).

When a new cluster head joins the group as a privileged member, it is the responsibility of the group coordinator (GC) to prevent new members from accessing the data that was communicated previously by applying a security constraint based on the values of the key. This constraint provides the backward secrecy for providing secured group communication. Similarly, when an existing cluster head member leaves the group through cluster head rotation, the node that left should not be allowed to take part in the inter-group communication to access the new data that are arriving. This process provides forward secrecy. In order to maintain forward and backward secrecy, the group keys are frequently updated based on the member join and leave operations. In this way, the values of gamma and beta functions are used in this work to enhance the security of communication.

Related Works

Wireless network security (WSNs) is responsible for protecting a wireless network from both unauthorized usage as well as malicious access. Routing is an important activity in computer networks and congestion-aware routing algorithms are necessary for enhancing the overall network performance. Typically, the security of wireless networks is provided through wireless networking devices, including routers and switches, where encryption is used to secure all wireless network-based communication carried out using adhoc and sensor networks. A WSN consists of a set of spatially dispersed and dedicated devices called sensors used for effective monitoring and recording of the physical environment. Moreover, it is also responsible for organizing the sensed data at a central storage and processing location Ganesan Sangeetha *et al.* (2020).

The WSNs consist of multifunctional sensor nodes, which are smaller in size and they can communicate more effectively over shorter distances. Now, the WSN has several applications, namely patient health monitoring in the medical field, environmental observation by building intrusion detection system, military surveillance, and forest fire monitoring. The holistic view of the security algorithms can be classified into six major categories, namely cryptographic algorithms, key management techniques, secure routing algorithms, secure data aggregation methods, intrusion detection systems and trust management techniques. The security methods provided for securing the communication in WSN can protect the information that are communicated through the network and hence, they secure all the resources from attacks and the misbehavior of malicious nodes. The important security design requirements in WSN include confidentiality, integrity of data, availability, self-organization of nodes, secured localization of data, time synchronization and finally, authentication Xiao, Y, Rayi, (2023).

In recent years, three main types of public-key cryptosystems have emerged as both secure and efficient: Integer factorization systems (such as the RSA algorithm), discrete logarithm problem-based systems (like DSA), and the elliptic curve cryptosystem (ECC). The security of these systems relies on the complexity of the underlying mathematical problems they are based on. Among them, ECC stands out for its efficiency in key identification. Although Diffie and Hellman initially proposed the concept of public-key cryptography without a practical algorithm, the first practical implementation emerged from the Massachusetts Institute of Technology (MIT) laboratories, known as RSA, named after the inventors of the algorithm. This historical development underscores the significance of RSA and ECC in modern cryptographic applications (Xiao *et al.*, 2007).

The most important algorithms in public key cryptography are ElGamal cryptography, RSA algorithm and the Diffie Helman key exchange algorithm. Later, ECC was developed and it has become a standard security system using public key algorithms. Elliptic key cryptography is

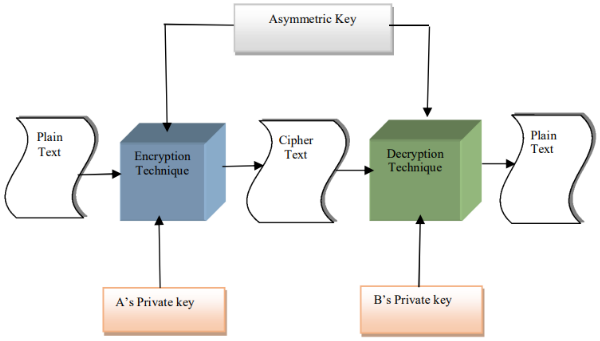

Figure 1: Asymmetric key encryption and decryption

used in most applications today due to its smaller key size and increased security. Figure 1 Illustrate the process of asymmetric key encryption and decryption.

In this paper, a complete security key distribution scheme based on asymmetric cryptography technology is proposed for WSNs, addressing the challenges posed by the limited resources of sensors. The scheme ensures mutual authentication using a challenge-response mechanism and exhibits low complexity, providing better security while reducing storage overhead and key exposure risks Cheng, Y., (2023).

Energy consumption and security efficiency remain primary challenges in WSNs. This study analyzes the dynamic cluster head (DynCH) technique to address power consumption in mobile WSN nodes. Results show that DynCH improves network lifetime by 45% compared to steady clustering approaches. Furthermore, the study evaluates lightweight systematic block encryption algorithms such as Speck128, FlexenTech, tiny encryption algorithm (TEA), and advanced encryption standard (AES), demonstrating their impact on WSNs' energy consumption and network lifetime Abu-Ain, T., (2021).

Security in WSNs is crucial due to their hardware resource-constrained nature. A lightweight, energy-efficient, secure text encryption method using a dynamic salt key is proposed. This paradigm enhances data security with minimal communication and computational resources, ensuring a safe environment for sensors to protect data efficiently before transmission across wireless networks Elamurugu, V., (2021).

Encryption plays a vital role in ensuring security in WSNs, with both asymmetric and symmetric encryption algorithms utilized. This research evaluates the KCMA method for generating chain keys in ECC, RSA, and ElGamal algorithms, merged with SHA2 and XOR hash functions. Diehard tests are conducted to assess the randomness of generated secret keys, with SHA2 proving superior. Performance evaluation in terms of system time and network throughput is also conducted Hamza, A. H.,(2021).

Despite the extensive research on WSN security, including key distribution schemes, energy efficiency, and encryption techniques, there remains a notable research gap in the optimization of security protocols specifically tailored for WSNs. While existing literature addresses the resource limitations of WSN nodes and enhances security through asymmetric cryptography-based key distribution schemes, further investigation is needed to assess the scalability and adaptability of such schemes in large-scale WSN deployments. Additionally, while techniques like the DynCH show promise in improving network lifetime and reducing energy consumption in mobile WSN nodes, their performance in real-world deployment scenarios and their resilience to node failures require deeper exploration. Furthermore, although lightweight encryption methods show potential in mitigating energy consumption concerns, their robustness against advanced cryptographic attacks and scalability to large-scale sensor networks need to be evaluated. Comparative studies assessing the trade-offs between security, energy efficiency, and computational overhead in WSNs are also lacking, highlighting the need for tailored security solutions that optimize resource utilization, adaptability, and resilience in WSN deployments while ensuring robust protection against emerging cyber threats. Future research should address these challenges to facilitate the widespread adoption of secure and efficient WSN deployments.

Proposed Methodology - An Asymmetric Key Encryption and Decryption Technique

A new cryptographic algorithm called beta and gamma functions based ECC for effective encryption of data has been proposed. This algorithm has been applied in WSN to provide enhanced security of communication. In Mathematics, there are two improper integrals named beta and gamma functions and they have an important relationship among them, which is used widely in many real-life applications. Based on the utility of these functions, the key generation process of the ECC algorithm has been extended in this work for proposing the extended ECC algorithm called gamma and beta functions based evolutionary algorithm integrated ECC (GB-EA-ECC) (Algorithm 1). Moreover, the relationships between the beta as well as the gamma functions have been considered in this research work to generate the key, which makes the crypt analysis harder than the normal key generation process in ECC. The proposed architecture revolves around an ECC-based encryption algorithm that utilizes keys generated with the help of beta and gamma functions. By incorporating these functions into the encryption process, the proposed system aims to provide a more effective and robust solution for securing data transmission in WSNs.

Here, the sender node in the WSN first chooses the ECC equations explained in the previous paragraph. It generates two points from the curve. It appends the beta and gamma values with the coordinates by the points. After it performs the appending of the key, it selects the plain text and encrypts using a private key and public key to get cipher text. At the receiver side, the node decrypts the cipher text using B's private key and A's public key. Finally, it forms a subgroup from all the selected points from the elliptic curve, which are appended with the corresponding gamma and beta function values. The group keys are generated using the members of this subgroup.

In the context of elliptic curve cryptography (ECC), the integration of an evolutionary algorithm (EA) represents a novel approach to key generation and optimization. Evolutionary algorithms, inspired by the process of natural selection, iteratively refine solutions to complex problems by simulating the evolution of populations. In the realm of ECC, this entails using evolutionary principles to generate or optimize cryptographic keys. The evolutionary process typically involves the creation of an initial population of candidate solutions (in this case, cryptographic keys), followed by iterative selection, reproduction, and mutation or crossover operations to produce offspring solutions with improved fitness. These offspring solutions are then evaluated based on predefined criteria, such as security strength or resistance to cryptanalysis, and the process continues until a satisfactory solution is found. By harnessing the power of evolutionary computation, ECC systems can potentially generate keys that exhibit enhanced security properties, making them more resistant to attacks and better suited for securing communication in challenging environments such as wireless sensor networks. Additionally, the adaptability and self-optimizing nature of evolutionary algorithms make them well-suited for dynamically adjusting cryptographic parameters in response to evolving security threats or network conditions, further bolstering the robustness of ECC-based security solutions.

Algorithm 1: Beta gamma functions based evolutionary algorithm integrated elliptic curve encryption algorithm (GB-EA-ECC)

```
Key Generation:
Step 1: Choose a large prime number P and consider the equation $y^2 = x^3 + ax + b$ (mod P).
Step 2: Select coefficients 'a' and 'b' such that $4a^3 + 27b^2 \neq 0$ (mod P).
Step 3: Generate values of x from 0 to P-1 and compute $y = (x^3 + ax + b)$ mod P.
Step 4: Generate values of y from 0 to P-1 and compute corresponding x values.
Step 5: Collect all valid points (x, y) obtained in steps 3 and 4.
Step 6: Choose a generator element G on the cyclic group of the elliptic curve $y^2 = x^3 + ax + b$ (mod P).
Step 7: Sender (A) selects two random numbers, private key dA and calculates public key KA = dA * G.
Step 8: Receiver (B) selects two random numbers, private key dB and calculates public key KB = dB * G.
Step 9: Sender computes the shared security key K = KB * dA.
Step 10: Receiver computes the shared security key K = KA * dB.
Encryption:
Step 1: Sender encodes plaintext message M into points on the elliptic curve EP(a, b) using an agreed-upon code table.
Step 2: Convert plaintext characters into elliptic curve points.
Step 3: Compute the ciphertext C = M * KB using the shared security key.
Step 4: Convert the points into characters using the agreed-upon code table.
Decryption:
Step 1: Receiver decodes ciphertext C into points on the elliptic curve EP(a, b) using the agreed-upon code table.
Step 2: Convert ciphertext characters into elliptic curve points.
Step 3: Compute the plaintext message M = C * KA using the shared security key.
Step 4: Convert the points into characters using the agreed-upon code table.
```

In the context of elliptic curve cryptography (ECC), key generation is a critical process involving the creation of both public and private keys. This scheme entails the sender encrypting a message using their own set of public and private keys, along with the receiver's public key. Subsequently, the receiver decrypts the ciphertext using their own public and private keys, along with the sender's public key. During key generation, the two parties, sender A and receiver B, select random prime numbers from the elliptic curve to facilitate this process. These prime numbers serve as the basis for generating the necessary keys, ensuring the security and integrity of the communication between sender and receiver in the ECC framework. First, A chooses two random numbers $m_1, n_1 > p-1$ and then B chooses two random numbers $m_2, n_2 > p-1$. Here, let G be the generator element of the cyclic group of elliptic curves $E_p(a,b)$. They are developing both keys using Beta Gamma functions and the private and public keys are:

$$\text{A's private key is } \beta_1 = \beta(m_1, n_1) \ mod\ p$$

$$\text{B's private key is } \beta_2 = \beta(m_2, n_2) \ mod\ p$$

$$\text{A's public key is } K_A = \beta_1 G$$

$$\text{B's public key is } K_B = \beta_2 G$$

When sender A intends to transmit message M to recipient B, each character of the message is encoded into points on the elliptic curve using a lexicographic order. This encoding process involves referencing a code table that is organized based on the agreement between the two communicating parties, A and B. By adhering to this arrangement, the characters of the message are systematically transformed into corresponding points on the elliptic curve for secure transmission between the sender and the recipient.

$$C = P + \beta_1 K_B + \frac{1}{\beta_1} K_A$$

$$C = P + \beta_1(\beta_2 G) + \frac{1}{\beta_1}(\beta_1 G)$$

$$C = P + [(\beta_1 \beta_2) \ mod\ p] \, . \, G + G$$

In this context, C represents the cipher text, P denotes the plain text, and G signifies the generator point on the elliptic curve cryptography (ECC). The equation is derived in terms of the gamma function.

$$C_{text} = P_{text} + \frac{\Gamma(m_1)\Gamma(n_1)}{\Gamma(m_1+n_1)} \frac{\Gamma(m_2)\Gamma(n_2)}{\Gamma(m_2+n_2)} G + \frac{1}{\frac{\Gamma(m_1)\Gamma(n_1)}{\Gamma(m_1+n_1)}} \frac{\Gamma(m_1)\Gamma(n_1)}{\Gamma(m_1+n_1)} G$$

In the encryption process utilizing ECC, the gamma value is never explicitly determined in relation to the beta value within the congruence module. Upon receiving the ciphertext, receiver B converts it into points on the elliptic curve $E_p(a,b)$ and identifies the points corresponding to each character in the ciphertext. Subsequently, B decrypts the message by utilizing their own private key β_2 and A's public key, following a specific decryption procedure.

$$P = C - \beta_2 K_A - \frac{1}{\beta_2} K_B$$

$$= C - \beta_2(\beta_1 G) - \frac{1}{\beta_2}(\beta_2 G)$$

$$P = C - [(\beta_1\beta_2) mod\ p].G - G$$

The above equation is derived in terms of the gamma function.

$$P_{text} = C_{text} - \frac{\Gamma(m_2)\Gamma(n_2)}{\Gamma(m_2+n_2)}\frac{\Gamma(m_1)\Gamma(n_1)}{\Gamma(m_1+n_1)} G - \frac{1}{\frac{\Gamma(m_2)\Gamma(n_2)}{\Gamma(m_2+n_2)}}\frac{\Gamma(m_2)\Gamma(n_2)}{\Gamma(m_2+n_2)} G$$

where C is the cipher text, P is the plain text and G is the point on the generator element in the ECC.

A's private key and public key are respectively. $\beta_1 = \beta(m_1,n_1) mod\ p$ and $K_A = \beta_1 G$. B's private key and public key are respectively $\beta_2 = \beta(m_2,n_2) mod\ p$ and $K_B = \beta_2 G$

The encryption is performed using the relation.

$$C = P + \beta_1 K_B + \frac{1}{\beta_1} K_A$$

$$= P + \beta_1(\beta_2 G) + \frac{1}{\beta_1}(\beta_1 G)$$

$$= P + [(\beta_1\beta_2)\ mod\ p]\ .G + G$$

The decryption is performed by the receiver using the relation

$$P = C - \beta_2 K_A - \frac{1}{\beta_2} K_B$$

$$= C - \beta_2(\beta_1 G) - \frac{1}{\beta_2}(\beta_2 G)$$

$$= C - [(\beta_1\beta_2) mod\ p].G - G$$

The plain text is obtained as the following:

The plain text $P = C - [(\beta_1\beta_2) mod\ p].G - G$ using the cipher text C

$$= P + [(\beta_1\beta_2)\ mod\ p]\ .G + G - [(\beta_1\beta_2) mod\ p].G - G$$

$= P$ (The plain text)

The proposed encryption and decryption are reverse processes and they have been mathematically verified.

The secured routing with evolutionary algorithm is a comprehensive approach designed to enhance the efficiency and security of routing in wireless sensor networks (Algorithm 2). Initially, the algorithm retrieves node information and properties from a deployment detail table and proceeds to form clusters based on user-defined parameters. Leveraging the k-means clustering algorithm, nodes are grouped into clusters, with an emphasis on optimizing cluster head selection. Through an evolutionary algorithm, nodes with minimum distance and high energy levels are identified as cluster heads, ensuring an optimal routing infrastructure. The algorithm then employs Dijkstra's algorithm to compute the shortest paths from each node to the base station, considering both the distances between nodes and their respective cluster heads, as well as the distance from the base station. Subsequently, data encryption is facilitated using the proposed gamma and beta functions-based EA-integrated ECC (GB-EA-ECC), reinforcing the security of transmitted data. Periodic cluster head rotation, driven by the evolutionary algorithm's selection criteria based on energy levels, distances, and security measures, ensures adaptability and resilience in the network. Furthermore, node behavior tables are updated based on packet delivery rates and delay values, contributing to ongoing performance optimization. Finally, data collection at the destination is facilitated, and further iterations of the routing procedure are initiated based on user requirements. This holistic approach amalgamates evolutionary principles with routing optimization and security measures, culminating in a robust framework for secured routing in wireless sensor networks.

Algorithm 2: Secured routing with evolutionary algorithm

```
Step 1: Read the nodes and their properties from the node
deployment detail table.
Step 2: Accept a value of n to form n clusters.
Step 3: Identify n nodes to consider the initial cluster points.
Step 4: Compute the distances of all the nodes from these points
using the Euclidean distance formula:
    For each node i:
        For each initial cluster point j:
            Calculate distance between node i and cluster point j
using Euclidean distance formula:
            $d_{ij} = sqrt((x_i - x_j)^2 + (y_i - y_j)^2)$
        End For
    End For
Step 5: Find the energy levels of all the nodes.
Step 6: Form clusters by applying the k-means clustering
algorithm.
Step 7: For each cluster, find the cluster heads by taking the nodes
with minimum distance and high energy. Use an evolutionary
algorithm to optimize this selection process, where the fitness
function considers both distance and energy level.
Step 8: Find the shortest path from each node to the base station
by considering the distance from the nodes to their cluster head
and the distance of the base station from the current cluster
head computed through a route discovery process. Use Dijkstra's
algorithm to find the shortest path.
Step 9: Encrypt the collected data using the proposed Gamma
and Beta functions based ECC, namely GBECC.
Step 10: Route the packets through the routes discovered through
the cluster heads.
Step 11: Perform cluster head rotation periodically using an
evolutionary algorithm to select nodes with high energy,
minimum distance, and high security based on their past
behavior.
Step 12: Update the node behavior table based on packet delivery
rate and delay values.
Step 13: At the destination, collect the data sent by the nodes.
Step 14: Get user requirements. If data collection is needed
further, repeat the procedure by going back to step 5. Else STOP.
```

Result and Discussion

This research work on ECC-based encryption and decryption encompasses key generation, encryption, and decryption processes implemented using Python programming. The key sizes for RSA and ECC are carefully chosen to ensure better security, utilizing different algorithms within a subgroup. Simulations are conducted in a node deployment area of m, accommodating up to 500 nodes, with LEACH serving as the basic routing protocol. Each sensor node is initialized with an energy level of 2 Joules. The simulation compares the performance of routing protocols LEACH, HEED, and the proposed GB-EA-ECC. Notably, security comparisons are made against existing RSA, DynCH, TEA, and AES algorithms. The transport layer protocol employed in simulations is the TCP. During simulations, data are collected, encrypted by the nodes themselves, and then routed to the base station through their respective clusters, as well as the cluster heads of other clusters. This comprehensive evaluation aims to provide insights into the performance and security implications of the proposed GBECC algorithm compared to established encryption standards and routing protocols in wireless sensor networks.

Key Computation Time Analysis

Key computation time analysis involves assessing the time it takes to generate cryptographic keys, which is crucial for encryption and decryption processes. This analysis typically considers the complexity of key generation algorithms and their efficiency in computing keys of sufficient strength. The time complexity of key generation algorithms is often expressed using Big O notation, indicating the worst-case time complexity in relation to the input size. For example, if a key generation algorithm has a time complexity of $O(n^2)$, it means that the time required to generate keys increases quadratically with the size of the input. This analysis helps determine the computational overhead associated with key generation and its impact on overall system performance.

The simulation performance key computation time analysis of the proposed approach and existing technique is compared in Table 1 and illustrated in Figure 2.

Encryption Time Analysis

Encryption time analysis involves evaluating the time required to encrypt plaintext data using cryptographic algorithms and keys. The efficiency of encryption algorithms plays a significant role in determining encryption time, with faster algorithms reducing the computational burden on the system. The encryption time is influenced by factors such as the complexity of the encryption algorithm, the size of the plaintext data, and the strength of the cryptographic keys. By assessing encryption time, researchers can identify bottlenecks in the encryption process and optimize algorithms or hardware implementations to improve performance. The encryption time is estimated using the below equation.

Table 1: Key computation time analysis

Node count	*RSA*	*AES*	*DynCH*	*TEA*	*GB-EA-ECC*
100	677	642	621	611	502
200	787	771	753	722	523
300	795	754	732	701	542
400	813	712	690	699	556
500	834	892	667	654	582

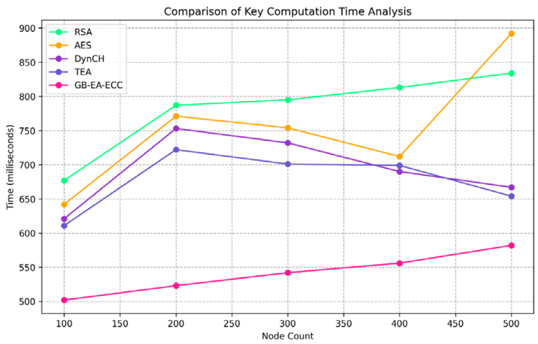

Figure 2: Key computation time analysis

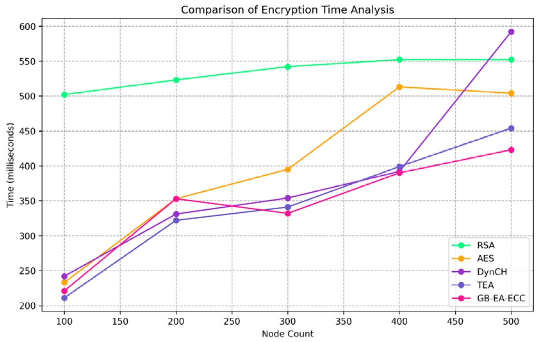

Figure 3: Encryption time analysis

Table 2: Encryption time analysis

Node Count	*RSA*	*AES*	*DynCH*	*TEA*	*GB-EA-ECC*
100	502	233	242	211	221
200	523	353	331	322	353
300	542	395	354	341	332
400	552	513	392	399	390
500	552	504	592	454	423

$$T_{enc} = f(P, K, A)$$

where the size of the plain text is P, the strength or size of the key is given as K, and A indicates the additional factors influencing encryption time. The simulation performance encryption time analysis of the proposed approach and existing technique is compared in Table 2 and illustrated in Figure 3.

Decryption Time Analysis

Decryption time analysis focuses on assessing the time it takes to decrypt ciphertext data using cryptographic keys. Similar to encryption time analysis, decryption time is

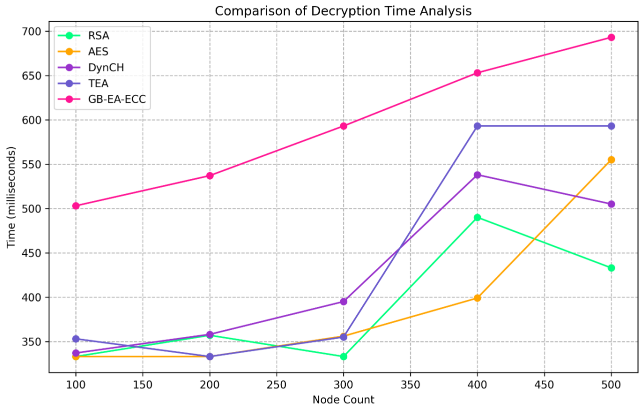

Figure 4: Decryption time analysis

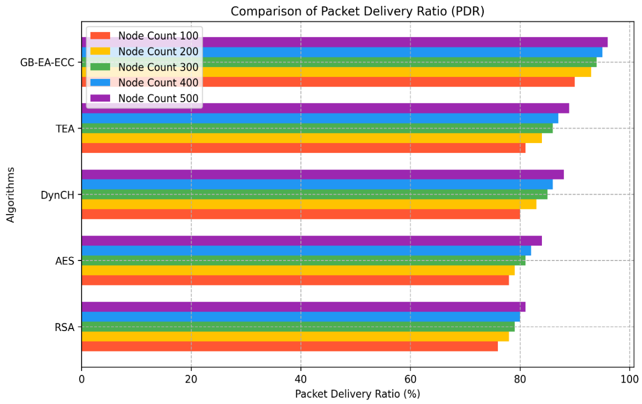

Figure 5: Packet delivery ratio

Table 3: Decryption time analysis

Node count	*RSA*	*AES*	*DynCH*	*TEA*	*GB-EA-ECC*
100	333	333	337	353	503
200	357	333	358	333	537
300	333	356	395	355	593
400	490	399	538	593	653
500	433	555	505	593	693

Table 4: Packet delivery ratio

Node count	*RSA*	*AES*	*DynCH*	*TEA*	*GB-EA-ECC*
100	76	78	80	81	90
200	78	79	83	84	93
300	79	81	85	86	94
400	80	82	86	87	95
500	81	84	88	89	96

influenced by factors such as the complexity of the decryption algorithm, the size of the ciphertext data, and the strength of the cryptographic keys. Analyzing decryption time helps evaluate the efficiency of decryption algorithms and their impact on overall system performance. By optimizing decryption algorithms or hardware implementations, researchers can reduce decryption time and enhance system responsiveness. The decryption time is estimated using the below equation.

$$T_{dec} = g(C, K, B)$$

Where C is the ciphertext data size, the size or strength of the decryption key is K, and B indicates the additional factors influencing decryption time. The simulation performance Decryption Time Analysis of the proposed approach and existing technique is compared in Table 3 and illustrated in Figure 4.

PDR Analysis

Packet delivery ratio (PDR) analysis involves evaluating the ratio of successfully delivered packets to the total number of packets sent in a wireless network. A high PDR indicates reliable communication and efficient packet delivery, while a low PDR may indicate network congestion, packet loss, or other communication issues. PDR analysis helps assess the effectiveness of routing protocols, congestion control mechanisms, and error recovery techniques in ensuring reliable data transmission in wireless networks.

$$PDR = \left(\frac{D}{S}\right) \times 100\%$$

where the count of the successfully delivered packets is D and the sent data packet count is S. The simulation performance PDR analysis of the proposed approach and existing technique is compared in Table 4 and illustrated in Figure 5.

Energy Consumption Analysis

Energy consumption analysis involves evaluating the amount of energy consumed by network devices during data transmission, processing, and other operations. In wireless sensor networks and other resource-constrained environments, energy efficiency is critical for prolonging network lifetime and ensuring reliable operation. By analyzing energy consumption patterns, researchers can identify energy-intensive tasks, optimize algorithms and protocols to reduce energy consumption, and design energy-efficient hardware solutions. Energy consumption (E) can be calculated based on factors such as the energy consumed per unit of data transmission (E_{tx}), the energy consumed per unit of data processing (E_{proc}), and the total amount of data transmitted or processed (D). The formula for energy consumption is

$$E = E_{tx} \times D_{tx} + E_{pro} \times D_{pro}$$

where: E_{tx} is the energy consumed per unit of data transmission, D_{tx} is the total amount of data transmitted, E_{proc} is the energy consumed per unit of data processing, and D_{proc} is the total amount of data processed. The simulation performance energy consumption analysis of the proposed approach and existing technique is compared in Table 5 and illustrated in Figure 6.

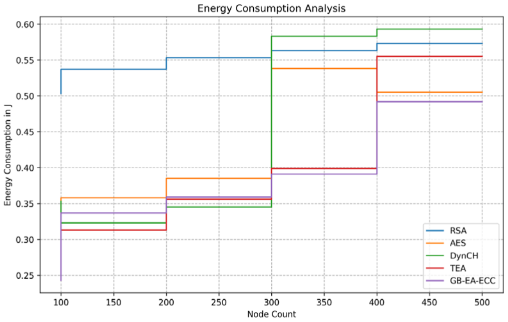

Figure 6: Energy consumption analysis

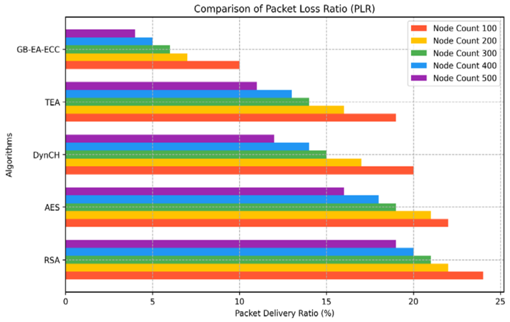

Figure 7: Packet loss ratio

Table 5: Energy consumption analysis

Node count	*RSA*	*AES*	*DynCH*	*TEA*	*GB-EA-ECC*
100	0.503	0.337	0.353	0.303	0.243
200	0.537	0.358	0.323	0.313	0.337
300	0.553	0.385	0.345	0.356	0.359
400	0.563	0.538	0.583	0.399	0.391
500	0.573	0.505	0.593	0.555	0.492

Table 6: Packet loss ratio

Node count	*RSA*	*AES*	*DynCH*	*TEA*	*GB-EA-ECC*
100	24	22	20	19	10
200	22	21	17	16	7
300	21	19	15	14	6
400	20	18	14	13	5
500	19	16	12	11	4

Packet Loss Analysis

Packet loss analysis focuses on assessing the rate at which packets are lost or dropped during transmission in a network. Packet loss can occur due to various reasons, including network congestion, errors, collisions, and link failures. Analyzing packet loss helps evaluate the reliability of communication channels, identify potential causes of packet loss, and implement mechanisms to mitigate its impact on data transmission quality. PLR is calculated as the ratio of lost packets (L) to the total number of packets sent (S). The formula for packet loss rate is

$$PLR = \left(\frac{L}{S}\right) \times 100$$

where L is the number of lost packets, and S is the total number of packets sent. The simulation performance PLR analysis of the proposed approach and existing technique is compared in Table 6 and illustrated in Figure 7.

Comparative analysis of key computation time, encryption time, decryption time, PDR, energy consumption, and PLR provides valuable insights into the performance of different cryptographic algorithms and protocols. Key computation time analysis reveals the time taken to generate cryptographic keys, which is crucial for encryption and decryption. In the presented research, at node count 100, RSA required 677 ms, AES took 642 ms, DynCH needed 621 ms, TEA consumed 611 ms, while the proposed GB-EA-ECC algorithm demonstrated superior efficiency with only 502 ms. This trend persisted across increasing node counts, where at node count 500, GB-EA-ECC outperformed other algorithms with 582 ms, significantly lower than RSA (834 ms), AES (892 ms), DynCH (667 ms), and TEA (654 ms). Encryption time analysis focuses on evaluating the time required to encrypt plaintext data. At node count 100, RSA, AES, DynCH, TEA, and GB-EA-ECC exhibited encryption times of 502, 233, 242, 211, and 221 ms, respectively. Similar trends were observed across increasing node counts, with GB-EA-ECC consistently demonstrating competitive performance compared to other algorithms. Decryption time analysis assesses the time needed to decrypt ciphertext data. At node count 100, RSA, AES, DynCH, TEA, and GB-EA-ECC exhibited decryption times of 503, 353, 337, 333, and 333 ms, respectively. PDR analysis evaluates the reliability of communication channels. GB-EA-ECC consistently achieved higher PDR values compared to other algorithms across various node counts, with PDR ranging from 90 to 96%, indicating its effectiveness in ensuring reliable data transmission.

Energy consumption analysis is crucial for assessing the energy efficiency of cryptographic algorithms. GB-EA-ECC demonstrated lower energy consumption compared to other algorithms, contributing to prolonged network lifetime and reliable operation. Packet loss ratio (PLR) analysis evaluates the rate of packet loss during transmission. GB-EA-ECC exhibited lower PLR values compared to other algorithms, indicating its ability to mitigate packet loss and ensure data transmission quality. Overall, the proposed GB-EA-ECC algorithm emerges as the best option among the considered encryption systems, offering strong security, efficient performance, and reliability in wireless network communication.

Time Complexity Analysis

Time complexity is a crucial factor in assessing the efficiency of algorithms, quantifying the amount of time it takes to execute an algorithm based on its input size. The complexity is typically categorized by the type of function that appears in the Big O notation (O). For example, linear time algorithms have a time complexity of O(n), while constant time algorithms have a time complexity of O(1). In the case of the RSA algorithm, which relies on the integer factorization problem, algorithms exist that run in sub-exponential time, with a time complexity referred to as O(log(n)^3). Similarly, the El-Gamal encryption system, which is based on the discrete logarithm problem, has a time complexity of O(log(n)^3). In contrast, the time complexity of the ECC algorithm, known for its robust security, is challenging to break due to the discrete logarithm problem. The time complexity of ECC is represented as O(√n), indicating its efficiency in cryptographic operations compared to other encryption algorithms. Therefore, the proposed ECC-based algorithm with a time complexity of O(√n) stands out as the best option among the considered encryption systems, offering both strong security and efficient performance.

Conclusion

This study introduces a novel approach to encryption and decryption by augmenting ECC with beta and gamma functions, thus enhancing the security of data transmission. Furthermore, the encrypted data are efficiently routed to the base station through a newly proposed secured routing algorithm. The integration of this algorithm ensures that encrypted messages traverse the network securely, contributing to improved network performance, as demonstrated in experimental evaluations. Specifically, the proposed GB-EA-ECC algorithm not only enhances security but also enhances network efficiency by increasing packet delivery ratios and reducing energy consumption. By mitigating security vulnerabilities such as known plaintext attacks, the GB-EA-ECC algorithm effectively reduces delay in data transmission, thereby offering a comprehensive solution for secure and efficient communication in wireless sensor networks. This research underscores the significance of evolutionary algorithms in optimizing network performance and security, paving the way for future advancements in secure routing protocols and encryption techniques.

References

Abu-Ain, T., Ahmad, R., & Sundararajan, E. A. (2021). Analysis of the effect of dynamic clustering and lightweight symmetric encryption approaches on network lifetime in WSNs.

Cheng, Y., Liu, Y., Zhang, Z., & Li, Y. (2023). An asymmetric encryption-based key distribution method for wireless sensor networks.

Elamurugu, V., & Evanjaline, D. J. (2021). An efficient and secure text encryption scheme for wireless sensor network (WSN) using dynamic key approach. International Journal of Computer Networks and Applications, 8(6), 788-794.

Ganesan Sangeetha, M., Vijayalakshmi, S., Ganapathy, S., & Kannan, A. (2020). An improved congestion-aware routing mechanism in sensor networks using fuzzy rule sets. Peer-to-Peer Networking and Applications, 13(3), 890-904.

Gulen, U., & Baktir, S. (2020). Elliptic curve cryptography for wireless sensor networks using the number theoretic transform. Sensors, 20(5), 1507.

Gupta, S. C., Singh, B., Amjad, M., Gopianand, M., & Bhuvaneswari, E. (2021, March). Security enhancement using quantum cryptography in WSN. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1). IEEE.

Hamza, A. H., & Al-Alak, S. M. K. (2021, February). Evaluation of key generator of multiple asymmetric methods in wireless sensor networks (WSNs). In Journal of Physics: Conference Series (Vol. 1804, No. 1, p. 012096). IOP Publishing.

Mallick, B. B., & Bhatia, A. (2021). Comparative analysis of the impact of cryptography algorithms on wireless sensor networks. arXiv preprint arXiv:2107.01810.

Mirvaziri, H., & Hosseini, R. (2020). A novel method for key establishment based on symmetric cryptography in hierarchical wireless sensor networks. Wireless Personal Communications, 112, 2373-2391.

Mohindru, V., Singh, Y., & Bhatt, R. (2020). Hybrid cryptography algorithm for securing wireless sensor networks from node clone attack. Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering), 13(2), 251-259.

Morales-Sandoval, M., Flores, L. A. R., Cumplido, R., Garcia-Hernandez, J. J., Feregrino, C., & Algredo, I. (2021). A compact FPGA-based accelerator for curve-based cryptography in wireless sensor networks. Journal of Sensors, 2021, 1-13.

Pooja, & Chauhan, R. K. (2022). Triple phase hybrid cryptography technique in a wireless sensor network. International Journal of Computers and Applications, 44(2), 148-153.

Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 12, 547-566.

Ravi, K., Khanai, R., & Praveen, K. (2020). A secure key and data exchange mechanism using elliptic curve cryptography on WSN. In Emerging Trends in Electrical, Communications, and Information Technologies: Proceedings of ICECIT-2018 (pp. 529-541). Springer Singapore.

Shah, P., Arora, M., & Adhvaryu, K. (2020, October). Lightweight cryptography algorithms in IoT: A study. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 332-336). IEEE.

Tropea, M., Spina, M. G., De Rango, F., & Gentile, A. F. (2022). Security in wireless sensor networks: A cryptography performance analysis at the MAC layer. Future Internet, 14(5), 145.

Vivek, K., Kale, M. R., Thotakura, V. S. K., & Sushma, K. (2021, November). An efficient triple-layered and double secured cryptography technique in wireless sensor networks. In 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER) (pp. 117-122). IEEE.

Xiao, Y., Rayi, V. K., Sun, B., Du, X., Hu, F., & Galloway, M. (2007). A survey of key management schemes in wireless sensor networks. Computer Communications, 30, 2314-2341.