*Computer Assisted Methods in Engineering and Science*
**31**(2): 213–240, 2024, doi: 10.24423/cames.2024.594

This article belongs to the *Special Issue on Scientific Computing and Learning Analytics for Smart Healthcare Systems* edited by Dr. C. Chakraborty, Dr. S. Barbosa and Dr. L. Garg

# Lightweight Hybrid Cryptography Algorithm for Wireless Body Area Sensor Networks Using Cipher Technique

Aizaz RAZIQ[1)], Kashif Naseer QURESHI[2)], Asfand YAR[1)], Kayhan Zrar GHAFOOR[3),4)], Gwanggil JEON[5)*]

[1)] *Department of Robotics and Artificial Intelligence, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST University), Islamabad, Pakistan*

[2)] *Department of Electronic & Computer Engineering, University of Limerick, Limerick, Ireland*

[3)] *Department of Information and Communication Technology Engineering, Erbil Polytechnic University, Erbil, Iraq*

[4)] *Department of Computer Science, Knowledge University, University Park, Erbil, Iraq*

[5)] *Department of Embedded Systems Engineering, Incheon National University, Incheon, South Korea*

[*] *Corresponding Author e-mail: gjeon@inu.ac.kr*

Wireless Body Area Networks (WBANs) are based on connected and dedicated sensor nodes for patient monitoring in the healthcare sector. The sensor nodes are implanted inside or outside the patient's body for sensing the vital signs and transmitting the sensed data to the end devices for decision-making. These sensor nodes use advanced communication technologies for data communication. However, they have limited capabilities in terms of computation power, battery life, storage, and memory, and these constraints make networks more vulnerable to security breaches and routing challenges. Important and sensitive information is exchanged over an unsecured channel in the network. Several devices are involved in handling the data in WBANs, including sink nodes, coordinator, or gateway nodes. Many cryptographic schemes have been introduced to ensure security in WBANs by using traditional confidentiality and key-sharing strategies. However, these techniques are not suitable for limited resource-based sensor nodes. In this paper, we propose a Lightweight Hybrid Cryptography Algorithm (LWHCA) that uses cryptographic-based techniques for WBAN networks to improve network security, minimize network overhead and delay issues, and improve the healthcare monitoring processes. The proposed solution is evaluated in a simulation scenario and compared with state-of-the-art schemes in terms of energy consumption, and ciphertext size.

**Keywords:** WBAN, healthcare, security, data, network, routing, cryptography, lightweight, mechanism.

Copyright © 2024 The Author(s).
Published by IPPT PAN. This work is licensed under the Creative Commons Attribution License CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Wireless Body Area Network (WBAN) plays a vital role in monitoring and sensing patients' vital signs in the healthcare sector using smart technologies, heterogeneous and connected network standards [1]. The WBAN systems are feasible for all types of patients, especially in remote areas where patients are monitored with portable monitoring devices. WBAN applications are used to improve and enhance the daily life and disease management of patients in hospitals and homes. Sensor nodes are employed to sense and monitor the patient data and transmit it for further measurement and analysis in medical centers. These sensor nodes are connected with wireless technologies and form a network as per the network requirement. However, sensor nodes have limited resources such as small size, limited battery life, low computational processing capabilities, limited bandwidth, less storage and memory space. Sensor nodes are categorized based on their resources into Sensor Node (SN), Gateway Node (GN), and Cluster Head (CL). The sensor nodes have several parts, including a micro-controller, radio transceiver, and embedded battery. The cost of sensor nodes is lower compared to traditional wireless network devices and it depends upon the network size [2]. The normal SNs are more numerous and deployed for sensing and monitoring. SN is connected to the GN or Base Station (BS) [3]. The user interface provides access to users for accessing SNs and GNs. The nodes collect information from patients using wearable and implant sensor nodes, whereas the GN enables the communication between users and SNs [4]. The responsibilities of SNs are greater compared to ordinary nodes because they collect the data from regular nodes and send it to other devices [5, 6]. Some key factors, such as node connectivity, coverage area, and energy consumption, should be considered for node deployment. The autonomous design of WBAN makes it more feasible for users. The microprocessors automatically initiate contact between SNs and transfer the data for further processing. SNs are deployed to effectively perform the expected task. The performance of heterogeneous sensor networks is more realistic, scalable, load-balanced mechanism, and able to handle delay tolerance. Real-time WBAN applications are used for temperature monitoring, heath management, brain and cancer signs monitoring in hospitals. The sensed data must be accessible at any time and from everywhere [7]. Figure 1 shows the working process of the WBAN network, where different sensor nodes are implanted inside or outside the patient's body for sensing vital signs. All sensor nodes are connected with each other and linked to a GN for forwarding the sensing data towards the sink node.

Despite the various benefits, WBAN networks suffer from a variety of challenges, including the lack of physical infrastructure, localization issues, data communication problems, fault tolerance, self-management risks and security

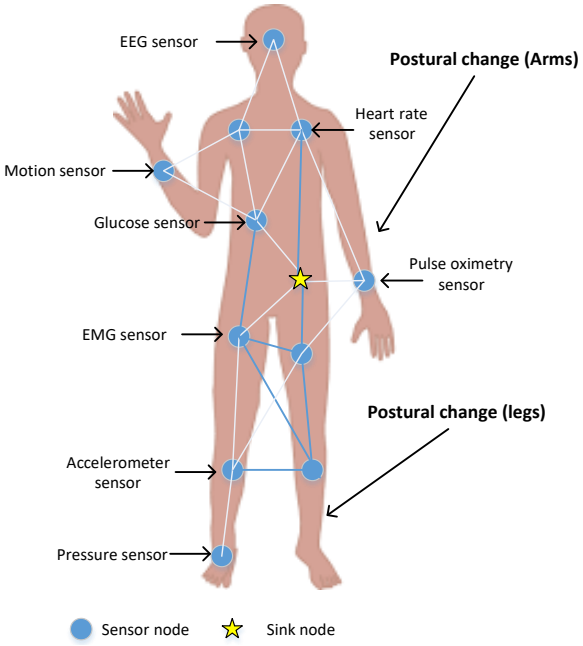


FIG. 1. Working process of WBAN network.

concerns. Due to these challenges, the performance of WBAN networks is degraded. Among these concerns, one of the significant issues is security, causing various threats and attacks. WBANs are established without proper security specifications and are more likely to be attacked during network deployment. Furthermore, due to a variety of complex issues such as compatibility, it is impossible to change the security specifications once the network is established. Several solutions have been proposed for security. For instance, asymmetric key algorithms use only a single key for encryption and decryption of data [3]. Data security and key sharing of nodes are also among the main issues in WBAN security, because SNs are mostly deployed randomly and in open areas where node communication becomes unsafe, especially in mobile conditions [7]. On the other hand, limited memory, computing processing power, the broadcasting of messages via wireless connection, and other factors make these networks more vulnerable [8]. Broadcasting messages through wireless network to SNs, exposes the signal to potential detection by anyone, causing a serious threat to network [9]. Since SNs have limited energy resources, sending secure data solutions must handle the energy issue to extend the network's lifetime [10].

In WBAN networks, node mobility creates a high risk of network security due to topology node mobility [7]. Attacks can be conducted internally or externally, depending on the applications involved, such as Denial of Service (DoS), eavesdropping, and sinkhole attacks. The requirements for these networks are

confidentiality, authentication, integrity, and availability. Encryption, authentication, latency management, authorization, and complexity handling are techniques used for security in WBANs. There are several different forms of security threats faced by WBANs. Different malicious attacks are launched to compromise the functionality of WBAN applications. As a result, it is necessary to provide WBANs with acceptable security standards. The computational complexity of ciphers is another concern and significant challenge. Security situation will become worse when SNs constantly exchange information, especially involving critical and sensitive information exchange [11]. The security architecture of WBAN heavily relies on cryptographic algorithms, which are not suitable for limited resources of SNS. In WBANs, there are two major issues associated with encryption algorithms. The first issue is an overhead in cryptographic algorithms and a decrease in the network's lifetime. Secondly, there is a concern regarding memory capacity, which corresponds to the size of an encrypted message, as well as the key size, which should be reduced. To meet security standards such as confidentiality, authentication, and integrity, various cryptographic algorithms have been proposed [8], but they often ignore various important factors.

Data transparency and traceability are significant factors for adopting any kind of security services aimed at mitigating risks. These factors play its role in encryption, where digital keys are used for data access. Another thing to remember when developing a lightweight stable protocol for WBAN are key management systems [12]. Key exchange is one of the most difficult issues in these networks. If a key is compromised, the security of the whole system is compromised. To address this challenge, researchers have suggested various protocols. However, in WBAN, not all key management system protocol can be used due to limited resources, which makes it impractical to implement. Since the majority of SNs have insufficient computing capacity, public-key cryptography is prohibitively costly in terms of device overhead [13, 14].

To utilize the available and fewer resources of the network, this paper designs a lightweight cryptographic security mechanism to increase the network lifetime, secure data communication, and reduce computation cost and overhead. The lightweight security protocol provides better security with less overhead and more data confidentiality. The paper's contributions are aligning with each other. The first objective is to review existing security solutions, especially designed for sensor networks. The existing solutions gap is the motivation for the second objective, where we design a lightweight crypto mechanism for SNs in WBAN networks. Thirdly, the proposed solution decreases the computational cost during data encryption and decryption processes in WBAN networks.

The rest of the paper is organized as follows: Sec. 2 elaborates on existing security solutions for sensor network, Sec. 3 discusses the design and development

of proposed solutions, Sec. 4 presents the results and discussion. The last section concludes the paper and outlines future directions.

## 2. Related work

Rizk and Alkady [8] proposed a new security algorithm to provide high security with reduced key maintenance using a combination of both symmetric and asymmetric cryptographic techniques by performing two parallel phases. By achieving a high degree of safety without increasing the execution time, these stages avoid the drawbacks of the current hybrid algorithms. It guarantees the confidentiality, authenticity, and integrity of three cryptographic primitives. To provide encryption, Elliptical Curve Cryptography (ECC) and Advanced Encryption Standard (AES) are combined. For authentication, the XOR-DUAL RSA algorithm and Message Digest-5 (MD5) for integrity are taken into consideration. The experiment results showed that the hybrid algorithm provides better performance in terms of ciphertext size, calculation time, and wireless sensor network (WSN) energy consumption. However, using both types of symmetric and asymmetric techniques creates extra overhead in the network especially for WBAN type networks.

Gope and Hwang [15] proposed a lightweight anonymous authentication protocol to secure the real-time data in WSNs. The user and the SN can create a session key, and a GN is used for mutual authentication. This scheme is based on hash and XOR functions. They introduce four phases: registration, anonymous authentication and key exchange, password renewal, and dynamic node addition. The authors claimed that this scheme provides the following security: user anonymity and traceability, resistance to node capture attack, and forward secrecy. However, this scheme has some major flaws related to security, as pointed out by Ghani *et al.* [16], including vulnerabilities against session key disclosure, password guessing, user traceability, de-synchronization attack, and stolen smart card. Since SN has limited resources and, after deployment, the battery power cannot be recharged, it is necessary to establish a protocol that uses fewer resources and provides better security in user authentication and key agreement protocols.

Elhoseny *et al.* [17] proposed secure data transmission by elliptic curve and homomorphic encryption for WSNs, building upon GASONeC algorithm. The best acceptable SN acting as the CH to convey messages to the BS was determined using a genetic algorithm. For key generation (both public and private keys) the ECC algorithm was employed, and utilized a 176-bit key size with a combination of some parameters including the distance between CH, node identification number, and ECC key. The CH used homomorphic encryption to collect encrypted data from its cluster members without needing to decrypt

them, subsequently sending the final message to the BS, and thereby helping to reduce the CH energy consumption. However, the cluster formation and head selection are time-consuming processes themselves.

Praveena and Smys [18] proposed an efficient cryptography approach for data security in WSNs. They combined two different encryption algorithms known as TTJSA and DJSA algorithms and presented a new Modern Encryption Standard (MES) version – 1 algorithm. They modified variable block size and variable key size, and following the completion of forwarding encryption, they divided the whole message into two parts, which were then swapped. Additionally, they improved Vernam cipher technique with feedback, and a new key was applied again. To increase the encryption procedure complexity, the entire operation was repeated several times.

Srinivas *et al.* [19] proposed a secure and efficient user authentication scheme for multi gateway WSN, providing key agreement and authentication using a bio-hashing technique. This scheme also allows for dynamic node addition and password changes. The advantage of bio-hashing is its zero error rate, imposter population, and clean separation of imposter population and genuine users. Essentially bio-hashing generates a vector of bits starting from the biometric features and a hash key known as the seed. It provides security against the following attacks: reply attacks, session key computation attacks, privileged insider attacks, stolen smart card attack, GN impersonation attacks, user impersonation attacks, SN spoofing attacks, etc. Unfortunately, it still has some vulnerabilities as revealed by Wang *et al.* [20], such as user anonymity violation, offline password guessing, and node capture attack.

Prakash and Rajput [21] proposed a new algorithm aimed at providing stable data communication while extending the duration of WSNs. They developed a hybrid WSN data encryption and decryption algorithm. Since key sharing was a major problem in the symmetric approach, they used the ECC algorithm, an asymmetric key algorithm, to perform the key generation, and next used this key to encrypt and decrypt data using the AES algorithm known as a symmetric key algorithm. They created an optimized algorithm that takes advantage of the benefits of two algorithms (EEC and AES). The proposed algorithm, which is simpler than ECC, is only used for key generation, not data encryption or decryption, and it is more reliable than AES because it avoids the issue of key sharing existing in AES.

Farooq *et al.* [22] discussed the necessity of encrypting data obtained by sensors from patients before transmission. The authors proposed a new concept called the Hybrid Encryption Algorithm (HEA), which can be used in both ad hoc and wired networks. For authentication, sensors are given unique IDs (NIDs) under registration numbers. As a result, each sensor on the patient's body will have its registration number (RN) and NID. For key exchange, the sink node

and BS share a key that will be used for data encryption and decryption on the sink node and base station sides, respectively. The elliptic curve Diffie–Hellman algorithm is used for key exchange between the sender and receiver. For the encryption/decryption process, a hybrid encryption algorithm (HEA) is used in the proposed system. Plaintext messages are divided into '$n$' smaller units or blocks for encryption. The first block is encoded with ECC 128-bit and the second block is encoded with AES 128-bit. This algorithm not only considers data protection but also addresses various limitations of sensor networks, such as battery power, bandwidth constraints, restricted processing capacity, and dynamic topology.

Many researchers focus on secure communication with session key establishment over an insecure channel to provide security against attacks. According to Alotaibi [23], the scheme proposed by Jung *et al.* [24] – an anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in WSN – has some security flaws and it cannot provide full security against user traceability attack, password guessing attack, and forgery attack. Additionally, Jung *et al.* [24] also did not consider the privacy of the session key in the scheme, and their proposed scheme fails to support node addition in a dynamic nature. To overcome this problem, Alotaibi [23] proposed an anonymous user authentication and session key establishment scheme that is based on an enhanced symmetric cryptosystem and biometrics. The protocol uses biometrics with a fuzzy extractor, providing efficient legitimate user login. This scheme provides security against password guessing attacks, privilege insider attack, user anonymity breaches, user and GN impersonation attack, node capture attack, man-in-the-middle attack, password guessing attack, etc. But still, there are some weakness in this scheme that as pointed out by Moghadam *et al.* [4] include vulnerability to stolen verification attack and perfect forward secrecy.

Naresh *et al.* [25] proposed a provable secure lightweight hyperelliptic curve-based communication (HECC) system for WSN. According to the authors, HECC is a lightweight protocol and more suitable for WSNs due to protocol less complexity and low-power system approach. The algorithm proposed by the authors is HEC with a discrete logarithm problem-based lightweight secure communication, known as HEC-DDH (HEC-Decision Diffie–Hellman). In this scheme, party A creates a private key, public key and signature, then sends the public key and signature to party B. Party B verifies the signature with the help of the public key. After successful verification, party B creates a public and private key pair and shares the public key with party A. Now, using a Diffie–Hellman they create a shared key. The authenticity is checked by signature verification. This algorithm provides confidentiality, integrity, non-repudiation, forward secrecy, authenticity, and enforceability. However, as the network size increases, the computation time also increases and optimized results are observed at lower

network sizes. HEEC with genus 2 using 80 bits provides the same security level as 160 bits of EEC.

Abdullah *et al.* [26] discussed the growing popularity of WSNs in recent years, and their important role in a variety of fields. However, deploying these technologies without necessary security measures has been a significant problem for many years. To address it, the authors proposed a hybrid cryptography algorithm for WSN. The algorithm used a combination of AES, RSA, and Lempel–Ziv–Welch (LZW) algorithms. The plaintext is divided into two parts with one part encrypted with AES and the other encrypted with RSA. LZW is used to compress the size of cipher message. Experimental results are performed in the network simulator NS-2 and demonstrate that compared to other hybrid cryptography methods, this approach reduces the rate of dropped packets and provides a good overall result.

Gupta and Kapoor [27] discussed that data encryption not only protects sensitive data but also adds to the computation and memory overhead of a web application when processing large volumes of data. To maintain fast computations, web applications require low computing overhead. The AES algorithm is replaced by the blowfish encryption algorithm, which is faster than AES, and the ECC algorithm is replaced by the RSA algorithm in the proposed hybrid model. The ECC algorithm can provide a large modulus and a corresponding large key with the same level of protection as RSA. The MD5 algorithm is used to verify the data's originality, and the modified Kerberos protocol is used to authenticate the client. The odd and even chunks are divided. After that, even chunks are encrypted using the blowfish algorithm, and odd chunks are encrypted using the ECC algorithm. Then, the cipher chunks are generated.

Yuvaraju and Pranesh [28] discussed an energy-proficient hybrid secure scheme (EPHSS) proposed to provide WSN with safe and energy-efficient data transmission. The node clustering process is employed to properly utilize the node resources. During node communication, the clustering process effectively eliminates excessive energy waste. AES is used for encryptions and level 2 security, while ECC is used for generating pairing keys for node identification in level 1 security. Graded coprime keys are created for private and public keys, and encryption functions are performed to provide security mechanisms. The CH checks every CM's authentication and provides data protection using the hybrid AES-ECC process. Each node is provided with security keys once the clustering process is completed. AES and ECC security schemes are used to produce certain security keys. The BS generates public keys and distributes them to all other nodes based on their NID. This process strengthens security measures and reduces energy consumption between nodes at the same time. A network simulator tool analyzes simulation presentations of the networks. The proposed scheme's efficiency is measured by comparing packet transmission speeds, which reveals

that it is 18% more effective than the standard elliptic curve cryptography-based secure authentication scheme (ECCSAS).

### 2.1. Discussion

To meet security standards such as confidentiality, authentication, and integrity, a variety of cryptographic algorithms have been proposed as discussed in the literature. The most important factor to consider in WBANs is their limited resources such as processing power, battery life, and communication range. Most of the existing solutions have suffered from network overhead and issues related to complexity. The WBAN network requires a lightweight security solution because traditional cryptography techniques are complex and consume more network resources. A lightweight security solution supports the sensor node battery management, as changing batteries and charging techniques are difficult to manage.

## 3. Proposed Lightweight Hybrid Cryptography Algorithm

A security protocol for WSNs is proposed, employing cryptographic techniques that use fewer resources compared to other cryptographic algorithms to overcome the data confidentiality issue in WSNs. The proposed security protocol known as the Lightweight Hybrid Cryptography Algorithm (LWHCA) is developed using a cryptographic-based technique by using encryption and decryption algorithms: AES [29] and secure efficient encryption algorithm (SEEA) [30]. We employ the ECC algorithm to generate a highly secure key for encryption and decryption of data. The LWHCA algorithm is designed with some assumptions as discussed in Sec. 2 There are two types of nodes: SNs and GNs. SNs are known as weak nodes and GNs are considered to be strong nodes. This categorization is made to improve system performance and protection. Key generation and sharing are crucial aspects of the protocol. The SEEA symmetric algorithm is used for encryption and decryption of data in SN-to-SN communication, and both AES and SEEA algorithms are used for encryption and decryption of data in GN-to-GN communication. Communication in SN-to-GN will be the same as SN-to-SN communication. The ECC algorithm is used for only one purpose – generating and sharing a key due to its smaller key size compared to other asymmetric algorithms.

In an LWHCA, the first phase involves generating public and private keys to create session keys for encryption and decryption of data during communication between nodes. Each node will generate a key with the help of asymmetric techniques, initiated by any random node in WSNs, and then it will share it with other nodes. All the nodes in WSNs generate public and private keys by using cryptographic techniques and each node will have different public and private keys. Public and private keys will be exchanged between nodes by using the

Elliptic Curve Diffie–Hellman (ECDH) algorithm. After the exchange of public keys, the nodes will create a session key, known as an asymmetric key. This session key will be used in the algorithm for encryption and decryption of data during communication between nodes. This ensures the establishment of secure communication between nodes.

To secure data communication in WSNs while using fewer resources, two communication mechanisms are employed. In SN-to-SN communication, first, nodes share a public key between them to create a session key. After that, they only use the SEEA symmetric algorithm with the session key to encrypt and decrypt data. In GN-to-GN communication, they also share a public key between them to securely create a session key. However, after that, two algorithms are employed for encryption and decryption data. First, the plaintext is divided into two equal blocks, the first block is encrypted using AES and the second block is encrypted with SEEA using the session key. Both algorithms are symmetric, and the same procedure is employed for the decryption process. These two mechanisms enable nodes to securely communicate and use less resources. The communication in SN-to-GN follows the same procedure as SN-to-SN, as SNs are weak nodes and GNs are strong nodes.

We assume that nodes are already authenticated, and the primary focus is on sharing a key between them, and encrypting and decrypting messages between them. To communicate between two nodes, the first node authenticates to the other to ensure they are not unauthorized or malicious nodes in the network. After authentication, the key sharing process starts. Both nodes have already created their public and private keys using ECC. Node A sends its ECC public key to node B, and node B sends its ECC public key to node A. When both nodes receive each other's public keys, they start to generate a symmetric key, known as a session key, using ECDH. If they use the same elliptic curve parameters, they will create the same symmetric key; otherwise, both nodes will have a different symmetric key and they cannot communicate with each other successfully.

When both nodes create a session key, then node A will start sending an encrypted message to node B. The plaintext message is converted into ciphertext using encryption/decryption algorithms and the session key used for it. When node B receives the ciphertext, it uses the session key on the decryption algorithm to decrypt the ciphertext into the plaintext. The communication process between nodes continues until they stop communicating with each other. Figure 2 shows the communication between two nodes where node A shares its public key A with node B, and in response, node B shares its public key B. A session key is then created by both nodes, and encrypted data transmission is initiated using the session keys from both nodes.

Figure 3 illustrates the cryptography communication mechanisms between two nodes, A and B. In step 1, node A sends a request to another node for

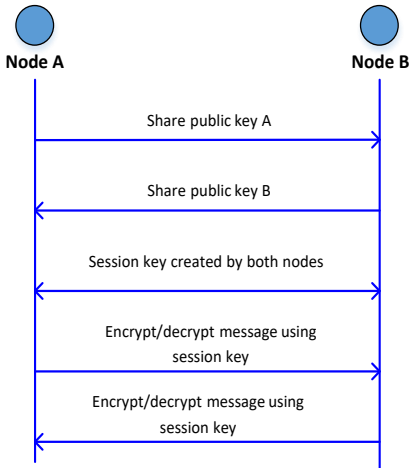


Fig. 2. Communication between node A and node B.

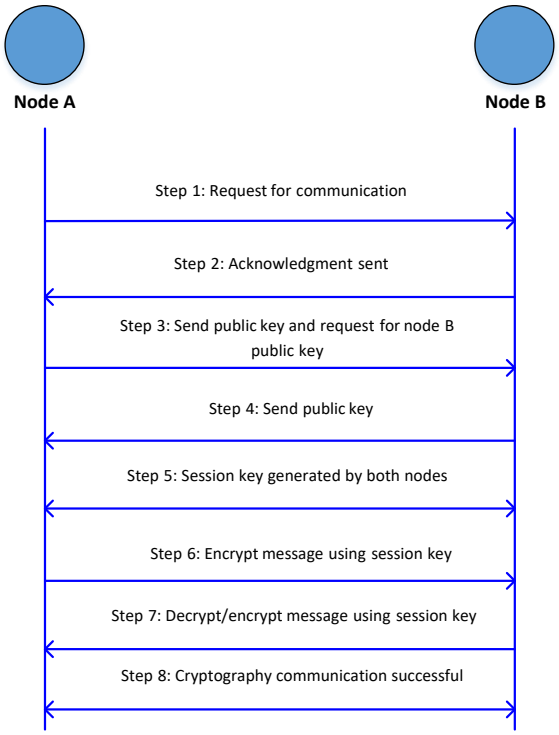


Fig. 3. Cryptography communication mechanisms.

communication, such as requesting to communicate with node B. In step 2, if node B is available, it replies with an acknowledgment to node A, indicating that it is available for communication. In step 3, node A sends its public key to

node B and also requests node B's public key. In step 4, when node B receives the public key from node A, it also sends its public key to node A.

In step 5, both nodes create a session key for encryption and decryption of messages, using the other node's public key and its private key. In step 6, after creating the session key, node A encrypts a message using the session key and sends it to node B. In step 7, when node B receives an encrypted message from node A, it decrypts the message using the same session key, and if node B also wants to send a message to node A, it encrypts the message and sends it to node A. In step 8, after successful encryption and decryption between the two nodes, the communication process terminates. Figure 3 presents the cryptography communication mechanism.

The flowchart shown in Fig. 4 illustrates the flow of cryptography mechanism of communication between two nodes in general. At the start of the mechanism,

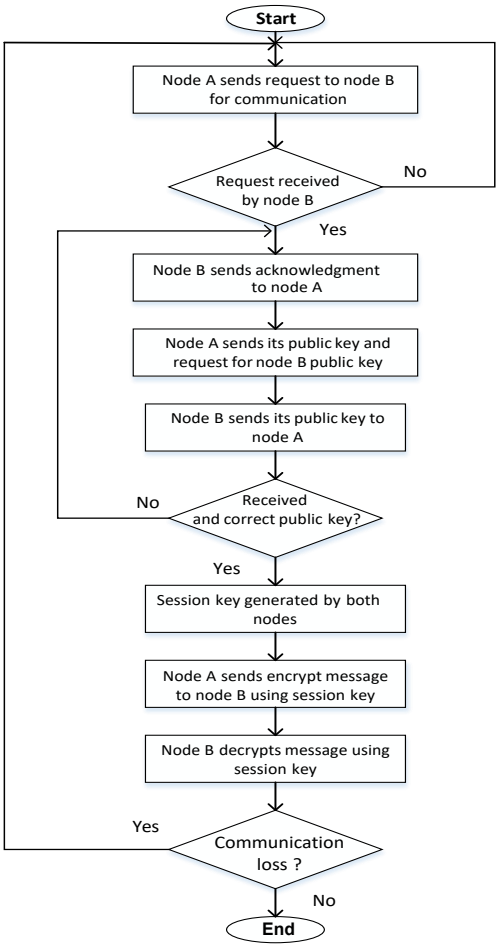


Fig. 4. Flowchart of node-to-node cryptography communications.

node A sends a request to node B for communication. Node A checks whether an acknowledgment is received from node B. If not, node A resends the request to node B. If an acknowledgment is received from node B, it means that node B is active and available for communication.

Then, node A sends its public key to node B and also requests node B's public key. When node B receives node A's public key, it sends its public key to node A to generate a session key between them. Now, the public keys are shared between the two nodes, and they check if they have the correct public keys or not. If not, the whole process restarts from requesting the public keys from each node until the correct public keys are obtained. If the public key is correct, the process moves to the next step. Using the public key of each node, they create a symmetric key known as the session key, and both nodes know the session secret key for encryption and decryption of messages during communication between them. Once the session key is generated, node A encrypts the message using the secret key and sends it to node B. Node B receives the encrypted message from node A and decrypts the message from the session key. After that, it checks if the communication is completed or lost. If communication is lost and not completed, the entire process restarts from the initial point such as requesting communication. If communication is completed and not lost, the communication

Table 1. Different notation used in proposed methodology.

| Symbol | Description |
|---|---|
| SN | Sensor Node |
| GN | Gateway Node |
| a | Private Key Node A |
| A | Public Key Node A |
| b | Private Key Node B |
| B | Public Key Node B |
| P | Plaintext |
| K | Secret Key |
| C | Ciphertext |
| BP | Plaintext in Binary Form |
| BK | The Secret Key in Binary Form |
| BC | Ciphertext in Binary Form |
| M1 | First Half of Plaintext |
| M2 | Second Half of Plaintext |
| C1 | First Half of Ciphertext |
| C2 | Second Half Ciphertext |
| CT | Concatenated Ciphertext |

between nodes is successfully completed and the process ends. Figure 4 shows the flowchart of node-to-node cryptography communications.

The proposed solution – LWHCA – is divided into three main phases: key sharing, node-to-node communication, and gateway-to-gateway node communication. These three modules are discussed in the following subsections. Table 1 shows the different notations used in the proposed methodology.

### 3.1. Key exchanged mechanism

There are two types of nodes in this network: SNs known as weak nodes and GNs known as strong nodes. The quantity of SNs will be large in WSNs to collect information from their sensing and the quantity of GNs will be smaller compared to the SNs. Two different scenarios are deployed in the network for node communication in WSNs: SN-to-SN and GN-to-GN communications. In SN-to-SN communication, the nodes first generate a pairwise key (a public and private key) by using ECC, and the public key is then shared with other nodes using ECDH. When the public key is shared between SNs, they compute a session key by using ECDH and this session key is used for encryption and decryption of data in algorithm. Only one algorithm is used for encryption and decryption of data in SN, making it more lightweight and efficient compared to other traditional cryptographic algorithms. In GN-to-GN communication, the same procedure is followed for key generation. First, GNs compute pairwise keys and share their public keys with other GNs to compute a session key using ECDH. The session key is then used for encryption and decryption of data. GN is a powerful node, so there are two algorithms used for encryption and decryption of data, which makes communication more secure in WSNs, and both algorithms are lightweight, requiring fewer resources compared to traditional cryptographic algorithms.

We have already discussed that for key exchange between nodes in LWHCA, we use the ECDH algorithm that provides better security compared to other algorithms. First, both nodes select the ECDH domain parameter, and they agree on these parameters. Otherwise, a wrong key is generated, and they will not successfully communicate with each other. The ECDH domain parameters, being public information, are known to both nodes, and this information creates a session key. In ECDH, node A selects its private key and computes its public key with the help of the initial point and private key. At the same time, node B also selects its private key and computes its public key. Then, node A shares its public key with node B and node B shares its public key with node A. When both nodes receive each other's public key, node A computes the session key with the help of node B's public key and its own private key, while node B computes the session key with the help of node A's public key and its own

private key. If the domain parameters and public key values are correct, the joint secret key between them will be correct; otherwise, both will create wrong keys and they cannot communicate with each other due to the fact that the decryption process is not possible. Figure 5 shows the key exchange between two nodes using ECDH.

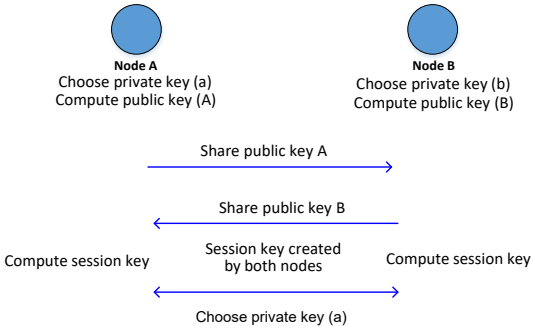


Fig. 5. Key exchange between nodes.

### 3.2. Node-to-Node communication (encryption)

SNs are weak nodes and cannot use multiple algorithms or heavy algorithms due to limited resources. Therefore, only one algorithm is employed for data encryption and decryption in LWHCA for SN-to-SN communication, reducing the overhead on SNs. When the key is successfully exchanged between two nodes using ECDH, they can perform encryption and decryption processes. When node A wishes to communicate with node B, the algorithm employed is based on an asymmetric algorithm with XNOR operation, which makes computation cost fast compared to other algorithms. Node A first converts the message known as plaintext into ASCII code as a decimal number, which is then converted into a binary number. The session key, generated by both nodes, is also converted into a binary number. When both plaintext and session key are converted into binary format, an XNOR operation is performed between them and this operation is performed between the session key and the plaintext. Then, a circular left shift operation is executed to change the position of binary digits from its actual position to a complex one. After the circular left shift operation, the result undergoes a 1's complement operation. The binary number is then divided into two equal blocks, denoted as set 1 and set 2. Set 1 block and set 2 block swap with each other to change their position. In the final result, we obtain the ciphertext in binary format. Then, the binary ciphertext is converted into ASCII code as a decimal number and again converted into text form. Once the encryption process is successfully performed by node A, then it is sent to node B, in encrypted form, and no one can read or decrypt it unless they know the session key. Figure 6 shows the encryption process performed by sensor node A.

**Algorithm 1.** SEEA encryption.

| | | |
|---|---|---|
| **Input** | : | *Plaintext (P), Secret Key (K)* |
| **Output** | : | *Ciphertext (C)* |
| 1 | | *Start* |
| 2 | : | *Divide plaintext into a 128-bit number of blocks* |
| 3 | : | ***If*** *the number of blocks is 128-bit, not full* |
| 4 | : | *Padding null in the last block* |
| 5 | : | ***Else*** |
| 7 | : | *BP = convert plaintext to binary* |
| 8 | : | *BK = convert ECDH secret key to binary* |
| 9 | : | *T1 = XNOR BP with BK* |
| 10 | : | *T2 = Circular left shift one of T1* |
| 11 | : | *T3 = 1's complement of T2* |
| 12 | : | *Set 1 = Divide T3 into the first half* |
| 13 | : | *Set 2 = Divide T3 into second half* |
| 14 | : | *C = Swap Set 1 and Set 2* |
| 15 | : | *Send ciphertext C to the other sensor node* |
| 16 | : | *End* |

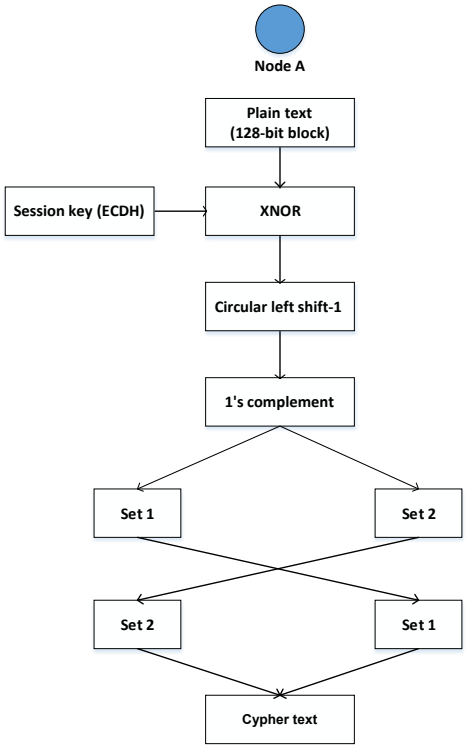


Fig. 6. Data encryption by sensor node A.

### 3.3. Node-to-Node communication (decryptions)

When node B receives a ciphertext from node A, the same algorithm used for encryption is employed to decrypt the cipher message. The decryption process is the opposite of the encryption process. At first, the ciphertext, which is in text form, is converted into ASCII code in decimal numbers and then into a binary format. Then, the binary ciphertext blocks are divided into two blocks with equal length, denoted as set 1 and set 2. Next, these blocks swap their block position to their original position, reversing the swapping position performed during the encryption process. Following this, the 1's complement is performed on them. After that, a circular right shift operation is performed on it, resulting in a binary number in the original position compared to the encryption algorithm.

A session key is generated by both node A and node B. Node B converts the session key into a binary number and performs an XNOR operation with it. If the wrong session key is used, it will give a wrong binary number, making the plaintext unreadable or meaningless. When the XNOR operation is performed successfully, it successfully decrypts the message into the original message, but the message is in binary format. The binary number is then converted into ASCII code in decimal and further into text from which node A sent the original message in encrypted form. Node B successfully decrypts the message. Figure 7 shows the decryption process by sensor node B.

**Algorithm 2.** SEEA decryption.

| | |
|---|---|
| **Input** | : *Ciphertext (C), Secret Key (K)* |
| **Output** | : *Plaintext (P)* |
| 1 | : *Start* |
| 2 | : *Divide ciphertext into a 128-bit number of blocks* |
| 7 | : *BC = Convert ciphertext into a binary number* |
| 8 | : *BK = Convert ECDH secret key into a binary number* |
| 9 | : *Set 1 = Divide BC into first half ciphertext* |
| 10 | : *Set 2 = Divide BC into second half ciphertext* |
| 11 | : *T3 = Swap Set 1 and Set 2* |
| 12 | : *T2 = 1's complement of T3* |
| 13 | : *T1 = Circular right shift one of T2* |
| 14 | : *P1 = XNOR T1 with BK* |
| 15 | : *P2 = Convert P1 into ASCI code decimal* |
| 16 | : *P = Convert P2 into character* |
| 17 | : *End* |

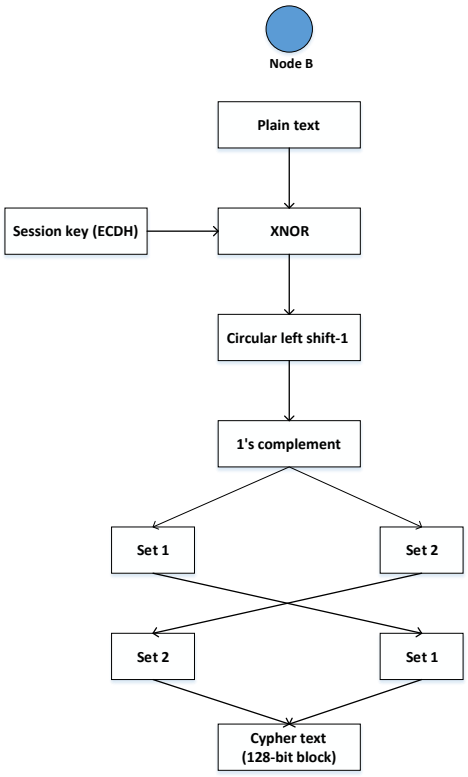


FIG. 7. Data decryption by sensor node B.

### 3.4. Encryption/decryption of GN-to-GN communication

In the LWHCA algorithm for GNs, a hybrid algorithm is adopted, combining two encryption/decryption algorithms, AES and SEEA, for GN-to-GN communication. They are executed in parallel. These algorithms are only used for data encryption between nodes. Both algorithms used ECDH keys, which makes data more secure, thus forming a hybrid cryptography scheme for data encryption and decryption. AES is a symmetric algorithm, which is fast and secure compared to other traditional algorithms. SEEA is also a symmetric algorithm, making it very fast compared to asymmetric algorithms.

When GN A wants to communicate with GN B via secure data transmission, GN A first converts data into ciphertext and sends it to GN B. After that, GN B converts the ciphertext into plaintext to get a readable text form. During encryption process in GN A, the plaintext is divided into 128-bit blocks, and if the last block is not a full 128 bits, it is padded with null bytes to ensure all blocks are of equal size (128-bit n blocks). After that, the blocks are divided into two equal sub-blocks. AES encryption algorithm is used in the first sub-block to produce ciphertext 1, while the SEEA encryption algorithm is used at the

same time in the second sub-block to obtain ciphertext 2. Subsequently, GN A combines ciphertext 1 and 2, which is the final ciphertext and sends it to GN B. For the decryption process, GN B receives the ciphertext from GN A, containing ciphertext 1 and ciphertext 2. Then, AES decryption algorithm is used on ciphertext 1 to get plaintext 1, and SEEA decryption algorithms are used on ciphertext 2 to get plaintext 2. When both ciphertexts are converted into plain-

**Algorithm 3.** Hybrid GN encryption.

1 : *Start*
2 : *Get plaintext*
3 : *Generate secret key (K) by ECDH*
4 : ***If*** *plaintext 128-bit n number block does not divide equally into* $\frac{n}{2}$
5 : ***then***
6 : *Padding in the last block to make even n number block 128-bit*
7 : ***Else***
8 : *Divide plaintext equal into* $\frac{n}{2}$ *block (each block 128-bit)*
9 : *M1 = First half plaintext*
10 : *C1 = AES Encrypt of M1*
11 : *Encrypt1 = C1*
12 : *M2 = Second half plaintext*
13 : *C2 = SEEA Encrypt of M2*
14 : *Encrypt2 = C2*
15 : *C = Encrypt1 + Encrypt2 (concatenate all the two-ciphertext values)*
16 : *Send ciphertext to another GN*
17 : *End*

**Algorithm 4.** Hybrid GN decryption.

1 : *Start*
2 : *Secret key already generated by ECDH*
3 : *Get ciphertext*
4 : *Divide ciphertext into n number of block 128bit*
5 : *C1 = First half ciphertext*
6 : *M1 = AES Decrypt of C1*
7 : *Decrypt1 = M1*
8 : *C2 = Second half ciphertext*
9 : *M2 = SEEA Decrypt of C2*
10 : *Decrypt2 = M2*
11 : *P = Decrypt1 + Decrypt2 (concatenate all the two-plaintext values)*
12 : *End*

texts, they are combined to give the final plaintext, which will be an original message sent by GN A to GN B. This successfully decrypts the message received by GN A. Figure 8 shows hybrid cryptography algorithm for GN.

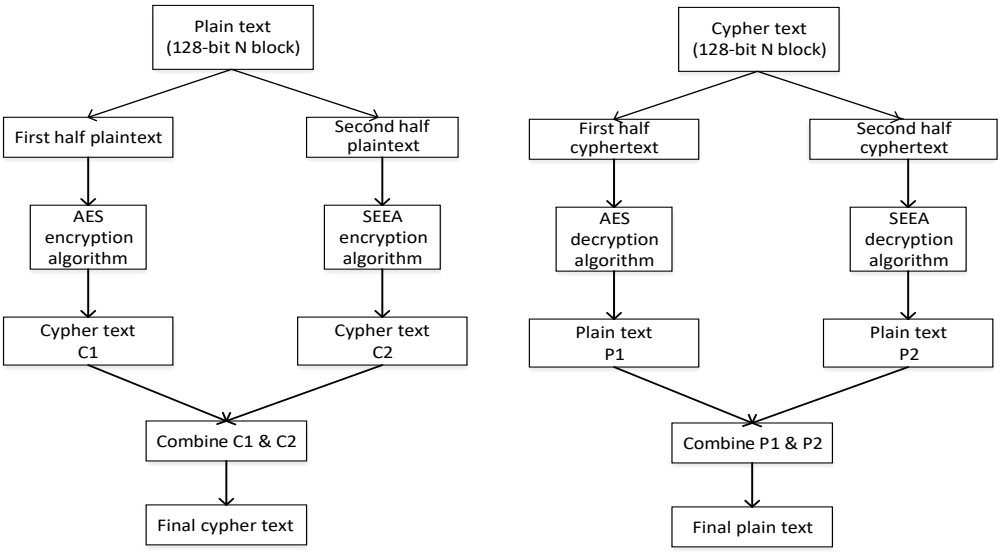


FIG. 8. Hybrid cryptography algorithm for GN.

We discussed the GN-to-GN communication algorithm and SN-to-SN communication algorithm in great detail, determining which algorithm will be used to encrypt and decrypt data securely considering the limitations of these devices. We also discussed in detail the key generation process between nodes, divided into three phases: key generation, SN-to-SN communication, and GN-to-GN communication. We already established that the communication between SN-to-GN algorithms will be the same as the communication between SN-to-SN, due to SN being a weak node and incapable to support multiple algorithms for data encryption. A 128-bit encryption block is used for both algorithms to secure the data efficiently. The asymmetric algorithm is used for key generation and the symmetric algorithm is used for data encryption and decryption.

## 4. Results and analysis

### 4.1. Symmetric and asymmetric cryptography

Symmetric cryptography is faster compared to asymmetric cryptography, but the challenge lies in key sharing in symmetric cryptography. Asymmetric algorithms are used for generating pairs of keys for communication, attackers cannot obtain information about the other key information even when one key is com-

promised. The issue in asymmetric cryptography is its slow process of encryption and encryption of data. To enhance network security for communication, hybrid techniques should be used and the selection of the algorithm for hybrid techniques must be conducted according to network resources such as processing speed, battery life, storage capacity, and memory. In hybrid techniques, ECC is a lightweight algorithm that is one of the best for secure data communication among nodes. Compared to the RSA algorithm, ECC supports a shorter key length and provides high security. ECDH is used for key generation, providing high security compared to other cryptography techniques for key generation. SEEA and AES are asymmetric cryptography algorithms, providing fast encryption and decryption processes compared to other symmetric or asymmetric algorithms. So, in the context of gateway-to-gateway node communication and SN-to-SN communication, an approach is adopted, where half of the plaintext is encrypted/decrypted using AES and the other half using SEEA with ECC keys. High security and fast performance speed are ensured, as only one algorithm is used for encryption and decryption in SN-to-SN communication.

### 4.2. Computation analysis

In an LWHCA, during communication between two nodes of same type and between GNs (GN-to-GN communication), they utilize pair-wise keys to generate symmetric keys for encryption/decryption algorithms. As already explained, communication between two SNs uses only one algorithm known as the SEEA symmetric algorithm, recognized for its very fast computation compared to other symmetric or asymmetric algorithms.

GN plays a pivotal role and it is a powerful node, and that is why communication between GNs uses two algorithms, both lightweight and providing fast computation, along with another hybrid algorithm, i.e., encrypting/decrypting half of the plaintext is conducted with AES symmetric algorithm, and the other half plaintext with the SEEA cryptography. This dual-algorithm approach in GNs makes it more secure. The performance of LWHCA was analyzed based on encryption, decryption, and ciphertext size of the message in GN-to-GN communication with different sizes of plaintext in terms of encryption and decryption process. To see how LWHCA performs compared to other algorithms, it is compared with already existing hybrid algorithms such as THCA [8] and HCA [26] in the form of a graph for better understanding. Datasets of varying sizes: 609, 25 615, 35 080, 61 386 and 184 162 bytes are used due to the fact that other algorithms also use the same text size. As the data size increases, encryption/decryption time also increases, making it easier to understand the performance differences among algorithms. Since different algorithms exhibit varying performance characteristics with different data sizes, it is recommended to choose algorithms

accordingly. In addition, the chosen data is also not equally divided into 128-bit blocks, so padding is needed. This impacts the ciphertext and provides insights into algorithm performance.

The encryption time is the time taken by the encryption algorithm to encrypt the plaintext and convert it into ciphertext. The decryption time is the time taken by the decryption algorithm to decrypt a ciphertext and convert it into plaintext. It is observed that the LWHCA technique takes the shortest time to encrypt and decrypt data. This is due to dividing plaintext into two parts for LWHCA in GNs and executing the algorithm in parallel to reduce encryption and decryption time and the lightweight symmetric algorithm used.

The results of LWHCA are presented in terms of time and compared to different sizes of plaintext. As we increase the size of plaintext in terms of bytes, encryption and decryption times increase as well. The encryption and decryption results show that LWHCA's encryption and decryption performance outperforms other two algorithms, as shown in Figs. 9 and 10. In Fig. 9, it is observed that

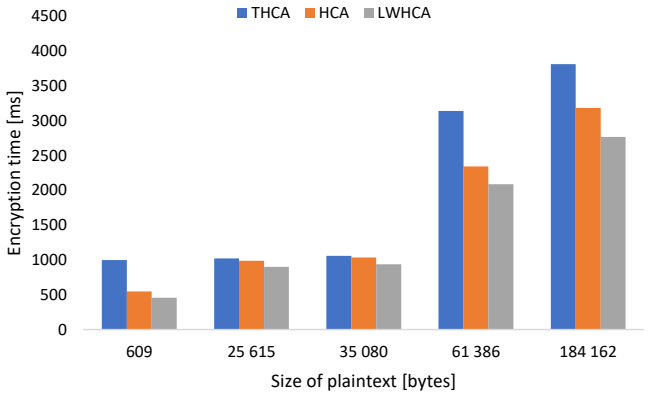


Fig. 9. Hybrid algorithms' encryption.

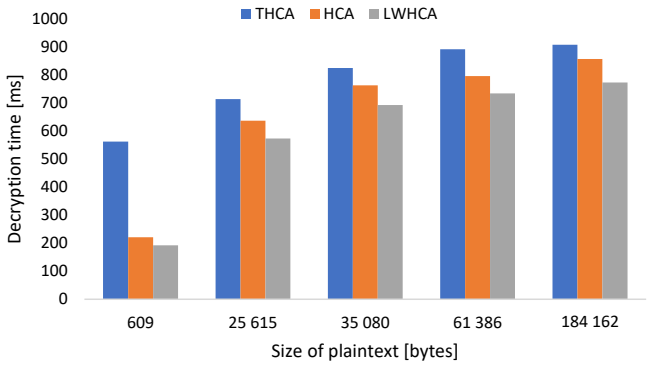


Fig. 10. Hybrid algorithms' decryption.

THCA and HCA algorithms encrypt slower compared to LWHCA across different sizes of plaintext. Figure 10 illustrates that THCA and HCA algorithms' decryption is slower compared to LWHCA across different plaintext sizes, highlighting LWHCA's better performance in encryption and decryption processes compared to the other two algorithms.

Then, we compared the computation time, which is the combined duration of encrypting and decrypting algorithms. Computation time is the time taken by the encryption and decryption algorithms to encrypt plaintext and convert it into ciphertext and decrypt a ciphertext into plaintext. This is compared with other algorithms. Figure 11 shows the result of computation in comparison with other algorithms across different text sizes. When the size of plaintext increases, the computation time for encryption and decryption also increases, due to mathematical calculation dependent on system performance. This helps in understanding the proposed methodology's efficiency when handling large volumes of data in encrypted or decrypted form. As we can see in Fig. 11, the THCA algorithm takes more computation time compared to HCA and LWHCA. Additionally, as the size of plaintext increases in terms of bytes, the computation time increases as well. The LWHCA algorithm takes the shortest computation time compared to other algorithms, highlighting its better performance overall. Figure 11 shows the computation time of the hybrid algorithms.

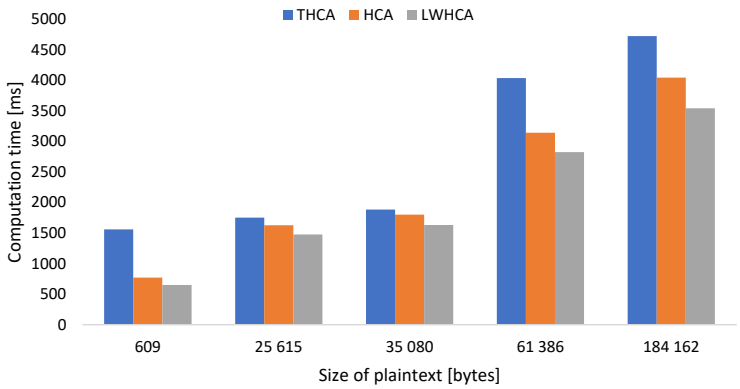


Fig. 11. Hybrid algorithms' computation time.

Figure 12 illustrates the ciphertext size, representing the output of the encryption process. It shows that THCA, HCA, and LWHCA algorithms' ciphertext sizes are very close to the plaintext size, with minimal reduction compared to existing algorithms. However, LWHCA maintains a ciphertext size very close to the plaintext size.

The topology of the WSN is assumed to be made up of twenty nodes for NS2 networks. The nodes are distributed around the network at random. During data

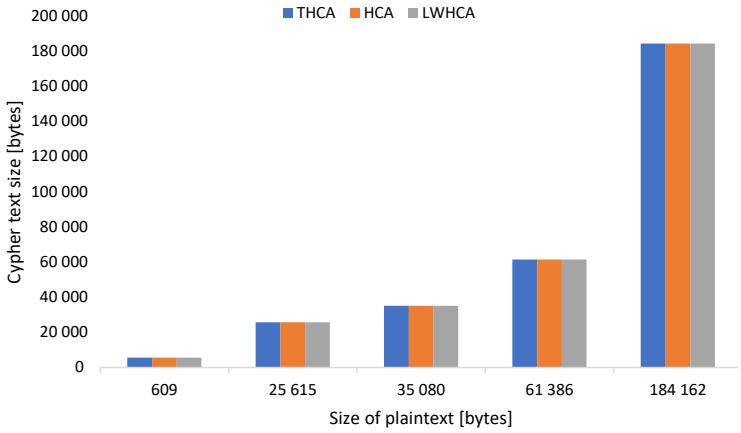


FIG. 12. Hybrid algorithms' ciphertext size.

transfer between several nodes, many circumstances are expected to arise. The information about the other nodes in the WSN must be provided to each node. When evaluating energy consumption, both the energy used during the execution of cryptographic algorithms and the energy used during communication are taken into consideration. The energy required to execute a cryptographic method is simply the product of the method's average power consumption and its execution time. Simulations were conducted to calculate energy use and the obtained results were compared with the two other hybrid algorithms, as shown in Fig. 13.

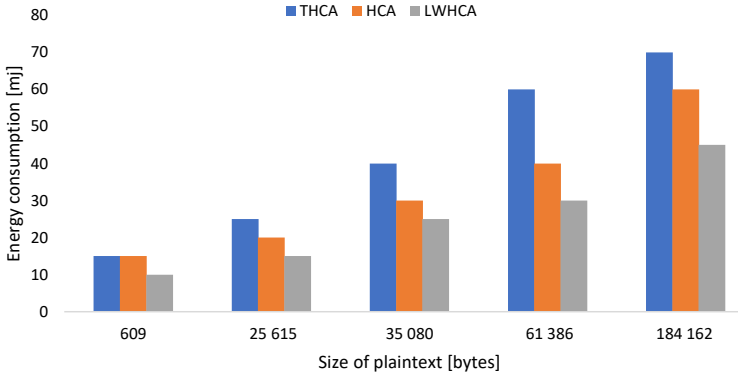


FIG. 13. Hybrid algorithms' energy consumption.

The transmission of incorrect packets over any link between two nodes can cause connection failures. Due to execution time delays, some packets may be dropped, and an increase in the number of dropped unsafe packets leads to the link being temporarily turned off, subsequently making the network to switch

to a different path. Then, we compared the packet drop ratio with other algorithms and it was observed that LWHCA provides the lowest rate of packet drops compared to other algorithms, as shown in Fig. 14.

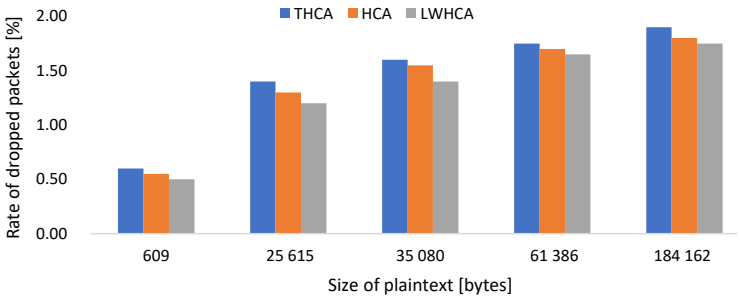


Fig. 14. Hybrid algorithms' rated of dropped packets.

## 5. Conclusion

The Lightweight Hybrid Cryptography Algorithm (LWHCA) is based on ECC, AES, and SEEA cryptography algorithms. The proposed cryptographic algorithm is based on the ECC method, considered a powerful public-key cryptographic mechanism for generating session keys for data encryption and decryption, thereby protecting the WBAN. The symmetric algorithm method is used for data encryption and decryption. In the last few years, both academics and industry have increasingly adopted ECC techniques, which rely on solving the discrete logarithm problem. Solving such a problem takes exponential time. On the other hand, RSA takes sub-exponential time to solve discrete logarithm problems. ECDH (a variant of ECC) is a technique that employs a small key size suitable for secure network communication. ECC has numerous parameters that can be employed more efficiently compared to RSA. A small key size, high security, fast computation, low power consumption, and lightweight make ECDH the best choice for session key generation. The symmetric algorithm is used to efficiently secure the communication over the network. Based on experiment results, it was proven that LWHCA takes very little time to encrypt and decrypt data, and the size of ciphertext is very close to the original plaintext size. We also compared LWHCA with existing hybrid algorithms for WBAN, and concluded that LWHCA provides an efficient encryption/decryption process.

For future work, a single or hybrid lightweight secure algorithm will be employed for encrypting and decrypting data for both SN and GN. Additionally, we will focus on the user and GN authentication using the same algorithm. The proposed solution will also be extended to other networks such as Internet of Things (IoT), WSN, and smart cities networks.

## References


1. M. Anwar, A.H. Abdullah, A. Altameem, K.N. Qureshi, F. Masud, M. Faheem, Y. Cao, R. Kharel, Green communication for wireless body area networks: Energy aware link efficient routing approach, *Sensors*, **18**(10): 3237, 2018, doi: 10.3390/s18103237.

2. M. Anwar, A.H. Abdullah, K.N. Qureshi, A.H. Majid, Wireless body area networks for healthcare applications: An overview, *TELKOMNIKA*, **15**(3): 1088–1095, 2017, doi: 10.12928/TELKOMNIKA.v15i3.5793.

3. R. Qazi, K.N. Qureshi, F. Bashir, N.U. Islam, S. Iqbal, A. Arshad, Security protocol using elliptic curve cryptography algorithm for wireless sensor networks, *Journal of Ambient Intelligence Humanized Computing*, **12**: 547–566, 2021, doi: 10.1007/s12652-020-02020-z.

4. M.F. Moghadam, M. Nikooghadam, M.A.B. Al Jabban, M. Alishahi, L. Mortazavi, A. Mohajerzadeh, An efficient authentication and key agreement scheme based on ECDH for wireless sensor network, *IEEE Access*, **8**: 73182–73192, 2020, doi: 10.1109/ACCESS.2020.2987764.

5. K. Awan, K.N. Qureshi, M. Mehwish, Wireless body area networks routing protocols: A review, *Indonesian Journal of Electrical Engineering Computer Science*, **4**(3): 594–604, 2016, doi: 10.11591/ijeecs.v4.i3.pp594-604.

6. K.N. Qureshi, S. Din, G. Jeon, F. Piccialli, Link quality and energy utilization based preferable next hop selection routing for wireless body area networks, *Computer Communications*, **149**: 382–392, 2020, doi: 10.1016/j.comcom.2019.10.030.

7. P. Sinha, V.K. Jha, A.K. Rai, B. Bhushan, Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: A survey, [in:] *2017 International Conference on Signal Processing and Communication (ICSPC)*, Coimbatore, India, pp. 288–293, 2017, doi: 10.1109/CSPC.2017.8305855.

8. R. Rizk, Y. Alkady, Two-phase hybrid cryptography algorithm for wireless sensor networks, *Journal of Electrical Systems Information Technology*, **2**(3): 296–313, 2015, doi: 10.1016/j.jesit.2015.11.005.

9. A. Karakaya, S. Akleylek, A survey on security threats and authentication approaches in wireless sensor networks, [in:] *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, Antalya, Turkey, pp. 1–4, 2018, doi: 10.1109/ISDFS.2018.8355381.

10. M.S. Yousefpoor, E. Yousefpoor, H. Barati, A. Barati, A. Movaghar, M. Hosseinzadeh, Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review, *Journal of Network and Computer Applications*, **190**: 103118, 2021, doi: 10.1016/j.jnca.2021.103118.

11. S.B. Sadkhan, A.O. Salman, *A survey on lightweight-cryptography status and future challenges*, [in:] *2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*, Wasit-Kut, Iraq, pp. 105–108, 2018, doi: 10.1109/ICASEA.2018.8370965.

12. K.N. Mishra, C. Chakraborty, A novel approach towards using big data and IoT for improving the efficiency of m-health systems, [in:] *Advanced Computational Intelligence Techniques for Virtual Reality in Healthcare*, D. Gupta, A. Hassanien, A. Khanna [Eds.], Studies in Computational Intelligence, Vol. 875, pp. 123–139, Springer, 2020, doi: 10.1007/978-3-030-35252-3_7.

13. A. Kishor, C. Chakraborty, W. Jeberson, Intelligent healthcare data segregation using fog computing with internet of things and machine learning, *International Journal of Engineering Systems Modelling and Simulation*, **12**(2–3): 188–194, 2021, doi: 10.1504/IJESMS.2021.115533.

14. C. Chakraborty, *Performance analysis of compression techniques for chronic wound image transmission under smartphone-enabled tele-wound network*, [in:] *Research Anthology on Telemedicine Efficacy, Adoption, and Impact on Healthcare Delivery*, IGI Global, pp. 345–364, 2021, doi: 10.4018/978-1-7998-8052-3.ch018.

15. P. Gope, T. Hwang, A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks, *IEEE Transactions on Industrial Electronics*, **63**(11): 7124–7132, 2016, doi: 10.1109/TIE.2016.2585081.

16. A. Ghani, K. Mansoor, S. Mehmood, S.A. Chaudhry, A.U. Rahman, M. Najmus Saqib, Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key, *International Journal of Communication Systems*, **32**(16): e4139, 2019, doi: 10.1002/dac.4139.

17. M. Elhoseny, H. Elminir, A. Riad, X. Yuan, A secure data routing schema for WSN using elliptic curve cryptography and homomorphic encryption, *Journal of King Saud University-Computer and Information Sciences*, **28**(3): 262–275, 2016, doi: 10.1016/j.jksuci.2015.11.001.

18. A. Praveena, S. Smys, Efficient cryptographic approach for data security in wireless sensor networks using MES V-U, [in:] *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, pp. 1–6, 2016, doi: 10.1109/ISCO.2016.7726911.

19. J. Srinivas, S. Mukhopadhyay, D. Mishra, Secure and efficient user authentication scheme for multi-gateway wireless sensor networks, *Ad Hoc Networks*, **54**: 147–169, 2017, doi: 10.1016/j.adhoc.2016.11.002.

20. D. Wang, W. Li, P. Wang, Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks, *IEEE Transactions on Industrial Informatics*, **14**(9): 4081–4092, 2018, doi: 10.1109/TII.2018.2834351.

21. S. Prakash, A. Rajput, Hybrid cryptography for secure data communication in wireless sensor networks, [in:] *Ambient Communications and Computer Systems*, G. Perez, S. Tiwari, M. Trivedi, K. Mishra [Eds.], Advances in Intelligent Systems and Computing, Vol. 696, pp. 401–410, Springer, 2018, doi: 10.1007/978-981-10-7386-1_50.

22. S. Farooq, D. Prashar, K. Jyoti, Hybrid encryption algorithm in wireless body area networks (WBAN), [in:] *Intelligent Communication, Control and Devices*, R. Singh, S. Choudhury, A. Gehlot [Eds.], Advances in Intelligent Systems and Computing, Vol. 624, pp. 401–410, Springer, 2018, doi: 10.1007/978-981-10-5903-2_41.

23. M. Alotaibi, An enhanced symmetric cryptosystem and biometric-based anonymous user authentication and session key establishment scheme for WSN, *IEEE Access*, **6**: 70072–70087, 2018, doi: 10.1109/ACCESS.2018.2880225.

24. J. Jung, J. Kim, Y. Choi, D. Won, An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks, *Sensors*, **16**(8): 1299, 2016, doi: 10.3390/s16081299.

25. V.S. Naresh, R. Sivaranjani, N.V.E.S. Murthy, Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor networks, *International Journal of Communication Systems*, **31**(15): e3763, 2018, doi: 10.1002/dac.3763.

26. K.M. Abdullah, E.H. Houssein, H.H. Zayed, New security protocol using hybrid cryptography algorithm for WSN, [in:] *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, pp. 1–6, 2018, doi: 10.1109/CAIS.2018.8442003.

27. N. Gupta, V. Kapoor, Hybrid cryptographic technique to secure data in web application, *Journal of Discrete Mathematical Sciences Cryptography*, **23**(1): 125–135, 2020, doi: 10.1080/09720529.2020.1721872.

28. M. Yuvaraju, K.A. Pranesh, Energy proficient hybrid secure scheme for wireless sensor networks, *Wireless Personal Communications*, **117**: 747–767, 2021, doi: 10.1007/s11277-020-07895-x.

29. C.-W. Hung, W.-T. Hsu, Power consumption and calculation requirement analysis of AES for WSN IoT, *Sensors*, **18**(6): 1675, 2018, doi: 10.3390/s18061675.

30. S. Ali *et al.*, An efficient cryptographic technique using modified Diffie–Hellman in wireless sensor networks, *International Journal of Distributed Sensor Networks*, **16**(6): 1550147720925772, 2020, doi: 10.1177/1550147720925772.

*Received April 28, 2022; revised version August 18, 2022;*
*accepted September 22, 2022; published online April 26, 2024.*