IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 July 2023, accepted 1 August 2023, date of publication 4 August 2023, date of current version 10 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3302251

RESEARCH ARTICLE

A Lightweight Authentication Framework for Fault-Tolerant Distributed WSN

KOLLU SIVA SAI[1], RADHAKRISHNA BHAT[1], (Member, IEEE), MANJUNATH HEGDE[2], AND J. ANDREW[1]

[1]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
[2]Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

Corresponding author: Radhakrishna Bhat (rsb567@gmail.com)

ABSTRACT The vast production of resource-constrained wireless communication devices and the development of various techniques in recent years opens room for security concerns to overcome potential attacks. However, efficient methods are needed to reduce the trade-off between communication and computation complexities in resource-constrained wireless device communication. In this paper, we propose a lightweight fault-tolerant secure data communication framework that consists of Elliptic Curve Diffie-Hellman (ECDH) secure communication scheme, Elliptic Curve Cryptography (ECC) based secure communication scheme and Elliptic Curve Integrated Encryption Scheme (ECIES) based authentication scheme for wireless sensor network communication using Message Passing Interface (MPI) parallel program platform. Further, we have implemented the proposed framework for a single sink node (scenario-1) and all sink nodes (scenario-2) scenarios with parallel threads using Linux Pthreads to improve the total execution time. It is observed that the overall execution time performance of ECC is better in scenario-2 whereas the performance of ECDH is better in scenario-1 when the number of sensors is greater than 200. It is also observed that enabling Linux Pthreads in ECC implementation guarantees the parallel execution of decryption process and the reduction in the overall execution time in both scenarios. The results demonstrate the superiority of the proposed framework in terms of execution time and memory use over simulated wireless network environments, making the proposed framework suitable for fault-tolerant wireless sensor communication applications.

INDEX TERMS Wireless sensor networks, fault-tolerant communication, elliptic curve cryptography, lightweight authentication, the Internet of Things, key management, data aggregation.

I. INTRODUCTION

The advent of large-scale production of Internet-of-Things (IoT) devices made the wireless communication easy and quickly accessible. The Wireless Sensor Network (WSN) is an inter-connected network of a large number of dedicated sensor devices which collect, transfer and analyse the useful information. The WSN is also used to detect the environment changes, transferring the collected data in a wireless communication medium to a fusion center through other intermediate sensors or nodes for further processing. The application domains of WSN are multi-fold including Healthcare, Agriculture, weather forecasting, surveillance of safety critical systems, disaster management, smart applications, vehicular technology, unmanned aerial vehicles, real-time systems, military etc. However, the accuracy of the post-processing result of collected data depends on the following persistently raising issues i) inaccurate data fusion ii) faulty information iii) insecure communication iv) unexpected exceptions like failure of a link, energy depletion, radio interference, environmental calamities and synchronization mismatch. As a consequence of the security vulnerabilities present in WSN [1] due to various reasons, an adversary can easily execute different attack strategies that disrupt either the communication or tamper the information. Therefore, the development of secure and fault-tolerant wireless sensor communication in redundant sensor systems have gained enormous attention in recent years.

To mitigate prominent faults in WSN [2], researchers have put their efforts through novel procedure-based [3], [4], [5],

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar.

[6] and prediction-based [7], [8], [9] solutions. Despite of few efforts [10], [11], achieving the best fault-tolerance with accurate data fusion was a nightmare. To bridge this gap, the first interval-based practical solution [12] was proposed to encourage the researchers to deep dive into interval-based data fusion. The proposed solution could be useful in many practical applications including the safety of cyber-physical systems, fault-tolerant scheduling in real-time operating systems, robot convergence, fault tolerant high performance computing, ensembling in artificial intelligence, software or hardware reliability etc. But, lack of security in fault-tolerant sensor fusion systems lead to different types of potential attacks such as man-in-the middle attack, Sybil attack, denial-of-service attack etc. To overcome such attacks, it is essential to look into the security of the data at rest and/or during transmission in fault-tolerant sensor fusion systems.

The motivation behind this work is that the security of fault-tolerant sensor fusion systems can be implemented broadly in two ways i.e., simulation-based implementation and actual implementation. There are three types of cryptographic methods to secure the underlying system i) asymmetric encryption ii) symmetric encryption iii) hybrid encryption which uses both asymmetric and symmetric. Out of existing cryptographic methods, it is found that Elliptic Curve Cryptography (ECC) based security solution [13] stands best for resource constrained devices in terms of computation and bandwidth use. But, not much work has been proposed in securing fault-tolerant WSN communication except for few ECC-based security solutions [14].

In this work, we have focused on actual implementation and propose a ECC-based security framework for fault-tolerant sensor fusion systems with the following salient features.

- The proposed security framework involves two types of security solutions. First, it provides Elliptic Curve Diffie-Hellman (ECDH) plus Advanced Encryption Standard (AES) based security solution. Second, it provides the complete ECC-based security solution.
- The proposed hybrid security framework is resistant to potential attacks such as man-in-the-middle attack, Sybil attack etc.
- The proposed security framework that can be used for both wired and wireless interval-based fault-tolerant sensor fusion systems.

The paper is organized as follows. Section II refers to the current state-of-art work in fault-tolerant wireless data communication. Section III covers the interval-based fault-tolerant sensor fusion in wireless sensor network. Section IV covers the proposed hybrid security framework for fault-tolerant wireless data communication. Section V covers the implementation details of the proposed framework. Section VI describes the obtained results of the proposed framework in terms of execution time. Section VII concludes with the conclusion and future scope.

II. RELATED WORK

The fault-tolerance in interval-based redundant sensor systems could be effectively achieved through any one of the following: scalar-based approximate consensus technique [15], interval-based Brooks-Iyengar Algorithm (BIA) [12], vector-based Byzantine vector consensus technique [10] and multi-dimensional agreement technique [11].

Many security protocols and fault tolerant schemes were proposed in recent years. Security means ensuring confidentiality, integrity, availability and authenticity of the data. Providing security for resource constrained WSN is totally different from providing security for the resource-rich environments. Therefore, lot of research work took place on providing the security for WSN taking the less computation power, limited resources of sensors into consideration. at starting symmetric key cryptography was mainly used to secure the confidentiality of the data. The study [16] showed the possibility of using the public key cryptography (PKC) in selecting the pair-wise keys (shared secret key between two nodes). Based on [16], new security scheme was proposed for establishing the symmetric pair-wise keys using asymmetric key cryptography (ECC) between two nodes [17]. The reasons for forming the key pairs for every two nodes are for ensuring confidentiality and authenticity. Authentication is a major aspect of security. A two factor authentication scheme is proposed in [18] which provides safety over impersonation attacks, offline password guessing attacks. An authentication scheme based on elliptical curve theory for health care management is proposed in [19] and [20] which take three factors namely user credentials, smart card, biometrics into consideration for authentication purposes. An idea which combines the blockchain and authentication to find a way to protect the data from being corrupted during worm attack in sensor network is proposed in [21] and proves helpful in authenticating genuine user even when WSN is under worm attack.

The sensors present in WSN generally have less computation power and are resource constrained saving the energy used while generating these pair-wise keys helps in increasing the lifetime of the WSN. An efficient combined system divides the entire WSN into different layers and uses Kerberos protocol in those layers which are very near to the base station and Elliptic Curve Menezes-Qu-Vanstone (ECMQV) in those layers which are somewhat far away from the base station for generating the shared key between the nodes helped in increasing energy efficiency in WSN. The key generation protocol based on PKC which diffuses part of the secret key on the sensor nodes while the other part would be kept inside the nodes of WSN proves helpful in saving the energy of the sensors. For all the nodes present in the network, sensors need to store the secret keys. This demands high storage facilities which are not possible for a resource constrained sensor. Therefore, achieving storage efficiency is important for WSN. This is where key distribution schemes come into play. The key distribution scheme which takes topology of

TABLE 1. The feature-wise comparison of existing schemes.

Scheme	Encryption		Authentication		Real Implementation	Facult-tolerant Data Generation	Self Key Distribution
	symmetric	asymmetric	single	multiple			
Nadir et al. [17]	✔	✔	✔	✘	✔	✘	✔
Hu et al. [18]	✔	✔	✘	✔	–	✘	✘
Dai and Xu [19]	✘	✔	✘	✔	–	✘	✘
Chen et al. [21]	✘	✘	✘	✘	✘	✘	✔
Havashemi and Barati [22]	✔	✔	✔	✘	✘	✘	✔
Zhang et al. [23]	✘	✔	✘	✘	–	✘	✘
Ullah et al. [24]	✔	✘	✘	✘	✔	✘	✔
Afsar et al. [25]	✘	✘	✘	✘	✘	✘	✔
Bashaa et al. [26]	✘	✔	✘	✘	✘	✘	✔
Proposed Scheme	✔	✔	✔	✘	✔	✔	✔

the WSN into account while generating the shared secret key between nodes and the results show that the number of shared keys stored in a node is decreased without affecting the other security parameters. The distributed architecture of wireless sensor networks need some robustness. This robustness is given by fault tolerant schemes. A cluster based fault tolerant protocol which divides the network into many clusters and makes the cluster head detect and resolve the fault issues without directly contacting the base station.

The features of recent contributions done in security and fault-tolerant wireless sensor communication are tabulated in TABLE 1. But, comparatively less work is done in incorporating both the features (fault-tolerance and security) in a single sensor network. Our work presented in this paper includes both the features in a sensor network and compares various security measures which are implemented on top of the fault-tolerant sensor networks. The security measures used are compared using different metrics to understand and observe the difference between them.

A. KEY MANAGEMENT

The secret link key-based scheme has been proposed by Deng and Han [27]. In this, the secret key is generated using the pre-distribution of the set of keys into each node before deployment. The co-operative secret delivery technique has been proposed to transfer the secret key generated by the source node to the insecure neighbor (the nodes with no common key with the source node). It takes the help of the bridge nodes that have at least one common key with the source node and an already predefined number of common keys with the any of the insecure neighbors. They compared the secret disclosure probabilities of different schemes and showed that the proposed scheme provides low secret disclosure probability compared to other schemes.

Pietro et al. [28] introduced two protocols namely direct protocol and co-operative protocol to secure the communication between two nodes. The co-operative protocol is adaptive, and its properties can be changed during the lifetime of the WSN. The probabilistic method is used to prove the adaptiveness of their scheme. In the direct protocol, the sensor node is pre-distributed with some set of keys selected from the key pool. These keys are indexed, indexes are stored in the sensor node, and a seed is assigned when passed to a generator which generates the indexes of the keys belonging to the certain node. The node generates the secret key using indexes of the keys. But, when there is no common key between two nodes, the shared key between those two nodes is generated using the co-operative protocol. The proposed scheme provides automatic authentication without any overhead and shows the corruption probabilities for different number of corrupted nodes for both direct and co-operative protocols.

Li et al. [29] introduced an energy efficient and high accuracy scheme to guarantee accuracy, privacy and reduced communication overhead. In this, an aggregation tree is constructed from the network topology. The leaf nodes sense the data; the intermediate nodes will aggregate the results sent by their child nodes, combines their own sensed value and transfers it to the base station. The shared keys established between the nodes are used to encrypt the data that is being transferred. To preserve the privacy of the data, the leaf nodes identify a set of nodes within h hops and divide its sensed data into m pieces where m is the number of nodes present in the set of nodes including itself. These slices are then encrypted and sent to the respective nodes. A node waits for a certain interval of time after sending the data to other nodes. The other nodes receive the information, decrypt the data, aggregate it with other received data, encrypts the aggregated data and send it to their parent node. The parent node in turn sends the final encrypted aggregated information to base station. The proposed scheme shows better results when compared with SMART model in terms of energy consumption, accuracy and communication overhead.

Rahman and Sampalli [30] presented an improved scheme over Blom scheme for establishing the keys between pairs of nodes. The proposed scheme adds the functionalities like key revocation, node addition to the Blom scheme. The node ids are used to generate the secret keys between the nodes. The nodes are pre-distributed with the respective columns of the private keys before deployment. When two nodes want to communicate, they share their ids and calculate the column of the respective nodes in the public matrix present in the base station. Then the key is calculated by multiplying the private row matrix and the public column matrix. Secure communication is achieved by encrypting the information

using the shared secret key. The process of key revocation starts by identifying the nodes that are compromised. The new secret key matrix is calculated and the message carrying the private row matrices of only those nodes that are not compromised are broadcasted across the network. After receiving the broadcasted message, the nodes decrypt their own row matrix and update the already present row matrix with the decrypted row matrix. The compromised nodes will not get any updates and hence they cannot establish connection with any other nodes thus they are excluded from the network.

A key management schemes presented in [22] and [31] decrease power consumption, increases efficiency, flexibility and scalability. The proposed scheme [22] follows a hierarchical model and uses three different types of keys namely area key, communication key and base station key. It uses asymmetric key encryption for encrypting the data with the base station key. The proposed method is simulated in NS2 and compared with HISCOM and MGHS. The results show that the proposed method reduces energy consumption and memory usage improves the flexibility and network lifetime. It is shown that the proposed method is resistant to node compromise attacks and replay attacks. In addition, the proposed method guarantees confidentiality, integrity, and authenticity of the data.

B. SECURE DATA AGGREGATION

Zhong et al. [32] presented a scheme that overcomes the disadvantages of homomorphic encryption like malleability, unauthorized aggregation, and limited aggregation functions by combining homomorphic encryption with a signature scheme. The proposed scheme reduces the transmission costs by doing in-network false data filtering. In this, the base station can identify the origin of the data it receives. Upon receiving the data, the base station decrypts it and performs the aggregation function. This scheme is evaluated against communication overhead, computational overhead, energy consumption and delay and found with satisfying results.

Zhang et al. [23] devised a method for tree-based homogeneous sensor network. The proposed scheme uses homomorphic encryption and executed in five phases. Since the proposed method follows the end-to-end encryption method, it reduces energy consumption by exempting the intermediate nodes to encrypt and decrypt the data. It supports multi-functional aggregation and helps in enhancing privacy and confidentiality of data as the intermediate nodes cannot decrypt the data. This scheme prevents eavesdropping and traffic analysis by adversaries. One of the drawbacks of the proposed scheme is that the sensors present in the upper layers of the tree have to work more compared to the sensors at the lower level. Hence, the upper layer sensor's lifetime is less compared to lower layer sensors. Therefore, this makes it not suitable for large-scale WSNs. In addition, the proposed scheme does not provide any mechanism for removing duplicate data which leads to an increase in the communication overhead and energy consumption.

TABLE 2. The summary of notations.

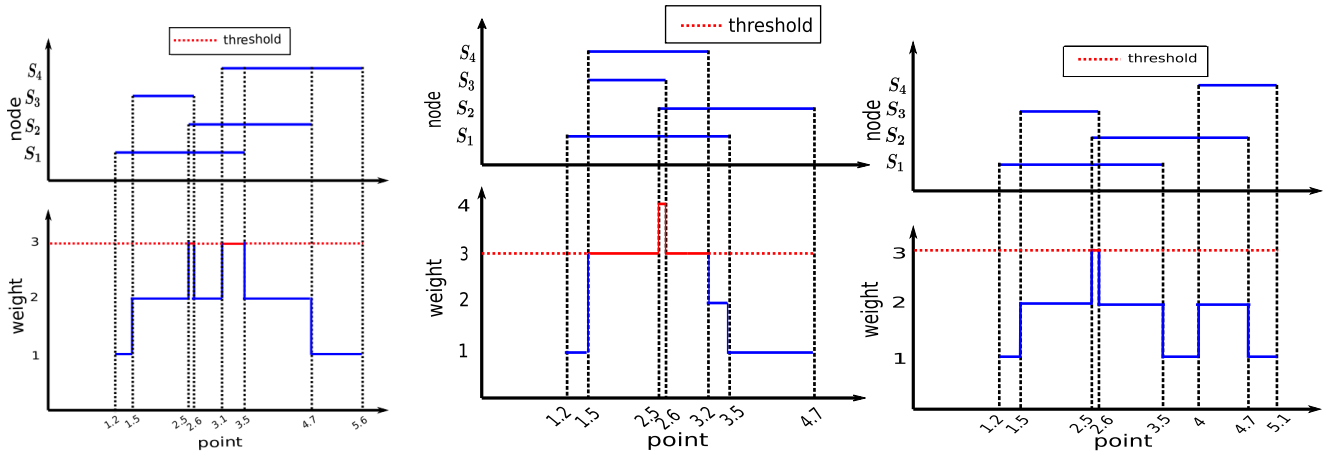
Symbol	Interpretation
S_i	i^{th} sensor node
S_{ij}	pair of sensors S_i and S_j
N	Number of sensor nodes
$(\mathcal{PU}_i, \mathcal{PR}_i)$	(public key, private key) pair of i^{th} node
$\mathcal{SK}_{ij}$	Shared secret key between ith and i^{th} and j^{th} node
IP_i	Plaintext of i^{th} node
C_{ij}	Ciphertext between i^{th} and j^{th} node
EP	Elliptic curve set
A	Symmetric key algorithm set
PE_i	Point estimation
τ	Number of faulty sensors
$EP(a, b)$	ECC curve set
E_{ij}, E_{ij}^{-1}	Encryption and decryption functions respectively

Ullah et al. [24] have designed a scheme for homogeneous sensor types using cluster-based hierarchical network topology. In this, the data is collected by the sensors, encrypted and hashed with a timestamp. The aggregator validates the hash value, encrypts the collected data, re-hash it (including the timestamp) and sends the compressed version to the next aggregator. Once the packet reaches the sink, the fog server validates the data and decrypts it. The proposed scheme guarantees data confidentiality, integrity, data freshness and privacy. It follows the end-to-end encryption method. Since it uses cluster-based network topology, the proposed scheme is highly scalable. Even though the proposed scheme has countermeasures for eavesdropping, traffic analysis, Sybil attack, replay attacks and flooding attacks, it does not provide any mechanism to detect data redundancy and fails to balance the network's energy consumption. As a result, the nodes nearer to the fog server have less lifetime than other nodes.

Boubiche et al. [33] have also designed a secure scheme for homogeneous WSNs. It uses cluster based hierarchical network topology. The proposed method uses a watermark technique to validate and secure the data. In this, each sensor calculates the watermark and fills the first few bits of the packet with the watermark and the remaining space with the collected data. The proposed scheme provides data integrity and is resistant to Sybil attack, packet alteration and injection attacks. Since every node must calculate the watermark and validate it, the proposed scheme suffers from high energy consumption and delay. In addition, the confidentiality of the data is also arguable here because the data transmitted is only secure until the adversary does not know about the watermark.

III. INTERVAL-BASED BROOKS-IYENGAR FAULT-TOLERANT SENSOR FUSION ALGORITHM

The summary of notations used in this paper is recorded in TABLE 2. The interval-based Brooks-Iyengar fault-tolerant sensor fusion algorithm runs on each sensor S_i, $1 \leq i \leq N$, in two phases: sensor data generation and output point estimation as follows.



(a) at S_1 (b) at S_2 (c) at S_3

FIGURE 1. The weighted region diagram at various nodes.

Phase-1: [Sensor data generation (SensorDataGen())]

for j=1 to t **begin**

$[l_j, h_j] \leftarrow 2 \cdot \sin(\frac{j\pi}{2})$+Rand();

end for

Sort($\{[l_1, h_1], \cdots, [l_t, h_t]\}$)

$\text{IP}_i \leftarrow$Assignweight($\{([l_1, h_1], w_1), \cdots, ([l_t, h_t], w_t)\}$)

▷ IP_i=interval-based sensor data

▷ t=interval size

▷ w_j=no. of intersecting intervals in the range $[l_j, h_j]$

Phase-2: [Output point estimation (PointEstimation())]

Construct Weighted Region Diagram (WRD) of IP_i

for j=1 to t **begin**

Select $[l_j, h_j]$ if $w_j > (N - \tau)$

end for

Select RI_i=$\{([l_1, h_1], w_1), \cdots, ([l_L, h_L], w_L)\}$

Calculate PE_i=$\sum_{j=1}^{L}(w_j \cdot (l_j + h_j))/2 \cdot \sum_{j=1}^{L} w_j$

▷ PE_i=point estimation

▷ N=no. of sensors, τ=no. of faulty sensors

▷ L=remaining interval size

To understand further, consider N= 4, τ=1. Let the interval data generated by the sensors S_1, S_2, S_3, S_4 are [1.2,3.5], [2.5,4.7], [1.5,2.6], [3.1,5.6] respectively. Assume S_4 is faulty. Each sensor executes the fault-tolerant sensor fusion algorithm by constructing WRD as shown in FIGURE 1 (i.e., FIGURE 1a for S_1, FIGURE 1b for S_2, FIGURE 1c for S_3). Each sensor then calculates intersection points, remaining intervals and output point estimation as shown in equation (1), at the bottom of the page.

IV. PROPOSED HYBRID SECURITY FRAMEWORK

The proposed security framework operates in three phases: sensor data generation, secure communication, sensor fusion. The sensor data generation and sensor fusion phases are exactly the same as explained in Section III. To achieve secure communication between sensors, the proposed framework broadly provides an ECC-based cryptographic solution for

$$
\begin{aligned}
&\text{Intersection points at } S_1 \text{ are } (\text{IP}_1) = \{([1.5, 2.5], 2), ([2.5, 2.6], 3), ([2.6, 3.1], 2), ([3.1, 3.5], 3), ([3.5, 4.7], 2)\}\\
&\text{Remaining intervals at } S_1 \text{ are } (\text{RI}_1) = \{([2.5, 2.6], 3)([3.1, 3.5], 3)\}\\
&\text{Output point estimate at } S_1 \text{ is } (\text{PE}_1) = \frac{3 \cdot (\frac{2.5+2.6}{2}) + 3 \cdot (\frac{3.1+3.5}{2})}{6} = 2.925\\
&\text{Intersection points at } S_2 \text{ are } (\text{IP}_2) = \{([3.2, 3.5], 2), ([1.5, 2.5], 3), ([2.6, 3.2], 3), ([2.5, 2.6], 4)\}\\
&\text{Remaining intervals at } S_2 \text{ are } (\text{RI}_2) = \{([1.5, 2.5], 3), ([2.6, 3.2], 3), ([2.5, 2.6], 4)\}\\
&\text{Output point estimate at } S_2 \text{ is } (\text{PE}_2) = \frac{3 \cdot (\frac{1.5+2.5}{2}) + 3 \cdot (\frac{2.6+3.2}{2}) + 4 \cdot (\frac{2.5+2.6}{2})}{10} = 2.49\\
&\text{Intersection points at } S_3 \text{ are } (\text{IP}_3) = \{([1.5, 2.5], 2), ([2.5, 2.6], 3), ([2.6, 3.5], 2), ([4, 4.7], 2)\}\\
&\text{Remaining intervals at } S_3 \text{ are } (\text{RI}_3) = \{([2.5, 2.6], 3)\}\\
&\text{Output point estimate at } S_3 \text{ is } (\text{PE}_3) = \frac{3 \cdot (\frac{2.5+2.6}{2})}{3} = 2.55
\end{aligned} \tag{1}
$$

two prominent sensor network scenarios. In the first scenario (scenario-1), each sensor is considered as sink and therefore, the fusion process will be carried out in every sensor. In the second scenario (scenario-2), only one of the sensors is considered as sink and therefore, fusion process will be carried out only in that sensor.

A. ECDH-BASED SECURE COMMUNICATION SCHEME

The proposed scheme is a hybrid two-party secure communication system where ECC-based (public key, private key) pair is used to generate a shared-secret and symmetric encryption (such as AES or DES or TripleDES) is used to encrypt the data. It is a three tuple ($\mathcal{KG}$, $\mathcal{E}$, $\mathcal{D}$) scheme where $\mathcal{KG}$ is key generation, $\mathcal{E}$ is encryption and $\mathcal{D}$ is decryption as described below.

KEY GENERATION ($\mathcal{KG}$):

1) Sensor S_i, $1 \leq i \leq N$, chooses an ECC curve from $EP(a, b)$={secp256k1,secp192k1,secp521r1, prime192v3, prime239v3} and generates ECC-based (public key, private key) pair ($\mathcal{PU}_i$, $\mathcal{PR}_i$) using equation (2).

$$\begin{aligned} \mathcal{PR}_i &\xleftarrow{R} \mathbb{Z} \\ \mathcal{PU}_i &= \mathcal{PR}_i \cdot G \end{aligned} \tag{2}$$

where $\mathbb{Z}$=integer set, G=ECC generator.

2) Using ECDH method, pair of sensors S_i and S_j (in short S_{ij}) generates a shared-secret $\mathcal{SK}_{ij}$ using equation (3).

$$\mathcal{SK}_{ij} = \mathcal{PR}_i \cdot \mathcal{PU}_j \tag{3}$$

where $\mathcal{SK}_{ij}$=shared-secret between S_i and S_j.

3) Output shared-secret $\mathcal{SK}_{ij}$.

ENCRYPTION ($\mathcal{E}$):

1) Using shared-secret $\mathcal{SK}_{ij}$ and symmetric key algorithm $A \in \{AES, DES, TrippleDES\}$, sensor S_i encrypts the interval-based sensor data IP_i using equation (4) and sends to S_j.

$$C_{ij} = E_{ij}(IP_i, \mathcal{SK}_{ij}, A) \tag{4}$$

where E_{ij}=symmetric key encryption.

2) Output the ciphertext C_{ij}.

DECRYPTION ($\mathcal{D}$):

1) Using the ciphertext C_{ij}, shared-secret $\mathcal{SK}_{ij}$, and the symmetric key algorithm from A, sensor S_j decrypts the ciphertext using equation (5).

$$IP_i = E_{ij}^{-1}(C_{ij}, \mathcal{SK}_{ij}, A) \tag{5}$$

where E_{ij}^{-1}=symmetric key decryption.

2) Output the decrypted data IP_i.

Since every sensor should carry out fusion in case of scenario-1, each sensor generates $(N-1)$ ECDH-based shared-secrets where N is the total number of sensors in the network. It is intuitive that each sensor should run $(N-1)$ instances of encryptions followed by $(N-1)$ instances of decryptions (as shown in FIGURE 2a). But, in case of scenario-2, each sensor except the sink sensor generates only one ECDH-based shared-secret and should run only one encryption instance. The sink sensor generates $(N-1)$ ECDH-based shared-secrets, runs $(N-1)$ instances of decryptions (as shown in FIGURE 2b).

B. ECC-BASED SECURE COMMUNICATION SCHEME

The proposed ECC-based secure communication is uses ECC-based asymmetric (public key, private key) pair to encrypt the data. It is also a three tuple ($\mathcal{KG}$, $\mathcal{E}$, $\mathcal{D}$) scheme where $\mathcal{KG}$ is key generation, $\mathcal{E}$ is encryption and $\mathcal{D}$ is decryption as described below.

KEY GENERATION ($\mathcal{KG}$):

1) Sensor S_i, $1 \leq i \leq N$, chooses an ECC curve $E_p(a, b)$ and generates ECC-based (public key, private key) pair ($\mathcal{PU}_i$, $\mathcal{PR}_i$) using equation (2).
2) Output the pair ($\mathcal{PU}_i$, $\mathcal{PR}_i$).

ENCRYPTION ($\mathcal{E}$):

1) Using public key ($\mathcal{PU}_j$) of the destination sensor S_j, sensor S_i encrypts the interval-based sensor data IP_i and sends to S_j using equation (6).

$$C_{ij} = E_{ij}(IP_i, \mathcal{PU}_j) \tag{6}$$

where E_{ij}=asymmetric key encryption.

2) Output the ciphertext C_{ij}.

DECRYPTION ($\mathcal{D}$):

1) Using the ciphertet C_{ij}, private key $\mathcal{PR}_j$, sensor S_j decrypts the ciphertext using equation (7).

$$IP_i = E_{ij}^{-1}(C_{ij}, \mathcal{PR}_j) \tag{7}$$

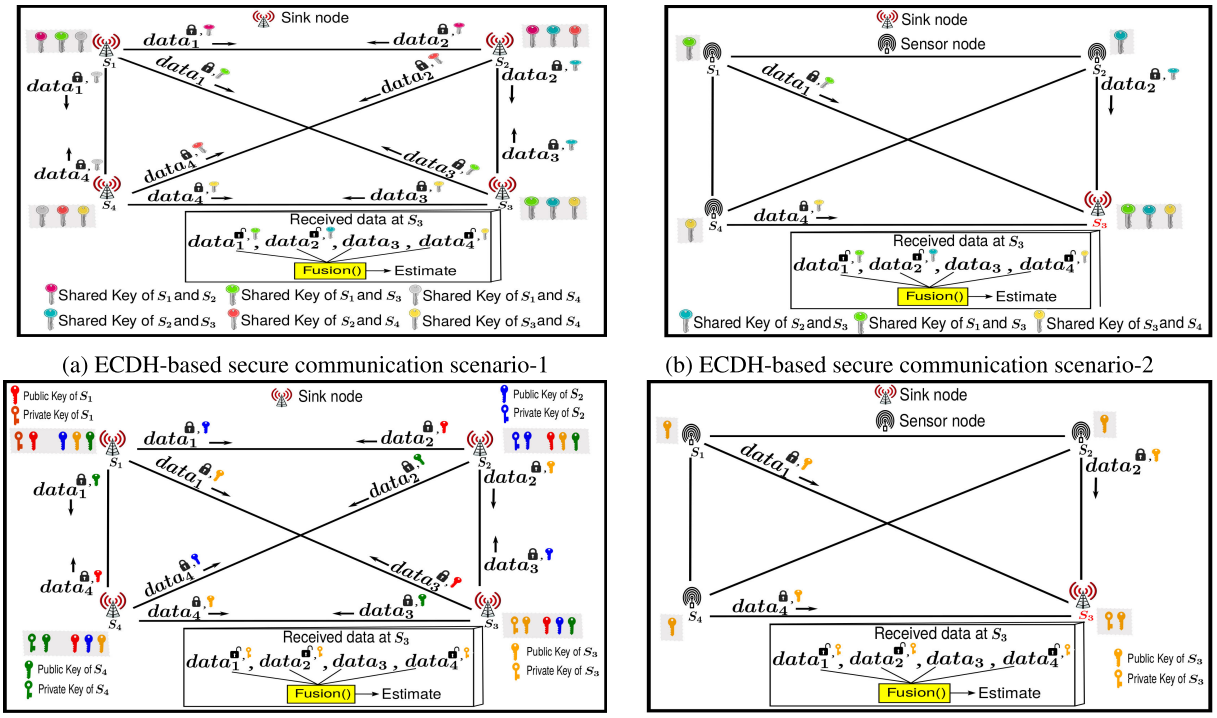
where E_{ij}^{-1}=asymmetric key decryption.

2) Output the decrypted data IP_i.

Since every sensor should carry out fusion in case of scenario-1, each sensor generates $(N-1)$ ECDH-based shared-secrets where N is the total number of sensors in the network. It is intuitive that each sensor should run $(N-1)$ instances of encryptions followed by $(N-1)$ instances of decryptions (as shown in FIGURE 2c). But, in case of scenario-2, each sensor except the sink sensor generates only one ECDH-based shared-secret and should run only one encryption instance. The sink sensor generates $(N-1)$ ECDH-based shared-secrets, runs $(N-1)$ instances of decryptions (as shown in FIGURE 2d).

C. LIGHT-WEIGHT AUTHENTICATION SCHEME

We have used the light-weight Elliptic Curve Integrated Encryption Scheme (ECIES) method proposed in [34] for fault-tolerant, secure and authenticated WSN communication on top of Brooks-Iyengar algorithm described in Section III. The ECIES method involves a hybrid combination of Elliptic Curve Digital Signature Algorithm (ECDSA), Secure Hash Algorithm (SHA) and CLAKE2b for generating and verifying the cryptograms. The process of generating key pairs,



(a) ECDH-based secure communication scenario-1

(b) ECDH-based secure communication scenario-2

(c) ECC-based secure communication scenario-1

(d) ECC-based secure communication scenario-2

FIGURE 2. ECC-based and ECDH-based secure communication scenarios.

generating cryptogram and verifying cryptogram in ECIES method is as follows.

1) KEY GENERATION ($\mathcal{KG}$)

It is a two-way communication between two sensor nodes S_i and S_j in which the node S_i initiates the communication by choosing random parameters such as groups G_1, G_2, identifiers Id_1, Id_2, primary key $\mathcal{K}_m$. The node S_i generates two pairs of (public, private) keys using equation (8) and sends its public keys to S_j in the form of key string KS (Refer equation (9) for key string formation). Upon receiving key string, the node S_j also generates two pairs of (public, private) keys using equation (8) and sends its public keys to S_i in the form of key string.

$$((\mathcal{PU}_1, \mathcal{PR}_1), (\mathcal{PU}_2, \mathcal{PR}_2)) \xleftarrow{R} GenerateKeyPair(\mathcal{P}) \quad (8)$$

$$KS = \{\mathcal{PU}_1 || \mathcal{PU}_2 || ((Id_1 + Id_2) \oplus \mathcal{K}_m)\} \quad (9)$$

2) CRYPTOGRAM GENERATION ($\mathcal{CG}$)

Using the plaintext IP_i, one of the private keys $\mathcal{PR}_{i,1}$, one of the received public keys $\mathcal{PU}_{j,2}$ of node S_j, and a shared secret $\mathcal{SK}_{ij}$ of both i-th and j-th nodes, the cryptogram is generated at the sending node S_i using equation (10) and then sent to the receiving node S_j for verification.

$$\mathcal{C}_{ij} = GenerateCryptogram(IP_i, \mathcal{PR}_{i,1}, \mathcal{PU}_{j,2}, \mathcal{SK}_{ij}) \quad (10)$$

where $\mathcal{C}_{ij}$ is the cryptogram.

CRYPTOGRAM VERIFICATION ($\mathcal{CV}$): Using the cryptogram $\mathcal{C}_{ij}$, shared secret $\mathcal{SK}_{ij}$, one of the public keys $\mathcal{PU}_{j,2}$, group G_1, one of the public keys $\mathcal{PU}_{i,1}$ of sending node S_i,the receiving node S_j verifies the cryptogram and extract the plaintext IP$_i$ using equation (11).

$$IP_i = VerifyCryptogram(\mathcal{C}_{ij}, \mathcal{SK}_{ij}, \mathcal{PU}_{j,2}, G_1, \mathcal{PU}_{i,1}) \quad (11)$$

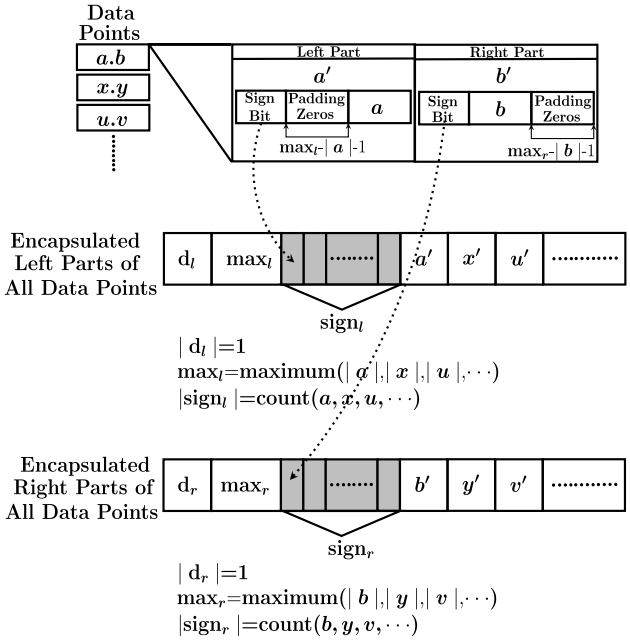


FIGURE 3. Data point representation and encapsulation before encryption in ECC implementation.

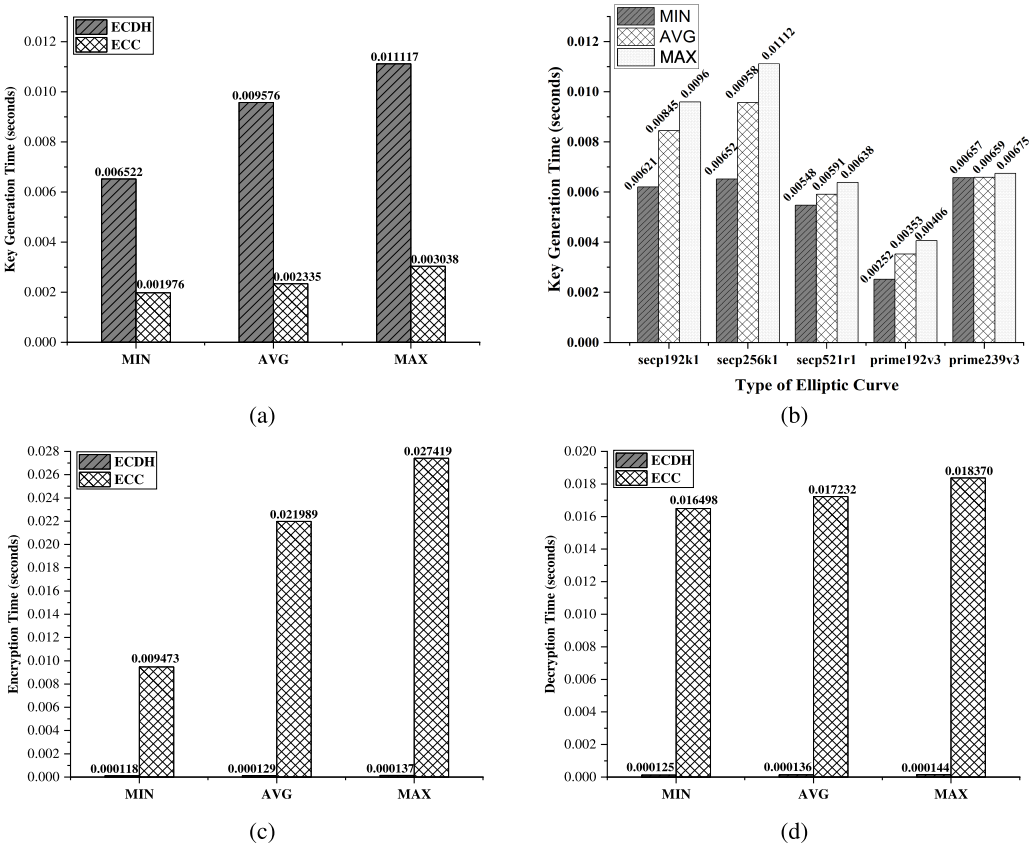


FIGURE 4. Performance of proposed schemes in scenario-1 with 4 nodes.

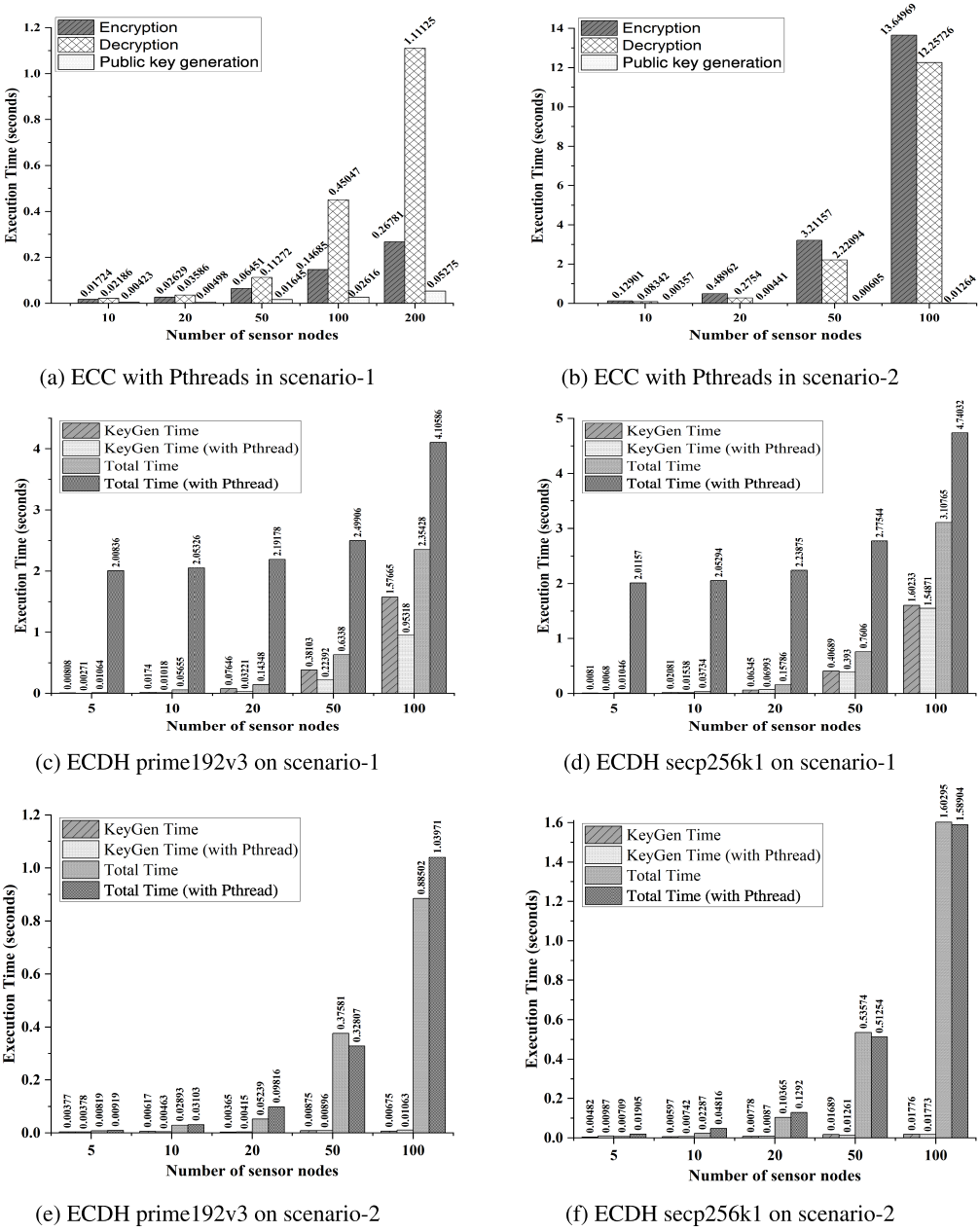
where IP_i is the plaintext.

The proposed scheme uses ECIES method to achieve authentication, integrity and confidentiality in fault-tolerant WSN communication. The description of the proposed scheme is as follows.

- Each sender node S_i chooses a parameter list $\mathcal{P}$ and generates required key pairs $(\mathcal{PU}_{i,1}, \mathcal{PR}_{i,1})$, $(\mathcal{PU}_{i,2}, \mathcal{PR}_{i,2})$ and a key string KS_i using key generation algorithm $\mathcal{KG}$. The key string is then sent to the receiver node or sink node S_j.
- After receiving the key string from all the sender nodes, the sink node decodes each key string and extracts the public key components $(\mathcal{PU}_{i,1}, \mathcal{PU}_{i,2})$ and group ids $(Id_{i,1}, Id_{i,2})$ of each sender node.
- Using the parameter list $\mathcal{P}$, sink node generates two pairs of keys $(\mathcal{PU}_{j,1}, \mathcal{PR}_{j,1})$, $(\mathcal{PU}_{j,2}, \mathcal{PR}_{j,2})$ and a key string KS_j using key generation algorithm $\mathcal{KG}$. Each key string is then sent to the respective sender node.
- Each sensor node decodes the received key string and extracts the public keys $(\mathcal{PU}_{j,1}, \mathcal{PU}_{j,2})$ of the sink node.
- Each sender node S_i generates the sensor data IP_i using interval-based Brooks-Iyengar fault-tolerant sensor fusion algorithm (Refer SensorDataGen() described in Section III). Using sensor data IP_i, one of its private keys $\mathcal{PR}_{i,1}$, one of the public keys $\mathcal{PU}_{j,2}$ of sink node, shared secret $\mathcal{SK}_{ij}$, each sender node generates the cryptogram $\mathcal{C}_{ij}$ using cryptogram generation algorithm $\mathcal{CG}$. The $\mathcal{CG}$ algorithm achieves encryption using AES algorithm, achieves hashing using CBLAKE2b and SHA techniques and signs the plaintext using ECDSA technique. Finally, the encrypted plaintext, hash of the plaintext and the parameters required for verifying the signature of the sensor node are bundled together as a cryptogram and sent to the sink node.
- Sink node verifies all the received cryptograms using cryptogram verification algorithm $\mathcal{CV}$. If the verification is successful, the plaintext is extracted and further processed by the sink node otherwise the cryptogram is simply rejected.

V. IMPLEMENTATION DETAILS

The setup consists of the following system configurations: Ubuntu 18.04.5 LTS OS, 8 GiB RAM, Intel Core i5-8265U CPU with 1.60GHz×8, 500GB hard disk. The proposed framework has been implemented using a distributed computing environment standard called Message Passing Interface (MPI). The MPI provides a distributed computing environment with a separate memory and computing capabilities to each processing element which helps to emulate a sensor behaviour on the processing element. The MPI provides



(a) ECC with Pthreads in scenario-1

(b) ECC with Pthreads in scenario-2

(c) ECDH prime192v3 on scenario-1

(d) ECDH secp256k1 on scenario-1

(e) ECDH prime192v3 on scenario-2

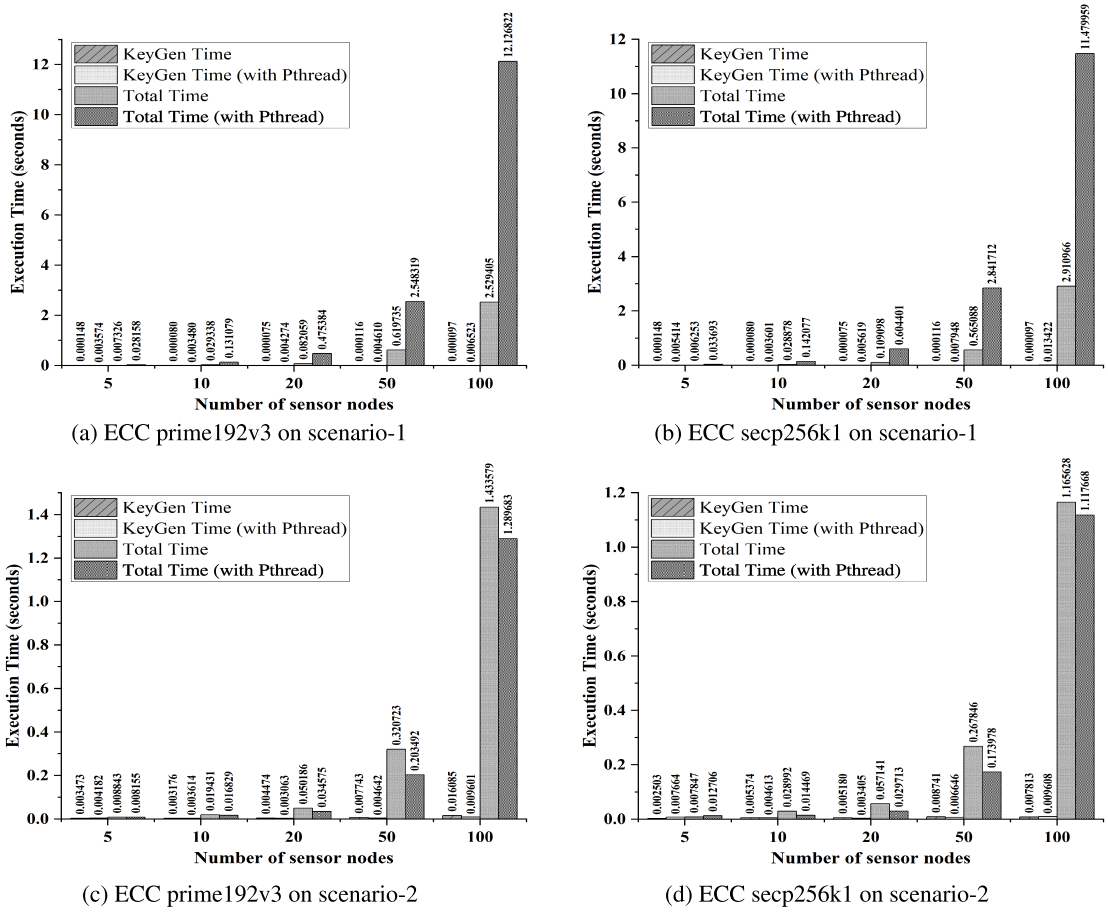
(f) ECDH secp256k1 on scenario-2

FIGURE 5. ECC and ECDH performance on both scenarios with and without Pthreads.

MPI_Send() and MPI_Recv() APIs to establish one-to-one communication between any two sensors and it provides MPI_Scatter() API to establish one-to-many communication from one sensor to other sensors. The proposed scheme uses mpz_class from GMP library to generate large integer numbers. All the operations in ECDH are carried out as string operations. However, all the operations in ECC are integer operations. Therefore, the floating point input data points must be converted into integers before encryption as shown in FIGURE 3. Consider three input data points 123.45, -10.687, 4847.3 where a=123, b=45,x=-10,y=687, u=4847, v=3. The components of the encapsulated left part are calculated as d_l=1, $\max_l$=4, sign_l={0,1,0}, a'=0123, x'=1010, u'=4847. Similarly, the components of the encapsulated right part are calculated as d_r=1, $\max_r$=4, sign_r={0,1,0}, b'=4500, y'=1687, v'=3000.

A. PTHREAD-ENABLED IMPLEMENTATION

We have extended the proposed schemes of Section IV with Linux Pthreads to speed up the overall process. In Pthread-enabled ECC implementation, both the encryption and decryption phases of the proposed ECC scheme of Section IV-B have been implemented and tested for execution



(a) ECC prime192v3 on scenario-1

(b) ECC secp256k1 on scenario-1

(c) ECC prime192v3 on scenario-2

(d) ECC secp256k1 on scenario-2

FIGURE 6. ECC performance on both scenarios with and without Pthreads.

time performance using linux Pthreads in both scenarios. The results confirm that incorporating Pthread has decreased the total execution time. In Pthread-enabled authentication implementation, only the verification phase of the proposed ECIES-based authentication scheme of Section IV-C has been implemented and tested for cryptogram generation time, cryptogram verification time, total execution time performance using linux Pthreads in single sink node scenario.

VI. RESULTS AND DISCUSSION

The proposed framework is initially implemented with four sensor nodes. The secp256k1 elliptic curve is used in ECDH to generate the 256-bit public and private key pairs. The CBC mode of AES with 256-bit key is used for encrypting the messages with the secret key generated from public and private key pair. The elliptic curve $y^2 = x^3\text{-}3x\text{-}20925$ is used in ECC with 77-bit private key and the corresponding public key is used to encrypt the data. FIGURE 4a shows the comparison of key generation time in ECC and ECDH-based communication. It is observed that ECDH is showing worst performance over ECC due to the additional shared-secret key generation. FIGURE 4b shows the comparison of key generation time among various ECC curves. Among secp192k1, secp256k1, secp521r1, prime193v3, prime239v3 curves, the prime192v3 curve is taking less time for key generation whereas secp256k1 is taking more time. FIGURE 4c and FIGURE 4d compare and show the encryption and decryption time using ECDH and ECC for four sensors. It is clear from the results that ECDH is taking very less time compared to the ECC for encrypting/decrypting the data. This large performance gap is due to the repeated point addition in ECC encryption/decryption process.

The proposed framework is also tested against different sensor nodes setting up to 200 and the performance is recorded in TABLE 3 and TABLE 4, TABLE 5, TABLE 6. From the results it is concluded that the overall execution time performance of ECC is better in scenario-2 whereas performance of ECDH is better in scenario-1 when number of sensors greater than 200. From FIGURE 5a and FIGURE 5b, it is clear that enabling Pthreads in ECC implementation guarantees the parallel execution of decryption process and reduction in the overall execution time in both the scenarios. However, this is may not be true for any number of nodes.

Since the proposed scheme has been implemented on real system, the comparison results of both ECDH and ECC based implementations using prime192v3 and secp256k1 curves are

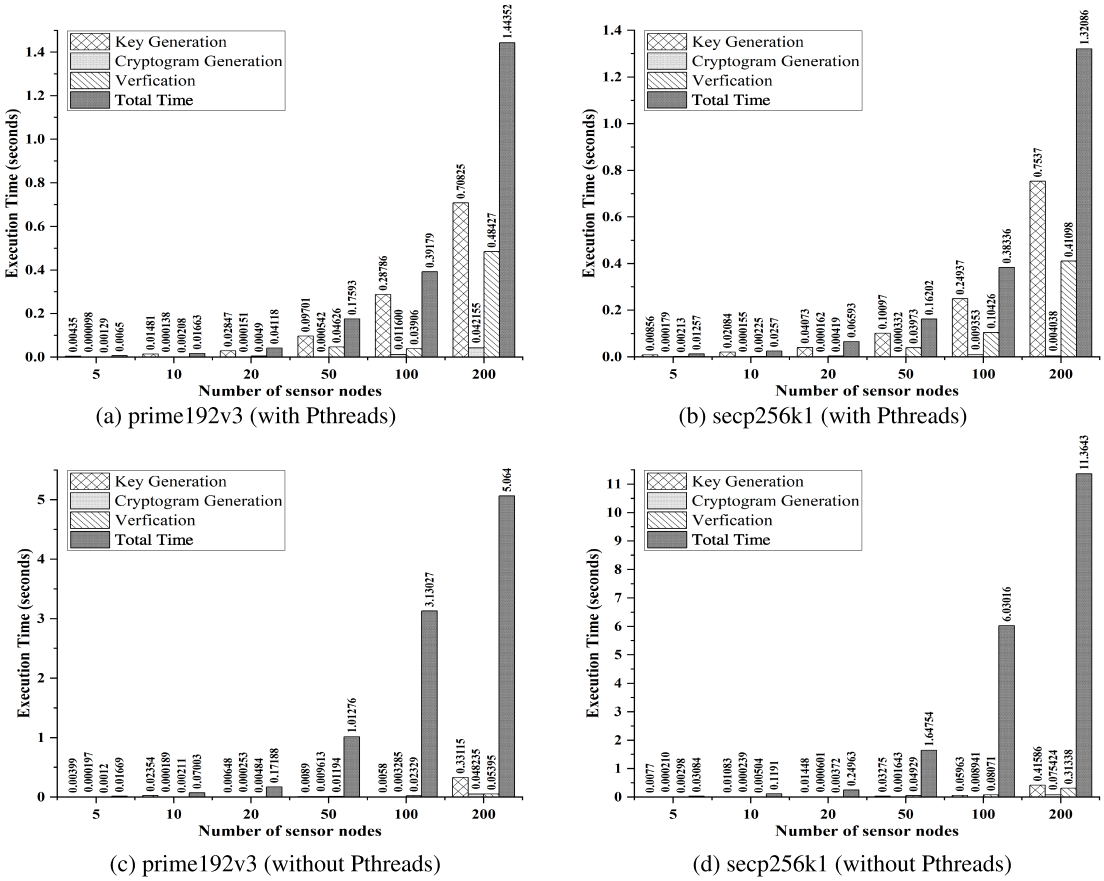


FIGURE 7. Implementation of ECIES-based authentication scheme of Section IV-C with and without Pthreads on scenario-2.

TABLE 3. Performance of ECDH in scenario-1 with different nodes.

Node	Encryption	Decryption	Fusion	Public Key Gen.	Secret Key Gen.	Total Time
10	0.00047	0.00047	0.00001	0.00496	0.01441	0.02033
20	0.00187	0.00188	0.00001	0.00545	0.04868	0.05791
50	0.03412	0.03415	0.00003	0.00450	0.40158	0.47440
100	0.32935	0.32941	0.00012	0.00313	1.44951	2.11154
200	3.54821	3.54839	0.00155	0.01031	6.86020	13.96868

TABLE 4. Performance of ECDH in scenario-2 with different nodes.

Node	Encryption	Decryption	Fusion	Public Key Gen.	Secret Key Gen.	Total Time
10	0.000009	0.000006	0.002102	1.1E-05	0.008368	0.010496
20	0.000011	0.000014	0.002637	0.000026	0.050366	0.053054
50	0.000011	0.000033	0.003322	0.000053	0.318926	0.322345
100	0.000011	0.000063	0.006700	0.000193	1.295854	1.302821
200	0.000012	0.000115	0.007487	0.000049	5.114338	5.122001

shown in FIGURE 5 and FIGURE 6. On the similar lines, performance of ECIES-based authentication scheme without Pthreads and with Pthreads using prime192v3, secp256k1 curves are shown in FIGURE 7. The experimental results

TABLE 5. Performance of ECC in scenario-1 with different nodes.

Nodes	Encryption	Decryption	Fusion	Public key generation	Total time
10	0.129969	0.087899	2.4E-05	0.003845	0.221737
20	0.530984	0.340896	2.4E-05	0.003799	0.875703
50	4.675096	0.519132	6E-05	0.004769	8.199057
100	15.300229	14.527483	0.000161	0.009092	29.836965
200	75.894176	73.156042	0.1539	0.041906	149.246024

TABLE 6. Performance of ECC in scenario-2 with different nodes.

Nodes	Encryption	Decryption	Fusion	Public key generation	Total time
10	0.016452	0.048267	0.004652	1.1E-05	0.069382
20	0.027244	0.11146	0.004504	1.7E-05	0.143225
50	0.061443	0.293645	0.007059	2.9E-05	0.362176
100	0.128405	0.685543	0.015128	9.9E-05	0.829175
200	0.261532	6.886953	0.051647	0.001358	7.20149

show that enabling Pthreads will linearly increase the key generation time and decreases overall execution time.

VII. CONCLUSION AND FUTURE SCOPE

The proposed lightweight fault-tolerant secure data communication framework consisting of Elliptic Curve Diffie-Hellman (ECDH)/Elliptic Curve Cryptography (ECC) based

secure communication and Elliptic Curve Integrated Encryption Scheme (ECIES) based authentication using Message Passing Interface (MPI) parallel program platform has been proved as a promising future for wireless sensor network communication. The proposed framework has been implemented on both single sink node and all sink nodes scenarios of WSN with parallel threads using Linux Pthreads and shows significant improvement in terms of overall speed. It is observed that the overall execution time performance of ECC is better in scenario-2 whereas the performance of ECDH is better in scenario-1 when number of sensors is greater than 200. It is also observed that enabling Linux Pthreads in ECC implementation guarantees the parallel execution of decryption process and a reduction in the overall execution time in both scenarios. Further, the dynamic addition and deletion of sensor nodes can make the proposed scheme more realistic. Also, with the help of massively parallel computing environments such as CUDA and OpenCL, the encryption and decryption phases of a node can be executed in parallel to reduce the overall execution time further. The key generation and distribution issues can be further exported to a trusted third-party so that sensor nodes can fully concentrate on the data fusion. Therefore, developing a dynamic WSN system, accelerating independent sequential processes through parallel computation and incorporating trusted third-party to handle key-related issues are the future directions.

CONFLICTS OF INTEREST

The authors have no conflicts of interest to declare.

REFERENCES

[1] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the Internet of Things: A review," in *Proc. Int. Conf. Comput. Sci. Electron. Eng.*, vol. 3, Mar. 2012, pp. 648–651.

[2] R. R. Swain, P. M. Khilar, and T. Dash, "Multifault diagnosis in WSN using a hybrid metaheuristic trained neural network," *Digit. Commun. Netw.*, vol. 6, no. 1, pp. 86–100, Feb. 2020.

[3] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient Byzantine fault-tolerance," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 16–30, Jan. 2013.

[4] D. D. Geeta, N. Nalini, and R. C. Biradar, "Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach," *J. Netw. Comput. Appl.*, vol. 36, no. 4, pp. 1174–1185, Jul. 2013.

[5] R. Klempous, J. Nikodem, L. Radosz, and N. Raus, "Byzantine algorithms in wireless sensors network," in *Proc. Int. Conf. Inf. Autom.*, Dec. 2006, pp. 319–324.

[6] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Deploying wireless sensor networks with fault-tolerance for structural health monitoring," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 382–395, Feb. 2015.

[7] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Saleem, M. Alzahrani, and O. Sheta, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, p. 1568, Apr. 2019.

[8] S. Jia, L. Ma, and D. Qin, "Fault detection modelling and analysis in a wireless sensor network," *J. Sensors*, vol. 2018, pp. 1–9, Oct. 2018.

[9] Y. Cheng, Q. Liu, J. Wang, S. Wan, and T. Umer, "Distributed fault detection for wireless sensor networks based on support vector regression," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–8, Oct. 2018.

[10] N. H. Vaidya and V. K. Garg, "Byzantine vector consensus in complete graphs," 2013, *arXiv:1302.2543*.

[11] H. Mendes and M. Herlihy, "Multidimensional approximate agreement in Byzantine asynchronous systems," in *Proc. 45th Annu. ACM Symp. Theory Comput.*, Jun. 2013, pp. 391–400.

[12] B. Ao, Y. Wang, L. Yu, R. R. Brooks, and S. S. Iyengar, "On precision bound of distributed fault-tolerant sensor fusion algorithms," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 1–23, Mar. 2017.

[13] N. A. Khan and A. Awang, "Elliptic curve cryptography for the security of insecure Internet of Things," in *Proc. Int. Conf. Future Trends Smart Communities (ICFTSC)*, Dec. 2022, pp. 59–64.

[14] R. Qazi, K. N. Qureshi, F. Bashir, N. U. Islam, S. Iqbal, and A. Arshad, "Security protocol using elliptic curve cryptography algorithm for wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 547–566, Jan. 2021.

[15] S. Bonomi, A. Del Pozzo, M. Potop-Butucaru, and S. Tixeuil, "Approximate agreement under mobile Byzantine faults," *Theor. Comput. Sci.*, vol. 758, pp. 17–29, Feb. 2019.

[16] L. Yuan and G. Qu, "Design space exploration for energy-efficient secure sensor network," in *Proc. IEEE Int. Conf. Application-Specific Syst., Archit., Processors*, 2002, pp. 88–97.

[17] I. Nadir, W. K. Zegeye, F. Moazzami, and Y. Astatke, "Establishing symmetric pairwise-keys using public-key cryptography in wireless sensor networks (WSN)," in *Proc. IEEE 7th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2016, pp. 1–6.

[18] B. Hu, W. Tang, and Q. Xie, "A two-factor security authentication scheme for wireless sensor networks in IoT environments," *Neurocomputing*, vol. 500, pp. 741–749, Aug. 2022.

[19] C. Dai and Z. Xu, "A secure three-factor authentication scheme for multi-gateway wireless sensor networks based on elliptic curve cryptography," *Ad Hoc Netw.*, vol. 127, Mar. 2022, Art. no. 102768.

[20] A. G. Mirsaraei, A. Barati, and H. Barati, "A secure three-factor authentication scheme for IoT environments," *J. Parallel Distrib. Comput.*, vol. 169, pp. 87–105, Nov. 2022.

[21] Y. Chen, X. Yang, T. Li, Y. Ren, and Y. Long, "A blockchain-empowered authentication scheme for worm detection in wireless sensor network," *Digit. Commun. Netw.*, 2022, doi: 10.1016/j.dcan.2022.04.007.

[22] K. H. Rezaeipour and H. Barati, "A hierarchical key management method for wireless sensor networks," *Microprocessors Microsyst.*, vol. 90, Apr. 2022, Art. no. 104489.

[23] P. Zhang, J. Wang, K. Guo, F. Wu, and G. Min, "Multi-functional secure data aggregation schemes for WSNs," *Ad Hoc Netw.*, vol. 69, pp. 86–99, Feb. 2018.

[24] A. Ullah, G. Said, M. Sher, and H. Ning, "Fog-assisted secure healthcare data aggregation scheme in IoT-enabled WSN," *Peer Peer Netw. Appl.*, vol. 13, no. 1, pp. 163–174, Jan. 2020.

[25] M. M. Afsar and E. Z. Kh, "A fault tolerant protocol for wireless sensor networks," in *Proc. 7th Int. Conf. Mobile Ad-Hoc Sensor Netw.*, Dec. 2011, pp. 475–478.

[26] M. H. Bashaa, S. M. Al-Alak, and A. K. Idrees, "Secret key generation in wireless sensor network using public key encryption," in *Proc. Int. Conf. Inf. Commun. Technol.*, Apr. 2019, pp. 106–112, doi: 10.1145/3321289.3321320.

[27] J. Deng and Y. S. Han, "Cooperative secret delivery in wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 14, no. 4, pp. 226–237, Jan. 2013, doi: 10.1504/IJAHUC.2013.058504.

[28] R. Di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proc. 1st ACM Workshop Secur. Ad Hoc Sensor Netw.*, Oct. 2003, pp. 62–71, doi: 10.1145/986858.986868.

[29] H. Li, K. Lin, and K. Li, "Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks," *Comput. Commun.*, vol. 34, no. 4, pp. 591–597, Apr. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366410001015

[30] M. Rahman and S. Sampalli, "An efficient pairwise and group key management protocol for wireless sensor network," *Wireless Pers. Commun.*, vol. 84, no. 3, pp. 2035–2053, Oct. 2015, doi: 10.1007/s11277-015-2546-4.

[31] P. Alimoradi, A. Barati, and H. Barati, "A hierarchical key management and authentication method for wireless sensor networks," *Int. J. Commun. Syst.*, vol. 35, no. 6, Apr. 2022, Art. no. e5076.

[32] H. Zhong, L. Shao, J. Cui, and Y. Xu, "An efficient and secure recoverable data aggregation scheme for heterogeneous wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 111, pp. 1–12, Jan. 2018.

[33] D. E. Boubiche, S. Boubiche, H. Toral-Cruz, A.-S.-K. Pathan, A. Bilami, and S. Athmani, "SDAW: Secure data aggregation watermarking-based scheme in homogeneous WSNs," *Telecommun. Syst.*, vol. 62, no. 2, pp. 277–288, Jun. 2016, doi: 10.1007/s11235-015-0047-0.

[34] V. Rao and K. V. Prema, "Light-weight hashing method for user authentication in Internet-of-Things," *Ad Hoc Netw.*, vol. 89, pp. 97–106, Jun. 2019.

KOLLU SIVA SAI is currently pursuing the B.Tech. degree in computer science and engineering (data science and analytics) from the Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. His current research interests include cryptography and machine learning.

RADHAKRISHNA BHAT (Member, IEEE) received the B.Eng. degree from Visveswaraya Technological University (VTU), India, and the integrated M.Tech. and Ph.D. degree from Visveswaraya Technological University, Belagavi, India. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal, India. He is an active researcher who has published more than 15 scientific research papers in reputed journals and conferences. His current research interests include information security, high-performance computing, blockchain technology, and machine learning.

MANJUNATH HEGDE received the master's degree in computer science from Mangalore University, Karnataka, India, in 2014, and the Ph.D. degree in mathematical and computational sciences from the National Institute of Technology Karnataka, India, in 2019. He is currently an Assistant Professor with the Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. He has published several papers in reputed journals and conferences. His current research interests include network security, information security, secure authentication, cryptography, and blockchain technology.

J. ANDREW received the B.E. and M.E. degrees in computer science from Anna University, Chennai, India, in 2011 and 2013, respectively, and the Ph.D. degree from the Vellore Institute of Technology(VIT), Vellore, India, in 2021. He is currently an Assistant Professor with the Department of Computer Science and Engineering (CSE), Manipal Institute of Technology (MIT), Manipal, India. He is an active researcher who has published more than 33 scientific research papers in reputed journals and conferences. He also served as a speaker at many prestigious conferences worldwide. He has ten years of teaching experience at the undergraduate (UG) and postgraduate (PG) levels. He has also supervised several projects at different levels at the university. His current research interests include data privacy, healthcare data analysis, deep learning, machine learning, computer vision, and blockchain technologies.