ECGSH: An Efficient Certificateless Group Signcryption based Homomorphic in Industrial IoT

Bei Gong, *Member, IEEE,* Mowei Gong, Zhe Li, Haotian Zhu, Weizhi Meng, *Senior, IEEE,* and Chong Guo

***Abstract*—With the growth of the Industrial Internet of Things (IIoT), millions of smart devices are transmitting and processing data globally. However, this extensive interconnectivity also poses significant security challenges, particularly in data transmission. Traditional security mechanisms often incur high computational costs and long processing times, which are impractical for resource-constrained devices. In this paper, we propose an efficient and secure data processing and transmission scheme for the IIoT called ECGSH. This scheme combines certificateless signcryption and homomorphic encryption to enable homomorphic processing in an encrypted state, thus enhancing both security and flexibility. Moreover, it reduces the complexity of large-scale data processing by eliminating bilinear pair computations. The ECGSH scheme also supports homomorphic data transmission in the IIoT. A rigorous security analysis proves that the scheme has the properties of confidentiality, non-repudiation, and forward security under the standard model. An attack resistance analysis proves that the scheme can effectively resist man-in-the-middle attacks, replay attacks, and eavesdropping attacks. The performance evaluation demonstrates that ECGSH excels in terms of security, computational efficiency, and communication overhead, making it particularly suitable for IIoT environments with limited resources and high transmission costs.**

***Index Terms*—Certificateless signcryption, homomorphic encryption, Industrial Internet of Things (IIoT), data transmission security, forward security**

I. Introduction

WITH the persistent advancement and application of Industry 4.0, the Industrial Internet of Things (IIoT) has developed as an important expansion of IoT innovations within the realm of industrial control. Fig. 1 shows a complex network architecture developed for the IIoT infrastructure [1]. This architecture achieves a high degree of interoperability and connectivity through real-time data exchange between cloud platforms and edge computing nodes. It helps improve production efficiency, achieve real-time data processing, and support predictive maintenance in essential industrial sectors such as manufacturing, energy, and transportation [2].

(Corresponding author: Weizhi Meng and Chong Guo.)

Bei Gong is with the Beijing Key Laboratory of Trusted Computing, College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: gongbei@bjut.edu.cn).

Mowei Gong, Zhe Li, Haotian Zhu and Chong Guo are with the College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: gongmowei@emails.bjut.edu.cn; leezhe627@emails.bjut.edu.cn; zhuhaotian@emails.bjut.edu.cn; 1594991663@qq.com).

Weizhi Meng is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark. (email: weme@dtu.dk)

Manuscript received XXXXX; revised XXXXXX.

The core of the IIoT is the perception and control domain, which performs data acquisition and feedback control. It comprises different complex perceptual heterogeneous systems with a large number of sensors, industrial instruments, control devices, and industrial control systems. In this domain, industrial data, including production data and equipment status, are collected, exchanged, and initially analyzed and processed in real-time, providing a decision-making foundation for automated industrial production through IIoT gateways [3]. While these advancements offer significant benefits, they also present considerable security challenges [4]. In particular, concerning data transmission, the heterogeneity of IIoT and frequent data exchange in production require that sensitive data be subjected to multiple forwarding, encryption, and decryption on resource-constrained devices. This inflexible transmission puts more performance pressure on IIoT environments with high real-time requirements and increases the data risk of tampering, eavesdropping, and forgery at each stage due to prevalent security vulnerabilities in industrial devices [5]. Moreover, when the IIoT dynamically adapts its devices to the demands of production tasks, uniform updates to security mechanisms on devices cause disruptions to data transmission, challenging the robustness of the IIoT network and leading to significant organizational losses.

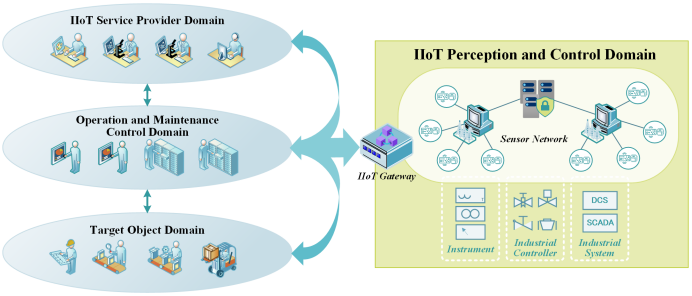


Fig. 1. The Architecture of the IIoT.

To address these challenges effectively, developing a secure data transmission scheme that can guarantee the security, reliability, and efficiency of data transmission in the perception and control domains is essential. It should be versatile for industrial devices with restricted computing control and capacity and should satisfy the strong real-time requirements of the IIoT while minimizing transmission overhead.

Traditional security mechanisms typically rely on separate encryption and signature processes to protect data in transit. However, these traditional methods can place significant per-

formance demands on the device and affect the timeliness of data transmission because they are computationally intensive [6].

To overcome these limitations, Zheng [7] proposed the concept of signcryption on the basis of the PKI framework; it accomplishes both encryption and signature verification in a single logical step. Even though the signcryption scheme under the PKI framework has high resource costs in terms of certificate and private key management, it laid the foundation for research on subsequent signcryption schemes [8]. To address the difficulties associated with the PKI framework, Barbosa et al. [9] presented a signcryption mechanism based on the certificateless PKC [10]. This approach has become central to secure data transmission in the IoT realm [11]. However, IIoT environments still face the critical challenge of reducing the number of computational steps and speeding up data processing while maintaining data confidentiality and integrity under low-overhead and real-time conditions [3], [12].

A. Related Work

In IIoT research, it is important to guarantee the security and real-time transmission of data, especially in the perception and control domain, which is based on the feedback control of a large amount of sensitive data on the production process. Certificateless signcryption has drawn considerable attention because of its potential to decrease the resource drain from key escrow and bypass the complexities of traditional public key infrastructure (PKI) and identity-based cryptographic (IDC) systems. Karati et al. [13] proposed a certificateless universal signcryption method optimized for resource-constrained devices, which increases computational and communication efficiency while maintaining security. However, its adaptability to device heterogeneity and dynamic environments needs further exploration. Additionally, Ahene et al. [14] and Cong et al. [15] made advancements in certificateless signcryption for smart grids and multicast edge environments, respectively, improving efficiency; however, the challenges of addressing extensive deployment and computational demands remain. Further contributions include that of Gong et al. [16], who designed a lightweight hybrid signcryption that offers multiple security features suitable for IIoT devices, albeit with potential transmission delays due to bilinear pairings. Chen et al. [17] proposed online/offline certificateless signcryption to increase the energy efficiency and responsiveness of IIoT devices, although its performance under unstable network conditions has yet to be thoroughly validated. Shim et al. [18] investigated a blockchain-based certificateless signature scheme that solves the key escrow problem and increases system security, although its complexity might restrict its application in resource-constrained environments.

Certificateless signcryption schemes provide effective security measures in the IIoT, but the scalability of certificateless signcryption remains a concern, particularly as data processing demands increase. Lu et al. [19] proposed a certificateless aggregated signcryption scheme based on pairing. Although no specific performance evaluation was conducted, this method still provides a conceptual solution to the poor scalability and low efficiency of certificateless signcryption as well as the key escrow problem in the identity-based aggregated signcryption scheme [20]. This pioneering work established a foundational framework for advancing research in certificateless aggregate signcryption in the IIoT. Similarly, Chen et al. [21] and Cui et al. [22] contributed to enhancing data transmission security in the IIoT; the former combined certificateless aggregate signatures with masked random numbers for smart grids, and the latter improved vehicle user security and message authentication efficiency. Unfortunately, both of these schemes lack empirical validation in IIoT environments.

Recent studies continue to explore advanced cryptographic techniques to address the inherent challenges of IIoT environments. Kim et al. [23] presented a certificateless aggregate signcryption framework that improves transmission efficiency at the expense of data security. Zhang [24] proposed a lightweight and efficient certificateless aggregation signcryption scheme for medical scenarios, but it has limited support for resource-constrained devices for data processing. Yu et al. [25] and Cong et al. [26] made strides in reducing the computational burden with elliptic curve-based and pairing-free certificateless signcryption methods; however, they also encountered issues with decryption errors and dynamic network adaptability. In the latest research, Gopisetti et al. [6] proposed a lightweight and secure certificateless aggregate signcryption scheme suitable for IIoT environments that is pairing-free. However, the scheme focuses more on theoretical performance analysis and does not consider the challenges of practical applications.

While certificateless aggregate signcryption schemes continue to improve in terms of verification cost efficiency, they still rely on traditional encryption and decryption processes, which place significant pressure on data collection and reception nodes when handling large-scale data. Homomorphic encryption provides a reasonable solution to this by enabling complex operations on encrypted data. Although this method faces great challenges in terms of computational efficiency, homomorphisms are particularly valuable for securely and efficiently processing sensitive industrial data.

To date, there has been limited research focusing on this application. Li et al. [27] analyzed security and privacy protection methods in smart grids, showing how homomorphic encryption can maintain data privacy while supporting efficient data aggregation. However, the computational complexity of their approach is high, and it has not been tested in real-world environments. Zhang et al. [28] examined methods of reducing overhead through network coding; however, the security of these methods heavily depends on the reliability of network coding, which may not be suitable for most IIoT scenarios. More recent studies by Lu et al. [29] and Li et al. [30] investigated lightweight homomorphic encryption privacy protection schemes for the IoT and IIoT, respectively, significantly reducing the burden of data encryption and decryption. However, Lu's model remains theoretical, whereas Li's model lacks crucial empirical evidence regarding security as well as performance comparisons.

Homomorphic encryption has great potential to enhance

TABLE I
COMPARISON OF EXISTING WORK AND THE PROPOSED SCHEME

Scheme	Basis	Applicability	TH	Security			Attack resistance			Efficiency	DS	SV
				S1	S2	S3	A1	A2	A3			
[19]	BP	–	⋆	IND-CCA2	EUF-CMA	–	⋆	✓	–	–	–	–
[27]	MO	✓	✓	IND-CPA	–	–	⋆	⋆	–	–	–	–
[29]	CRT	–	✓	IND-CPA	–	–	⋆	⋆	–	–	–	–
[13]	BP	✓	–	IND-CCA2	EUF-CMA	–	⋆	✓	–	–	–	✓
[23]	BP	✓	⋆	IND-CPA	EUF-CMA	⋆	✓	✓	⋆	✓	–	✓
[16]	BP	✓	–	IND-CCA2	EUF-CMA	⋆	⋆	✓	⋆	–	–	–
[25]	EC	✓	–	IND-CCA2	SUF-CMA	–	✓	✓	–	✓	–	–
[30]	LO	✓	✓	IND-CPA	–	–	⋆	⋆	–	✓	–	✓
[26]	EC	✓	⋆	IND-CCA2	EUF-CMA	⋆	⋆	✓	⋆	✓	–	–
[18]	BP	✓	–	IND-CPA	EUF-CMA	–	✓	✓	–	–	–	–
[6]	EC	✓	⋆	IND-CCA2	EUF-CMA	✓	✓	✓	✓	–	✓	–
ECGSH	EC	✓	✓	IND-CCA2	EUF-CMA	✓	✓	✓	✓	✓	✓	✓

Notes: TH:Transmission Homomorphism; DS: Dynamic Scalability; SV: Simulation Verification; S1: Confidentiality; S2: Non-repudiation; S3: Forward Security; A1: MITM; A2: Eavesdropping; A3: Replay; BP: Bilinear Pairing; MO: Modular Operation; CRT: Chinese Remainder Theorem; EC: Elliptic Curve; LO: Linear Operation; ⋆: Not fully satisfied.

data privacy and reduce node processing loads. Therefore, research on combining certificateless signcryption with homomorphic encryption is highly important. Table I summarizes classic and recent research on certificateless signcryption and homomorphic encryption within IIoT, focusing on applicability, transmission homomorphism, security, efficiency, etc. In the existing work, most schemes remain at the theoretical analysis stage and are limited by the security of historical data, which is crucial for IIoT. Moreover, the high cost of data transmission in most schemes compromises the real-time efficiency required for processing large volumes of data. Our research not only secures data but also effectively decentralizes the computational burden during processing, which is particularly suitable for resource-constrained devices. Additionally, the dynamic scalability of the ECGSH scheme enables it to adapt flexibly to the evolving architecture in IIoT as operational demands change.

B. Contributions

In this article, we propose an efficient certificateless group signcryption homomorphic data transmission scheme (ECGSH), which achieves secure transmission and processing of data in the perception and control domain of the IIoT. The main contributions of this article are summarized as follows.

- **Transmission homomorphism.** We integrate homomorphic encryption into the certificateless signcryption framework, effectively reducing the overhead associated with certificate management. Moreover, the transmission homomorphism of the scheme allows data to be processed while keeping it encrypted, thus enhancing the security and flexibility of data processing.
- **Efficiency.** We eliminate expensive bilinear pairwise computations and test them in the range of the IIoT communication data volume. The results show that the ECGSH scheme has obvious advantages for large-scale data processing. Compared with related schemes [31]–[38], the ECGSH scheme significantly reduces the computational cost in the signcryption, forwarded signcryption, and designcryption phases. Moreover, it reduces the impact of the security mechanism on the transmission speed, which makes it particularly suitable for industrial IoT environments with bandwidth constraints or high transmission costs.
- **Security.** We conducted an extensive security analysis of the ECGSH and verified that it satisfies IND-CCA2, EUF-CMA, and forward security requirements in the standard model. Additionally, the scheme is effective against common man-in-the-middle, replay, and eavesdropping attacks in the perception and control domain of the IIoT.
- **Dynamic scalability.** ECGSH supports the dynamic scaling of nodes within the IIoT perception and control domain. It allows new devices to join without large-scale system reconfiguration and adapts them to dynamic changes in network structure.

C. Organization of this paper

The remainder of this paper is structured as follows. Section II presents the preliminaries and describes the participating entities, the system model, and the security goals of the ECGSH scheme. We present the detailed construction of the ECGSH scheme in Section III. The security analyses and performance evaluation of ECGSH are presented in Section IV. Finally, Section V summarizes this work.

II. SYSTEM MODEL AND SECURITY GOALS

In this section, we review some background knowledge related to the ECGSH scheme and its security analysis, describe the participating entities involved in our scheme, and present the system model of the proposed scheme. Additionally, the security objectives that the scheme should satisfy are described.

A. Preliminaries

The knowledge of the Edwards curve [39], complexity assumption, homomorphic encryption of ECC, and adversary models are described in *Supplemental Material A*.

B. Formalization of participating entities

The nodes in the perception and control domain of the IIoT are distributed over a great physical distance and in a wide range according to the service requirements of different services. As shown in Fig. 2, industrial sensor nodes (ISNs) are grouped into clusters on the basis of their computational capacities, data flow, and physical locations, and they are managed by an edge cluster node (ECN). The edge data aggregation nodes (DANs) oversee these clusters within the network communication domain, acting as trusted entities and generating essential public parameters for the system. Additionally, the IIoT gateway functions as a boundary computation and management node, participating in operations related to edge data aggregation.

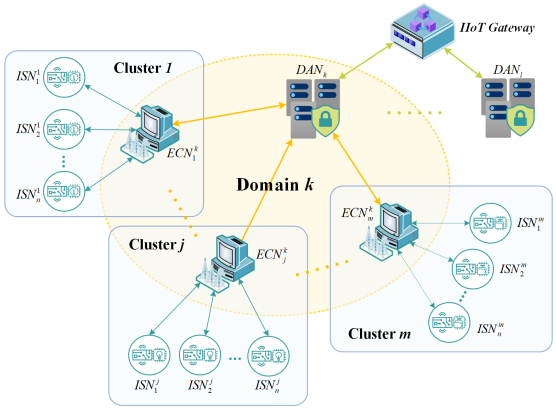


Fig. 2. The participating entities of the ECGSH scheme.

- **Industrial Sensor Node (ISN)**: In our scheme, the ISN is the basic unit of the perception and control domain. It comprises various sensors, RFIDs, multimedia, and industrial controls tasked with collecting state information on industrial resources such as machines, raw materials, processes, and environments. Typically, these nodes have low computational power and limited storage capacity.
- **Edge Cluster Node (ECN)**: In our scheme, the ECN is a node that processes, filters, compresses, and aggregates data from physically adjacent ISNs, reducing redundancy and noise to enhance data quality and efficiency.
- **Edge Data Aggregation Node (DAN)**: In our scheme, the DAN is the node with the strongest computational and storage capabilities. It processes, analyses, mines, and optimizes data transmitted by the ECNs to extract their value and significance, thereby supporting the service provider domain of the IIoT.

To clearly illustrate the relationships among the three types of nodes, the ECGSH uses ISN_i^j to represent the i^{th} ISN in cluster j and uses ECN_j^k to represent the j^{th} ECN in network region k.

C. Syntax of the Data Transmission Scheme

As shown in Fig. 3, the process of the ECGSH scheme consists of six phases: system initialization, key generation, member joining, signcryption, designcryption, and key update. The general implementation process of those phases is described as follows.

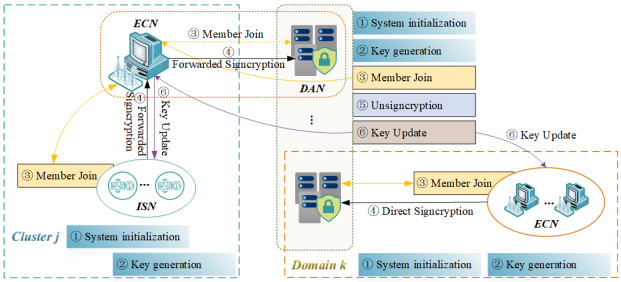


Fig. 3. The architecture of the ECGSH scheme model.

1) ***System Initialization***: This phase includes the identity information initialization algorithm $NodeInfoInit$, which is executed independently by each type of node entity and the system parameter initialization algorithm $GPInit$ executed by the DAN.
2) ***Key Generation***: This phase includes the system master key generation algorithm $GKeyGen$ executed by the DAN alone, the partial key generation algorithm $PaKeyGen$ executed in concert with the ECN and ISN, and the complete key generation algorithm $AllKeyGen$ executed by the ECN and ISN.
3) ***Member Joining***: This phase is executed by a node that has completed identity authentication and obtained member key pairs with the management node to which it belongs, using the algorithm $MemJoin$ to generate group membership credentials and group signature public keys.
4) ***Signcryption***: This phase involves two scenarios. One is executed by senders, using the $EnSign$ algorithm for direct communications; the other one is executed by intermediaries, using the $TrSign$ algorithm for forwarded communications.
5) ***Designcryption***: This phase is executed by data receivers, using the designcryption algorithms $DeEnSign$ to verify and decrypt the received information.
6) ***Key Update***: This phase is executed by the DAN, which uses the key update algorithm $ESKUpdate$ to send the update information to the management nodes within the relevant communication group and update the keys within the communication group.

D. Security Goals

To effectively counter eavesdropping, replay, man-in-the-middle, and other malicious attacks in the IIoT, the ECGSH scheme should ensure both the existence of homomorphisms during transmission and robust security. The specific security goals are as follows.

1) ***Transmission Homomorphism***: Considering the limited computing power of most industrial nodes in the IIOT

environment, industrial data are mainly collected and uploaded by ISNs and forwarded by ECNs. Therefore, our scheme needs to alleviate the pressure of decryption and distribute the data processing task to the forwarded nodes via homomorphic encryption.

2) ***Confidentiality***: To achieve confidentiality, our scheme needs to ensure that communication data are not vulnerable to eavesdropping or decryption by unauthorized persons during attacks. The security model is defined on the basis of the semantic security of signcryption data transmission under adaptive chosen ciphertext attacks (IND-ECGSH-CCA2).
3) ***Non-repudiation***: To achieve non-repudiation, our scheme needs to ensure that no valid signcryption information can be generated for any illegal participating entity during data transmission. The security model is based on the existential unforgeability of the signed data under adaptive chosen-plaintext attacks (EUF-ECGSH-CMA).
4) ***Forward security***: To achieve forward security, our scheme needs to ensure that a malicious attacker cannot obtain the session key of the previous period, even if the current session key is obtained. The security model is based on the long-term security of industrial data under adaptive selective ciphertext attacks (ECGSH-Fsec).
5) ***Attack resistance***: Security vulnerabilities in devices, network architectures, and communication protocols in the IIoT [40], [41] make industrial data vulnerable to replay, man-in-the-middle, and eavesdropping attacks [42]. Therefore, our scheme needs to be resistant to common malicious attacks.

III. Proposed Scheme

In this section, we give the details of the ECGSH scheme. The detailed flows of the system initialization, key generation, and member joining phases are shown in Fig. 4, the detailed flows of the signcryption and designcryption phases are shown in Fig. 5, and the detailed flow of the key update phase is shown in Fig. 6.

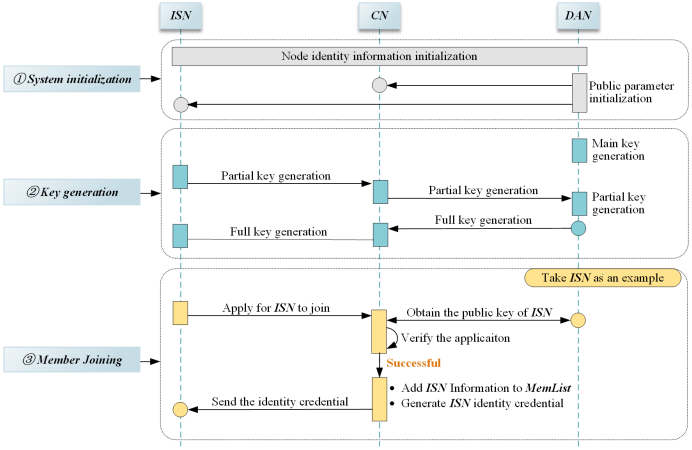


Fig. 4. The detailed process of the initialization, key generation and member joining phases.

(1) ***System Initialization Phase***

- $NodeInfoInit\left(\eta, Node\right) \rightarrow NodeID$:

The identity information of nodes $NodeID$ is initialized by the activation parameters η and the attributes of ISN_i^j, ECN_j^k and DAN_k.

- $GPInit\left(\theta\right) \rightarrow GP$:

Given the security parameter θ, DAN_k selects a large prime number p as the prime field order of the curve, and computes $E = \left\{(x,y) \in F_p \times F_p : x^2 + y^2 = c^2\left(1 + dx^2y^2\right)\right\}$ for node communication. Subsequently, E determines the isomorphic cyclic group G and its generator g, where $c, d \in F_p$ and $c^2d(1-d) \neq 0$. Then DAN_k selects four collision-resistant one-way hash functions: $h_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $h_2 : \{0,1\}^* \times G \rightarrow \mathbb{Z}_p^*$, $h_3 : \{0,1\}^* \times G \times G \rightarrow \mathbb{Z}_p^*$, and $h_4 : \{0,1\}^* \times G \times G \times G \rightarrow \mathbb{Z}_p^*$. Finally, DAN_k publishes the system parameters $GP = \{p, E, G, h_1, h_2, h_3, h_4\}$.

(2) ***Key Generation Phase***

- $GKeyGen\left(GP\right) \rightarrow (MaSK, MaPK)$:

DAN_k randomly selects $MaSK \in \mathbb{Z}_p^*$ as the master private key of domain k, and computes $MaPK = MaSK \cdot g$ as the system master public key. DAN_k stores $MaSK$ secretly and publishes $MaPK$.

- $PaKeyGen\left(GP, NodeID, MaSK, \psi\right) \rightarrow (PAPK, PASK)$:
  1) ISN_i^j completes authentication through *PUF* [43] to obtain the confirmation parameter ψ and updates $NodeID_{ISN_i^j}$. Next, it randomly selects $r_{pi1}, r_{pi2} \in \mathbb{Z}_p^*$ to compute $MI_i^j = r_{pi2}g$, $C_{p1} = h_1\left(NodeID_{ISN}\right) + r_{pi1}MaPK$, and $C_{p2} = r_{pi1}g$. Then, ISN_i^j sends $\left(MI_i^j, C_{p1}, C_{p2}\right)$ to ECN_j^k for group member private key request forwarding.
  2) ECN_j^k completes authentication through *PUF* to obtain the confirmation parameter ψ and updates $NodeID_{ECN_j^k}$. Next, it randomly selects $r_{pe1}, r_{pe2}, r_{pe3} \in \mathbb{Z}_p^*$ to compute $ME_j^k = r_{pe3}g$, $C_{p3} = C_{p1} + h_2\left(i \parallel j \parallel k, MI_i^j\right) + r_{pe1}MaPK$, $C_{p4} = C_{p2} + r_{pe1}g$, $C_{p6} = r_{pe2}g$, and $C_{p5} = h_1\left(NodeID_{ECN} \parallel j \parallel k\right) + r_{pe2}MaPK$. Then, ECN_j^k sends $\left(MI_i^j, ME_j^k, C_{p3}, C_{p4}, C_{p5}, C_{p6}\right)$ to DAN_k for the group partial private key and group member partial private key application.
  3) DAN_k randomly selects $r_{pd1}, r_{pd2}, r_{pd3} \in \mathbb{Z}_p^*$ to compute $MD_k = r_{pd1}g$, $HID_{Gr} = C_{p5} - MaSK \cdot C_{p6}$, $HID_{Mem} = C_{p3} - MaSK \cdot C_{p4}$, $PASK_{Gr} = r_{pd1} + MaSK \cdot h_3\left(HID_{Gr}, ME_j^k, MD_k\right)$, $PASK_{Mem} = r_{pd1} + MaSK \cdot h_3\left(HID_{Mem}, MI_i^j, MD_k\right)$, $C_{p7} = PASK_{Mem} + r_{pd2}MI_i^j$, $C_{p8} = r_{pd2}g$, $C_{p9} = PASK_{Gr} + r_{pd3}ME_j^k$, and $C_{p10} = r_{pd3}g$. Then, it sends $(MD_k, C_{p7}, C_{p8}, C_{p9}, C_{p10})$ to ECN_j^k for complete group key pair generation and group member key pair generation message forwarding.
- $AllKeyGen\left(GP, PASK\right) \rightarrow (ESPK, ESSK)$ or $\perp$:
  1) After ECN_j^k receives $(MD_k, C_{p7}, C_{p8}, C_{p9}, C_{p10})$, it computes $PASK'_{Gr} = C_{p9} - r_{pe3}C_{p10}$

and verifies whether $PASK'_{Gr}g$ equals $MD_k + MaPK \cdot h_3\left(HID_{Gr}, ME_j^k, MD_k\right)$. If not, the algorithm outputs $\perp$. Otherwise, ECN_j^k generates $ESSK_{Gr} = (r_{pe3}, PASK_{Gr})$ and stores $ESSK_{Gr}$. Then, it computes $ESPK_{Gr} = \left(ME_j^k, PASK_{Gr}g\right)$ and publishes $ESPK_{Gr}$.

2) ECN_j^k forwards (MD_k, C_{p7}, C_{p8}) to ISN_i^j. When ISN_i^j is received, it computes $PASK'_{Mem} = C_{p7} - r_{pi2}C_{p8}$ and verifies whether $PASK'_{Mem}g$ equals $MD_k + MaPK \cdot h_3\left(HID_{Mem}, MI_i^j, MD_k\right)$. If not, the algorithm outputs $\perp$. Otherwise, ISN_i^j generates $ESSK_{Mem} = (r_{pi2}, PASK_{Mem})$ and stores $ESSK_{Mem}$. Then, it computes $ESPK_{Mem} = \left(MI_i^j, PASK_{Mem}g\right)$ and publishes $ESPK_{Mem}$.

(3) ***Member Joining Phase***

- $MemJoin\,(GP, ESPK, ESSK, NodeID) \rightarrow (Cert, MemList, GrSI)$ or $\perp$:

1) ISN_i^j randomly selects $r_{mi1}, r_{mi2}, r_{mi3} \in \mathbb{Z}_p^*$ and records tp_{mi1} as the timestamp to compute $R_{mi} = r_{mi1}g$, $MID_{ISN} = HID_{ISN} + h_1\left(tp_{mi1}\right)$, $Q_{mi} = h_4\left(MID_{ISN}, R_{mi}, ESPK_{ISN}^{Mem}, h_1\left(tp_{mi1}\right)g\right)$, $C_{m1} = MID_{ISN} + r_{mi2}ESPK_{ISN}^{Gr}$, $C_{m2} = r_{mi2}g$, $C_{m3} = HID_{ISN} + r_{mi3}MaPK$, $C_{m4} = r_{mi3}g$, $V_{mi} = Q_{mi}ESSK_{ISN}^{Mem} + r_{mi1}$. Then, it sends $(C_{m1}, C_{m2}, C_{m3}, C_{m4}, R_{mi}, Q_{mi}, V_{mi})$ to the managing node of this cluster ECN_j^k to request to join and stores MID_{ISN} secretly.

2) After ECN_j^k receives the application from ISN_i^j, it verifies whether $V_{mi}g$ equals $Q_{mi}ESPK_{ISN}^{Mem} + R_{mi}$. If not, the algorithm outputs $\perp$. Otherwise, ECN_j^k sends (C_{m3}, C_{m4}) to DAN_k to obtain $ESPK_{ISN}$.

3) ECN_j^k computes $C_{m1} - C_{m2}ESSK_{ISN}^{Gr}$ and adds the result to $MemList_j^k$.

4) ECN_j^k randomly selects $r_{cert}, \xi_j^{GS} \in \mathbb{Z}_p^*$ and records tp_{cert} as the timestamp to generate the group signature public key $GrSI_j = \xi_j^{GS}g$.

5) Then, ECN_j^k generates the group membership credentials for the member node. It computes $R_{cert} = r_{cert}g$, $S_{cert} = r_{cert}GrSI_j$, $Q_{cert} = h_4\left(MID_{ISN} \parallel tp_{cert}, ESPK_{ISN}^{Gr}, R_{cert}, S_{cert}\right)$, $U_{cert} = Q_{cert}\xi_j^{GS} + r_{cert}$, and $Y_{cert} = h_1\left(Cert_{ISN} \parallel GrSI_j\right)$. Subsequently, ECN_j^k records that the group membership credential of ISN_i^j is $Cert_{ISN} = (C_{m1} \parallel C_{m2}, R_{cert}, S_{cert}, U_{cert})$ and sends $(Y_{cert}, Cert_{ISN}, GrSI_j, tp_{cert})$ to ISN_i^j.

6) After ISN_i^j receives the information, it verifies whether $h_1\left(Cert_{ISN} \parallel GrSI_j\right)$ equals Y_{cert} and $U_{cert}g$ equals $Q'_{cert}GrSI_j + R_{cert}$, where $Q'_{cert} = h_4\left(MID_{ISN} \parallel tp_{cert}, ESPK_{ISN}^{Gr}, R_{cert}, S_{cert}\right)$. If both are verified as equal, ISN_i^j stores $(Cert_{ISN}, GrSI_j, tp_{cert})$. Otherwise, the algorithm outputs $\perp$.

(4) ***Signcryption Phase***

We take the industrial data $m_1, m_2, ..., m_q$ collected by ISN_i^j and sent to DAN_k, which are forwarded by ECN_j^k, as an example to illustrate the details of the scheme in the signcryption phase.

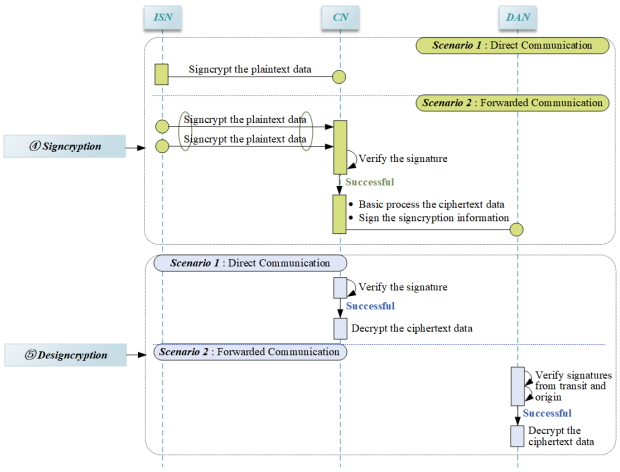


Fig. 5. The detailed process of the signcryption and designcryption phases.

- $EnSign\,(Msg, GP, ESPK, ESSK, Cert, GrSI) \rightarrow \sigma$:

1) Given the plaintext data $Msg = \{m_1, m_2, ..., m_q\}$, ISN_i^j generates a sequence of random numbers $Yr = \{\gamma_1, \gamma_2, ..., \gamma_q\}$ and computes the ciphertext data $CM = \{(c_{11}, c_{21}), ..., (c_{1q}, c_{2q})\}$ with Eq. (1).

$$\begin{cases} c_{1\vartheta} = m_\vartheta + \gamma_\vartheta ESPK_k^{Mem} \\ c_{2\vartheta} = \gamma_\vartheta g \end{cases} \tag{1}$$

2) ISN_i^j randomly selects $r_{es1} \in \mathbb{Z}_p^*$ and records $TPO_{es} = \{tpo_1, tpo_2, ..., tpo_q\}$ as the time series. It computes the relevant parameters via Eq. (2), where $\vartheta \in [1, q]$, and sends the signcryption information $\sigma = (CM, Cert_{ISN}, KRS, KV, UO, TPO_{es})$ to ECN_j^k, where $KRS = K_{es1} \parallel R_{os} \parallel S_{os}$ and $KV = K_{es2} \parallel V_{os}$.

$$\begin{cases} K_{es1} = MD_k r_{pi2} \\ K_{es2} = h_2\left(ESPK_j^{Gr}, K_{es1}\right) \\ R_{os} = r_{es1}g \\ S_{os} = r_{es1}GrSI_j \\ hcm_\vartheta = h_1\left(c_{1\vartheta} \parallel c_{2\vartheta}\right) \\ ts_\vartheta = h_3\left(hcm_\vartheta \parallel tpo_\vartheta, R_{os}, S_{os}\right) \\ UO = \left\{uo_1, ..., uo_q \,\middle|\, uo_\vartheta = ts_\vartheta PASK_i^{Mem} + r_{es1}tpo_\vartheta\right\} \\ V_{os} = h_4\left(Cert_{ISN}, MaPK, ESPK_j^{Gr}, GrSI_j\right) \end{cases} \tag{2}$$

- $TrSign\,(\sigma, GP, ESPK, ESSK, Cert, GrSI, \xi_{GS}) \rightarrow \sigma_{Tr}$ or $\perp$:

1) ECN_j^k is the forwarded node for this data transmission. First, it verifies the source trustworthiness of σ. ECN_j^k computes $MID_{ISN}^* = C_{m1} - C_{m2}ESSK_j^{Gr}$ and extracts the group membership credential $Cert_{ISN}^*$.

2) ECN_j^k verifies whether $V_{os} \stackrel{?}{=} V'_{os}$, $R_{os} \stackrel{?}{=} \frac{1}{q}\sum_{\vartheta=1}^{q} r_{u\vartheta}$, and $S_{os} \stackrel{?}{=} \frac{1}{q}\sum_{\vartheta=1}^{q} s_{u\vartheta}$ with Eq. (3), where $\vartheta \in [1, q]$. If all are verified as equal,

ECN_j^k forwards the message with its signature. Otherwise, the algorithm outputs $\perp$.

$$\begin{cases} V'_{os} = h_4\left(Cert^*_{ISN}, MaPK, ESPK_j^{Gr}, GrSI_j\right) \\ hcm'_\vartheta = h_1\left(c_{1\vartheta} \parallel c_{2\vartheta}\right) \\ ts'_\vartheta = h_3\left(hcm'_\vartheta \parallel tpo_\vartheta, R_{os}, S_{os}\right) \\ r_{u\vartheta} = \frac{1}{tpo_\vartheta}\left(uo_\vartheta g - ts'_\vartheta PAPK_i^{Mem}\right) \\ s_{u\vartheta} = \frac{1}{tpo_\vartheta}\left(uo_\vartheta GrSI_j - ts'_\vartheta \xi_j^{GS} PAPK_i^{Mem}\right) \end{cases} \tag{3}$$

3) Forwarded nodes can perform initial addition, subtraction, and scalar multiplication on encrypted data and then forward the processed signcrypted information after signing. We illustrate this by summing the encrypted data $y \in [1, w]$, resulting in $DealCM_j^{ECN} = \left\{\left(\sum_{e=1}^{x} c_{1e}, \sum_{e=1}^{x} c_{2e}\right), \left(c_{1(x+1)}, c_{2(x+1)}\right), ..., \left(c_{1q}, c_{2q}\right)\right\}$, where $x \in [1, q]$.
4) Then, ECN_j^k randomly selects $r_{es3}, r_{es4} \in \mathbb{Z}_p^*$ and records $TPR_{es} = \{tpr_1, tpr_2, ..., tpr_w\}$ as the time series, where $w = q - x + 1$. It computes the relevant parameters via Eq. (4), where $y \in [1, w]$, and sends the signcryption information $\sigma_{Tr} = (DCM, Cert_{ECN}, KRS_{tr}, KV_{tr}, UT, TPR_{es})$ to DAN_k, where $KRS_{tr} = K_{es3} \parallel R_{tr} \parallel S_{tr}$, $KV_{tr} = K_{es4} \parallel V_{tr}$, and $DCM = \left\{CM, DealCM_j^{ECN}\right\}$.

$$\begin{cases} K_{es3} = K_{es1} r_{pe3} \\ K_{es4} = h_1\left(K_{es2} \parallel K_{es3} g\right) \\ R_{tr} = r_{es3} g \\ S_{tr} = r_{es3} GrSI_k \\ hdm_y = h_1\left(c_{1y} \parallel c_{2y}\right) \\ tsr_y = h_3\left(hdm_y \parallel tpr_y, R_{tr}, S_{tr}\right) \\ UT = \left\{ut_1, ..., ut_w \,\middle|\, ut_y = tsr_y PASK_j^{Mem} + r_{es3} tpr_y\right\} \\ V_{tr} = h_4\left(Cert_{ECN}, MaPK, ESPK_k^{Gr}, GrSI_k\right) \end{cases} \tag{4}$$

(5) ***Designcryption Phase***

- $DeEnSign\left(\sigma_{(Tr)}, GP, Cert, ESPK, ESSK, \xi_{GrSI}\right) \rightarrow Msg^*$ or $\perp$:
  1) After DAN_k receives the signcryption message σ_{Tr}, it performs trusted verification of the forwarded node and the data transmission process in turn.
  2) DAN_k computes $MID^*_{ECN} = C_{m1} - C_{m2} ESSK_j^{Gr}$ and extracts the group membership credential $Cert^*_{ECN}$.
  3) Next, DAN_k uses Eq. (5) to verify the trustworthiness of ECN_j^k by checking whether $V_{tr} \stackrel{?}{=} V'_{tr}$, $R_d \stackrel{?}{=} \frac{1}{w}\sum_{y=1}^{w} r_{dy}$, and $S_d \stackrel{?}{=} \frac{1}{w}\sum_{y=1}^{w} s_{dy}$, where $\left\{\left(c_{1y}, c_{2y}\right), y \in [1, w]\right\} \subseteq DealCM_j^{ECN}$ and w is the number of processed ciphertexts.

$$\begin{cases} V'_{tr} = h_4\left(Cert^*_{ECN}, MaPK, ESPK_k^{Gr}, GrSI_k\right) \\ hdm'_y = h_1\left(c_{1y} \parallel c_{2y}\right) \\ tsr'_y = h_3\left(hdm'_y \parallel tpr_y, R_{tr}, S_{tr}\right) \\ r_{dy} = \frac{1}{tpr_y}\left(ut_y g - tsr'_y PAPK_j^{Mem}\right) \\ s_{dy} = \frac{1}{tpr_y}\left(ut_y GrSI_k - tsr'_y \xi_k^{GS} PAPK_j^{Mem}\right) \end{cases} \tag{5}$$

4) Then, DAN_kuses Eq. (6) to confirm the credibility of the data transmission process by determining whether $\kappa_1 \stackrel{?}{=} K_{es3} g$ and $\kappa_2 \stackrel{?}{=} K_{es4}$.

$$\begin{cases} \kappa_1 = r_{pd1} MI_i^j ME_j^k \\ \kappa_2 = h_1\left(h_2\left(ESPK_i^{Gr}, r_{pd1} MI_i^j\right) \parallel \kappa_1\right) \end{cases} \tag{6}$$

5) If both are verified as equal, DAN_k stores the original encrypted data and decrypts the processed information. Otherwise, the algorithm outputs $\perp$.
6) DAN_k decrypts the processed information according to homomorphism and obtains $Msg^* = \left\{m_1^*, m_2^*, ..., m_w^* \,\middle|\, m_y^* = c_{1y} - c_{2y} ESSK_k^{Mem}\right\}$, where $y \in [1, w]$.

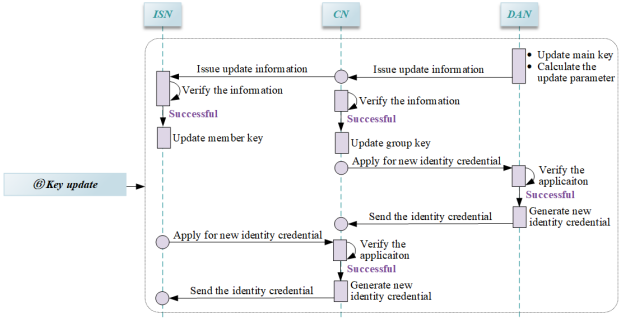


Fig. 6. The detailed process of the key update phase.

(6) ***Key Update Phase***

- $ESKUpdate\left(GP, \rho\right) \rightarrow \varpi$ or $\perp$:
  1) Given the key update parameter ρ, DAN_k sends the validity period of the communication key pair and group membership credentials to ρ. When a transmission is sent at time t_x, the nodes use the key pair or credential of the period τ, where $\tau = \left\lceil \frac{t_x - t_0}{\rho} \right\rceil - 1$ and t_0 is the start time of the period.
  2) DAN_k first updates the system master key pair in domain k, records the timestamp tp_ρ and randomly selects $MaSK_\rho \in \mathbb{Z}_p^*$, which is unrelated to $MaSK$. Subsequently, DAN_k computes $MaSK_{ud} = MaSK + MaSK_\rho \cdot h_1\left(tp_\rho\right)$ and $MaPK_{ud} = MaSK_{ud} \cdot g$, stores $MaSK_{ud}$ secretly, and publishes $MaPK_{ud}$ as the system master public key.
  3) Then, DAN_k randomly selects $\mu_\rho \in \mathbb{Z}_p^*$ and computes the key update factor ψ_ρ and the public update parameter Ψ_ρ for cluster j. Specifically, $\psi_\rho = \mu_\rho f\left(\tau\right) = \mu_\rho \sum_{x=0}^{\tau}\left(x+1\right) h_1\left(t_x\right)$is stored secretly within the node as the key update factor for the period, and $\Psi_\rho = \psi_\rho g$ is published as the public update parameter.
  4) DAN_k randomly selects $r_{ud1}, r_{ud2} \in \mathbb{Z}_p^*$, computes the update information $\varpi = \left(C_{u1} \parallel C_{u2}, C_{u3} \parallel C_{u4}\right)$ with Eq. (7) and sends it to ECN_j^k for key update.

$$\begin{cases} PASK_{ud}^{Gr} = PASK_j^{Gr} + \psi_\rho \cdot h_2\left(PASK_j^{Gr}, MaPK_{ud}\right) \\ PASK_{ud}^{Mem} = PASK_i^{Mem} + \psi_\rho \cdot h_2\left(PASK_i^{Mem}, MaPK_{ud}\right) \\ C_{u1} = PASK_{Gr}^{*} + r_{ud1} ME_j^k \\ C_{u2} = r_{ud1} g \\ C_{u3} = (PASK_{Mem}^{*} \parallel PASK_{Gr}^{*}) + r_{ud2} MI_i^j \\ C_{u4} = r_{ud2} g \end{cases} \tag{7}$$

5) After ECN_j^k receives ϖ, it computes $PASK_{Gr}^{*}{}' = C_{u1} - r_{pe3} C_{u2}$ and verifies whether $PAPK_j^{Gr} + \Psi_\rho \cdot h_2\left(PASK_j^{Gr}, MaPK_{update}\right)$ equals $PASK_{Gr}^{*}{}' g$. If not, the algorithm outputs $\perp$. Otherwise, ECN_j^k updates $\left(ESSK_{ud}^{Gr}, ESPK_{ud}^{Gr}\right)$ and publishes $ESPK_{ud}^{Gr}$.
6) ECN_j^k randomly selects $r_{ue1} \in \mathbb{Z}_p^*$ and sends (C_{u5}, C_{u6}) to DAN_k for credential update, where $C_{u5} = MID_{ECN} + r_{ue1} MaPK_{ud}$,$C_{u6} = r_{ue1} g$.
7) DAN_k verifies the identity credibility of ECN_j^k on the basis of (C_{u5}, C_{u6}). If the verification fails, the algorithm outputs $\perp$. Otherwise, DAN_k records the timestamp tp_{ECN}^* and updates $Cert_{ud}^{ECN} = \left(C_{u5} \parallel C_{u6}, R_{cert}, S_{cert}, U_{cert}^{ECN}\right)$, where $U_{ECN}^* = U_{cert}^{ECN} + Q_{ECN}^* MaSK_{update} + \psi_\rho$. Then, DAN_k computes $Y_{ECN}^* = h_1\left(Cert_{ud}^{ECN} \parallel GrSI_k\right)$ and sends $\left(Y_{ECN}^*, Cert_{ud}^{ECN}, tp_{ECN}^*\right)$ to ECN_j^k.
8) Finally, ECN_j^k receives this information and verifies it. If validation is passed, ECN_j^k updates the identity credential. Otherwise, the algorithm outputs $\perp$.
9) ISN_i^j is a member node in cluster j, and it updates its communication key pairs and the group membership credential in a similar way. Additionally, each node needs to delete the communication key and membership credentials used in the previous period as soon as it completes its respective update.

IV. Security and Performance

In this section, we analyze the security of the ECGSH scheme, evaluate its performance, and compare it with existing related schemes [31]–[38] in terms of computational cost, transmission overhead, and security characteristics. Specifically, we focus on the computational cost and transmission overhead of the ECGSH scheme during the signcryption, forwarded signcryption, and designcryption phases. This comparative evaluation aims to highlight the relative performance and security advantages of the ECGSH scheme in addressing challenges in the IIoT security landscape.

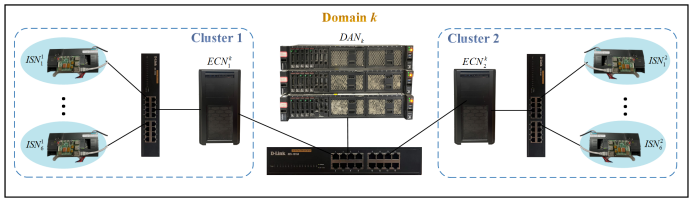


Fig. 7. The architecture of the ECGSH simulation experiment.

TABLE II
The run times for basic operations

Symbol	Operation	Runtime for 1 byte data (*ms*)		
		ISN	ECN	DAN
T_p	A bilinear pairing operation	5.7582	3.0156	1.357
T_{em}	Point multiplication on EC	1.2796	0.6701	0.3016
T_{ea}	Point addition on EC	0.4	0.08	0.01
T_{es}	Point subtraction on EC	0.5	0.1	0.015
T_h	A hash function operation	0.05	0.01	0.001
T_o	Other constant operations	0.01	0.001	0.0005

A. Security Analysis and Proof

We analyze the security of the ECGSH scheme using the two types of adversaries, which are given in *Supplemental Material A*. Detailed proofs concerning the transmission homomorphism, confidentiality, non-repudiation, forward security, and attack resistance of the ECGSH scheme are demonstrated in *Supplemental Material B*.

B. Efficiency and Applicability

To ensure that our solution is suitable for the IIoT environment, we consider a situation with low power consumption and resource constraints, which is typical of edge sensing devices within the perception and control domain of the IIoT. As depicted in Fig. 7, our evaluation environment simulates ISNs via a HollySys LE5109L PLC and Raspberry Pi 4B. The ECNs are simulated on PCs with Intel i7-8700 CPUs, 16 GB RAM, and Windows 10, whereas the DAN runs on servers with Intel Xeon E5 CPUs, 128 GB RAM, and CentOS 7. The hardware diversity reflects the varying computational capacities in IIoT, enabling us to evaluate the operational efficiency of our ECGSH scheme under realistic conditions. It also helps us to analyze how different data volumes and homomorphic encryption operations affect performance across various nodes.

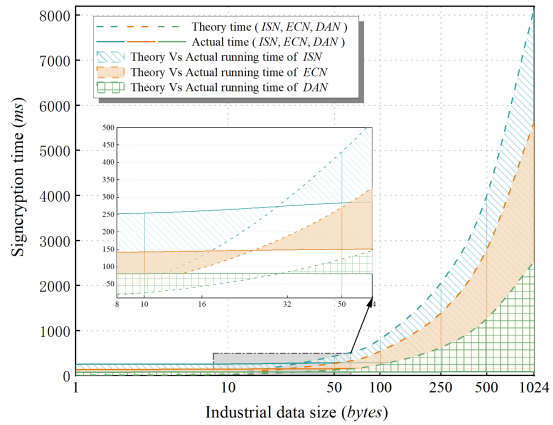
The runtimes of the basic operations on the ISN, ECN, and DAN are detailed in Table II. To evaluate the actual performance of our scheme in IIoT, we use these runtimes as a theoretical performance baseline and compare them with actual results.

Considering that data in the perception and control domain of the IIoT are typically transmitted and processed in small segments, we compared the theoretical and actual runtimes of the ISN, ECN, and DAN during the signcryption and designcryption phases for data volumes from tens to several hundreds of bytes in Fig. 8. The results indicate that the actual runtime of all nodes generally falls below the theoretical prediction starting at 50 bytes, underscoring the efficiency of the ECGSH scheme in IIoT. This discrepancy may be attributed to hardware optimization, parallel processing capabilities, and real-time algorithmic optimization during execution.

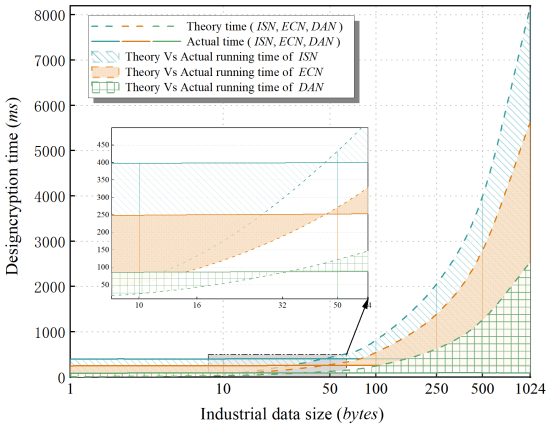
To further assess how the number of homomorphic encryption operations and data sizes affect performance, we designed a series of experiments covering numbers of encryption operations ranging from 1 to 30 and data sizes ranging from 1 byte to 1024 bytes. These tests are essential to validate the performance of the ECGSH scheme under different load

TABLE III
COMPARISON OF THE THEORETICAL COMPUTATION COSTS OF THE SCHEMES

Scheme	Signcryption	Designcryption	Direct Comm. on ISN (*ms*)	Forwarded Signcryption	Forwarded Signcryption on ECN in (*ms*)			
					$x=1$	$x=5$	$x=10$	$x=15$
Ara Anees (2017) [31]	$T_p+4T_{em}+T_h+T_o$	$2T_p+3T_{em}+T_h+2T_o$	26.3618	$(x+1)T_p+3xT_{em}+2xT_h+6xT_o$	8.0417	28.1461	53.2766	78.4071
Sui Zhiyuan (2020) [32]	$5T_{em}+3T_{ea}+3T_h$	$2T_p+3T_{em}+3T_{ea}+2T_h+T_o$	24.4132	$2T_p+T_{em}+(5x+1)T_{ea}+xT_h$	7.1814	8.7814	10.7814	12.7814
Rezaeibagha Fatemeh (2021) [33]	$T_p+4T_{em}+T_o$	$2T_p+4T_{em}+T_o$	27.5314	$T_p+(3x+1)T_{em}+xT_o$	5.6962	13.7378	23.7898	33.8418
Dohare Indu (2022) [34]	$9T_{em}+6T_{ea}+4T_h+T_o$	$T_p+3T_{em}+4T_h+3T_o$	23.9534	$T_p+2xT_{em}+2xT_{ea}+(2x+1)T_h+T_o$	4.5158	10.5166	18.0176	25.5186
Chen Xin (2023) [35]	$7T_{em}+4T_{ea}+4T_h+9T_o$	$6T_{em}+2T_{ea}+T_{es}+2T_h+4T_o$	19.9648	$(6x+7)T_{em}+(2x+4)T_{ea}+xT_{es}+(2x+4)T_h+(5x+9)T_o$	9.2914	26.413	47.815	69.217
Ramadan Mohammed (2023) [36]	$T_p+2T_{em}+5T_h+6T_o$	$5T_p+4T_h+4T_o$	37.6584	$(4x+1)T_p+(3x+1)T_h+(4x+1)T_o$	15.0781	63.3281	123.6406	183.9531
Xu Guangxia (2023) [37]	$T_p+3T_{em}+T_{ea}+T_h+T_o$	$2T_p+T_{em}+T_o$	22.863	$3xT_p+4xT_{em}+xT_{ea}+xT_h+2xT_o$	11.8075	59.0375	118.075	177.1125
Zhang Jianhong (2024) [38]	$5T_{em}+2T_{ea}+5T_h+8T_o$	$9T_{em}+6T_{ea}+3T_h+5T_o$	21.6444	$14xT_{em}+8xT_{ea}+8xT_h+13xT_o$	10.022	50.11	100.22	150.33
Our scheme (ECGSH)	$5T_{em}+T_{ea}+4T_h+3T_o$	$5T_{em}+3T_{es}+4T_h+2T_o$	15.146	$8T_{em}+2xT_{ea}+3T_{es}+(2x+5)T_h+(3x+3)T_o$	5.8211	6.4611	7.2611	8.0611



(a) Signcryption phase

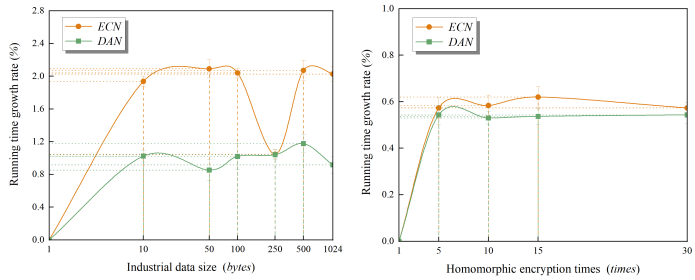


(b) Designcryption phase

Fig. 8. Theoretical vs actual runtime of the ECGSH scheme.

conditions, ensuring its reliability and effectiveness in IIoT environments.

Fig. 9 provides a detailed illustration of the variations in runtime growth rates when the ECN or DAN executes forwarded signcryption under different conditions. The results reveal that the number of homomorphic encryption operations has a relatively modest effect on performance, with variations ranging from 0.52% to 0.61%. In contrast, an increase in data size significantly affects performance, showing a greater variation of 0.8% to 2.09%. This suggests that compared with the number of homomorphic encryption operations, the amounts of data used for processing and transmission are more likely to be performance bottlenecks in IIoT environments.



(a) (b)

Fig. 9. Runtime growth rates of the ECGSH scheme in the forwarded signcryption phase: (a) 10 homomorphic encryption operations; (b) data size of 100 bytes.

Importantly, within the perception and control domain of the IIoT, data processing at any given time is typically minimal. In this context, our ECGSH scheme exhibits high efficiency

TABLE IV
COMPARISON OF THEORETICAL TRANSMISSION OVERHEADS

Scheme	Direct Signcryption in *bytes*	Forwarded Signcryption in *bytes*
(2017) [31]	$3\|G\|+\|Z_p\|+\|m\|=178$	$3\|G\|+\|Z_p\|+x\|m\|=128+50x$
(2020) [32]	$2\|G\|+2\|Z_p\|=128$	$2x\|G\|+2x\|Z_p\|=128x$
(2021) [33]	$4\|G\|=128$	$4\|G\|+x\|m\|=128+50x$
(2022) [34]	$2\|G\|+2\|Z_p\|+\|m\|=178$	$2x\|G\|+(x+1)\|Z_p\|+x\|m\|=32+146x$
(2023) [35]	$2\|G\|+2\|Z_p\|+\|m\|=178$	$2x\|G\|+2x\|Z_p\|+x\|m\|=178x$
(2023) [36]	$\|G\|+3\|Z_p\|+\|m\|=178$	$x\|G\|+3\|Z_p\|+x\|m\|=96+82x$
(2023) [37]	$4\|G\|+\|m\|=178$	$(3+x)\|G\|+x\|m\|=96+82x$
(2024) [38]	$2\|G\|+\|Z_p\|+\|m\|=146$	$2\|G\|+x\|Z_p\|+x\|m\|=64+82x$
Our scheme (ECGSH)	$2\|G\|+2\|Z_p\|=128$	$(2+x)\|G\|+2\|Z_p\|=128+32x$

in data processing and transmission, making it highly suitable for resource-constrained environments that demand rapid responsiveness. These results not only affirm the utility of our solution but also provide valuable data for its future optimization.

C. Computational Cost

In Table III, we present the results of a theoretical comparison with other relevant schemes ([31], [32], [33], [34], [35], [36], [37], [38]) in terms of computational cost. These schemes were selected because they represent innovative approaches to solving the security of data transmission in different IIoT scenarios. Moreover, the challenges addressed by these schemes are aligned closely with the focus of our scheme. We consider the cost of hash operations and other constant operations on the ISN in addition to the standard computational costs. Including this cost is critical, as our simulation experiments demonstrated that for ISNs, which are typically resource-limited, the cost of these additional operations becomes increasingly significant as the data volume expands.

From a theoretical perspective, using direct communication on the ISN as an example, we find that the ECGSH scheme incurs a signcryption cost of 7.028 ms and a designcryption cost of 8.118 ms, totaling approximately 15.146 ms. In comparison, nonbilinear pair-based signcryption schemes have higher costs: Sui's method [32] costs 24.4132 ms, Indu's method [34] costs 23.9534 ms, Chen's method [35] costs 19.9648 ms, and Zhang's method [38] costs 21.6444 ms. The computational costs for bilinear pair-based signcryption schemes in the IIoT environment are also presented in Table III.

Further analysis of IIoT data transmission patterns reveals that ECNs and DANs often serve as forwarders in forwarded signcryption scenarios. Therefore, we compared the computational cost under forwarded signcryption with that of other related scenarios by taking the example of homomorphic data processing on ECNs with different computational intensities (low-level, $x = 2$; medium-level, $x = 5, 10$; and high-level, $x = 15$). According to Table III, the ECGSH scheme shows a reduction in computational costs at a low computational intensity of up to 77.96% compared with that of Mohammed's [36] scheme and a reduction of as little as 0.58% compared with that of Indu's scheme [34]. At medium and high computational intensities, particularly when homomorphic data are processed 15 times on the ECN, the cost advantage of the ECGSH scheme is even more pronounced, ranging from a minimum of 36.93% compared with that of Sui's [32] to a maximum of 95.62% compared with that of Mohammed's [36].

To fully assess the computational cost benefits of the ECGSH scheme across its signcryption, forwarded signcryption, and designcryption phases, we set up a simulation experiment with the ISN as the sender, the ECN as the forwarder, and the DAN as the receiver. We compared the ECGSH scheme against similar schemes in terms of theoretical computational costs, including the schemes proposed by [32], [33], and [34]. The comparative results for transmitting 100 bytes of data at various computational intensities are shown in Fig. 10. The ECGSH scheme consistently shows at least a 15% reduction in time cost across all configurations, which increases to 25% as the computational intensity increases.

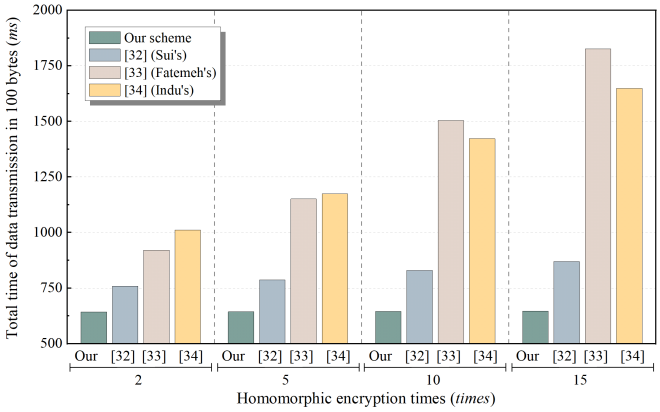


Fig. 10. Comparison of the total time of data transmission for x=2, 5, 10, and 15.

These experimental findings show that the ECGSH scheme offers enhanced applicability and computational cost benefits for transmitting tens to hundreds of bytes of data within the perception and control domain of the IIoT. Moreover, the potential for performance optimization of the scheme increases with increasing computational processing intensity. These results also suggest that the proposed scheme is more effective in mitigating the impact of security mechanisms on real-time data transmission in resource-constrained IIoT environments while still ensuring secure system operations.

D. Transmission Overhead

We assessed the transmission overhead in the signcryption phase and compared it with those of other relevant schemes ([31], [32], [33], [34], [35], [36], [37], [38]). The main

TABLE V
COMPARISON OF SECURITY CHARACTERISTICS

Scheme	Hard Problem Assumption	Transmission Homomorphism	Confidentialiy	Non-repudiation	Foward Security	Attack Resistance		
						MITM	Eavesdropping	Replay
[31]	CDH,BDH,DBDH	✓	IND-CPA	EUF-CMA	✓	✓	✓	✓
[32]	CDH,DDH	✓	IND-CCA2	EUF-CMA	✓	✓	✓	✓
[33]	BDH,DBDH	✓	IND-CPA	WUF	-	✓	-	-
[34]	CDH,q-BSDH	✓	IND-CCA2	EUF-CMA	-	✓	✓	-
[35]	ECDLP,CDH	-	IND-CPA	EUF-CMA	-	✓	-	-
[36]	DH,CDH,BDH	-	IND-CCA2	EUF-CMA	-	✓	✓	-
[37]	ECDLP,BDHDL	-	IND-CCA2	EUF-CMA	-	✓	✓	-
[38]	CDH,SDH	-	IND-CCA2	EUF-CMA	-	✓	✓	-
ECGSH	ECDLP,CDH	✓	IND-CCA2	EUF-CMA	✓	✓	✓	✓

theoretical transmission overheads for both the direct and forwarded signcryption scenarios are detailed in Table IV and were further validated through simulation experiments.

In our experiments, the size of the point on the Edwards-EC $|G|$ was 32 bytes, and the size of the constant $|Z_p|$ on Z_p^* was 32 bytes. We assumed that the size of the transmitted plaintext message $|m|$ was 50 bytes.

In the direct signcryption scenario, the transmission overhead of the ECGSH scheme is approximately 128 bytes, which is the same as in [32] and [33]. However, an analysis of the transmission overhead in forwarded signcryption scenarios with different computational intensities reveals that the overhead increases significantly with the number of homomorphic encryption operations. As shown in Fig. 11, the transmission overheads of Sui's [32] and Fatemeh's [33] schemes increase by 15.79% to 68.33% with increasing computational intensity compared with that of the ECGSH scheme.

These results demonstrate that the ECGSH scheme uses significantly fewer network resources than competing schemes do in IIoT environments characterized by limited bandwidth or high transmission costs, particularly in scenarios in which tens to hundreds of bytes of data are processed. This efficiency underscores the suitability of the ECGSH scheme for application scenarios with constrained network resources, confirming its practical potential and utility.

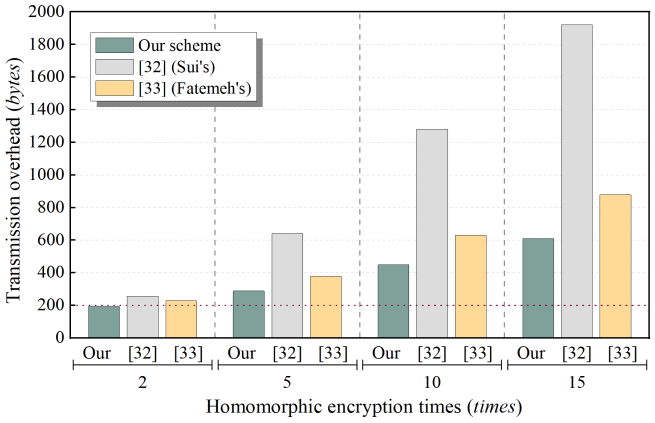


Fig. 11. Comparison of the transmission overhead in the forwarded signcryption phase.

E. Security Characteristics

The security characteristics of a scheme are critically important in transmitting sensitive industrial data. We therefore compared our scheme with other related schemes ([31], [32], [33], [34], [35], [36], [37], [38]) in hard problem assumptions, the security properties of the schemes, and their resistance to common attacks. The results are shown in Table V.

As indicated in Table V, the schemes ([31], [32], [33], [36]) based on the DH or BDH assumption generally require longer keys to maintain equivalent levels of security. In contrast, the schemes of [37] and [38], which rely on SDH or DBDH assumptions, provide enhanced security features within certain cryptographic protocols but have lower efficiency, which is essential in industrial applications. Our scheme employs the ECDLP and CDH assumptions, which offer robust security with shorter keys, effectively balancing the demands of efficiency and security in resource-constrained IIoT environments.

In terms of security characteristics, the ECGSH scheme has significant advantages over other related schemes. As demonstrated in Table V, the schemes in [31], [33], and [35] support only IND-CPA level confidentiality. In particular, the scheme in [33] supports only the weaker WUF standard in terms of non-repudiation, which may not be sufficient to protect sensitive data transmission in the IIoT. Additionally, the schemes in [33]–[38] do not support forward security, rendering them vulnerable to replay attacks.

However, the ECGSH scheme supports a higher level of IND-CCA2 confidentiality and EUF-CMA non-repudiation, including features such as transmission homomorphism and forward security. These attributes significantly increase the resistance of the ECGSH to man-in-the-middle (MITM), eavesdropping, and replay attacks.

In conclusion, the ECGSH method shows significant advantages in terms of comprehensive signcryption processing efficiency and security. It provides robust security protection for data transmission in IIoT environments and satisfies the real-time operational demands of the IIoT perception and control domain. This is achieved through its comprehensive security features and the homomorphic computing characteristics of signcryption information. However, real-machine experiments reveal that the ECGSH scheme does not perform optimally for transmissions below 50 bytes, which could limit its use in

IIoT where only small amounts of data are transmitted. Despite this, the scheme remains highly advantageous in medium or large IIoT environments. Thus, the ECGSH scheme is not only theoretically secure but also effective and applicable in practical settings.

V. Conclusion

In this paper, we proposed an efficient certificateless signcryption homomorphic data transmission scheme optimized for IIoT environments with limited device resources, extended distribution distances, and high real-time data transmission demands. The proposed ECGSH scheme enhances security through short key lengths and uses homomorphic encryption at edge nodes to improve data processing efficiency. It not only ensures real-time data transmission but also minimizes the computational load at collection points. The proposed ECGSH scheme has been verified to have the properties of transmission homomorphism, confidentiality, non-repudiation, and forward security, and its robustness against common attacks such as man-in-the-middle, replay, and eavesdropping was demonstrated. Through simulation experiments and theoretical comparisons with other related schemes, our ECGSH scheme was found to outperform existing techniques in terms of efficiency, lightweight, and security. Future research will focus on enhancing the scheme's performance on small data volume transmissions and investigate its adaptability to cross-domain communication, aiming to expand its applicability across a broader spectrum of industrial applications.

Acknowledgments

This work was supported in part by the Major Science and Technology Projects in Yunnan Province (202202AD080013).

References

[1] H. Xu, J. Wu, Q. Pan, X. Guan, and M. Guizani, "A survey on digital twin for industrial internet of things: Applications, technologies and tools," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2569–2598, 2023.

[2] R. Tallat, A. Hawbani, X. Wang, A. Al-Dubai, L. Zhao, Z. Liu, G. Min, A. Y. Zomaya, and S. Hamood Alsamhi, "Navigating industry 5.0: A survey of key enabling technologies, trends, challenges, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 1080–1126, 2024.

[3] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2021.

[4] Z. Hussain, A. Akhunzada, J. Iqbal, I. Bibi, and A. Gani, "Secure IIoT-enabled industry 4.0," *Sustainability*, vol. 13, no. 22, p. 12384, 2021.

[5] A. Saxena and S. Mittal, "Advanced persistent threat datasets for industrial IoT : A survey," in *2023 Second International Conference on Informatics (ICI)*, pp. 1–6, IEEE, 2023.

[6] S. R. Gopisetti, G. Thumbur, R. B. Amarapu, G. N. Bhagya, and P. V. Reddy, "A new lightweight and secure certificateless aggregate signcryption scheme for industrial internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2024. rate: 3.

[7] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption)," in *Advances in Cryptology — CRYPTO '97* (B. S. Kaliski, ed.), vol. 1294, pp. 165–179, Springer Berlin Heidelberg, 1997. Series Title: Lecture Notes in Computer Science.

[8] S. A. Hamad, Q. Z. Sheng, W. E. Zhang, and S. Nepal, "Realizing an internet of secure things: A survey on issues and enabling technologies," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, 2020.

[9] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pp. 369–372, ACM, 2008.

[10] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology - ASIACRYPT 2003* (C.-S. Laih, ed.), vol. 2894, pp. 452–473, Springer Berlin Heidelberg, 2003. Series Title: Lecture Notes in Computer Science.

[11] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4004–4022, 2021.

[12] S. F. Tan and A. Samsudin, "Recent technologies, security countermeasure and ongoing challenges of industrial internet of things (IIoT): A survey," *Sensors*, vol. 21, no. 19, p. 6647, 2021.

[13] A. Karati, C.-I. Fan, and R.-H. Hsu, "Provably secure and generalized signcryption with public verifiability for secure data transmission between resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10431–10440, 2019.

[14] E. Ahene, Z. Qin, A. K. Adusei, and F. Li, "Efficient signcryption with proxy re-encryption and its application in smart grid," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9722–9737, 2019.

[15] C. Peng, J. Chen, M. S. Obaidat, P. Vijayakumar, and D. He, "Efficient and provably secure multireceiver signcryption scheme for multicast communication in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6056–6068, 2020.

[16] B. Gong, Y. Wu, Q. Wang, Y.-h. Ren, and C. Guo, "A secure and lightweight certificateless hybrid signcryption scheme for internet of things," *Future Generation Computer Systems*, vol. 127, pp. 23–30, 2022. rate: 2.

[17] J. Chen, L. Wang, M. Wen, K. Zhang, and K. Chen, "Efficient certificateless online/offline signcryption scheme for edge IoT devices," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8967–8979, 2022.

[18] K.-A. Shim, "A secure certificateless signature scheme for cloud-assisted industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 6834–6843, 2024.

[19] H. Lu and Q. Xie, "An efficient certificateless aggregate signcryption scheme from pairings," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, pp. 132–135, IEEE, 2011.

[20] S. S. D. Selvi, S. S. Vivek, J. Shriram, S. Kalaivani, and C. P. Rangan, "Identity based aggregate signcryption schemes," in *Progress in Cryptology - INDOCRYPT 2009* (B. Roy and N. Sendrier, eds.), vol. 5922, pp. 378–397, Springer Berlin Heidelberg, 2009. Series Title: Lecture Notes in Computer Science.

[21] J. Chen and X. Ren, "A privacy protection scheme based on certificateless aggregate signcryption and masking random number in smart grid," in *Proceedings of the 2016 4th International Conference on Mechanical Materials and Manufacturing Engineering*, Atlantis Press, 2016.

[22] M. Cui, D. Han, and J. Wang, "An efficient and safe road condition monitoring authentication scheme based on fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 5, 2019.

[23] T.-H. Kim, G. Kumar, R. Saha, M. Alazab, W. J. Buchanan, M. K. Rai, G. Geetha, and R. Thomas, "CASCF: Certificateless aggregated SignCryption framework for internet-of-things infrastructure," *IEEE Access*, vol. 8, pp. 94748–94756, 2020.

[24] B. Zhang, "A lightweight data aggregation protocol with privacy-preserving for healthcare wireless sensor networks," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1705–1716, 2021.

[25] H. Yu and R. Ren, "Certificateless elliptic curve aggregate signcryption scheme," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2347–2354, 2022.

[26] C. Dai and Z. Xu, "Pairing-free certificateless aggregate signcryption scheme for vehicular sensor networks," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5063–5072, 2023.

[27] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *International Journal of Security and Networks*, vol. 6, no. 1, p. 28, 2011.

[28] P. Zhang, C. Lin, Y. Jiang, Y. Fan, and X. Shen, "A lightweight encryption scheme for network-coded mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2211–2221, 2014.

[29] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[30] S. Li, S. Zhao, G. Min, L. Qi, and G. Liu, "Lightweight privacy-preserving scheme using homomorphic encryption in industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14542–14550, 2022.

[31] A. Ara, M. Al-Rodhaan, Y. Tian, and A. Al-Dhelaan, "A secure privacy-preserving data aggregation scheme based on bilinear ElGamal cryptosystem for remote health monitoring systems," *IEEE Access*, vol. 5, pp. 12601–12617, 2017.

[32] Z. Sui and H. D. Meer, "An efficient signcryption protocol for hop-by-hop data aggregations in smart grids," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 132–140, 2020.

[33] F. Rezaeibagha, Y. Mu, and X. Huang, "Secure and privacy-preserved data collection for IoT wireless sensors," *IEEE INTERNET OF THINGS JOURNAL*, vol. 8, no. 24, 2021.

[34] I. Dohare, K. Singh, A. Ahmadian, S. Mohan, and P. Kumar Reddy M, "Certificateless aggregated signcryption scheme (CLASS) for cloud-fog centric industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6349–6357, 2022.

[35] X. Chen, D. He, M. K. Khan, M. Luo, and C. Peng, "A secure certificateless signcryption scheme without pairing for internet of medical things," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 9136–9147, 2023.

[36] M. Ramadan, G. Elbez, and V. Hagenmeyer, "Verifiable certificateless signcryption scheme for smart grids," in *2023 7th International Conference on System Reliability and Safety (ICSRS)*, pp. 181–189, IEEE, 2023. rate: 2.

[37] G. Xu, J. Dong, C. Ma, J. Liu, and U. G. Omar Cliff, "A certificateless signcryption mechanism based on blockchain for edge computing," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 11960–11974, 2023.

[38] J. Zhang, C. Dong, and Y. Liu, "Efficient pairing-free certificateless signcryption scheme for secure data transmission in IoMT," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4348–4361, 2024.

[39] H. Edwards, "A normal form for elliptic curves," *Bulletin of the American mathematical society*, vol. 44, no. 3, pp. 393–422, 2007.

[40] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[41] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.

[42] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, "Security issues in IIoT: A comprehensive survey of attacks on IIoT and its countermeasures," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, pp. 124–130, IEEE, 2018.

[43] X. Shan, H. Yu, Y. Chen, and Z. Yang, "Physical unclonable function-based lightweight and verifiable data stream transmission for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11573–11583, 2023.

APPENDIX A
SUPPLEMENTAL MATERIAL A

A. Edwards Curve

The Edwards curve is a specific form of elliptic curve proposed by Harold Edwards. The curve is defined as $x^2 + y^2 = c^2\left(1 + dx^2y^2\right)$, where $c, d \neq 0$ and $c^2d\left(1-d\right) \neq 0$. The advantage of the Edwards curve over traditional elliptic curves is that it maintains uniform speed and high resistance to attacks when performing group operations. It is also less susceptible to side-channel attacks because operations on the curves allow unconditional branching.

B. Complexity assumption

Definite 1 *(Elliptic Curve DL Problem (ECDLP))*: Consider an elliptic curve E over a finite field F_p, where p is a large prime. For two given points $P, Q \in E$, it is difficult to find a $k \in Z_q^*$ in any PPT algorithm such that $Q = kP$ with any PPT algorithm.

Definite 2 *(Computational Diffie-Hellman (CDH))*: Consider an elliptic curve E over a finite field F_p, where p is a large prime and where G is the generator of E. For a given tuple (G, aG, bG), it is difficult to compute abG by any PPT algorithm with unknown values of a and b.

C. Homomorphic Encryption of ECC

In elliptic curve cryptography (ECC), homomorphic encryption allows certain operations to be performed on encrypted data, and the results of these operations remain valid after decryption. Specifically, if $enc\left(m_1\right) = c_1$ and $enc\left(m_2\right) = c_2$, the properties of the encrypted data are as follows.

- *Additive homomorphism*: $dec\left(c_1 + c_2\right) = m_1 + m_2$
- *Subtractive homomorphism*: $dec\left(c_1 - c_2\right) = m_1 - m_2$
- *Scalar multiplicative homomorphism*: $dec\left(k \cdot c_2\right) = k \cdot m_2$, $k \in Z_q^*$

D. Adversary models

In the IIoT, malicious attacks such as eavesdropping, replay, and man-in-the-middle attacks will not only come from the outside but also from malicious internal attackers; thus, to prove the security of our scheme, two types of adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$ are assumed, and the capabilities of each adversary are as follows.

Adversary $\mathcal{A}_I$: This adversary can simulate the case where a dishonest node entity exists. This means that the adversary is able to replace the public key of any node entity with a value of its own choosing but does not have access to the communicating master's private key or part of the corresponding entity's private key.

Adversary $\mathcal{A}_{II}$: This adversary can simulate the presence of a malicious but passive node entity. This means that the adversary is able to access or establish the system parameters and communication master key pair on the basis of their choices and knows the partial private keys of individual node entities within the communication scope but cannot replace the public key of the corresponding node entity.

APPENDIX B
SUPPLEMENTAL MATERIAL B

A. Transmission Homomorphism

Theorem 1: For an ECGSH scheme, the scheme has transmission homomorphism if the signcryption message of an honest data sender or data forwarder, after undergoing mathematical operations (addition, subtraction, scalar multiplication), can be legitimately verified and correctly decrypted by an honest receiver.

Proof: Before transmission begins, honest data senders, forwarders, and receivers have honestly executed the algorithms of the initialization, key generation, and member join phases in their respective domains.

Assuming that data transmission occurs in domain k, the public parameters are $GP_k = \{p, E, G, h_1, h_2, h_3, h_4\}$, and the main communication key pair is $(MaPK_k, MaSK_k)$. Moreover, an honest node with $(Cert, ESSK, GrSI)$ can generate a signcryption message Λ for this data communication object.

Let $M_1 = \{m_1, m_2\}$, $M_2 = \{m_3, m_4\}$, and $M_3 = \{m_5\}$ be the industrial datasets collected by ISN_1^j, ISN_2^j, and ISN_3^j, respectively. In this data transmission, ECN_j^k is the forwarder, and the receiver is DAN_k. The sender executes $EnSign$ to output the signcryption information as follows.

$$\Lambda_1 = \left\{\left(c_{11}, c_{21}\right), \left(c_{12}, c_{22}\right), Cert_{ISN_1^j}, KRS^{ISN_1^j}, KV^{ISN_1^j}, UO_{ISN_1^j}, TPO_{es}^{ISN_1^j}\right\}$$

$$\Lambda_2 = \left\{\left(c_{13}, c_{23}\right), \left(c_{14}, c_{24}\right), Cert_{ISN_2^j}, KRS^{ISN_2^j}, KV^{ISN_2^j}, UO_{ISN_2^j}, TPO_{es}^{ISN_2^j}\right\}$$

$$\Lambda_3 = \left\{\left(c_{15}, c_{25}\right), Cert_{ISN_3^j}, KRS^{ISN_3^j}, KV^{ISN_3^j}, UO_{ISN_3^j}, TPO_{es}^{ISN_3^j}\right\}$$

After an honest data forwarder ECN_j^k receives $(\Lambda_1, \Lambda_2, \Lambda_3)$, it first verifies the source trustworthiness. We take Λ_3 as an example, and ECN_j^k is computed and verified as follows.

$$\widetilde{V_{os}^{ISN_3^j}} = h_4\left(Cert_{ISN_3^j}^*, MaPK, ESPK_j^{Gr}, GrSI_j\right) = V_{os}^{ISN_3^j}$$

$$TS^* = h_3\left(h_1\left(c_{15} \parallel c_{25}\right) \parallel tpo_1^{ISN_3^j}, R_{os}^{ISN_3^j}, S_{os}^{ISN_3^j}\right)$$

$$\widetilde{R_{os}^{ISN_3^j}} = \frac{1}{tpo_1^{ISN_3^j}}\left(uo_1^{ISN_3^j} g - TS^* PAPK_{ISN_3^j}^{Mem}\right)$$

$$= \frac{1}{tpo_1^{ISN_3^j}}\left(ts_1^{ISN_3^j} PASK_{ISN_3^j}^{Mem} g + r_{es1}^{ISN_3^j} tpo_1^{ISN_3^j} g - TS^* PAPK_{ISN_3^j}^{Mem}\right)$$

$$= r_{es1}^{ISN_3^j} g = R_{os}^{ISN_3^j}$$

$$\widetilde{S_{os}^{ISN_3^j}} = \frac{1}{tpo_1^{ISN_3^j}}\left(uo_1^{ISN_3^j} GrSI_j - TS^* \xi_j^{GS} PAPK_{ISN_3^j}^{Mem}\right)$$

$$= \frac{1}{tpo_1^{ISN_3^j}}\left(ts_1^{ISN_3^j} PASK_{ISN_3^j}^{Mem} GrSI_j + r_{es1}^{ISN_3^j} tpo_1^{ISN_3^j} GrSI_j - TS^* \xi_j^{GS} PAPK_{ISN_3^j}^{Mem}\right)$$

$$= r_{es1}^{ISN_3^j} GrSI_j = S_{os}^{ISN_3^j}$$

Subsequently, ECN_j^k applies addition, subtraction, and scalar multiplication operations to ciphertexts in Λ_1, Λ_2, and Λ_3, respectively. This results in Λ_1^{Tr}, Λ_2^{Tr}, and Λ_3^{Tr}, which are

then sent to the receiver. Specifically, the operations are defined as $(dc_{11}, dc_{21}) = (c_{11} + c_{12}, c_{21} + c_{22})$, $(dc_{12}, dc_{22}) = (c_{13} - c_{14}, c_{23} - c_{24})$, and $(dc_{13}, dc_{23}) = (2c_{15}, 2c_{25})$.

After an honest data receiver DAN_k receives $\left(\Lambda_1^{Tr}, \Lambda_2^{Tr}, \Lambda_3^{Tr}\right)$, it executes $DeEnSign$ to verify and decrypt these messages. The verification process is similar to that in the forwarding phase, and the decryption of the processed ciphertext is performed as follows.

$$
\begin{aligned}
M_1^* &= dc_{11} - dc_{21} ESSK_{DAN_k}^{Mem} \\
&= (c_{11} + c_{12}) - (c_{21} + c_{22}) \, ESSK_{DAN_k}^{Mem} \\
&= (m_1 + m_2) + (\gamma_1 + \gamma_2) \, ESPK_{DAN_k}^{Mem} - (\gamma_1 g + \gamma_2 g) \, ESSK_{DAN_k}^{Mem} \\
&= m_1 + m_2 \\
M_2^* &= dc_{12} - dc_{22} ESSK_{DAN_k}^{Mem} \\
&= (c_{13} - c_{14}) - (c_{23} - c_{24}) \, ESSK_{DAN_k}^{Mem} \\
&= (m_3 - m_4) + (\gamma_3 - \gamma_4) \, ESPK_{DAN_k}^{Mem} - (\gamma_3 g - \gamma_4 g) \, ESSK_{DAN_k}^{Mem} \\
&= m_3 - m_4 \\
M_3^* &= dc_{13} - dc_{23} ESSK_{DAN_k}^{Mem} = 2c_{15} - 2c_{25} ESSK_{DAN_k}^{Mem} \\
&= 2m_5 + 2\gamma_5 ESPK_{DAN_k}^{Mem} - 2\gamma_5 g ESSK_{DAN_k}^{Mem} \\
&= 2m_5
\end{aligned}
$$

Thus, an honest data receiver is able to accurately verify signed messages and decrypt ciphertexts that underwent operations such as addition, subtraction, and scalar multiplication during transmission. Consequently, the decrypted results will match precisely with those derived from direct computations on the associated plaintexts.

B. Confidentiality

Theorem 2: For data transmission in the IIoT, the ECGSH scheme satisfies confidentiality under adaptive chosen ciphertext attacks when neither $\mathcal{A}_I$ nor $\mathcal{A}_{II}$ can win the game in polynomial time $\mathcal{P}$ with the advantage $Adv_{\mu}^{IND-ECGSH-CCA2}$, where$\mu = \mathcal{A}_I, \mathcal{A}_{II}$.

Proof: Assuming that the adversary is able to win the game, there exists an algorithm $\mathcal{Y}$ that can obtain a solution to the ECDLP with the advantage $Adv_{IND-ECGSH-CCA2}$.

- **Adversary $\mathcal{A}_I$ of type I**

Setup: Algorithm $\mathcal{Y}$ initializes the node identity via $NodeInfoInit$ and generates public parameters $Gparams = \{p, E, G, h_1, h_2, h_3, h_4\}$ and the master key pair (mpk, msk) through $GPInit$ and $GKeyGen$. While $\mathcal{Y}$ shares $(p, E, G, h_1, h_2, h_3, h_4, mpk)$ with $\mathcal{A}_I$, it keeps msk secret and treats $\mathcal{A}_I$ as a subroutine for the challenger role in game interaction. Additionally, $\mathcal{Y}$ maintains lists L_1, L_2, L_3, L_4, L_c, and L_w, which are initially empty. We assume that $\mathcal{Y}$ selects the ith node in the jth cluster in domain k as a challenge node and that the identity information of $NodeID_{ISN_i^j}$ is not available to $\mathcal{A}_I$.

Phase 1: $\mathcal{A}_I$ executes a finite number of adaptive queries within polynomial bounds.

1) H_1 *query*: L_1 is used to track the results of this query. When $\mathcal{A}_I$ queries H_1 about the $info_x$ of $NodeID_x$, $\mathcal{Y}$ checks whether $(info, hinfo)$ of L_1 includes $info_x$. If so, the corresponding $hinfo_x$ is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ randomly selects $hinfo_x \in \mathbb{Z}_p^*$ to return and updates L_1.
2) H_2 *query*: L_2 is used to track the results of this query. When $\mathcal{A}_I$ queries H_2 about the $ESPK_x^{Gr}$ and K_x of $NodeID_x$, $\mathcal{Y}$ checks whether the $(ESPK, K, hpk)$ of L_2 includes $\left(ESPK_x^{Gr}, K_x\right)$. If so, the corresponding hpk_x is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ randomly selects $hpk_x \in \mathbb{Z}_p^*$ to return and updates L_2.
3) H_3 *query*: L_3 is used to track the results of this query. When $\mathcal{A}_I$ queries H_3 about the $HTPR_x$, R_x and S_x of $NodeID_x$, $\mathcal{Y}$ checks whether the $(HTPR, R, S, TSR)$ of L_3 includes $(HTPR_x, R_x, S_x)$. If so, the corresponding TSR_x is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ randomly selects $TSR_x \in \mathbb{Z}_p^*$ to return and updates L_3.
4) H_4 *query*: L_4 is used to track the results of this query. When $\mathcal{A}_I$ queries H_4 about the $Cert_x$, mpk, $ESPK_x^{Gr}$, $GrSI_x$ of $NodeID_x$, $\mathcal{Y}$ checks whether the $(Cert, MaPK, ESPK, GRSI, V)$ of L_4 includes $(Cert_x, mpk, ESPK_x^{Gr}, GrSI_x)$. If so, the corrresponding V_x is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ randomly selects $V_x \in \mathbb{Z}_p^*$ to return and updates L_4.
5) $OMJoin$ *query*: L_c is used to track the results of this query. When $\mathcal{A}_I$ queries $OMJoin$ via $NodeID_x$, $NodeLC_x$ and $ESPK_x^{Gr}$, $\mathcal{Y}$ checks whether $\left(NodeID, NodeLC, ESPK^{Gr}, Cert, GrSI, tp_{cert}\right)$ of L_c includes $\left(NodeID_x, NodeLC_x, ESPK_x^{Gr}\right)$. If so, the corresponding $\left(Cert_x, GrSI_x^j, tp_{cert_x}\right)$ is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ randomly selects $r_{m1}, r_\zeta, r_{mc}, tp_{x1}, tp_{x2} \in \mathbb{Z}_p^*$ to compute $MID_x = h_1\left(NodeID_x\right) + h_2\left(i \parallel j \parallel k, MI_i^j\right) + h_1\left(tp_{x1}\right)$, $GrSI_x^j = r_\zeta g$, $R_x = r_{mc} g$, $S_x = r_{mc} GrSI_x^j$, $C_{m1} = MID_x + r_{m1} ESPK_x^{Gr}$, $C_{m2} = r_{m1} g$, and $U_x = h_4\left(MID_x \parallel tp_{x2}, ESPK_x^{Gr}, R_x, S_x\right) r_\zeta + r_{mc}$ and generates $Cert_x = \left(C_{m1} \parallel C_{m2}, R_x, S_x, U_x\right)$ as the answer to return. On the basis of the answer, $\mathcal{Y}$ updates L_c.
6) *Public key query*: L_w is used to track the results of this query. When $\mathcal{A}_I$ queries the public key via $NodeID_x$ and its $NodeLC_x$, $\mathcal{Y}$ checks whether $\left(NodeID, NodeLC, \psi, ESPK^{Mem}, ESPK^{Gr}, PASK^{Mem}, ESSK^{Mem}\right)$ of L_w includes $(NodeID_x, NodeLC_x)$. If so, the corresponding $\left(ESPK_x^{Mem}, ESPK_x^{Gr}\right)$ is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ checks whether $NodeID_x$ equals $NodeID_{ISN_i^j}$ and updates L_w. If not, $\mathcal{Y}$ executes both the partial private key extraction(***PPKE***) query and the private key extraction(***PKE***) query. Otherwise, $\mathcal{Y}$ randomly selects $r_{pk1} \in \mathbb{Z}_p^*$ and $mgr_{pk} \in G$ to compute $ESPK_x^{Mem} = (r_{pk1} g, r_{pk1} g + e_{h1} mpk)$ and $ESPK_x^{Gr} = (mgr_{pk}, r_{pk1} g + e_{h2} mpk)$, where e_{h1} and e_{h2} are obtained with the query result of H_3. After these computations, $\mathcal{Y}$ returns $\left(ESPK_x^{Mem}, ESPK_x^{Gr}\right)$ to $\mathcal{A}_I$.
7) *Partial Private Key Extraction (PPKE) query*: L_w is used to track the results of this query.

When $\mathcal{A}_I$ is queries the partial private key using $NodeID_x$, $NodeLC_x$ and ψ_x, $\mathcal{Y}$ checks whether $(NodeID, NodeLC, \psi, ESPK^{Mem}, ESPK^{Gr}, PASK^{Mem}, ESSK^{Mem})$ of L_w includes $(NodeID_x, NodeLC_x, \psi_x)$. If so, the corresponding $PASK_x^{Mem}$ is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ checks whether $NodeID_x$ equals $NodeID_{ISN_i^j}$. If so, $\mathcal{Y}$ abandons this game. Otherwise, $\mathcal{Y}$ randomly selects $r_{psk} \in \mathbb{Z}_p^*$ to compute $PASK_x^{Mem} = r_{psk} + e_{h1} msk$, where e_{h1} is obtained from the query result of H_3. Then, $\mathcal{Y}$ returns $PASK_x^{Mem}$ to $\mathcal{A}_I$ and updates L_w.

8) *Replace Public Key (RPK) query*: L_w is used to track the results of this query. When $\mathcal{A}_I$ uses $NodeID_{ISN_x^j}$ and $ESPK_x^{Mem}$ to perform this query, $\mathcal{Y}$ updates L_w and returns $(NodeID, \phi, \phi, ESPK^{Mem}, \phi, \phi, \phi)$ as the answer.
9) *Private Key Extraction (PKE) query*: L_w is used to track the results of this query. When $\mathcal{A}_I$ queries the partial private key via $NodeID_x$ and $PASK_x^{Mem}$, $\mathcal{Y}$ checks whether $(NodeID, NodeLC, \psi, ESPK^{Mem}, ESPK^{Gr}, PASK^{Mem}, ESSK^{Mem})$ of L_w includes $(NodeID_x, PASK_x^{Mem})$. If so, the corresponding $ESSK_x^{Mem}$ is returned to $\mathcal{A}_I$ as the answer. Otherwise, $\mathcal{Y}$ checks whether $NodeID_x$ equals $NodeID_{ISN_i^j}$. If so, $\mathcal{Y}$ abandons this game. Otherwise, $\mathcal{Y}$ randomly selects $r_{sk} \in \mathbb{Z}_p^*$ to compute $ESSK_x^{Mem} = (PASK_x^{Mem} - e_{hsk} msk, PASK_x^{Mem})$, where e_{hsk} is obtained with the query result of H_3. Then, $\mathcal{Y}$ returns $ESSK_x^{Mem}$ to $\mathcal{A}_I$ and updates L_w.
10) *Signcryption query*: Assume that $\mathcal{Y}$ has finished querying the relevant oracle machine. When $\mathcal{A}_I$ performs a signcryption query regarding the data transmission between $NodeIDse$ and $NodeIDre$, $\mathcal{Y}$ checks whether $NodeID_{se}$ equals $NodeID_{ISN_i^j}$. If not, $\mathcal{Y}$ executes the signcryption process of the ECGSH scheme and returns σ_{ES} to $\mathcal{A}_I$ as the result. Otherwise, $\mathcal{Y}$ performs the public key query, PPE query, and PKE query on $NodeID_{re}$ to generate the signcryption information σ_{ES} as follows. Additionally, we assume that the sender is an ECN and that the receiver is a DAN.

    a) Select $\gamma_1 \in \mathbb{Z}_p^*$ to compute $c_{2x} = \gamma_1 g$, $c_{1x} = m_x + \gamma_1 ESPK_{re}^{Mem}$ and $cm_x = c_{1x} \parallel c_{2x}$.
    b) Select $r_{1x} \in \mathbb{Z}_p^*$ and record the timestamp tpo_x to compute $K_{es1} = r_{pd1} ME_{se}$, $K_{es2} = h_2(ESPK_{se}^{Gr}, K_{es1})$ and update L_2.
    c) Compute $hcm_x = h_1(cm_x)$ and update L_1.
    d) Compute $R_{os} = r_{1x} g$, $S_{os} = r_{1x} GrSI_{re}$, and $ts_x = h_3(hcm_x \parallel tpo_x, R_{os}, S_{os})$ and update L_3.
    e) Compute $NID_{se} = h_1(NodeID_{se}) + h_2(i \parallel j \parallel k, ME_{se}^j)$, $PSK_{se}^{Mem} = r_{pd1} + msk \cdot h_3(NID_{se}, ME_{se}^j, MD_{se})$, and $V_{os} = h_4(Cert_{se}, mpk, ESPK_{se}^{Gr}, GrSI_{se})$ and updates L_4.
    f) Generate signcryption information $\sigma_{ES} = (cm_x, Cert_{se}, KRS_{se}, KV_{se}, uo_x, tpo_x)$ as the answer to $\mathcal{A}_I$. When there is forwarded communication, $\mathcal{Y}$ is obtained in a similar way as described above.
    g) After $NodeID_{re}$ receives σ_{ES}, it computes and verifies V'_{os}, R'_{os} and S'_{os}.

11) *Designcryption query*: Assume that $\mathcal{Y}$ has finished querying the relevant oracle machine. When $\mathcal{A}_I$ performs a designcryption query regarding the signcryption information σ_{DES} between $NodeIDse$ and $NodeIDre$, $\mathcal{Y}$ checks whether $NodeID_{se}$ equals $NodeID_{ISN_i^j}$. If not, $\mathcal{Y}$ executes the designcryption process of the ECGSH scheme and returns m_{dx} to $\mathcal{A}_I$ as the result. Otherwise, $\mathcal{Y}$ performs a public key query, PPE query, and PKE query on $NodeID_{se}$ to obtain the data in plaintext m_{dx} as follows. Additionally, we assume that the sender is an ECN and that the receiver is a DAN.

    a) Compute $hdm_{dx} = h_1(cm_{dx})$ and update L_1.
    b) Compute $ts_{dx} = h_3(hdm_{dx} \parallel tpo_{dx}, R_{ds}, S_{ds})$ and update L_3.
    c) Retrieve L_4 and verify whether $V' = V_{ds}$. If not, the algorithm outputs $\perp$.
    d) Retrieve L_3 and verify whether $R'_{ds} = R_{ds}$ and $S'_{ds} = S_{ds}$, where $R'_{ds} = \frac{1}{tpo_{dx}}(uo_{dx} g - TSR' PAPK_{se}^{Mem})$ and $S'_{ds} = \frac{1}{tpo_{dx}}(uo_{dx} GrSI_{re} - TSR' \xi_{re}^{GS} PAPK_{se}^{Mem})$. If either equation is not true, the algorithm outputs $\perp$.
    e) If both of the above equations are true, $\mathcal{Y}$ computes $r_{pd} = PASK_{se}^{Mem} - msk \cdot h_3(NID'_{se}, ME_{se}^j, MD_{re})$ and $PASK_{re}^{Mem} = r_{pd} + msk \cdot h_3(NID'_{re}, MD_{re}, MD_{re})$, where $NID'_{se} = h_1(NodeID_{se}) + h_2(i \parallel j \parallel k, ME_{se}^j)$, $NID'_{re} = h_1(NodeID_{re}) + h_2(i \parallel j \parallel k, MD_{re})$.
    f) Compute $m_{dx} = c_{1x} - c_{2x} ESSK_{re}^{Mem}$ as the answer to $\mathcal{A}_I$, where $ESSK_{re}^{Mem} = (r_{pd}, PASK_{re}^{Mem})$.
    g) When there is forwarded communication, $\mathcal{Y}$ also needs to verify whether (K_{ds3}, K_{ds4}) satisfies $ME_{se} MX_{trans} MD_{re} = K_{ds3} g$ and $h_1(K' \parallel K_{ds3}) = K_{ds4}$ by retrieving L_2. If both are verified as equal, $\mathcal{Y}$ executes (e)-(f). Otherwise, the algorithm outputs $\perp$.

Challenge: After completing the *Phase 1* queries, $\mathcal{A}_I$ selects two different nodes $NodeID_s^*$ and $NodeID_r^*$ and obtains the group membership credential $Cert_s^*$ of $NodeID_s^*$ by using $OMJoin$. Subsequently, $\mathcal{A}_I$ chooses two messages (m_0, m_1) of the same length from the plaintext space $|\mathcal{M}|$ and sends them to $\mathcal{Y}$. $\mathcal{Y}$ then randomly signcrypts $m_b (b \in (0,1))$ and sends the result σ^* to $\mathcal{A}_I$ as an answer. Moreover, $\mathcal{Y}$ is not allowed to perform PKE or RPK queries on $NodeID_r^*$ in this phase. When $NodeID_r^* = NodeID_{ISN_i^j}$, $\mathcal{Y}$ executes the normal signcryption process of ECGSH scheme and returns the result to $\mathcal{A}_I$. Otherwise, $\mathcal{Y}$ abandons this challenge.

Phase 2: $\mathcal{A}_I$ executes a finite number of adaptive queries within polynomial bounds, as in *Phase 1*. However, $\mathcal{A}_I$ is not allowed to perform a designcryption query with $NodeID_s^*$ as the sender and $NodeID_r^*$ as the receiver. Moreover, $NodeID_r^*$ cannot be queried by RPK or PKE queries.

Guess: According to the above process, when $\mathcal{A}_I$ wins the game, $\mathcal{Y}$ is able to solve the ECDLP with $MD^* = r_{pd1} g = PASKg - msk \cdot$

$h_3\left(H_{L1}^* + h_2\left(i \parallel j \parallel k, MI_i^j\right), ME_j^k, MD_k\right)g = k^*g.$

- **Adversary $\mathcal{A}_{II}$ of type II**

Setup: Algorithm $\mathcal{Y}$ plays the role of the challenger to complete the game, and adversary $\mathcal{A}_{II}$ initializes node identification via $NodeInfoInit$ and generates public parameters $Gparams = \{p, E, G, h_1, h_2, h_3, h_4\}$ and master key pair (mpk, msk) through $GPInit$ and $GKeyGen$. After these operations, $\mathcal{A}_{II}$ shares the information with $\mathcal{Y}$.

Phase 1, Challenge and Phase 2: These phases are basically the same as those in the interactive game played by the adversary of type I $\mathcal{A}_I$. However, $\mathcal{A}_{II}$ does not perform RPK or PPKE queries in this game.

Probability Analysis: $\mathcal{A}_I$ has a game mode similar to that of $\mathcal{A}_{II}$. When the adversary can makeat most q_δ queries among the $H_\delta(\delta = 1, 2, 3, 4)$ queries, q_{OM} among the $OmJoin$ queries, q_P among the public key queries, q_{paS} among the PPKE queries, q_{reP} among the RPK queries, q_S among the PKE queries, q_{SC} among the signcryption queries and q_{DeSC} among the designcryption queries, we assume that $q_\Sigma = q_{OM} + q_P + q_{paS} + q_S$, where q_P and q_{reP} are zero in the game of $\mathcal{A}_{II}$. Then, the probability that the adversary wins the game is at most $\left(1 - \frac{q_S}{q_\Sigma}\right)\left(1 - \frac{q_{reP}}{q_\Sigma}\right)\left(1 - \frac{q_{paS}}{q_\Sigma}\right)\left(\frac{1}{q_1}\right)\varepsilon$. Therefore, we have the advantage

$$Adv_{IND-ECGSH-CCA2} > \left(1 - \frac{q_S}{q_\Sigma}\right)\left(1 - \frac{q_{reP}}{q_\Sigma}\right)\left(1 - \frac{q_{paS}}{q_\Sigma}\right)\left(\frac{1}{q_1}\right)\varepsilon$$

where q_{reP} and q_P are zero in the game of $\mathcal{A}_{II}$.

C. Nonrepudiation

Theorem 3: In the data transmission process of the IIoT, the ECGSH scheme satisfies nonrepudiation under adaptive chosen plaintext attacks when neither $\mathcal{A}_I$ nor $\mathcal{A}_{II}$ can win the game in polynomial time $\mathcal{P}$ with the advantage $Adv_\mu^{EUF-ECGSH-CMA}$, where $\mu = \mathcal{A}_I, \mathcal{A}_{II}$.

Proof: Assuming that the adversary is able to win the game, there exists an algorithm $\mathcal{Y}$ that can obtain a solution to the ECDLP with the advantage $Adv_{EUF-ECGSH-CMA}$.

- **Adversary $\mathcal{A}_I$ of type I**

Setup: In this phase, algorithm $\mathcal{Y}$ executes system initialization, which is the same as when this phase is performed by the Type I adversary $\mathcal{A}_I$ described in *Theorem 2*.

Phase 1: $\mathcal{A}_I$ executes a finite number of adaptive queries within polynomial bounds, which is the same as when this phase is performed by the Type I adversary $\mathcal{A}_I$ described in *Theorem 2*.

Challenge: After completing *Phase 1* queries, $\mathcal{A}_I$ selects two different nodes $NodeID_s^*$ and $NodeID_r^*$ and obtains the group membership credential $Cert_s^*$ of $NodeID_s^*$ by using $OMJoin$. Subsequently, $\mathcal{A}_I$ chooses a message m^* ($m^* \neq \{m_x, m_{dx}\}$) from the plaintext space $|\mathcal{M}|$ and signcrypts it. Moreover, $\mathcal{A}_I$ is not allowed to perform PKE or RPK queries on $NodeID_s^*$ or a signcryption query with $NodeID_s^*$ as the sender and $NodeID_r^*$ as the receiver in the signcryption process. Then, $\mathcal{A}_I$ sends the signcryption result σ^* and the information of $NodeID_s^*$ to $\mathcal{Y}$. When $NodeID_s^* = NodeID_{ISN_i^j}$, $\mathcal{Y}$ executes the normal designcryption process of the ECGSH scheme. Otherwise, $\mathcal{Y}$ abandons this challenge.

Guess: According to the above process, when $\mathcal{A}_I$ wins the game, $\mathcal{Y}$ is able to solve the ECDLP with $MD^* = r_{pd1}g = PASKg - msk \cdot h_3\left(H_{L1}^* + h_2\left(i \parallel j \parallel k, MI_i^j\right), ME_j^k, MD_k\right)g = k^*g$.

- **Adversary $\mathcal{A}_{II}$ of type II**

Setup: Algorithm $\mathcal{Y}$ executes system initialization, which is the same as when this phase is performed by the Type II adversary $\mathcal{A}_{II}$ described in *Theorem 2*.

Phase 1: $\mathcal{A}_{II}$ executes a finite number of adaptive queries within polynomial bounds, which is the same as when this phase is performed by the Type II adversary $\mathcal{A}_{II}$ described in *Theorem 2*.

Challenge: This phase is essentially the same as that in the interactive game played by the adversary of type I $\mathcal{A}_I$, except that $\mathcal{A}_{II}$ does not perform RPK or PPKE queries in this game.

Probability Analysis: $\mathcal{A}_I$ has a game mode similar to that of $\mathcal{A}_{II}$. When the adversary can make at most q_δ queries among the $H_\delta(\delta = 1, 2, 3, 4)$ queries, q_{OM} among the $OmJoin$ queries, q_P among the public key queries, q_{paS} among the PPKE queries, q_{reP} among the RPK queries, q_S among the PKE queries, q_{SC} among the signcryption queries and q_{DeSC} among the designcryption queries, we assume $q_\Sigma = q_{OM} + q_P + q_{paS} + q_S$, where q_P and q_{reP} is zero in the game of $\mathcal{A}_{II}$. Then, the probability that the adversary wins the game is at most $\left(1 - \frac{q_{paS}q_S}{q_\Sigma^2}\right)\left(1 - \frac{q_{reP}}{q_\Sigma}\right)\left(\frac{1}{q_1}\right)\varepsilon$. Therefore, we have the advantage

$$Adv_{EUF-ECGSH-CMA} > \left(1 - \frac{q_{paS}q_S}{q_\Sigma^2}\right)\left(1 - \frac{q_{reP}}{q_\Sigma}\right)\left(\frac{1}{q_1}\right)\varepsilon$$

where q_{reP} and q_P are zero in the game of $\mathcal{A}_{II}$.

D. Forward security

Theorem 4: In the data transmission process of the IIoT, the ECGSH scheme satisfies forward security when neither $\mathcal{A}_I$ nor $\mathcal{A}_{II}$ can win the game in polynomial time $\mathcal{P}$ with the advantage $Adv_\mu^{ECGSH-FSec}$, where$\mu = \mathcal{A}_I, \mathcal{A}_{II}$.

Proof: Assuming that the adversary is able to win the game, there exists an algorithm $\mathcal{Y}$ that can obtain a solution to the CDH problem with the advantage $Adv_{ECGSH-FSec}$.

- **Adversary $\mathcal{A}_I$ of type I**

Setup: In this phase, algorithm $\mathcal{Y}$ executes system initialization, which is the same as when this phase is performed by the Type I adversary $\mathcal{A}_I$ described in *Theorem 2*.

Phase 1: $\mathcal{A}_I$ executes a finite number of adaptive queries within polynomial bounds, which is the same as when this phase is performed by the Type I adversary $\mathcal{A}_I$ described in *Theorem 2*.

Challenge: After completing the *Phase 1* queries, $\mathcal{A}_I$ selects two different nodes $NodeID_s^*$ and $NodeID_r^*$ and obtains the group membership credential $Cert_s^*$ of $NodeID_s^*$ by using $OMJoin$. Subsequently, $\mathcal{A}_I$chooses two messages (m_0, m_1) of the same length from the plaintext space $|\mathcal{M}|$ and sends them to $\mathcal{Y}$. $\mathcal{Y}$ executes the key update algorithm

and then randomly signcrypts $m_b\,(b \in (0,1))$ with the updated key. Afterwards, $\mathcal{Y}$ sends the result σ^* to $\mathcal{A}_I$ as an answer. Moreover, $\mathcal{Y}$ is not allowed to perform PKE or RPK queries on $NodeID_r^*$ in this phase. When $NodeID_r^* = NodeID_{ISN_i^j}$, $\mathcal{Y}$ executes the normal signcryption process of the ECGSH scheme and returns the result to $\mathcal{A}_I$. Otherwise, $\mathcal{Y}$ abandons this challenge.

Phase 2: $\mathcal{A}_I$ executes a finite number of adaptive queries within polynomial bounds, as in *Phase 1*. However, $\mathcal{A}_I$ is not allowed to perform a designcryption query with $NodeID_s^*$ as the sender and $NodeID_r^*$ as the receiver. Moreover, $NodeID_r^*$ cannot be queried by RPK or PKE queries.

Guess: According to the above process, when $\mathcal{A}_I$ wins the game, $\mathcal{Y}$ is able to solve the CDH problem with $MD^* = r_{pd1}g = PASK^*g - msk^*\tau g = PASKg - msk\tau g + H_{L2}^*\psi_\rho g - \rho H_{L1}^*\tau g = \Lambda + k^*g$.

- **Adversary $\mathcal{A}_{II}$ of type II**

Setup: Algorithm $\mathcal{Y}$ executes system initialization, which is the same as when this phase is performed in the Type II adversary $\mathcal{A}_{II}$ described in *Theorem 2*.

Phase 1, Challenge and Phase 2: These phases are essentially the same as those in the interactive game played by the adversary of type I $\mathcal{A}_I$. However, $\mathcal{A}_{II}$ does not perform RPK or PPKE queries in this game.

Probability Analysis: $\mathcal{A}_I$ has a similar game mode to that of $\mathcal{A}_{II}$. When the adversary can make at most q_δ queries among the $H_\delta(\delta = 1,2,3,4)$ queries, q_{OM} among the $OmJoin$ queries, q_P among the public key queries, q_{paS} among the PPKE queries, q_{reP} among the RPK queries, q_S among the PKE queries, q_{SC} among the signcryption queries and q_{DeSC} among the designcryption queries, we assume that $q_\Sigma = q_{OM} + q_P + q_{paS} + q_S$, where q_P and q_{reP} are zero in the game of $\mathcal{A}_{II}$. Then, the probability that the adversary wins the game is at most $\left(1 - \frac{q_S}{q_\Sigma}\right)\left(1 - \frac{q_{reP}}{q_\Sigma}\right)\left(1 - \frac{q_{paS}}{q_\Sigma}\right)\left(\frac{1}{q_1}\right)\left(\frac{1}{q_2}\right)\varepsilon$. Therefore, we have the advantage

$$Adv_{ECGSH-FSec} > \left(1 - \frac{q_S}{q_\Sigma}\right)\left(1 - \frac{q_{reP}}{q_\Sigma}\right)\left(1 - \frac{q_{paS}}{q_\Sigma}\right)\left(\frac{1}{q_1}\right)\left(\frac{1}{q_2}\right)\varepsilon$$

where q_{reP} and q_P are zero in the game of $\mathcal{A}_{II}$.

E. Attack resistance

In the perception and control domain of the IIoT, the variety of devices and the complex network architectures and communication protocols expose the system to various security threats, including replay attacks, man-in-the-middle attacks, and eavesdropping. These threats not only endanger the security and integrity of data but also pose significant risks to operational continuity.

Robust security measures are therefore critical. According to *Theorem 2* and *Theorem 3*, the ECGSH scheme is specifically engineered to offer strong resistance against such attacks. It effectively shields against selective ciphertext attacks, ensuring data confidentiality even under potential man-in-the-middle attacks. It further guarantees that the data cannot be tampered with or forged, thus significantly bolstering defences against eavesdropping and unauthorized data alteration. Additionally, as outlined in *Theorem 4*, the ECGSH scheme provides strong protection for historical transmission data against the threat of replay attacks, even if the encryption key is later compromised.

Overall, the ECGSH scheme is notable for its robust attack resistance, providing a secure framework that addresses multiple layers of potential threats within the IIoT environment.