SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2025, Vol. 68, Iss. 8, 182303:1–182303:21
https://doi.org/10.1007/s11432-024-4269-5

Secure data transmission and classification for digital twin

Weizheng WANG[1], Dequan XU[2*], Zhusen LIU[3], Qipeng XIE[4], Chunhua SU[5] & Changgen PENG[2]

[1]*Department of Computer Science, City University of Hong Kong, Hong Kong 999077, China*
[2]*Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China*
[3]*Hangzhou Innovation Institute of Beihang University, Hangzhou 311121, China*
[4]*Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong 999077, China*
[5]*Division of Computer Science, University of Aizu, Fukushima 965-8580, Japan*

Received 30 August 2024/Revised 12 December 2024/Accepted 3 January 2025/Published online 10 July 2025

Abstract In Industry 4.0, digital twin (DT) technology plays an increasingly vital role in enabling intelligent and automated manufacturing and management. However, the utilization of DT in Industry 4.0 environments raises significant security concerns, particularly regarding data transmission and protection. This underscores the critical need for comprehensive and robust security frameworks specifically designed for data transmission and classification in DT-based systems. In this paper, we present a novel secure solution based on the purified Paillier cryptosystem to handle sensitive and categorical information through specialized verification keys and aggregation mechanisms. Our framework implements a three-layer architecture: the device layer uses trusted authority (TA) issued parameters to generate encrypted data types, content, and signatures; the edge layer employs verification keys to filter and aggregate required data types; and the DT layer performs final assessment and decryption. Additionally, we introduce an LSTM-RNN-based reverse data control strategy for DT network formulation and anomaly detection. Through extensive evaluation and testing, we demonstrate both the security robustness and performance efficiency of our proposed approach in realistic deployment scenarios.

Keywords digital twin (DT), secure data classification and transmission, security and privacy

Citation Wang W Z, Xu D Q, Liu Z S, et al. Secure data transmission and classification for digital twin. Sci China Inf Sci, 2025, 68(8): 182303, https://doi.org/10.1007/s11432-024-4269-5

1 Introduction

The industrial sector has experienced a profound digital transformation, evolving from the standalone physical systems of Industry 3.0 to the integrated cyber-physical systems (CPS) of Industry 4.0. In this new paradigm, CPS connects physical devices to external Internet services, enabling data-driven control through real-time sensing and analysis. This evolution has been powered by emerging technologies including the Industrial Internet of Things (IIoT), mobile edge computing (MEC), and artificial intelligence (AI), collectively enabling fully automated distributed control and service orientation [1]. Digital twin (DT) technology represents a significant advancement beyond traditional CPS, offering enhanced capabilities to map physical processes into virtual space for real-time monitoring, prediction, and optimization in Industry 4.0 [2]. A key distinguishing feature of DT is its bidirectional data flow, where modifications to the digital object directly influence physical world execution, thereby creating a continuous feedback loop for performance optimization. The concept of DT traces its origins to NASA's Apollo program in the 1960s, where two identical spacecraft were built — one launched into space and another kept on Earth to mirror the spacecraft's real-time state, enabling scientists to monitor and adjust equipment parameters [3]. This pioneering implementation laid the groundwork for modern DT applications. In 2012, NASA and the U.S. Air Force formalized the DT definition, establishing it as a strategic emerging technology [4]. Today, DT applications have proliferated across various sectors. In supply chain management, DT systems analyze traffic patterns and topographic data to optimize distribution routes [5].

* Corresponding author (email: dqxu@gyu.edu.cn)

Manufacturing applications enable product customization through virtual modeling of external surfaces and internal configurations [6]. In healthcare, DT technology models patient genome codes and lifestyle factors to enable personalized treatment approaches [7]. The global DT market, valued at 7.48 billion dollars in 2021, is projected to maintain a compound annual growth rate of 39.1% over the next decade[1)]. Modern DT architecture typically comprises three layers: the device layer (consisting of physical actuators and sensors), the edge layer (providing intermediary data filtering and aggregation), and the DT layer (assembling and managing the virtual network). This architecture supports sophisticated retroregulation mechanisms, where virtual modifications trigger corresponding physical changes. For instance, in smart city applications, DT systems model electricity loads across Internet of Things (IoT) devices to optimize power distribution and rapidly identify anomalies, enabling proactive maintenance and efficient resource allocation.

Although the popularity of research concerned with DT and corresponding applications has been rising, some security and privacy issues have not been well considered in the traditional three-layer DT network architecture, which mainly comes from data dissemination and monitor parts [8]. Regarding the data dissemination part, the edge layer first needs to collect operating data from devices in its region. However, in this step, once the transmitted data is unencrypted, the adversary may intercept or even tamper with the uploaded message [9]. Once large amounts of dirty data are generated, system reliability and effectiveness cannot be guaranteed. Besides, depending on the individual requirements of the DT network, uncategorized device data will slow down data processing and transmission rate in the DT layer. Even if the devices are not compromised, some internal exceptions may happen, leading to abnormal operational parameters received by DT. Hence, for the monitoring phase, the DT needs to keep watch for device status and locate the error-prone devices in time. Based on the above-mentioned concerns, in this paper, we propose a secure data transmission and classification framework for DT, composed of a downlink and uplink dual communication process. In the uplink transmission period, the device first utilizes the system parameters to handle original data, which results in data type ciphertexts, encrypted data, and signatures. When the edge layer receives this information, the data type ciphertext selection algorithm will be applied to the unqualified data filter. Moreover, we utilize the purified Paillier cryptosystem that can aggregate specific types of data and signatures in a privacy-preserving way for the edge layer. Finally, the DT layer can verify and decrypt device operational data. In the downlink process, the DT layer utilizes the historical device data to generate a long short-term memory (LSTM) plus recurrent neural network (RNN)-based data trend prediction model, which can analyze and compare the real-time device execution data for error device report. In this paper, we propose a comprehensive security framework for data transmission and classification in DT environments. The main contributions of this article are as follows.

(1) We first propose a secure data transmission and classification framework for DT, which can select and aggregate ciphertexts and signatures for efficient transmission.

(2) We utilize the purified Paillier cryptosystem to design a novel cryptographic protocol that enables secure data type selection under ciphertexts.

(3) We design a data monitor model that can predict and analyze the data trend for the device layer, which then can help DT find out the error devices.

(4) We conduct the comprehensive security analysis and performance evaluation that verify the security and effectiveness of our proposed method for data transmission and monitor in DT.

The remainder of this article is organized as follows. In Section 2, we discuss history development, recent research and applications of DT. In Section 3, we introduce some cryptographic primitives such as purified Paillier cryptosystem, bilinear pairings and computational Diffie-Hellman (CDH) assumption for our scheme construction. In Section 4, we briefly introduce our proposed secure data transmission and classification framework for DT, threat model, design goal, schemes outlines. The security model and definition for this work are presented in Section 5. In Section 6, we illustrate the detailed proposed secure DT scheme step by step. The security for our proposed protocol is analyzed in Section 7. In Section 8, we evaluate the performance of our proposed scheme in terms of computational/communication overhead, along with practical case studies. Moreover, a case study for reverse regulation (i.e., data monitor) is also given in this section. In Section 9, we benchmark our framework against existing DT security solutions. Finally, we conclude this article and give future directions in Section 10.

1) https://www.grandviewresearch.com/industry-analysis/digital-twin-market.

2 Related approaches

In 2002, Grieves [10] first proposed the prototype of DT conception in his presentation called "Conceptual ideal for product life-cycle management (PLM)". In 2012, NASA and U.S. Air Force [4] formally published a report for DT, which illustrates the paradigm for future air force vehicle design. Several years later in 2017, Grieves and Vickers [11] gave a comprehensive definition for DT, which contains three major parts: the digital object in the virtual space, the physical object in the real environment, and the data link between these two objects. Simultaneously, Erikstad [12] illustrated the fundamental constitution of the DT network and compared its solution with AI. Following this, a series of DT-based research and applications have sprung up. For the DT-based IoT, Riemer [13] established semantic and lightweight DT models that simulate sensors and data in a graphic processing pipeline. Steinmetz et al. [14] stated basic concepts for DT modeling under the CPS circumstance, whose case study also verifies the feasibility and effectiveness of mapping IoT devices and attributes into the virtual space. Sleuters et al. [15] applied DT to large-scale distributed IoT systems, including a smart office light system. The experiment results show the DT-based solutions can conduct anomaly detection and reason in a root-cause analysis. Song et al. [16] illustrated the IEEE 1451 DT that emulates each mode of a real sensor such as success and failure. Moreover, a federated experiment for these two statuses is described and analyzed in detail. Eckhart et al. [17] proposed a novel conception called CPS twinning that can generate virtual environments from a real industrial scenario. Considering the potential attacks, security modules are appended for this framework, which has been demonstrated for man-in-the-middle (MITM) attack defense in motor speed control.

For DT-based manufacturing and industry, Bao et al. [18] developed product DT, process DT and operation DT in the manufacturing scene that creates replicas of the state and execution of the objects in the real world. Then the derived optimized solutions for the manufacturing process in the virtual world can be returned to adjust the operations in the physical world. Kritzler et al. [19] introduced a virtual twin that can implement the 3D presentation of modern factories. This system predicts possible emergencies and guarantees the smooth execution of factories. Uhlemann et al. [20] compared DT and value-stream mapping (VSM). The analysis result shows DT can achieve better real-time data acquisition and simulation performance. For the healthcare field, Tao et al. [21] implemented a prognostics and health management (PHM) system in DT. Based on a case study about gearbox prognosis, the DT-empowered PHM can significantly improve prognosis accuracy. To alleviate the existing issues of surveillance and alarm for the elderly, Liu et al. [22] proposed a DT healthcare (DTH) implemented on the cloud. Furthermore, an electrocardiogram device accessed to the DTH demonstrated the two features mentioned above. Karakra et al. [7] utilized the DT to simulate hospital services that need to serve patients in a daily route and investigated whether the modification can influence the normal pathway for patients. For the DT-based logistics, Korth et al. [23] proposed a logistics management system based on the DT which collects the data from true surrounding environments. The combination of the logistics system and simulation logic relieves the pressure for logistics task modeling. Abideen et al. [24] combined the DT with reinforcement learning to propose an operational framework for logistic and supply chain. Lee et al. [5] established a DT framework for the supply chain, which can forecast the unknown risks and delivery time for real-time logistic simulation.

While existing research has made significant progress in DT applications across various domains including IoT, manufacturing, healthcare, and logistics, several critical limitations remain unaddressed in current DT security frameworks.

(1) Most existing studies focus on functional aspects of DT while lacking comprehensive security mechanisms, particularly in data transmission and classification.

(2) Current security solutions do not adequately address the challenge of efficient encrypted data aggregation at the edge layer, which is crucial for large-scale DT deployments.

(3) Existing frameworks lack effective mechanisms for monitoring and detecting anomalous devices in real-time while maintaining data privacy.

(4) The integration of secure data classification with privacy-preserving data transmission remains largely unexplored in DT contexts.

To address these limitations, we propose a secure data transmission and classification framework based on DT specifications, which not only protects against potential attackers but also enables efficient data processing and anomaly detection while preserving privacy.

3 Preliminaries

This section introduces the purified Paillier cryptosystem, bilinear pairings, and CDH assumption for our scheme construction.

3.1 Purified Paillier cryptosystem

Paillier cryptosystem is a partially homomorphic encryption scheme relying on composite residual classes. Moreover, homomorphic features enable users to perform mathematical or rational operations on the encrypted data. In this work, we modify some parameters of the Paillier cryptosystem for supporting encrypted data type selection. The detailed description is given as follows.

(1) **Key generation.** We select p' and q' of two large prime numbers, and $a \in Z_{N^2}^*$, calculate $p = 2p' + 1$, $q = 2q' + 1$, $N = pq$, $g = -a^{2N} \mod N^2$. Then, we pick up g as a generator of order $(p-1)(q-1)/2$, $\theta \in [1, N/4]$ as a private key sk and then calculate the corresponding public key as $\mathrm{pk} = (N, g, h)$, where $h = g^{\theta} \mod N^2$.

(2) **Encryption.** Suppose $m \in Z_N$ is a plaintext message and $r \in [1, N/4]$ is a random number, the ciphertext $C = [m]_{\mathrm{pk}} = \{T_{i,1}, T_{i,2}\}$ is computed under pk, where $T_{i,1} = g^r \mod N^2$, $T_{i,2} = h^r \cdot (1 + m \cdot N) \mod N^2$.

(3) **Decryption.** Let C be a ciphertext with pk encryption. The plaintext message $m = L(T_{i,2}/T_{i,1}^{\theta} \mod N^2)$ can be recovered by using private key $\mathrm{sk} = \theta$, where $L(x) = \frac{x-1 \mod N^2}{N}$.

(4) **Homomorphic property.** Let $[m_1]_{\mathrm{pk}}$ and $[m_2]_{\mathrm{pk}}$ be two ciphertexts encrypted the same public key pk for the purified homomorphic cryptosystem [25], it has the following homomorphic properties:

$$D_{\mathrm{sk}}\left([m_1]_{\mathrm{pk}} \cdot [m_2]_{\mathrm{pk}}\right) \equiv m_1 + m_2 \bmod N, \tag{1}$$

$$D_{\mathrm{sk}}\left([m_1]_{\mathrm{pk}}^{Y}\right) \equiv Y m_1 \bmod N. \tag{2}$$

3.2 Bilinear pairings

Assume two cyclic groups $\mathbb{G}, \mathbb{G}_{\mathbb{T}}$, whose prime order and generator are q and g, respectively. The bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathbb{T}}$ that satisfies the three following properties.

(1) **Bilinearity.**

$\forall g_1, g_2 \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$,

$\forall a, b, y \in \mathbb{G}$, $e(a \cdot b, y) = e(a, y) \cdot e(b, y)$,

$\forall a, b, y \in \mathbb{G}$, $e(y, a \cdot b) = e(y, a) \cdot e(y, b)$.

(2) **non-degeneracy.** $\exists g_1, g_2$, $e(g_1, g_2) \neq 1$.

(3) **Computability.** $\forall g_1, g_2 \in \mathbb{G}$, $e(g_1, g_2)$ is computable.

3.3 CDH assumption

Given $g, g^a, g^b \in \mathbb{G}$ where $a, b \in \{0, \ldots, q-1\}$, it is computationally infeasible to compute the value g^{ab}.

4 System model

In this section, we give a brief illustration of our proposed secure data transmission and classification framework and define the threat model and design goal for DT. The framework for our scheme can refer to Figure 1.

4.1 System overview

Our proposed DT network architecture is a three-tier architecture (i.e., DT layer, edge layer and IoT device layer) and contains four entities: trust authority (TA), a group of heterogeneous IoT devices, the deployed edge layers at the network edges for data type filter and aggregation, and some DT applications run on cloud platform for environment simulation. The comprehensive description for these four parties is illustrated as follows.

(1) TA is responsible for system initialization, parameter generation and public/secret key distribution for the remaining parts.

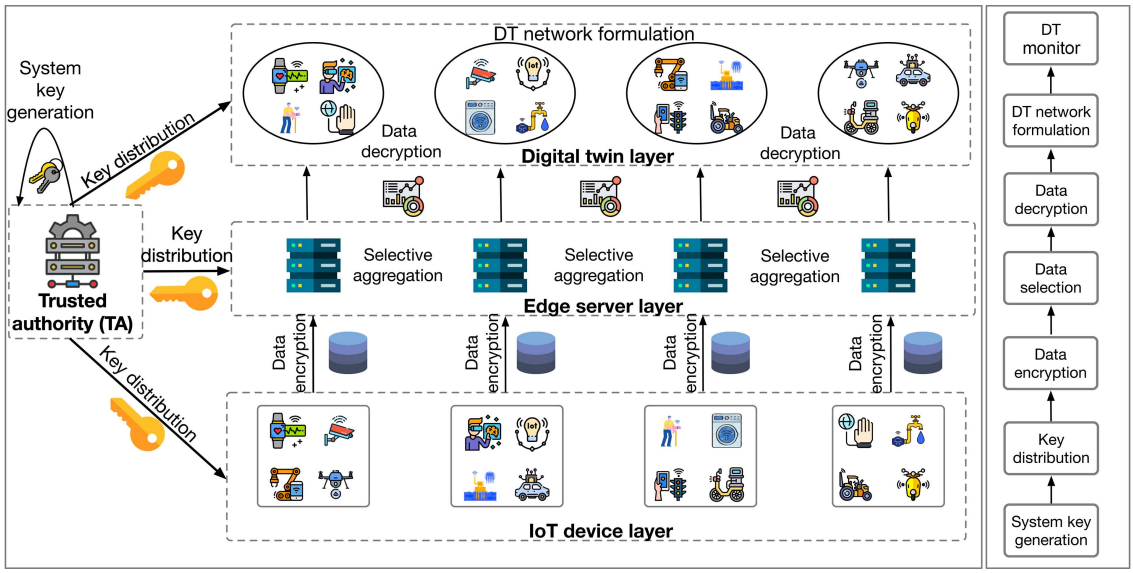

Figure 1 (Color online) Secure data transmission and classification framework for DT.

(2) IoT device can be any terminal device with embedded sensors and communication module, and periodically provide its execution parameters and some sensing data to the DT for network simulation. However, given the wide range of devices, the related data should be uploaded to the edge with proximity for data aggregation first. Moreover, IoT device generates some ciphertexts and signatures to guarantee privacy and integrity of device data.

(3) Edge is a powerful computer that locates at the network edge. Upon receiving ciphertexts regarding data type and content from the IoT devices, edge conducts some computation to filter unrequired data types and then aggregates encrypted data and signatures that satisfy the requirement. Finally, fitted data and signatures are transmitted to the DT for further decryption and verification.

(4) DT is a computer program that gathers the data from IoT devices in the real world and then creates simulations for process prediction. Firstly, DT receives the same type of data transmitted from the edges. Then DT utilizes the private keys issued by the TA for data verification and decryption. Finally, DT forms a data trend prediction model that can find out the error-prone devices. Note that the falsified devices will be notified to the related edges and prohibited to participate in the DT network formulation in the next turn.

4.2 Threat model

(1) The TA is considered a fully trusted party in our DT system, which is responsible for system initialization and key distribution. Moreover, the communication between TA and other parts is conducted in a secure channel.

(2) The IoT device is regarded as unreliable, which may provide fabricated sensing data and interfere with the ultimate DT network formulation.

(3) The edge server is curious-but-honest. The edge may be interested in the content transmitted by the IoT device, but cannot collude with other IoT devices and DT to compromise real data.

(4) The DT is reliable and generates a simulated network according to its receiving data honestly.

More detailed, an active adversary $\mathcal{A}^*$ is introduced in our scheme. The goal of $\mathcal{A}^*$ is to infer the original data of the challenged IoT device and aggregation result of challenged edge server in the following ways: (1) $\mathcal{A}^*$ may eavesdrop all communications of the sensing report process to obtain the encrypted data; (2) $\mathcal{A}^*$ may compromise one or more IoT devices except from the challenged IoT device to guess the challenged value; (3) $\mathcal{A}^*$ may compromise the challenged edge server to guess the original value of all aggregated ciphertexts; (4) $\mathcal{A}^*$ may compromise the IoT devices and the edge to provide fabricated sensing data.

4.3 Design goal

(1) **Data privacy preservation.** The data privacy must be guaranteed during the uploading process, so the transmitted data needs to be encrypted.

(2) **Selective data aggregation.** The encrypted data needs to be filtered and aggregated for the same type in the edge layer without any security breach.

(3) **Data decryption and verification.** Upon receiving the device data, DT needs to restore the original content depending on the private keys and verify the correctness of the data.

(4) **Data quality monitor.** DT needs to monitor the sensing data value during the network life cycle and block the fabricated device for the edge layer.

4.4 Scheme outline

In this subsection, we illustrate the outlines of our scheme which consist of six polynomial time algorithms: system key generation (**SKeyGen**), key distribution (**KeyDist**), device layer data encryption (**DLDataEnc**), edge layer selective aggregation (**ELSeleAggr**), twin layer data decryption (**TLDataDec**), and DT network monitor (**DTNMoni**).

(1) **SKeyGen**$(k) \to$ (Params)**.** It is executed by the trusted authority (TA). Input a security parameter k to the TA, which outputs the system public parameter Params $= \{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, P\}$.

(2) **KeyDist**$(\mathrm{ID}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \{\mathrm{TYPE}_1, \ldots, \mathrm{TYPE}_i\}, P) \to (\mathrm{PK}_{\mathrm{DT}_i}, \mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i)$**.** It is still run by the TA. It inputs identities of DT_i and IoT_i as $\mathrm{ID}_{\mathrm{DT}_i}$ and $\mathrm{ID}_{\mathrm{IoT}_i}$, the i data types of DT_i, a large prime P. Then TA outputs public key of DT_i as $\mathrm{PK}_{\mathrm{DT}_i}$, secret key of DT_i as $\mathrm{SK}_{\mathrm{DT}_i}$, secret key of IoT_i as $\mathrm{SK}_{\mathrm{IoT}_i}$, the ith encrypted data type ciphertexts as $\mathrm{VK}_{\mathrm{TYPE}_i}$. TA transmits them to corresponding entities in a secure channel.

(3) **DLDataEnc**$(\mathrm{SK}_{\mathrm{IoT}_i}, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{TYPE}_i, m_i) \to (c_i, C_i, \sigma_i)$**.** It is executed by the IoT device. Input secret key of IoT_i as $\mathrm{SK}_{\mathrm{IoT}_i}$, public key of DT_i as $\mathrm{PK}_{\mathrm{DT}_i}$, required data type TYPE_i, and message m_i. IoT_i outputs the ciphertext c_i, ciphertext on encrypted data type C_i, signature σ_i.

(4) **ELSeleAggr**$(\mathrm{TYPE}_i, \mathrm{VK}_{\mathrm{TYPE}_i}) \to (c_{\mathrm{all}}, \sigma_{\mathrm{all}})$**.** This step is executed by the edge server. It inputs required data type TYPE_i and the ith data type verification key $\mathrm{VK}_{\mathrm{TYPE}_i}$ and outputs the aggregated ciphertext c_{all} and signature σ_{all}.

(5) **TLDataDec**$(\mathrm{SK}_{\mathrm{DT}_i}) \to (m)$**.** This step is executed by the server in the DT layer. It inputs the secret key of DT_i as $\mathrm{SK}_{\mathrm{DT}_i}$, and outputs the aggregated result of each data type $m = \sum m_i P^i$.

(6) **DTNMoni**$(\mathrm{Dataset}_{\mathrm{IoT}_i}) \to (\mathrm{IoT}_i)$**.** Upon inputting the historical execution data $\mathrm{Dataset}_{\mathrm{IoT}_i}$ for the ith IoT device, the DT can generate a data trend prediction model that outputs the identity of the ith error device $\mathrm{ID}_{\mathrm{IoT}_i}$.

5 Security model

In this section, we define the security model for our proposed DT transmission and classification framework.

5.1 Security definition

Ciphertext indistinguishability. When the encrypted data is uploaded to the edge server from the IoT device, it is required that the encrypted data should not leak any information about its underlying original data. It is worth noting that even though the edge can select and aggregate multiple ciphertexts, the final aggregation result can only be accessed by the specific DT server. Given multiple IoT devices existing in our system model, to capture the security of the chosen plaintext attack, the indistinguishability game under the adaptive chosen plaintext attack can be established for our scheme. The adversary chooses two distinct plaintexts and sends them to the challenger. The challenger randomly chooses one of them for encryption and returns the ciphertext. Then, the adversary tries to guess which one is selected and encrypted by the challenger. The game of indistinguishability under ciphertext indistinguishability (IND-CI) is an interactive game between the adversary $\mathcal{A}$ and challenger $\mathcal{C}$ as follows.

Initialization. Challenger $\mathcal{C}$ neends to first execute the initial algorithm **SKeyGen**$(k) \to (\{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, P\})$, and the **KeyDist**$(\mathrm{ID}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \{\mathrm{TYPE}_1, \ldots, \mathrm{TYPE}_i\}, P) \to (\{\mathrm{PK}_{\mathrm{DT}_i},$

$\mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i\}$). Then, $\mathcal{C}$ publishes $\{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i\}$ to adversary $\mathcal{A}$ and keeps ($\mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}$) secret.

Query. Adversary $\mathcal{A}$ can adaptively request the ciphertext c for any input plaintext m from $\mathcal{C}$ in a polynomial time. Then $\mathcal{C}$ encrypts the m as c and returns it to $\mathcal{A}$.

Challenge. Adversary $\mathcal{A}$ chooses two distinct plaintexts m_1 and $m_2 \in Z_N$, which will be sent to the $\mathcal{C}$. Both the m_1 and m_2 have to fulfill the criteria with equal length and cannot be any plaintext m required in the **Query**. After receiving the m_1 and m_2, $\mathcal{C}$ chooses $b \in_R \{0,1\}$ and performs **DLDataEnc**$(\mathrm{SK}_{\mathrm{IoT}_i}, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{TYPE}_i, m_b) \to (c_b, C_i, \sigma_i)$. Finally, the challenged ciphertext c_b is returned to $\mathcal{A}$.

Guess. Adversary $\mathcal{A}$ outputs b' as its guess, then wins the game if $b' = b$.

Definition 1. Our scheme satisfies IND-CI, if the advantage

$$\mathrm{Adv}^{\text{IND-CI}}_{\text{DT-IoT},\mathcal{A}}(k) = \left|\Pr[b' = b] - \frac{1}{2}\right| \tag{3}$$

for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ in the above game is negligible.

Signature privacy. When an IoT device transmits encrypted data to the edge server, the corresponding signatures are also attached. Thus, it is required that the signatures should not leak any information about the underlying content. Furthermore, even though the edge server can select and aggregate multiple signatures, the final aggregation result also cannot leak any information about the underlying data content. Due to the multiple IoT devices in our system model, to capture the security of the chosen plaintext attack, the indistinguishability game under the adaptive chosen plaintext attack can be given as follows: the adversary $\mathcal{A}$ chooses two distinct plaintext m_1 and m_2, which are sent to the challenger $\mathcal{C}$. Then $\mathcal{C}$ randomly chooses one m_b to generate signature σ_b and returns it. Finally, adversary $\mathcal{A}$ tries to guess the chosen plaintext m_b. The game of indistinguishability under signature indistinguishability (IND-SI) is an interactive game between the $\mathcal{A}$ and $\mathcal{C}$ as follows.

Initialization. Challenger $\mathcal{C}$ executes the initial algorithm **SKeyGen**$(k) \to (\{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(x), N, P\})$, and the **KeyDist**$(\mathrm{ID}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \{\mathrm{TYPE}_1, \ldots, \mathrm{TYPE}_i\}, P) \to (\{\mathrm{PK}_{\mathrm{DT}_i}, \mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i\})$. Then, $\mathcal{C}$ publishes $\{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i\}$ to adversary $\mathcal{A}$ and keeps ($\mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}$) secret.

Query. Adversary $\mathcal{A}$ can adaptively request the signature σ for any plaintext m from $\mathcal{C}$ in a polynomial time. The $\mathcal{C}$ responds with σ generated by the input m and returns it to $\mathcal{A}$.

Challenge. Adversary $\mathcal{A}$ chooses two distinct plaintext $m_1, m_2 \in Z_N$. Both the m_1 and m_2 own the same length and cannot equal to the previous m in the **Query**. After receiving the m_1, m_2, $\mathcal{C}$ chooses $b \in_R \{0,1\}$ and performs **DLDataEnc**$(\mathrm{SK}_{\mathrm{IoT}_i}, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{TYPE}_i, D_b) \to (c_i, C_i, \sigma_b)$. The challenged signature c_b will be responded to $\mathcal{A}$.

Guess. Adversary $\mathcal{A}$ outputs b' as its guess, then wins the game if $b' = b$.

Definition 2. We say that our scheme satisfies IND-SI, if for any PPT adversary $\mathcal{A}$ in the above game, the advantage

$$\mathrm{Adv}^{\text{IND-SI}}_{\text{DT-IoT},\mathcal{A}}(k) = \left|\Pr[b' = b] - \frac{1}{2}\right| \tag{4}$$

is negligible.

Type privacy. During device data transmission, the encrypted type should not leak any sensitive information about the data type. The remaining definition for **Type privacy** is similar with **Signature privacy**. The game of indistinguishability under type ciphertext indistinguishability (IND-TCI) is an interactive game between the $\mathcal{A}$ and $\mathcal{C}$ as follows.

Initialization. Challenger $\mathcal{C}$ executes the initial algorithm **SKeyGen**$(k) \to (\{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, P\})$, and the **KeyDist**$(\mathrm{ID}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \{\mathrm{TYPE}_1, \ldots, \mathrm{TYPE}_i\}, P) \to (\{\mathrm{PK}_{\mathrm{DT}_i}, \mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i\})$. Then, $\mathcal{C}$ publishes $\{g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{VK}_{\mathrm{TYPE}_i}, P^i\}$ to adversary $\mathcal{A}$ and keeps ($\mathrm{SK}_{\mathrm{DT}_i/\mathrm{IoT}_i}$) secret.

Query. Adversary $\mathcal{A}$ can adaptively request the ciphertext of data types C for any plaintext m from $\mathcal{C}$ in polynomial time. The $\mathcal{C}$ responds with C derived from the m and returns it to $\mathcal{A}$.

Challenge. Adversary $\mathcal{A}$ chooses two distinct integers m_1 and $m_2 \in Z_N$, whose types are TYPE_1 and TYPE_2. After receiving the m_1, m_2, TYPE_1, TYPE_2, $\mathcal{C}$ randomly chooses $b \in_R \{0,1\}$, and then performs **DLDataEnc**$(\mathrm{SK}_{\mathrm{IoT}_i}, \mathrm{PK}_{\mathrm{DT}_i}, \mathrm{TYPE}_b, m_b) \to (c_i, C_b, \sigma_i)$. Finally, the challenged ciphertext of the data types C_b is returned to $\mathcal{A}$.

Guess. Adversary $\mathcal{A}$ outputs b' as its guess, then wins the game if $b' = b$.

Definition 3. We say that our scheme satisfies IND-TCI, if for any PPT adversary $\mathcal{A}$ in the above game, the advantage

$$\mathrm{Adv}^{\text{IND-TCI}}_{\text{DT-IoT},\mathcal{A}}(k) = \left|\Pr[b' = b] - \frac{1}{2}\right| \tag{5}$$

is negligible.

5.2 Simulation-based security definition

To prove the above-mentioned security definition, we introduce the simulation-based security model when non-colluding semi-honest adversaries exist.

Assume an IoT device D_a and an edge server E_b are in our scheme. Let $\mathcal{P} = (D_a, E_b)$ represent all participants in the protocol execution interval, and $\mathcal{A}_{D_a}, \mathcal{A}_{E_b}$ represent two adversaries of corrupted D_a, E_b.

In a real word, entity D_a runs with given x, y, z as input (i.e., with additional auxiliary w_x, w_y, w_z as input), while E_b runs with receiving w_1, w_2, w_3 as input. Let $\mathcal{H} \subset \mathcal{P}$ represents the set of honest entities. When P is honest (i.e., $P \in \mathcal{H}$), out_p is the output of entity P. When P is corrupted (i.e., $P \in \mathcal{P}\backslash\mathcal{H}$), out^*_p is the view of entity P in the protocol Π running interval.

The protocol Π is run for each $P^* \in \mathcal{P}$, when entities $\mathcal{P} = (D_a, E_b)$ and adversaries $\mathcal{A} = (\mathcal{A}_{D_a}, \mathcal{A}_{E_b})$ exist. Here, we output the partial view of P^* as follows:

$$\mathrm{REAL}^{P^*}_{\Pi,\mathcal{A},\mathcal{P},w}(x, y, z) = \{\mathrm{out}_p\} \cup \{\mathrm{out}^*_P : P \in \mathcal{P}\backslash\mathcal{H}\}. \tag{6}$$

In an ideal world, the ideal functionality $\boldsymbol{f}$ denotes a trusted entity responsible for interaction with all other entities. The challenge D_a uploads x, y, z to $\boldsymbol{f}$. Suppose one of (x, y, z) is $\perp$, $\boldsymbol{f}$ returns $\perp$. Otherwise, $\boldsymbol{f}$ responds $\boldsymbol{f}(x, y, z)$ to the D_a. Let $\mathcal{H} \subset \mathcal{P}$ be the set of honest entities. When P is honest (i.e., $P \in \mathcal{H}$), out_p is the output returned by $\boldsymbol{f}$ to entity P. When P is corrupted (i.e., $P \in \mathcal{P}\backslash\mathcal{H}$), out^*_p is the output of some random value from P in the protocol Π execution interval.

The protocol Π is run for each $P^* \in \mathcal{P}$ in an ideal world. When entities $\mathcal{P} = (D_a, E_b)$ and the independent simulators $\mathcal{S} = (\mathcal{S}_{D_a}, \mathcal{S}_{E_b})$ are present. Here, we output the partial view of P^* as follows:

$$\mathrm{IDEAL}^{P^*}_{\boldsymbol{f},\mathcal{S},\mathcal{P},w}(x, y, z) = \{\mathrm{out}_p\} \cup \{\mathrm{out}^*_P : P \in \mathcal{P}\backslash\mathcal{H}\}. \tag{7}$$

Informally, the protocol Π is secure in non-colluding semi-honest adversaries, and an ideal functionality $\boldsymbol{f}$ in the ideal world can partially be emulated in the real world.

Definition 4. Let $\boldsymbol{f}$ be a deterministic function among entities $\mathcal{P} = (D_a, E_b)$ and Π be a protocol among entities $\mathcal{P} = (D_a, E_b)$. When $\mathcal{H} \subset \mathcal{P}$ represents the subset of honest entities, let $\mathcal{H} = \emptyset$ (i.e., each entity $P \in \mathcal{P}$ denotes semi-honest non-colluding entities). We say that Π can securely emulate $\boldsymbol{f}$ if there exists a set $\mathrm{Sim} = (\mathrm{Sim}_{D_a}, \mathrm{Sim}_{E_b})$ of PPT mutations (e.g., $S_{D_a} = \mathrm{Sim}_{D_a}(\mathcal{A}_{D_a}), S_{E_b} = \mathrm{Sim}_{E_b}(\mathcal{A}_{E_b})$). Hence, for all semi-honest non-colluding adversaries $\mathcal{A} = (\mathcal{A}_{D_a}, \mathcal{A}_{E_b})$, for all $x, y \in Z_N$ and auxiliary $z \in Z_N$, and all entities $P \in \mathcal{P}$, the following equation holds:

$$\mathrm{REAL}^{P^*}_{\Pi,\mathcal{A},\mathcal{P},w}(x, y, z) \overset{\mathrm{c}}{\approx} \mathrm{IDEAL}^{P^*}_{\boldsymbol{f},\mathcal{S},\mathcal{P},w}(x, y, z), \tag{8}$$

where $\overset{\mathrm{c}}{\approx}$ represents computational indistinguishability.

6 Proposed scheme construction

6.1 Overview of construction steps

In this subsection, we present the detailed construction of our secure data transmission and classification scheme. The scheme consists of six major steps, each serving a specific purpose in achieving our security and functionality goals.

• **System key generation.** This step establishes the cryptographic foundation of our framework by generating necessary system parameters and keys, ensuring the security of subsequent operations.

• **Key distribution.** This phase securely distributes different types of keys to various entities (DT, IoT devices, and edge servers), establishing trust relationships and enabling secure communication channels.

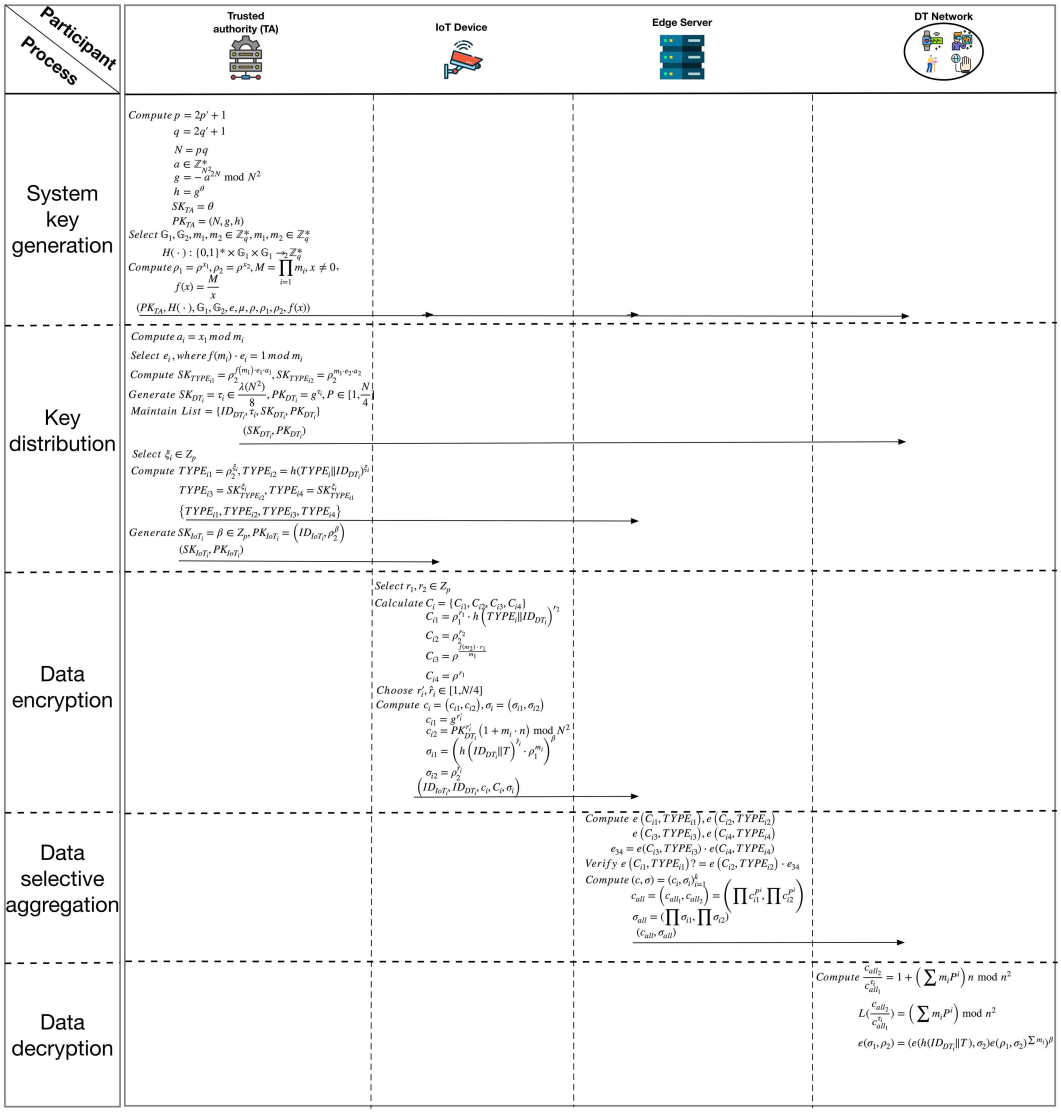

Figure 2 (Color online) Proposed protocol for secure data transmission and classification for DT.

• **Device layer data encryption.** This step enables IoT devices to securely encrypt their data and data types, ensuring data privacy during transmission while allowing for selective processing at the edge layer.

• **Edge layer data selective aggregation.** This phase allows edge servers to efficiently filter and aggregate encrypted data based on type verification, reducing communication overhead while maintaining data privacy.

• **Twin layer data decryption.** This step enables the DT layer to verify and decrypt the aggregated data, ensuring data integrity and confidentiality throughout the process.

• **DT network monitor.** This final phase implements real-time monitoring and anomaly detection, enabling the DT to maintain network health and security.

6.2 Detailed construction process

The detailed construction of each step is outlined below. Note that Table 1 lists the notation used in this subsection. The interaction process for our protocol can refer to Figure 2.

• **System key generation.**

(1) Firstly, given a security parameter k, TA randomly selects two large prime numbers $p', q' \in \mathbb{Z}_N^*$, and figures out $p = 2p' + 1$, $q = 2q' + 1$, $N = pq$. Then, TA randomly picks $a \in \mathbb{Z}_{N^2}^*$ and computes

Table 1 Notations.

Symbol	Description
p', q', μ, m_1, m_2	Prime number in Z_N^*
p, q	Large prime number
a	Random number in $Z_{N^2}^*$
P	Super-increasing sequence
g, ρ	Group generator
x_1, x_2	Random number in Z_N^*
$\mathbb{G}_1, \mathbb{G}_2$	Cyclic group
e	Bilinear map
$H(\cdot)$	Hash function
ρ_i	Element in group $\mathbb{G}_1$
M	Multiplication of m_i
$f(m)$	The coefficient of China remainder theorem
$\mathrm{SK}_{\mathrm{DT}_i}$	Secret key of DT_i
$\mathrm{PK}_{\mathrm{DT}_i}$	Public key of DT_i
$\mathrm{ID}_{\mathrm{DT}_i}$	Identity of DT_i
TYPE_i	The ith data type
ξ_i, r_1, r_2	Random number in Z_μ
$\mathrm{VK}_{\mathrm{TYPE}_i}$	The ith type verification key
$\mathrm{VK}_{\mathrm{TYPE}_{i1/2/3/4}}$	The ith type verification subkey
$\mathrm{SK}_{\mathrm{IoT}}$	Common secret key for all IoT
C_i	The ith type selection key
$C_{i1/2/3/4}$	The ith type selection subkey
c_i	Ciphertext of the ith type data
σ_i	Signature of the ith type data
c_{all}	Aggregated ciphertext of the ith type data
σ_{all}	Aggregated signature of the ith type data
IoT_i	The ith IoT device
edge_i	The ith edge server
DT_i	The ith DT server
m	The data context
r_i', $\hat{r}_i$	Random number in range from 1 to $\frac{N}{4}$

$g = -a^{2N} \bmod N^2$. Finally, TA selects a large prime $P \in [1, \frac{N}{4}]$ to construct a super-increasing sequence $\left(1, P, P^2, \ldots, P^i\right)$.

(2) TA selects two cyclic groups $\mathbb{G}_1$ with generator $\rho \in \mathbb{Z}^+$ and $\mathbb{G}_2$, where the order of both $\mathbb{G}_1$ and $\mathbb{G}_2$ is the prime μ and a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ holds.

(3) TA chooses a strong collision-resistant hash function $H(\cdot) : \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_N^*$, and randomly chooses two primes $m_1, m_2 \in \mathbb{Z}_N^*$ and two integer $x_1, x_2 \in Z_N$.

(4) Finally, TA publishes the public parameters $(g, H(\cdot), \mathbb{G}_1, \mathbb{G}_2, e, \mu, \rho, \rho_1, \rho_2, f(m), N, P)$, where $\rho_i = \rho^{x_i}$ and $f(m_i) = \frac{M}{m_i}$. Note that $M = \prod_{i=1}^{2} m_i$ and $m_i \neq 0$.

• **Key distribution.**

(1) TA computes $a_i = x_1 \bmod m_i$ and selects e_i satisfying with $f(m_i) \cdot e_i = 1 \bmod m_i$.

(2) TA selects $\mathrm{SK}_{\mathrm{DT}_i} = \tau_i \in \frac{\lambda(N^2)}{8}$ as the secret key of DT_i, and generates the public key $\mathrm{PK}_{\mathrm{DT}_i} = g^{\tau_i}$ for DT_i.

(3) TA maintains a DT's key label list as $\mathrm{List} = \{\mathrm{ID}_{\mathrm{DT}_i}, \mathrm{SK}_{\mathrm{DT}_i}, \mathrm{PK}_{\mathrm{DT}_i}\}$, where $\mathrm{ID}_{\mathrm{DT}_i}$ is the identity of DT_i.

(4) TA transmits $(\mathrm{SK}_{\mathrm{DT}_i}, \mathrm{PK}_{\mathrm{DT}_i})$ to corresponding DT_i in a secure channel. Then, the DT_i publishes $\mathrm{PK}_{\mathrm{DT}_i}$.

(5) Assume each DT_i has i kinds of data type that can be denoted as $\{\mathrm{TYPE}_1, \ldots, \mathrm{TYPE}_i\}$. Then, DT_i randomly picks $\xi_i \in Z_\mu$ to figure out the ith type verification key $\mathrm{VK}_{\mathrm{TYPE}_i} = \{\mathrm{VK}_{\mathrm{TYPE}_{i1}}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}, \mathrm{VK}_{\mathrm{TYPE}_{i3}}, \mathrm{VK}_{\mathrm{TYPE}_{i4}}\}$, where $\mathrm{VK}_{\mathrm{TYPE}_{i1}} = \rho_2^{\xi_i}$, $\mathrm{VK}_{\mathrm{TYPE}_{i2}} = H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{\xi_i}$, $\mathrm{VK}_{\mathrm{TYPE}_{i3}} = \rho_2^{e_2 \cdot a_2 \cdot \xi_i}$, $\mathrm{VK}_{\mathrm{TYPE}_{i4}} = \rho_2^{f(m_1) \cdot e_1 \cdot a_1 \cdot \xi_i}$.

(6) DT_i uploads the ith data type ciphertexts $\mathrm{VK}_{\mathrm{TYPE}_i} = \{\mathrm{VK}_{\mathrm{TYPE}_{i1}}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}, \mathrm{VK}_{\mathrm{TYPE}_{i3}},$

Algorithm 1 Data type selection algorithm.

Input: Ciphertext $(\mathrm{VK}_{\mathrm{TYPE}_{i1}}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}, \mathrm{VK}_{\mathrm{TYPE}_{i3}}, \mathrm{VK}_{\mathrm{TYPE}_{i4}})$, $i > 0$, and selected type $C_i = (C_{i1}, C_{i2}, C_{i3}, C_{i4})$, $i = 1, \ldots, \pi$.
Output: c_i, $i > 0$.

for $i = 1$ to k **do**
  $c_i = (c_{i1} = 0, c_{i2} = 0)$;
  $\sigma_i = (\sigma_{i1} = 0, \sigma_{i2} = 0)$;
  for $j = 1$ to π **do**
    label1 $= e\left(C_{j1}, \mathrm{TYPE}_{i1}\right)$;
    label2 $= e\left(C_{j2}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}\right)$;
    label3 $= e\left(C_{j3}, \mathrm{VK}_{\mathrm{TYPE}_{i3}}\right) \cdot e\left(C_{j4}, \mathrm{VK}_{\mathrm{TYPE}_{i4}}\right)$;
    if label1 == label2 $\cdot$ label3 **then**
      $c_i = (c_{i1} \cdot c_{j1}, c_{i2} \cdot c_{j2})$;
      $\sigma_i = (\sigma_{i1} \cdot \sigma_{j1}, \sigma_{i2} \cdot \sigma_{j2})$;
    end if
  end for
end for
return c_i, $i > 0$.

$\mathrm{VK}_{\mathrm{TYPE}_{i4}}\}$ and P to the ith edge node edge_i for data selection and filter.

(7) For each participated IoT device IoT_i, TA randomly chooses $\beta \in [1, N/4]$ as their common secret key $\mathrm{SK}_{\mathrm{IoT}}$ and computes public key $\mathrm{PK}_{\mathrm{IoT}} = \rho_2^{\beta}$, which will be distributed to all IoT devices in a secure channel.

• **Device layer data encryption.**

(1) IoT_i selects $r_1, r_2 \in Z_\mu$ to calculate $C_i = \{C_{i1}, C_{i2}, C_{i3}, C_{i4}\}$ for the ith type encrypted data selection, where $C_{i1} = \rho_1^{r_1} \cdot H\left(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i}\right)^{r_2}$, $C_{i2} = \rho_2^{r_2}, C_{i3} = \rho^{f(m_2) \cdot r_1}$ and $C_{i4} = \rho^{r_1}$.

(2) IoT_i chooses two random numbers $r_i', \hat{r}_i \in [1, N/4]$ to encrypt the data context m_i as $c_i = (c_{i1}, c_{i2}) = (g^{r_i'}, \mathrm{PK}_{\mathrm{DT}_i}^{r_i'}(1 + m_i \cdot n) \bmod N^2), \sigma_i = (\sigma_{i1}, \sigma_{i2}) = (H(\mathrm{ID}_{\mathrm{DT}_i} \| T)^{\hat{r}_i} \cdot \rho_1^{m_i \mathrm{SK}_{\mathrm{IoT}}}, \rho_2^{\hat{r}_i})$.

(3) IoT_i broadcasts $(\mathrm{ID}_{\mathrm{IoT}_i}, \mathrm{ID}_{\mathrm{DT}_i}, c_i, C_i, \sigma_i)$ to the edge_i.

• **Edge layer data selective aggregation.**

(1) edge_i needs to distinguish whether the received data type TYPE_i is the expected one. For achieving this goal, edge_i calculates $e(C_{i1}, \mathrm{VK}_{\mathrm{TYPE}_{i1}}) = e(\rho_1^{r_1} \cdot H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{r_2}, \rho_2^{\xi_i}) = e(\rho_1^{r_1}, \rho_2^{\xi_i}) \cdot e(H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{r_2}, \rho_2^{\xi_i})$, $e(C_{i2}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}) = e(\rho_2^{r_2}, H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{\xi_i})$, $e(C_{i3}, \mathrm{VK}_{\mathrm{TYPE}_{i3}}) = e(\rho^{f(m_2) \cdot r_1}, \rho_2^{\xi_i \cdot e_2 \cdot a_2}) = e(\rho, \rho_2)^{r_1 \xi_i f(m_2) \cdot e_2 \cdot a_2}$, $e(C_{i4}, \mathrm{VK}_{\mathrm{TYPE}_{i4}}) = e(\rho^{r_1}, \rho_2^{\xi_i f(m_1) \cdot e_1 \cdot a_1}) = e(\rho, \rho_2)^{r_1 \xi_i f(m_1) \cdot e_1 \cdot a_1}$.

(2) Subsequently, edge_i needs to verify whether the following equation $e(C_{i1}, \mathrm{VK}_{\mathrm{TYPE}_{i1}}) = e(C_{i2}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}) \cdot e(C_{i3}, \mathrm{VK}_{\mathrm{TYPE}_{i3}}) \cdot e(C_{i4}, \mathrm{VK}_{\mathrm{TYPE}_{i4}})$ holds. If it does not hold, the encrypted sensing data will be dropped. The correctness of this equation can be verified here: $e(C_{i2}, \mathrm{VK}_{\mathrm{TYPE}_{i2}}) \cdot e(C_{i3}, \mathrm{VK}_{\mathrm{TYPE}_{i3}}) \cdot e(C_{i4}, \mathrm{VK}_{\mathrm{TYPE}_{i4}}) = e(\rho_2^{r_2}, H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{\xi_i}) \cdot e(\rho^{\frac{f(m_2) \cdot r_1}{m_1}}, \rho_2^{m_1 \cdot \xi_i \cdot e_2 \cdot a_2}) \cdot e(\rho^{r_1}, \rho_2^{\xi_i f(m_1) \cdot e_1 \cdot a_1}) = e(\rho_2^{r_2}, H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{\xi_i}) \cdot e(\rho, \rho_2)^{r_1 \xi_i (f(m_2) \cdot e_2 \cdot a_2 + f(m_1) \cdot e_1 \cdot a_1)} = e(\rho_2^{r_2}, H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{\xi_i}) \cdot e(\rho, \rho_2)^{x_1 r_1 \xi_i} = e(\rho_2^{r_2}, H(\mathrm{TYPE}_i \| \mathrm{ID}_{\mathrm{DT}_i})^{\xi_i}) \cdot e(\rho_1, \rho_2)^{r_1 \xi_i} = e(C_{i1}, \mathrm{VK}_{\mathrm{TYPE}_{i1}})$. The detailed data type selection process can be found in Algorithm 1.

(3) edge_i aggregates the same type of all encrypted sensing message c_i from multiple IoT_i as $(c, \sigma) = (c_i, \sigma_i)_{i=1}^k$. Note that the detailed process is shown in Algorithm 2.

(4) edge_i aggregates all encrypted sensing message c_{all} using P as $c_{\mathrm{all}} = (c_{\mathrm{all}_1}, c_{\mathrm{all}_2}) = (\prod c_{i1}^{P^i}, \prod c_{i2}^{P^i})$ and signatures σ_{all} as $\sigma_{\mathrm{all}} = (\prod \sigma_{i1}, \prod \sigma_{i2})$. Note that signatures can be aggregated due to shared secret key $\mathrm{SK}_{\mathrm{IoT}}$ between all IoT devices.

(5) edge_i sends c_{all} and σ_{all} to the DT_i.

• **Twin layer data decryption.**

(1) After receiving the aggregated results c_{all} and σ_{all} at time period T, DT_i firstly uses its secret key τ_i to decrypt the aggregated results of data type TYPE_i as $\frac{c_{\mathrm{all}_2}}{c_{\mathrm{all}_1}^{\tau_i}} = \frac{\mathrm{PK}_{\mathrm{DT}_i}^{\sum r_i'}(1 + \left(\sum m_i P^i\right) N) \bmod N^2}{g^{\tau_i \sum r_i'}} = \frac{g^{\tau_i \sum r_i'}(1 + \left(\sum m_i P^i\right) N) \bmod N^2}{g^{\tau_i \sum r_i'}} = 1 + \left(\sum m_i P^i\right) N \bmod N^2$, where m_i represents the aggregated result of all sensing reports of data type TYPE_i.

(2) The result is recovered by executing $L(x)$ function: $L(\frac{c_{\mathrm{all}_2}}{c_{\mathrm{all}_1}^{\tau_i}}) = L((1 + \left(\sum m_i P^i\right) N) \bmod N^2) = \frac{(1 + \left(\sum m_i P^i\right) N) - 1 \bmod N^2}{N} = \sum m_i P^i$, where $L(x) = \frac{x-1}{N}$. Note that for $m = \sum m_i P^i$, the detailed process is given in the Algorithm 3 to recover the aggregated result of each data type.

Algorithm 2 Data content aggregation algorithm.

Input: Selected ciphertext collection (c_i), $i = 1, \ldots, \pi$, and signature collection $\sigma_i = (\sigma_{i1}, \sigma_{i2})$, $i = 1, \ldots, \pi$.
Output: (c_i, σ_i).

for $i = 1$ to size(c_i) **do**
  $c_i = 0, \sigma_i = 0$;
  for $j = 1$ to π **do**
    $c_{ij} = (c_{ij1}, c_{ij2})$;
    $c_i = c_i \cdot c_{ij}$;
    $\sigma_{ij} = (\sigma_{ij1}, \sigma_{ij2})$;
    $\sigma_i = \sigma_i \cdot \sigma_{ij}$;
  end for
end for
return (c_i, σ_i).

Algorithm 3 Aggregated result of each data type recover.

Input: $m = m_1 + m_2 P + \cdots + m_i P^i$, and a super-increasing sequence $(1, P, \ldots, P^i)$ with $m_i < P - 1$.
Output: $(m_1, m_2, \ldots, m_i)$.

for i to 0 **do**
  $m_{i-1} = m \bmod P^i$;
  $m_i = \frac{m - m_{i-1}}{P^i}$;
end for
return $(m_1, m_2, \ldots, m_i)$.

• **DT network monitor.**

(1) Prior to DT network formulation, the integrity of the aggregation data needs to be verified according to the following equation: $e(\sigma_{\text{all}_1}, \rho_2) = e(\prod H(\text{ID}_{\text{DT}_i} \| T)^{\hat{r}_i} \rho_1^{m_i \text{SK}_{\text{IoT}}}, \rho_2) = e(\prod H(\text{ID}_{\text{DT}_i} \| T)^{\hat{r}_i}, \rho_2) \cdot e(\rho_1^{\sum m_i \text{SK}_{\text{IoT}}}, \rho_2) = e(\prod H(\text{ID}_{\text{DT}_i} \| T)^{\hat{r}_i}, \rho_2) \cdot e(\rho_1^{\sum m_i}, \rho_2^{\text{SK}_{\text{IoT}}}) = e(\prod H(\text{ID}_{\text{DT}_i} \| T)^{\hat{r}_i}, \rho_2) \cdot e(\rho_1^{\sum m_i}, \rho_2^{\beta}) = e(\prod H(\text{ID}_{\text{DT}_i} \| T)^{\hat{r}_i}, \rho_2) \cdot e(\rho_1^{\sum m_i}, \text{PK}_{\text{IoT}}) = e(H(\text{ID}_{\text{DT}_i} \| T), \prod \rho_2^{\hat{r}_i}) \cdot e(\rho_1^{\sum m_i}, \rho_2^{\beta}) = e(H(\text{ID}_{\text{DT}_i} \| T), \sigma_{\text{all}_2}) \cdot e(\rho_1, \rho_2^{\beta})^{\sum m_i}$.

(2) After the DT network is formulated, it needs to ensure the correctness and availability of the uploaded data. In other words, the DT network needs to real-time detect malicious or error IoT devices that transmit abnormal data. On the one hand, a large amount of data collected from IoT devices presents a non-linear pattern. On the other hand, these data are stored in chronological order and depend on the current state perceived by dynamically operating IoT devices. Hence, the RNN model can well capture complex non-linear relationships among various parameters by exploring the relation between the inputs and outputs. Moreover, the LSTM is a self-loop structure that stores temporal information and shows strength in a prediction model construction by combining the RNN and the LSTM for the time series sensing data. Thus, we generate an LSTM-RNN future data trend prediction model to verify the correctness and availability of IoT data and build the reverse regulation in the DT network.

First, DT_i processes sequential data and trains the LSTM-RNN model as $h^t, y^t = f(h^{t-1}, x^t)$, where x^t represents the data input in the current state, h^{t-1} represents the received input from the previous node, y^t represents the output in the current state, and h^t represents the output passed to the next node. Thus, our LSTM-RNN model is constructed according to the state transitions. The detailed process is shown as follows: $c^t = z^f \odot c^{t-1} + z^i \odot z, h^t = z^o \odot \tanh(c^t), y^t = \sigma(W' h^t)$, where z^f, z^i and z^o are converted into values 0 or 1 by a sigmoid activation function after the concatenation vector is multiplied by the weight matrix. z is converted into values 0 or 1 by a tanh activation function. $\odot$ represents the Hadamard Product. W' represents the state transition control variable. c^{t-1} represents the forgetting information.

Then, DT_i uses the LSTM-RNN model to formulate the deviation for each data uploaded from IoT devices. Once either h^t or y^t, in practice, deviates more than a predefined threshold according to the LSTM-RNN model predicted in advance, the related device will be marked temporarily.

Finally, the DT network will analyze each data submitted from the IoT device and locate the broken devices according to the corrective information derived from the LSTM-RNN model to achieve reverse regulation. Further, to monitor the dynamic change of DT_i network, we give a dynamic feedback mechanism to discard or add new IoT devices and maintain the stability of the proposed DT network.

• **Side-channel attack protection.** To protect against potential side-channel attacks in our proposed scheme, we implement comprehensive protection mechanisms while maintaining system efficiency. For timing attacks, we employ constant-time operations in all cryptographic computations and incorporate random delays in key operations. Against power analysis, our framework implements balanced power

consumption patterns and random operation scheduling at the device layer to mask power signatures. To mitigate cache-based attacks, we utilize pre-loading of critical data and cache-resistant algorithms, combined with randomized memory access patterns. These protective measures are carefully integrated with our security mechanisms to ensure comprehensive protection while maintaining operational efficiency.

7 Security analysis

In this section, we firstly give proof that our scheme is securely emulated by Definition 4. Then we analyze the security of our scheme under the Definition 4, which can verify whether our scheme satisfies the security concerns of ciphertext indistinguishability (i.e., Definition 1), signature privacy (i.e., Definition 2), and type privacy (i.e., Definition 3).

Theorem 1. Suppose our scheme can be securely emulated according to Definition 4 with adversaries $\mathcal{A} = (\mathcal{A}_{D_a}, \mathcal{A}_{E_b})$. Our scheme satisfies ciphertext indistinguishability, signature privacy, and type privacy.

Proof. Assume that our scheme does not satisfy ciphertext indistinguishability or signature privacy, or type privacy. We say that it cannot be securely emulated according to Definition 4.

Let there be a distinguisher $\mathcal{Z}$, who invariably tries to distinguish the real world from the ideal world.

• Suppose that our scheme does not satisfy the ciphertext indistinguishability (i.e., Definition 1). In other words, adversary $\mathcal{B}$ exists so that Eq. (3) shows a non-negligible advantage. Then distinguisher $\mathcal{Z}$ instigates $\mathcal{A}$ or $\mathcal{S}$ to corrupt edge server E_b, where E_b should behave honestly to relay each message received from IoT device D_a to $\mathcal{Z}$. Adversary $\mathcal{B}$ run by internal adversary chooses two distinct plaintexts $m_1, m_2 \subseteq Z_N$ and sends them to challenger $\mathcal{C}$.

• $\mathcal{Z}$ stimulates D_a with input **DLDataEnc**, sid, D_b, where $b \in \{0, 1\}$ is the random bit and sid is a counter.

• In the real world, D_a first uploads c_b to $E_b(\mathcal{A})$, then $E_b(\mathcal{A})$ relays it to $\mathcal{Z}$. In the ideal world, D_a first uploads **DLDataEnc**, sid, D_b to $\boldsymbol{f}$, then $\boldsymbol{f}$ relays $|D_b|$ to $\mathcal{S}$. Finally, $\mathcal{S}$ computes c'_b and relays it to $\mathcal{Z}$.

When $\mathcal{B}$ wins the game, $\mathcal{Z}$ can distinguish between the real world and the ideal world. If $\mathcal{Z}$ interacts with the protocol Π, $\mathcal{B}$ generates the c_b, then adversary $\mathcal{A}$ plays the role of $\mathcal{A}_{E_b}$ in the real world. However, if $\mathcal{Z}$ interacts with $\mathcal{S}$, $\mathcal{B}$ generates the c'_b, where the adversary $\mathcal{S}$ plays the role of $\mathcal{S}_{E_b}$ in the ideal world. The proof for the remaining assumptions that our scheme does not satisfy signature privacy (i.e., Definition 2), and type privacy (i.e., Definition 3) are similar to ciphertext indistinguishability (i.e., Definition 1), so we omit the detailed description here.

For the above assumption, adversary $\mathcal{B}$ distinguishes ciphertext in the real world, and wins the game by outputting 1 with the non-negligible advantage over 0. However, in the ideal world, $\mathcal{B}$ outputs with probability $\frac{1}{2}$. Obliviously, $\mathcal{Z}$ runs $\mathcal{B}$ as a subroutine that can distinguish the partial view of the entity E_b between the real world and the ideal world execution. It proves that the protocol without ciphertext indistinguishability, signature privacy and type privacy cannot securely emulate our scheme.

Theorem 2. The protocol introduced in Section 6 securely realizes our DT framework based on Definition 4 when adversaries $\mathcal{A} = (\mathcal{A}_{D_a}, \mathcal{A}_{E_b})$ exist.

Proof. Sim_{D_a} receives (x) as input and emulates $\mathcal{A}_{D_a}$ as follows: Sim_{D_a} computes $(C) \leftarrow$ **DLDataEnc**(x) by adopting a purified Paillier cryptosystem and returns ciphertext (C) to $\mathcal{A}_{D_a}$. Since $\mathcal{A}_{D_a}$ does not know the corresponding private key for decryption so that (C) cannot be restored. Note that the entire view of $\mathcal{A}_{D_a}$ is (C) and (C) are indistinguishable between the real world and the ideal world executions due to the semantic security of the purified Paillier cryptosystem. In addition, although $\mathcal{A}_{D_a}$ can obtain the ciphertext $(C = ((g^{r'_i}, \mathrm{PK}_{\mathrm{DT}_i}^{r'_i}(1 + m_i \cdot n) \bmod N^2)))$, where $\mathrm{PK}_{\mathrm{DT}_i} = g^{\tau_i}$. However, it is impossible to decrypt x from the (C) since $\mathcal{A}_{D_a}$ does not know $\mathrm{PK}_{\mathrm{DT}_i}^{r'_i}$. Furthermore, computing $\mathrm{PK}_{\mathrm{DT}_i}^{r'_i}$ from $(g^{r'_i}, \mathrm{PK}_{\mathrm{DT}_i} = g^{\tau_i})$ is similar to solve the CDH problem. It is in contradiction with the difficulty of solving the CDH assumption. Thus, encrypted data privacy can be guaranteed.

Sim_{D_a} receives (TYPE) as input and emulates $\mathcal{A}_{D_a}$ as follows: Sim_{D_a} computes $(C) \leftarrow$ **DLDataEnc** (TYPE) and returns ciphertext (C) to $\mathcal{A}_{D_a}$. For this date type ciphertext (C), $C_1 = \rho_1^{r_1} \cdot H(\mathrm{TYPE} \| \mathrm{ID}_{\mathrm{DT}_i})^{r_2}$, $C_2 = \rho_2^{r_2}$, $C_3 = \rho^{f(m_2) \cdot r_1}$, $C_4 = \rho^{r_1}$, where r_1 and r_2 are random numbers. Moreover, $\mathcal{A}_{D_a}$ does not know $\rho_1^{r_1}$ in the data type selection phase. Thus, $\mathcal{A}_{D_a}$ cannot obtain anything about data type (TYPE) from $\{C_{i1}, C_{i2}, C_{i3}, C_{i4}\}$. Note that the entire view of $\mathcal{A}_{D_a}$ is (C) and the (C) is indistinguishable. To distinguish the data type between the real world and the ideal world executions, $\mathcal{A}_{D_a}$

can randomly construct a ciphertext (C') and check whether the following equation holds $e(\frac{C/C'}{\rho_1^{r_1-r_1'}}, \rho_2) = e(H(\text{TYPE}\|\text{ID}_{\text{DT}_i}), \rho_2^{r_2-r_2'})$. Furthermore, obtaining $\rho_2^{r_2-r_2'}$ from the equation $(\rho, \rho_1 = \rho^{x_1}, \rho^{r_2-r_2'})$ is equivalent to solve the CDH problem. As we all know, it is computationally infeasible to solve CDH. Thus, the encrypted data type can be preserved.

Sim_{D_a} receives (x, TYPE) as input and emulates $\mathcal{A}_{D_a}$ as follows: Sim_{D_a} computes $(\sigma) \leftarrow$ **DLDataEnc** (x, TYPE) and returns ciphertext (σ) to $\mathcal{A}_{D_a}$. Given the above proof, the privacy of encrypted data and data type is preserved so that the entire view of $\mathcal{A}_{D_a}$ is (σ). Hence, the (σ) is indistinguishable between the real world and the ideal world executions due to the semantic security of our DT network.

Since Theorems 1 and 2 have been proved, it is clear that our scheme satisfies ciphertext indistinguishability, signature privacy, and type privacy.

8 Performance evaluation

In this section, we compare our scheme with four recent secure data aggregation schemes (i.e., LVPDA [26], AMDA [27], EdgeVANET [28], and CBACS [29]) in terms of communication and computation cost. Moreover, a case study is put for reverse regulation verification in DT. Note that we conclude the batch verification stage in EdgeVANET and CBACS as the data aggregation phase in this scheme. Then we give a case study for reverse regulation in DT and conduct computational complexity analysis. Finally, the user experience and system usability analysis are discussed.

8.1 Computation cost analysis

For computation cost evaluation, we first count the number of cryptographic operations in each phase and then figure out the overall cost for comparison. Since the computation cost for addition is smaller than bilinear pairing and other operations, we omit their calculation here. We implement our experiment on a laptop with Intel(R) Core(TM) i5-6300U CPU of 2.40 GHz and 6.0 GB RAM and Windows 10 system.

Since the system key generation and distribution only take one-time execution, we mainly consider the computation cost for device layer data encryption, edge layer data selective aggregation and twin layer data decryption during the whole procedure. In the device layer data encryption phase, our protocol first needs to run $4T_{\text{EXP}_\mathbb{G}}$, $2T_{\text{MUL}_\mathbb{Z}}$, T_H and $T_{\text{EXP}_\mathbb{Z}}$ to generate ciphertext on encrypted data type $C_i = \{C_{i1}, C_{i2}, C_{i3}, C_{i4}\}$. Then $4T_{\text{EXP}_\mathbb{G}}$, $3T_{\text{MUL}_\mathbb{Z}}$, T_H and $T_{\text{EXP}_\mathbb{Z}}$ are consumed for ciphertext and signature generation. Hence, the total time cost for device layer data encryption is $8T_{\text{EXP}_\mathbb{G}}+2T_H+2T_{\text{EXP}_\mathbb{Z}}+5T_{\text{MUL}_\mathbb{Z}}$. In the edge layer data selective aggregation phase, our protocol needs to execute $4nT_{\text{BP}}$ to figure out four corresponding data type verification value $e(C_{i1}, \text{VK}_{\text{TYPE}_{i1}}), e(C_{i2}, \text{VK}_{\text{TYPE}_{i2}}), e(C_{i3}, \text{VK}_{\text{TYPE}_{i3}}), e(C_{i4}, \text{VK}_{\text{TYPE}_{i4}})$. Subsequently, $2nT_{\text{MUL}_\mathbb{G}}$ are used for data type verification. At last, $2nT_{\text{EXP}_\mathbb{G}}$ and $(4n-2)T_{\text{MUL}_\mathbb{G}}$ are used for ciphertext and signature aggregation. Note that the n defines the number of data types. The total time cost for data selective aggregation is $2nT_{\text{EXP}_\mathbb{G}} + (6n-2)T_{\text{MUL}_\mathbb{G}} + 4nT_{\text{BP}}$. In the twin layer data decryption phase, our protocol requires $2T_{\text{EXP}_\mathbb{G}} + 2T_{\text{MUL}_\mathbb{G}} + 3T_{\text{BP}} + T_{\text{MUL}_\mathbb{Z}} + T_H$ for data decryption and signature verification. Hence, the total time cost for our protocol is $(2n+10)T_{\text{EXP}_\mathbb{G}} + 3T_H + 2T_{\text{EXP}_\mathbb{Z}} + 6nT_{\text{MUL}_\mathbb{G}} + 6T_{\text{MUL}_\mathbb{Z}} + (4n+3)T_{\text{BP}}$. The data aggregation protocol LVPDA [26] takes $8T_{\text{EXP}_\mathbb{G}}$, $4T_H$, $3T_{\text{EXP}_\mathbb{Z}}$, $5T_{\text{MUL}_\mathbb{Z}}$ and $2T_{\text{MUL}_\mathbb{G}}$ in the IoT data encryption process. For the edge server side, LVPDA takes $(n-1)T_{\text{MUL}_\mathbb{G}}$, $(3n+1)T_H$, $(2n-2)T_{\text{BP}}$ and $T_{\text{EXP}_\mathbb{Z}}$ to check data integrity and aggregate the received data. $2T_{\text{EXP}_\mathbb{G}}$, $2T_{\text{MUL}_\mathbb{Z}}$, $2T_H$, $2T_{\text{BP}}$ and $T_{\text{MUL}_\mathbb{G}}$ are used for data decryption and verification. Therefore, the total computation cost for LVPDA is $7T_{\text{MUL}_\mathbb{Z}} + 10T_{\text{EXP}_\mathbb{G}} + (n+2)T_{\text{MUL}_\mathbb{G}} + (3n+7)T_H + 2nT_{\text{BP}} + 4T_{\text{EXP}_\mathbb{Z}}$. For the AMDA [27], it consumes $T_{\text{EXP}_\mathbb{G}}$, T_H, $2T_{\text{EXP}_\mathbb{Z}}$ and $3T_{\text{MUL}_\mathbb{Z}}$ for the privacy-preserving data generation. Regarding the data verification and aggregation, edge server takes $(n+2)T_{\text{MUL}_\mathbb{Z}}$, $nT_{\text{EXP}_\mathbb{G}}$, $(3n-2)T_{\text{MUL}_\mathbb{G}}$, $(n+1)T_H$ and nT_{BP}. In the final data decryption and verification stage, AMDA takes $2T_{\text{EXP}_\mathbb{G}}$, $3T_{\text{MUL}_\mathbb{G}}$, T_H and $2T_{\text{BP}}$. Hence, the total computation cost for AMDA is $(n+5)T_{\text{MUL}_\mathbb{Z}} + (n+3)T_{\text{EXP}_\mathbb{G}} + 2T_{\text{EXP}_\mathbb{Z}} + (3n+1)T_{\text{MUL}_\mathbb{G}} + (n+3)T_H + (n+2)T_{\text{BP}}$.

Regarding the EdgeVANET scheme [28], the vehicle first spends $3T_{\text{MUL}_\mathbb{G}} + 7T_H + T_{\text{AES}} + T_{\text{EXP}_\mathbb{G}}$ for data encryption. For the batch verification in the edge server, the computation needs to take $(n+2)T_{\text{MUL}_\mathbb{G}} + 2nT_H$. When the server or the target vehicle receives the transmitted data, $3T_{\text{MUL}_\mathbb{G}} + +2T_H + T_{\text{EXP}_\mathbb{G}}$ are necessary for decryption. From the above analysis, we can know the total computation cost for the EdgeVANET scheme is $(n+5)T_{\text{MUL}_\mathbb{G}} + (9+2n)T_H + T_{\text{AES}} + 2T_{\text{EXP}_\mathbb{G}}$. Concerning

Table 2 Time cost for different operations.

Abbreviation	Operation	Time cost (μs)
$T_{\mathrm{EXP}_{\mathbb{Z}}}$	Modular exponentiation operation in $\mathbb{Z}_{\mathbb{N}}$	100
$T_{\mathrm{EXP}_{\mathbb{G}}}$	Modular exponentiation operation in $\mathbb{G}$	800
T_{BP}	Bilinear pairing operation	1200
$T_{\mathrm{MUL}_{\mathbb{Z}}}$	Modular multiplication operation in $\mathbb{Z}_{\mathbb{N}}$	200
$T_{\mathrm{MUL}_{\mathbb{G}}}$	Modular multiplication operation in $\mathbb{G}$	1600
T_H	Hash operation	15000
T_{AES}	AES operation	300

CBACS [29], the computation cost for device, edge server and destination server is $9T_{\mathrm{MUL}_{\mathbb{G}}}+7T_H+3T_{\mathrm{AES}}$, $(2n+2)T_{\mathrm{MUL}_{\mathbb{G}}}+2nT_H$ and $10T_{\mathrm{MUL}_{\mathbb{G}}}+8T_H+3T_{\mathrm{AES}}$, respectively. The overall computation cost for CBACS is $(2n+21)T_{\mathrm{MUL}_{\mathbb{G}}}+(2n+15)T_H+6T_{\mathrm{AES}}$. According to the measurement result for each cryptographic primitive provided in Table 2, the comparative computation cost for ours, LVPDA [26], AMDA [27], EdgeVANET [28] and CBACS [29] are presented in Figure 3 and Table 3 [27–30]. It can be seen from Figure 3(a), our protocol, LVPDA, AMDA, EdgeVANET [28] and CBACS [29] costs 376000, 709000, 166000, 110900, and 120300 μs, respectively. Compared with the LVPDA scheme, our data aggregation method reduces nearly two times the computation cost for ten concurrent IoT devices in the data encryption process. The computation has been reduced by nearly 2/3 when compared with EdgeVANET and CBACS schemes under the same ten devices' concurrent conditions. Although our proposed protocol takes more computation cost than the AMDA scheme, selective encrypted data aggregation is supported in our scheme, which greatly strengthens the communication security and releases the computation pressure from the DT server. As shown in Figure 3(b), each IoT computation cost for our scheme, LVPDA, AMDA, EdgeVANET and CBACS are 28800, 109100, 56200, 66400, and 69600 μs, where our protocol has achieved the least communication cost in the data aggregation process. Note that we assume the default required number of data type is 2. Figure 3(c) presents the computation cost for twin layer data decryption. It is obvious that our protocol also obtains the least computation cost since the edge server shares some work of data verification. The total communication cost for the above-mentioned three stages is illustrated in Figure 3(d). Our protocol can maintain the most efficient computation cost (i.e., 236000 μs for ten concurrent IoT devices) with the increasing number of IoT devices. Furthermore, the relationship between data type and computation cost is explored in Figures 3(e) and (f), respectively. From these two experiment results, we know that our scheme achieves the most efficient computation cost with the increasing number of types when compared with LVPDA, AMDA, EdgeVANET and CBACS schemes.

8.2 Communication cost analysis

For communication cost evaluation, we mainly consider the message transmission between the device-to-edge server and edge-to-DT server. In our protocol, each IoT device needs to send $(\mathrm{ID}_{\mathrm{IoT}_i}, \mathrm{ID}_{\mathrm{DT}_i}, c_i, C_i, \sigma_i)$ to the edge server, which costs $8|\mathbb{G}|+|\mathrm{ID}_{\mathrm{IoT}_i}|+|\mathrm{ID}_{\mathrm{DT}_i}|$ bits. Then the edge server transmits $c_{\mathrm{all}} = (c_{\mathrm{all}_1}, c_{\mathrm{all}_2})$, which consumes $4|\mathbb{G}|$ bits. Hence, the total communication cost for our protocol is around $12|\mathbb{G}|+|\mathrm{ID}_{\mathrm{IoT}_i}|+|\mathrm{ID}_{\mathrm{DT}_i}|$ bits. Regarding the LVPDA [26], the device sends $2|N|+5|\mathbb{G}|+2|q_1|+2|T_{\mathrm{Edge}}|+2|\mathrm{ID}_{\mathrm{Edge}}|$ bits to the edge server, which contains offline and online signature generation. From edge server to DT server, $4|N|+|\mathbb{G}|+|T_{\mathrm{Edge}}|$ bits are needed for LVPDA. Therefore, the total communication cost for LVPDA is $4|N|+6|\mathbb{G}|+2|q_1|+2|T_{\mathrm{Device}}|+2|\mathrm{ID}_{\mathrm{Device}}|+|T_{\mathrm{Edge}}|+|\mathrm{ID}_{\mathrm{Edge}}|$ bits, where T_{Device} and T_{Edge} are the timestamp for IoT device and edge server, respectively. For the AMDA [27], the message transmission for IoT device takes $4|N|+|\mathbb{G}|+|T_{\mathrm{Device}}|$ bits. From the edge server to the DT server, the communication cost is $4|N|+|\mathbb{G}|+|T_{\mathrm{Edge}}|$ bits. Hence, the total computation cost for AMDA is $8|N|+|2\mathbb{G}|+|T_{\mathrm{Device}}|+|T_{\mathrm{Edge}}|$ bits. Regarding the EdgeVANET [28], the communication cost from the vehicle to the edge server is $5|N|+2|\mathbb{G}|+|T_{\mathrm{Device}}|$. The consumed bits between the edge server and the destination server are $|N|+|\mathbb{G}|+|T_{\mathrm{Edge}}|$. Therefore, the overall transmission cost for EdgeVANET is $6|N|+|3\mathbb{G}|+|T_{\mathrm{Device}}|+|T_{\mathrm{Edge}}|$. Regarding the CBACS [29], the total communication cost $4|N|+|8\mathbb{G}|+2|T_{\mathrm{Device}}|+|T_{\mathrm{Edge}}|$ consists of two phases including $2|N|+5|\mathbb{G}|+2|T_{\mathrm{Device}}|$ and $2|N|+3|\mathbb{G}|+|T_{\mathrm{Edge}}|$.

We assume the bit length for Paillier cryptosystem parameter N, user identity $\mathrm{ID}_{\mathrm{Device}}/\mathrm{ID}_{\mathrm{Edge}}$, timestamp $T_{\mathrm{Device}}/T_{\mathrm{Edge}}$, prime q_1 and element in a group G are 320, 32, 32, 320 and 160 bits, respectively.

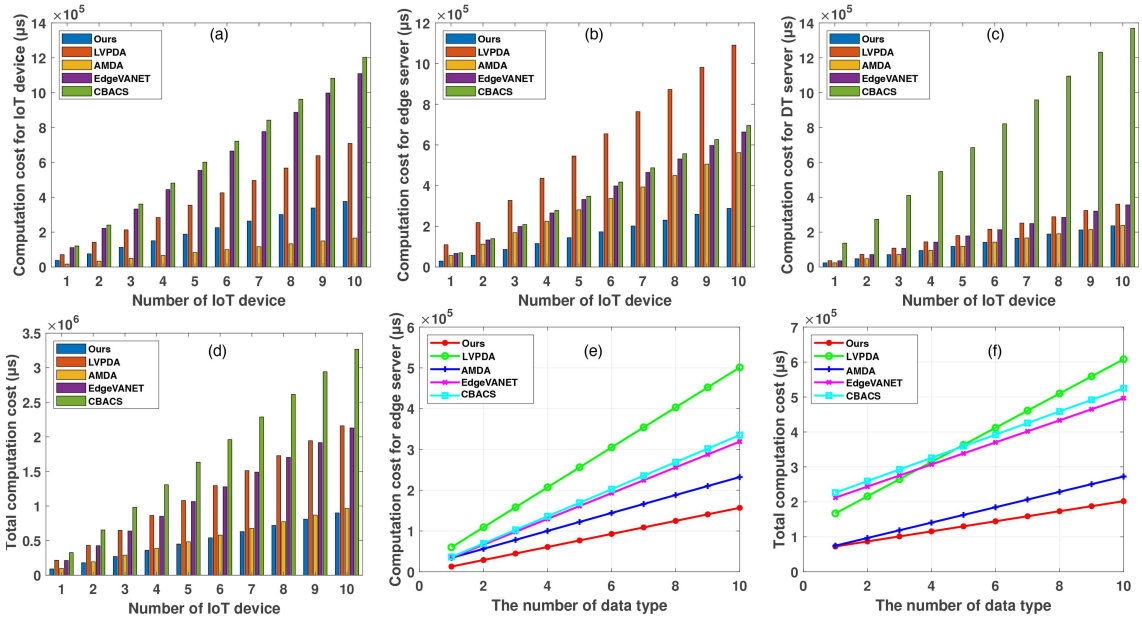

Figure 3 (Color online) Comparison of computation cost. (a) Device side; (b) edge server side ($|\text{Type}| = 2$); (c) DT server side; (d) total cost; (e) edge server side with types; (f) total cost with types.

Table 3 Comparison of computation cost.

Scheme	Device	Edge server	DT server	Total
Ours	$8T_{\text{EXP}_{\mathbb{G}}} + 2T_H$ $+2T_{\text{EXP}_{\mathbb{Z}}} + 5T_{\text{MUL}_{\mathbb{Z}}}$	$2nT_{\text{EXP}_{\mathbb{G}}} + (6n$ $-2)T_{\text{MUL}_{\mathbb{G}}} + 4nT_{\text{BP}}$	$2T_{\text{EXP}_{\mathbb{G}}} + 2T_{\text{MUL}_{\mathbb{G}}}$ $+3T_{\text{BP}} + T_{\text{MUL}_{\mathbb{Z}}} + T_H$	$(2n+10)T_{\text{EXP}_{\mathbb{G}}} + 3T_H$ $+2T_{\text{EXP}_{\mathbb{Z}}} + 6nT_{\text{MUL}_{\mathbb{G}}}$ $+6T_{\text{MUL}_{\mathbb{Z}}} + (4n+3)T_{\text{BP}}$
LVPDA [30]	$8T_{\text{EXP}_{\mathbb{G}}} + 4T_H + 3T_{\text{EXP}_{\mathbb{Z}}}$ $+5T_{\text{MUL}_{\mathbb{Z}}} + 2T_{\text{MUL}_{\mathbb{G}}}$	$(n-1)T_{\text{MUL}_{\mathbb{G}}} + (3n$ $+1)T_H + (2n-2)T_{\text{BP}}$ $+T_{\text{EXP}_{\mathbb{Z}}}$	$2T_{\text{EXP}_{\mathbb{G}}} + 2T_{\text{MUL}_{\mathbb{Z}}}$ $+2T_H + 2T_{\text{BP}} + T_{\text{MUL}_{\mathbb{G}}}$	$7T_{\text{MUL}_{\mathbb{Z}}} + 10T_{\text{EXP}_{\mathbb{G}}}$ $+(n+2)T_{\text{MUL}_{\mathbb{G}}} + (3n$ $+7)T_H + 2nT_{\text{BP}} + 4T_{\text{EXP}_{\mathbb{Z}}}$
AMDA [27]	$T_{\text{EXP}_{\mathbb{G}}} + T_H + 2T_{\text{EXP}_{\mathbb{Z}}}$ $+3T_{\text{MUL}_{\mathbb{Z}}}$	$(n+2)T_{\text{MUL}_{\mathbb{Z}}} + nT_{\text{EXP}_{\mathbb{G}}}$ $+(3n-2)T_{\text{MUL}_{\mathbb{G}}} + (n$ $+1)T_H + nT_{\text{BP}}$	$2T_{\text{EXP}_{\mathbb{G}}} + 3T_{\text{MUL}_{\mathbb{G}}}$ $+T_H + 2T_{\text{BP}}$	$(n+5)T_{\text{MUL}_{\mathbb{Z}}} + (n$ $+3)T_{\text{EXP}_{\mathbb{G}}} + 2T_{\text{EXP}_{\mathbb{Z}}}$ $+(3n+1)T_{\text{MUL}_{\mathbb{G}}} + (n$ $+3)T_H + (n+2)T_{\text{BP}}$
EdgeVANET [28]	$3T_{\text{MUL}_{\mathbb{G}}} + 7T_H + T_{\text{AES}}$ $+T_{\text{EXP}_{\mathbb{G}}}$	$(n+2)T_{\text{MUL}_{\mathbb{G}}} + 2nT_H$	$3T_{\text{MUL}_{\mathbb{G}}} + 2T_H$ $+T_{\text{EXP}_{\mathbb{G}}}$	$(n+5)T_{\text{MUL}_{\mathbb{G}}} + (9$ $+2n)T_H + T_{\text{AES}} + 2T_{\text{EXP}_{\mathbb{G}}}$
CBACS [29]	$9T_{\text{MUL}_{\mathbb{G}}} + 7T_H$ $+3T_{\text{AES}}$	$(2n+2)T_{\text{MUL}_{\mathbb{G}}} + 2nT_H$	$10T_{\text{MUL}_{\mathbb{G}}} + 8T_H$ $+3T_{\text{AES}}$	$(2n+21)T_{\text{MUL}_{\mathbb{G}}}$ $+(2n+15)T_H + 6T_{\text{AES}}$

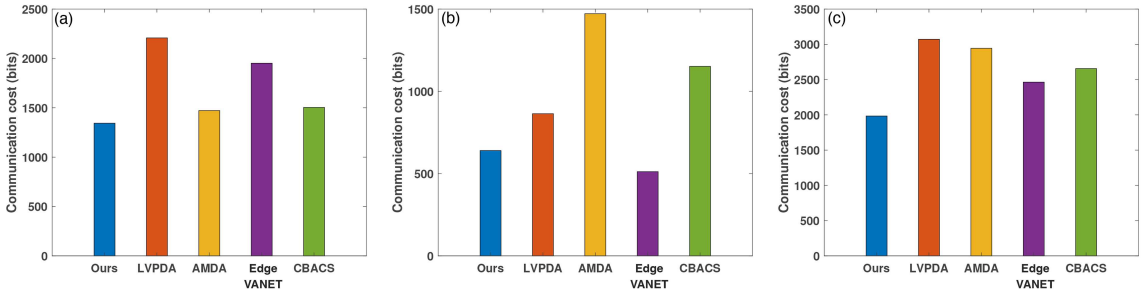

Figure 4 (Color online) Comparison of communication cost. (a) Device to edge; (b) edge to DT; (c) total cost.

Figure 4 and Table 4 [27–30] present the communication cost for our work compared with LVPDA [26], AMDA [27], EdgeVANET [28] and CBACS [29]. From the device-to-edge phase shown in Figure 4(a), our protocol costs 1344 bits which is lower than 2208 bits in LVPDA, 1472 bits in AMDA, 1952 bits in EdgeVANET and 1504 bits in CBACS. For the edge-to-DT phase presented in Figure 4(b), 640 bits are consumed in our protocol, which is much lower than 864, 1472, and 1152 bits in LVPDA, AMDA, and CBACS, respectively. However, EdgeVANET requires less transmission cost (i.e., 512 bits) at this stage. Nevertheless, in Figure 4(c), it is clear that our protocol achieves the least total communication cost — 1984 bits, which is reduced by 54.8%, 48.4%, 24.2% and 33.9% compared with LVPDA, AMDA,

Table 4 Comparison of communication cost.

Scheme	Device	Edge server	Total
Ours	$8\|G\| + \|\mathrm{ID}_{\mathrm{IoT}_i}\| + \|\mathrm{ID}_{\mathrm{DT}_i}\|$	$4\|\mathbb{G}\|$	$12\|\mathbb{G}\| + \|\mathrm{ID}_{\mathrm{IoT}_i}\| + \|\mathrm{ID}_{\mathrm{DT}_i}\|$
LVPDA [30]	$2\|N\| + 5\|\mathbb{G}\| + 2\|q_1\| + 2\|T_{\mathrm{Device}}\| + 2\|\mathrm{ID}_{\mathrm{Device}}\|$	$2\|N\| + \|\mathbb{G}\| + \|T_{\mathrm{Edge}}\| + \|\mathrm{ID}_{\mathrm{Edge}}\|$	$4\|N\| + 6\|\mathbb{G}\| + 2\|q_1\| + 2\|T_{\mathrm{Device}}\| + 2\|\mathrm{ID}_{\mathrm{Device}}\| + \|T_{\mathrm{Edge}}\| + \|\mathrm{ID}_{\mathrm{Edge}}\|$
AMDA [27]	$4\|N\| + \|\mathbb{G}\| + \|T_{\mathrm{Device}}\|$	$4\|N\| + \|\mathbb{G}\| + \|T_{\mathrm{Edge}}\|$	$8\|N\| + \|2\mathbb{G}\| + \|T_{\mathrm{Device}}\| + \|T_{\mathrm{Edge}}\|$
EdgeVANET [28]	$5\|N\| + 2\|\mathbb{G}\| + \|T_{\mathrm{Device}}\|$	$\|N\| + \|\mathbb{G}\| + \|T_{\mathrm{Edge}}\|$	$6\|N\| + \|3\mathbb{G}\| + \|T_{\mathrm{Device}}\| + \|T_{\mathrm{Edge}}\|$
CBACS [29]	$2\|N\| + 5\|\mathbb{G}\| + 2\|T_{\mathrm{Device}}\|$	$2\|N\| + 3\|\mathbb{G}\| + \|T_{\mathrm{Edge}}\|$	$4\|N\| + \|8\mathbb{G}\| + 2\|T_{\mathrm{Device}}\| + \|T_{\mathrm{Edge}}\|$

EdgeVANET, and CBACS, respectively.

8.3 Case study for reverse regulation in DT

To verify the reverse regulation in the DT network, we develop a case study that can detect abnormal behaviours of IoT devices [31] by analysing the electric load information received in the DT network. In this experiment, the dataset is referred to hourly power consumption in Toronto[2)]. Moreover, some interference is appended into this dataset to test reverse regulation function provided in our DT network. We utilize LSTM+RNN to train a reliable machine learning model that can predict the hourly electricity load in Toronto based on the loads of the previous 23 h. As can be seen from Figure 5(a), when the IoT device operates normally, the curve of real electricity load fits with the predicted value. However, when some accidents happen in the IoT device, an obvious variance appears between the real value and predicted value. As shown in Figure 5(b), the real-time electricity load for IoT devices becomes abnormal from the 300th hour, which may reflect the execution error for the current IoT devices. Then the DT network will analyse each data submitted from the IoT device and finally locate the broken devices to achieve reverse regulation.

Building upon the power load analysis, we further developed an industrial sensor simulator to validate our framework's effectiveness in multi-sensor scenarios. The simulator generates three types of industrial sensor data with realistic characteristics: temperature sensor data with daily periodic variations (baseline 25°C, amplitude 3°C) and environmental noise (standard deviation 0.2); pressure sensor data with a 100 kPa baseline incorporating gradual drift (0.005/h); and vibration sensor data centered at 2 mm/s with operational fluctuations. As shown in Figure 6, blue lines represent normal operational data while red lines indicate anomalous behavior. Three types of anomalies were injected: sudden spikes (2–4× amplification), gradual drifts (2× increase over 10 time points), and stuck values (5-point duration). The figure demonstrates clear distinctions between normal operations and anomalous states — temperature anomalies reaching 140°C compared to normal 20°C–30°C fluctuations, pressure surges to 350 kPa from the 100 kPa baseline, and vibration spikes up to 9 mm/s from the normal 2 mm/s range. The LSTM-RNN model trained on this dataset achieved consistent improvement in performance, with the loss value decreasing from 0.1450 to 0.0312. These results demonstrate our DT framework's capability to detect various anomalies in multi-sensor industrial systems, providing robust support for reverse regulation.

8.4 Computational complexity analysis

While the security data processing and intelligent prediction technologies introduced in this scheme effectively enhance the security and reliability of data transmission and classification, they inevitably increase computational and communication overhead. From a quantitative perspective, let n denote the number of concurrent IoT device connections and m represent the number of data types. At the device-end encryption and signature phase, the typical operational complexity for each data item can be expressed as $O(T_{\mathrm{EXP}_{\mathbb{G}}} + T_{\mathrm{H}} + T_{\mathrm{EXP}_{\mathbb{Z}}} + T_{\mathrm{MUL}_{\mathbb{Z}}})$. When n devices upload simultaneously, the overall complexity exhibits growth of $O(n \cdot (T_{\mathrm{EXP}_{\mathbb{G}}} + T_{\mathrm{H}} + T_{\mathrm{EXP}_{\mathbb{Z}}} + T_{\mathrm{MUL}_{\mathbb{Z}}}))$. At the edge layer, data type verification and selective aggregation require multiple bilinear pairings and exponential operations for each data item, resulting in a complexity of approximately $O(n \cdot m \cdot (T_{\mathrm{BP}} + T_{\mathrm{MUL}_{\mathbb{G}}} + T_{\mathrm{EXP}_{\mathbb{G}}}))$. During data decryption and verification at the DT layer, the computational cost increases linearly with the

2) https://www.torontohydro.com/SITES/ELECTRICSYSTEM/BUSINESS/YOURBILLOVERVIEW/NETSYSTEMLOADSHAPE.

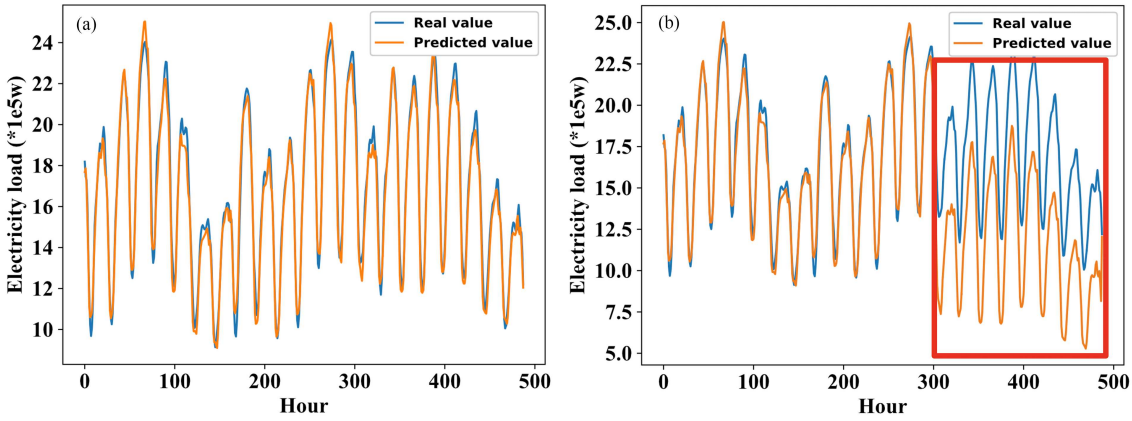

Figure 5 (Color online) Electricity load prediction and monitoring. (a) Normal IoT devices; (b) abnormal IoT devices.

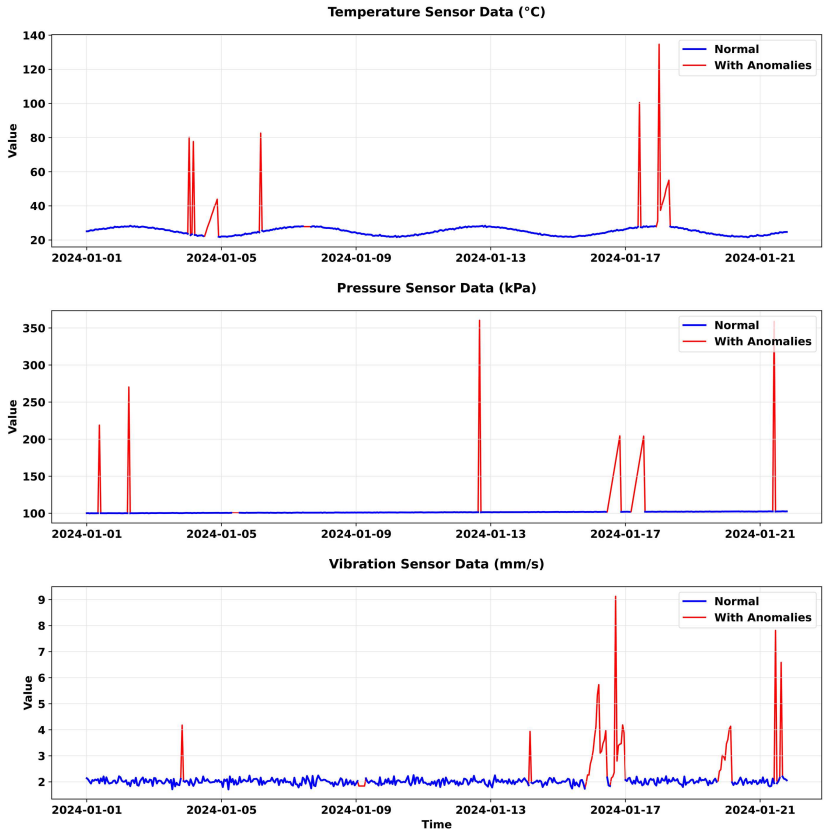

Figure 6 (Color online) DT simulation of industrial sensor data with anomaly injection.

number of data types m, approximated by $O(m \cdot (T_{\mathrm{BP}} + T_{\mathrm{MUL}_{\mathbb{G}}} + T_{\mathrm{EXP}_{\mathbb{G}}} + T_{\mathrm{MUL}_{\mathbb{Z}}}))$. Furthermore, while the prediction and detection capabilities of the LSTM-RNN model enhance anomaly identification, they introduce additional computational costs. The complexity can be expressed in terms of model parameters and layers as $O(d^2 \cdot T_{\mathrm{L}})$, where d represents a composite indicator of model depth and parameter scale.

In conclusion, while achieving enhanced security and intelligent monitoring, the system must carefully balance key length, encryption algorithm complexity, data type scale, and model depth. This balance aims to minimize computational and communication load while meeting the security and real-time requirements of the application scenario.

Table 5 Performance comparison of DT security frameworks.

Framework	Data size (bytes)	Encryption time (ms)	Decryption time (ms)	Error rate (%)	Throughput (bytes/s)
Our scheme	1024	0.117	1.148	8.00	802869
	4096	0.121	1.139	8.00	3200136
	16384	0.168	1.219	6.00	11284134
[32]	1024	2.974	2.644	4.00	181840
	4096	3.142	2.819	2.00	684118
	16384	3.061	2.634	14.00	2843581
[33]	1024	1.203	2.363	8.00	286037
	4096	1.193	2.368	8.00	1142640
	16384	1.306	2.491	14.00	4242351
[34]	1024	1.386	2.645	12.00	253118
	4096	1.200	2.375	8.00	1138237
	16384	1.332	2.505	8.00	4199019

8.5 User experience and system usability analysis

While our framework enhances security and monitoring capabilities for DT systems, we carefully optimize the impact on end-users' experience and system usability. Based on our industrial deployment experience, we analyze this impact across three primary user categories.

For IoT device operators, our framework maintains a transparent security mechanism with only a one-time device registration process. All subsequent security operations are handled automatically, allowing operators to maintain their familiar operational workflow without additional complexity.

System administrators benefit from a centralized management interface with automated tools for key distribution, certificate management, and security policy enforcement. The system provides real-time monitoring and automated alerts, enabling efficient security management with minimal manual intervention.

For data analysts and engineers, our framework implements role-based access control while ensuring transparent data access. The encryption processes run seamlessly in the background, allowing analysts to focus on data interpretation through an intuitive interface rather than security procedures.

These results demonstrate our framework's ability to enhance security while maintaining system usability across different user roles.

9 Benchmarking

9.1 Experimental setup

We conducted a comprehensive performance evaluation of our proposed scheme against three state-of-the-art DT security frameworks, including a synchronized data verification scheme [32], a privacy-preserving vehicular network solution [33], and a decentralized authentication framework [34]. While these approaches differ in their protection goals and architectural designs, we extract and simplify their core cryptographic components for fair performance comparison. All experiments were implemented in Python 3.10 using PyCryptodome for cryptographic operations and Python-Paillier for homomorphic encryption. The evaluation was performed with data sizes of 1024, 4096, and 16384 bytes, executing 50 test runs per size with a 10% error injection probability. To ensure meaningful security testing, we implemented controlled error injection targeting cryptographic tags rather than random data corruption. Moreover, our evaluation framework measures encryption time (average duration required for data encryption), decryption time (average duration required for data recovery), error rate (percentage of detected errors under controlled injection), and throughput (total data processing rate in bytes per second).

9.2 Results and analysis

Table 5 [32–34] presents the empirical results across all evaluated frameworks. The experimental data reveal several significant findings.

Computational efficiency. The empirical results demonstrate that our scheme achieves substantial improvements in computational efficiency. Specifically, the encryption latency exhibits a reduction factor of 15–25× compared to [32] and 8–10× compared to [33, 34]. The encryption performance maintains remarkable stability across all data sizes (0.117–0.168 ms), whereas competing frameworks demonstrate higher variability (1.2–3.1 ms). Similarly, the decryption process demonstrates consistent performance optimization, maintaining latencies of 1.1–1.2 ms compared to 2.3–2.8 ms in alternative frameworks.

System throughput. Analysis of system throughput reveals significant scalability advantages, with our scheme achieving processing rates of 802869 bytes/s, 3.2 MB/s, and 11.2 MB/s for increasing data sizes. These results represent a performance improvement factor of 4.4× over [32] and 2.7× over [33, 34] at maximum data capacity.

Security robustness. Our scheme achieves stable error detection rates of 6%–8% across all data sizes, outperforming the variable rates observed in competing frameworks: [32] (2%–14%), [33] (8%–14%), and [34] (8%–12%). This consistency demonstrates the robust security characteristics of our approach.

Note that these good performance metrics in our scheme stem from two key optimizations: the precomputation strategy for common data sizes and a carefully balanced 512-bit Paillier cryptosystem, which together enable significant efficiency improvements while maintaining strong security guarantees for real-time DT operations.

10 Conclusion and future work

We have described our proposed secure data transmission and classification framework designed for the DT environment. Given the varying distribution of IoT devices that underpin typical DT systems, we used edge servers to aggregate data type of interest in the encrypted form and filter irrelevant data. Subsequently, DT verifies the data correctness and performs decryption for network formulation. Moreover, based on historical device statistics, we designed an LSTM-RNN-based data trend prediction model to help DT locate the malfunctioning device more efficiently. Finally, the security and efficiency of our framework were demonstrated through theoretical analysis and experiments. One future extension to this work is to include a dynamic incentive mechanism to reward IoT devices in submitting accurate data. Additionally, enhancing the scalability of the framework for larger IoT deployments and extending the LSTM-RNN model to support more sophisticated real-time anomaly detection patterns would further strengthen its practical utility.

Acknowledgements This work was supported in part by Science and Technology Program of Guizhou Province (Grant No. [2023]434), National Natural Science Foundation of China (Grant Nos. 62272124, U1836205), and National Key Research and Development Program of China (Grant No. 2022YFB2701400).

References

1 Groshev M, Guimarães C, Martín-Pérez J, et al. Toward intelligent cyber-physical systems: digital twin meets artificial intelligence. IEEE Commun Mag, 2021, 59: 14–20
2 Khan L U, Saad W, Niyato D, et al. Digital-twin-enabled 6G: vision, architectural trends, and future directions. IEEE Commun Mag, 2022, 60: 74–80
3 Portela R, Varsakelis C, Richelle A, et al. When is an in silico representation a digital twin? A biopharmaceutical industry approach to the digital twin concept. In: Digital Twins. Berlin: Springer, 2020. 35–55
4 Glaessgen E, Stargel D. The digital twin paradigm for future NASA and US Air Force vehicles. In: Proceedings the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2012
5 Lee D, Lee S H. Digital twin for supply chain coordination in modular construction. Appl Sci, 2021, 11: 5909
6 Glatt M, Sinnwell C, Yi L, et al. Modeling and implementation of a digital twin of material flows based on physics simulation. J Manufacturing Syst, 2021, 58: 231–245
7 Karakra A, Fontanili F, Lamine E, et al. Pervasive computing integrated discrete event simulation for a hospital digital twin. In: Proceedings of the 15th International Conference on Computer Systems and Applications, 2018. 1–6
8 Alcaraz C, Lopez J. Digital twin: a comprehensive survey of security threats. IEEE Commun Surv Tut, 2022, 24: 1475–1503
9 Hu X Y, Wan Z, Huang K Z, et al. Modulated symbol-based one-time pad secure transmission scheme using physical layer keys. Sci China Inf Sci, 2024, 67: 112303
10 Githens M. Product Lifecycle Management: Driving the Next Generation of Lean Thinking. Hoboken: Wiley, 2007
11 Grieves M, Vickers J. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. In: Transdisciplinary Perspectives on Complex Systems. Berlin: Springer, 2017. 85–113
12 Erikstad S O. Merging physics, big data analytics and simulation for the next-generation digital twins. In: Proceedings of High-Performance Marine Vehicles, 2017. 141–151
13 Riemer D. Feeding the digital twin: basics, models and lessons learned from building an IoT analytics toolbox (invited talk). In: Proceedings of IEEE International Conference on Big Data, 2018
14 Steinmetz C, Rettberg A, Ribeiro F G C, et al. Internet of Things ontology for digital twin in cyber physical systems. In: Proceedings of VIII Brazilian Symposium on Computing Systems Engineering (SBESC), 2018. 154–159
15 Sleuters J, Li Y, Verriet J, et al. A digital twin method for automated behavior analysis of large-scale distributed IoT systems. In: Proceedings of the 14th Annual Conference System of Systems Engineering (SoSE), 2019. 7–12
16 Song E Y, Burns M, Pandey A, et al. IEEE 1451 smart sensor digital twin federation for IoT/CPS research. In: Proceedings of IEEE Sensors Applications Symposium (SAS), 2019. 1–6

17 Eckhart M, Ekelhart A. Towards security-aware virtual environments for digital twins. In: Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, 2018. 61–72
18 Bao J, Guo D, Li J, et al. The modelling and operations for the digital twin in the context of manufacturing. Enterprise Inf Syst, 2019, 13: 534–556
19 Kritzler M, Funk M, Michahelles F, et al. The virtual twin: controlling smart factories using a spatially-correct augmented reality representation. In: Proceedings of the 7th International Conference on the Internet of Things, 2017. 1–2
20 Uhlemann T H J, Schock C, Lehmann C, et al. The digital twin: demonstrating the potential of real time data acquisition in production systems. Procedia Manuf, 2017, 9: 113–120
21 Tao F, Zhang M, Liu Y, et al. Digital twin driven prognostics and health management for complex equipment. CIRP Ann, 2018, 67: 169–172
22 Liu Y, Zhang L, Yang Y, et al. A novel cloud-based framework for the elderly healthcare services using digital twin. IEEE Access, 2019, 7: 49088–49101
23 Korth B, Schwede C, Zajac M. Simulation-ready digital twin for realtime management of logistics systems. In: Proceedings of IEEE International Conference on Big Data, 2018. 4194–4201
24 Abideen A Z, Sundram V P K, Pyeman J, et al. Digital twin integrated reinforced learning in supply chain and logistics. Logistics, 2021, 5: 84
25 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, 1999. 223–238
26 Zhang J, Zhao Y, Wu J, et al. LVPDA: a lightweight and verifiable privacy-preserving data aggregation scheme for edge-enabled IoT. IEEE Int Things J, 2020, 7: 4016–4027
27 Shen H, Liu Y, Xia Z, et al. An efficient aggregation scheme resisting on malicious data mining attacks for smart grid. Inf Sci, 2020, 526: 289–300
28 Cui J, Wei L, Zhong H, et al. Edge computing in VANETs — an efficient and privacy-preserving cooperative downloading scheme. IEEE J Sel Areas Commun, 2020, 38: 1191–1204
29 Zhang X, Zhong H, Fan C, et al. CBACS: a privacy-preserving and efficient cache-based access control scheme for software defined vehicular networks. IEEE Trans Inform Forensic Secur, 2022, 17: 1930–1945
30 Ding Y, Wang B, Wang Y, et al. Secure metering data aggregation with batch verification in industrial smart grid. IEEE Trans Ind Inf, 2020, 16: 6607–6616
31 Sun L, Wang Y Y, Ren Y J, et al. Path signature-based XAI-enabled network time series classification. Sci China Inf Sci, 2024, 67: 170305
32 Li T, Wang H, He D, et al. Synchronized provable data possession based on blockchain for digital twin. IEEE Trans Inform Forensic Secur, 2022, 17: 472–485
33 Wang C, Ming Y, Liu H, et al. Secure and flexible data sharing with dual privacy protection in vehicular digital twin networks. IEEE Trans Intell Transp Syst, 2024, 25: 12407–12420
34 Patel C, Pasikhani A, Gope P, et al. User-empowered secure privacy-preserving authentication scheme for digital twin. Comput Secur, 2024, 140: 103793