LVPDA: A Lightweight and Verifiable Privacy-Preserving Data Aggregation Scheme for Edge-Enabled IoT

Jiale Zhang, *Student Member, IEEE,* Yanchao Zhao, *Member, IEEE,* Jie Wu, *Fellow, IEEE,* and Bing Chen, *Member, IEEE*

***Abstract*—Edge computing is envisioned to be a powerful platform that provides efficient data storage and computation services in smart IoT systems. In this data-intensive architecture, protecting user-side data privacy is one of the most critical concerns to prevent privacy leakage from any other untrusted entities. Aiming to resist this concern, lots of privacy-preserving data aggregation (PPDA) schemes have been proposed for various cloud-enabled IoT applications. However, due to the resource-constraint nature of smart IoT devices, the conventional PPDA solutions, in terms of both privacy and performance requirements, are unsuitable in edge computing. To address this challenge, we propose a lightweight and verifiable privacy-preserving data aggregation scheme, named LVPDA, for the edge computing enabled IoT system, where the Paillier homomorphic encryption method and online/offline signature technique are combined to ensure the privacy-preserving and integrity verification during the data aggregation process. Detailed security analysis indicates that LVPDA is existentially unforgeable under the chosen message attack (EU-CMA) and the data integrity can be guaranteed with formal proof under q-Strong Diffie-Hellman (q-SDH) assumptions. Compared with other PPDA methods, our scheme can achieve lightweight privacy-preserving data aggregation in terms of less computational complexity and communication overhead.**

***Index Terms*—Edge computing, Privacy-preserving, Data aggregation, Paillier cryptosystem, Online/offline signature.**

I. Introduction

WITH the widely deployed Internet of Things (IoT) infrastructures, the IoT technologies have shown great potential in smart services like smart grid [1, 2], smart healthcare [3, 4], smart city [5, 6], and vehicular sensing system [7]. However, the conventional cloud-based data processing paradigm [8] could hardly meet the requirements of these smart services, especially those serving in the real time manner, due to the bandwidth limitation and computation resources constraint [9] in the edge. To realize these envisions, the computation paradigm of IoT is developing in the track of edge computing[10], rather than traditional cloud computing, for supporting the real-time processing of the big sensing data generated by IoT devices. As shown in Fig. 1, the sensing data are gathered by the smart IoT devices and forwarded to the edge server for the local processing, such as aggregation, sharing, and mining. Then, the locally processed data are sent to the remote cloud center for further processing and analysis, providing various data services for IoT applications. Here, the edge server can be seen as a preliminary processing unit to provide efficient local services through the combination of the cloud server [11]. In this way, the resources of computation and communication can be significantly reduced, overcoming the bottleneck of conventional cloud-based architecture.

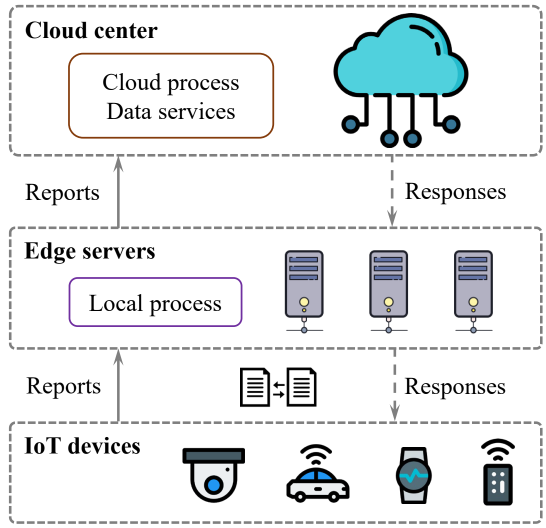


Fig. 1. Edge computing enhanced privacy-preserving data aggregation

Although the edge computing IoT system is beneficial for big data analysis, potential security and privacy risks are still present since the distributed nature of edge computing also enhances the activity of internal and external attackers [12]. Firstly, these additional edge nodes are not fully trusted, which might leak users' private data and thus destroy the privacy, accuracy, and robustness of the data aggregation protocol [13]. For example, in edge-cloud and smart grid systems [14], customers frequently transmit their sensitive data, e.g, electricity usage information, to the central server in order to benefit from centralized services, while these data usually contain users' privacy information [15]. Moreover, the external attackers could also eavesdrop on the communication channel among the involved entities, so as to modify the in-network messages, forge the signatures, or even launch a replay attack to compromise the normal data transmission procedure.

To solve this privacy issue, lots of privacy-preserving data aggregation (PPDA) schemes have been proposed to prevent privacy leakage from the untrusted entities [16–22]. Most of them are using the homomorphic cryptosystem to realize specific functions, such as Min, Max, and Sum, which can

Jiale Zhang, Yanchao Zhao and Bing Chen are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China and Science and Technology on Avionics Integration Laboratory, China e-mail: ({jlzhang; yczhao; cb_china}@nuaa.edu.cn).

Jie Wu was with the Center for Networked Computing, Temple University, Philadelphia 19122, USA e-mail: (jiewu@temple.edu).

Corresponding authors: Yanchao Zhao and Bing Chen.

guarantee data confidentiality and further preserve privacy. Li et al. [16] present the first data aggregation scheme for smart grid systems by using a homomorphic cryptosystem. Following this work, many embedded functionalities were explored to enhance the security and availability of PPDA, such as the high-dimensional reduction [17], key evolution technique [18], resisting internal attackers [19], data integrity verification [20], random noisy technique [21] and so on.

However, the aforementioned PPDA schemes are facing several practical challenges. Firstly, frequent data transmission requirements are crucial to edge computing IoT systems. It is impractical to tolerate high communication delay when executing real-time data processing tasks. Secondly, data source authentication and verification are necessary to prevent the attackers from forging, modifying, and replaying the messages and signatures. At last, the huge computation requirements of the authentication and verification operations greatly hinder their realization in the resource-constraint IoT devices. Therefore, it is highly desirable to design a novel PPDA scheme that can reduce the computational overheads on mobile devices while still fulfilling the data privacy requirements.

In this paper, to address the above challenges, we propose LVPDA: a Lightweight and Verifiable Privacy-preserving Data Aggregation scheme for edge computing enabled smart IoT systems, which simultaneously supports the data source authentication and lightweight verification. In LVPDA, the heavy computation cost of data integrity operations can be significantly reduced by an online/offline signature mechanism, hence is more suitable for the resource-constraint smart IoT devices. The main contributions of this paper are summarized as follows:

- We adopt the edge computing enabled IoT architecture to improve the computational efficiency of data aggregation and meanwhile present the corresponding PPDA framework.
- We further propose a novel lightweight and verifiable PPDA scheme based on the designed edge-enabled IoT system, called LVPDA, where the time-consuming operations are securely outsourced to the edge servers, so as to reduce the computing burden of the smart IoT devices.
- We give the detailed security analysis to show how our proposed LVPDA scheme can achieve data integrity, authentication, confidentiality and privacy-preserving under our defined security model.
- We conduct the exhaustive experiments of LVPDA and the results indicate that the computation and communication overheads are significantly reduced.

The remainder of this paper is going to be structured in the following way. We summarize the related work in Section II and introduce the preliminaries in Section III. The system model and design goals are introduced in Section IV. The description of our proposed LVPDA is detailed in Section V. Security and performance analysis of the proposed scheme are demonstrated in Section VI and Section VII, respectively. Finally, we conclude the paper in Section VIII.

II. Related Work

Privacy-preserving data aggregation (PPDA) has attracted more and more attention among different fields in recent years, such as smart grids, vehicular sensing systems, and other related smart IoT systems. Some previous works [16–22] have studied secure data aggregation in smart grids by using homomorphic cryptosystem. Specifically, in [16–18], the authors aim to find an efficient way to successfully construct PPDA schemes through different kinds of homomorphic encryption methods, i.e., additive and multiplicative homomorphism. After that, to improve the privacy guarantee, Fan et al. [19] presented a privacy-enhanced PPDA scheme by adding the blinding factors in the encryption step, which can resist in the internal attackers. In terms of enhancing security property against a malicious aggregator, Ni et al. [20, 21] introduced the trapdoor hash function and random noisy technique into PPDA schemes, achieving data integrity during the ciphertexts transmission phase. However, these schemes mainly focus on the privacy-preserving, reliability, communication overhead, and some other related functionalities, while the computation costs brought by cryptosystem operations are ignored.

Recently, as the research goes deep, researchers are devoted to reducing the computation costs of cryptographic-related operations in the conventional PPDA schemes. Abdallah et al. [23] proposed a lightweight security and privacy-preserving scheme by predicting the expected electricity demand for a cluster of houses in the smart grid system. This scheme can efficiently satisfy the security and privacy requirements and further reduce the communication overhead. Lu et al. [24] presented a lightweight privacy-preserving data aggregation scheme for the fog computing-enhance IoT system by extending [17], which can compress multidimensional data into one composite and early filter the injected false data at the fog node. Xu et al. [25] proposed a privacy-preserving data classification and aggregation scheme for vehicular sensing systems, which is the first work to resist data link attack and ensure data security. Most recently, Guan et al. [26] presented an anonymous PPDA scheme for fog-enhanced IoT by assuming multiple authorities to certify IoT devices and fog nodes locally. In our previous work [27], we presented an online/offline signature and verification method based on the double trapdoor Chameleon hash function to reduce the computational costs of data integrity mechanism, which can be seen as a basic theoretical exploration of our LVPDA scheme.

However, there are almost no works aiming to reduce the high computational complexity in data integrity mechanisms, especially the signature and verification operations. Hence, we propose a novel lightweight privacy-preserving data aggregation scheme for the edge computing enabled IoT system that can achieve lightweight integrity verification while protecting users' data privacy.

III. Preliminaries

In this section, we briefly introduce several definitions and notations used in our proposed LVPDA scheme, including bilinear pairing, the Paillier homomorphic cryptosystem, online/offline signatures, and security definitions.

A. Bilinear Pairing Setting

We assume that $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups with the prime order p, and g is a generator of group $\mathbb{G}$. Consider a nodegnerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ satisfies the following properties [28]:

- *Bilinear:* For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
- *Nondegenerate:* The generator g of group $\mathbb{G}$ should satisfy $e(g, g) \neq 1_{\mathbb{G}_T}$.
- *Computable:* For any $u, v \in \mathbb{G}$, there exists an efficient algorithm to compute $e(u, v)$.

To prove the security of LVPDA, we recall the following complexity problems:

Definition 1. *(q-Strong Diffie-Hellman Problem (q-SDH)). Let $\mathbb{G}$ be a cyclic group of prime order p, g be a generator of $\mathbb{G}$, and x be a random element in $\mathbb{Z}_p^*$. For a given $(q+1)$-tuple $(g, g^x, g^{(x^2)}, ..., g^{(x^q)})$, q-SDH problem is to calculate a pair (m, Σ_x) where $m \in \mathbb{Z}_p^*$. We define the q-SDH as (q, t, ϵ)-hard problem, only if the following equation holds for any t-time adversary $\mathcal{A}$.*

$$Pr[\mathcal{A}(g, g^x, g^{(x^2)}, ..., g^{(x^q)}) = (m, \Sigma_x), m \in \mathbb{Z}_p^*] < \epsilon. \quad (1)$$

Theorem 1. *We define that the (q, t, ϵ)-SDH assumption holds if and only if the advantage to solve the q-SDH problem in $\mathbb{G}$ for any t-time algorithm is less than ϵ.*

Note that, the probability is decided by the random choice of x in $\mathbb{Z}_p^*$ and the random bits consumed by $\mathcal{A}$. The detailed proof of *Theorem 1* can be found in [29], so we skipped the detailed description.

B. Paillier Homomorphic Cryptosystem

To guarantee the data confidentiality during the aggregation process, we utilize the Paillier homomorphic cryptosystem, which can achieve additive homomorphism property. It can be described as the following three algorithms.

- *KeyGen:* Input two large primes (p, q), calculate the RSA modulus and Carmichael function as $n = pq$ and $\lambda = (p-1)(q-1)$. Define a function $L(u) = \frac{u-1}{n}$ and compute $\mu = \big(L(g^{\lambda} \mod n^2)\big)^{-1}$. Then, the key materials can be formed as $(pk, sk) = \{(n, g), (\lambda, \mu)\}$.
- *ENC:* For any plaintext $m \in \mathbb{Z}_n$, randomly generate a number r where $gcd(r, n) = 1$, the ciphertext can be calculated as $c = g^m \cdot r^n \mod n^2$.
- *DEC:* Given a ciphertext $c \in \mathbb{Z}_{n^2}^*$, the corresponding plaintext message can be recovered as $m = L(c^{\lambda} \mod n^2)\mu \mod n$.

Paillier cryptosystem is proved to be semantically secure a-gainst chosen plaintext attacks, which means the mathematical expression is to decide whether an integer s is an n-residue modulo n^2 for some composite n.

C. Online/Offline Signatures

In an online/offline signature and verification method, the whole procedure can be divided into online and offline phases, where the latter can be outsourced to an untrusted third party. Double Trapdoor Chameleon Hash (DTCH) function [30] is an efficient mathematical tool to achieve online/offline signatures. For a given large prime p_1 and a generator g_1 from $\mathbb{G}_{p_1}$, pick two trapdoor keys $y, z \in \mathbb{Z}_{p_1}^*$. Then the DTCH function can be computed as $H_{ch}(r, s, u) = g_1^r \cdot g_2^s \cdot g_3^u$, where $g_2 = g_1^y, g_3 = g_1^z$ and (r, s, u) are elements generated from the chameleon hash. Note that, the DTCH function carries the following properties:

- *Computable:* Given a public key $pk \in \mathbb{G}$ and an input triple $(r, \circ, \circ) \in \mathbb{Z}_p$, the DTCH function $H_{ch}(r, \circ, \circ)$ is computable in polynomial time.
- *Collision Resistance:* Without at least one of the trapdoor keys, it is infeasible to find two chameleon hash pairs $(r_1, s_1, u_1), (r_2, s_2, u_2)$ which satisfy $r_1 \neq r_2$ and $H_{ch}(r_1, s_1, u_1) = H_{ch}(r_2, s_2, u_2)$.
- *Trapdoor Collision:* Given the hash function H_{ch} and public/private key pair (pk, sk), also given a chameleon hash pair (r_1, s_1, u_1) and an additional message $r_2 \in \mathbb{Z}_p$, we want to find $s_2 \in \mathbb{Z}_p$ (or $u_2 \in \mathbb{Z}_p$) such that $H_{ch}(r_1, s_1, u_1) = H_{ch}(r_2, s_2, u_2)$. The value of s_2 (or u_2) can be calculated in polynomial time as $s_2 = \big((r_1 - r_2) + (u_1 - u_2)y + s_1 z\big)z^{-1}$ (or $u_2 = \big((r_1 - r_2) + (s_1 - s_2)y + u_1 z\big)z^{-1}$ when first choose a random u_2 (or s_2).

According to the above-described properties of DTCH function, the online/offline signature and verification method used in our scheme can be constructed using the "hash-sign-switch" method, which consists of the following algorithms.

- *Setup:* On input a security parameter 1^{λ}, the Setup algorithm returns a random verification (public) key Ver_{pk} and the corresponding signature (private) key Sig_{sk}.
- *Sign.off:* On input signature key Sig_{sk}, the offline signature algorithm returns an offline signature token Σ_{off} and the state information St.
- *Ver.off:* On input verification key Ver_{pk} and the offline signature Σ_{off}, the offline verification algorithm returns *accept* if Σ_{off} is valid; Otherwise, outputs *reject*.
- *Sign.on:* On input Sig_{sk}, the state information St and a message m, the online signature algorithm returns an online signature token Σ_{on}.
- *Ver.on:* On input Ver_{pk}, a message m, the online signature Σ_{on} and the offline signature Σ_{off}, the verification algorithm returns *accept* if Σ_{on} is valid; Otherwise, it outputs *reject*. The signature of m is defined as $\Sigma = (\Sigma_{off}, \Sigma_{on})$.

D. Security Definitions

Definition 2. *(Unforgebility). The security definition of an online/offline signature and verification mechanism is existential unforgeability under chosen message attacks (EU-CMA), which can be formalized as an adversary-challenger game. We assume that the adversary $\mathcal{A}$ can make multi-times queries to the online and offline signature oracles $(sig^{on}(sk, St_i, m_i), sig^{off}(sk))$, where st_i is the state information of the signer.*

By this way, the EU-CMA can be illustrated as follows [31]:

- *Initiation:* The Challenger $\mathcal{C}$ runs the key generation algorithm on input 1^k to obtain a pair of public/private key (pk, sk). Then, pk is given to the adversary $\mathcal{A}$.

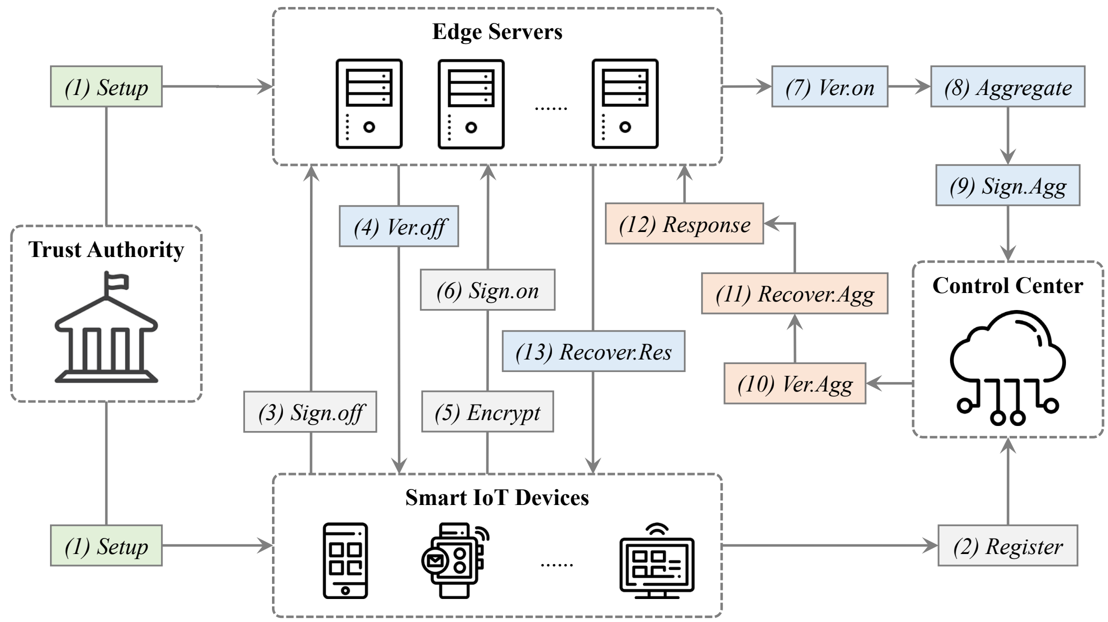


Fig. 2. System architecture

- *Sign.off Queries:* The adversary requests the i-th offline signature, and the challenger replies to the adversary with Σ_i^{off} while the state information St_i is stored by itself. Assume that the adversary can make q_1 queries at most in this phase.
- *Sign.on Queries:* The adversary requests the i-th online signature of message m_i, and the challenger $\mathcal{C}$ computes the online signature Σ_i^{on} using st_i and returns Σ_i^{on} to the adversary. Assume that the adversary can make q_2 queries at most in this phase.
- *Forgery:* The adversary $\mathcal{A}$ forges a message-signature pair (m^*, Σ^*), and sends it to the challenger $\mathcal{C}$. The challenger checks the validity of the signature by computing $Ver_{on}(pk, m^*, \Sigma^*)$, it outputs 1 (*success*) when the forged signature is valid; otherwise outputs 0 (*failure*).

The advantage in existentially forging a signature of the adversary $\mathcal{A}$ is:

$$Adv_{\mathcal{A}} = Pr\begin{bmatrix} Ver_{on}(pk, m^*, \Sigma^*) = 1 : (pk, sk) \leftarrow \\ KeyGen(1^k); (m^*, \Sigma^*) \leftarrow \mathcal{A}^{(\Sigma^{off}, \Sigma^{on})} \end{bmatrix}. \quad (2)$$

Where $\mathcal{A}$ has never requested the signature of m^* from the online signing oracle.

IV. System Model and Design Goals

In this section, we present the system model, workflow of LVPDA scheme, security model and design goals.

A. System Model

The system model of the proposed LVPDA scheme is shown in Fig. 2, which consists of four entities: control center (CC) like cloud server, edge servers (ES), smart IoT devices (SD), and a trust authority (TA).

- *TA* bootstraps the whole system and distributes key materials as well as system parameters (see step 1 in Fig. 2). We assume that the communication channels between TA and other entities are secure to transmit private key information. After the setup phase, TA will turn to offline.
- *CC* can collect all the aggregated data packages from edge servers and further make some intelligent decisions. Then it sends the corresponding responses back to the edge servers (see step 10, 11, 12 in Fig. 2). CC also provides the registration service for smart IoTs.
- *ES* plays the role as aggregators to aggregate the encrypted data from SD and transmit the aggregated data and responses between CC and SD (see step 8, 9, 13 in Fig. 2). ES also executes the online/offline integrity verification phases (see step 4, 7 in Fig. 2).
- *SD* represents a set of smart IoT devices owned by users. The private data m_i are collected by SD through sensors on the registered devices and transmitted to the CC via ES in encrypted form (see step 2, 3, 5, 6 in Fig. 2).

Note that, since the SD are usually resource constrained equipment, the privacy-preserving data aggregation processes with high computational complexity, especially the cryptographic operations involved in data integrity mechanism, cannot be efficiently executed. This main drawback motivates us to explore a lightweight PPDA mechanism for edge computing enabled smart IoT systems.

B. Workflow of LVPDA

According to the above-described system model, the proposed LVPDA scheme can be divided into the following phases and algorithms:

1) System Initialization Phase.

- *Setup* $(k, k_1) \rightarrow (SP_{pub}, msk)$: on input two security parameters (k, k_1), it outputs the system parameters SP_{pub} and the master key msk.

2) Registration Phase.

- *Register* $(X_i, k_i) \rightarrow (\alpha_i, \beta_i)$: on input a random value X_i and a blind factor k_i, it outputs the verification public key Y_i and the knowledge of registration (α_i, β_i).
- *Sign.off* $(y, z, s_i, u_i) \rightarrow (St, H_{ch_i}, \Sigma_i^{off}, Ver_{on})$: on input two sets of random values (y, z) and integers (s_i, u_i), it outputs the state information St, DTCH function, offline signature Σ_i^{off}, and online verification key Ver_{on}.

3) Report Generation Phase.

- *Ver.off* $(Ver_{pk}, \Sigma_i^{off}) \rightarrow b_1$: on input Ver_{pk} and Σ_i^{off}, it outputs a bit $b_1 \in \{0, 1\}$, where $b_1 = 1$ indicates the result of offline verification is *accept* and $b_1 = 0$ represents *reject*.
- *Encrypt* $(PK_P, m_i, v_i) \rightarrow c_i$: on input the public key PK_P, a message m_i, and an integer v_i, it outputs a encrypted report c_i.
- *Sign.on* $(c_i, St, s_i{}') \rightarrow \Sigma_i^{on}$: on input c_i, St, and a number $s_i{}'$, it outputs the online signature Σ_i^{on}.

4) Report Aggregation Phase.

- *Ver.on* $(\Sigma_i^{on}, Ver_{on}) \rightarrow b_2$: on input Ver_{on} and Σ_i^{on}, it outputs a bit $b_2 \in \{0, 1\}$, where $b_2 = 1$ indicates the result of online verification is *accept* and $b_2 = 0$ represents *reject*.
- *Aggregate* $(c_i) \rightarrow c$: on input c_i, it outputs the aggregation result c.
- *Sign.Agg* $(X_j, c) \rightarrow (Y_j, \Sigma_{Agg})$: on input c and a random number X_j, it outputs the aggregation signature public key Y_j and the aggregation signature Σ_{Agg}.

5) Report Reading Phase.

- *Ver.Agg* $(Y_j, \Sigma_{Agg}) \rightarrow b_3$: on input Y_j and Σ_{Agg}, it outputs a bit $b_3 \in \{0, 1\}$, where $b_3 = 1$ indicates the verification result of the aggregation phase is *accept* and $b_3 = 0$ represents *reject*.
- *Recover.Agg* $(c) \rightarrow m$: on input c, it outputs the aggregated plaintext m.

6) Response Phase.

- *Response* $(e(g_1, g_1)^{\tilde{\alpha}}, \tilde{\beta}, Q, Y, M_R) \rightarrow (\tilde{C}_1, \tilde{C}_2, \tilde{C}_3)$: on input two random numbers $(\tilde{\beta}, Q)$, the respond public key $(e(g_1, g_1)^{\tilde{\alpha}}, Y)$ and the respond message M_R, it outputs respond ciphertexts $(\tilde{C}_1, \tilde{C}_2, \tilde{C}_3)$.
- *Recover.Res* $(ak_i, \tilde{C}_1, \tilde{C}_2, \tilde{C}_3) \rightarrow M_R$: on input the respond ciphertexts $(\tilde{C}_1, \tilde{C}_2, \tilde{C}_3)$ and the authorized key ak_i, it outputs the respond plaintext M_R.

C. Security Model

Security is crucial for the success of privacy-preserving data aggregation. In our security model, the TA and CC are assumed to be fully trusted. However, the trustworthy of ES may be semi-trusted or honest-but-curious. That is, it will not arbitrarily tamper with the user's sensitive data, but try to reveal the embedded private information during the aggregation procedure. Moreover, we also consider an external adversary $\mathcal{A}$ hidden in the communication channels, whose main purpose is to threaten the data integrity mechanism and steal the private data by launching the active attacks. On the one hand, $\mathcal{A}$ can eavesdrop on the data in transit or intrude the servers in ES and CC to steal the data in process. On the other hand, the adversary $\mathcal{A}$ could actively forge the signatures of data reports and further compromise the data integrity. In summary, our security model should satisfy the data confidentiality, authentication, integrity, and privacy-preserving simultaneously. The corresponding security analyses of our proposed LVPDA scheme are detailed in Section VI.

D. Design Goals

Based on the above-mentioned system and security models, the design goal of our scheme can be described as the following four objectives:

- *Confidentiality and privacy-preserving*: user's sensitive raw data should always remain in ciphertext form once it departs from the devices. Meanwhile, the internal adversary, such as the ES, cannot access any individual's data except with aggregated results.
- *Authentication and Integrity*: all the users that participate in our LVPDA system should be authorized as the legal participants by CC. Besides, adversaries cannot modify the data in transit and any illegal operations of data packets can be detected by the CC and ES.
- *Computation efficiency*: the complex computation operations on the smart IoT devices should be reduced as much as possible. In addition, high communication efficiency is also expected to handle the frequent aggregation requests.
- *Scalability*: the designed LVPDA scheme can be easily applied to other networking scenarios, such as the smart grid and vehicle sensing systems. In addition, the lightweight properties embedded in our LVPDA scheme can be perfectly inherited.

V. PROPOSED LVPDA SCHEME

This section presents the proposed lightweight and verifiable privacy-preserving data aggregation scheme, LVPDA, for edge computing enabled smart IoT systems, which consists of six phases: system setup, registration, report generation, report aggregation, report reading, and response. In addition, the correctness of the LVPDA scheme is given.

A. System Setup

When receiving the data aggregation request sent by CC, TA first generates the Paillier public and private key pair $(SK_P, PK_P) = \{(\mu, \lambda), (n, g)\}$ based on the homomorphic cryptosystem described in III-B. After that, TA chooses two security parameters (k, k_1) randomly and a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ of prime order p_1, where $|p_1| = k_1$. Then, TA further defines three one-way hash functions: $H_0 : \{0, 1\}^* \rightarrow \mathbb{G}$, $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_{p_1}^*$, $H_2 : \mathbb{G} \rightarrow \mathbb{Z}_{p_1}^*$, a Chameleon hash function $H_{ch} : \mathbb{Z}_{p_1}^* \rightarrow \mathbb{G}$, three random elements $\tilde{\alpha}, \tilde{x} \in \mathbb{Z}_{p_1}^*$, $Q \in \mathbb{G}$, and computes $e(g_1, g_1)^{\tilde{\alpha}}$, $\tilde{Y} = g_1^{\tilde{x}}$. In addition, we assume that the number of IoT devices in a certain aggregation request period is ω. At last, TA publishes the system parameters as

$$SP_{pub} = \begin{Bmatrix} p_1, n, g, \mathbb{G}, \mathbb{G}_T, e, g_1, \omega, \tilde{Y}, Q, \\ e(g_1, g_1)^{\tilde{\alpha}}, H_0, H_1, H_2, H_{ch} \end{Bmatrix}. \tag{3}$$

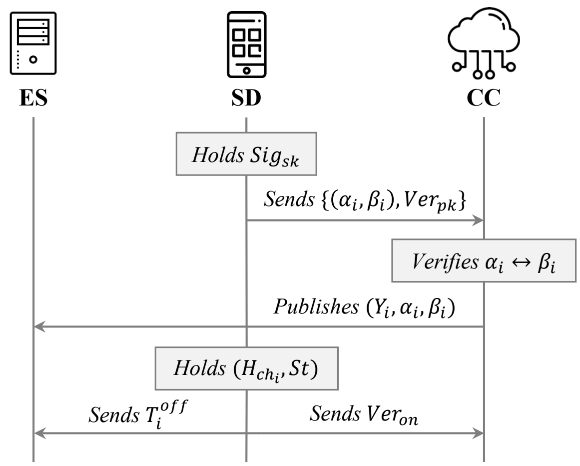


Fig. 3. The high level description of registration process

Correspondingly, the master private keys will be kept secret and sent to CC via a secure channel as

$$msk = (p, q, \lambda, \mu, \tilde{\alpha}, \tilde{x}). \tag{4}$$

B. Registration

When a user's smart device SD_i first participates in the LVPDA system, it is required to register to the CC for the purpose of authentication. Then, the offline signature step will be executed once the authentication succeeds. The registration process is shown in Fig. 3 and the descriptions are as follows.

- *User Registration:* SD_i first selects a value $X_i \in \mathbb{Z}_{p_1}^*$ and computes $Y_i = g_1^{X_i}$ based on the random signature method $Sig_{sk}()/Ver_{pk}()$. Correspondingly, the signature private key and verification public key can be formed as $(Sig_{sk}, Ver_{pk}) = (X_i, Y_i)$. Then, SD_i further picks a blinding factor $k_i \in \mathbb{Z}_{p_1}^*$ and calculates $r_i = H_1(ID_i||TS_i||k_i)$, where ID_i is SD_i's identity and TS_i is the current time slot. At last, SD_i generates the registration knowledge $\{\alpha_i = g_1^{r_i}, \beta_i = r_i - X_i H_2(\alpha_i)\}$ and sends $\{Y_i, \alpha_i, \beta_i\}$ to the CC.
- *Authentication:* Upon receiving $\{Y_i, \alpha_i, \beta_i\}$ from SD_i, CC verifies α_i by checking $\alpha_i = g_1^{\beta_i} Y_i^{H_2(\alpha_i)}$ based on the discrete logarithm problem. Once a user ω_i is successfully authenticated, CC firstly chooses a random number $t_i \in \mathbb{Z}_{p_1}^*$, and further generates the authorized user-related key ak_i to ω_i, where $ak_i = ({g_1}^{\tilde{\alpha}} \cdot Y^{t_i}, Q^{t_i}, {g_1}^{t_i})$. Then, it publishes $\{Y_i, \alpha_i, \beta_i\}$.
- *Offline Signature Generation:* SD_i firstly chooses two random values $y, z \in \mathbb{Z}_{p_1}^*$, and sets $g_2 = g_1^y$, $g_3 = g_1^z$. Without loss of generality, our LVPDA scheme would select the *BLS* signature method [32] Σ_{BLS} as the basic construction of the offline signature. SD_i also selects two integers $(s_i, u_i) \in \mathbb{Z}_{p_1}^*$ and stores $St = (r_i, s_i, u_i)$ as the state information, where $r_i = H_1(ID_i||TS_i||k_i)$. Then, SD_i calculates the DTCH function value as

$$H_{ch_i} = g_1^{r_i} \cdot g_2^{s_i} \cdot g_3^{u_i}, \tag{5}$$

and the *BLS* signature on H_{ch_i} can be formed as

$$\Sigma_i^{BLS} = \big(H_0(H_{ch_i})\big)^{X_i}, \tag{6}$$

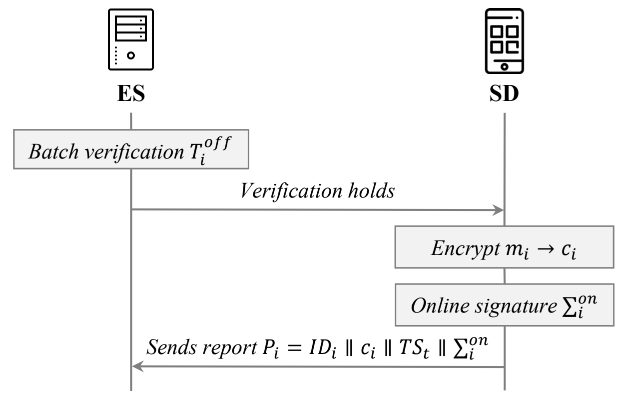


Fig. 4. The high level description of report generation phase

At last, SD_i sends the offline tag $T_i^{off} = (ID_i||TS_i||\Sigma_i^{off})$ to the ES, where $\Sigma_i^{off} = (\Sigma_i^{BLS}, H_{ch_i})$, and publishes the online verification key $Ver_{on} = (g_1, g_2, g_3)$ to the CC.

C. Report Generation

After receiving the offline tag T_i^{off} from SD_i, ES first executes the offline verification algorithm and SD_i sends the ciphertext along with online signature to ES once the offline signature is verified. Fig. 4 shows the report generation processes and the detailed steps are as follows.

- *Offline Signatures Batch Verification:* Upon ES receiving the offline tags from SD_i ($1 \le i \le \omega$), it verifies all signatures by checking if $e(g_1, \Sigma_i^{BLS}) = e\big(Y_i, H_0(H_{ch_i})\big)$ holds with the verification public key Ver_{pk}. To reduce the computation costs on repeatably verifying ω signatures, we utilize the batch verification method as

$$\begin{aligned}\prod_{i=1}^{\omega} e\big(Y_i, H_0(H_{ch_i})\big) &= \prod_{i=1}^{\omega} e\big(g_1^{X_i}, H_0(H_{ch_i})\big)\\ &= \prod_{i=1}^{\omega} e\big(g_1, (H_0(H_{ch_i}))^{X_i}\big) = \prod_{i=1}^{\omega} e(g_1, \Sigma_i^{BLS})\\ &= e(g_1, \prod_{i=1}^{\omega} \Sigma_i^{BLS}).\end{aligned} \tag{7}$$

If it does hold, the algorithm outputs *accept*, otherwise outputs *reject*.
- *Data Encryption:* Once the offline signature has been successfully verified, SD_i collects the sensitive data m_i and calculates the ciphertext based on the Paillier encryption mechanism as

$$c_i = g^{m_i} \cdot v_i^n \mod n^2, \tag{8}$$

where v_i is a randomly selected integer in $\mathbb{Z}_{n^2}^*$.
- *Online Signature Generation:* SD_i uses the state information $St = (r_i, s_i, u_i)$ to compute the online signature as

$${u_i}' = \big((r_i - c_i) + (s_i - {s_i}')y + u_i z\big)z^{-1}, \tag{9}$$

where ${s_i}' \in \mathbb{Z}_{p_1}^*$ and $\Sigma_i^{on} = ({s_i}', {u_i}')$. At last, SD_i sends its data report $P_i = ID_i||c_i||TS_t||\Sigma_i^{on}$ to ES nearby, where TS_t is the current timestamp.

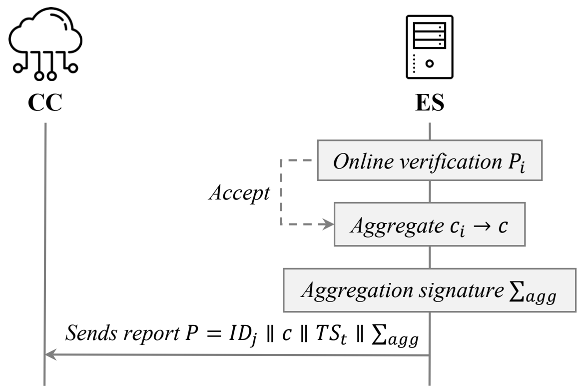


Fig. 5. The high level description of report aggregation phase

D. Report Aggregation

Upon ES receiving the users' reports P_i, $1 \leq i \leq \omega$, from SD, it adopts the online verification algorithm to check the validity of Σ_i^{on} and aggregates all the ciphertexts. The high-level illustration is shown in Fig. 5.

- *Online Signature Verification:* After receiving Σ_i^{on}, ES first checks the timestamp TS_t and uses Ver_{on} to verify the validity by checking whether $H_{ch}(r_i, s_i, u_i) = H_{ch}(c_i, s_i', u_i')$ holds or not. The correctness of online signature verification phase is shown as follows.

$$\begin{aligned} &H_{ch}(c_i, s_i', u_i') = g_1^{c_i'} \cdot g_2^{s_i'} \cdot g_3^{u_i'} \\ &= g_1^{c_i} \cdot (g_1^y)^{s_i'} \cdot g_1^{z\left((r_i - c_i) + (s_i - s_i')y + u_i z\right) z^{-1}} \\ &= g_1^{c_i} \cdot (g_1^y)^{s_i'} \cdot g_1^{r_i} \cdot g_1^{-c_i} \cdot g_1^{y \cdot s_i} \cdot (g_1^y)^{-s_i'} \cdot g_1^{z \cdot u_i} \\ &= (g_1)^{r_i} \cdot (g_1^y)^{s_i} \cdot (g_1^z)^{u_i} = g_1^{r_i} \cdot g_2^{s_i} \cdot g_3^{u_i} \\ &= H_{ch}(r_i, s_i, u_i). \end{aligned} \tag{10}$$

  this step outputs *accept* if the above equation holds, otherwise outputs *reject*.
- *Report Aggregation:* Once the online signature is verified, ES computes the aggregated ciphertext as

$$c = \prod_{i=1}^{\omega} c_i \mod n^2. \tag{11}$$

- *Aggregation Signature Generation:* ES randomly selects an aggregation signature private key $X_j \in \mathbb{Z}_{p_1}^*$ to generate the aggregation signature as

$$\Sigma_{Agg} = \left(H_0(ID_j||c||TS_t)\right)^{X_j}, \tag{12}$$

  where ID_j is the identity of a certain edge server ES_j. At last, ES_j sends the aggregated report $P = ID_j||c||TS_t||\Sigma_{Agg}$ to the control center.

E. Report Reading

When receiving P from ES_j, CC performs the following steps to read the aggregated result and sends the corresponding response to SD. The detailed description of report reading and the response phase was shown in Fig. 6.

- *Aggregation Signature Verification:* CC first verifies the received data report P by checking the validity of aggregation signature Σ_{Agg} as

$$\begin{aligned} e(g_1, \Sigma_{Agg}) &= e\big(g_1, (H_0(ID_j||c||TS_t))^{X_j}\big) \\ &= e\big(g_1^{X_j}, H_0(ID_j||c||TS_t)\big) \\ &= e\big(Y_j, H_0(ID_j||c||TS_t)\big) \end{aligned} \tag{13}$$

  where $Y_j = g_1^{X_j}$. If it does hold, the verification algorithm outputs *accept*, otherwise outputs *reject*.
- *Report Reading and Decryption:* Upon the aggregation signature has been verified, CC transforms the aggregated ciphertext c as

$$\begin{aligned} c &= \prod_{i=1}^{\omega} c_i \mod n^2 = \prod_{i=1}^{\omega} g^{m_i} \cdot v_i^n \mod n^2 \\ &= g^{\sum_{i=1}^{\omega} m_i} \cdot \prod_{i=1}^{\omega} v_i^n \mod n^2 \\ &= g^m \cdot \prod_{i=1}^{\omega} v_i^n \mod n^2. \end{aligned} \tag{14}$$

  Since the above-transformed ciphertext is also satisfied with the form of Paillier cryptosystem, thus CC can easily decrypt it and obtain the aggregated plaintext as

$$m = \sum_{i=1}^{\omega} m_i = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n. \tag{15}$$

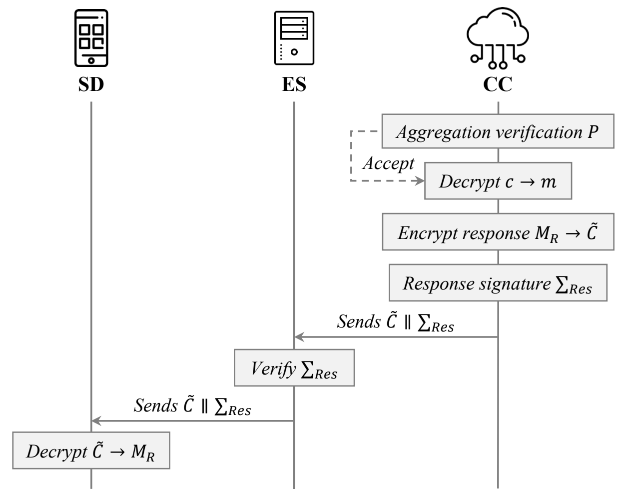


Fig. 6. The high level description of report reading and response phase

F. Response

After analyzing the aggregated plaintext m, the CC response with a message $M_R \in \mathbb{G}_T$ to the edge server in a certain coverage area. To guarantee the privacy of respond message, M_R should be transmitted under a ciphertext form. The concrete steps are performed as follows:

- *Step-1:* The CC firstly chooses a random number $\tilde{\beta} \in \mathbb{Z}_{p_1}^*$, and computes $\tilde{C} = (\tilde{C}_1, \tilde{C}_2, \tilde{C}_3)$, where

$$\begin{cases} \tilde{C}_1 = M_R \cdot e(g_1, g_1)^{\tilde{\alpha}\tilde{\beta}} \mod n, \\ \tilde{C}_2 = g_1{}^{\tilde{\beta}}, \tilde{C}_3 = (Y/Q)^{\tilde{\beta}}. \end{cases} \tag{16}$$

  Then, the CC makes the signature $\Sigma_{Res} = \left(H_0(\tilde{C}||TS_c)\right)^x$, where TS_c is the current time stamp, and sends back $\tilde{C}||\Sigma_{Res}$ to the edge server, which covered some smart IoT devices.

- *Step-2:* Upon receiving $\tilde{C}||\Sigma_{Res}$, the edge server verifies the validity of $\tilde{C}$ by checking whether $e(g_1, \Sigma_{Res}) = e\big(Y, H_0(\tilde{C}||TS_c)\big)$. If it does hold, the edge server broadcasts $\tilde{C}$ in its covered area.
- *Step-3:* After receiving the authenticated $\tilde{C}$ from the edge server, each user $\omega_i \in \omega$ uses the authorized key $ak_i = ({g_1}^{\tilde{\alpha}} \cdot Y^{t_i}, Q^{t_i}, {g_1}^{t_i})$ to recover M_R from $\tilde{C}$ in the followings:

$$\begin{aligned}
&\frac{e(\tilde{C}_2, {g_1}^{\tilde{\alpha}} \cdot Y^{t_i})}{e(\tilde{C}_2, Q^{t_i})e(\tilde{C}_3, {g_1}^{t_i})} = \frac{e({g_1}^{\tilde{\beta}}, {g_1}^{\tilde{\alpha}} \cdot Y^{t_i})}{e(\tilde{C}_2, Q^{t_i})e((Y/Q)^{\tilde{\beta}}, {g_1}^{t_i})} \\
&= \frac{e({g_1}^{\tilde{\beta}}, {g_1}^{\tilde{\alpha}})e({g_1}^{\tilde{\beta}}, Y^{t_i})}{e({g_1}^{\tilde{\beta}}, Q^{t_i})\big(e(Y^{\tilde{\beta}}, {g_1}^{t_i})/e(Q^{\tilde{\beta}}, {g_1}^{t_i})\big)} \\
&= \frac{e({g_1}^{\tilde{\beta}}, {g_1}^{\tilde{\alpha}})e({g_1}^{\tilde{\beta}}, Y^{t_i})}{e(Y^{\tilde{\beta}}, {g_1}^{t_i})} = e(g_1, g_1)^{\tilde{\alpha}\tilde{\beta}}.
\end{aligned} \tag{17}$$

$$\frac{\tilde{C}_1}{e(g_1, g_1)^{\tilde{\alpha}\tilde{\beta}}} = \frac{M_R \cdot e(g_1, g_1)^{\tilde{\alpha}\tilde{\beta}}}{e(g_1, g_1)^{\tilde{\alpha}\tilde{\beta}}} = M_R \tag{18}$$

When the recovered information M_R, ω_i can dynamically make the intelligent decisions while ensuring the privacy preservation.

VI. Security analysis

This section gives a detailed analysis of security properties followed by the security model and design goals described in Section IV. Especially, we mainly concentrate on the authentication, confidentiality and privacy-preserving, as well as integrity and unforgeability.

A. Authentication

In the proposed LVPDA scheme, we embed an authentication mechanism in the registration phase based on the extended Schnorr's signature method [33], which is proved to be secure under the discrete logarithm assumption. The correctness of authentication can be presented as follows.

$$g_1^{\beta_i} Y_i^{H_2(\alpha_i)} = g_1^{(r_i - X_i H_2(\alpha_i))} \cdot g_1^{X_i H_2(\alpha_i)} = g_1^{r_i} = \alpha_i. \tag{19}$$

Particularly, an attacker cannot forge the registration knowledge $\{\alpha_i, \beta_i\}$ without obtaining the real identity information ID_i of SD_i, since ID_i is protected by a secure one-way hash function H_1 and kept secretly. Moreover, even if the attacker can steal SD_i's real identifier ID_i, it still cannot get the hash function value r_i because r_i is further hidden by using a random selected blinding factor k_i, thus ensuring the security of the signature private key X_i. Therefore, the authentication between SD and CC is proved to be secure in our scheme.

B. Confidentiality and Privacy-preserving

In our LVPDA scheme, we utilize the Paillier cryptosystem to encrypt all the sensed data and aggregate the ciphertext based on the additively homomorphic property. The confidentiality and privacy of sensing data can be guaranteed for the following three aspects.

Firstly, in the report generation phase, SD_i's private data m_i are encrypted as $c_i = g^{m_i} \cdot {v_i}^n \mod n^2$, which is a standard ciphertext form of Paillier cryptosystem. Since the Paillier cryptosystem is proved to be semantically secure against the Chosen Plaintext Attack (CPA) based on the decisional Diffie-Hellman problem [34], no sensitive information will be leaked.

Secondly, in the report aggregation phase, ES cannot recover each individual's plaintext without the private key (λ, μ), but aggregate all the received ciphertexts as $c = g^{(\sum_{i=1}^{\omega} m_i)} \cdot (\prod_{i=1}^{\omega} v_i)^n \mod n^2$, which is still a valid ciphertext form of Paillier cryptosystem. Therefore, the users' data confidentiality and privacy can be ensured even when ES is untrusted.

Thirdly, imagine there exists an external attacker who can eavesdrop on the whole communication channel from SD to CC and obtain both the individual ciphertexts c_i, aggregated ciphertext c, and aggregated plaintext m, then he is still unable to recover the individual plaintext m_i, since all the plaintexts are compressed through the report aggregation process. In summary, the confidentiality and privacy of each individual SD_i's sensitive data can be perfectly protected.

C. Integrity and Unforgeability

In the proposed LVPDA, we designed an online/offline signature method to ensure the data integrity and meanwhile reduce the computation costs. Here, we prove that our scheme is existentially unforgeable under the chosen message attack (EU-CMA), thus guaranteeing the data integrity. According to the *Definition 2*, without querying the online signing oracle token on a given $m^* \in \mathbb{Z}_{p_1}^*$, an adversary $\mathcal{A}$ cannot forge any pair (m^*, Σ^*) to ensure the validity of signature Σ^* with private key $Sig_{sk}()$ in probabilistic polynomial time. Combined with *Definition 1* and *Theorem 1*, the problem can be transformed into proving the following theorem.

Theorem 2. *We say that an online/offline signature scheme is (t, q_1, q_2, ϵ) secure against EU-CMA if the q-SDH problem can be solved by an algorithm $\mathcal{B}$ in polynomial time with a non-negligible probability $\epsilon' \geq \frac{\epsilon}{3} - \frac{q_2}{p}$.*

Proof. We use the contradiction method to prove this theorem, assume that $\mathcal{A}$ queries the offline and online signature oracle on message m_i for q_1 and q_2 times respectively, where $q_2 = q \leq q_1$. Let $(\Sigma_{off}, \Sigma_{on})$ be the full signatures from the real online/offline signing oracle after q_2 queries by $\mathcal{A}$, and $\mathcal{A}$ returns a valid forgery signature $(\Sigma_{off}^*, \Sigma_{on}^*)$ on a new message m^* with probability of at least ϵ. Moreover, suppose $(g, g^{\tau}, g^{(\tau^2)}, \cdots, g^{(\tau^q)})$ is a q-SDH instance generated by algorithm $\mathcal{B}$, which aims to construct a new valid online/offline signature $(\Sigma_{off}^*, \Sigma_{on}^*)$ and successfully solve the q-SDH problem. In this way, the attacks from $\mathcal{A}$ fall into the following cases:

Case 1: $g^{m^*} g_2^{s^*} g_3^{u^*} \neq g^{m_i} g_2^{s_i} g_3^{u_i}$ for all $i \in \{1, \cdots, q_2\}$.

Case 2: $g^{m^*} g_2^{s^*} g_3^{u^*} = g^{m_i} g_2^{s_i} g_3^{u_i}$ for some $i \in \{1, \cdots, q_2\}$, and $s^* \neq s_i$.

Case 3: $g^{m^*} g_2^{s^*} g_3^{u^*} = g^{m_i} g_2^{s_i} g_3^{u_i}$ for some $i \in \{1, \cdots, q_2\}$, and $s^* = s_i$, but $u^* \neq u_i$.

[CASE 1.]

- *Initiation:* Algorithm $\mathcal{B}$ randomly chooses two numbers $y, z \in \mathbb{Z}_p^*$ and sets signature private key as $SK =$

(a, y, z). Then, it gives the verification public key $VK = (g, g_1, g_2, g_3)$ to $\mathcal{A}$, where $g_1 = g^a$, $g_2 = g^y$, $g_3 = g^z$.

- *Sign.off Queries:* Adversary $\mathcal{A}$ first takes the i-th offline query, where $1 \leq i \leq q_1$. Then, $\mathcal{B}$ responds with $\Sigma_i^{off} = \big(H_0(H_{ch_i})^a, H_{ch_i}\big)$ to $\mathcal{A}$ as the i-th offline signature, where $H_{ch_i} = g^{r_i} g_2^{s_i} g_3^{u_i} = g^{(r_i + s_i y + u_i z)}$ and $(r_i, s_i, u_i) \in \mathbb{Z}_p^*$ are stored by $\mathcal{B}$. Apparently, Σ_i^{off} is valid because $e\big(g, H_0(H_{ch_i})^a\big) = e\big(g_1, H_0(H_{ch_i})\big)$. For simplicity, we use c_i to represent $= r_i + s_i y + u_i z$.
- *Sign.on Queries:* Adversary $\mathcal{A}$ takes the i-th offline query, where $1 \leq i \leq q_2$. Correspondingly, $\mathcal{B}$ returns $\Sigma_i^{on} = (s_i', u_i')$ to $\mathcal{A}$ as the i-th offline signature, where $u_i' = \big((r_i - m_i) + (s_i - s_i')y + u_i z\big)z^{-1}$ and $s_i' \in \mathbb{Z}_p^*$. Also, the validity of Σ_i^{on} can be guaranteed by $H_{ch_i}(r_i, s_i, u_i) = H_{ch_i}(m_i, s_i', u_i')$.
- *Forgery:* Finally, $\mathcal{A}$ submits a valid forgery signature $(m^*, s^*, u^*, s_*', u_*')$ satisfying the condition in *Case 1*. Since $g^{m^*} g_2^{s^*} g_3^{u^*} \neq g^{m_i} g_2^{s_i} g_3^{u_i}$, then we have $c^* = m^* + s^* y + u^* z \neq c_i$, which means there exists an algorithm $\mathcal{B}$ to solve the q-SDH problem with probability at least $\epsilon/3$ (the same with *Case 1* occurred).

Note that, the only difference between *Case 1*, *Case 2*, and *Case 3* on the *Initiation*, *Sign.off Queries*, and *Sign.on Queries* phases is that the algorithm $\mathcal{B}$ forges a new Chameleon hash function value H_{ch}^* in *Case 1* while the trapdoor y and z are forged in *Case 2* and *Case 3*. Therefore, we only focus on the forging step in the subsequent two cases.

[CASE 2.]

- *Forgery:* In *Case 2*, one of the double trapdoors y is forged by $\mathcal{B}$, whose the signature private key is set as $SK = (x, a, z)$. As described above, we know that the probability of *Case 2* occurring is $\epsilon/3$ at least, and $s^* = s_i$ occurs with probability $1/p$ since s_i is randomly selected from $\mathbb{Z}_p^*$. Thus, for the whole game, $s^* = s_i$ occurs with probability of at most q_2/p. In this situation, once adversary $\mathcal{A}$ returns a forged signature $\big(m^*, \Sigma_{off}^*(a, r^*, s^*, u^*), \Sigma_{on}^*(s_*', u_*')\big)$ which fulfills the conditions in *Case 2*, then algorithm $\mathcal{B}$ can successfully calculate $a = y = \big((m^* - m_i) + (u^* - u_i)z\big)(s_i - s^*)^{-1}$. In other words, $\mathcal{B}$ can succeed with probability of at least $\epsilon/3 - q_2/p$ to solve the q-SDH problem.

[CASE 3.]

- *Forgery:* In *Case 3*, $\mathcal{B}$ forges another trapdoor z and sets the corresponding signature private key as $SK = (x, y, a)$, where the maximum probability of $u^* = u_i$ is q_2/p. Similar to *Case 2*, $\mathcal{B}$ can solve the q-SDH problem with probability of at least $\epsilon/3 - q_2/p$ in polynomial time by computing $a = z = \big((m^* - m_i) + (s^* - s_i)z\big)(u_i - u^*)^{-1}$ for some i, where $\big(m^*, \Sigma_{off}^*(a, r^*, s^*, u^*), \Sigma_{on}^*(s_*', u_*')\big)$ is a valid signature forged by $\mathcal{A}$ which meets the condition of *Case 3*.

In summary, there exists an algorithm $\mathcal{B}$ to solve the q-SDH problem with probability at least $\epsilon/3 - q_2/p$ in polynomial time. Correspondingly, *Theorem 2* is proved due to the contradictions between the reasoning result and the original q-SDH assumption. □

TABLE I
NOTATIONS IN EVALUATIONS

Notations	Descriptions	Time Cost (ms)
T_{E_1}	Exponentiation in $\mathbb{Z}_{n^2}$	1.58
T_{E_2}	Exponentiation in $\mathbb{G}$	1.62
T_M	Multiplication in $\mathbb{G}$	0.06
T_P	Pairing Operation	17.62

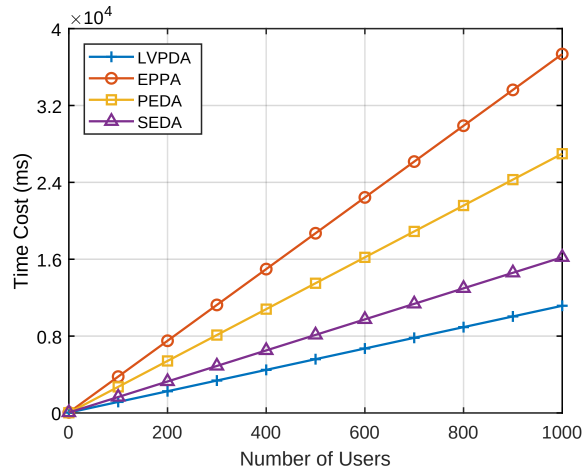


Fig. 7. Overall computational cost comparison

VII. NUMERICAL EVALUATION

This section evaluates the performance of the proposed LVPDA scheme, in terms of the computational complexity as well as communication overhead. As a comparison, we take three classic homomorphic cryptosystem based schemes into consideration, namely EPPA [17], PEDA [19], and SEDA [20], to demonstrate the efficiency of our scheme. Without loss of generality, we use the public Pairing-Based Cryptography (PBC) library to estimate the time costs operations in Paillier cryptosystem, in which the RSA modulus n is set to 1024 bits and the security parameter p_1 is 160 bits. All the experiments are implemented on a Linux machine with Intel Core i7-4710U CPU at 2.5GHz and 4.00 GB memory. The notations of cryptographic operations and corresponding time costs are shown in Table I.

A. Computational Complexity

For the proposed LVPDA scheme, the report generation of a new smart device SD_i requires two exponentiation operations in $\mathbb{Z}_{n^2}$ to generate ciphertext c_i, and three multiplication operation in $\mathbb{G}$ to compute the online signature Σ_i^{on}. In the report aggregation phase, ES needs to verify the online signature and further aggregates all the collected ciphertexts, which consumes three exponentiation operations in $\mathbb{G}$ and ω multiplication operations in $\mathbb{Z}_{n^2}$. Note that, the Hash operations and multiplication operations in $\mathbb{Z}_{n^2}$ are regarded as negligible compared to exponentiation and pairing operations. Then, ES also performs one exponentiation operation in $\mathbb{G}$ to generate the aggregation signature Σ_{Agg}. Upon receiving the aggregated report from ES_j, CC verifies Σ_{Agg} and decrypts the aggregated ciphertext c to obtain sum-plaintext which exe-

TABLE II
SIGNATURE AND VERIFICATION COMPUTATION COST COMPARISONS

Scheme	Cost
LVPDA	$2T_P + (3\omega+1)T_{E_2} + \omega T_M$
EPPA [17]	$(\omega+3)T_P + (\omega+1)T_M$
PEDA [19]	$(\omega+1)T_P + (2\omega+1)T_{E_2} + (\omega+1)T_M$
SEDA [20]	$2T_P + (6\omega+3)T_{E_2} + \omega T_M$

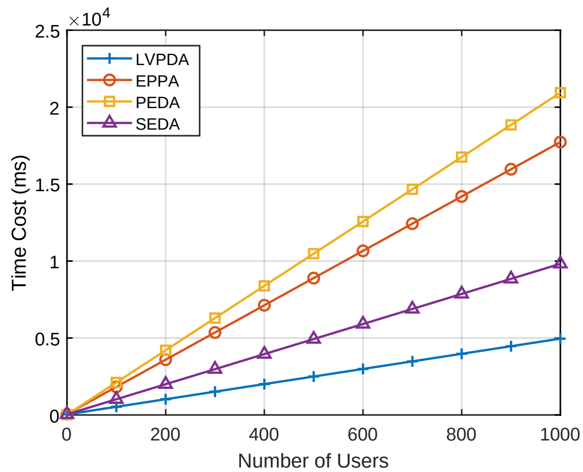


Fig. 8. Signature and Verification Cost Comparison

cutes two pairing operations and two exponentiation operations in $\mathbb{Z}_{n^2}$.

The above computation complexity analysis indicates that there are fewer time-consuming cryptographic operations required in terms of SD_i, especially the signature operations. The overall computational costs comparison in the four schemes are illustrated in Fig. 7. It shows that our proposed LVPDA scheme has significant efficiency in computational costs compared to the other three schemes [17, 19, 20] because a major part of complex operations is shifted into the offline phase. In particular, Table II shows the detailed computation cost comparisons on signature and verification method and Fig. 8 further depicts the change tendency of time cost among the four schemes. Obviously, the time cost of signature and verification in our LVPDA scheme is at least 50% lower than EPPA, PEDA, and SEDA. Furthermore, we also compare the computation costs in the aggregation phase and the result is demonstrated in Fig. 9, which shows that our scheme also has an advantage in aggregation computation costs comparison. In summary, the above evaluation results indicate that our scheme is more efficient than the other three schemes in terms of signature, verification, aggregation, and overall computation costs. However, it requires sufficient computational resources on the registration phase since the system needs to execute the authentication and offline signature operations.

B. Communication Overhead

According to our system model described in IV-A, the communication interactions involved in our LVPDA scheme fall into two phases: one phase is from smart devices SD to edge server communication, noted as SD-to-ES, and the other phase is from edge server ES to control center CC

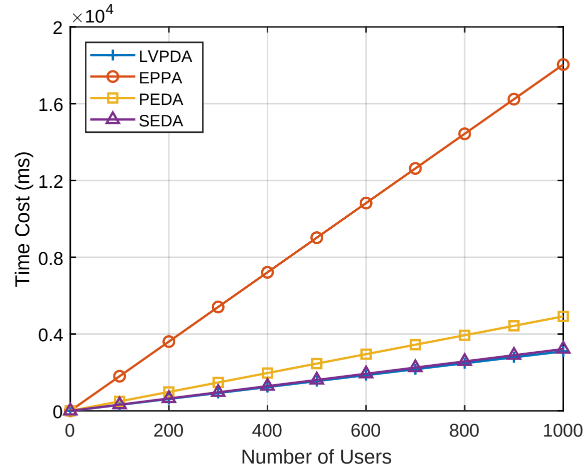
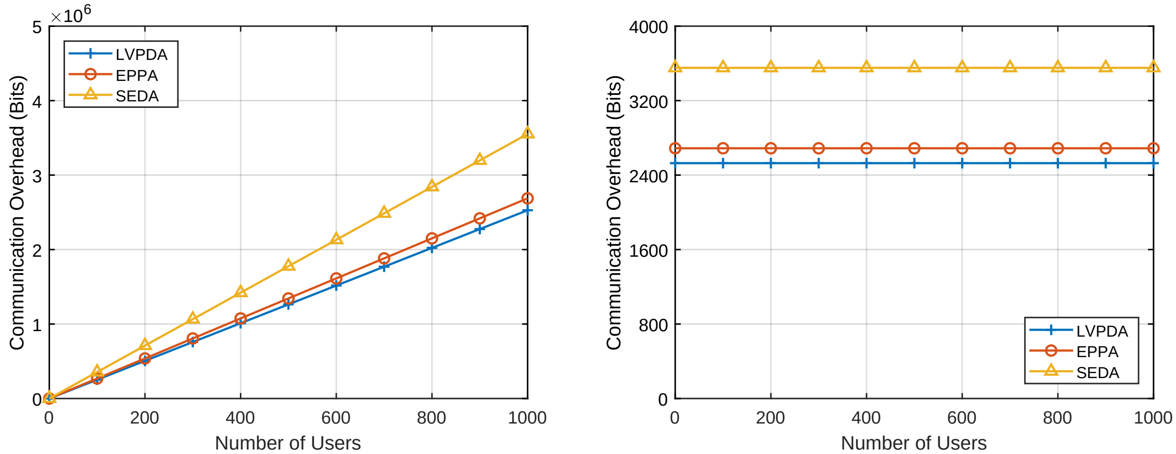


Fig. 9. Aggregation Cost Comparison

communication, abbreviated as ES-to-CC. In the SD-to-ES phase, the data report generated by SD_i is sent to the target ES_j which can be formed as $P_i = ID_i||c_i||TS_i||\Sigma_i^{on}$. Here, we set the RSA modulus $n = 1024$ bits and security parameter $p_1 = 106$ bits and the size of data report should be $S_{SD_i} = |ID_i| + 2048 + |TS_i| + 160$ bits. Considering there exist a total of ω users to participate in our LVPDA system in a certain time slot, thus the overall communication overheads are $S_{TS} = \omega S_{SD_i}$. In the second communication phase, ES_j aggregates ω users' data reports and sends the aggregated report $P = ID_j||c||TS_t||\Sigma_{Agg}$ to CC, where the report size is $S_{SC} = |ID_j| + 2048 + |TS_t| + 160$ bits. Note that, the aggregation mechanism can significantly reduce the communication overhead compared with the conventional cloud-based data transmission scenario where each individual's data report is separately transmitted to the CC and the total data size is $(|ID_j| + 2048 + |TS_t| + 160) * \omega$ bits. Since the PEDA scheme [19] does not consider the perspective of communication overhead, we mainly focus on the EPPA [17], SEDA [20], and our proposed LVPDA scheme. Fig. 10 presents the communication overhead comparison on both SD-to-ES and ES-to-CC phases, where the size of $|ID|$ and $|TS|$ is set to be 160 bits. The results indicate that the LVPDA scheme is indeed more efficient than the other two schemes. Particularly, we can see that the evaluation results shown in Fig. 10(b) are close to the constant, which is mainly because the communication overheads in the ES-to-CC phase have no correlation with the number of users.

VIII. CONCLUSION

In this paper, we present a lightweight and verifiable privacy-preserving data aggregation scheme for smart IoT systems, named LVPDA, which simultaneously achieves the authentication, lightweight integrity verification, confidentiality, and privacy-preserving. The scheme exploits the Paillier homomorphic cryptosystem and online/offline signature method to significantly reduce the computation and communication costs of conventional PPDA schemes. Moreover, benefiting from the edge computing, LVPDA can efficiently shift the time-consuming cryptographic operations to the edge server



(a) (SD-to-ES Overhead)

(b) (ES-to-CC Overhead)

Fig. 10. Communication overhead comparison

and meanwhile minimize the online computation costs. Due to the efficiency property, our designed LVPDA scheme can be used in lots of smart IoT systems, such as the smart grid and vehicular network. Thorough security analysis illustrated that the proposed scheme is secure under our defined security model. Extensive evaluation results demonstrated the lightweight and effectiveness of LVPDA. However, our method, to an extent, is vulnerable to collusion attacks launched by edge servers and malicious users. In regard to future work, we plan to further improve the security properties under more powerful adversaries and active attack models.

ACKNOWLEDGMENT

This work was supported in part by National Key R&D Program of China under Grant 2019YFB2102000, in part by the National Natural Science Foundation of China under Grant 61672283 and Grant 61602238, in part by the Science and Technology on Avionics Integration Laboratory and the National Aeronautical Science Foundation of China under Grant 20175552039, in part by the Postgraduate Research&Practice Innovation Program of Jiangsu Province under Grant KYCX18_0308, and in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

REFERENCES

[1] Z. Erkin, J. R. Troncoso-Pastoriza, R. L. Lagendijk, and F. Pérez-González, "Privacy-preserving data aggregation in smart metering systems: An overview," *Computer networks*, vol. 30, no. 2, pp. 75–86, 2013.

[2] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, 2017.

[3] L. Catarinucci, D. De Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An iot-aware architecture for smart healthcare systems," *IEEE Internet of Things Journal*, vol. 2, no. 6, pp. 515–526, 2015.

[4] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, no. part 2, pp. 641–658, 2018.

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[6] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the internet of things in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 84–91, 2017.

[7] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE VEHICULAR TECHNOLOGY MAGAZINE*, vol. 12, no. 2, pp. 36–44, 2017.

[8] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[9] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[11] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[12] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, no. part 2, pp. 680–698, 2018.

[13] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed privacy-preserving data aggregation against dishonest nodes in network systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1462–1470, 2019.

[14] X. Gong, Q.-S. Hua, L. Qian, D. Yu, and H. Jin, "Communication-efficient and privacy-preserving data aggregation without trusted authority," in *IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, April*, 2018, pp. 1250–1258.

[15] J. Zhang, B. Chen, Y. Zhao, X. Chen, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.

[16] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *IEEE International Conference on Smart Grid Communications, SmartGrid-*

Comm 2010, Gaithersburg, MD, Oct, 2010, pp. 13–16.
[17] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.
[18] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "Eppdr: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2014.
[19] C.-I. Fan, S.-Y. Huang, and Y.-L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial informatics*, vol. 10, no. 1, pp. 666–675, 2014.
[20] J. Ni, K. Alharbi, X. Lin, and X. Shen, "Security-enhanced data aggregation against malicious gateways in smart grid," in *IEEE Global Communications Conference, GLOBECOM 2015, San Diego, CA, USA, Dec*, 2015, pp. 1–6.
[21] J. Ni, K. Zhang, X. Lin, and X. Shen, "Edat: Efficient data aggregation without ttp for privacy-assured smart metering," in *IEEE International Conference on Communications, ICC 2016, Kuala Lumpur, Malaysia, May*, 2015, pp. 1–6.
[22] Z. Guan, Y. Zhang, L. Zhu, L. Wu, and S. Yu, "Effect: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid," *Science China Information Sciences*, vol. 62, no. 3, pp. 1–14, 2019.
[23] A. Abdallah and X. Shen, "Lightweight security and privacy preserving scheme for smart grid customer-side networks," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1064–1074, 2017.
[24] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.
[25] C. Xu, R. Lu, H. Wang, L. Zhu, and C. Huang, "Pavs: a new privacy-preserving data aggregation scheme for vehicle sensing systems," *Sensors*, vol. 17, no. 3, p. 500, 2017.
[26] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, and J. Hu, "Appa: An anonymous and privacy preserving data aggregation scheme for fog-enhanced iot," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.
[27] J. Zhang, Y. Zhao, J. Wu, and B. Chen, "Lpda-ec: A lightweight privacy-preserving data aggregation scheme for edge computing," in *IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2018, Chengdu, China, Oct*, 2018, pp. 98–106.
[28] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual International Cryptology Conference, CRYPTO 2001, Santa Barbara, California, USA, Aug*, 2001, pp. 213–229.
[29] C. Gao, B. Wei, D. Xie, and C. Tang, "Divisible on-line/off-line signatures," in *The Cryptographers' Track at the RSA Conference, CT-RSA 2009, San Francisco, CA, USA, Apr*, 2009, pp. 148–163.
[30] E. Bresson, D. Catalano, and R. Gennaro, "Improved on-line/off-line threshold signatures," in *International Conference on Practice and Theory in Public-Key Cryptography, PKC 2007, Beijing, China, Apr*, 2007, pp. 217–232.
[31] Y. Zhang, Z. Chen, and F. Guo, "Online/offline verification of short signatures," in *International Conference on Information Security and Cryptology, Inscrypt 2010, Shanghai, China, Oct*, 2010, pp. 350–358.
[32] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of cryptology*, vol. 17, no. 4, pp. 294–319, 2004.
[33] E.-J. Goh, S. Jarecki, J. Katz, and N. Wang, "Efficient signature schemes with tight reductions to the diffie-hellman problems," *Journal of cryptology*, vol. 20, no. 4, pp. 493–514, 2007.
[34] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 1999, Prague, Czech Republic, May*, 1999, pp. 223–238.

JIALE ZHANG received the M.E. degree in computer technology from the Tianjin Polytechnic University, Tianjin, China, in 2017. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests are mainly edge computing, privacy-preserving, and machine learning.

YANCHAO ZHAO received his B.S. degree and Ph.D. degree in Computer Science from Nanjing University in 2007 and 2015. He is currently an associate professor in college of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. In 2011 he was a visiting student in the Department of Computer and Information Sciences, Temple University, Philadelphia, USA. His research interests include wireless network, mobile computing, edge computing and device-free sensing.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visotor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) overseas Outstanding Achievement Award.

BING CHEN received his B.S. and M.S. degree in computer engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1992 and 1995, respectively. He received Ph.D. degree in the College of Information Science and Technology from NUAA in 2008. He has worked for NUAA since 1998, he is currently a Professor at the computer science and technology department of NUAA. His main research interests include cloud computing, wireless communications and cognitive radio networks.