APPA: An Anonymous and Privacy Preserving Data Aggregation

Scheme for Fog-Enhanced IoT

Zhitao Guan\textsuperscript{1}, Yue Zhang\textsuperscript{1}, Longfei Wu\textsuperscript{2}, Jun Wu\textsuperscript{2}, Jing Li\textsuperscript{1}, Yinglong Ma\textsuperscript{2}, Jingjing Hu\textsuperscript{4*}

1. School of Control and Computer Engineering, North China Electric Power University, Beijing, China
2. Department of Mathematics and Computer Science, Fayetteville State University, Fayetteville, NC, USA
3. School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
4. School of Computer, Beijing Institute of Technology, Beijing, China

Abstract

Fog computing is a modern computing platform that connects the cloud with the edge smart devices located at the edge of the network. The fog computing platform has several characteristics desirable for Internet of Things (IoT) systems, such as the efficient data access, low latency, and location awareness. Data aggregation is a common operation in IoT systems. However, for data aggregation applications in the fog-enhanced IoT environment, how to efficiently preserve the privacy of sensitive data is a major concern. To address this challenge, we propose APPA: a device-oriented Anonymous Privacy-Preserving scheme with Authentication for data aggregation applications in fog-enhanced IoT systems, which also supports multi-authority to manage smart devices and fog nodes locally. In APPA scheme, the anonymity and authenticity of the device is guaranteed with pseudonym and pseudonym certificate, which can be updated autonomously. Taking the advantage of a local certification authority, the pseudonym management can be shifted to specialized fogs at the network edge, which provide real-time service for device registration and update. The data privacy can be ensured during data aggregation by using the Paillier algorithm. Detailed security analysis is conducted to show that our scheme can achieve security and privacy-preservation properties in the fog-enhanced IoT systems. Additionally, we compare our scheme with existing schemes to demonstrate the effectiveness and efficiency of our proposed scheme in terms of low computational complexity and communication overhead.

Keywords: IoT, Fog Computing, Data privacy, Anonymity.

1 Introduction

Internet of things (IoT) has emerged as one of the most significant technology in recent years, which has received a great attention and has been regarded as the next generation of the Internet [1]. Data sensed by smart devices (SD), like sensors, mobile phones, vehicles, are transmitted to the cloud for a various of purposes, such as automated management of industrial plants, smart transportation control and healthcare, etc. All of these can be realized by big data analysis and learning [2]. The rapid development of IoT technologies has boosted the increase of smart devices, which brings a substantial pressure and overhead on data processing and storage to the IoT system. Intuitively, offloading the computation and storage tasks to the cloud can break through the limits [3, 4, 5]. However, cloud computing may not be an ideal solution for the latency-sensitive IoT applications, considering the drawbacks and flaws in the current cloud computing paradigm including high latency, lake of reliability, and lake of support for security, mobility, and location awareness [6, 7]. How
to integrate the cloud and IoT is a big challenge to be solved [8]. Fog computing was proposed by CISCO researchers as a new alternative paradigm that can meet the requirements in IoT [9]; fog computing enables the partial shift of storage and computing tasks from the cloud to end users or near-user edge devices, particularly when it comes to data analysis and management [10, 11], to provide efficient data access and processing in IoT systems.

Fog computing can be viewed as a distributed computing paradigm that extends the cloud to the smart device side. As shown in Fig.1, there are three main layers in the fog-enhanced IoT architecture, the lower layer, the middle layer, and the upper layer. Sensing data are collected by smart devices (SD) at the lower layer and transmitted to the fog nodes (FN) (e.g. gateway) in the middle layer for some local processing. Then, the preliminary processed data are sent to the upper layer for visualization and analysis. Fog computing supports ephemeral storage at the lower layer and semi-permanent storage at the upper layer [12, 13]. It also provides real-time high-quality intelligence management to smart devices in IoT [14, 15].

![Fig. 1: The illustration of fog-enhanced IoT](image)

Although IoT integrated with fog computing can play a critical role in delivering various services to SDs in a highly efficient and effective way, potential security and privacy risks also need to be resolved [16, 17]. First of all, high-frequency data collection may cause great concerns over the location privacy. To prevent the SDs from being tracked by the adversary, it is significant to ensure that the real identities of SDs are never exposed [18, 19]. Additionally, the identities of FNs and SDs could be impersonated to upload malicious data or illegally collect data for malicious purposes. Traditional pseudonym schemes can protect the identities of SDs. However, since the frequency of data collection on different SDs may vary, periodically updating all the certificates by the central server cannot meet the privacy protection requirements of high-frequency data collection tasks. In addition, the authentication between devices in different layers is another major issue to solve [20, 21, 22]. Due to the limitations of storage and computing capability in SDs, complicated cryptographic operations cannot be adopted. Moreover, since there are huge amounts of SDs and FNs serving the various IoT applications, the dynamics of incoming and outgoing SDs increases the system management complexity. Furthermore, all devices are supposed to be managed by a single central authority in the traditional architecture, this premise may not satisfy the practical requirements of such a large-scale distributed IoT system [23]. Besides, the leak of SDs sensitive data like location and time schedule should also be taken
seriously. Overall, the resource-constrained SDs are highly vulnerable in the heterogeneous and distributed IoT system, devising an effective privacy-preserving scheme with low communication cost and computation overhead is significant for providing reliable real-time services to SDs [24].

Hence, we propose APPA: a device-oriented Anonymous Privacy-Preserving data aggregation scheme for the fog-enhanced IoT system, which also supports multi-authority for the local management of SDs and FNs. The main contributions of this paper are summarized as follows:

1) We adopt the pseudonym certificates to realize the anonymity of SDs, and set up local authorities for certificate register and update. SDs collaborate with the local certificate authority (LCA) and the trusted certificate authority (TCA) to generate the pseudonym certificates, which can prevent the certificate forgery.

2) We realize the autonomous pseudonym certificate update for SDs in an on-demand manner, which better satisfy the SDs requirement in certificate update.

3) We realize reliable data aggregation in the fog-enhanced IoT system, achieving data authentication, integrity verification, and privacy preservation.

4) We demonstrate the security and efficiency of the proposed scheme with security analysis and performance evaluation.

The remainder of this paper is organized as follows. Section 2 introduces the related work, section 3 reviews some preliminaries, section 4 describes the system model, security requirements and the corresponding design goals. In section 5, our scheme is described in detail. In Section 6, security analysis is given. In Section 7, the performance evaluation of our scheme is presented. Section 8 concludes the paper.

2 Related Work

Some works [25, 26, 27] have studied the IoT systems with a fog-enhanced architecture, such as the smart grid, the vehicular ad hoc networks, and the smart city. However, these works mainly focus on the reliability, efficiency, and scalability aspects, while the privacy and security challenges brought by fog computing are neglected.

Some previous works [28, 29, 30, 31] has summarized the major security and privacy challenges of IoT systems, which includes authentication, device trust and data privacy. Stojmenovic et al [32] considered the authentications at different layer of devices, however, the dynamic update of SD information has not been considered. Additionally, other lightweight authentication protocols have been proposed in [33, 34, 35], for the distributed communication environment.

As for the device trust in distributed environment, Kai Hwang et al. [36] developed a reputation-based trust management scheme for both user devices and data centers in the cloud systems. However, the device trust management is more complicated because of the more dynamics in IoT system.

In terms of privacy issues in the fog-enhanced IoT systems, both the identity privacy and data privacy should be taken into consideration. Existing works [37, 38, 39], focus on protecting the privacy of devices, but its not ideal in efficiency for real-time IoT applications and the SD and FN in fog computing cannot meet the requirements on computing power in these schemes. The pseudonym scheme has been considered an important technique for the protection of identity privacy, work [40] proposed a authentication protocol with anonymity for wireless body area networks. Specifically, the real identity of a person is replaced by a pseudonym, to prevent the person from being tracked.

In order to protect the data privacy for SDs with low communication overhead, efficient and privacy-preserving aggregation scheme should be developed. Lu et al. [41] proposed to use the gateway to collect data from the smart meters. Some other works [42, 43] tried to perform secure data aggregation using blockchain technology.
3. Preliminaries

In this section, we briefly introduce the Paillier cryptosystem and some notations used in our scheme.

3.1. Notations

In Table. 1, the notations used in this paper are listed.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>Timestamp of event</td>
</tr>
<tr>
<td>SD&lt;i&gt;</td>
<td>The i&lt;sup&gt;th&lt;/sup&gt; smart device</td>
</tr>
<tr>
<td>FN&lt;k&gt;</td>
<td>The k&lt;sup&gt;th&lt;/sup&gt; fog node</td>
</tr>
<tr>
<td>Fog&lt;j&gt;</td>
<td>The j&lt;sup&gt;th&lt;/sup&gt; fog area</td>
</tr>
<tr>
<td>LCA&lt;sub&gt;j&lt;/sub&gt;</td>
<td>Local certificate authority in Fog&lt;sub&gt;j&lt;/sub&gt;</td>
</tr>
<tr>
<td>TCA</td>
<td>Trusted certificate authority</td>
</tr>
<tr>
<td>Gen&lt;sub&gt;rsa&lt;/sub&gt;</td>
<td>The Initialization of RSA encryption</td>
</tr>
<tr>
<td>Gen&lt;sub&gt;pai&lt;/sub&gt;</td>
<td>The Initialization of Paillier encryption</td>
</tr>
<tr>
<td>pk&lt;sub&gt;pai&lt;/sub&gt;, sk&lt;sub&gt;pai&lt;/sub&gt;</td>
<td>Public key and private key of Paillier</td>
</tr>
<tr>
<td>Pseu&lt;sub&gt;sd&lt;/sub&gt;, Pseu&lt;sub&gt;fn&lt;/sub&gt;</td>
<td>SD’s and FN’s pseudonyms</td>
</tr>
<tr>
<td>Cerp&lt;sub&gt;sd&lt;/sub&gt;, Cerp&lt;sub&gt;fn&lt;/sub&gt;</td>
<td>SD’s and FN’s pseudonym certificates</td>
</tr>
</tbody>
</table>

3.2. Paillier cryptosystem

Paillier cryptosystem [44] is an asymmetric encryption algorithm, which can achieve additive homomorphism property efficiently. It has three phases: key generation, encryption and decryption.

1) Key generation: given the security parameter \( \kappa \), choose two prime numbers \( p, q \), where \( |p|=|q| \), and calculate \( \lambda = \text{lcm}(p-1,q-1) \). Then defined a function \( L(u) = \frac{(u-1)}{n} \), where \( n = pq \), and choose a generator of cyclic group \( g \), then calculate \( \mu = (L(g^2 \mod n^2))^{-1} \mod n \). The public key is \( (n, g) \), and the corresponding private key is \( (\lambda, \mu) \).

2) Encryption phase: given the plain text \( m \), select a random number \( r \in \mathbb{Z}_n^* \). Then, the ciphertext can be calculated as: \( C = \text{Enc}(m) = g^mr^n \mod n^3 \).

3) Decryption phase: Given the ciphertext \( C \), the plain text can be recovered with the private key \( (\lambda, \mu) \) as: \( m = \text{Dec}(C) = L(C^{\lambda \mod \lambda}) \cdot \mu \mod n \).

The Paillier cryptosystem has been proved to be provably secure against chosen plaintext attack [44].

3.3 Signature-of-knowledge

In the zero-knowledge proof protocol [45], the validity of statement can be verified while the verifier has learned nothing about the specific statement. Firstly, calculate a certain quantity \( \omega \), which can satisfy some kinds of relation \( R \) with respect to a commonly known string \( x \). Verifier can be convinced by prover that he holds \( \omega \) such that \( (\omega, x) \in R \). If a proof-of-knowledge protocol can be done in such a way that the verifier learns nothing other than the validity of the statement, this protocol is called a zero-knowledge Proof of Knowledge (PoK) protocol [45].
PoK protocol is a three-round protocol between two prover P and verifier V with a binary relation R. Input $x$ to P and $(\alpha, x)$ to V. In the first round, P sends a commitment $t$ to V. In the second round, V responds with a challenge $c$. P sends the corresponding reply $y$ to conclude the conversation in the last round. V outputs the validity: accept or reject. If the output is accepted by an honest verifier, the protocol transcript is valid, which is also proved as completeness property.

Any PoK protocol can be realized in a noninteractive form, which is named Signature of Knowledge (SoK) [46]. In SoK protocol, a challenge $c$ is set to the hash value of the commitment $t$ together with the message to be signed [47].

**Definition 1** (Computational Diffie-Hellman assumption): this assumption claims that given $g^a, g^b$, computing $g^{ab}$ is computationally hard.

**Definition 2** (Decisional Diffie-Hellman Problem): Given a tuple $(P, aP, bP, cP) \in G$, where $a, b, c$ are unknown, decide whether or not $c = ab \in Z_q$. If no $\tau$-time algorithm has advantage at least $\epsilon$ in solving the DDH problem in $G$, we say that $(\tau, \epsilon)$ holds in $G$.

**Definition 3** (RSA assumption): given $x, y \in Z_N$, and $a, b \in Z$ such that $x^a = y^b$, if $\gcd(a, b) = 1$, it is hard to find out an $x' \in Z_N$ such that $(x')^a = y$.

4. System Model and Design Goal

In our scheme, we consider a layered architecture in the fog-enhanced IoT system, as shown in Fig.2, which consists of five well-defined entities, smart device (SD), fog node (FN), public cloud server (PCS), trusted certification authority (TCA) and local certification authority (LCA). Data are collected from SDs on the edge of the system, then transported to the FNs in the middle layer through data transmission network. Fog nodes may perform some local processing over the raw data before sending them to the PCS. PCS will conduct data processing and analysis as required by the specific applications. TCA and LCA are independent agencies for the certification management of the whole system.

4.1. System model

The detailed description of the five entities is as follows.

1) SD: smart devices are deployed near the end user. Each SD is equipped with the sensing, storage, and communication capabilities. SDs have only limited computing capacity, hence complex computations are not executed on SDs.

2) FN: fog nodes are critical local computing task forces in fog computing, which are deployed at the edge of the network between SDs and PCS. FNs serve the vast SDs in local data processing, storage and communication. They can also be used to manage SDs in a particular area. In our scheme, the FNs will aggregate the collected data to reduce the bandwidth using between FNs and PCS.

3) PCS: the public cloud server is in the upper layer, which provides scalable, schedulable and manageable virtual servers with strong computing power and virtually unlimited storage space. The PCS processes and analyzes the received data to help IoT applications to better serve the users.

4) TCA: the trusted certification authority is an independent entity for key and certificate management.

5) LCA: the local certification authority is set up for improving the efficiency of certificate management. Specifically, each fog owns a local certification authority for the generation and update of pseudonym certificate for new SDs.
4.2. Adversarial model

In our scheme, we assume that the SDs deployed on the user side are vulnerable to an adversary, and the adversary can abuse a legal devices identity to steal or tamper the sensing data for malicious purposes. Besides, an adversary can track SDs to infer the true identity of a user. FNs and the PCS are honest but curious. That is to say, they do not manipulate the sensing data, but they are always attempting to sniff the sensitive data passing through.

In addition, the communication channel between SDs and PCS may not have been fully encrypted, an adversary could eavesdrop to get valuable information. Additionally, the adversary could launch active attacks to compromise the data integrity during transmission.

4.3. Design goal

Considering the adversarial model above, our design goal includes four aspects.

1) Device anonymity and unforgeability: the real identities of the SDs should be kept invisible in the system. The pseudonym certificates are used instead which cannot be fabricated or used fraudulently.

2) Certificate management: the local certificate management can greatly reduce the computational and communication cost for the generation and update of pseudonyms and pseudonym certificates.

3) Privacy-preservation: the SDs’ sensing data are inaccessible to other devices, even if they collude with each other.

4) Data source authentication and integrity verification: in the communication process, all messages are supposed to be coming from the claimed senders, and the messages have not been modified by an adversary during the transmission.

5. APPA Scheme

5.1. System initialization

When PCS sends data collection requests to SDs in $fog_i$, TCA and LCA generates corresponding parameters to initialize the system for the following operations. The initialization process is shown in Fig.3.
Step 1: Each LCA randomly chooses two large prime numbers \( p_0, q_0 \) by running \( \text{Gen}_{\text{RSA}} \) for SDs in \( \text{fog}_j \), and calculates the RSA modulus \( N_0 = p_0q_0 \). Then it chooses a prime number \( e \), and computes \( d \equiv e^{-1} \mod \varphi(N_0) \), the public key is \( (e, N_0) \) and corresponding private key is \( d \).

Step 2: The TCA randomly chooses two large prime numbers \( p_1, q_1 \), by running \( \text{Gen}_{\text{RSA}} \), and calculates the RSA modulus \( N_1 = p_1q_1 \). Then, it chooses a prime number \( \beta \), and computes \( \alpha \equiv \beta^{-1} \mod \varphi(N_1) \). The public key is \( (\beta, N_1) \), and the corresponding private key is \( \alpha \).

Step 3: Given the security parameters \( \kappa \), TCA chooses two large prime numbers \( p, q \), where \( |p|=|q|=\kappa \), and runs \( \text{Gen}_{\text{Paillier}}(\kappa) \) of Paillier Cryptosystem to generates the public key is \( pk_{\text{paillier}} = (N, g) \), \( N = pq \), and the corresponding private key \( sk_{\text{paillier}} = (\lambda, \mu) \).

Step 4: The TCA defines three one-way hash functions: \( H_1 : \{0,1\}^* \rightarrow Z_N^* \), \( H_2 : \{0,1\}^* \rightarrow Z_N^* \), \( H_3 : \{0,1\}^* \rightarrow Z_N^* \), which are random oracle mappings.

### 5.2. Registration

The registration process is divided into two parts: the new smart device registration and fog node registration. The SD's identity is denoted by \( ID_{\text{sd}} \) and FN's identity is denoted by \( ID_{\text{fn}} \). Below are the detailed steps.

#### 5.2.1 SD registration

The registration process of SD is shown in Fig. 4.
Step1: an unregistered smart device \( SD_i \) in \( fog_j \) has to request a pseudonym for privacy protection. Specifically, \( SD_i \) chooses a random number \( r_i \in \mathbb{Z} \), and calculates \( P = g^{r_i} \). Then it computes the signature \( \delta_{SD_i} = H_i(P)^{r_i} \mod N_0 \) and \( M_{SD} = H_i(P \parallel ID_{SD_i} \parallel r_i \mod N_0 \parallel TS) \). The certificate information \( Cr_{SD} \) is calculated based on the signature, where \( Cr_{SD} = \frac{r_i}{\delta_{SD_i} \cdot e} \mod N_0 \). Finally, \( SD_i \) sends the request packet to \( LCA_j \) in \( fog_j \).

\[
SD_i \rightarrow LCA_j : \{M_{SD} \parallel Cr_{SD} \parallel ID_{SD_i} \parallel P \parallel TS\} \quad (1)
\]

Step2: after receiving the packet from \( SD_i \), \( LCA_j \) verifies \( M_{SD} \) by checking \( \delta_{SD_i} \) with public key \( e \), and checks if \( M_{SD} = H_i(P \parallel ID_{SD_i} \parallel Cr_{SD} \cdot H_i(P)^{M_{SD}} \mod N_0 \parallel TS) \) holds, if it does holds, the identity of \( SD_i \) is validated, since

\[
Cr_{SD} \cdot H_i(P)^{M_{SD}} \mod N_0 = \frac{r_i}{H_i(P)^{\delta_{SD_i} \cdot e} \cdot H_i(P)^{M_{SD}} \mod N_0}

= r_i^* \mod N_0 \quad (2)
\]
LCA$_j$ chooses a random number $r_j \in Z_n^*$, and computes $Q = g^{r_j}$. Then it sends $Enc(P || Q || TS, Pub_{SD})$, which is an encrypted response packet to $SD_j$. (The Enc algorithm can be any public key encryption technique, depending on the specific need in the system).

LCA$_j$ calculates the pseudonym of $SD_j$: $Pseu_{SD} = p^r = g^{r_j}$, then stores and updates the $Pseu_{fog}$ of $SD_j$ in $fog_j$. Finally, it sends the pseudonym of $SD_j$ to TCA.

$$LCA_j \rightarrow SD_j : (Enc[P || Q || TS], Pub_{SD}) \quad (3)$$

Step3: upon receiving the response packet from LCA$_j$, $SD_j$ decrypted the packet with its private key. If the recovered $P$ is correct, then $SD_j$ calculates its pseudonym with $r \in Z_n^*$, and calculates $P' = g^{r'}$. $r'$ meets $r + r' = 1 \mod n$, then $SD_j$ sends a request packet to TCA to get pseudonym certificate for secure communication.

$$SD_j \rightarrow TCA : Enc(P' || Pseu_{SD} || TS, Pub_{TCA}) \quad (4)$$

Step4: after receiving the certificate request, the TCA decrypts the message and verifies the validity of $SD_j$’s pseudonym and timestamp TS. Then, TCA chooses a random number $r \in Z_n^*$, and calculates $I = g^{r_j}$. The pseudonym certificate is generated as $Cerp_{SD} = p^{r'} = g^{r_j}$. Finally, TCA stores and updates $Cerp_{fog}$ of $SD_j$ in $fog_j$, and sends the encrypted response packet to $SD_j$.

$$TCA \rightarrow SD_j : Enc(P', I || TS, Pub_{TCA}) \quad (5)$$

Step5: $SD_j$ decrypts the data packet with its private key. If the recovered $P'$ is correct, it can calculates its pseudonym certificate. The registration process of $SD_j$ is completed.

5.2.2 FN registration

The registration process of FN is shown in Fig. 5.
Fig.5 : FN registration process

Step1: suppose a new fog node $FN_i$ in $fog_j$ is willing to join the system, $FN_i$ first chooses a random number $r_i \in Z_n$, and calculates $L = g^{r_i}$. Then it computes signature $\delta_{FN_i} = H_i(L)^a \mod N_0$, and $M_{FN} = H_j(L \parallel ID_{FN_i} \parallel r_i^a \mod N_0 \parallel TS)$. The certificate information $Cr_{FN}$ can be calculated based on the signature $Cr_{FN} := \frac{r_i}{\delta_{FN_i}^\beta} \mod N_0$. Finally, $FN_i$ sends the request packet to $LCA_j$ in $fog_j$.

$$FN_i \rightarrow LCA_j : \{M_{FN} \parallel Cr_{FN} \parallel ID_{FN_i} \parallel L \parallel TS\} \quad (6)$$

Step2: after receiving the packet from $FN_i$, $LCA_j$ verifies $M_{FN}$ by checking $Cr_{FN}$ with its public key $\beta$, and checks if $M_{FN} = H_j(L \parallel ID_{FN_i} \parallel Cr_{FN} \cdot H_i(L)^a \mod N_0) \mod N_0$ holds, if it does hold, the identity of $FN_i$ is validated. It then calculates $r'_i \in Z_n$, which meets: $r'_i + r_i = 1 \mod n$ ($r'_i$ is a random number, which was chosen in SD's registration process), and calculates the pseudonym for new FN $Pseu_{FN_i} = L' = g^{r'_i a}$. After registering the new FN, $fog_j$ sends a pseudonym certificate request to TCA.

$$LCA_j \rightarrow TCA : Enc((L \parallel Pseu_{FN_i} \parallel TS), Pub_{TCA}) \quad (7)$$

Step3: the TCA decrypts the request packet and chooses a random number $r'_i \in Z_n$, which meets: $r'_i + r_i = 1 \mod n$ (random number $r'_i$ is a random number, which was chosen in SD's registration process). Then, it calculates the $FN_i$'s pseudonym certificate $Cerp_{FN} = Pseu_{FN_i} \cdot g^{r'_i a}$, stores and updates $Cerp_{List}$ of FNs in $fog_j$, the TCA sends the encrypted response packet to $FN_i$.

$$TCA \rightarrow FN_i : Enc((L \parallel Cerp_{FN_i} \parallel TS), Pub_{FN_i}) \quad (8)$$

5.3. Privacy-preserving data aggregation
5.3.1 Data collection
The smart devices in fog collect data \( (d_1, d_2, \ldots, d_n) \) at time point \( t \in T \), and perform the following steps to generate \( SD_i \)'s report:

**Step1:** \( SD_i \) picks a random number \( r_i \in Z_n^* \), and computes the encrypted data \( C_i \) based on \( SD_i \)'s pseudonym.

\[
C_i = (P_{\text{seu}_{SD_i}})^{r_i} \cdot r_i \mod n^2
\]

(9)

**Step2:** \( SD_i \) computes the message digest \( \sigma_i = H_i(C_i) \mod n \), then \( SD_i \) sends the data packet to the corresponding \( FN_k \) in fog.

\[
SD_i \rightarrow FN_k : [C_i, \| \sigma_i \| \text{Cerp}_{SD_i} \| TS]
\]

(10)

### 5.3.2 Data aggregation

**Step1:** after receiving the packet from SDs in fog, \( FN_k \) verifies if \( \sigma_i = H_i(C_i) \mod n \) holds, if it holds, \( FN_k \) calculates the aggregated data with SDs' pseudonym certificates, in order to prevent illegal SDs sending malicious data to the system:

\[
\sum_{i=1}^{k}(C_i, \text{Cerp}_{SD_i}) = \sum_{i=1}^{k}[(g^{r_i})^{d_i} \cdot g^{\gamma_i}] \mod n^2
\]

(11)

**Step2:** \( FN_k \) computes the final aggregation data with its certificate and \( r_i \), \( r_i \) meets \( r_i + r_i = 1 \mod n \) (\( r_i \) is a random number, which was chosen in FN's pseudonym generation process). Only if it holds the valid certificate of target fog node, can it gets the correct aggregated data:

\[
C_a = \sum_{i=1}^{k}(C_i, \text{Cerp}_{SD_i}) \cdot \text{Cerp}_{FN_k} \cdot g^{r_i}
\]

\[
= g^{(d_i + d_i \ldots + d_i) r_i} \cdot g^{\gamma_i} \cdot g^{r_i} \mod n^2
\]

(12)

**Step3:** \( FN_k \) calculates the message digest \( \sigma_{C_a} = H_i(C_a) \mod n \), and sends report packet to the PCS.

\[
FN_k \rightarrow PCS : [\sigma_{C_a} \| C_a \| TS \| \text{Cerp}_{FN_k}]
\]

(13)

### 5.3.3 Secure report reading

Upon receiving the packet from \( FN_k \), PCA verifies (1) the validity of the packet by checking the timestamp TS, (2) the integrity of the packet by verifying if \( \sigma_{C_a} = H_i(C_a) \mod n \) holds, (3) the validity of \( FN_k \)'s certificate.

Then PCS uses the private key \( sk_p : (\lambda, \mu) \) to recover the aggregated data in plaintext based on Paillier decryption[44].

### 5.4. Pseudonym and certificate updating

In our scheme, we consider the management of the pseudonyms and certificates in practical cases. Note that the update procedures of FN are the same with SD, hence we will only give a detailed description in the case of SD.

#### 5.4.1 Case 1: autonomous updating
For each $SD_i$ which frequently interacts with the system, periodical update of the pseudonyms and certificates cannot meet the requirements for security and anonymity. In our scheme, $SD_i$ is able to update its pseudonym and certificate as per its own demand.

When $SD_i$ wants to update its pseudonym and certificate, $SD_i$ informs its corresponding $LCA_j$ it associated with. Specifically, it chooses a new random number $r_{new} \in Z_n$, and calculates $P_{new} = g^{r_{new}}$. Then it sends an update request to $LCA_j$, $SD_i \rightarrow LCA_j : \{P_{\text{seu}_{SD_i}} \parallel P_{\text{new}} \parallel Q \parallel \text{Re}_{\text{update}} \parallel |TS|\}$, where $Q$ is calculated by $LCA_j$ in registration phase. In the meanwhile, $SD_i$ will reject any incoming data collection request.

Upon receiving an update request, $LCA_j$ will verify the consistency of $Q \oplus P_{\text{new}}$ with $Q$, if satisfied, $LCA_j$ will send the response packet to $SD_i$, $LCA_j \rightarrow SD_i : \{\text{Enc}(P_{\text{new}} \parallel |TS|, Pub_{\text{SD_i}})\}$, in order to ensure the normal data collection, other parameters involved in the registration remain the same.

After the step 3 to 5 in $SD_i$’s registration phase are finished, pseudonym and certificate will be updated.

### 5.4.1 Case 2: expired information updating

When $SD_i$ has not interact with the system for a long time or if the pseudonym and certificate have not been updated in time, $SD_i$ becomes illegal in the system. Under such circumstances, previously registered information is no longer be trusted by the system, and $SD_i$ needs to go through the registration process again. Specifically $SD_i$ sends an update request to $LCA_j$, $SD_i \rightarrow LCA_j : \{P_{\text{seu}_{SD_i}} \parallel P_{\text{seu}_{SD_i}} \oplus Q \parallel \text{Re}_{\text{update}} \parallel |TS|\}$ $LCA_j$ verifies the existing registration information of $SD_i$ with $Q$, if satisfied, $LCA_j$ will choose new parameters as described in step 2 of the registration phase and perform the following steps of the registration.

### 5.5. Revocation

In our scheme, we consider the revocation of the pseudonyms and certificates in practical cases. For $SD_i$, there are two cases of revocation, active revocation and passive revocation, which means the revocation can be triggered by $SD_i$ itself or by $TCA$. When an $SD_i$ leaves the system, the active revocation will be triggered. The passive revocation is triggered only when a malicious $SD_i$ has been detected. In our scheme, an $SD_i$ is considered to be malicious if the behaviors of the $SD_i$ have negative impact on data aggregation, e.g., the malicious $SD_i$ may deliberately upload false data to cause wrong data aggregation results. For $FN$, since it plays a key role in fog-enhanced IoT System, the revocation can only triggered by $TCA$. In the following, the revocation procedures of $SD_i$ are given. The revocation of $FN$ is similar to the case 2 of $SD_i$.

#### 5.5.1 Case 1: active revocation

When $SD_i$ wants to be disconnected from the system, it sends a revocation request to the corresponding $LCA_j$. In the meanwhile, $SD_i$ will reject data collection request. $SD_i \rightarrow LCA_j : \{P_{\text{seu}_{SD_i}} \parallel P_{\text{seu}_{SD_i}} \oplus Q \parallel \text{Re}_{\text{rej}} \parallel |TS|\}$ where $Q$ has been calculated by $LCA_j$ in the registration phase.

Upon receiving the request packet, $LCA_j$ will verify the consistency of $Q \oplus P_{\text{seu}_{SD_i}} \oplus Q$. If satisfied, $LCA_j$ sends a revocation request to $TCA$. While the $TCA$ would revoke the certificate of $SD_i$, the $SD_i$’s pseudonym would remain in the system for the re-registration of $SD_i$.

#### 5.5.2 Case 2: passive revocation

When malicious behaviors of a $SD_i$ are detected by another $SD_i$, $FN$ or certification authorities, $LCA_j$ will keep tracking of the suspicious $SD_i$’s identity. The $TCA$ will revoke the pseudonym and certificate of any malicious $SD_i$. Additionally, the malicious $SD_i$s will be added to the blacklist of the system, which means they would never be accepted into the system again.
6. Security Analysis

In this section, we analyze the security properties of our APPA scheme, particularly in the security and privacy-preservation perspectives. Our analysis will focus on how the proposed scheme realizes the anonymity of SD, and why the identities of legal SDs cannot be used fraudulently. Additionally, with data authentication and integrity check, we demonstrate that our scheme is resistant to various passive and active attacks.

6.1. Anonymity and unforgeability of SDs

In the APPA scheme, the anonymity and unforgeability of SDs are guaranteed due to the following four measures.

Firstly, when a new device sends a registration request to \( LCA_j \), \( \{ M_{SD} \| C_{SD} \| ID_{SD} \| P \| TS \} \), where \( M_{SD} = H_j(P \| ID_{SD} \| r^d \mod N_o \| TS) \) is a digest of RSA signature of zero-knowledge, \( SD_i \) will generate the signature \( \delta_{SD_i} = H_j(P)^d \mod N_o \) with its own private key \( d \). The signature is then used for the generation of certification information \( C_{SD} := r \cdot \delta_{SD_i} \mod N_o \). Upon receiving this request, \( LCA_j \) will verify \( SD_i \)'s real identity with its public key \( e \) without knowing the specific signature \( \delta_{SD_i} \) based on the zero-knowledge proof protocol[45]. Besides, the signature \( \delta_{SD_i} \) is secure according to the Definition 3 (RSA assumption) with large public exponents in the random oracle model (which is provably secure as in work [48]): even if an adversary \( A \) holds \( d \), it is hard to find an \( x \in Z^*_N \) such that \( (x)^d = \delta_{SD_i} \). Therefore, the identity of the device can be verified by \( LCA_j \) while the signature is not revealed and cannot be forged either.

Secondly, \( LCA_j \) calculates and sends \( LCA_j \rightarrow SD_i : (Enc(P \| Q \| TS), Pub_{SD}) \) to \( SD_i \). Note that packet has been encrypted with \( SD_i \)'s public key, hence only \( SD_i \) can get the correct packet and verify if \( P = g^r \) has been changed or not, and generate the its pseudonym \( P_{seu_{SD_i}} = p^r = g^{x^r} \). Besides, the exchange of parameters used to generate the pseudonym is under Definition 1(CDH assumption): given \( g^x, g^y \), it is computationally hard to compute \( g^{xy} \), which means that even an adversary gets \( P, Q \), it is still impossible to calculate the corresponding pseudonym, hence preventing \( SD_i \)'s pseudonym from being stolen and forged.

Thirdly, in the pseudonym certificate generation phase, \( SD_i \) sends the request packet \( Enc(P \| P_{seu_{SD_i}} \| TS), Pub_{TCA} \) to the TCA. The TCA can determine whether the user is legal by validating the pseudonym \( P_{seu_{SD_i}} \) without the knowledge of \( SD_i \)'s actual identity. The generation of the pseudonym
certificate is the same as pseudonyms, which both rely on Definition 1. Therefore, an adversary is not able to forge or steal an SD's pseudonym certificate.

Lastly, SD can change its pseudonym and pseudonym certificate whenever necessary, making it hard for a global passive adversary to continuously track an SD. This can protect the real identity of the SD from being exposed to the adversary. Additionally, the generation of pseudonyms and pseudonym certificates involves multiple entities, which can decrease the risk of leak when compared with traditional pseudonym schemes with single authority.

In sum, APPA scheme can guarantee the anonymity of SDs and prevent their real identities from being forged.

6.2. Privacy-preservation

In the APPA scheme, the privacy of sensing data can be guaranteed with three measures.

Firstly, in the data collection phase, SD’s data are encrypted as $C_i = (g^{r_i})^s \cdot r_i \mod n^2$, which are valid ciphertext of Paillier Cryptosystem. Since Paillier cryptosystem is provably secure against the chosen plaintext attack based on the Definition 2 (Decisional Diffie-Hellman Problem), the data $d_i$ is also semantic secure and privacy-preserving [30]. Therefore, even though the adversary $A$ eavesdrops $C_i$, it is impossible for him to know the plaintext.

Secondly, in the data aggregation process, FN cannot recover the plaintext of individual SD's data, it just computes aggregation data $C_s = \sum_{i=1}^{n} (C_i \cdot Cer_p_{sd} \cdot Cer_p_{fn} \cdot g^{r_i})$. Besides, FN’s valid certificate is involved in the calculation of aggregation, an adversary cannot obtain the correct aggregated data even if FN is compromised by the adversary. Besides, data are store in the formed of $C_s = g^{\sum_{i=1}^{n} d_i} \mod n^2$, which means, even if an adversary intrudes in FN’s database, it cannot get the individual data $d_i$.

Thirdly, when the PCS receives $C_s = g^{\sum_{i=1}^{n} d_i} \mod n^2$ from FN, the PCS recovers $C_s$ as $\sum_{i=1}^{n} d_i$ using its private key. Since the PCS only has the aggregated data from SDs in fog, individual SD's sensing data would not be exposed even if the adversary $A$ hacks into the PCS.

In sum, APPA scheme can ensure the privacy of SD's data in both processing and transmission phases.

6.3. Data authentication and integrity check

In APPA scheme, the authenticity and integrity of data can be guaranteed.

The data authentication is performed using the pseudonym certificates of SDs and FNs (the reliability of pseudonym and pseudonym certificate have been proved in the subsection above). During the data processing phase, an SD's sensing data is encrypted with its pseudonym, and then uploaded to the FN. Before the FN aggregates the encrypted data, the SD's pseudonym certificate must be used to process the encrypted data. If the pseudonym certificate does not match with the pseudonym, the data will not be correctly aggregated, which ensures the authenticity of the data source and prevents illegal SDs who maliciously submit invalid data.
Besides, the certificates of FNs are also required in data aggregation. It is impossible for an adversary to submit data to the PCS without a valid FN certificate.

As for data integrity, all entities will verify the integrity of data with digest during transmissions. The hash functions $H_2 : \{0,1\}^* \rightarrow Z_N^*$, $H_3 : \{0,1\}^* \rightarrow Z_N^*$ are random oracle mappings. If data has been modified during the transmission, the data receiver can easily detect by verifying the message digest.

In sum, our scheme can ensure that the received messages are indeed from the claimed senders and have not been manipulated during the transmission.

7. Performance Evaluation

In this section, we evaluate the performance of the proposed APPA scheme, in terms of the computation complexity as well as the communication overhead.

7.1. Computation complexity

In APPA scheme, the computation complexity is mainly composed of two parts: new device registration and data processing. The data processing contains three phases: data encryption, aggregation and decryption.

Firstly, we look into the computation complexity in device registration. Since the registration protocol is based on RSA signature and Diffie-Hellman, the most complex operations involved are the exponentiation, hash function, and multiplication operation.

The registration of a new smart device requires six exponentiation operations in $Z_N^*$ and one hash function. In data processing, when the PCS sends a data request to SD in $fog_i$, $SD_i$ will generate report $\{ \parallel \parallel \parallel \}^{i}^{i}^{i}^{i}^{i}^{i}^{i}^{i}^{i}^{i}$ by the Paillier encryption, which needs two exponentiation operations in $Z_N^*$, one hash function and one multiplication operation. After $FN_k$ receives $n$ response packets from $fog_j$, $FN_k$ verifies the validity of the received data which performs $n$ hash functions, then it aggregates the received ciphertexts with its own certificates which executes $n+2$ multiplication operations, two hash functions and one exponentiation operation in $Z_N^*$. Upon receiving the report from $FN_k$, the PCS authenticates the sender's identity and verifies the integrity of the report which needs one hash operation, then it decrypts the ciphertext with Paillier decryption. For simplicity, we denote the computational cost of an exponentiation operation, multiplication operation and hash function, as $e$, $m$ and $h$ respectively. And the computation complexities of the major entities in APPA scheme are as shown in Table 2.

In order to evaluate the efficiency of our scheme, we compare it with the recent ASAS scheme [49] in terms of the computation overhead. The ASAS scheme also focuses on realizing data aggregation with authentication in fog-based IoT systems. We conduct experiments with JBPC libraries [50] on a PC with 3.2GHz-processor and 8GB-memory. Compared with the exponentiation operation and multiplication operation, the hash operation can be regarded as negligible [50]. As shown in Fig.6, our scheme has an advantage in computational cost compared with the ASAS scheme. Since the fog-based IoT system may contain thousands of SDs, even a slight improvement in efficiency may bring huge benefits. Therefore, APPA scheme is very promising for the fog-based IoT system, especially for the resource-limited end SDs.
Table 2: Computation Complexity

<table>
<thead>
<tr>
<th>Entity Name</th>
<th>Involving Operations</th>
<th>Computation Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>1) SD's data collection</td>
<td>( C_n + C_m + 2C_e )</td>
</tr>
<tr>
<td></td>
<td>2) Data encryption</td>
<td>( (n+2)* (C_n + C_m) )</td>
</tr>
<tr>
<td></td>
<td>3) Message digest ( \sigma ) generation</td>
<td>( C_n + C_m + C_e )</td>
</tr>
<tr>
<td>FN</td>
<td>1) SD's data integrity verification</td>
<td>( C_n + C_m + C_e )</td>
</tr>
<tr>
<td></td>
<td>2) SD's validity and legality verification and data aggregation</td>
<td>( C_n + C_m + C_e )</td>
</tr>
<tr>
<td></td>
<td>3) Aggregated data processing</td>
<td>( C_n + C_m + C_e )</td>
</tr>
<tr>
<td>PCS</td>
<td>1) Aggregated data integrity verification and sender authentication</td>
<td>( C_n + C_m + C_e )</td>
</tr>
<tr>
<td></td>
<td>2) Aggregated data decryption</td>
<td>( C_n + C_m + C_e )</td>
</tr>
</tbody>
</table>

Fig. 6: Comparison of computational cost

7.2. Communication Overhead

The communication overhead in our APPA scheme can be divided into two parts: one part is in the registration and authentication, the other is in data collection and processing.

In the registration and authentication process, SD sends  \( \{ M_{SD} \| C_{rsd} \| ID_{sd} \| P \| TS \} \) to the associated LCA to request for pseudonym, which then sends  \( Enc(P', Pseu_{sd}, TS, Pub_{TCA}) \) to the TCA to request for the certificate. The main size of the report for registration and authentication is \( S_r = |Pseu_{sd}| + |ID_{sd}| + |P| + |TS| \).

We will compare the communication overhead in registration and authentication process with the recent ASC [51] and ASAS [49] schemes, which also focus on authenticating SDs in an high efficient way. Their communication overheads are listed as follows:

\[ ASC: S_{rk} = |Pseu| + |C| + |\delta| + |TS| \quad (14) \]

\[ ASAS: S_{rk} = |Pseu| + |ID_{rsd}| + |\delta| + |TS| \quad (15) \]
For the ease of comparison, we choose the same parameters for ASC and ASAS schemes, namely 160-bit $Z_N$ and 160-bit $G$. Fig. 7 presents the communication overhead in the registration and authentication phases of ASC, ASAS and our scheme, versus the number of SDs.

![Fig. 7: Comparison of communication overhead in registration and authentication process](image)

Although the device registration and generation of pseudonyms require multiple entities to participate in our APPA scheme, it does not bring too much overhead in communication, and it remains efficient with the increase of the number of SDs.

In the data collection and processing process, the communication overhead of APPA scheme contains two major parts: one is the communication from $SD_i$ to $FN_k$ in fog, and the other is the communication from $FN_k$ to the PCS, which are abbreviated as $SD\rightarrow FN$, $FN\rightarrow PCS$, respectively. In the first part, the SD generates the response report as $\{C_i\|\sigma_i\|\text{Cerp}_{SD_i}\|TS\}$ and sends it to $FN_k$. The size of the report is $S_{SD_i} = |C_i| + |\sigma_i| + |\text{Cerp}_{SD_i}| + |TS|$, for a fog node $FN_k$ managing $n$ smart devices, the maximum communication overhead in this phase is $S_{SD_{max}} = n^* (|C_i| + |\sigma_i| + |\text{Cerp}_{SD_i}| + |TS|)$. In the second part, $FN_k$ sends the aggregated report to the PCS. The report is calculated as $\{\sigma_{C_k}\|C_k\|\text{TS}\|\text{Cerp}_{FN_k}\}$, so the size of the report is $S_{FN_k} = |\sigma_{C_k}| + |C_k| + |\text{TS}| + |\text{Cerp}_{FN_k}|$.

The ASAS scheme also implements data aggregation in the fog-enhanced system. We compare the communication overhead in the data collection and processing process in our scheme with ASAS. The communication overheads of ASAS is listed as follows:

$$SD(TD)\rightarrow FN : S_{SD_{max}} = n^* (|C_i| + |\sigma| + |Pseu| + |TS|) \quad (16)$$

$$FN\rightarrow PCS : S_{FN_k} = n^* (|\sigma| + |C| + |Pseu| + |ID| + |TS|) \quad (17)$$
For the ease of comparison, we suppose that each user generates a 2,048-bit ciphertext and chooses 160-bit $Z_N$ and 160-bit $G$. Fig. 8 presents the communication overhead in the data collection and processing process of ASAS and our scheme versus the number of devices.

![Fig. 8: Comparison of communication overhead](image)

It is obvious that the total communication cost of our APPA scheme is lower than the ASAS scheme.

8. Conclusion

In this paper, we propose APPA scheme: a device-oriented Anonymous Privacy-Preserving scheme with Authentication in fog-enhanced IoT system. We realize the anonymity and multi-layer authentication of device by pseudonym and pseudonym certificate. The proposed scheme also supports autonomous update of pseudonym and certificate by SDs. Additionally, the privacy of sensing data can be guaranteed in a highly efficient way. Compared with the existing schemes of this kind, APPA provides flexible, efficient, and autonomous management of devices. We provide security analysis to demonstrate the security and privacy-preservation properties of our scheme. The performance evaluations show that APPA is a better choice for fog-enhanced IoT system with resource-limited devices and real-time communications. For future work, we will apply the proposed scheme to some specific scenarios, i.e., data aggregation in smart grid. Furthermore, we will also work on other challenging security and privacy issues in the fog-enhanced IoT system.

Acknowledgement

This work is partially supported by Beijing Natural Science Foundation under grant 4182060, the National Key R&D Program of China under grant 2018YFC0830605 and 2018YFC0831504, the Fundamental Research Funds for the Central Universities under grant 2018ZD06.
References


[34] Z. Qikun, G. Yong, Z. Quanxin, W. Rifang, T. Yu-An, A dynamic and cross-domain authentication asymmetric group key agreement in telemedicine application, IEEE Access.


[38] T. Mao, C. Cao, X. Peng, W. Han, A privacy preserving data aggregation scheme to investigate apps installment in massive mobile devices, Procedia Computer Science 129 (2018) 331–340.


Zhitao Guan is currently an Associate Professor at the School of Control and Computer Engineering, North China Electric Power University (NCEPU). He received his BEng degree and PhD in Computer Application from Beijing Institute of Technology (BIT), China, in 2002 and 2008, respectively. His current research focuses on smart grid security, wireless security and cloud security. Dr. Guan has authored over 40 peer-reviewed journal and conference papers in these areas. He is a Member of the IEEE.

Yue Zhang is currently a Master candidate at the School of Control and Computer Engineering, North China Electric Power University (NCEPU). She received her BEng degree from North China Electric Power University in 2016. Her current research focuses on cloud security and smart grid security.

Longfei Wu is currently an assistant professor in the Department of Mathematics and Computer Science at Fayetteville State University. He received his Ph.D. degree in computer and information sciences from Temple University in July 2017. He obtained his B.E. degree in telecommunication engineering from Beijing University of Posts and Telecommunications in July 2012. His research interests are the security and privacy of networked systems and modern computing devices, including mobile devices, Internet-of-Things, implantable medical devices, and wireless networks.

Jun Wu is an Associate Professor of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He obtained his PH.D. Degree in Information and Telecommunication Studies (GITS) at Waseda University, Japan. He was a postdoctoral researcher for the Research Institute for Secure Systems (RISEC), National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He worked as a researcher for the Global Information and Telecommunication Institute (GITI), Waseda University, Japan, from 2011 to 2013. His research interests include the advanced computation and
communications techniques of smart sensors, wireless communication systems, industrial control systems, wireless sensor networks, smart grids, and more.

**Jing Li** received her master’s degree in computer science from North China Electric Power University in April 2018. Following graduation, Jing plans to continue her studies at the University of Houston, where she will pursue her Ph.D in Electrical and Computer Engineering. Her current research focuses on cloud security and smart grid security.

**Yinglong Ma** is a professor with the School of Control and Computer Engineering, North China Electric Power University, Beijing, China. He received his Ph.D degree from the Institute of Software, Chinese Academy of Sciences, Peking, China, in 2006. His main research areas include Knowledge Engineering, Big Data Analysis and Information Forensics. He has published more than 60 scientific papers in some international/domestic journals and conferences in the last decades.

**Jingjing Hu** received the PhD degree in computer science from Beijing Institute of Technology, Beijing, China. She is currently an associate professor in the School of Computer of Beijing Institute of Technology. Her research interests are in the areas of service computing, web intelligence, and information security.