Anonymous and Secure Aggregation Scheme in Fog-Based Public Cloud Computing

Huaqun Wang

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China. E-mail: wanghuaqun@aliyun.com

Zhiwei Wang

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China. E-mail: zhwwang@njupt.edu.cn

Josep Domingo-Ferrer

Universitat Rovira i Virgili, Department of Computer Engineering and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia. E-mail: josep.domingo@urv.cat

Abstract

By using fog computing, cloud computing can be extended to the edge of the network. Generally, in the public cloud, fog computing comprises three components: terminal device, fog node and public cloud server (PCS). In this paper, we propose the concept of anonymous and secure aggregation scheme (ASAS) in fog-based public cloud computing. In the ASAS model, a fog node aggregates the data from terminal nodes and forwards the aggregated data to the public cloud server. By using the ASAS scheme, the fog node can help terminal devices upload their data to PCS. By using the data aggregation technique, our ASAS scheme can save bandwidth between the fog node and PCS. At the same time, our ASAS scheme not only protects the identities of terminal devices by using pseudonyms but it also guarantees data secrecy

via a homomorphic encryption technique. In this paper, we design the first concrete ASAS scheme. We also examine the security and the performance of our proposal, which we show to be provably secure and efficient.
Keywords: Fog computing, Cloud computing, Terminal device, Anonymity, Homomorphic encryption

1. Introduction

Cloud computing develops very quickly along with the rapid rise of big data. The cloud makes big data processing possible by providing storage and computing power. In turn, big data also push cloud computing forward, as more and more users would like to make use of the cloud to store and process their data. In general, three different types of clouds can be distinguished: public clouds, private clouds, and hybrid clouds. A public cloud is an external or publicly available cloud environment that can be accessed by any users on a pay-per-use model. Public cloud services are attractive because they are inexpensive, they are easy to set up, they scale well, and they make an efficient use of resources. Although public cloud computing develops very quickly, it poses some problems that have to be solved; these include lack of reliability, high latency, lack of support for mobility and location awareness, security [1, 2, 3]. Fog computing was proposed in 2012 by CISCO researchers Bonomi *et al.* [4] as an alternative paradigm to solve the above issues: the idea is to partly shift storage and computing from cloud data centers to a multitude of end-user or near-user edge terminal devices coordinates by the so-called fog nodes.

In their seminal paper [4], Bonomi *et al.* demonstrated that fog comput-

ing can be used in the following three scenarios: connected vehicle, smart grid, and wireless sensor and actuator networks [4]. In 2014, Bonomi *et al.* [5] proposed a hierarchical distributed architecture extending from terminal devices situated on the edge of the network to the network core: in particular, the Internet of Things (IoT) and the resulting big data analytics were presented as a use case for such an architecture. Yi *et al.* [6] continued work on fog computing by introducing representative application scenarios and considering design issues of fog computing systems. Wang *et al.* [7] presented and discussed new security and forensics problems that arise with fog computing. Other security and privacy issues of fog computing have also been surveyed in [8, 9].

In order to save bandwidth between fog nodes (FNs) and the public cloud server (PCS), and to save computation at PCS, it is important to aggregate the data from terminal devices (TDs) and upload the aggregated data to PCS: aggregation is especially suited when the tasks delegated by PCS to TDs involve obtaining partial results that need to be added to get the total result (like adding frequencies of pattern occurrence in parts of a data set each processed by a different TD to get the frequency in the total data set). Since, for safety reasons, data stored at TDs must be encrypted, aggregation has to be performed on ciphertext. Homomorphic encryption can be utilized to allow privacy-preserving aggregation at the local gateways without decryption; it is a form of encryption where the algebraic operation performed on the plaintext is equivalent to another algebraic operation performed on the ciphertext. The homomorphic cryptosystems by Domingo-Ferrer [10, 11] allow two operations on encrypted data, but they are only

secure against ciphertext-only attacks. That is to say, the attacker has no channel providing access to the plaintext prior to encryption. In the practical attack environment, the attacker can get some knowledge of the plaintext. Thus, Domingo-Ferrer [10, 11]'s schemes are not secure in the stronger security model. In 1999, Paillier designed an efficient homomorphic encryption system [12]. After that, Catalano *et al.* proposed an efficient probabilistic cryptosystem which works in the ring $\mathbb{Z}/n^2\mathbb{Z}$, where n is a RSA integer [13]. In 2003, Galindo *et al.* proposed a cryptosystem in the group of points of an elliptic curve [14]. The elliptic curve is defined in the ring $\mathbb{Z}/n^2\mathbb{Z}$ where n is an RSA integer and its security comes from the difficulty of factoring n. In 2007, by using the LUC function [15], Castagnos adapted Catalano *et al.*'s scheme in a group simpler than the elliptic curve over $\mathbb{Z}/n^2\mathbb{Z}$ [16]. In 2012, Lu *et al.* designed an efficient and privacy-preserving aggregation scheme for smart grid communications [17]; they use a super-increasing sequence to structure multi-dimensional data and encrypt the structured data. Viejo *et al.* [18] presented in 2012 a method for secure and scalable many-to-one lossy transmission via homomorphic ciphertext aggregation that enables the root node of a communication tree to compute any mathematical function (like minimum, maximum, average) on the data sent by the leaf nodes. In CT-RSA 2015, Castagnos and Laguillaumie solved an open problem: to design a linearly homomorphic scheme based on the sole hardness of the discrete logarithm problem [19]. The security of this scheme relies on the hardness of the decisional Diffie-Hellman problem. The scheme can be efficiently implemented within some specific group, namely the class group of imaginary quadratic fields. We will review the Castagnos-Laguillaumie scheme [19] in Section 2.2.

The cryptographic technique has already been used to design cryptosystems, including, for instance, Bresson et al.'s encryption scheme [20]. Homomorphic encryption and aggregation techniques can be used in many different fields, such as health fog [21], smart grid [17], *etc.*

In the modern information society, more and more terminal devices wish to protect their identity privacy [22, 23, 24, 25]. In cloud computing and fog computing, privacy and efficiency are the important problems which have been studied by many experts [26, 27, 28, 29]. Unless privacy is guaranteed, the TD's owner will not allow the TD to help any PCS, even if it has available idle resources. For example, when the TD is a smart meter, its identity can be mapped to its owner's identity; hence privacy for the smart meter's identity must be provided. Pseudonyms are a technique for identity privacy that has been used in many fields, such as vehicular networks [30], mobile social networks [31], IPTV (Internet Protocol Television) [32], among others [33].

Contribution and plan of this paper

In this paper, we consider a fog computing scenario in which a public cloud PCS leverages the available idle computing power of a multitude of terminal devices (which can be connected devices in the Internet of Things). The PCS delegates computing tasks to the TDs such that PCS can obtain the total result by aggregating the partial results obtained by the various TDs. As previously mentioned, one example can be counting the frequency of occurrence of a certain pattern in a very large data set that has been split into subsets each of which is processed by a different TD.

The above task delegation requires security for PCS (no one other than PCS should be able to recover the results obtained) and privacy for the

TDs (if helping PCS with their idle computing resources implies losing their anonymity, they will be reluctant to help).

Thus, we tackle the need for secure aggregation and identity privacy in fog computing. First, we introduce the concept of anonymous and secure aggregation scheme (ASAS) in fog computing. Second, we give a formal system model and security model for ASAS. Third, we instantiate a concrete ASAS scheme based on bilinear pairings, short signatures and the Castagnos-Laguillaumie cryptosystem. Fourth, we give a performance analysis and a security analysis, which show that our concrete ASAS is efficient and provably secure.

The remainder of this paper is organized as follows. Section 2 gives the cryptographic preliminaries. Section 3 presents the formal system model and the security model of ASAS. Section 4 describes a concrete ASAS scheme. Section 5 analyzes the security and the performance of our concrete ASAS. Finally, Section 6 is a conclusion.

2. Preliminaries

Our ASAS scheme is predicated on cryptographic techniques that include signature, encryption and aggregation in public-key cryptography. Specifically, we use elliptic curve public-key cryptography and the Castagnos-Laguillaumie cryptosystem. In this section, we review these building blocks as well as some underlying cryptographic notions.

2.1. Bilinear pairings and computational assumptions

A bilinear pairing is a bilinear map $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$, where $\mathcal{G}_1$ is a cyclic additive group and $\mathcal{G}_2$ is a cyclic multiplicative group, both with prime order

q. If P is a generator of $\mathcal{G}_1$, a bilinear pairing satisfies the following properties:

1. Bilinearity: $\forall a, b \in \mathbb{Z}/q\mathbb{Z}^*$, $e(aP, bP) = e(P, P)^{ab}$.
2. Non-degeneracy: $e(P, P) \neq 1_{\mathcal{G}_2}$.
3. Computability: $\forall Q, R \in \mathcal{G}_1$, there is an efficient algorithm to calculate $e(Q, R)$.

On a supersingular elliptic curve, we can construct a bilinear pairing e by using the modified Weil pairing [34] or the Tate pairing [35]. When using bilinear pairings, it turns out that the Computational Diffie-Hellman (CDH) problem is computationally hard while the Decisional Diffie-Hellman (DDH) problem is easy [36], where both problems are defined as follows:

Definition 1 (CDH)**.** *Given* $(P, aP, bP) \in \mathcal{G}_1^3$ *where* $a, b \in \mathbb{Z}/q\mathbb{Z}$ *are unknown, compute* abP*.*

Definition 2 (DDH)**.** *Given* $(P, aP, bP, cP) \in \mathcal{G}_1^4$ *where* a, b, c *are unknown, determine whether* $c = ab \bmod q$*.*

To make a distinction between the discrete logarithm (resp. decisional Diffie-Hellman) problem in a specific group and the general discrete logarithm (resp. decisional Diffie-Hellman) problem in elliptic curve groups, we abbreviate the former problems as $\widehat{DL}$ (resp. $\widehat{DDH}$). Next, we define a group G in which $\widehat{DDH}$ is assumed to be difficult, but which has a subgroup F where $\widehat{DL}$ is easy. G and F can be efficiently implemented within the class group of imaginary quadratic fields, as described in the following definition 3 [19].

Definition 3. *Consider the pair of algorithms* $(Gen, Solve)$*. The Gen algorithm is a group generator that takes as input two parameters* λ, μ *and outputs a tuple* (B, n, p, s, g, f, G, F)*. The output satisfies that* s *is a* λ*-bit integer,* p *is a* μ*-bit integer,* $\gcd(p, s) = 1$*,* $n = p \cdot s$ *and* B *is an upper bound for* s*.* $(G, \cdot)$ *is a cyclic group of order* n *generated by* g*, and* $F \subset G$ *is the subgroup of* G *of order* p *generated by* f*.* B *satisfies that the distribution of* $\{g^r, r \leftarrow \{0, 1, \cdots, Bp - 1\}\}$ *is statistically indistinguishable from the uniform distribution on* G*. The canonical surjection* $\pi : G \to G/F$ *is efficiently computable and, given* $h \in G/F$*, it is also efficient to compute* $h_l \in \pi^{-1}(h)$*. The Solve algorithm is a deterministic polynomial-time algorithm such that:*

1. *The* $\widehat{DL}$ *problem in* F *can be solved using the Solve algorithm and is therefore easy:*

$$\Pr[x = x^* : (B, n, p, s, g, f, G, F) \leftarrow Gen(1^\lambda, 1^\mu), x \leftarrow \mathbb{Z}/p\mathbb{Z},$$
$$X = f^x, x^* \leftarrow Solve(B, p, g, f, G, F, X)] = 1$$

2. *The* $\widehat{DDH}$ *problem is hard in* G *even with access to the Solve algorithm:*

$$|\Pr[b = b^* : (B, n, p, s, g, f, G, F) \leftarrow Gen(1^\lambda, 1^\mu), x, y, z \leftarrow \mathbb{Z}/n\mathbb{Z},$$
$$X = g^x, Y = g^y, b \leftarrow \{0, 1\}, Z_0 = g^z, Z_1 = g^{xy},$$
$$b^* \leftarrow \mathcal{A}(B, p, g, f, G, F, X, Y, Z_b, Solve(\cdot))] - \frac{1}{2}|$$

   is negligible for all probabilistic polynomial-time adversary $\mathcal{A}$*.*

See [19] for details on how to construct G and F.

2.2. The Castagnos-Laguillaumie cryptosystem

By using the groups G and F as per Definition 3, Castagnos and Laguillaumie proposed in [19] an efficient cryptosystem that can be described by

the following three algorithms:

1. KeyGen: On input the security parameters $\lambda,\ \mu$, output (B, n, p, s, g, f, G, F). Then pick a random $\bar{x} \in \{0, 1, \cdots, Bp-1\}$ and compute $\bar{X} = g^{\bar{x}}$. Let the public key be $pk = (B, p, g, \bar{X}, f)$ and the private key be $sk = \bar{x}$.
2. Encryption: Given a plaintext m, pick a random $r \in \{0, 1, \cdots, Bp-1\}$ and compute $c_1 = g^r,\ c_2 = f^m h^r$. The ciphertext is $(c_1,\ c_2)$.
3. Decryption: For the ciphertext (c_1, c_2), compute $M = c_2/c_1^{\bar{x}}$ and get the plaintext $m = Solve(p, g, f, G, F, M)$.

This cryptosystem is linearly homomorphic, that is:

- If (c_1, c_2) is the ciphertext corresponding to m and (c_1', c_2') is the ciphertext corresponding to m', then decrypting (c_1c_1', c_2c_2') returns $m + m' \bmod p$.
- If (c_1, c_2) is the ciphertext corresponding to m, then decrypting $(c_1^{\alpha}, c_2^{\alpha})$ for any scalar $\alpha \in \{0, \ldots, p-1\}$ returns $\alpha m \bmod p$.

3. System model and security model of ASAS

3.1. System model and definition of ASAS

First, we give the formal system model. Our anonymous and secure aggregation scheme consists of four different entities:

1. System manager: This entity generates the system parameters and helps the other entities generate the private key/public key pairs.

2. TD: Terminal devices are user or near-user entities on the edge of the network (such as connected devices in the Internet of Things). They may have idle computing and communication capacities that can be leveraged by a public cloud server PCS.
3. FN: A fog node bridges TD and PCS. It uses a collaborative multitude of TDs for storage (rather than primarily storing data in cloud data centers), for communication (rather than routing data over the internet backbone), and for control, configuration, measurement and management (rather than primarily relying on network gateways).
4. PCS: A public cloud server has a large storage space and a strong computing capacity to process data coming from TDs or FNs. In our model, PCS is trusted by TDs and FNs.

In a realistic application, the user and a PKI (public key infrastructure) are also indispensable. By using the PKI, every entity gets the certification that builds the relation between the entity's identity and its public key. At least two fog computing business models are conceivable:

- The owners of TDs (where a single owner may own many TDs) make the idle computing power of their TDs available to PCS. In return, PCS may compensate them in some agreed way (maybe some rent).

- The users of the PCS cloud own TDs whose idle resources they want to use to supplement PCS's computing resources. In this way, they need to rent less resources from PCS and/or get a better service.

In either model, the fog nodes FN are provided by PCS. When all the encrypted final results have been uploaded to PCS, the corresponding user will

run some methods to process them. For example, it can download them and decrypt them locally. Of course, it can also authorize PCS to decrypt the ciphertext partially (or wholly). In order to simplify our construction, we omit the methods for data processing and PKI authentication.

Definition 4 (ASAS)**.** *An ASAS scheme consists of six phases conducted among TD, FN, PCS and the system manager:*

1. *Set-up: On input the security parameters, the set-up algorithm generates the system parameters. It also generates the private key/public key pairs for TD, FN and PCS, respectively. Finally, it publishes some public system parameters and sends the private keys to TD, FN and PCS.*
2. *Registration: In order to register, TD (resp. FN) begins an interactive protocol with PCS. After this interactive protocol, TD (resp. FN) gets an authorization from PCS. This authorization allows TD (resp. FN) to upload its data.*
3. *TD data processing and upload: After processing its data, TD encrypts them and uploads the ciphertexts to FN. Before the upload takes place, TD and FN must authenticate each other.*
4. *Secure data aggregation and upload: After receiving many ciphertexts from TD (or several TDs), FN aggregates these ciphertexts into one ciphertext. Then, FN uploads the aggregated ciphertext to PCS.*
5. *Decryption: PCS performs the decryption process and obtains the plaintext.*
6. *TD and FN revocation: TD or FN can be revoked actively or passively.*

3.2. Security model

An ASAS scheme must be efficient and secure. The efficiency analysis focuses on the computation overhead and communication overhead. The security analysis aims at proving security. In order to formally model security, we identify the following security requirements:

1. Anonymity: TD remains unconditionally anonymous to FN.
2. Indistinguishability of ciphertexts: It is infeasible to distinguish the ciphertexts of two plaintexts (of the same length). That is, such ciphertexts are computationally indistinguishable.
3. Revocation: TD and FN can be revoked easily and efficiently.

According to the above requirements, we give the corresponding formal definitions:

Definition 5 (Anonymity)**.** *Let the possible set of TDs be $\mathcal{S}$. Without loss of generality, let the cardinality of the set $\mathcal{S}$ be n, i.e., $|\mathcal{S}| = n$. Anonymity of TD is satisfied if no adversary can find the real TD identity with probability greater than $\frac{1}{n}$.*

Next, we formally define indistinguishability of encryptions resistant against a chosen-plaintext attack (Ind-CPA).

Definition 6 (Ind-CPA)**.** *An encryption scheme $(params, E, D)$, where $params$ is the public parameters and (E, D) is the encryption/decryption algorithm pair, satisfies Ind-CPA if the probabilistic polynomial-time adversary $\mathcal{A}$ has a non-negligible advantage in the following game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. The game is given below:*

1. *On input the security parameter k, the challenger $\mathcal{C}$ runs Set-up procedure. $\mathcal{C}$ creates the system parameters params. At the same time, PCS's private key/public key pair, FN's private key/public key pair and TD's private key/public key pair are also created by $\mathcal{C}$.*
2. *$\mathcal{A}$ submits two different messages (m_0, m_1) to $\mathcal{C}$ where $|m_0| = |m_1|$. $\mathcal{C}$ picks α uniformly at random from $\{0, 1\}$. Then, $\mathcal{C}$ encrypts m_α and sends the corresponding ciphertext c_α to $\mathcal{A}$.*
3. *Taking use of the public parameters and c_α, $\mathcal{A}$ outputs the guess β.*

We say the proposed scheme satisfies Ind-CPA if for every positive polynomial p, all sufficiently large n, the following inequality holds:

$$\Big| \Pr[\beta = \alpha | \mathcal{A}(params, m_0, m_1, c_\alpha) = \beta] - \frac{1}{2} \Big| < \frac{1}{p(n)}$$

In Table 1, we summarize the notations used in the rest of this paper and their descriptions.

4. Design of the concrete ASAS scheme

To instantiate our concrecte ASAS scheme, we make use of the Castagnos-Laguillaumie cryptosystem, Boneh et al.'s short signature [36] and data aggregation. We next detail the six phases of our ASAS scheme, listed in Definition 4.

4.1. Set-up

Pick a generator P of $\mathcal{G}_1$. Let $e : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a bilinear pairing where $\mathcal{G}_1$ is a cyclic additive group and $\mathcal{G}_2$ be a cyclic multiplicative group

Table 1: Notations and descriptions

Notations	Descriptions
TD	Terminal device
FN	Fog node
PCS	Public cloud server
CPA	Chosen plaintext attack
p, s	p is μ-bit integer and s is λ-bit integer, $gcd(p, s) = 1$
B, n	B is an upper bound for s and $n = p \cdot s$
G, F	$(G, \cdot)$ is cyclic group of order n and $F \subset G$ is the subgroup of G
g, f	g is a generator of G and f is a generator of F
$\mathcal{G}_1, \mathcal{G}_2$	Two groups with the same prime order q
e	Bilinear pairing $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$
P	A generator of $\mathcal{G}_1$
$(x, \bar{x})$	PCS's private key
$(X, \bar{X})$	PCS's public key $X = xP, \bar{X} = g^{\bar{x}}$
(y, Y)	FN's private key/public key pair, $Y = yP$
(z, Z)	TD's private key/public key pair, $Z = zP$
H	Cryptographic hash function $H : \{0, 1\}^* \to \mathcal{G}_1$
ID_{td}, PS_{td}	TD's identifier and pseudonym
I_{td}	TD's information
Au_{td}	Authorization information for TD from PCS
ID_{fn}	FN's identifier
I_{fn}	FN's information
Au_{fn}	Authorization information for FN from PCS
T_i, T	Timestamp
c_i, c	Ciphertext

with the same prime order q. On input two parameters λ and μ, output (B, p, s, g, f, G, F). Pick a random $\bar{x} \in \{0, 1, \cdots, Bp-1\}$ and calculate $\bar{X} = g^{\bar{x}}$. PCS's private key is $(x, \bar{x})$ and its public key is $(X, \bar{X})$ where $1 \leq x \leq q$ and $X = xP$. Let FN's private key/public key pair be (y, Y) where $1 \leq y \leq q$ and $Y = yP$. Let TD's private key/public key pair be (z, Z) where $1 \leq z \leq q$ and $Z = zP$. The cryptographic hash function H is defined as $H : \{0,1\}^* \rightarrow \mathcal{G}_1$. In the end, the system public parameters are

$$params = \{\mathcal{G}_1, \mathcal{G}_2, e, P, B, p, g, f, \bar{X}, X, Y, Z, H\}.$$

4.2. Registration

Denote TD's identity by ID_{td} and FN's identity by ID_{fn}. TD picks a random string PS_{td} as its pseudonym and sends $(ID_{td}, I_{td}, PS_{td}, Sig_{td-pcs})$ to PCS, where I_{td} denotes ID_{td}'s corresponding information and $Sig_{td-pcs} = zH(ID_{td}, I_{td}, PS_{td})$. PCS verifies $e(Sig_{td-pcs}, P) \stackrel{?}{=} e(H(ID_{td}, I_{td}, PS_{td}), Z)$. If it does not hold, PCS rejects the query; otherwise, PCS and ID_{td} perform the following protocol:

1. PCS accepts TD's query and generates the authorization information Au_{td}. Then PCS calculates $Sig_{pcs-td} = xH(PS_{td}, Au_{td})$ and sends $(PS_{td}, Au_{td}, Sig_{pcs-td})$ to ID_{td}.
2. ID_{td} verifies $e(Sig_{pcs-td}, P) \stackrel{?}{=} e(H(PS_{td}, Au_{td}), X)$. If it holds, the registration process ends; otherwise, TD informs PCS and begins the registration process again.

FN sends $(ID_{fn}, I_{fn}, Sig_{fn-pcs})$ to PCS, where I_{fn} denotes FN's corresponding information and $Sig_{fn-pcs} = yH(ID_{fn}, I_{fn})$. PCS verifies $e(Sig_{fn-pcs}, P) \stackrel{?}{=} e(H(ID_{fn}, I_{fn}), Y)$. If it does not hold, PCS rejects the query; otherwise,

PCS and FN engage in a two-step protocol similar to the one above between PCS and TD to finish FN's registration. Finally, FN gets PCS's authorization (Au_{fn}, Sig_{pcs-fn}) where $Sig_{pcs-fn} = xH(ID_{fn}, Au_{fn})$ and Au_{fn} denotes the authorization information from PCS.

4.3. TD data processing and upload

In order to process the data in near-real time and upload them to FN, let us assume without loss of generality that in one time period of length η, each of n_1 TDs uploads one data item. Hence, the n_1 TDs will upload n_1 data items $(m_1, m_2, \cdots, m_{n_1})$ to FN one by one. These processed data satisfy the condition that $\sum_{i=1}^{n_1} m_i < p$.

Note. In general, in what follows each data item m_i is uploaded by a different TD_i. However, to simplify the notation, we will drop the subscript and we will use TD, z, Z, I_{td}, Au_{td}, ID_{td} and PS_{td} (to be strict, one would need to add the subscript i to the previous notations, which is rather cumbersome).

For each $i \in \{1, 2, \cdots, n_1\}$, TD encrypts m_i and signs the corresponding ciphertext as follows:

1. Pick a random number $r_i \in \{0, 1, 2, \cdots, Bp - 1\}$ and compute

$$c_{i1} = g^{r_i},\ c_{i2} = f^{m_i} \bar{X}^{r_i}.$$

2. Based on the ciphertext $c_i = (c_{i1}, c_{i2})$, FN, PS_{td} and the current time stamp T_i, TD creates the signature σ_i below:

$$\sigma_i = zH(c_i, ID_{fn}, PS_{td}, T_i).$$

Before beginning the data upload, TD interacts with PCS as follows:

1. TD sends the data uploading query $(PS_{td}, Au_{td}, Sig_{pcs-td})$ to FN.
2. FN verifies $e(Sig_{pcs-td}, P) \stackrel{?}{=} e(H(PS_{td}, Au_{td}), X)$. If it does not hold, FN refuses TD's query; otherwise, FN sends $(ID_{fn}, Au_{fn}, Sig_{pcs-fn})$ to TD.
3. TD verifies $e(Sig_{pcs-fn}, P) \stackrel{?}{=} e(H(ID_{fn}, Au_{fn}), X)$. If it does not hold, TD refuses to upload the data to FN; otherwise, the interaction succeeds.

When the above authentication interaction succeeds, TD uploads the data $(c_i, PS_{td}, T_i, \sigma_i), 1 \leq i \leq n_1$ to FN one by one. That is, TD uploads $(c_1, PS_{td}, T_1, \sigma_1)$ to FN; after a designated time interval η/n_1, TD uploads $(c_2, PS_{td}, T_2, \sigma_2)$ to FN; and so on.

4.4. Secure data aggregation and upload

In one period, we assume that FN receives n_1 messages $(c_i, PS_{td}, T_i, \sigma_i)$ for $i = 1, 2, \cdots, n_1$. FN does the following to aggregate the received messages:

1. FN verifies

$$e(\sum_{i=1}^{n_1} \sigma_i, P) \stackrel{?}{=} e(\sum_{i=1}^{n_1} H(c_i, ID_{fn}, PS_{td}, T_i), Z).$$

   If it does not hold, FN looks for any invalid messages by using dichotomic search and informs TD to retransmit them until they are valid; otherwise, FN goes on.
2. Compute the aggregated ciphertext $c = (\bar{c}_1, \bar{c}_2)$:

$$\bar{c}_1 = \prod_{i=1}^{n_1} c_{i1}, \quad \bar{c}_2 = \prod_{i=1}^{n_1} c_{i2}.$$

3. Create the signature

$$\sigma_{fn} = yH(c, ID_{fn}, FN, T),$$

where T is the time stamp.

4. Upload the message $(c, PS_{td}, ID_{fn}, T, \sigma_{fn})$ to PCS.

4.5. Decryption

Upon receiving the message $(c, PS_{td}, ID_{fn}, T, \sigma_{fn})$ from FN, PCS verifies

$$e(\sigma_{fn}, P) \stackrel{?}{=} e(H(c, ID_{fn}, FN, T), Y).$$

If it does not hold, PCS rejects FN's query; otherwise, PCS does:

1. Compute $M = \bar{c}_2/\bar{c}_1^{\bar{x}}$;
2. Compute the plaintext $m = Solve(p, g, f, G, F, M)$.

4.6. TD and FN revocation

We consider the following cases for TD and FN revocation:

1. When PCS wants to revoke TD, PCS informs FN; FN will reject TD's new requests; thus, TD becomes invalid. When PCS wants to revoke FN, PCS will reject FN's new requests; thus, FN becomes invalid.
2. When TD wants to be revoked, it will not access FN. When FN wants to be revoked, it will reject TD's new requests and will not access PCS. Further, if TD (resp. FN) does not access FN (resp. PCS) until the expiration date, TD (resp. FN) can also be revoked.

5. Security and performance

Security and efficiency are two important properties of the proposed ASAS scheme. We give a security analysis of our concrete ASAS in Section 5.1 and a performance analysis in Section 5.2.

5.1. Security analysis

To analyze security, we will focus on correctness, ciphertext indistinguishability, authentication security and TD's anonymity.

Theorem 1 (Signature correctness)**.** *The generated signatures* Sig_{td-pcs}, Sig_{pcs-td}, Sig_{fn-pcs} *and* Sig_{pcs-fn} *can pass the verification. In other words, if PCS, FN and TD honestly perform the signature and verification procedures, these signatures are accepted.*

Proof: According to the generation process of Sig_{td-pcs}, Sig_{pcs-td}, Sig_{fn-pcs} and Sig_{pcs-fn}, we have:

$$\begin{aligned} e(Sig_{td-pcs}, P) =& e(zH(ID_{td}, I_{td}, PS_{td}), P) \\ =& e(H(ID_{td}, I_{td}, PS_{td}), zP) \\ =& e(H(ID_{td}, I_{td}, PS_{td}), Z); \end{aligned}$$

$$\begin{aligned} e(Sig_{pcs-td}, P) =& e(xH(PS_{td}, Au_{td}), P) \\ =& e(H(PS_{td}, Au_{td}), xP) \\ =& e(H(PS_{td}, Au_{td}), X). \end{aligned}$$

By using the same method, the correctness of Sig_{fn-pcs} and Sig_{pcs-fn} can also be proved. □

Theorem 2 (Decryption correctness)**.** *If TD, FN and PCS are honest, the ciphertext c can be decrypted by PCS. In other words, the decrypted plaintext* m *is* $m = \sum_{i=1}^{n_1} m_i$.

Proof: According to the encryption process, we know that

$$c_{i1} = g^{r_i},\ c_{i2} = f^{m_i}\bar{X}^{r_i}.$$

By aggregating these data, we get

$$\bar{c}_1 = \prod_{i=1}^{n_1} c_{i1} = g^{\sum_{i=1}^{n_1} r_i},\ \bar{c}_2 = \prod_{i=1}^{n_1} c_{i2} = f^{\sum_{i=1}^{n_1} m_i}\bar{X}^{\sum_{i=1}^{n_1} r_i}.$$

Let $r = \sum_{i=1}^{n_1} r_i,\ m = \sum_{i=1}^{n_1} m_i$. Then, $\bar{c}_1 = g^r,\ \bar{c}_2 = f^m\bar{X}^r$. We can get $\bar{c}_2/\bar{c}_1^{\bar{x}} = f^m$. Since the discrete logarithm problem $\widehat{DL}$ is easy in the subgroup F, the plaintext m can be computed as $m = Solve(p, g, f, G, F, f^m)$. □

Theorem 3 (Ind-CPA)**.** *Based on the assumption that the* $\widehat{DDH}$ *problem is hard, the aggregated ciphertext c satisfies Ind-CPA.*

Proof: For encryption of a single message, satisfaction of Ind-CPA has been proved in [19]. We will prove that the aggregated ciphertext satisfies Ind-CPA. We will prove the theorem by contradiction. Let $\mathcal{C}$ be a challenger and $\mathcal{A}$ be an adversary in our ASAS scheme. We assume that the theorem does not hold, so that the adversary $\mathcal{A}$'s advantage is a non-negligible ϵ. On input a $\widehat{DDH}$ instance $(B, p, g, f, G, F, X, Y, Z)$, $\mathcal{C}$ picks a random $X' \in \mathcal{G}_1$ and sets PCS's public key $pk = (X', X)$. Thus, for the input (g, X, Y, Z) where $X = g^x, Y = g^y$ and x, y are kept unknown, $\mathcal{C}$ will decide whether or not $Z = g^{xy}$ holds. In the proof, we use the proof by contradiction. If $\mathcal{A}$ can break the proposed scheme, $\widehat{DDH}$ problem can be broken by $\mathcal{C}$.

In the challenge phase, $\mathcal{A}$ submits two different message groups $m_0 = (m_{01}, m_{02}, \cdots, m_{0n_1})$ and $m_1 = (m_{11}, m_{12}, \cdots, m_{1n_1})$ to $\mathcal{C}$. $\mathcal{C}$ picks $\alpha \in \{0, 1\}$ and computes $m_\alpha = \sum_{i=1}^{n_1} m_{\alpha i}$. Then, $\mathcal{C}$ computes $c = (Y, f^{m_\alpha} Z)$. Finally, $\mathcal{C}$ sends c to $\mathcal{A}$. If Z is not a random element, c will be indistinguishable from a true ciphertext of m_α. Otherwise, the ciphertext is independent of the message m_α. $\mathcal{A}$ outputs its guess α' to $\mathcal{C}$. If $\alpha' = \alpha$, $\mathcal{A}$ succeeds; otherwise, $\mathcal{A}$ fails. Thus, we get the success probability of $\mathcal{C}$ to solve the $\widehat{DDH}$ problem below:

$$\begin{aligned}\Pr[\mathcal{C}\ succeeds] = &\frac{1}{2}\Pr[\mathcal{A}\ succeeds | (X, Y, Z)\ satisfies\ (g^x, g^y, g^{xy})] \\ &+ \frac{1}{2}\Pr[\mathcal{A}\ succeeds | Z\ is\ a\ random\ element\ of\ G] \\ =&\frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2} * \frac{1}{2} = \frac{1}{2} + \frac{1}{2}\epsilon.\end{aligned}$$

In the process of solving the $\widehat{DDH}$ problem, the advantage of challenger $\mathcal{C}$ is

$$Adv_{\mathcal{C}} = \Pr[\mathcal{C}\ succeeds] - \frac{1}{2} = \frac{1}{2}\epsilon.$$

By assumption, ϵ is non-negligible. Hence, $Adv_{\mathcal{C}} = \frac{1}{2}\epsilon$ contradicts the hardness assumption on $\widehat{DDH}$. Thus, the aggregated ciphertext satisfies Ind-CPA. □

Authentication security. In order to realize the secure authentication and authorization, we make use of Boneh *et al.*'s short signature [36]. Since Boneh *et al.*'s short signature is provably secure, so is our authentication and authorization.

Anonymity. The pseudonym PS_{td} is picked randomly by TD. In TD's data processing and upload, TD makes use of the pseudonym PS_{td}. Since PS_{td} is a random string, TD remains unconditionally anonymous with respect

to FN.

5.2. Performance analysis

We analyze the performance of our scheme in terms of computation and communication.

5.2.1. Computation cost

In our concrete ASAS, three entities *TD, FN, PCS* interact to finish the data processing and upload. In the *Set-up* and *Registration* phases, the system parameters and the registration interaction are computed once and for all. Thus, we do not consider the computation overhead in these two phases.

In the *TD data processing and upload* phase, for every message TD performs two exponentiations and one multiplication in group G, along with one exponentiation in group F. At the same time, TD calculates one hash function and performs one scalar mulplication on group $\mathcal{G}_1$. Before the data uploading, TD and FN authenticate each other. In the authentication, TD computes one hash function and two bilinear pairings. On its side, FN computes one hash function and two bilinear pairings.

In the *Secure data aggregation and upload* phase, for the received n_1 messages $(c_i, PS_{td}, T_i, \sigma_i)$ $(i = 1, 2, \cdots, n_1)$, FN computes two bilinear pairings, $2(n_1 - 1)$ point additions and one scalar multiplication in group $\mathcal{G}_1$. Also, FN computes $2(n_1 - 1)$ multiplications in group G.

In the *Decryption* phase, PCS computes one hash function and two bilinear pairings. Also, PCS computes one exponentiation, one inverse and one multiplication in group G. Further, it computes one $Solve(\cdot)$ in subgroup F.

In order to evaluate how long the above operations take in a usual computing environment, we implemented a prototype. We used the C programming language with the GMP (GMP-5.0.5), Miracl and PBC (pbc-0.5.13) libraries. We used the following equipment to run PCS and FN:

- CPU: Intel Core i7-3517U @ 1.90GHz
- Physical Memory: 4GB DDR3 1600MHz
- OS: Ubuntu 13.04 Linux 3.8.0-19-generic SMP i686

On the other side, we ran TD on a SAMSUNG N9100 smarthphone with the following features:

- CPU: Qualcomm 2.7GHz
- Physical Memory: 3GB RAM
- OS: Android 5.0

Figure 1 depicts the computation overhead (in milliseconds) for the concatenation of the *TD data process and upload*, *Secure data aggregation and upload* and *Decryption* phases. We separately report the computing time invested by TD, FN and PCS, where TD actually accounts for the set of terminal devices sending the n_1 messages (in general each message m_i is sent by a different TD_i). The number n_1 of messages encrypted by TD is represented in the X-axis. In our implementation, we assume that n_1 ciphertexts are aggregated into one ciphertext by FN. The Y-axis denotes the computing time. From this real implementation, it can be seen that our concrete ASAS scheme takes very affordable execution times.

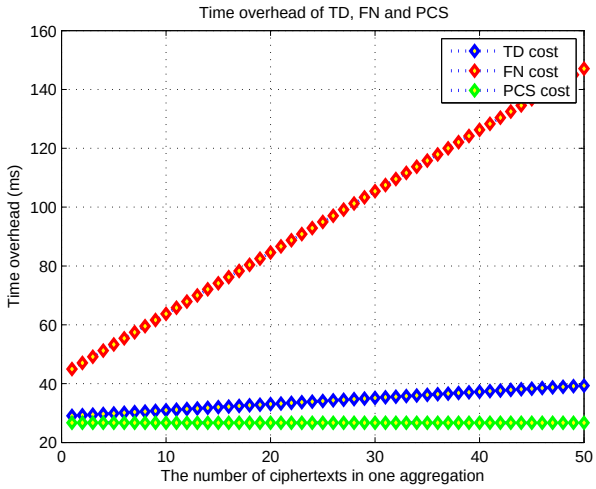

Figure 1: Time overhead (ms) of TD, FN and PCS

5.3. Communication analysis

We make use of a supersingular elliptic curve with a 160-bit order. The base field of this elliptic curve has 512-bit elements.

In the *TD data processing and upload* phase, in order to realize the authentication, TD sends $1024+|PS_{td}|+|Au_{td}|$ bits to FN and FN sends $1024+|ID_{fn}|+|Au_{fn}|$ bits to TD. For one ciphertext, TD will send $2048+|PS_{td}|+|T_i|+|c_i|$ bits to FN.

In the *Secure data aggregation and upload* phase, we assume that n_1 ciphertexts are aggregated. After the aggregation, FN sends $2048+|ID_{fn}|+|PS_{td}|+|T|+|c|$ bits to PCS. Hence, the consumed bandwidth after aggregation is roughly $\frac{1}{n_1}$ of the bandwidth that would be needed with unaggregated data. More precisely, the saving is:

$$\frac{2048+|ID_{fn}|+|PS_{td}|+|T|+|c|}{n_1(2048+|PS_{td}|+|T_i|+|c_i|)}=\frac{1}{n_1}+\frac{|ID_{fn}|}{n_1(2048+|PS_{td}|+|T_i|+|c_i|)}\approx\frac{1}{n_1}$$

6. Conclusion

This paper has presented the novel concept of anonymous and secure aggregation in fog-based public cloud computing. We have formalized the system model and the security model of ASAS. Then, we have presented a concrete instantiation of ASAS by using bilinear pairings and the Castagnos-Laguillaumie cryptosystem. Finally, we have proved the security of this instantiation and we have evaluated its performance. As a result, we can assert that our concrete ASAS system is provably secure and efficient enough to be deployed in practice.

Acknowledgments

The work of H. Wang was supported the National Natural Science Foundation of China (No. 61272522). The work of Z. Wang was supported the National Natural Science Foundation of China (No. 61373006, 61672016). The work of J. Domingo-Ferrer was supported by the European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"), from the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2014 SGR 537), and from the Spanish Government (project TIN2014-57364-C2-1-R "SmartGlacis" and TIN 2015-70054-REDC). The third author is with the UNESCO Chair in Data Privacy, but the views in this paper are the authors' own and are not necessarily shared by UNESCO.

References

[1] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, K. Ren. A privacy-preserving and copy-deterrence content-based image retrieval scheme

in cloud computing. IEEE Transactions on Information Forensics and Security. 11(11): 2594-2608, 2016.

[2] Z. Fu, K. Ren, J. Shu, X. Sun, F. Huang. Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Transactions on Parallel and Distributed Systems. 27(9), 2546C2559, 2016.

[3] Z. Xia, X. Wang, X. Sun, Q. Wang. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems. 27(2), 340-352, 2015.

[4] F. Bonomi, R. Milito, J. Zhu, S. Addepalli. Fog computing and its role in the internet of things. In: Proceedings of MCC'12, Helsinki, Finland, Aug. 13-17, 2012, pp. 13-16.

[5] F. Bonomi, R. Milito, P. Natarajan, J. Zhu. Fog computing: A platform for internet of things and analytics. In: Big Data and Internet of Things: A Roadmap for Smart Environments, Springer International Publishing, 2014, pp. 169-186.

[6] S. Yi, C. Li, Q. Li. A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou, China, Jun. 22-25, 2015, pp. 37-42.

[7] Y. Wang, T. Uehara, R. Sasaki. Fog computing: issues and challenges in security and forensics. In: Proceedings of COMPSAC'15, Taichung, July 1-5, 2015, pp. 53-59.

[8] S. Yi, Z. Qin, Q. Li. Security and privacy issues of fog computing: a survey. In: Proceedings of WASA 2015, LNCS 9204, Qufu, China, Aug. 10-12, 2015, pp. 685-695.

[9] I. Stojmenovic, S. Wen, X. Huang, H. Luan. An overview of fog computing and its security issues. Concurrency and Computation: Practice and Experience, DOI: 10.1002/cpe.3485.

[10] J. Domingo-Ferrer. A new privacy homomorphism and applications. Information Processing Letters, 60(5):277-282, 1996.

[11] J. Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In: Proceedings of ISC 2002, LNCS 2433, São Paulo, Brazil, Sep. 30-Oct. 2, 2002, pp. 471-483.

[12] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of EUROCRYPT'99, Prague, Czech Republic, May 2-6, 1999, pp. 223-238.

[13] D. Catalano, R. Gennaro, N. Howgrave-Graham, P. Q. Nguyen. Paillier's cryptosystem revisited. In: Proceedings of CCS'01, Swissôtel Chicago, Chicago, USA, Oct. 17-21, 2011, pp. 206-214.

[14] D. Galindo, S. Martín, P. Morillo, J. L. Villar. An efficient semantically secure elliptic curve cryptosystem based on KMOV. In: Proceedings of WCC'03, Versailles, France, Mar. 24-28, 2003, pp. 213-221.

[15] P. Smith, M. J.J. Lennon. LUC: A new public key system. In: Proceedings of IFIP/Sec'93, Toronto, Canada, May 12-14, 1993, pp. 103-117.

[16] G. Castagnos. An efficient probabilistic public-key cryptosystem over quadratic fields quotients. Finite Fields and Their Applications, 13(3)563-576, 2007.

[17] R. Lu, X. Liang, X. Li, X. Lin, X. Shen. Eppa: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Transactions on Parallel and Distributed Systems, 23(9)1621-1631, 2012.

[18] A. Viejo, Q. Wu, J. Domingo-Ferrer. Asymmetric homomorphisms for secure aggregation in heterogeneous scenarios. Information Fusion, 13(4)285-295, 2012.

[19] G. Castagnos, F. Laguillaumie. Linearly homomorphic encryption from DDH. In: Proceedings of CT-RSA'15, San Francisco, CA, USA, April 20-24, 2015, pp. 487-505.

[20] E. Bresson, D. Catalano, D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: ASIACRYPT 2003. LNCS, vol. 2894, pp. 37-54. Springer, Heidelberg (2003)

[21] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, T. Cheong, S. Lee. Health fog: a novel framework for health and wellness applications. The Journal of Supercomputing, 2016. DOI:10.1007/s11227-016-1634-x.

[22] C. Orazio, K. Choo. Circumventing iOS security mechanisms for APT forensic investigations: A security taxonomy for cloud apps. Future Generation Computer Systems, 2016. DOI: 10.1016/j.future.2016.11.010

[23] C. Orazio, K. Choo. A technique to circumvent SSL/TLS validations on iOS devices. Future Generation Computer Systems, 2016. DOI: 10.1016/j.future.2016.08.019

[24] C. Orazio, K. Choo, L. Yang. Data exfiltration from Internet of Things devices: iOS devices as case studies. IEEE Internet of Things Journal, 2016. DOI: 10.1109/JIOT.2016.2569094

[25] N. Cahyani, N. Ab Rahman, W. Glisson, K. Choo. The role of mobile forensics in terrorism investigations involving the use of cloud storage service and communication apps. Mobile Networks and Applications, 2016. DOI: 10.1007/s11036-016-0791-8

[26] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu. Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Transactions on Communications. E98-B(1), 190-200, 2015.

[27] T. Ma, J. Zhou, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, S. Lee. Social network and tag sources based augmenting collaborative recommender system. IEICE transactions on Information and Systems. E98-D(4), 902-910, 2015.

[28] Q. Liu, W. Cai, J. Shen, Z. Fu, X. Liu, N. Linge. A speculative approach to spatialtemporal efficiency with multiobjective optimization in a heterogeneous cloud environment. Security and Communication Networks. 9(17), 4002-4012, 2016.

[29] Y. Kong, M. Zhang, D. Ye. A belief propagation-based method for task allocation in open and dynamic cloud environments. Knowledge-based Systems. 115, 123-132, 2016.

[30] R. Lu, X. Li, T. H. Luan, X. Liang, X. Shen. Pseudonym changing at social spots: an effective strategy for location privacy in VANETs. IEEE Transactions on Vehicular Technology, 61(1), 86-96, 2012.

[31] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, X. Shen. Fully anonymous profile matching in mobile social networks. IEEE Journal on Selected Areas in Communications, 31(9)641-655, 2013.

[32] L.J. Khayati, C. Orencik, E. Savas, B. Ustaoglu. A practical privacy-preserving targeted advertising scheme for IPTV users. International Journal of Information Security, 2015. DOI: 10.1007/s10207-015-0296-7.

[33] Z. Fu, X. Sun, S. Ji, G. Xie. Towards efficient content-aware search over encrypted outsourced data in cloud. Proceedings of the 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM), San Francisco, CA, 2016, DOI: 10.1109/INFOCOM.2016.7524606

[34] D. Boneh, M. Franklin. Identity-based encryption from the Weil pairing. In: Proceedings of CRYPTO'01, Santa Barbara, California, USA, Aug. 19-23, 2001, pp. 213-229.

[35] A. Miyaji, M. Nakabayashi, S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. IEICE Transactions Fundamentals, 5:1234-1243, 2001.

[36] D. Boneh, B. Lynn, H. Shacham. Short signatures from the Weil pairing. In: Proceedings of ASIACRYPT'01, Gold Coast, Australia, Dec. 9-13, 2001, pp. 514-532.