Coop-DAAB: Cooperative Attribute Based Data Aggregation for Internet of Things Applications

Belguith, Sana, Kaaniche, Nesrine, Mohamed, Mohamed and Russello, Giovanni

http://dx.doi.org/10.1007/978-3-030-02610-3_28

Coop-DAAB: Cooperative Attribute Based Data Aggregation for Internet of Things Applications

Sana Belguith[1], Nesrine Kaaniche[2], Mohamed Mohamed[3], and Giovanni Russello[4]

[1] School of Computing, Science and Engineering, University of Salford, Manchester, UK
[2] SAMOVAR, Telecom SudParis,University Paris-Saclay, France
[3] IBM Research, Almaden Research Center, San Jose, CA, USA
[4] The Cyber Security Foundry,
The University of Auckland, New Zealand
belguith.sana@gmail.com,nesrine.kaaniche@telecom-sudparis.eu,
mmohamed@us.ibm.com,g.russello@auckland.ac.nz

Abstract. The deployment of IoT devices is gaining an expanding interest in our daily life. Indeed, IoT networks consist in interconnecting several smart and resource constrained devices to enable advanced services. Security management in IoT is a big challenge as personal data are shared by a huge number of distributed services and devices. In this paper, we propose a **Coop**erative **D**ata **A**ggregation solution based on a novel use of **A**ttribute **B**ased signcryption scheme (Coop-DAAB). Coop-DAAB consists in distributing data signcryption operation between different participating entities (i.e., IoT devices). Indeed, each IoT device encrypts and signs in only one step the collected data with respect to a selected sub-predicate of a general access predicate before forwarding to an aggregating entity. This latter is able to aggregate and decrypt collected data if a sufficient number of IoT devices cooperates without learning any personal information about each participating device. Thanks to the use of an attribute based signcryption scheme, authenticity of data collected by IoT devices is proved while protecting them from any unauthorized access.

Keywords: IoT data aggregation · IoT applications · Resource-constrained devices· Constant size attribute based signcryption

1 Introduction

The Internet of Things (IoT) applications are deployed in several fields such as health care, smart cities, smart monitoring [1, 5]. IoT systems connect loosely defined objects, gateways and services that may exchange data about peoples' body state, life events, habits, location or professional information. In most of the cases, this data is sensitive and should be processed and managed with high security measures. Security management in IoT is a big challenge as sensitive data

are shared by a huge number of distributed services and devices. To face the exponential growth of data being generated by IoT devices and to efficiently ensure their collection, aggregation and sharing, fog and cloud computing are usually used to assist IoT devices since these latters are resource constrained. Although the usage of these environments has clear benefits, it brought new security and privacy threats as the data might be outsourced to untrusted environments. To countermeasure these threats, the data is usually obfuscated before being outsourced. In addition, the collected data from different IoT devices is generally aggregated to considerably save the energy resources and extend the lifetime of the IoT devices. However, the aggregation affects the security properties that might be provided by the protection schemes. For instance, the data needs to be authenticated at the aggregation phase to ensure that it is outsourced from benign devices while preserving their privacy.

Several works have been proposed to ensure data authentication based on signature schemes. However, these techniques are usually combined with encryption to provide data contents' secrecy. This combination incurs heavy computation and communication overhead due to the cumulative costs of encryption and signature. Signcryption [18] has been proposed by Zheng et al. as a cryptographic mechanism combining signature and encryption in only one phase. Signcryption allows an entity to encrypt and sign the ciphertext while incurring reduced computation costs compared to executing the encryption and the signature algorithms separately.

Attribute based signcryption (ABSC) is a signcryption primitive that ensures fine grained access control, data origin authenticity and data confidentiality thanks to the combination of attribute based encryption and signature in one logic step. Similar to other attribute based techniques, ABSC introduces one main drawback related to high computation and storage costs which depends on the size of used access policies. To mitigate this limitation, thus, attribute based signcryption schemes with constant computation costs and ciphertext sizes have been proposed [2].

Contributions — This paper extends our previous work [4] and it introduces Coop-DAAB, a cooperative privacy preserving attribute based signcryption mechanism based on the constant-size attribute based signcryption (ABSC) technique [2]. As such, our Coop-DAAB construction consists on performing the combined signing and encrypting processes of a set of data devices' inputs in a secure collaborative manner. The main idea behind Coop-DAAB relies on the distribution of the signcrypting operation among different devices, with respect to selected sub-sets of a general access predicate. That is, each device signcrypts its input data and sends the partial signcrypted information to an untrusted aggregator. This latter is capable of decrypting the received data only if a sufficient number of IoT devices cooperates.

Paper organization — section 2 introduces potential applications while identifying major security requirements. Section 3 introduces the network model and details Coop-DAAB concrete construction. The security analysis of Coop-

DAAB is discussed in section 4. Finally, performances analysis is detailed in section 5 and related work is discussed in Section 6 before concluding in section 7.

2 Motivating Applications and Requirements

The deployment of IoT devices is gaining an expanding interest in our daily life. Indeed, IoT networks consist in interconnecting several smart and resource constrained devices to enable advanced services. Security management in IoT is a big challenge as personal data are shared by a huge number of distributed services and devices. As reducing the amount of transmitted data can effectively save energy, aggregation services are generally applied to derive succinct contents. This technique can be an alternative that aims at providing security enhancement while reducing processing and communication overheads in several applications, namely *vehicular networks*, *mobile crowd sensing* and *service level agreement monitoring*.

Intelligent Transport Systems (ITSs) — In ITSs, connected cars are responsible for continuously publishing data related to their location and traffic status among the network. These broadcasted data need to be genuine in order to avoid injecting false data in the network. However, ensuring data authentication should not lead to a privacy leakage of personal data. To fulfill tis trade-off, collected data should be authenticated using privacy preserving techniques. The most prominent C-ITS solutions today allows the detection of traffic jam based on received data from the connected vehicles. Each connected car sends a set of collected data information, based on embedded sensors, to a central node. This latter needs to make sure that the received data are authentic. Then, to decrypt data, the aggregator merges received data from different devices and decrypts the contents to monitor the state of the road and supply the centralized infrastructure with necessary information.

Mobile Crowd Sensing — The popularity of increasingly capable human carried mobile devices such as smartphones and smart-watches involved several embedded sensors. This led to the appearance of the Mobile Crowd Sensing (MCS) paradigm. MCS is a sensing paradigm able to outsource collected data by sensors to a group of participating users, namely (crowd) workers. MCS is mainly based on the use of mobiles devices and their resources. Thus, data aggregation can be an efficient technique to ensure data collection while saving costs at the crowd worker side. Obviously, privacy preservation of the participating entities and data authentication should be considered in such environment to ensure the collection of benign data.

SLA Monitoring — Service Level Agreements (SLAs) are widely used to describe the agreements between customers and service providers regarding the quality of the provided services [11]. To measure the providers compliance to the SLA, monitoring tools should be setup to collect specific metrics that can quantify the Quality of Service (QoS). These monitoring mechanisms should run continuously or periodically but in either cases can generate huge amounts of data. Eventually, this data is shared with other components responsible for

performing analysis, reporting and executing SLA enforcement measures. The amounts of monitoring data can be huge and sending it through the wire can be an overkill for any system. Moreover, in most of the cases, this data is confidential and should be processed in a secure manner. Consequently, monitoring modules can act as an aggregator to reduce the amount of generated data. And to protect data from any tampering, they should be collected, authenticated and aggregated before being used by SLA services to estimate the QoS.

As described above, it is clear that designing a secure, privacy preserving and efficient data aggregation solution, for IoT applications has a high importance. The proposed scheme must fulfill the following properties:

- ***data confidentiality*** – the proposed aggregation mechanism should protect data contents from being accessed by unauthorized users.
- ***data origin authenticity*** – the aggregation scheme needs to ensure that data are created by authorized entities.
- ***privacy*** – the secrecy of devices' access pattern must be protected. Indeed, the aggregator must be able to verify data origin authenticity without leaking extra information about signing devices.
- ***low computation and storage costs*** – low processing complexities and storage costs need to be provided, by the proposed mechanism mainly for resource-constrained devices.

3 Coop-DAAB: Cooperative Attribute based Data Aggregation Scheme

In this section, we expose our network model then we present a general overview of Coop-DAAB. Finally, we detail the different phases of the proposed solution.

3.1 Network Model

Coop-DAAB network model relies on five different actors: an attribute authority, an administrator, a set of IoT devices $\{d_i\}_{\{i=1,\cdots,M\}}$, a set of gateway aggregating entities $\{Gw_j\}_{\{j=1,\cdots,L\}}$ and a selected IoT trusted node. These different entities are shown in Figure 1 and defined as follows:

- **attribute authority** AA – is responsible for bootstrapping the whole system in the initialization phase. We assume that AA is a trusted entity. It also issues certified attributes and related secret keys for the aggregating gateways and IoT devices.
- **administrator** $admin$ – is the system administrator responsible for generating the general access predicates used to encrypt data contents. In addition, $admin$ generates the signing access predicate submitted to the IoT devices to sign the collected data.
- **aggregating entity** (Gw) – is considered as a local network gateway. It is responsible of collecting, deciphering and verifying the authenticity of data contents.

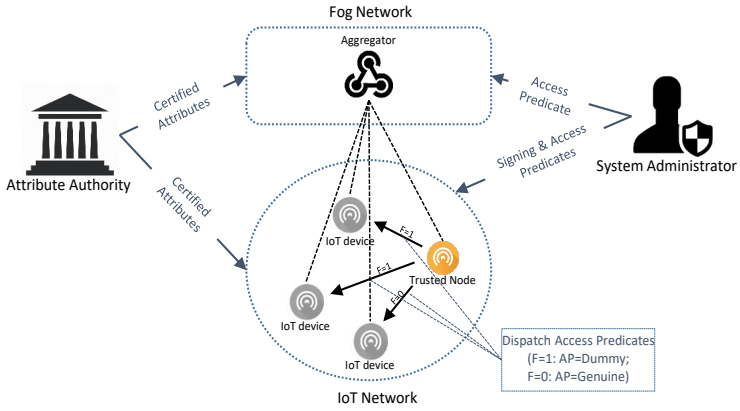

Fig. 1. Network Model

- **IoT device** (d) – collects, encrypts data then signs ciphertexts before forwarding to the aggregating entity.
- **trusted IoT node** (d_s) – is a trusted selected IoT device, periodically assigns to each involved IoT device d_i a sub-access predicate to be used for encrypting data.

3.2 Overview

In this paper, we design a new cooperative privacy preserving encryption scheme, for IoT signed data contents, denoted by Coop-DAAB with constant ciphertext size. Our proposal relies on the constant size attribute based signcryption proposed by Belguith et al. [2], which has been extended to support collaborative encryption of a set of data inputs, collected by IoT devices and gathered by an aggregator with respect to his granted privileges.

Our proposed Coop-DAAB construction involves three phases, namely, Sys_Init, Data_SignCrypt and Data_Agg.
During the first Sys_Init phase, three randomized algorithms are executed. First, the attribute authority AA performs the stp and keygen algorithms to generate the global public parameters and derive secret keys associated with each involved entity's attributes (i.e., IoT device, gateway). In addition, the system administrator executes accgen algorithm to generate the general access signing and enciphering predicates, denoted by Γ_s and Γ_e.
The second phase occurs periodically, such that each involved IoT device d_i has to signcrypt its collected data content and independently sends the resulting signcrypted information to the aggregating gateway. Note that the time period T is specified by the system administrator, during the Sys_Init phase. For the second Data_SignCrypt phase, two algorithms are performed, namely enc and sign, by each involved IoT device. For this purpose, a trusted selected IoT device

(d_s) periodically assigns to each involved IoT node a sub-access predicate. That is, the general enciphering access predicate is split by d_s into a set of dummy and genuine sub-access predicates. Note that a dummy access predicate refers to an access predicate such that the aggregating gateway Gw does not satisfy the required threshold (i.e., generally, the aggregator does not have any required attributes), while for a genuine access predicate, the gateway may have some of the required attributes. As such, when an IoT device d_i is assigned a dummy access predicate, denoted by γ_{d,d_i}, it has to encipher and sign its collected data content. And, when assigned a genuine access predicate, denoted by γ_{g,d_i}, it enciphers and signs a neutral group element $1_{\mathbb{G}}$. Afterwards, each IoT device d_i sends the signcrypted result to the aggregating node.
During the DATA_AGG phase, the aggregating gateway Gw gathers signcrypted data contents from involved IoT devices. Note that the aggregating gateway ignores assigned sub-access predicates, and is only aware of the general enciphering access predicate Γ_e, published by the system administrator. During this phase, four different algorithms are run by the aggregating gateway Gw, namely vrfchunk, agg, dec and vrfd algorithms. As such, Gw first verifies signed data chunks based on vrfchunk algorithm. Then, it merges the received signcrypted results, in order to generate a global ciphertext, based on the agg algorithm, deciphers the resulting global ciphertext, relying on dec algorithm and checks the authenticity of the received global data content, based on vrfd algorithm.

3.3 Coop-DAAB Phases

This section details the Coop-DAAB main phases and their algorithms.

3.3.1 Sys_Init Phase This first phase includes three randomized algorithms defined as follows:

- stp — the setup algorithm is executed by the attribute authority AA. The stp is responsible for generating the public parameters pp and the master secret key msk while taking as input security parameter ξ. For this purpose, the trusted authority defines a bilinear setting $(\hat{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}, g, h)$ of prime order p such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}$, $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$. In addition, It specifies an encoding function τ such that $\tau : \mathbb{U} \rightarrow (\mathbb{Z}/p\mathbb{Z})^*$, where $\mathbb{U}$ is the attribute universe of cardinal n. The function τ is chosen such that for each encoded attribute values $\tau(a) = x$ are pairwise different.
  Then, the stp algorithm selects a set $\mathcal{D} = \{d_1, ..., d_{n-1}\}$ consisting of $n-1$ pairwise different elements of $(\mathbb{Z}/p\mathbb{Z})^*$ (i.e; dummy users), which must also be different to the values $\tau(a_i)$, for all $a_i \in \mathbb{U}$. Note that for any integer i lower or equal to $n-1$, we denote as $\mathcal{D}_i$ the set $\{d_1, ..., d_i\}$. Finally, the stp algorithm computes u defined as $u = g^{\alpha \cdot \gamma}$ and outputs the global public parameters pp as follows:

$$\mathtt{pp} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}, \hat{e}, h, u, \{h^{\alpha\gamma^i}\}_{\{i=0,\cdots,2n-1\}}, \mathcal{D}, \tau, \hat{e}(g^{\alpha}, h)\}$$

The master key is set to be $\texttt{msk} = (g, \alpha, \gamma)$ where α, γ are two values randomly selected from $(\mathbb{Z}/p\mathbb{Z})^*$.

– keygen — the attribute authority executes the keygen algorithm once it receives a key generation request from any participating entity (i.e; IoT device, aggregating gateway). We denote by E, any participating entity, such that $E \in \{d, Gw\}$. The keygen algorithm takes as input the participating entity's set of attributes, denoted by A_E and the master key of the attributes authority msk and generates the corresponding secret key sk_E.
For any subset $A_E \subset \mathbb{U}$ of attributes associated with E, keygen picks a random value $r_E \in (\mathbb{Z}/p\mathbb{Z})^*$ and derives the secret key as follows:

$$\begin{aligned} sk_E &= (\{g^{\frac{r_E}{\gamma+\tau(a)}}\}_{a\in A_E}, \{h^{r_E\gamma^i}\}_{i=0,\cdots,m-2}, h^{\frac{r_E-1}{\gamma}}) \\ &= (sk_{E_1}, sk_{E_2}, sk_{E_3}) \end{aligned}$$

Remark 1. Communication overhead optimization for IoT devices' key distribution — Considering resource constraints of IoT devices, in terms of storage, processing and communications, our Coop-DAAB scheme assumes that IoT devices are pre-configured with corresponding secret keys. That is, the IoT device's manufacturer is responsible for contacting the attribute authority to physically embed the secret keys into the IoT device.

– accgen — the system administrator runs the accgen algorithm. It takes as input the attributes universe $\mathbb{U}$ and outputs the time period T, the access signing predicate Γ_s and the general access enciphering predicate Γ_e. Recall that T permits to regulate processing and transmitting signcrypted contents by IoT devices, periodically. Each access predicate is represented by a set of attributes $S \in \mathbb{U}$ and a threshold value t, such that at least t attributes need to be satisfied by an IoT device to sign or encrypt a data message. In the following, we denote by $\Gamma_s = (S_s, t_s)$ the access signing predicate, mainly used by IoT devices to prove the authenticity of their data contents, and $\Gamma_e = (S_e, t_e)$ the access enciphering predicate, mainly used by the aggregating gateway to decipher the resulting data contents.

3.3.2 Data_SignCrypt phase The second phase occurs periodically, such that each involved IoT device d_i has to signcrypt its collected data content and independently sends the resulting signcrypted information to the aggregating gateway. Recall that a trusted selected IoT device d_s periodically assigns to each involved IoT node a sub-access predicate. That is, the general enciphering access predicate Γ_e is divided into a set of dummy and genuine sub-access predicates such that:

$$\Gamma_e = \bigcup\{\{\gamma_{g,d_i}\}_{i=1,\cdots,k}, \{\gamma_{d,d_i}\}_{i=k+1,\cdots,l}\}$$

l represents the number of all participating IoT devices, k is randomly selected by d_s and represents the number of IoT devices that are assigned genuine access

predicates such that they have to encipher and sign a neutral group element $1_{\mathbb{G}}$, while the remaining $l-k$ devices are assigned dummy access predicates such that they encipher and sign their collected data contents [5].

This phase consists of two randomized algorithms, performed by each involved IoT device, detailed hereafter:

- enc — this algorithm takes as input a message m_i and the assigned sub-access tree γ_{F,d_i}, where $F=0$ presents a genuine sub-access predicate and $F=1$ denotes a dummy sub-access predicate. It outputs the encrypted message C_{d_i}. Note that each sub-access predicate is presented as $\gamma_{F,d_i}=(S_F,t_F)$, where S_F is the set of required deciphering attributes by γ_{F,d_i}, $|S_F|=s_F$ is the number of attributes of S_F and t_F is the threshold value w.r.t. γ_{F,d_i}.
  First, the device d_i picks a random $\kappa_i \in (\mathbb{Z}/p\mathbb{Z})^*$ and computes the enciphered message C_{d_i} defined as $C_{d_i}=(C_{1,i},C_{2,i},C_{3,i})$, defined as follows:

$$\begin{cases} C_{1,i} = (g^{\alpha\cdot\gamma})^{-\kappa_i} \\ C_{2,i} = h^{\kappa_i\alpha\cdot\prod_{a\in S_F}(\gamma+\tau(a))\prod_{d\in\mathcal{D}_{n+t_F-1-s_F}}(\gamma+d)} \\ C_{3,i} = \hat{e}(g,h)^{\alpha\cdot(\kappa_i+m_i)}\cdot f(m_i) = K_i\cdot f(m_i) \end{cases}$$

  where f is a bijective and semi-homomorphic function that is specified by *admin* (i.e; f depends on the use case), supporting the following property: $\prod_i f(m_i) = f(\sum_i m_i)$. For example, f may be the exponential function exp, where $\prod_i exp(m_i) = exp(\sum_i(m_i))$.
- sign — this algorithm is performed by the IoT device d_i to sign his encrypted data input, with respect to $\Gamma_s=(S_s,t_s)$, defined by the system administrator, where $S_s\subset\mathbb{U}$ is an attribute set of size $s=|S_s|$ such that $1\leq t_s\leq|S_s|$. Let A_{d_i} be the subset of attribute set related to the signing IoT device where $|A_{d_i}\cap S_s|=t_s$.
  For this purpose, each device d_i uses his secret key sk_{d_i} and the aggregate algorithm `aggreg` [6] [6] to output a signature σ_{d_i}. Indeed, d_i first computes T_{1,d_i} such as:

$$T_{1,d_i} = \texttt{aggreg}(\{g^{\frac{r_{d_i}}{\gamma+\tau(a)}},\tau(a)\}_{a\in A_{d_i}}) = g^{\frac{r_{d_i}}{\prod_{a\in A_{d_i}}(\gamma+\tau(a))}}$$

  Then, d_i defines the polynomial $P_{(A_{d_i},S_s)}(\gamma)$ such as:

$$P_{(A_{d_i},S_s)}(\gamma) = \frac{1}{\gamma}\Big(\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}\setminus A_{d_i}}(\gamma+\tau(a)) - B_{1,d_i}\Big)$$

  Where $B_{1,d_i}=\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}\setminus A_{d_i}}\tau(a)$

[5] Assigning sub-access dummy and genuine predicates may be set via activating a flag F, where $F=0$ presents a genuine sub-access predicate and $F=1$ denotes a dummy sub-access predicate

[6] Coop-DAAB relies on the aggregate algorithm `aggreg` introduced by Delerablee et al. [9].

Afterwards, using the element $sk_{d_i 2}$, the signcrypting device d_i derives B_{2,d_i} as follows:

$$B_{2,d_i} = h^{r_{d_i} P_{(A_{d_i}, S_s)}(\gamma)/B_{1,d_i}}$$

In the sequel, d_i generates the signature $\sigma_{d_i} = (\sigma_{1,i}, \sigma_{2,i}, \sigma_{3,i})$ defined as:

$$\begin{cases} \sigma_{1,i} = T_{1,d_i} \cdot g^{\frac{m_i}{\prod_{a \in A_{d_i}} (\gamma + \tau(a))}} \\ \sigma_{2,i} = sk_{d_i 3} \cdot B_{2,d_i} \cdot h^{m_i P_{(A_{d_i}, S_s)}(\gamma)/B_{1,d_i}} \\ \sigma_{3,i} = \hat{e}(g^{\alpha}, h)^{m_i} \end{cases}$$

Finally, the signcrypting IoT device d_i outputs the signcryption of the message m_i as follows:

$$\Sigma_i = (C_{d_i}, \sigma_{d_i}, B_{1,i})$$

Remark 2. Processing cost optimization for IoT devices' encryption and signature algorithms — Considering resource constraints of IoT devices, in terms of storage and processing, our Coop-DAAB scheme assumes that several elementary functions (i.e., computation of signcrypted message elements based on public parameters, such as $P_{(A_{d_i}, S_s)}(\gamma)$, ${C_{2,i}}^{{\kappa_i}^{-1}}, \cdots$) are outsourced to a semi-trusted device, with sufficient computation capacities [3]. As such, only a few number of exponentiations and multiplication is required by each single IoT device.

3.3.3 Data_Agg phase The DATA_AGG phase involves four algorithms, executed by the aggregating gateway Gw, defined as follows:

- vrfchunk — before starting the aggregation process, Gw has to verify that each received signcrypted message m_i has been correctly signed by a related device d_i. The vrfchunk algorithm takes as input the set of received signcrypted messages $\{\Sigma_i\}_{\{i=1,\cdots,l\}}$ and the public parameters pp. It outputs a boolean value $b \in \{0, 1\}$, where 0 means *reject* and 1 means *accept*. For this purpose, the aggregating gateway Gw has to check the following equality:

$$\sigma_{3,i} \stackrel{?}{=} \hat{e}(u^{-1}, \sigma_{2,i}) \cdot \hat{e}(g^{\alpha}, h)^{-1} \cdot \hat{e}(\sigma_{1,i}^{\frac{1}{B_{1,i}}}, h^{\alpha \cdot \prod_{a \in S_s \cup \mathcal{D}_{n+t_s-1-s}} (\gamma + \tau(a))}) \quad (1)$$

  Note that the aggregating gateway Gw performs the following algorithms of DATA_AGG phase relying on correctly signed data contents. That is, Gw withdraws inaccurate signatures.
- agg — this algorithm takes as input the set of signcrypted data chunks $\{\Sigma_i\}_{\{i \in [1,l]\}}$, where l is the number of participating IoT devices [7]. It outputs an aggregated signcrypted data message Σ w.r.t. a global message $M = \sum_{i=1}^{l} (m_i)$, as follows:

[7] For ease of presentation, we consider that all received signcrypted contents are correctly verified.

$$\begin{cases} C_1 = \prod_{i=1}^{l} C_{1,i} \\ C_2 = \prod_{i=1}^{l} C_{2,i} \\ C_3 = \prod_{i=1}^{l} C_{3,i} = \prod_{i=1}^{l} K_i \cdot f(m_i) \\ \sigma_3 = \prod_{i=1}^{l} \sigma_{3,i} \end{cases}$$

– dec — This algorithm is executed by the aggregating gateway. It takes as input the aggregated signcrypted message Σ, the set of attributes A_{Gw}, the secret key sk_{Gw} of the aggregating gateway and the enciphering access predicate Γ_e. It outputs the message M, such that $M = \sum_{i=1}^{l} m_i$. Indeed, any aggregating gateway Gw having a set of attributes A_{Gw} where $|A_{Gw} \cap S_e| = t_e$ can verify and decrypt the signcrypted message under the access policy $\Gamma_e = (S_e, t_e)$. Then, for all $a \in A_{Gw}$, Gw has to aggregate its secret keys related to the required attributes such as:

$$A_2 = \texttt{aggreg}(\{g^{\frac{r_{Gw}}{\gamma+\tau(a)}}, \tau(a)\}_{a \in A_{Gw}}) = g^{\frac{r_{Gw}}{\prod_{a \in A_{Gw}} (\gamma+\tau(a))}} \tag{2}$$

Afterwards, Gw defines the polynomial $P_{A_{Gw}}(\gamma)$ such as:

$$P_{A_{Gw}}(\gamma) = \frac{1}{\gamma}\Big(\prod_{a \in S_e \cup \mathcal{D}_{n+t_e-1-s_e} \setminus A_{Gw}} (\gamma + \tau(a)) - B_{Gw}\Big)$$

where $B_{Gw} = \prod_{a \in S_e \cup \mathcal{D}_{n+t_e-1-s_e} \setminus A_{Gw}} \tau(a)$

Afterwards, Gw uses the aggregated secret key A_2 and the $sk_{E_{Gw}}$ key element to compute:

$$[\hat{e}(C_1, h^{r_{Gw} P_{A_{Gw}}(\gamma)}) \cdot \hat{e}(A_2, C_2)]^{\frac{1}{B_{Gw}}} = e(g,h)^{(\sum_{i=1}^{l} \kappa_i \cdot \alpha) \cdot r_{Gw}}$$

Then, the aggregating gateway Gw deduces the deciphering key $K = \prod_{i=1}^{l} K_i$ such as:

$$\begin{aligned} K &= \hat{e}(C_1, sk_{Gw_3}) \cdot \sigma_3 \cdot \hat{e}(g,h)^{\sum_{i=1}^{l} \kappa_i \cdot r_{Gw} \cdot \alpha} \\ &= \hat{e}(g,h)^{\alpha \cdot (\sum_{i=1}^{l} m_i + \kappa_i)} \end{aligned}$$

Finally, the aggregating gateway Gw recovers the message $M = \sum_{i=1}^{l} m_i$ as follows:

$$M = f^{-1}(\frac{C_3}{K}) = f^{-1}(f(\sum_{i=1}^{l} m_i)) = \sum_{i=1}^{l} m_i$$

– vrfd — to verify the authenticity of the signature of the resulting message M, the gateway Gw uses the retrieved message M and the aggregated σ_3. That is, Gw verifies the correctness of following equality:

$$\sigma_3 \stackrel{?}{=} \hat{e}(g^{\alpha}, h)^{M} \tag{3}$$

4 Security Discussion

In this section, we analyse the security resistance of the proposed cooperative aggregation scheme Coop-DAAB with respect to the threat model presented in section 4.1 while proving the fulfillment of the security challenges defined in Section 2.

4.1 Threat model

In order to design a relevant aggregation scheme to secure IoT assisted applications, we define two potential attackers: *malicious IoT external device* and *honest but curious aggregating gateway*, defined as follows:

- *honest but curious aggregating gateway* – this aggregating gateway is honest in term of executing the protocols included in our proposed Coop-DAAB scheme. However, it may be curious about the participating entities sensitive data.
- *malicious IoT device* – this IoT device may be an external device trying to access aggregated data, to forge the signed data contents or to transmit false inputs. This adversary aims at persuading the gateway that he is a genuine IoT user.

4.2 Confidentiality

In our proposed Coop-DAAB scheme, data are encrypted before being stored using an attribute based signcryption scheme. Therefore, the secrecy of data is inherited from the used ABSC scheme.

Theorem 1. Coop-DAAB *ensures the secrecy of both encrypted data chunks and aggregated data contents.*

Sketch of Proof — the proof of Theorem 1 is twofold. First, the secrecy of signcrypted data contents depends on the security of the signcryption algorithm used to encrypt data chunks provided by IoT devices. Thus, Coop-DAAB inherits the indistinguishability property from [2], such that if a malicious attacker knows some data about the plaintext, it can not leak information about the ciphertext. Note that in ABSC schemes, the adversary may try to overcome the indistinguishability property using his own attributes or by colluding with other compromised users. Indeed, similar to [2], in Coop-DAAB the users secret keys are randomised using the r_E value which is unique for each participating entity. This stops the collusion attacks as users can not put their secret keys together and override their access rights. In addition, sub-access encrypting predicates used to encrypt data chunks are not communicated to the aggregator Gw. Furthermore, Gw's attributes do not satisfy sub-access policies used for encrypting genuine data contents phase thanks to the use of dummy sub-access polices. Consequently, an aggregator cannot deduce data chunk content.

Second, the secrecy of resulting aggregated contents depends on the consistency of the aggregating algorithm agg, such that aggregated data contents are only accessed by the authorized aggregator. Indeed, the general enciphering access predicate is published by the system administrator to the involved aggregating entities. As such, thanks to the use of the ABSC scheme, data are only accessed by users whose attributes match the defined access policy [2].

4.3 Privacy

Theorem 2. Coop-DAAB *ensures the privacy property, such that signing attributes are indistinguishable against a curious aggregating entity.*

Sketch of Proof — the proof of Theorem 2 states that an aggregating entity cannot guess which attributes have been used to signcrypt the data chunk. That is, let us consider a signing IoT device d, holding two different sets of attributes $A_{d,1}$ and $A_{d,2}$, that both satisfy a fixed access predicate $\Gamma^* = (S^*, t^*)$. Thus, d randomly selects a set $A_{d,b}$, where $b \in \{1, 2\}$ to sign a data message m, chosen by a malicious entity. As $|A_{d,b} \cap S^*| = |A_{d,1} \cap S^*| = |A_{d,2} \cap S^*| = t^*$, we deduct that the two signatures computed with respect to the two set of attributes $A_{d,1}$ and $A_{d,2}$ have similar distribution. Moreover, the used signcryption scheme [2] is demonstrated to be privacy preserving in the standard model. That is, while signing data, the identity of the signcrypting entity and its set of attributes are kept hidden from any verifying entity. Thus, Coop-DAAB guarantee that the applied signature does not leak any extra information about the signing entity neither its signing attributes except what can be already inferred from the used signing access policy.

4.4 Access Control to Data

Theorem 3. Coop-DAAB *provides an access to authenticated data contents.*

Sketch of Proof — the resistance of Coop-DAAB against unauthorized access to data relies on the correctness of the aggregation agg and decryption dec algorithms, as detailed in Lemma 1 and Lemma 2. In addition, the support of data origin authentication is provided by the correctness of the signing algorithm sign (c.f., Lemma 3) and its resistance against forgery attacks (c.f., Lemma 4).

Lemma 1. *Correctness of the aggregation of data chunks algorithm* agg — *The correctness of the aggregation of received signcrypted data chunks is detailed hereafter.*

$$\begin{cases} C_1 = \prod_{i=1}^{l} (g^{\alpha \cdot \gamma})^{-\kappa_i} \\ C_2 = \prod_{i=1}^{l} h^{\kappa_i \alpha \cdot \prod_{a \in S_F} (\gamma + \tau(a)) \prod_{d \in \mathcal{D}_{n+t_F-1-s_F}} (\gamma + d)} \\ C_3 = \prod_{i=1}^{l} \hat{e}(g,h)^{\alpha \cdot (\kappa_i + m_i)} \cdot f(m_i) \\ \sigma_3 = \prod_{i=1}^{l} \hat{e}(g^{\alpha}, h)^{m_i} \end{cases}$$

$$\begin{cases} C_1 = (g^{\alpha\cdot\gamma})^{-\sum_{i=1}^{l}\kappa_i} \\ C_2 = h^{\sum_{i=1}^{l}\kappa_i\alpha\cdot\prod_{a\in S_e}(\gamma+\tau(a))\prod_{d\in\mathcal{D}_{n+t_e-1-s_e}}(\gamma+d)} \\ C_3 = \hat{e}(g,h)^{\alpha\cdot\sum_{i=1}^{l}(\kappa_i+m_i)}\cdot f(\sum_{i=1}^{l}m_i) \\ \sigma_3 = \hat{e}(g^{\alpha},h)^{\sum_{i=1}^{l}(m_i)} \end{cases}$$

Lemma 2. *Correctness of the decryption algorithm* dec *— After aggregating its secret key relying on Equation 2, the aggregator calculates the decryption key K by executing the following equations:*
$\hat{e}(g,h)^{\sum_{i=1}^{l}\kappa_i\cdot r_{Gw}\cdot\alpha} = (\hat{e}(C_1, h^{r_{Gw}P_{A_{Gw}}(\gamma)}))\cdot\hat{e}(A_2,C_2))^{\prod_{a\in S_e\cup\mathcal{D}_{n+t_e-1-s_e}\setminus A_{Gw}}\tau(a)}$
$\hat{e}(g,h)^{\sum_{i=1}^{l}\kappa_i\cdot\alpha} = \hat{e}(C_1, h^{\frac{r_{Gw}-1}{\gamma}})\hat{e}(g,h)^{\sum_{i=1}^{l}\kappa_i\cdot r_{Gw}\cdot\alpha}$
$\hat{e}(g,\sigma_3) = \hat{e}(g,h^{\alpha\cdot\sum_{i=1}^{l}m_i}) = \hat{e}(g,h)^{\alpha\cdot\sum_{i=1}^{l}m_i}$

Finally, the aggregator may decrypt the message as follows:

$$\begin{aligned} M &= f^{-1}(\frac{C_3}{\hat{e}(g,h)^{\alpha\cdot\sum_{i=1}^{l}m_i}\cdot\hat{e}(g,h)^{\alpha\cdot\sum_{i=1}^{l}\kappa_i)}}) \\ &= f^{-1}(\frac{C_3}{K}) = f^{-1}(f(\sum_{i=1}^{l}m_i)) = \sum_{i=1}^{l}m_i \end{aligned}$$

Lemma 3. *Correctness of the signature verification algorithms* vrfchunk *and* vrfd *— First, the correctness of the algorithm* vrfchunk *relies on the correctness of Equation 4. In the following, we set the quantities* Ⓢ $= \hat{e}(u^{-1},\sigma_{2,i})\cdot\hat{e}(g^{\alpha},h)^{-1}\cdot\hat{e}(\sigma_{1,i}^{\frac{1}{B_{1,i}}}, h^{\alpha\cdot\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}}(\gamma+\tau(a))})$ *and* $\tau_\gamma(a) = \gamma+\tau(a)$*. The aggregating entity has to check if $\sigma_{3,i}$ is equal to* Ⓢ *such as:*

$$\begin{aligned} Ⓢ &= \hat{e}(u^{-1},\sigma_{2,i})\cdot\hat{e}(g^{\alpha},h)^{-1} \\ &\quad\cdot\hat{e}(\sigma_{1,i}^{\frac{1}{B_{1,i}}}, h^{\alpha\cdot\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}}(\tau_\gamma(a))}) \\ &= \hat{e}(g^{-\alpha\gamma}, h^{\frac{r_{d_i}-1}{\gamma}}\cdot h^{(r_{d_i}+m_i)P_{(A_{d_i},S_s)}(\gamma)/B_{1,d_i}}) \\ &\quad\cdot\hat{e}(g^{\alpha},h)^{-1}\cdot\hat{e}(\sigma_{1,i}^{\frac{1}{B_{1,i}}}, h^{\alpha\cdot\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}}(\tau_\gamma(a))}) \\ &= \hat{e}(g^{-\alpha}, h^{(r_{d_i}+m_i)\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}\setminus A_{d_i}}(\tau_\gamma(a))}) \\ &\quad\cdot\hat{e}(g,h)^{\alpha(1-r_{d_i})}\cdot\hat{e}(g^{\alpha},h^{(r_{d_i}+m_i)})\cdot\hat{e}(g^{\alpha},h)^{-1} \\ &\quad\cdot\hat{e}(\sigma_{1,i}^{\frac{1}{B_{1,i}}}, h^{\alpha\cdot\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}}(\tau_\gamma(a))}) \\ &= \hat{e}(g^{-\alpha}, h^{\frac{r_{d_i}+m_i}{B_{1,d_i}}\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}\setminus A_{d_i}}\tau_\gamma(a)})\hat{e}(g^{\alpha},h)^{m_i} \\ &\quad\cdot\hat{e}(g^{\frac{r_{d_i}+m_i}{B_{1,d_i}\prod_{a\in A_{d_i}}(\tau_\gamma(a))}}, h^{\alpha\cdot\prod_{a\in S_s\cup\mathcal{D}_{n+t_s-1-s}}(\tau_\gamma(a))}) \\ &= \sigma_{3,i} \end{aligned} \tag{4}$$

Second, the correctness of the signature verification vrfd *of the resulting aggregated data is based on the correctness of Equation 3. Such that, the aggregating entity relies on the received message* $M = \sum_{i=1}^{l}(m_i)$*, to check that* $\sigma_3 = \hat{e}(g^{\alpha}, h)^{\sum_{i=1}^{l}(m_i)} = \hat{e}(g^{\alpha}, h)^{M}$.

Lemma 4. *Unforgeability of the signing algorithm* sign *— As our* Coop-DAAB *is based on [2], it supports the unforgeability property of the signing algorithm, such that a malicious external device cannot provide a valid signcrypted data message as it does not satisfy the signing predicate* Γ_s*. Indeed, thanks to the randomization of the secret keys, unauthorized entities can not pool their attributes together to sign the data chunks. Thus, only authorized devices can generate genuine signcrypted data chunks.*

5 Performance Analysis

In this analysis we compare the computation and the storage overheads of the closely related attribute based signature schemes. In most ABSC schemes, the size of a signcrypted data grows along with the size of the encryption access policies [7, 13]. As Coop-DAAB relies on the constant ciphertext size ABSC scheme introduced in [2], it presents an efficient aggregation scheme in terms of processing and storage overheads. Indeed, as shown by Table 1, our contribution introduces a ciphertext size which is independent from the size of the used access policy.

Table 1. Features, Computation and Storage Costs Comparison of ABSC Schemes

Scheme	*Type*	Access Policy	*Key size*	*Signcryption size*	*Signcrypt time*	*Unsigncrypt time*
[13]	CP-ABE	Monotone	$l_s + 2, l_e + 2$	$\mathcal{O}(l_s) + \mathcal{O}(l_e)$	$\tau_p + E_T + E_1(3 + l_e + 3l_s)$	$\tau_p(3l_s + 5) + 2l_sE_T + 2l_sE_1$
[7]	CP-ABE	Threshold	$3l_e, 2l_s$	$\mathcal{O}(M) + \mathcal{O}(l_e) + \mathcal{O}(l_s)$	$E_1(2\mathcal{O}(M) + ml_e + 4l_e + 3 + l_s)$	$2E_1 + \tau_p(2t + 2 + l_s) + tE_T$
[16]	KP-ABE	Monotone	$m + l_s, m + l_e$	8	$E_1(10 + l_e + 4l_s)$	$6\tau_p + E_1(l_e + 3l_e)$
Coop-DAAB	CP-ABE	Aggregate-Threshold	$n_E + m + 1$	8	$E_1(t_s + 2) + 2E_2 + 2E_T$	$E_1t_e + 7\tau_p + 2E_T$

s denote both the size of the signing policy and the encryption policy. t_e and t_s, $\mathcal{O}(M)$ and m respectively define the encryption threshold, the signing threshold value, the size of the plaintext message M and the cardinal of the attributes' universe $\mathcal{U}$. n_E presents the cardinal of the set of attributes of the user. E_1, E_2, E_T represent exponentiation cost in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, while τ_p is the cost of a pairing operation.

Attribute based techniques have been implemented in resources-constrained devices in several research works [3, 8]. Based on our on going-implementation, we deduce that in Data_SignCrypt phase, the computation overhead raises linearly with the size of the access signing predicate (t_s) (c.f., Figure 3), due to the execution of aggreg [6] during the sign algorithm. Similarly, dec computation cost grows linearly with the size of the access enciphering predicate t_e. The algorithms vrfchunk, agg, enc and vrfd is independent of the sizes of the encryption and signing access policy. Recall that agg algorithm only involves multiplications which requires a negligible computation cost compared to pairing and exponentiation times.

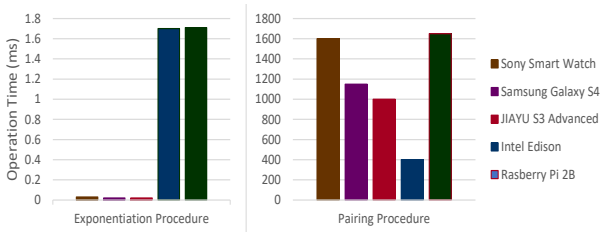

Fig. 2. Elementary functions Computation Costs

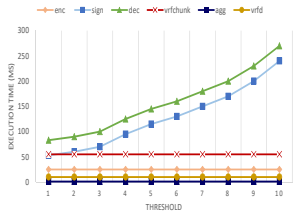

Fig. 3. Estimation of Coop-DAAB Computation Costs

The performances of attribute based techniques have been studied in several research works especially in IoT applications [3,8]. Our ongoing implementation of the Coop-DAAB's Proof of Concept (PoC) consists in evaluating the execution costs of the most known elementary cryptographic functions while performed in different IoT devices as presented in [3]. To evaluate the performances of Coop-DAAB, we executed three main cryptographic operations in different types of IoT devices, namely bilinear maps and exponentiation functions (cf. Figure 2).

Table 2. Selected Devices [3]

Device	Type	Processor
Sony SmartWatch 3 SWR50	Smart Watch	520 MHz Single-core Cortex-A7
Samsung I9500 Galaxy S4	Smartphone	1.6 GHz Dual-Core Cortex-A15
Jiayu S3 Advanced	Smartphone	1.7 GHz Octa-Core 64bit Cortex A53
Intel Edison	IoT Development Board	500 MHz Dual-Core Intel AtomTM CPU, 100 Mhz MCU
Raspberry Pi 2 model B	IoT Development Board	900 MHz Quad-Core ARM Cortex-A7

6 Related Work

IoT networks consist in interconnecting several smart and resource constrained devices to enable advanced services. Data aggregation has been widely explored, yet, few research works have focused on data aggregation in IoT networks while considering data secrecy and privacy preserving requirements.

Shi et al. [17] have proposed a privacy preserving aggregation technique. In this scheme, an aggregator is able to collect participants' data and run statistics over them without learning private information about each participant.

In [14], the authors design a data aggregation scheme adapted for fog computing systems. This solution relies on the use of the Chinese Remainder Theorem to aggregate received data from different parties. This scheme applies one-way hash chain to ensure data origin authentication. In the same vein, Lyu et al. proposed PPFA, a Privacy Preserving Fog-enabled Aggregation for smart grid environments [15]. Their construction relies on fog nodes computation resources to perform heavy computation-consuming functions. Later, Hu et al. introduced a privacy preserving data aggregation scheme for IoT applications [10]. The proposed scheme relies on Secure Multiparty Computation (SMC) techniques, such that each device has to first divide sensory data, locally keeps one piece, and sends the remaining pieces to other group devices. Then each IoT device adds the received shares and the held piece together to get immediate result. The [10] construction provides heavy computation and communication costs. Hence, it makes it unsuitable for resource-constrained devices.
Recently, in 2018, a lightweight aggregation signature scheme for IoT environments have been proposed in [12]. This scheme is based on the use of a set homomorphic signature scheme to aggregate received signed data from IoT devices without learning secret keys of each participant. However, data are transmitted in clear text, between the different involved devices.

Signcryption schemes are generally considered as a logic combination of encryption and signature schemes that enables a data owner to encrypt and sign data in one step. This cryptographic technique allows data origin authentication as the receiver verifies the data owner signature before decrypting. Attribute based signcryption schemes have been introduced by Gagné et al. [7], in 2010. This first proposed construction combines attribute based encryption and signature schemes, based on the same access structure. As such, data secrecy and data origin authentication and flexible fine-grained access control features are provided. Nevertheless, Gagné et al. construction suffers from an important communication overhead that increases dependently with the number of attributes involved in the access structure. Recently, Belguith et al. introduced a constant-size threshold attribute based signcryption, referred to as t-ABSC, for cloud applications [2]. The proposed construction ensures both fine-grained access control and data origin authentication, thanks to the usage of two different access policies, assigned respectively to the enciphering and signing process. As such, users' privacy and outsourced data confidentiality are ensured. In addition, the size of signcrypted messages does not depend on the number of attributes involved in the threshold access structure, which makes t-ABSC scheme suitable for bandwidth-limited applications and resource-constrained devices.

7 Conclusion

Several security and privacy concerns have raised, due to the emergence of Cloud assisted IoT applications, considered as highly dynamic and distributed environments. This lead us to design a new cryptographic mechanism to ensure cooperative data aggregation for IoT applications while preserving devices' privacy, thanks to the attractive properties of attribute based cryptographic techniques. The proposed Coop-DAAB scheme enables an edge device, i.e., aggregator to collect sensory data from different IoT devices and verify their authenticity using an attribute based signcryption scheme. The privacy of involved IoT devices is ensured, thanks to the intrinsic properties of the signing procedure, as it does not reveal more information other than the accuracy of data integrity verification. Furthermore, compared to most closely related work, Coop-DAAB is suitable for resource-constrained devices based on an ongoing implementation of the proposed construction and a detailed theoretical performance analysis w.r.t. computational, communication and storage costs. The analysis clearly shows that the size of the signcrypted data does not depend on the number of attributes involved in the threshold access structure.

References

1. Atwady, Y., Hammoudeh, M.: A survey on authentication techniques for the internet of things. In: Proceedings of the International Conference on Future Networks and Distributed Systems. p. 8. ACM (2017)
2. Belguith, S., Kaaniche, N., Laurent, M., Jemai, A., Attia, R.: Constant-size threshold attribute based signcryption for cloud applications. In: SECRYPT 2017: 14th International Conference on Security and Cryptography. vol. 6, pp. 212–225. Scitepress (2017)
3. Belguith, S., Kaaniche, N., Laurent, M., Jemai, A., Attia, R.: Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot. Computer Networks **133**, 141–156 (2018)
4. Belguith, S., Kaaniche, N., Mohamed, M., Russello, G.: C-absc: Cooperative attribute based signcryption scheme for internet of things applications. In: Proceedings of the International Conference On Services Computing IEEE SCC. p. 6. IEEE (2018)
5. Coates, A., Hammoudeh, M., Holmes, K.G.: Internet of things for buildings monitoring: Experiences and challenges. In: Proceedings of the International Conference on Future Networks and Distributed Systems. p. 38. ACM (2017)
6. Delerablée, C., Pointcheval, D.: Dynamic threshold public-key encryption. In: Annual International Cryptology Conference. Springer (2008)
7. Gagne, M., Narayan, S., Safavi-Naini, R.: Threshold attribute-based signcryption. In: Security and Cryptography for Networks. Springer (2010)
8. Guo, L., Zhang, C., Yue, H., Fang, Y.: Psad: A privacy-preserving social-assisted content dissemination scheme in dtns. IEEE Transactions on Mobile Computing **13**(12), 2903–2918 (2014)
9. Herranz, J., Laguillaumie, F., Ràfols, C.: Constant size ciphertexts in threshold attribute-based encryption. In: International Workshop on Public Key Cryptography. pp. 19–34. Springer (2010)

10. Hu, C., Luo, J., Pu, Y., Yu, J., Zhao, R., Huang, H., Xiang, T.: An efficient privacy-preserving data aggregation scheme for iot. In: International Conference on Wireless Algorithms, Systems, and Applications. pp. 164–176. Springer (2018)
11. Kaaniche, N., Mohamed, M., Laurent, M., Ludwig, H.: Security sla based monitoring in clouds. In: 2017 IEEE International Conference on Edge Computing (EDGE). pp. 90–97 (June 2017)
12. Kaâniche, N., Jung, E.E., Gehani, A.: Efficiently validating aggregated iot data integrity
13. Liu, J., Huang, X., Liu, J.K.: Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption. Future Generation Computer Systems **52** (2015)
14. Lu, R., Heung, K., Lashkari, A.H., Ghorbani, A.A.: A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot. IEEE Access **5**, 3302–3312 (2017)
15. Lyu, L., Nandakumar, K., Rubinstein, B., Jin, J., Bedo, J., Palaniswami, M.: Ppfa: Privacy preserving fog-enabled aggregation in smart grid. IEEE Transactions on Industrial Informatics (2018)
16. Rao, Y.S., Dutta, R.: Efficient attribute-based signature and signcryption realizing expressive access structures. International Journal of Information Security **15**(1), 81–109 (2016)
17. Shi, E., Chan, H., Rieffel, E., Chow, R., Song, D.: Privacy-preserving aggregation of time-series data. In: Annual Network & Distributed System Security Symposium (NDSS). Internet Society. (2011)
18. Zheng, Y.: Digital signcryption or how to achieve cost (signature & encryption) ¡ cost (signature)+ cost (encryption). In: Advances in Cryptology, Crypto97. Springer (1997)