Computers, Materials & Continua
DOI:10.32604/cmc.2021.018549
Article

Tech Science Press

A Lightweight Certificate-Based Aggregate Signature Scheme Providing Key Insulation

Yong-Woon Hwang and Im-Yeong Lee*

Department of Software Convergence in Soonchunhyang University, Asan-si, 31538, Korea
*Corresponding Author: Im-Yeong Lee. Email: imylee@sch.ac.kr
Received: 12 March 2021; Accepted: 16 April 2021

Abstract: Recently, with the advancement of Information and Communications Technology (ICT), Internet of Things (IoT) has been connected to the cloud and used in industrial sectors, medical environments, and smart grids. However, if data is transmitted in plain text when collecting data in an IoT-cloud environment, it can be exposed to various security threats such as replay attacks and data forgery. Thus, digital signatures are required. Data integrity is ensured when a user (or a device) transmits data using a signature. In addition, the concept of data aggregation is important to efficiently collect data transmitted from multiple users (or a devices) in an industrial IoT environment. However, signatures based on pairing during aggregation compromise efficiency as the number of signatories increases. Aggregate signature methods (e.g., identity-based and certificateless cryptography) have been studied. Both methods pose key escrow and key distribution problems. In order to solve these problems, the use of aggregate signatures in certificate-based cryptography is being studied, and studies to satisfy the prevention of forgery of signatures and other security problems are being conducted. In this paper, we propose a new lightweight signature scheme that uses a certificate-based aggregate signature and can generate and verify signed messages from IoT devices in an IoT-cloud environment. In this proposed method, by providing key insulation, security threats that occur when keys are exposed due to physical attacks such as side channels can be solved. This can be applied to create an environment in which data is collected safely and efficiently in IoT-cloud is environments.

Keywords: Internet of things; certificate-based aggregate signature; key insulation; cloud; lightweight; physical attack

1 Introduction

Recent developments in Information and Communications Technology (ICT), the Internet of Things (IoT) have facilitated industrial "smartization"; smart factories and smart industries that link the real and virtual worlds via Cyber Physical Systems (CPS). A CPS processes tasks and information of the physical world in virtual space using IoT and other networks, and continuously adapts to changes without human intervention. If a CPS is to function well, the nature of the IoT environment is important because many physical things must be connected to sensors

and communication devices. In an IoT-cloud environment, data is collected from IoT devices and stored safely in the cloud, whereby legitimate users can access the cloud and check the data. It is currently mainly used in Industrial Internet of Things (IIoT) environments such as manufacturing, transportation, and energy fields, as well, as medical environments and smart homes [1–3]. However, if sensors transmit plain text, data forgery and replay attacks are possible. In particular, millions of dollars worth of assets could be at risk if communications are not secured in large network systems such as IIoT environments [4].

To solve this problem, a lightweight cryptography technology is required to provide data confidentiality, and a digital signature technology that ensures the integrity of the data generated by a sensor device in an IoT environment is necessary. A sensor signs all messages and passes them (through a gateway) to the cloud as shown in Fig. 1. A user verifies the signature, ensuring message integrity. In an IoT environment, users can receive safe services only when the integrity of data collected from all devices is ensured. In addition, in an environment where large-scale data is collected such as IIoT, the concept of aggregation is important to efficiently distribute data.

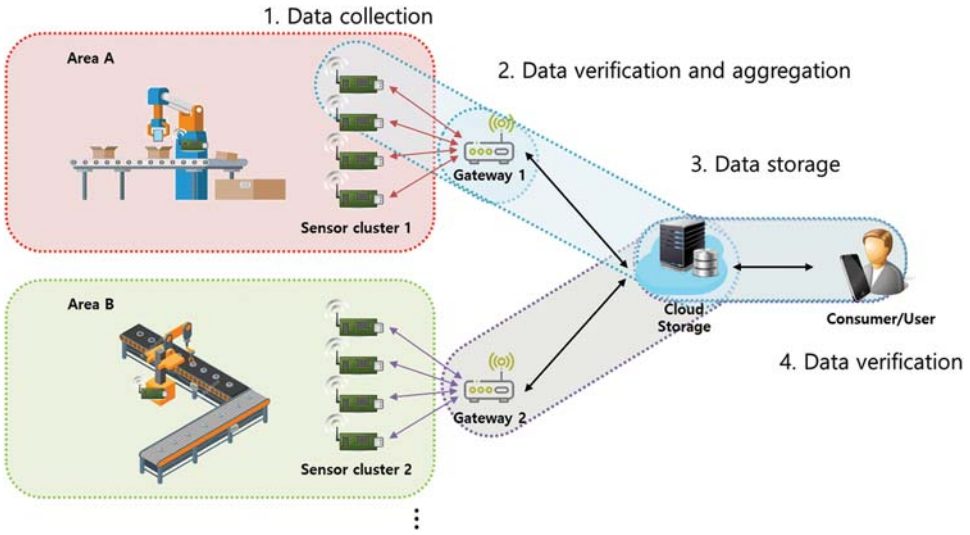

Figure 1: Data signing and verification process in the IoT-cloud (e.g., smart factory)

However, among previously studied aggregate signature methods, there are many methods that create aggregate signatures using pairing operations. This is inefficient. The computational burden rises as the number of signatories increases. An early digital signature method used a Public Key Infrastructure (PKI) encryption system. This was followed by signature and aggregate signature methods employing Identity-Based Cryptography (IBC), Identity-Based Signatures (IBS), Identity-Based Aggregate Signatures (IBAS), and Certificateless Cryptography (CLC). However, both the IBC and CLC methods pose key escrow and/or key distribution problems [5–7]. To solve these problems, Certificate-Based Signatures (CBS) and Certificate-Based Aggregate Signatures (CB-AS) have been proposed, and studies are being conducted to ensure they meet a number of security requirements, such as data integrity, non-repudiation, and resistance to the forgery of signatures.

In this paper, we analyze CB-AS and propose a new lightweight CB-AS scheme for IoT-cloud environments. Specifically, this proposed method is an efficient CB-AS method that provides key insulation, and the contribution of the paper is as follows: 1) The signature cannot be forged

by an attacker. 2) In order to solve the key exposure problem due to physical attacks such as side-channel attacks, key insulation is applied to continuously update the signature keys generated in the sensor devices. 3) The size of the entire signature is reduced in group environments, such as IIoT, by a gateway collecting and aggregating signed messages from multiple sensor devices. The final verifier can verify the signatures for multiple messages with one verification, and this provides integrity and non-repudiation functions for messages. 4) Since it is performed using pairing-free operations in the signature generation, aggregation, and verification steps, the operation efficiency is high compared to the aggregate signature methods proposed using pairings.

The proposed scheme in this paper focuses on integrity rather than confidentiality of data. It will be applicable to a sensor device network environment that safely collects data continuously in real time in an IIoT environment that has been expanded to an IoT-cloud environment. The paper is structured as follows. Section 2 deals with related studies and explains the CBS and CB-AS constructs, security threats, and previously studied key insulation and CB-AS systems. Section 3 introduces the security requirements of Certificate-Based Key Insulated Aggregate Signatures (CB-KIAS). Section 4 describes our method in detail. Section 5 analyzes the security and efficiency of the method, and Section 6 is the conclusion.

2 Background

This section describes the Elliptic Curve Discrete Logarithm Problem (ECDLP), digital signatures, and key insulation. It also reviews studies on CB-AS.

2.1 ECDLP

Elliptic curve encryption is a public key encryption method using the fact that the discrete logarithm problem on the elliptic curve is difficult. Compared to conventional public key cryptography, the same security can be obtained with fewer bits, encryption is processed at a high speed, and key management is easy because a short key is used. Therefore, it is widely used in IoT and other lightweight environments. The elliptic curve for elliptic curve encryption is a set of solutions (X, Y) of the equation $y^2 = x^3 + ax + b\,(mod\,p)$ defined for arbitrary integers a, b. The fact that the point P = (X, Y) is on the elliptic curve means that the above equation is satisfied, and $Q = x \cdot P$ can be defined for any integer x for two points P, Q. Finding the solution x is a problem of discrete logarithm elliptic curves. In other words, it is easy to find Q using $x \cdot P$ in $Q = x \cdot P$, but it is very difficult to infer the x value even if you know Q and P [8].

2.2 Digital Signatures

A digital signature is a method of verifying one's identity in a network, and generally uses cryptographic technology related to Public Key Cryptography (PKC). When a message is signed with the sender's private key and transmitted, the receiver uses the sender's public key to verify the validity of the signature. This provides a non-repudiation function via the signed message, that can prove that a message has been transmitted from the sender and provides the signer's authentication and message integrity.

Digital signatures started from PKC based signatures and have evolved into various types of signatures such as IBS, Certificateless Signature (CLS), and CBS, and research on signatures that aggregate many signatures into one signature is also being conducted [9–12]. PKC is a security technology that utilizes public key cryptography and can provide various functions such as signing and authentication. However, since PKC uses a certificate to verify a user's public key, there is a

storage problem for the certificate and overhead for management such as distribution, verification, and revocation.

Therefore, Shamir developed an IBS. In IBS the public key is a user identifier, eliminating the certificate management overhead [13–15]. However, a key escrow problem occurs because all users require private keys [16–18].

To solve this problem, Gentry et al. (2003) developed a (CBC) scheme that combined the advantages of the PKC and IBC. A user creates a public/private key pair and receives a certificate with an identity and a public key from a trusted Certification Authority (CA). The CBC certificate serves as a private (secret) key for the user. Signing and decryption are performed employing the user's certificate and private key simultaneously. Thus, the key escrow problem caused by the Key Generation Center (KGC) issuance of keys to each IBS owner is solved, and certificate management (problematic with the earlier PKC schemes) is simplified. Additionally, public key verification overload is eliminated. There are CBS and CB-AS techniques using CBC [19–26].

2.3 Key Insulation

In signature-based methods, the private key used in signing must be absolutely secure. If this key is exposed, there are several security threats to the smart factory of Fig. 1. Assuming that the attacker has exposed a key by physically attacking a sensor in the factory, the attacker can use that key to forge a message and signature from the sensor. This can trigger errors in manufacturing and the entire system could stop. This is fatal in a large, interconnected environment such as a smart factory or other IIoT. It is essential to safely manage and store the private keys, and among the various methods to do this is key insulation technology.

Key insulation was introduced by Dodis in 2002. Users update their private keys using a physically secure device termed a Helper (Fig. 2) [27]. Each user creates a public key, a private key, and a temporary secret key for initial signature. After that, the helper takes the user's public key, creates key, known as an update key, that can be used for period t, and sends it to the user. With the received update key, the user updates the existing signature key with a signature key that can be used for period t. Then, the Helper issues a new key to be used during the following period, t′. The IBC, CBC, and CLS methods use key insulation methods to solve the problem of key exposure [28–31].

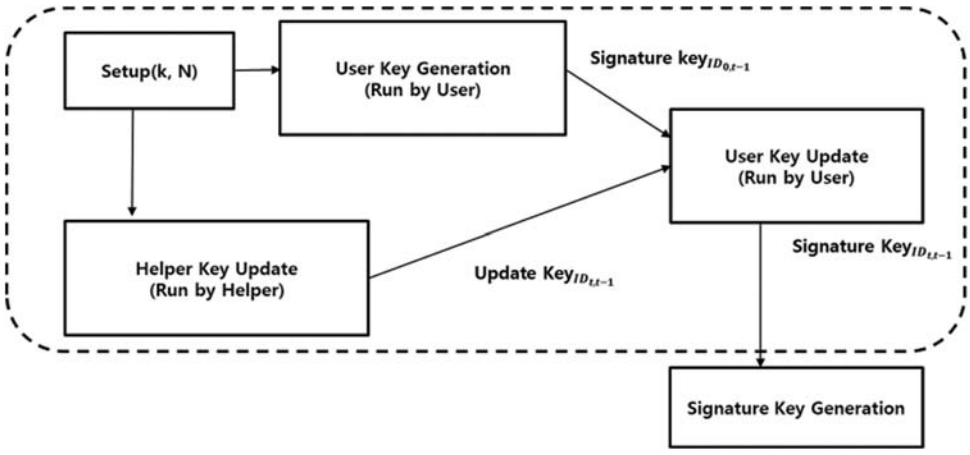

Figure 2: Signature key generation structure using key insulation

2.4 Certificate-Based Aggregate Signatures (CB-AS)

A CBS is a signature for a single message. The CB-AS method collects multiple messages and creates an aggregated signature. In CBS, if multiple senders transmit data, a signature is generated for each message. Thus, the number of signatures increases as the number of signatories rises, and the verifier must check all signatures. CB-AS aggregates the multiple signatures into one. Sender public keys are used for verification, but all signatures are verified together. This reduces verification and memory overheads, computing cost, and bandwidth. Many pairing-free-based CB-AS methods have been applied in IIoT environments such as smart factories [23–26].

2.4.1 CB-AS Model

Fig. 3 is a diagram showing the basic structure of CB-AS. The CB-AS method features a CBS that first registers a sensor with a CA and then issues a certificate. In general, the technique features seven phases. The Setup and Certificate Generate (CertGen) phases are performed by the CA. During Setup, the public parameters and a master CA key are created. In the CertGen phase, when a user requests, a certificate corresponding to the user ID and public key is generated and transmitted. In the Key Generate (KeyGen) phase, the user (signatory) generates a public/private key pair and then a signature for messages using the certificate, the public and private keys, and his/her identity. In the Sign phase, the user signs the message and sends it to the gateway, which verifies the message and signature. Messages and signatures from multiple signatories are aggregated into one by the gateway and transmitted to cloud storage. The verifier checks all messages and the aggregated signature [23–26].

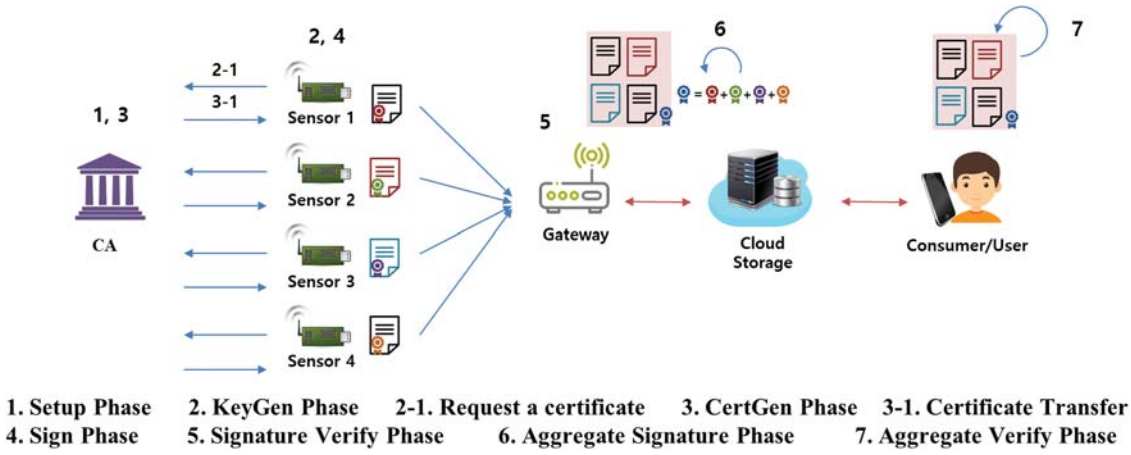

Figure 3: Structure of certificate-based aggregate signature

2.4.2 CB-AS Attack Model

The CB-AS method is vulnerable to forgery of messages and signatures. The public CB-PKC key can be authenticated via a certificate, but an attacker may substitute the public key of another user. A public key replacement attack on a CBS may forge the signature that device (A) sends to device (B) and replace the public key of device (A) (which is open for signature verification) with a public key generated by the attacker. Such an attack is possible because the substituted public key of the attacker, which can bypass verification of a signature generated using the private key of device (A), cannot be authenticated as the public key of device (A). In addition, the CA can forge the signature of device (A) using a certificate [25–27].

The security model of CB-AS must counter two types of attacks. The two models are similar to games in which competent attackers A_1 and A_2 communicate with Challenger (C) to successfully forge signatures. In the oracle model, Challenger (C) is responsible for calculating and executing a value when a user requests it. A_1 serves as an outsider who can arbitrarily substitute a new key for the public key of a legitimate user, but does not know the certificate or master key. A_2 is an attacker who can act as a malicious CA or control the CA master key, but cannot replace the public keys of users. If A_1 or A_2want to forge signatures, the signature can be forged by repeatedly executing a number of queries with attack type 1 and 2 below. The security threats mentioned above occur not only with CB-AS, but also with CB-KIAS where key insulation is provided. When comparing CB-KIAS and CB-AS, it includes a key update phase and a signing key generation phase.

Security Attack Type I, Adversary A_1

In Security Attack Type I, II, an attack is performed based on the CB-KIAS model, which provides a key insulation function.

• Setup: Challenger (C) creates a CA master public/private key and system parameters by executing this phase.

• KeyGen: The counterfeiter (attacker) sends its identity to obtain a key. Challenger (C) gets the key, creates a public key PU_i and delivers it to the attacker.

• PublicKeyReplace: The attacker can replace the user's public key with PU_i. It is not necessary to obtain the user's private key. The attacker can repeat this phase.

• CertGen: The attacker requests authentication for (ID, PU_i) and Challenger (C) sends the certificate obtained by executing CertGen to the attacker.

• Signature Key: The attacker makes a request to obtain a signing key, and Challenger (C) generates a temporary signature key $Tsk_{ID_i,0}$ and sends it to the attacker A_1.

• Sign Generate (SignGen): The attacker requests the (ID, m) signature and Challenger (C) executes SignGen to obtain that signature. Then, Challenger (C) sends the signature to the counterfeiter and records the response.

It is possible to use the oracle to query Setup, KeyGen, PublicKeyReplace, CertGen, Signature Key and SignGen (depending on the competence of A_1). Then the attacker (counterfeiter) can output $\left(ID', PU_i', \sigma', \mathrm{m}', t'\right)$.

–A_I has never requested ID' through the Signature Key oracle.

–A_I has never requested (ID', PU_i') through the Signature Key oracle.

–$(\sigma', \mathrm{m}', t')$ is a legitimate counterfeit message and signature pair, but A_1 has never requested $\left(ID', \mathrm{m}', t'\right)$ through the SignGen oracle.

Definition 1. In CB-AS Attack Type I, if there is no attacker A_1 who can win with a non-negligible probability within probabilistic polynomial time (PPT), it is existentially impossible to engage in forgery.

Security Attack Type II, Adversary A_2

• Setup, KeyGen, CertGen, Signature Key, SignGen: Same as the setup of Attack Type I, and parameter values are send to A_2.

• Helper Key: When the counterfeiter receives the Helper key query, Challenger(C) generates the Helper's private key and public key $(\mathrm{hsk}_1, \mathrm{hsk}_2, \mathrm{hpk}_1, \mathrm{hpk}_2)$ and sends them to the counterfeiter A_2. As A_2 knows the CA master key, she/he can mount a forgery attack, and can execute the Setup, KeyGen, Helper Key, Signature Key, SignGen oracle queries. Then, the counterfeiter can output $(ID', PU_i', \sigma', \mathrm{m}')$.

–A_2 has never requested an ID' through the Signature Key oracle.

–$(\sigma', \mathrm{m}', t)$ is a valid counterfeit message and signature pair, but A_2 has never requested (ID', m', t) through the SignGen oracle.

Definition 2. In CB-KIAS Attack Type II, it is existentially impossible to engage in forgery in the absence of an attacker A_2, who can win with a non-negligible probability within probabilistic polynomial time (PPT).

2.5 Analysis of Existing CB-AS Schemes

CB-AS was first proposed by Gentry in 2003, and recently proposed CB-AS schemes have many efficient methods that do not use pairing operations. Tab. 1 shows existing CBS, IBAS, and CB-AS methods that have been studied as well as the proposed method. The Li et al. [32] scheme solves the problem of key exposure due to physical attack by using a certificate-based signature using a key insulation technique. However, since it uses a pairing operation, there is a disadvantage in that the computational amount is high, and in 2017, Lu et al. [30] raised the problem that the signature could be forged by a malicious CA in the Li et al. scheme. This is because some (e.g., $\mathrm{l(M)}^{r_m}$) of the signature that contains the message is not related to the signer's private key. Thus, a malicious CA could forge the signature by removing $\mathrm{l(M)}^{r_m}$ and adding $\mathrm{l(M')}^{r'_m}$ without affecting the validity of the signature.

Table 1: Comparison of certificate-based signature, certificate-based aggregate signature, identity-based aggregate signatures schemes

Scheme	Type	Key escrow	Aggregate	Key insulated	Pairing operation
Li et al. [32] scheme	CBS	No	X	X	◯
Xiong et al. [22] scheme 1		No	X	◯	X
Reddy et al. [29] scheme	IBAS	Yes	◯	◯	◯
Shen et al. [11] scheme		Yes	◯	X	◯
Chen et al. [24] scheme	CB-AS	No	◯	X	X
Verma et al. [25] scheme 1		No	◯	X	X
Verma et al. [26] scheme 2		No	◯	X	X
Xiong et al. [31] scheme 2		No	◯	◯	◯
Proposed scheme goal		No	◯	◯	X

Notes: ◯: Provided; X: Not provided .

Xiong et al. [22] proposed an efficient CBS scheme that does not use a pairing operation. As a feature, a key insulation technique is applied to solve the security threats that occur when the key is exposed by a physical attack. The system affords high computational efficiency in lightweight IoT environments. However, as it lacks an aggregation function, it is necessary to individually verify all data from multiple sensors.

The Reddy et al. [29] scheme and the Shen et al. [11] scheme employ identity-based aggregate signatures, but only the Reddy et al. scheme features key insulation. Since both methods use pairing, and the larger the number of signatures, the higher the amount of computation. Also, since IBS allow the KGC to create a user's key, curiously, a key escrow issue can arise from KGC. To solve the key escrow problem, use need to create their own keys for signing.

The methods of the Chen et al. [24] scheme and the Verma et al. second method feature CB-AS. Either can be used in IoT and IIoT environments because computation is efficient (pairing is not employed). However, key insulation is not provided. Xiong et al. showed that one of the methods of Verma et al. was susceptible to public key replacement attack and signature forgery by a malicious CA. As in Attack Type I in 2.4.2, assuming that the attacker A_1 tries to forge a valid signature on all messages m_i represented by users with ID_i and public keys Yi, in Verma et al. $H_0(ID_i \| Yi)$, which is calculated during the signature process of method 1, can be removed and replaced with $H_0(ID_i' \| Yi')$. Since this does not contain the random value generated during the signing process, the attacker can easily erase $H_0(ID_i \| Yi)$ and replace it. Attack Type II is similar to Attack Type I, but in Attack Type II, an attacker can substitute $H_1(m_i' \| ID_i' \| Yi' \| par)$ for $H_1(m_i \| ID_i \| Yi \| par)$ in the signing process, and can successfully forge the signature even when the user's secret key is not known [31].

Xiong et al. [31] proposed a second method. It is a CB-KIAS proposed to solve the signature forgery problem occurring in the Verma et al. method discussed above. This Xiong et al. method includes both the aggregation function and key insulation function required by this proposed scheme. However, since a pairing operation is used, there is a disadvantage considering the amount of operations increase according to the number of signatures when generating signatures, verifying signatures, and verifying aggregated signatures.

The goal of our proposed method is to provide overall operational efficiency by not using a pairing operation unlike the second Xiong et al. method. In addition, we will solve the problem of forged signatures due to the attacks presented in Section 2.4.2, and respond to security threats that occur when keys are exposed due to physical attacks through periodic key updates in a key insulation method.

3 Security Requirements

• **Data Integrity and Reliability:** It is essential to ensure the integrity and reliability of data (messages) transmitted to/from sensors in an IoT environment. In existing CB-AS systems, the gateway individually verifies data transmitted by multiple sensors, and aggregates and transmits this information. A user who wishes to check the data must verify the final aggregated signature. Such verification is essential to ensure message integrity/non-repudiation, and the reliability of devices that transmit and receive messages.

• **Unforgeablity:** As described in Section 2.4.2, signature forgery can occur via an A_1 instigated public key substitution attack or when A_2employs the CA master key to create a user certificate. A_1 should not be able to generate a legitimate signature even via public key replacement employing a valid user identity. A_2 should not be able to forge a signature even if that signature is generated using both the signatory's key and certificate. Therefore, a verifier should not be able to verify a forged signature.

• **Side-channel Attack Key Exposure Prevention:** Messages are signed to ensure integrity and sensor reliability. That is, the signer's (sensor device's) signature key should not be leaked to the outside or extracted through a public value. If an attacker can deduce or steal the signature key

through a physical attack, such as a side-channel attack, the attacker can forge the signature of the generated messages. This reduces the reliability of IoT sensor devices, and an attacker can forge and transmit any number of messages through the extracted signature key value. Therefore, by applying a method such as key insulation, the signature key of the signer will be continuously updated.

4 The Proposed CB-KIAS Scheme

In this paper, we propose a certificate-based aggregate signature scheme that provides key insulation for secure data collection and processing through sensor networks in IoT-cloud environments. Fig. 4 is a schematic diagram of the scenario of this proposed method. In a network environment connected to a virtual CPS space, sensors generate data, sign messages, and send them to a gateway. Each gateway serves as an aggregator that verifies, collects, and aggregates data from multiple sensors and stores the information in the cloud. A user who wants to check the aggregated data can download the information and perform a single verification. The CA initially registers each sensor and issues certificates when requested. The Helper creates and updates (partial) keys for each sensor. The proposed scheme satisfies the security requirements for signature forgery prevention and non-repudiation of signatures. Aggregate signatures based on pairing-free operations can reduce the size of signatures in storage and reduce the amount of verification computations for validators. In addition, when a sensor device generates a signature key, a key insulation method is applied to periodically update the key used for signatures. This can prevent a security threat that may occur due to the exposure of the sensor key due to a physical attack such as a side-channel attack.

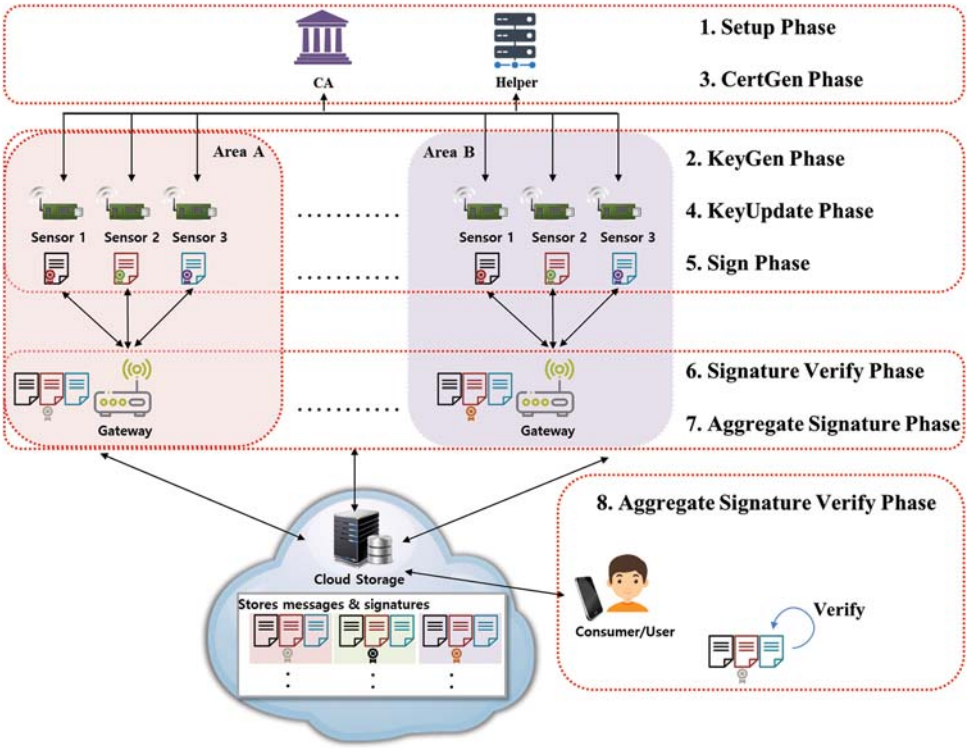

Figure 4: Scenario of the proposed scheme

4.1 System Parameters

The system parameters are:

- ID_i: Identifier of sensor i
- E, P: Elliptic curve on group G with prime order q, generator point of group G
- $Cert_i$: Certificate of sensor i
- MK, P_{CA}: CA's master private key/master public key pair
- DPK_{ID_i}, DSK_{ID_i}: Public key/private key pair of sensor ID_i
- s, z_i: Random values
- hSK_{ID_i}: The helper private key for sensor ID_i
- $udsk_{IDi,t}$: A value used by sensor ID_i to update the signature key
- σ_i: The signature for the message
- $TSK_{ID_i,0}$: The initial temporary signature key generated by the sensor
- $TSK_{ID_i,t}$: The updated signature keys generated by sensor ID_i
- $H_0(\cdot)$: Cryptographic hash function $\left(\{0,1\}^* \times G \times G \rightarrow Z_q^*\right)$
- $H_1(\cdot)$: Cryptographic hash function $\left(\{0,1\}^* \times \{0,1\}^* \times G \times G \rightarrow Z_q^*\right)$
- $H_2(\cdot)$: Cryptographic hash function $\left(\{0,1\}^* \times \{0,1\}^* \times Z_q^* \times G \times G \rightarrow Z_q^*\right)$.

4.2 System Scenario

The eight phases of our scheme include a KeyUpdate key insulation process and the seven phases of the CB-AS listed in Section 2.4.1.

• Setup: The CA receives the security parameters and generates the public parameters, the master secret key, and the CA public key. (Step 1 of Fig. 4)

• KeyGen: Each sensor generates a public/private key pair using the public parameters and random values, and requests a certificate from the CA. (Step 2 of Fig. 4)

• CertGen: The CA generates a certificate $Cert_{\mathrm{ID}_i}$ corresponding to ID_i using ID_i, DPK_{ID_i}, a public parameter, and the master secret key of the CA. The certificate is transmitted to the sensor. The CA creates a random value and sends it to the Helper. (Step 3 of in Fig. 4)

• KeyUpdate: Initially the helper creates a temporary value hSK_{ID_i} for signature generation and passes it to the sensor, and the sensor creates a temporary key $TSK_{ID_i,0}$ that can be used for signing. The Helper creates an updated value $udsk_{IDi,t}$ and sends it to the sensor, which then updates $TSK_{ID_i,0}$ to $TSK_{ID_i,t}$ for signature using the public hash functions and update value received from the Helper. (Step 4 of Fig. 4)

• Sign: The sensor signs the message using the ID_i, DPK_{ID_i}, $Cert_{ID_i}$ and signature key $TSK_{ID_i,t}$ then sends the message and signature to the gateway. (Step 5 of Fig. 4)

• Signature Verify: The gateway verifies the signature of the received messages using $ID_i, DPK_{\mathrm{ID}_i}$ and certificate $Cert_{\mathrm{ID}_i}$ for each sensor device. (Step 6 of Fig. 4)

• Aggregate Signature: In the proposed method, the gateway serves as both a validator and aggregator. Messages and signatures $(m_1, \ldots, m_n, \sigma_1, \ldots, \sigma_n)$ collected from multiple sensor devices

are verified with$(ID_1,\ldots,ID_i,DPK_{\mathrm{ID}_1},\ldots,DPK_{\mathrm{ID}_i},Cert_{\mathrm{ID}_1},\ldots,Cert_{\mathrm{ID}_i})$ then aggregated through the gateway, and the aggregated data is transmitted to cloud storage. After aggregation, n signatures become one signature, reducing signature size. A verifier can check n messages in a single step. (Step 7 of Fig. 4)

- Aggregate Signature Verify: A user who wants to check the aggregated data can verify the integrity of the message collected from each sensor device by retrieving the aggregated data $(m_1,\ldots,m_n,\sigma)$ from the cloud storage as a final verifier and performing verification with $(ID_1,\ldots,ID_i,DPK_{\mathrm{ID}_1},..,DPK_{\mathrm{ID}_i},Cert_{\mathrm{ID}_1},\ldots,Cert_{\mathrm{ID}_i})$. (Step 8 of Fig. 4).

4.3 Description of the Proposed Scheme

4.3.1 Setup Phase

Setup is performed by entering the security parameter K into the CA, and the public key $PK_{CA}=MK\cdot P$ is generated as shown in Eq. (1) after selecting a random value $s\in \mathrm{Z}_{\mathrm{q}}^{*}$. Then, the public parameter PP is created as shown in Eq. (2). Since s is used as the master key value, it is kept secret while $\mathrm{PK}_{\mathrm{CA}}$ and PP are disclosed.

$$MK=s\in Z_q^*,\quad PK_{CA}=s\cdot P \tag{1}$$

$$PP=\{P,q,\mathrm{G},p,PK_{CA},H_0,\ H_1,H_2\} \tag{2}$$

4.3.2 KeyGen Phase

Sensor ID_i selects a random value $x_{ID_i}\in Z_q^*$ and generates a public/private key pair (DPK_{ID_i},DSK_{ID_i}) as shown in Eq. (3). Then, a certificate is requested by sending $\mathrm{ID_i}$and DPK_{ID_i} to the CA.

$$(DPK_{ID_i},\ DSK_{ID_i})=(x_{ID_i}P,x_{ID_i}) \tag{3}$$

4.3.3 CertGen Phase

The CA receives ID_i and DPK_{ID_i} from the sensor and generates the certificate $Cert_{\mathrm{ID}_i}$ corresponding to the sensor device ID_i using the PP and MK as follows.

$$z_i\in Z_q^*,\quad Z_i=z_i\cdot P \tag{4}$$

$$d_{ID}=z_i+sH_0(ID_iPK_{CA},\ Z_i) \tag{5}$$

$$Cert_i=(Z_i,\ d_{ID}) \tag{6}$$

The CA transmits the certificate to the sensor and sends the registered sensor information to the Helper via a secure channel.

4.3.4 Key Update Phase

The KeyUpdate phase is divided into three steps. First, when the Helper sends a temporary value for signing key generation to the sensor device, the device creates a temporary key for signing. The second step is for the Helper to generate and transmit an update value $udsk_{ID_i,t}$ that the sensor uses to update the signature key of the sensor device. Finally, the sensor takes the received update value and updates the signature key, $TSK_{ID_i,\mathrm{t}}$ using it. Eq. (7) shows how the sensor generates temporary key $TSK_{ID_i,0}$ for signing, assuming that it was initially sent $hSK_{ID_i}H_1\left(ID_i,DPK_{ID_i},Z_i,0\right)$ from the Helper to create the signature key.

$$TSK_{ID_i,0}=x_{ID_i}H_1\left(ID_i,DPK_{ID_i},Z_i,0\right)+hSK_{ID_i}H_1\left(ID_i,DPK_{ID_i},Z_i,0\right) \tag{7}$$

Helper key update step: The Helper generates the update value $udsk_{IDi,t}$ employing the user information received from the CA and transmits it to the sensor.

$$hSK_{ID_i} \in Z_q^*, \quad hPK_{ID_i} = hSK_{ID_i} \cdot P \tag{8}$$

$$udsk_{IDi,t} = hSK_{ID_i}(H_1\left(ID_i, DPK_{ID_i}, Z_i, t\right) - H_1\left(ID_i, DPK_{ID_i}, Z_i, t-1\right)) \tag{9}$$

Sensor signature key update: The sensor receives $udsk_{IDi,t}$ from the Helper and can update the existing temporary signature key $TSK_{ID_{i,t-1}}$ to $TSK_{ID_i,t}$ as follows. (The $TSK_{ID_i,t}$ key can be used for time t.)

$$\begin{aligned} TSK_{ID_i,t} &= TSK_{ID_i,t-1} + udsk_{IDi,t-1} + x_{ID_i}\left(H_1\left(ID_i, DPK_{ID_i}, Z_i, t\right) - H_1\left(ID_i, DPK_{ID_i}, Z_i, t-1\right)\right) \\ &= hSK_{ID_i}H_1\left(ID_i, DPK_{ID_i}, Z_i, t\right) + x_{ID_i}H_1\left(ID_i, DPK_{ID_i}, Z_i, t\right) \end{aligned} \tag{10}$$

4.3.5 Sign Phase

The sensor signs message $m_i \in \{0,1\}^*$ using the signature key $TSK_{ID_i,t}$ and its ID_i, DPK_{ID_i}, and $Cert_{\mathrm{ID}_i}$ and sends the result to the gateway.

Step 1. The sensor device calculates $U_i = r_i \cdot P$ by selecting a random value $r_i \in Z_q^*$.

Step 2. The value required for signature is generated as follows and the signature is output.

$$W_i = d_{ID} + TSK_{ID_{i,t}} + r_i H_2(m_i,\ \ t, ID_i\ \ U_i, DPK_{\mathrm{ID}_i}) \tag{11}$$

$$\sigma_i = (U_i, W_i, Z_i) \tag{12}$$

4.3.6 Signature Verify Phase

The gateway receives signed messages from sensors and verifies the signatures using ID_i, DPK_{ID_i}, and $Cert_{\mathrm{ID}_i}$ as shown in Eq. (13). If the criterion is met, the message is regarded as legitimate, and if the criterion is not met, it is not.

$$\begin{aligned} W_i \cdot P = {} & Z_i + H_0\left(ID_i, DPK_{\mathrm{ID}_i}, Z_i\right) PK_{CA} + H_1\left(ID_i, DPK_{\mathrm{ID}_i}, Z_i, t\right) DPK_{\mathrm{ID}_i} \\ & + H_1\left(ID_i, DPK_{\mathrm{ID}_i}, Z_i, t\right) hPK_{\mathrm{ID}_i} + H_2\left(m_i, t, ID_i, U_i, DPK_{\mathrm{ID}_i}\right) U_i \end{aligned} \tag{13}$$

4.3.7 Aggregate Signature Phase

When messages are received from multiple sensors, the gateway performs verification, and then aggregates messages with valid signatures. Signed messages received from n sensors are collected under one signature. Aggregation can be expressed as in Eq. (14), where W is the signature of message $(m_1, \ldots, m_n)$.

$$W = \sum_{i=1}^{n} W_i \tag{14}$$

4.3.8 Aggregate Signature Verify Phase

Users who wish to check aggregated messages download them from the cloud. Then the users verify the validity of the aggregated signature W using $(ID_1, \ldots, ID_i), \left(DPK_{\mathrm{ID}_1}, \ldots, DPK_{\mathrm{ID}_i}\right)$, certificates $(Cert_{\mathrm{ID}_1}, \ldots, Cert_{\mathrm{ID}_i})$ and the aggregated messages $(m_1, \ldots, m_n)$. The user can verify the validity of the signatures of multiple messages with a single signature verification, and

can therefore verify the integrity of the messages collected from all sensor devices. Verification proceeds using Eq. (15).

$$WP = \sum_{i=1}^{n} Z_i + \sum_{i=1}^{n} H_0\left(ID_i, DPK_{\mathrm{ID}_i}, Z_i\right) PK_{CA} + H_1\left(ID_i, DPK_{\mathrm{ID}_i}, Z_i, t\right) DPK_{\mathrm{ID}_i} + H_1\left(ID_i, DPK_{\mathrm{ID}_i}, Z_i, t\right) hPK_{\mathrm{ID}_i} + H_2\left(m_i, t, ID_i, U_i, DPK_{\mathrm{ID}_i}\right) U_i \quad (15)$$

5 Analysis of Proposed Scheme

In this section, we will identify whether the proposed method meets the data integrity, forgery resistance, and key leakage resilience requirements from Section 3.

5.1 Security Analysis

• **Data Integrity and Reliability:** Verification is essential to ensure message integrity, reliability, and non-repudiation. In this paper, the signature $\sigma_i = (U_i, W_i, Z_i)$ generated by each sensor device uses a certificate value, the device's private key, and a public key. Verifying $W_i \cdot P$ includes the operation $H_2\left(m_i, t, ID_i, U_i, DPK_{\mathrm{ID}_i}\right)$, which verifies message integrity. In addition, the verifier can know that the data has been sent from sensor ID_i by verifying σ_i using the public key of the CA, the public key of the sensor device, and the public key of the Helper, as well as public parameters and the message. Thus, it can be seen that the message has not been modified during transmission.

• **Unforgeability:** In a CB-PKC signature protocol, an attack on unforgeability can occur. The types of attacks can be divided into an attack by an attacker A_1, capable of public key replacement, and an attack by an attacker A_2, a malicious CA. A_1 has the ability to replace the public keys of other users with their own generated keys. Due to the safety of ECDLP, private keys corresponding to users' public keys cannot be determined, but only public keys need be replaced to bypass validation. Public key replacement attacks often occur in certificateless methods where there is no certificate that can authenticate the public key of a sensor device, e.g., the signer. However, even if there are certificates, attacker A_1can perform a public key replacement attack by creating a new private key/public key pair like the attack model in Section 2.4.2. This is very difficult because it has to be performed within the polynomial time, and because the public key is usually authenticated by a CA. However, it was analyzed that Verma et al. scheme 2 is vulnerable to a public key exchange attack by Xiong et al. This attack modifies the operation of the signature value regardless of the issued certificate, so that the signature can be forged, and the verifier can still process it as valid. In our proposed method, the form of individual signature is $\sigma_i = (U_i, W_i, Z_i)$, and the signature $\sigma_i' = (U_i', W_i', Z_i')$ that could create a valid message cannot be generated using the forged public key $DPK'_{ID_i} = \left(x'_{ID_i} P\right)$ by attacker A_1. The signature generation formula for the message is $W_i = d_{ID} + TSK_{ID_i,t} + r_i H_2(m_i, t, ID_i, U_i, DPK_{\mathrm{ID}_i})$, and the attacker A_1 cannot create a legitimate W_i because the $TSK_{ID_i,t}$ are unknown. The signature verification formula (Eq. (13)) is calculated using the public key of CA, the public key of the sensor device, the public key of the Helper, other public parameters, and the message. It is impossible for attacker A_1 to forge the signature by replacing it with the public key of another device without CA assistance.

Attacker A_2 is a malicious CA and has the ability to know all the certificates of registered sensor devices because it knows MK. Because A_2 does not have the ability to replace public keys, if A_2 wants to forge a signature of a sensor device, it will try to generate a signature with the value of the sensor's certificate. The signature generation formula for a message is

$W_i = d_{ID} + TSK_{ID_{i,t}} + r_i H_2(m_i, t, ID_i, U_i, DPK_{\mathrm{ID}_i})$. Since the signature key $TSK_{ID_{i,t}}$ generated by the sensor device is used to generate the signature, the CA cannot forge the signature with only Z_i, and d_{ID}, and cannot extract $TSK_{ID_{i,t}}$. The signer's signature cannot be forged using only external public parameters, including in the CA A_2 scenario. In particular, in the proposed scheme, the signature key used must include $udsk_{IDi,t}$, created through the Helper, the previous temporary key value $TSK_{ID_i,0}$ of the sensor device, and the certificate. Therefore, a malicious CA cannot forge the sensor device's signature using the MK.

• **Key Exposure Resilience:** In our proposed method, the sensor devices initially register with the CA, receive a certificate, and receive $hSK_{ID_i} H_1\left(ID_i, DPK_{ID_i}, Z_i, 0\right)$ from the Helper in order to generate keys. Then the devices create temporary keys $TSK_{ID_i,0}$ for signing. Next, $udsk_{IDi,t}$ is received from the Hepler in the KeyUpdate Phase, and a key $TSK_{ID_i,t}$, that can be used to sign for a period of time t, is generated and used when signing. $TSK_{ID_i,t}$ includes the sensor device identifier ID_i, the device public key/private key pair (DPK_{ID_i}, DSK_{ID_i}), the Helper's private key hSK_{ID_i}, and Z_i from the certificate. At this point, even if $TSK_{ID_i,t}$ is exposed by a physical attack, each sensor device can update its signature key through the Helper and use the new key for signing. Later, when a verifier checks the signature, the $TSK_{ID_i,t}$ is verified with the public key DPK_{ID_i} of the sensor device and the public key of the Helper. Since the period t in which the signature can be used is included as a parameter, when an attacker who obtained the previous $TSK_{ID_i,t}$ signs with it the verifier will be unable to verify. Therefore, it is impossible to perform various attacks such as signature forgery or masquerade attacks by obtaining the signature key $TSK_{ID_i,t}$ through physical attack.

5.2 Efficiency

An IoT environment requires efficient computation. In a network environment in which many sensor devices participate in communications, the system is required to work without problems using low-performance devices with either Elliptic Curve Cryptography (ECC) or paring-free based lightweight cryptographic operations rather than heavy cryptographic operations using on Rivest–Shamir–Adleman (RSA) and pairing operations. In the aggregation process, as the number of messages and signatures increases, the total time for signature, aggregation, and verification increases in direct proportion. In this proposed method, pairing operations are not used. Compared to other pairing-free methods, ECC-based elliptic curve addition and multiplication operations are efficiently applied to reduce the total operation time, and the amount of calculation is more efficient than the pairing operations applied in the existing Xiong et al. [31] scheme. In addition, compared to the Chen et al. and Reddy et al. methods, which do not use pairing operations, as shown in Tab. 2, the computational efficiency is high, and the computational amount is similar to the Verma et al. methods. For the simulation environment of this proposed method and the existing methods, Windows 10 was used, running on an Intel i5-4690 processor with 3.50 GHz clock rate and 8 GB of memory. The ECC implementation used a Koblitz elliptic curve $y^2 = x^3 + ax + b\,(modp)$ with a = 1 and b = 1, with a 163-bit random prime defined in $F_{2^{163}}$. The execution time T_{sm} of the scalar multiplication operation in ECC was 0.431 ms, and the addition operation T_{sa} between points in ECC was 0.017 ms. In addition, the hash operation execution time H was 0.079 ms, the scalar exponential operation execution time T_{se} was 4.416 ms, and finally, the pairing operation p was 20.225 ms. Tab. 2 compares the efficiency of other recently proposed CB-AS methods with the proposed method, and Fig. 5 is a graph of the expected execution times. As shown in Fig. 5, this proposed method has a higher computational efficiency than Verma et al. schemes 1 and 2, which do not use pairing operations but also don't provide key insulation. In addition, compared to the Xiong et al. scheme, which is an aggregate signature

method that provides key insulation, the amount of computation is reduced, thereby increasing the efficiency of the overall computational amount in the proposed scheme.

Table 2: Security and efficiency analysis of existing schemes and proposed scheme

Scheme items	Chen et al. [24] scheme	Verma et al. [26] scheme 2	Verma et al. [25] scheme 1	Xiong et al. [31] scheme	Proposed scheme
Type	CB – AS	CB – AS	CB – AS	CB – KIAS	CB – KIAS
Key exposure problem due to side channel attack	Vulnerable when key is exposed			Safe even if keys are exposed (key update due to key insulation)	
Forgery with public key replacement (A_1)	Safe (the signature contains a hashed random value. The attacker cannot know about the hash value contained in the signature.)		Not safe (signature forgery is possible by attack type I)	Safe (since the random value generated by the signer is included in the signature value, the attacker cannot know the signature value)	Safe (5.1 refer to the unforgeability part of the security analysis)
Forgery with CA master key (A_2)			Not safe (signature forgery is possible by attack type II)		
Key update operation	Not provided	Not provided	Not provided	$2T_{sm}+2T_{sa}$	T_{sm}
Sign operation	$3T_{sa}+2T_{sm}+h$	T_{sm}	T_{sm}	$5T_{sm}+2T_{sa}$	T_{sm}‘
Verify operation	$2T_{sa}+2T_{sm}$	$4T_{sm}$	$3T_{sm}$	$3p$	$4T_{sm}$
Aggregate operation	nT_{sa}	nT_{sa}	nT_{sa}	np	nT_{sa}
Aggregate verify operation	$2nT_{sa}+(2n+1)\,T_{sm}+nh$	$(2n+2)\,T_{sm}$	$(n+2)\,T_{sm}$	$(n+3)p+(2n+4)\,T_{sa}$	$(2n+2)\,T_{sm}$
Total operations	$(3n+5)\,T_{sa}+(2n+5)\,T_{sm}+(n+1)\,h$	$(2n+7)\,T_{sm}+nT_{sa}$	$(n+6)\,T_{sm}+nT_{sa}$	$(2n+6)p+7T_{sm}+(2n+8)\,T_{sa}$	$(2n+8)\,T_{sm}+nT_{sa}$

Notes: p: *Pairing operation*; T_{sm}: *Multiplication operation*; T_{sa}: *Addition operation*; n: *Number of signatures*; T_{se}: *Exponentiation operation*; H: *Hash function*.

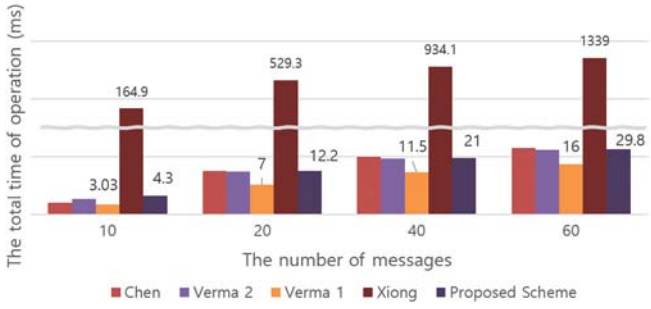

Figure 5: Comparison of total computation time between the proposed scheme and existing schemes

6 Conclusion

The proposed scheme is a certificate-based aggregate signature scheme that provides key insulation for secure and efficient data collection and provision in IoT-cloud environments. As a feature of this proposed scheme, signatures cannot be forged by an attacker, and the signature keys generated in the sensor devices are continuously updated by applying a key insulation technique to solve the key exposure problem caused by physical attacks such as side-channels attacks. In addition, the aggregate signature function allows the signatures of messages received from multiple sensor devices to be aggregated into one signature, thereby reducing the total size of the signature. Additionally, the final verifier can verify the signature of multiple messages with a single verification. This provides integrity and reliability for messages, and can be used in environments such as IIoT. Finally, since it is performed using pairing-free operations in the signature generation, aggregation, and verification phases, the operation efficiency is high compared to CB-AS methods proposed using pairings (Tab. 2).

For future research, it is necessary to study signcryption that can add data encryption for the confidentiality of data transmitted in the IoT environment. In particular, research on weight reduction to reduce the amount of computations performed in each phase for efficient communication between IoT devices should also be conducted.

Funding Statement: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1085718) and was supported by the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, 2019.

[2] J. W. Jang, S. Kwon, S. Kim, J. Seo, J. Oh *et al.,* "Cybersecurity framework for IIoT-based power system connected to microgrid," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2221–2235, 2020.

[3] M. M.Sha and M. P. Rahamathulla, "Cloud-based healthcare data management framework," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 3, pp. 1014–1025, 2020.

[4] A. R. Sadeghi, C. Wachsmann and M. Waidner, "Security and privacy challenges in industrial internet of things," in *2015 52nd ACM/EDAC/IEEE Design Automation Conf.*, San Francisco, CA, USA, pp. 1–6, 2015.

[5] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[6] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*, Berlin, Heidelberg: Springer, pp. 47–53, 1984.

[7] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Int. Conf. on the Theory and Application of Cryptology and Information Security*, Berlin, Heidelberg: Springer, pp. 452–473, 2003.

[8] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[9] D. Boneh, C. Gentry, B. Lynn and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg: Springer, pp. 416–432, 2003.

[10] K. A. Shim, "An ID-based aggregate signature scheme with constant pairing computations," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1873–1880, 2010.

[11] L. Shen, J. Ma, X. Liu, F. Wei and M. Miao, "A secure and efficient id-based aggregate signature scheme for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 546–554, 2016.

[12] X. Yang, R. Liu, M. Wang and G. Chen, "Identity-based aggregate signature scheme in vehicle ad-hoc network," in *2019 4th Int. Conf. on Mechanical, Control and Computer Engineering*, IEEE, pp. 1046–10463, 2019.

[13] F. Hess, "Efficient identity-based signature schemes based on pairings," in *Int. Workshop on Selected Areas in Cryptography*, pp. 310–324, 2002.

[14] H. Du and Q. Wen, "An efficient identity-based short signature scheme from bilinear pairings," in *Int. Conf. on Computational Intelligence and Security*, pp. 725–729, 2007.

[15] I. Ali, T. Lawrence and F. Li, "An efficient identity-based signature scheme without bilinear pairing for vehicle-to-vehicle communication in VANETs," *Journal of Systems Architecture*, vol. 103, no. 8, pp. 101692, 2020.

[16] P. Kumar, S. Kumari, V. Sharma, A. K. Sangaiah, J. Wei *et al.,* "A certificateless aggregate signature scheme for healthcare wireless sensor network, sustainable computing," *Informatics and Systems*, vol. 18, pp. 80–89, 2018.

[17] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," *Computer Communications*, vol. 32, no. 6, pp. 1079–1085, 2009.

[18] L. Deng, Y. Yang, Y. Chen and X. Wang, "Aggregate signature without pairing from certificateless cryptography," *Journal of Internet Technology*, vol. 19, no. 5, pp. 1479–1486, 2018.

[19] J. Li, X. Huang, Y. Mu, W. Susilo and Q. Wu, "Certificate-based signature: Security model and efficient construction," in *European Public Key Infrastructure Workshop*, Berlin, Heidelberg: Springer, pp. 110–125, 2007.

[20] J. Li, X. Huang, Y. Zhang and L. Xu, "An efficient short certificate-based signature scheme," *Journal of Systems & Software*, vol. 85, no. 2, pp. 314–322, 2012.

[21] J. Zhang, "On the security of a certificate-based signature scheme and its improvement with pairings," in *Int. Conf. on Information Security Practice and Experience*, Berlin, Heidelberg: Springer, pp. 47–58, 2009.

[22] H. Xiong, S. Wu, J. Geng, E. Ahene, S. Wu *et al.,* "A pairing-free key-insulated certificate-based signature scheme with provable security," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 3, pp. 1246–1259, 2015.

[23] X. Ma, J. Shao, C. Zuo and R. Meng, "Efficient certificate-based signature and its aggregation," in *Int. Conf. on Information Security Practice and Experience*, Cham: Springer, pp. 391–408, 2017.

[24] J. N. Chen, Q. S. Chen and F. M. Zou, "Certificate-based aggregate signature scheme without bilinear pairings," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 6, pp. 1330–1336, 2016.

[25] G. K. Verma, B. B. Singh, N. Kumar and V. Chamola, "CB-CAS: Certificate-based efficient signature scheme with compact aggregation for industrial internet of things environment," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2563–2572, 2019.

[26] G. K. Verma, B. B. Singh, N. Kumar and O. Kaiwartya, "PFCBAS: Pairing free and provable certificate-based aggregate signature scheme for the e-healthcare monitoring system," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1704–1715, 2019.

[27] Y. Dodis, J. Katz, S. Xu and M. Yung, "Key-insulated public key cryptosystems," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg: Springer, pp. 65–82, 2002.

[28] G. Hanaoka, Y. Hanaoka and H. Imai, "Parallel key insulated public key encryption," in *Int. Workshop on Public Key Cryptography*, Berlin, Heidelberg: Springer, pp. 105–122, 2006.

[29] P. V. Reddy and P. V. S. S. N. Gopal, "Identity-based key-insulated aggregate signature scheme," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 3, pp. 303–310, 2017.

[30] Y. Lu, J. Li and J. Shen, "Weakness and improvement of a certificate-based key-insulated signature in the standard model," *The Computer Journal*, vol. 60, no. 12, pp. 1729–1744, 2017.

[31] H. Xiong, Y. Hou, X. Huang and S. Kumari, "Certificate-based parallel key-insulated aggregate signature against fully chosen-key attacks for industrial internet of things," *IEEE Internet of Thing Journal*, Early Access, pp. 1–14, 2020.

[32] J. Li, H. Du and Y. Zhang, "Certificate-based key-insulated signature in the standard model," *The Computer Journal*, vol. 59, no. 7, pp. 1028–1039, 2016.