

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/149001/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Verma, Girraj, Gope, Prosanta, Saxena, Neetesh and Kumar, Neeraj 2023. CB-DA: Lightweight and escrow-free certificate-based data aggregation for smart grid. IEEE Transactions on Dependable and Secure Computing 20 (3) , pp. 2011-2024. 10.1109/TDSC.2022.3169952

Publishers page: <https://doi.org/10.1109/TDSC.2022.3169952>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



# CB-DA: Lightweight and Escrow-Free Certificate-Based Data Aggregation for Smart Grid

Girraj Kumar Verma, Prosanta Gope, *Senior Member, IEEE*, Neetesh Saxena, *Senior Member, IEEE*, and Neeraj Kumar, *Senior Member, IEEE*

**Abstract**—Recent development of smart cities includes advanced and optimized use of modern smart grid (SG) than the traditional power grid. The paradigm of SG has also transformed houses into a home area networks, where several smart devices and appliances are connected to the electricity control centers (ECC). Appliances share their load and consumption related information to ECC through smart meters. This consumption data may be used for supply-demand management, for example, by ramping production up or down as needed. However, security and privacy of the consumer's data are greatly important since fine-grained smart meter data may reveal users presence/absence in their house. To address this issue, several public-key-based or identity-based data aggregation schemes have been proposed in the literature. However, most of such schemes either suffer from the complexity of certificate management or key escrow problem. To eliminate these issues, in this paper, we propose an efficient certificate-based data aggregation (CB-DA) scheme. In the proposed CB-DA scheme, the owner selects a secret key and then use the secret key along with certificates as decryption/signing keys. Our analysis shows that the proposed CB-DA is secure under the random oracle model and more efficient than the existing data aggregation schemes.

**Index Terms**—Data Aggregation, Smart Grid, Certificate-Based Encryption, Identity-Based Cryptography.

## I. INTRODUCTION

The intelligent electricity distribution framework (IEDF) is a key requirement of smart grid infrastructure (SGI). In SGI, IEDF is defined as the framework of electric vehicles (EV), charging stations, houses (or buildings), base stations (BS), power grids and electricity control center (ECC) connected through internet [1]. The houses equipped with smart meters (SMs) create a home area networks (HAN). In HAN, smart appliances (SA) share their consumption and load data to ECC through SM/BS as shown in Figure 1. Therefore, all entities in IEDF communicate to each other using various modern communication technologies. For example, within HAN, communication links between different entities (SA-to-SA or SA-to-SM) are short distance ZigBee or Bluetooth links.

G. K. Verma is with the Department of Mathematics, Amity School of Engineering and Technology, Amity University Madhya Pradesh, Gwalior, India (Email: gkverma@gwa.amity.edu, girrajv@gmail.com)

P. Gope is with the Department of Computer Science, University of Sheffield, Sheffield S1 4DP, United Kingdom (Email: p.gope@sheffield.ac.uk).

N. Saxena is with the School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom (Email: nsaxena@ieee.org).

N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala, India. N. Kumar is also associated with Department of Computer Science and Information Engineering, Asia University, Taiwan, Department of Computing and Informatics, King Abdul Aziz University, Saudi Arabia, and School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand (Email: neeraj.kumar@thapar.edu).

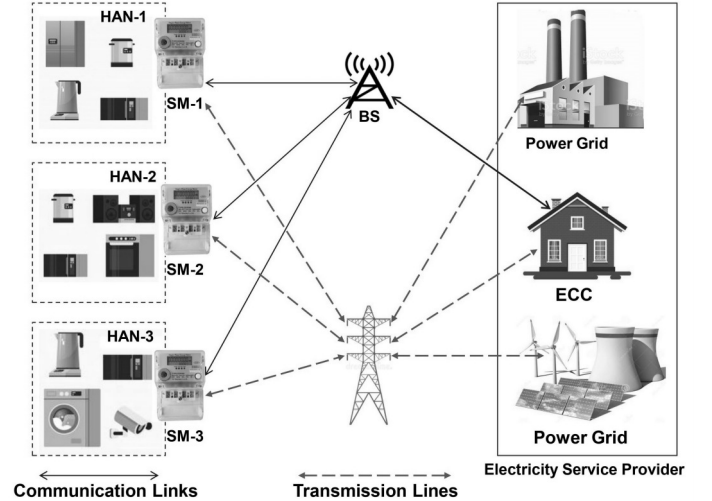


Fig. 1. Intelligent electricity distribution framework under advanced metering infrastructure of the smart grid.

Links between SM-to-BS and BS-to-ECC may be inexpensive wireless gateway and optical fiber respectively [8]. In this framework, based on the shared load or consumption records, ECC can compute the total load or future demand and supply, so that the availability of electricity can be confirmed [5].

As per the survey in Europe by the end of the year 2020, 72% of home area networks (HAN) will be equipped with SMs [16]. However, most of the communication links are wireless and hence an unauthorized entity (i.e. an adversary) can easily target the information sent by SM [11]. Using this information, an adversary can observe and track the pattern related to the sensitive information (such as family lifestyle) of the consumer. Based on this personal data, the adversary may get inspired to commit a crime. Besides, the real time data sent by SM is utilized by ECC to predict the future balance between demand and supply. Therefore, it may be that adversary sends modified/false information to ECC on behalf of SM. Due to this false data, smart grid system may suffer with unbalancing between demand and supply and this may result as blackout. Thus, it is a high priority to protect the communication links from such attackers [12]. On the other hand, in SGI system, data from several SMs is processed by ECC. Therefore, efficient utilization of ECC resources must also be guaranteed.

To ensure this efficient utilization alongwith security, the cryptographic data aggregation (DA) (i.e. aggregation of sig-

natures as well as encrypted text) technique is a suitable choice [4], [5], [12], [13], [14], [15], [18], [19]. Generally, in a DA protocol, HAN (equipped with SM) is comprised of a finite number of SAs. These SAs send their encrypted information (encrypted and signed by SM) to a designated SGI device/BS that is working as an aggregator (AG) [17]. AG first checks the validity of the signature sent by SM and then computes the compressed encrypted value. After it, AG creates a signature on the aggregated ciphertext. This signed ciphertext is sent to ECC. At last, ECC confirms the validity of AG/BS's signature and if it is correct, it computes the total consumption of energy units by decryption [8], [11].

#### A. Related work

In literature, to secure the SGI communication in an efficient manner, Fan *et al.* [2] devised the first privacy-preserving DA (PP-DA) protocol secure scheme against internal adversaries. However, Bao *et al.* [10] cryptanalysed this scheme successfully and suggested some guidelines to improve the security of [2]. In [4], Shiobara *et al.* devised an effective DA scheme to protect SGI. In this, authors used concentrators to reduce the message size by 98.5%. Besides [10], He *et al.* [5] also reported a successful key leakage attack on [2]. The authors also proposed an improved version of the PP-DA scheme, more efficient than [2]. In [3], Li *et al.* proposed a forward secret and efficient privacy-preserving demand response scheme (DRS) by the amalgamation of Homomorphic Encryption (HE) and a key evolution technique. Later, to improve DRS, Wang, Mu and Chen [6] devised a PP-DA protocol supporting the billing process by combining HE and verifiable secret sharing. In [7], Jo *et al.* also devised a robust privacy-preserving metering protocol to deal with the adversaries against compromised node attack. Later, as a post-quantum candidate for SGI security, Abdallah and Shen [8] devised a lattice-based PP-DA scheme by using a post-quantum HE from [9]. However, in this PP-DA scheme, no impact of the electric vehicle has been considered. To reduce the burden of certificate management on SGI, Z. Wang [11] devised an identity-based PP-DA scheme. The author implemented his scheme on the Intel Edison platform to realize efficiency. In 2017, He *et al.* [12] devised a PP-DA scheme secure against internal attackers, which consumes less computational cost than existing PP-DA protocols. However, the key size can further be reduced by taking elliptic curves as a base structure. To improve the recovery process of aggregated consumption, Li *et al.* [13] proposed a multi-subset PP-DA protocol by considering the aggregation of customer's subsets having different consumption ranges. To lower the key size, Vahedi *et al.* [14] devised an elliptic curve-based PP-DA scheme for SGI, which enjoys less computational cost alongwith short key. In [15], Gope and Sikadar devised a symmetric key cryptography-based efficient PP-DA scheme for SGI. It is the first scheme using dynamic pricing-based demand response management technique. As an improved post-quantum mechanism, Chaudhary *et al.* [16] designed a lattice theory-based crypto-system using software-defined networking technology. By using a modified Paillier cryptosystem [26] and Fog-enabled edge computing, Saleem *et*

TABLE I  
VARIOUS SYMBOLS USED IN THE PAPER

Notation	Description
$1^\gamma$	Security parameter
$q$	Prime number
$\mathcal{G}, \mathcal{G}_T$	Two cyclic $q$ order groups
$g$	Generator of $\mathcal{G}$
$g_t$	Generator of $\mathcal{G}_T$
$pk_{ca}$	CA's public key
$x_{ca}$	CA's secret key
$SA_i$	$i^{th}$ smart appliance in HAN
$SA_{AG}$	Aggregation computing edge device
$BS_i$	$i^{th}$ base station
$SM_i$	Smart meter deployed in $i^{th}$ HAN
$pk_{sa_i}, sk_{sa_i}$ and $Cert_{sa_i}$	Public key, secret key and certificate of $SA_i$
$pk_{sm_i}, sk_{sm_i}$ and $Cert_{sm_i}$	Public key, secret key and certificate of $SM_i$
$pk_{bs_i}, sk_{bs_i}$ and $Cert_{bs_i}$	Public key, secret key and certificate of $BS_i$
$pk_{ecc}, sk_{ecc}$ and $Cert_{ecc}$	Public key, secret key and certificate of ECC
$H_c : \{0, 1\}^* \times \mathcal{G} \rightarrow \mathcal{G}$	Cryptographic hash
$H_s : \mathcal{G} \times \mathcal{G} \times \{0, 1\}^* \rightarrow \mathcal{G}$	Cryptographic hash
$(C_{1_i}, C_{2_i})$	Ciphertext from $SA_i$ of $j^{th}$ HAN
$(C_{1_j}, C_{2_j})$	Aggregated ciphertext from $j^{th}$ HAN

*al.* [18] designed a secure PP-DA protocol for SGI. Recently, to make the joining and leaving of the smart meter in an SGI more flexible, Chen *et al.* [19] devised a scalable PP-DA scheme. In parallel, Xue *et al.* [20] also designed a robust and efficient PP-DA scheme. Their scheme supports dynamic user management. In the literature, to devise more secure PP-DA schemes the concept of masking has also been used by several authors [21], [22], [23], [24]. However, most of the proposed solutions in the literature suffer from several security issues such as lack of privacy protection, sender's authentication, collusion attacks, etc. [25].

As discussed, majority of the existing PP-DA schemes in the literature are either based on the public key infrastructure (PKI) (such as [2], [5], [6], [7], [8], [12], [13], [14]) or identity-based infrastructure (IBI) (such as [3], [11]). However, PKI or IBI based constructions have the following drawbacks:

- 1) Due to PKI-based construction, the schemes presented in [2], [5], [6], [7], [8], [12], [13], [14] suffer with certificate management overhead. Each smart meter needs verification of ECC's certificate before encryption of consumption data. This verification causes an additional load on the device, which is not suitable for a resource-constrained environment like SGI.
- 2) In IBI the secret key is chosen by a trusted authority like CA. Therefore, in SGI environment, CA knows the secret key of each SGI entity (like SM, BS or ECC). Thus, CA can create a forged signature or can get user's consumption or related information; this serious issue is known as the key-escrow problem.
- 3) Besides key escrow, for sharing secret keys a secure channel is also needed.

To overcome such issues, C. Gentry [27] proposed a new paradigm of certificate-based (CB) encryption. This paradigm integrates the merits of PKI and IBI by removing the certificate management (or Certificate Revocation) and key escrow. The important features of CB-infrastructure are:

- 1) In the CB scheme, like PKI the secret key is chosen by

TABLE II  
COMPARATIVE ANALYSIS OF RELATED SCHEMES

Scheme	Primitives Used	SP1	SP2
Fan <i>et al.</i> [2]	Bilinear Pairing, PKI	Yes	Yes
Li <i>et al.</i> [3]	Bilinear Pairing, IBI	No	No
He <i>et al.</i> [5]	Bilinear Pairing, PKI	Yes	Yes
Wang <i>et al.</i> [6]	Bilinear Pairing, PKI	Yes	Yes
Jo <i>et al.</i> [7]	Bilinear Pairing, PKI	Yes	Yes
Abdallah <i>et al.</i> [8]	Lattice Based PKI	Yes	Yes
Wang <i>et al.</i> [11]	Bilinear Pairing, IBI	No	No
He <i>et al.</i> [12]	DLP based PKI	Yes	Yes
Li <i>et al.</i> [13]	DLP based PKI	Yes	Yes
Vahedi <i>et al.</i> [14]	Bilinear Pairing, PKI	Yes	Yes
Proposed Scheme	Bilinear Pairing, CBE	No	Yes

SP1: Certificate Management Required, SP2: Robust Against Key Escrow

the user/owner (in place of CA of IBI).

- 2) CA is responsible to issue certification of (ID, public key) only and it is unable to know secret key of any user/owner.
- 3) The certificate (issued by CA) is used along with the secret key by the owner for decryption or signing. So, only the user/owner knows the secret key and needs its certificate. Any entity other than user/owner, needs only the (ID, Public Key) of user/owner for encryption/signature verification.

Therefore, all the issues (certificate management, key escrow and need of secure channel for secret key sharing) have been resolved by the CB scheme.

### B. Motivation and Contribution

As discussed, the key escrow and burden of certificate management are still regarded as the serious issues for SGI (Table II). Previously, to resolve these issues, various schemes on CB infrastructure [28], [29], [30], [31] have been proposed in the literature. However, these constructions are not adequate to get PP-DA for SGI communication. Therefore, this paper seeks to address all such serious issues by proposing a *new* and lightweight CB-DA scheme for SGI communication (HAN-to-ECC). The implementation of cryptographic operations shows that the proposed CB-DA scheme can provide an efficient and escrow free data aggregation environment for SGI. The major contributions of this paper are as follows: are:

- This paper proposes the *first* certificate-based homomorphic encryption (CB-HE) from pairing. Because of the homomorphic property, the proposed scheme can support the aggregation of data from various HANs. Therefore, along with efficiency improvement it also enhances the security level of the system by removing key escrow. It also ensure the efficient utilization of SM and BS by removing the burden of certificate management on these resource constrained devices in SGI [27].
- To ensure key escrow free signing, a lightweight CB signature scheme is also proposed, which can mitigate the possibility of forgery in signatures (in place of SG devices) by CA [28].
- To show enhanced security level, a detailed formal security analysis is presented by considering two stronger

adversaries (malicious CA and malicious user) in ROM. During informal security analysis, it has been observed that proposed CB-DA is robust against both internal (such as BS and ECC) and external attackers.

- To demonstrate the performance in practical SGI environment, a realistic implementation is done on two devices as emulator. During performance comparison, it has been observed that the proposed CB-DA scheme is more efficient than competitive schemes [2], [11], [14] and [19].

The article is organized as follows: Section-II describes basics and definitions on preliminaries such as groups, system model, the framework of CB-DA, and security definitions. Section-III presents the proposed CB-DA scheme. Section-IV discusses security analysis, Section-V presents the performance discussion and Section-VI concludes the paper.

## II. PRELIMINARIES

This section elaborates the required mathematical background, system model and framework of the proposed CB-DA, security goals and an adversarial model to achieve the goals.

### A. Mathematical Preliminaries

Suppose  $\mathcal{G}$  and  $\mathcal{G}_T$  be two cyclic prime ( $q$ ) order groups. A bilinear map  $e : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}_T$  having following attributes has been defined.

- 1) Bilinear:  $\forall g_1, g_2 \in \mathcal{G}$  and  $\forall \alpha, \beta \in \mathbb{Z}_q^*$ , we have  $e(g_1^\alpha, g_2^\beta) = e(g_1, g_2)^{\alpha\beta} = e(g_1, g_2)^{\alpha\beta}$ .
- 2) Nondegenerate:  $\exists g \in \mathcal{G}$ , we have  $e(g, g) \neq 1_{\mathcal{G}_T}$ .
- 3) Computable:  $\forall g_1, g_2 \in \mathcal{G}$  it is easy to compute  $e(g_1, g_2)$ .

The following assumptions will be the base for the security of the proposed CB-DA scheme.

**Definition-1 Computational Diffie-Hellman (CDH) Assumption:** The CDH assumption states that for a given instance  $(g, g^\alpha, g^\beta)$ ,  $\forall \alpha, \beta \in \mathbb{Z}_q^*$  and generator  $\forall g \in \mathcal{G}$  adversary  $\mathcal{A}$  computes the output  $g^{\alpha\beta}$  with negligible probability  $\epsilon(\zeta)$  in time  $(\leq \tau)$ .

**Definition-2 Decisional Bilinear Diffie-Hellman (DBDH) Assumption:** The DBDH assumption states that for a given  $(g, g^\alpha, g^\beta, g^\gamma)$ ,  $\forall \alpha, \beta, \gamma \in \mathbb{Z}_q^*$  and generator  $\forall g \in \mathcal{G}$  adversary  $\mathcal{A}$  computes the output  $b \in \{0, 1\}$  with negligible probability  $|Pr[\mathcal{A}(g, e, g^\alpha, g^\beta, g^\gamma, h_b) = b] - 1/2|$  in time  $(\leq \tau)$  where  $h_0 = e(g, g)^{\alpha\beta\gamma}$  and  $h_1 = e(g, g)^c$  for  $c \in \mathbb{Z}_q^*$ .

### B. System Model

Our system model for the proposed CB-DA scheme is shown in Figure 2, which consists of six major entities: Smart Appliances (SAs), Aggregators ( $SA_{AG}$ ), Smart Meters (SMs), Base Stations (BSs), Electricity Control Center (ECC) and an off-line Certification Authority (CA). In our system model, SAs,  $SA_{AG}$ , SMs, BSs and ECC generate their key pairs and get certification on ID and public key from CA. According to this model, in a HAN, SAs send their encrypted data (readings) to  $SA_{AG}$ , which is basically an edge computing device (may be a smart tab/phone/wireless gateway). Then,  $SA_{AG}$  sends the aggregated encrypted data to SM. After that,

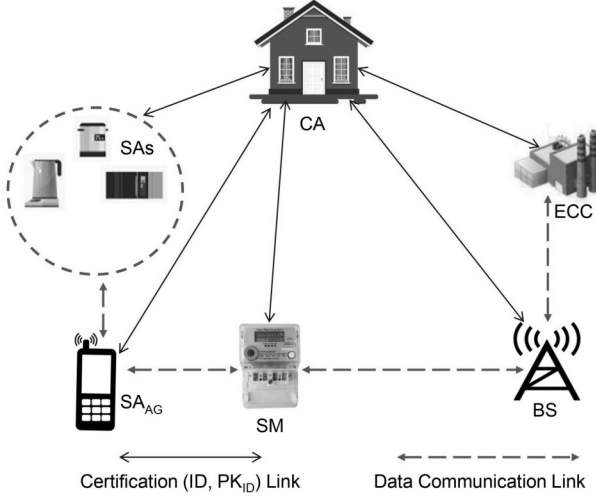


Fig. 2. System model and communications among entities.

the SM forwards this data to the nearest BS after signing on it. BS gets the data from several SMs, which are deployed in different HANs. BS then checks the validity of the signatures of all SMs and further aggregates the received ciphertext. Thereafter, BS sends the aggregated ciphertext to ECC after signing. In the end, ECC verifies the signature of the BS and further decrypts the received aggregated text. In this way, ECC gets the consumption data of from various HANs. It should be noted that, in the previous schemes [2], [11], [14] and [19] the aggregation, encryption and signing of entire local data is carried out at the SM. Within HAN, short range protocols like IEEE 802.15.4 or IEEE 802.15.1 may be applied. For other communication (SM-to-BS and BS-to-ECC), IEEE 802.11e, Fiber Optic, PLC, IEEE 802.11n, IEEE 802.16, etc. may be used.

### C. Framework of the Proposed CB-DA Scheme

The proposed CB-DA scheme is divided into five phases. The detailed description of the phases is given below.

- 1) Initialization: In this phase, CA generates system parameters and master secret key using the security parameter. Further, all entities SAs,  $SA_{AG}$ , SMs, BSs and ECC create their key pairs by using system parameters.
- 2) Certification: In this phase, all SAs,  $SA_{AG}$ , SMs, BSs and ECC send their IDs and public keys to CA. Further, CA issues certificate for (ID, public key) pair.
- 3) Data-Aggregate ( $SA_{AG}$ ): In this phase, SAs send encrypted consumption value to  $SA_{AG}$ , which aggregates and then signed it. SM receives this aggregated data and verifies the signature of  $SA_{AG}$  and then creates a new signature on it. This data with new signature is sent to BS.
- 4) Data-Aggregate (BS): During the phase, BS verifies the signatures made by SMs on received data from various HANs. Then, BS aggregates the received data and sends to ECC after signing.
- 5) Data-Decryption (ECC): The output is the plain text of aggregated ciphertexts. In this phase, ECC verifies

the signatures made by BSs on received data. After it, ECC does the decryption of the data and gets total consumption data.

### D. Security Goals

The following security goals are considered in the proposed CB-DA scheme.

- 1) Authentication: The authentication of source and data are important security goals to be considered.
- 2) Confidentiality and Integrity: The integrity and confidentiality of the data of the consumers are the next vital issues to be considered.
- 3) Privacy: The user's information privacy is to be considered in designing of the proposed CB-DA scheme.

By informal security analysis in Section IV-C, a detailed discussion to achieve these goals is presented. For the analysis, attacks such as Man-in-the-Middle (MITM) attacks and Internal attacks have been considered. In MITM, an active attacker uses three attacks namely *Impersonation Attack*, *Replay Attack* and *Modification Attack*. In *Internal Attack* an internal entity (such as BS or ECC) tries to get the individual user's data (i.e. from a single SM).

### E. Adversary and Security Model

Now, for the security of our proposed CB-DA scheme, it is important to consider the underlying security of the encryption and signature schemes. Informally, we can say that an encryption scheme is regarded as secure if an attacker is not able to find the plain-text corresponding to a chosen ciphertext. While a signature scheme is considered to be secure if an attacker is not able to create a forged signature on a message of its choice. Now, the security of the homomorphic encryption scheme can be defined as *Indistinguishability under chosen (non-adaptive) ciphertext attack (IND CCA1)* and security of the signature scheme as *Weak Unforgeability (WUF)*. To discuss in detail, we now model two kinds of adversaries: a malicious CA ( $\mathcal{A}_{ca}$ ) (who can try to forge a signature or decryption on behalf of the signer/receiver) and an uncertified user ( $\mathcal{A}_{ucu}$ ) [27], [28]. Accordingly, the security of the proposed CB-DA in the random oracle model (ROM) [35] has been defined by two attack games: Game-1(Security Against  $\mathcal{A}_{ca}$ ) and Game-2(Security Against  $\mathcal{A}_{ucu}$ ).

- Game-1(Security Against  $\mathcal{A}_{ca}$ ): Challenger  $\mathcal{C}$  runs Par-Generate( $1^\gamma$ ) to get system parameters  $\Theta$  and generates CA's secret key. Further,  $\mathcal{C}$  gives all outputs to  $\mathcal{A}_{ca}$ .  $\mathcal{A}_{ca}$  interleaves the following queries in a serial manner:

- 1)  $\mathcal{A}_{ca}$  makes various requests to Encrypt(.), Sign(.) and Decrypt(.) oracles to get corresponding outputs in non adaptive manner. Corresponding to these requests, challenger  $\mathcal{C}$  runs the oracles and gives outputs as response.
- 2)  $\mathcal{A}_{ca}$  generates two messages  $m_0, m_1$  and sends to  $\mathcal{C}$  to get corresponding ciphertexts.
- 3)  $\mathcal{C}$  returns challenged ciphertext  $C_b$  for a random  $b \in \{0, 1\}$ .

- 4) Finally,  $\mathcal{A}_{ca}$  outputs a forged signature  $\sigma^*$  and/or a guessed message  $m'_b$ ,  $b' \in \{0, 1\}$  corresponding to challenged ciphertext  $C_b$ .

$\mathcal{A}_{ca}$  wins the game if verification of  $\sigma^*$  is valid and/or guessed bit  $b' = b$ . The advantage of adversary is success probability defined as  $Adv_{CB-DA}^{CMA}(\mathcal{A}_{ca}) = Pr[SignVerify(\sigma^*) = 1]$  (corresponding to unforgeability) and/or  $Adv_{CB-DA}^{IND-CCA1}(\mathcal{A}_{ca}) = Pr[b = b'] - \frac{1}{2}$  (corresponding to IND CCA1).

- **Game-2(Security Against  $\mathcal{A}_{ucu}$ ):** Challenger  $\mathcal{C}$  runs Par-Generate( $1^\gamma$ ) to get system parameters  $\Theta$ . Further,  $\mathcal{C}$  gives all outputs to  $\mathcal{A}_{ucu}$ .  $\mathcal{A}_{ucu}$  interleaves the following queries in a serial manner:

- 1)  $\mathcal{A}_{ucu}$  makes various requests to Certify(.), Encrypt(.), Sign(.) and Decrypt(.) oracles to get corresponding outputs in non adaptive manner. Corresponding to these requests, challenger  $\mathcal{C}$  runs the oracles and gives outputs as response.
- 2)  $\mathcal{A}_{ucu}$  generates two messages  $m_0, m_1$  and sends to  $\mathcal{C}$  to get corresponding ciphertexts.
- 3)  $\mathcal{C}$  returns challenged ciphertext  $C_b$  for a random  $b \in \{0, 1\}$ .
- 4) Finally,  $\mathcal{A}_{ucu}$  outputs a forged signature  $\sigma^*$  and/or a guessed message  $m'_b$ ,  $b' \in \{0, 1\}$  corresponding to challenged ciphertext  $C_b$ .

$\mathcal{A}_{ucu}$  wins the game if verification of  $\sigma^*$  is valid and/or guessed bit  $b' = b$ . The advantage of adversary is success probability defined as  $Adv_{CB-DA}^{CMA}(\mathcal{A}_{ucu}) = Pr[SignVerify(\sigma^*) = 1]$  (corresponding to unforgeability) and/or  $Adv_{CB-DA}^{IND-CCA1}(\mathcal{A}_{ucu}) = Pr[b = b'] - \frac{1}{2}$  (corresponding to IND CCA1).

The proposed CB-DA scheme is CMA unforgeable and IND-CCA1-secure if the advantage of winning either Game-1 or Game-2 for an adversary is negligible. It should be noted that because of the IND CCA1 security, during each game, after step 3 adversary is not allowed to run Decrypt(.) oracle.

### III. PROPOSED CB-DA SCHEME

The CB-DA scheme is executed in the following five phases:

- 1) **Initialization:** This phase of the proposed scheme is executed in two steps:
  - a) *Par-Generate:* CA runs the step *Par-Generate*( $1^\gamma$ ) to generate system parameters  $\Theta = (1^\gamma, q, \mathcal{G}, \mathcal{G}_T, g, g_t, e, H_c, H_s, pk_{ca})$ . Where,  $1^\gamma$  is security parameter,  $q$  a prime,  $\mathcal{G}$  a order  $q$  cyclic group with generator  $g$ ,  $\mathcal{G}_T$  be another order  $q$  cyclic group with generator  $g_t$ ,  $e$  bilinear map and  $pk_{ca} = g^{x_{ca}}$  be master public key of CA. The secret key of CA is  $x_{ca}$ .  $H_c : \{0, 1\}^* \times \mathcal{G} \rightarrow \mathcal{G}$  and  $H_s : \mathcal{G} \times \mathcal{G} \times \{0, 1\}^* \rightarrow \mathcal{G}$  be two cryptographic hash.
  - b) *Extract:* Each user (i.e. SAs, SMs, BSs and ECC) runs this by using  $\Theta$  and outputs key pair. Suppose,  $(SA_1, SA_2, \dots, SA_k)$  be the set of SAs in a HAN,  $(SM_1, SM_2, \dots, SM_l)$  be the set of SMs in a region belonging to a BS,  $(BS_1, BS_2, \dots, BS_m)$  be the set of BSs connected to ECC. The (secret

key, public key) pairs of  $SA_i, SM_j, BS_u$  and ECC are  $(sk_{sa_i} = x_{sa_i}, pk_{sa_i} = g^{x_{sa_i}})$ ,  $(sk_{sm_j} = x_{sm_j}, pk_{sm_j} = g^{x_{sm_j}})$ ,  $(sk_{bs_u} = x_{bs_u}, pk_{bs_u} = g^{x_{bs_u}})$  and  $(sk_{ecc} = x_{ecc}, pk_{ecc} = g^{x_{ecc}})$  respectively.

- 2) **Certification:** This is executed between each user (i.e. SAs, SMs, BSs and ECC) and CA. CA certifies the identity and public key of each entity by using  $\Theta$  and  $x_{ca}$ . The certificates of  $SA_i, SM_j, BS_u$  and ECC are  $Cert_{sa_i} = H_c(ID_{sa_i} || pk_{sa_i})^{x_{ca}}$ ,  $Cert_{sm_j} = H_c(ID_{sm_j} || pk_{sm_j})^{x_{ca}}$ ,  $Cert_{bs_u} = H_c(ID_{bs_u} || pk_{bs_u})^{x_{ca}}$  and  $Cert_{ecc} = H_c(ID_{ecc} || pk_{ecc})^{x_{ca}}$  respectively.
- 3) **Data-Aggregate ( $SA_{AG}$ ):** This phase is completed using the following steps:
  - a) **Encrypt:** From  $HAN_j$ , each  $SA_i$  for  $(i = 1, 2, \dots, m)$  selects  $r_{ij} \in_R \mathbb{Z}_q^*$  and computes  $C_{1ij} = g^{r_{ij}}$  and  $C_{2ij} = g^{m_{ij} \cdot W^{r_{ij}}}$ . Where  $W = e(H_c(ID_{ecc} || pk_{ecc}), pk_{ca} \cdot pk_{ecc})$  is pre-stored in each SAs. Then,  $SA_i$  send  $M_1 = (C_{1ij}, C_{2ij})$  to  $SA_{AG}$  as the ciphertext.
  - b) **Aggregate:** The  $SA_{AG}$  computes  $C_{1j} = \prod_{i=1}^m C_{1ij}$ ,  $C_{2j} = \prod_{i=1}^m C_{2ij}$ .
  - c) **Sign ( $SA_{AG}$ ):** Then  $SA_{AG}$  selects  $r_j \in_R \mathbb{Z}_q^*$  and computes  $y_j = g^{r_j}$  and  $h_j = H_s(C_{2j} || y_j || t_j)$  for time stamp  $t_j$ . Then, computes  $\sigma_j = h_j^{(r_j + x_{ag})} \cdot Cert_{ag}$  and transfer  $M_2 = (C_{1j}, C_{2j}, y_j, \sigma_j, t_j)$  to  $SM_j$ .
  - d) **SignVerify (SM):** The  $SM_j$  checks the validity of signature  $(y_j, \sigma_j)$  by  $SA_{AG}$ .  $SM_j$  verify that  $e(\sigma_j, g) = e(H_s(C_{2j} || y_j || t_j), y_j \cdot pk_{ag}) \cdot e(H_c(ID_{ag} || pk_{ag}), pk_{ca})$ <sup>1</sup>
  - e) **Sign (SM):** If the verification is correct,  $SM_j$  computes the signature on  $(C_{1j}, C_{2j})$  by using similar steps as  $SA_{AG}$  has done. Suppose,  $(y_{sm_j}, \sigma_{sm_j})$  be the signature.  $SM_j$  transfers  $M_3 = (C_{1j}, C_{2j}, y_{sm_j}, \sigma_{sm_j}, t_{sm_j})$  to BS.
- 4) **Data-Aggregate (BS):** This phase has the following steps:

- a) **SignVerify(BS):** BS receives  $l$  signatures  $(y_{sm_1}, \sigma_{sm_1}), (y_{sm_2}, \sigma_{sm_2}), \dots, (y_{sm_l}, \sigma_{sm_l})$  on  $l$  ciphertext  $(C_{11}, C_{21}), (C_{12}, C_{22}), \dots, (C_{1l}, C_{2l})$ . For  $\theta_j \in_R \mathbb{Z}_q^*, 1 \leq j \leq l$  [2], BS checks the validity of  $e(\prod_{j=1}^l (\sigma_{sm_j})^{\theta_j}, g) = \prod_{j=1}^l e(H_s(C_{2j} || y_{sm_j} || t_{sm_j})^{\theta_j}, y_{sm_j} \cdot pk_{sm_j}) \cdot e(\prod_{j=1}^l H_c(ID_{sm_j} || pk_{sm_j})^{\theta_j}, pk_{ca})$  and displays accept/reject.
- b) **Aggregate (BS):** If output of validation is accept, BS computes the aggregated data as  $C_1 = \prod_{j=1}^l C_{1j}$

<sup>1</sup>It can be precomputed.



and  $C_2 = \prod_{j=1}^l C_{2j}$ .

- c) Sign (BS): Then BS selects  $r_{bs} \in_R \mathbb{Z}_q^*$  and computes  $y_{bs} = g^{r_{bs}}$  and  $h_{bs} = H_s(C_2 \| y_{bs} \| t_{bs})$  for time stamp  $t_{bs}$ . Then, computes  $\sigma_{bs} = h_{bs}^{(r_{bs} + x_{bs})}$ .  $Cert_{bs}$  and display  $(y_{bs}, \sigma_{bs})$  as signature on  $(C_1, C_2)$ . BS transfers  $M_4 = (C_1, C_2, y_{bs}, \sigma_{bs}, t_{bs})$  to ECC.

- 5) **Decryption (ECC):** ECC receives the ciphertext along with signatures from  $(BS_1, BS_2, \dots, BS_n)$ . For  $\theta_k \in_R \mathbb{Z}_q^*, 1 \leq k \leq n$  [2], ECC checks the validity of  $e(\prod_{k=1}^n (\sigma_{bs_k})^{\theta_k}, g) = \prod_{k=1}^n e(H_s(C_{2k} \| y_{bs_k} \| t_{bs_k})^{\theta_k}, y_{bs_k} \cdot pk_{bs_k})$ . If verified, ECC computes  $g_t^{\sum_{j=1}^l \sum_{i=1}^m m_{ij}} = \frac{C_2}{e(C_1, Cert_{ecc}, H_c(ID_{ecc} \| pk_{ecc})^{x_{ecc}})}$  and then compute total consumption data from  $g_t^{\sum_{j=1}^l \sum_{i=1}^m m_{ij}}$  by using [32], [36] method.

It should be noted that, since the consumer data is collected in every 15 min, hence the value of  $\sum_{j=1}^l \sum_{i=1}^m m_{ij}$  will be relatively small. Therefore, as per the discussion in [11], [14], for small exponents it is feasible to compute  $\sum_{j=1}^l \sum_{i=1}^m m_{ij}$  from  $g_t^{\sum_{j=1}^l \sum_{i=1}^m m_{ij}}$  by using [32] method in polynomial time.

#### IV. ANALYSIS OF THE PROPOSED CB-DA SCHEME

During the execution of the proposed CB-DA, smart grid entities ( $SA_{AG}$ , SAs, SMs, BSs and ECC) are signing (or verifying the signatures) associated with ciphertext. At the last, ECC decrypts the received aggregated ciphertext. Thus, a detailed proof of correctness for verification/decryption is presented. Since, the security of CB-DA is an important attribute. Therefore, a detailed security analysis in both, formal and informal ways, has also been presented.

##### A. Proof of Correctness

During execution, the correctness of signatures verification and data decryption are being the most important requirement to confirm authentication and integrity. Therefore, to discuss the correctness of signature verification and data decryption steps the following lemmas have been considered.

**Lemma-1.1:** The SignVerify(.) executed by SM is correct and it supports authentication to sign by  $SA_{AG}$ .

**Proof:** In the execution of Data-Aggregate( $SA_{AG}$ ),  $SA_{AG}$  sends  $(y_j, \sigma_j)$  along with ciphertext  $(C_{1j}, C_{2j})$  to SM. Thus, the authenticity of signature is checked by SM by running SignVerify(.). The detailed proof of correctness of SignVerify(.) is as follows:

$$\begin{aligned} e(\sigma_j, g) &= e(H_s(C_{2j} \| y_j \| t_j)^{(r_j + x_{ag})}, Cert_{ag}, g) \\ &= e(H_s(C_{2j} \| y_j \| t_j)^{(r_j + x_{ag})}, H_c(ID_{ag} \| pk_{ag})^{x_{ca}}, g) \\ &= e(H_s(C_{2j} \| y_j \| t_j)^{(r_j + x_{ag})}, g) \cdot e(H_c(ID_{ag} \| pk_{ag})^{x_{ca}}, g) \end{aligned}$$

$$\begin{aligned} &= e(H_s(C_{2j} \| y_j \| t_j), g^{(r_j + x_{ag})}) \cdot e(H_c(ID_{ag} \| pk_{ag}), g^{x_{ca}}) \\ &= e(H_s(C_{2j} \| y_j \| t_j), y_j \cdot pk_{ag}) \cdot e(H_c(ID_{ag} \| pk_{ag}), pk_{ca}). \end{aligned}$$

Therefore, correctness proves that the  $(ID_{ag} \| pk_{ag})$  pair of  $SA_{AG}$  are the key ingredients of verification. Thus, it supports the authentication and integrity of (ciphertext, signature) pair by  $SA_{AG}$ .

**Lemma-1.2:** The batch verification of SignVerify(.) executed by BS is correct.

**Proof:** During the Data-Aggregate(BS) phase execution, BS verifies the signatures  $(y_{sm_1}, \sigma_{sm_1}), (y_{sm_2}, \sigma_{sm_2}), \dots, (y_{sm_l}, \sigma_{sm_l})$  on  $l$  ciphertext  $(C_{11}, C_{21}), (C_{12}, C_{22}), \dots, (C_{1l}, C_{2l})$  in a batch. Thus, the correctness of batch verification is required. The detailed proof of the correctness is as follows

$$\begin{aligned} &e(\prod_{j=1}^l (\sigma_{sm_j})^{\theta_j}, g) \\ &= e(\prod_{j=1}^l (H_s(C_{2j} \| y_j \| t_j)^{(r_{sm_j} + x_{sm_j})}, Cert_{sm_j})^{\theta_j}, g) \\ &= \prod_{j=1}^l e(H_s(C_{2j} \| y_j \| t_j)^{\theta_j}, g^{(r_{sm_j} + x_{sm_j})}) \\ &\cdot e(\prod_{j=1}^l H_c(ID_{sm_j} \| pk_{sm_j})^{\theta_j}, g^{x_{ca}}) \\ &= \prod_{j=1}^l e(H_s(C_{2j} \| y_j \| t_j)^{\theta_j}, y_{sm_j} \cdot pk_{sm_j}) \\ &\cdot e(\prod_{j=1}^l H_c(ID_{sm_j} \| pk_{sm_j})^{\theta_j}, pk_{ca}). \end{aligned}$$

In this proof of correctness, it has been observed that  $(ID_{sm_j} \| pk_{sm_j})$  of each  $SM_j$  (jth smart meter) are the key ingredients. The hashed of ciphertext are also required to prove it. Thus, It supports the authentication and integrity.

**Lemma-1.3:** The batch verification of SignVerify(.) executed by ECC is correct.

**Proof:** During the Decryption(ECC) phase execution, ECC performs the verification of  $(y_{bs_1}, \sigma_{bs_1}), (y_{bs_2}, \sigma_{bs_2}), \dots, (y_{bs_n}, \sigma_{bs_n})$  signatures in a batch. Thus, the batch verification should work correctly. The detailed proof is as follows:

$$\begin{aligned} &e(\prod_{k=1}^n (\sigma_{bs_k})^{\theta_k}, g) \\ &= e(\prod_{k=1}^n (H_s(C_{2k} \| y_{bs_k} \| t_{bs_k})^{(r_{bs_k} + x_{bs_k})}, Cert_{bs_k})^{\theta_k}, g) \\ &= \prod_{k=1}^n e(H_s(C_{2k} \| y_{bs_k} \| t_{bs_k})^{\theta_k}, g^{(r_{bs_k} + x_{bs_k})}) \\ &\cdot e(\prod_{k=1}^n H_c(ID_{bs_k} \| pk_{bs_k})^{\theta_k}, g^{x_{ca}}) \\ &= \prod_{k=1}^n e(H_s(C_{2k} \| y_{bs_k} \| t_{bs_k})^{\theta_k}, y_{bs_k} \cdot pk_{bs_k}) \\ &\cdot e(\prod_{k=1}^n H_c(ID_{bs_k} \| pk_{bs_k})^{\theta_k}, pk_{ca}). \end{aligned}$$

The proof requires the pairs  $(ID_{bs_k} \| pk_{bs_k})$  of each  $BS_k$  (kth base station) as essential ingredient. Besides, the hashed values of ciphertext from each  $BS_k$  are also required. Thus, it supports authentication and integrity.

**Lemma-1.4:** The decryption executed by ECC is correct.

**Proof:** The receiving of original messages to ECC depends on the correctness of decryption algorithm. Thus, it is needed that the Decryption(.) run by ECC should be mathematically

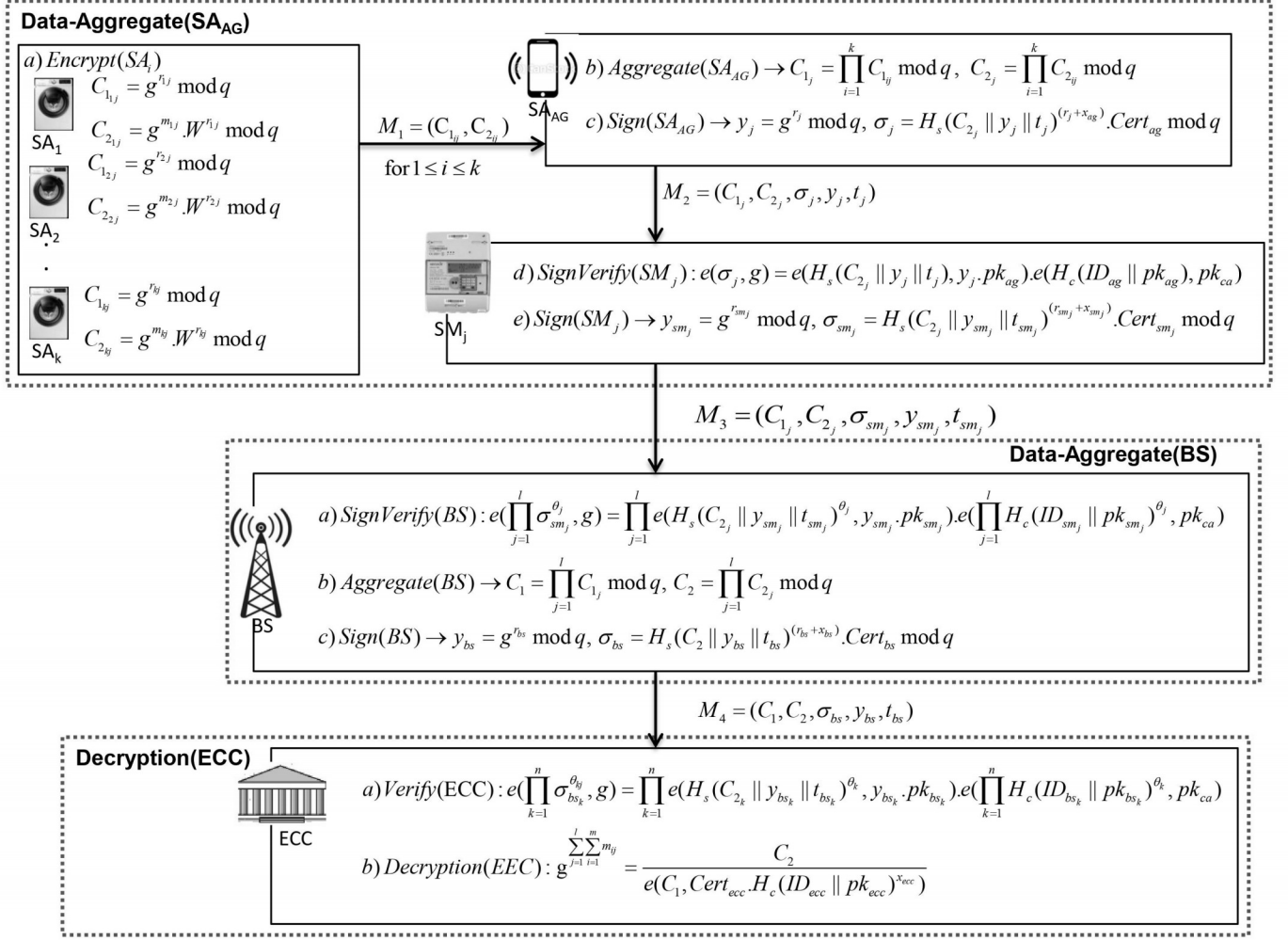


Fig. 3. Flow Diagram of Data Aggregation

correct. The detailed proof is as follows:

$$\begin{aligned}
 \text{Here, } C_1 &= \prod_{j=1}^l C_{1_j} = \prod_{j=1}^l \left( \prod_{i=1}^m C_{1_{ij}} \right) = \prod_{j=1}^l \left( \prod_{i=1}^m g^{r_{ij}} \right) = \\
 &= g^{\sum_{j=1}^l \sum_{i=1}^m r_{ij}} \quad \text{and} \quad C_2 = \prod_{j=1}^l C_{2_j} = \prod_{j=1}^l \left( \prod_{i=1}^m C_{2_{ij}} \right) = \\
 &= \prod_{j=1}^l \left( \prod_{i=1}^m g^{m_{ij}} \cdot W^{r_{ij}} \right) = g^{\sum_{j=1}^l \sum_{i=1}^m m_{ij}} \cdot W^{\sum_{j=1}^l \sum_{i=1}^m r_{ij}}. \\
 W^{\sum_{j=1}^l \sum_{i=1}^m r_{ij}} &= e(H_c(ID_{ecc} \parallel pk_{ecc}), g^{x_{ca}} \cdot g^{x_{ecc}})^{\sum_{j=1}^l \sum_{i=1}^m r_{ij}} \\
 &= e(H_c(ID_{ecc} \parallel pk_{ecc})^{(x_{ca} + x_{ecc})}, g^{\sum_{j=1}^l \sum_{i=1}^m r_{ij}}) \\
 &= e(C_1, Cert_{ecc} \cdot H_c(ID_{ecc} \parallel pk_{ecc})^{x_{ecc}}). \\
 \text{So, } g_t^{\sum_{j=1}^l \sum_{i=1}^m m_{ij}} &= \frac{C_2}{e(C_1, Cert_{ecc} \cdot H_c(ID_{ecc} \parallel pk_{ecc})^{x_{ecc}})}.
 \end{aligned}$$

To get original message correctly, the (secret key, certificate) pair of ECC is utilized. Without it, the proof of correctness will not work. Hence, it supports confidentiality.

### B. IND-CCA1 and Unforgeability Analysis

In the proposed CB-DA scheme for the smart grid communication, smart grid entities such as smart meters and base stations create signatures. Adversaries can try to create forgery signatures. Therefore, the unforgeability of the signing algorithms is been presented in this section. Further, ECC performs decryption to get the original message. Adversaries can also try to decrypt the received messages. As per the adversary model (Section II(E)), corresponding to each signer/receiver an adversarial uncertified user ( $\mathcal{A}_{ucc}$ ) is modeled. Another adversary who tries to create forgery signatures is malicious CA ( $\mathcal{A}_{ca}$ ). Here, we present the detailed security analysis definitions and related theorems.

To discuss the proof of the IND-CCA1 and unforgeability under CMA, the following two lemmas are presented. In these lemmas, in ROM, the simulation of the above two games is presented. Lemma 2.1 simulates the Game-1 and Lemma 2.2 simulates the Game-2.

**Lemma-2.1:** The adversary  $\mathcal{A}_{ca}$  can  $(t, q, \epsilon)$  break the proposed CB-DA scheme, then there exists a probabilistic poly-no-



mial time (PPT) adversary who can solve the random instance  $(g, g^a, g^b)$  of CDH-Problem with non-negligible probability. Here,  $t$  be the total running time of  $\mathcal{A}_{ca}$ ,  $q$  be the total number of queries during Game-1 and  $\epsilon$  be the success probability to break the CB-DA scheme.

**Proof:** Suppose,  $\mathcal{A}_{ca}$  can break the proposed encryption and signature scheme with success probability  $\epsilon$ . Then, challenger  $\mathcal{C}$  can compute  $g^{ab}$  as a solution to CDH-Problem  $(g, g^a, g^b)$ . The following interactive game is played between  $\mathcal{A}_{ca}$  and  $\mathcal{C}$ .

- **Par-Generate(.):**  $\mathcal{C}$  runs *Par-Generate*( $1^\gamma$ ) and gives  $\Theta = (1^\gamma, q, \mathcal{G}, \mathcal{G}_T, g, g_t, e, H_c, H_s, pk_{ca})$  and  $sk_{ca} = s$  to  $\mathcal{A}_{ca}$ .  $\mathcal{C}$  also sets  $h_0 = e(g, g)^{ab}$  and  $h_1 = e(g, g)^c$  for  $c \in_R \mathbb{Z}_q$ . Here,  $H_c(\cdot)$  and  $H_s(\cdot)$  works as random oracles.
- **Extract(.):**  $\mathcal{C}$  records the responses in  $L_{ext}$  of tuple  $(ID, pk_{id}, x_{id})$ .  $\mathcal{A}_{ca}$  sends  $ID_i, (1 \leq i \leq q_{ext})$  to *Extract*(.)-Oracle.  $\mathcal{C}$  scans  $L_{ext}$  and if  $(ID_i, pk_{id_i}, x_{id_i})$  exists, forward it to  $\mathcal{A}_{ca}$ . Otherwise, response is following:
  - If  $i \neq j$ ,  $\mathcal{C}$  sets  $pk_{id_i} = g^{x_{id_i}}$  for  $x_{id_i} \in_R \mathbb{Z}_q$  and sends to  $\mathcal{A}_{ca}$ .
  - If  $i = j$ ,  $\mathcal{C}$  sets  $pk_{id_i} = g^a$  and sends to  $\mathcal{A}_{ca}$ .

Update  $L_{ext}$ .

- **$H_c(\cdot)$ -Oracle:**  $\mathcal{C}$  stores the responses in  $L_{H_c}$  of tuple  $(ID, pk_{id}, v)$ . On a queried  $(ID_i, pk_{id_i})$ ,  $\mathcal{C}$  scans  $L_{H_c}$  and if entry exists, sends  $H_c(ID_i || pk_{id_i}) = g^{v_i}$ . Otherwise, computes  $H_c(ID_i || pk_{id_i}) = g^{v_i}$  for  $v_i \in_R \mathbb{Z}_q$  and sends to  $\mathcal{A}_{ca}$ . Update  $L_{H_c}$ .
- **$H_s(\cdot)$ -Oracle:**  $\mathcal{C}$  stores the responses in  $L_{H_s}$  of tuple  $(ID_i, pk_{id_i}, C_{1i}, C_{2i}, h_{s_i})$ . On a queried  $(ID_i, pk_{id_i}, C_{1i}, C_{2i})$ ,  $\mathcal{C}$  scans  $L_{H_s}$  and if entry exists forwards to  $\mathcal{A}_{ca}$ . Otherwise, does as follows:
  - If  $i \neq j$ ,  $\mathcal{C}$  sets  $h_{s_i} = g^{\alpha_i}$  for  $\alpha_i \in_R \mathbb{Z}_q$  and sends to  $\mathcal{A}_{ca}$ .
  - If  $i = j$ ,  $\mathcal{C}$  sets  $h_{s_i} = g^b$  and sends to  $\mathcal{A}_{ca}$ .
- **Corruption(.)-Oracle:**  $\mathcal{C}$  responds as follows:
  - If  $i \neq j$ ,  $\mathcal{C}$  scans  $L_{ext}$  and sends  $x_{id_i}$  to  $\mathcal{A}_{ca}$ .
  - If  $i = j$ ,  $\mathcal{C}$  sends  $\perp$  to  $\mathcal{A}_{ca}$ .
- **Sign(.)-Oracle:** To respond  $(ID_i, C_{1i}, C_{2i})$ ,  $\mathcal{C}$  runs all the above oracles and get the outputs. Then, does the following:
  - If  $i \neq j$ ,  $\mathcal{C}$  runs *Certify*(.)-Oracle and *Sign*(.)-Oracle and outputs the signature and certificate by using the responses.
  - If  $i = j$ ,  $\mathcal{C}$  selects  $e_j, d_j, z_j \in_R \mathbb{Z}_q$  and computes  $y_j = g^{(z_j - e_j \cdot s) \cdot d_j^{-1} \cdot pk_{ID_j}^{-1}}$ . Then, sets  $\sigma_j = g^{z_j}$ ,  $H_s(C_{2j} || y_j || t_j) = g^{d_j}$  and  $H_c(ID_j || pk_{ID_j}) = g^{e_j}$ . Then,  $\mathcal{C}$  scans lists  $L_{H_s}$  and  $L_{H_c}$  and check the hashed values. If, collision occurs, re-selects the random values and do the above computation. Finally,  $\mathcal{C}$  sends  $(\sigma_j, y_j)$  as signature to  $\mathcal{A}_{ca}$ .
- **Decryption(.)-Oracle:** To respond  $(ID_i, C_{1i}, C_{2i})$ ,  $\mathcal{C}$  acts as follows:
  - If  $i = j$ ,  $\mathcal{C}$  scans  $L_{H_c}$  and  $L_{ext}$  and picks the tuples  $(ID_i, v_i, g^{v_i})$  and  $(ID_i, pk_{ID_i}, \perp)$  respectively. Then, computes  $g_t^m = C_{2i} \cdot W^{-1}$  where

$W = e((pk_{ID_i}, pk_{ca})^{v_i}, C_{1i})$ . Then recover  $m$  and forwards the outputs to  $\mathcal{A}_{ca}$ .

- If  $i \neq j$ ,  $\mathcal{C}$  runs the *Decryption*(.) Oracle and sends output to  $\mathcal{A}_{ca}$ .

- **Challenge:**  $\mathcal{A}_{ca}$  outputs two aggregated messages  $(\sum m)_0$  and  $(\sum m)_1$  of same length corresponding to an identity  $ID^* \notin L_{ext}$  and also output a ciphertext  $(C_1^*, C_2^*) = (g^a, g_t^{(\sum m)_b} \cdot h_b)$  where  $b \in \{0, 1\}$  after requesting to  $\mathcal{C}$ .

**Output:**  $\mathcal{A}_{ca}$  outputs  $b'$  as its guess and a forged signature  $(\sigma^*, y_j^*)$  corresponding to  $ID^* \notin L_{ext}$ .  $\mathcal{A}_{ca}$  wins the game, either  $b' = b$  or  $SignVerify(\sigma^*, y_j^*) = \text{accepts}$ .

**Note-1:** If  $b = 0$ , then  $(C_1^*, C_2^*)$  is a valid ciphertext of  $(\sum m)_b$ . Otherwise for  $b = 1$ ,  $h_1$  is a random from  $\mathcal{G}_T$  and therefore  $(\sum m)_b$  is completely hidden by  $C_2^*$ . The difference of probability of these cases is  $\epsilon$  as  $\mathcal{A}_{ca}$  break the CB-DA with probability  $\epsilon$ . Thus, it reveals that  $Pr[b = b'] = 1/2 + \epsilon$ . Here, we have taken the probability  $\epsilon$  as non-negligible. Therefore, adversary is able to identify the bit with probability greater than 1/2. It contradicts the assumption that DBDH being hard. Thus, our assumption is incorrect and therefore the proposed CB-DA is IND-CCA1 secure.

**Note-2:**  $\mathcal{A}_{ca}$  outputs a forged signature  $(\sigma^*, y_j^*)$  on  $(C_1^*, C_2^*)$  with  $ID^* \notin L_{ext}$ . Then, the equation  $e(\sigma^*, g) = e(H_s(C_2^* || y_j^* || t_j^*), y_j^* \cdot pk_{ID^*}) e(H_c(ID^* || pk_{ID^*}), pk_{ca})$  is verified. From the simulation, by setting  $H_s(C_2^* || y_j^* || t_j^*) = g^b$ ,  $pk_{ID^*} = g^a$ ,  $y_j^* = g^\delta$ ,  $pk_{ca}$  and  $H_c(ID^* || pk_{ID^*}) = g^v$ ,  $\mathcal{C}$  can compute  $\sigma^* \cdot (g^{vx} \cdot (g^b)^v)^{-1}$  as a solution to CDH-Problem.

The success probability depends on the events:  $E_1$ -Challenger's output is not  $\perp$ ,  $E_2$ -Adversary outputs forgery successfully and  $E_3$ -Forged signature's identity is  $ID_j$ . Thus,  $P[E_1] \geq \left(1 - \frac{1}{q_{H_c}}\right)^{q_1 + q_2}$ ,  $P[E_2|E_1] \geq \epsilon$  and  $P[E_3|E_1 \wedge E_2] \geq \frac{1}{q_{H_c}}$ , where  $q_1, q_2$  and  $q_{H_c}$  be the number of corruption, signing and  $H_c(\cdot)$  oracle queries respectively. So, success probability to solve CDH-problem  $\geq \frac{1}{q_{H_c}} \left(1 - \frac{1}{q_{H_c}}\right)^{q_1 + q_2} \epsilon$ .

**Remark-1:** In the proposed CB-DA scheme, public and secret keys are known to owner only (such as SAs, SMs, BSs and ECC). Therefore, CA or any other entity except owner cannot create a valid signature (or cannot decrypt the ciphertext). Thus, as per the adversary model, in the smart grid communication infrastructure CA may try to forge the signatures created by  $SA_{AG}$  (or SM or BS). CA may also try to decrypt the ciphertext from HAN. In the Lemma-2.1, the security of the proposed CB-DA scheme has been proved against the adversary  $\mathcal{A}_{ca}$ . The adversary  $\mathcal{A}_{ca}$  is playing the role of malicious CA. Therefore, in the proposed CB-DA scheme CA is not able to forge signatures of any smart grid entity. Besides, the encryption in CB-DA is also secure. Thus, CA also cannot decrypt the messages from  $SA_{AG}$ . Therefore, the CB-DA scheme is secure against malicious CA attack.

**Lemma-2.2:** The adversary  $\mathcal{A}_{ucu}$  can  $(t, q, \epsilon)$  break the proposed CB-DA scheme, then there exists a PPT adversary who can solve the random instance  $(g, g^a, g^b)$  of CDH-Problem with non-negligible probability. Here,  $t$  be the total running time of  $\mathcal{A}_{ucu}$ ,  $q$  be the total number of queries during Game-2 and  $\epsilon$  be the success probability to break the CB-DA scheme.

**Proof:** Suppose,  $\mathcal{A}_{ucv}$  can break the proposed encryption and signature scheme with success probability  $\epsilon$ . Then, challenger  $\mathcal{C}$  can compute  $g^{ab}$  as a solution to CDH-Problem  $(g, g^a, g^b)$ . The following interactive game is played between  $\mathcal{A}_{ucv}$  and  $\mathcal{C}$ .

- **Par-Generate(.):**  $\mathcal{C}$  runs *Par-Generate*( $1^\gamma$ ) and gives  $\Theta = (1^\gamma, q, \mathcal{G}, \mathcal{G}_T, g, g_t, e, H_c, H_s, pk_{ca} = g^a)$  to  $\mathcal{A}_{ucv}$ .  $\mathcal{C}$  also sets  $h_0 = e(g, g)^{ab}$  and  $h_1 = e(g, g)^c$  for a  $c \in_R \mathbb{Z}_q$ . Here,  $H_c(\cdot)$  and  $H_s(\cdot)$  works as random oracles.
- **Extract(.):**  $\mathcal{C}$  records the responses in  $L_{ext}$  of tuple  $(ID, pk_{id}, x_{id})$ .  $\mathcal{A}_{ucv}$  sends  $ID_i, (1 \leq i \leq q_{ext})$  to *Extract*(.)-Oracle.  $\mathcal{C}$  scans  $L_{ext}$  and if  $(ID_i, pk_{id_i}, x_{id_i})$  exists, forward it to  $\mathcal{A}_{ucv}$ . Otherwise, sets  $pk_{id_i} = g^{x_{id_i}}$  for  $x_{id_i} \in_R \mathbb{Z}_q$  and sends to  $\mathcal{A}_{ucv}$ . Update  $L_{ext}$ .
- **$H_s(\cdot)$ -Oracle:**  $\mathcal{C}$  stores the responses in  $L_{H_s}$  of tuple  $(ID_i, pk_{id_i}, C_{1i}, C_{2i}, h_{si})$ . On a queried  $(ID_i, pk_{id_i}, C_{1i}, C_{2i})$ ,  $\mathcal{C}$  scans  $L_{H_s}$  and if entry exists, sends  $h_{si}$  to  $\mathcal{A}_{ucv}$ . Otherwise, computes  $h_{si} = g^{\alpha_i}$  for  $\alpha_i \in_R \mathbb{Z}_q$  and sends to  $\mathcal{A}_{ucv}$ . Update  $L_{H_s}$ .
- **$H_c(\cdot)$ -Oracle:**  $\mathcal{C}$  stores the responses in  $L_{H_c}$  of tuple  $(ID, pk_{id}, v)$ . On a queried  $(ID_i, pk_{id_i})$ ,  $\mathcal{C}$  scans  $L_{H_c}$  and if entry exists forwards to  $\mathcal{A}_{ucv}$ . Otherwise, does as follows:
  - If  $i \neq j$ ,  $\mathcal{C}$  sets  $h_{ci} = g^{v_i}$  for  $v_i \in_R \mathbb{Z}_q$  and sends to  $\mathcal{A}_{ucv}$ .
  - If  $i = j$ ,  $\mathcal{C}$  sets  $h_{ci} = g^b$  and sends to  $\mathcal{A}_{ucv}$ .
- **Corruption(.)-Oracle:** To respond,  $\mathcal{C}$  scans  $L_{ext}$  and if entry exists, sends  $x_{id_i}$  to  $\mathcal{A}_{ucv}$ . Otherwise, picks  $x_{id_i} \in_R \mathbb{Z}_q$  and computes  $pk_{ID_i} = g^{x_{id_i}}$  and sends  $x_{id_i}$  to  $\mathcal{A}_{ucv}$ .
- **Certify(.)-Oracle:** On requested  $(ID, pk_{id})$ ,  $\mathcal{C}$  scans  $L_{H_c}$  and get the value  $h_c$  and then does as follows:
  - If  $i \neq j$ ,  $\mathcal{C}$  runs *Certify*(.)-Oracle and sends output to  $\mathcal{A}_{ucv}$ .
  - If  $i = j$ ,  $\mathcal{C}$  scans  $L_{H_c}$  and picks  $v_j$  from the list. Then, sets  $Cert_{ID_j} = (g^a)^{v_j}$  and  $H_c(ID_j || pk_{ID_j}) = g^{v_j}$ . Sends  $Cert_{ID_j}$  to  $\mathcal{A}_{ucv}$ .
- **Sign(.)-Oracle:** To respond  $(ID_i, C_{1i}, C_{2i})$ ,  $\mathcal{C}$  runs all the above oracles and get the outputs. Then, does the following:
  - If  $i \neq j$ ,  $\mathcal{C}$  runs *Sign*(.)-Oracle and outputs the signature by using the responses.
  - If  $i = j$ ,  $\mathcal{C}$  selects  $e_j, d_j, z_j \in_R \mathbb{Z}_q$  and computes  $y_j = g^{(z_j - e_j \cdot a) \cdot d_j^{-1} \cdot pk_{ID_j}^{-1}}$ . Then, sets  $\sigma_j = g^{z_j}$ ,  $H_s(C_{2j} || y_j || t_j) = g^{d_j}$  and  $H_c(ID_j || pk_{ID_j}) = g^{e_j}$ . Then,  $\mathcal{C}$  scans lists  $L_{H_s}$  and  $L_{H_c}$  and check the hashed values. If, collision occurs, re-selects the random values and do the above computation. Finally,  $\mathcal{C}$  sends  $(\sigma_j, y_j)$  as signature to  $\mathcal{A}_{ucv}$ .
- **Decryption(.)-Oracle:** To respond  $(ID_i, C_{1i}, C_{2i})$ ,  $\mathcal{C}$  acts as follows:
  - If  $i = j$ ,  $\mathcal{C}$  scans  $L_{H_c}$  and  $L_{ext}$  and picks the tuples  $(ID_i, v_i, g^{v_i})$  and  $(ID_i, pk_{ID_i}, \perp)$  respectively. Then, computes  $g_t^m = C_{2i} \cdot W^{-1}$  where  $W = e((pk_{ID_i} \cdot pk_{ca})^{v_i}, C_{1i})$ . Then recover  $m$  and forwards the outputs to  $\mathcal{A}_{ucv}$ .
  - If  $i \neq j$ ,  $\mathcal{C}$  runs *Decryption*(.) oracle and sends output to  $\mathcal{A}_{ucv}$ .

- **Challenge:**  $\mathcal{A}_{ucv}$  obtains challenged ciphertext  $(C_1^*, C_2^*) = (g^a, g_t^{(\sum m)_b} \cdot h_b)$  in the similar way as in the previous Lemma 2.1.

**Output:**  $\mathcal{A}_{ucv}$  outputs a forged signature  $(\sigma^*, y_j^*)$  corresponding to  $ID^* \notin L_{ext}$ .  $\mathcal{A}_{ca}$  wins the game if *SignVerify*( $\sigma^*, y_j^*$ )=accepts.

**Note-1:** If  $b = 0$ , then  $(C_1^*, C_2^*)$  is a valid ciphertext. Otherwise for  $b = 1$ ,  $h_1$  is a random from  $\mathcal{G}_T$ . Thus,  $Pr[b = b'] = 1/2 + \epsilon > 1/2$  as  $\mathcal{A}_{ucv}$  break the CB-DA with non negligible probability  $\epsilon$ . Thus, it contradicts the assumption that DBDH is hard. So, our assumption that  $\mathcal{A}_{ucv}$  break the CB-DA is incorrect. Thus, CB-DA is IND-CCA1 secure.

**Note-2:**  $\mathcal{A}_{ucv}$  outputs a forged signature  $(\sigma^*, y_j^*)$  on  $(C_1^*, C_2^*)$  with  $ID^* \notin L_{ext}$ . Then, the equation  $c(\sigma^*, g) = e(H_s(C_2^* || y_j^* || t_j^*), y_j^* \cdot pk_{ID^*}) e(H_c(ID^* || pk_{ID^*}), pk_{ca})$  is verified. From the simulation, by setting  $H_s(C_2^* || y_j^* || t_j^*) = g^v$ ,  $pk_{ID^*} = g^x$ ,  $y_j^* \cdot pk_{ca} = g^a$  and  $H_c(ID^* || pk_{ID^*}) = g^b$ ,  $\mathcal{C}$  can compute  $\sigma^* \cdot (y_j^* \cdot g^x)^{-v}$  as a solution to CDH-Problem.

The success probability depends on the events:  $E_1$ -Challenger's output is not  $\perp$ ,  $E_2$ -Adversary outputs forgery successfully and  $E_3$ -Forged signature's identity is  $ID_j$ .

Thus,  $P[E_1] \geq \left(1 - \frac{1}{q_{H_c}}\right)^{q_1 + q_2}$ ,  $P[E_2|E_1] \geq \epsilon$  and  $P[E_3|E_1 \wedge E_2] \geq \frac{1}{q_{H_c}}$ , where  $q_1, q_2$  and  $q_{H_c}$  be the number of certification, signing and  $H_c(\cdot)$  oracle queries respectively. So, success probability to solve CDH-problem  $\geq \frac{1}{q_{H_c}} \left(1 - \frac{1}{q_{H_c}}\right)^{q_1 + q_2} \epsilon$ .

**Remark-2:** As per the description of the proposed CB-DA scheme for smart grid, each smart grid entity SAs, SMs, BSs and ECC creates their key pair. Then, get certificate from CA. So, as per the adversarial model, any attacker can define a duplicate smart grid entity. This, duplicate smart grid entity can generate their keys but do not request certificate. Thus, it is defined as uncertified user and denoted by  $\mathcal{A}_{ucv}$ . This attacker may try to create a forged certificate to generate a genuine signature or to decrypt a ciphertext. However, in the Lemma-2.2 the proposed CB-DA scheme is proved secure against such attacker. Therefore, attacker is not able to create a forgery of signature by any smart grid entity. The encryption devised in CB-DA is also secure. Thus, attacker cannot decrypt the ciphertext in unauthorized way. Therefore, proposed CB-DA is secure against such attacker  $\mathcal{A}_{ucv}$ .

**Lemma-2.3:** The proposed batch verification of the signatures in CB-DA scheme is secure.

**Proof:** The batch verification is done by

$$\begin{aligned}
 & e\left(\prod_{k=1}^n (\sigma_k)^{\theta_k}, g\right) \\
 &= e\left(\prod_{k=1}^n (H_s(C_{2k} || y_k || t_k)^{(r_k + x_k)} \cdot Cert_k)^{\theta_k}, g\right) \\
 &= \prod_{k=1}^n e(H_s(C_{2k} || y_k || t_k)^{\theta_k}, g^{(r_k + x_k)}) \cdot e\left(\prod_{k=1}^n H_c(ID_k || pk_k)^{\theta_k}, g^x\right) \\
 &= \prod_{k=1}^n e(H_s(C_{2k} || y_k || t_k)^{\theta_k}, y_k \cdot pk_k) \cdot e\left(\prod_{k=1}^n H_c(ID_k || pk_k)^{\theta_k}, pk_{ca}\right).
 \end{aligned}$$

Where,  $\sigma_k, H_s(C_{2k} || y_k || t_k)$  and  $H_c(ID_k || pk_k)$  are the elements from  $\mathcal{G}$ . So, let  $\sigma_k = g^{\alpha_k}$ ,  $H_s(C_{2k} || y_k || t_k) = g^{\beta_k}$  and  $H_c(ID_k || pk_k) = g^{\gamma_k}$  for some  $\alpha_k, \beta_k$  and  $\gamma_k$  from  $\mathbb{Z}_q^*$ . Thus, the batch verification is now

$$e(\prod_{k=1}^n (g^{\alpha_k \cdot \theta_k}, g) = \prod_{k=1}^n e(g^{\beta_k \cdot \theta_k}, g^{r_k + x_k}) \cdot e(\prod_{k=1}^n g^{\gamma_k \cdot \theta_k}, g^x).$$

$$\text{So, } e(g, g)^{\sum_{k=1}^n \alpha_k \cdot \theta_k} = e(g, g)^{\sum_{k=1}^n (r_k + x_k) \beta_k \cdot \theta_k} \cdot e(g, g)^{\sum_{k=1}^n x \cdot \gamma_k \cdot \theta_k}.$$

$$\text{So, } \sum_{k=1}^n \alpha_k \cdot \theta_k - (\sum_{k=1}^n (r_k + x_k) \beta_k \cdot \theta_k + \sum_{k=1}^n x \cdot \gamma_k \cdot \theta_k) \bmod q = 0.$$

$$\text{Let, } \delta_k = \alpha_k - ((r_k + x_k) \beta_k + x \gamma_k).$$
 Assume that  $\text{BatchVerify}((C_k, ID_k, \sigma_k), k = 1, 2, \dots, n) = \text{Accept}$ , but for some  $j$ , there exists an event such that  $\text{Verify}(C_j, ID_j, \sigma_j) = \text{Reject}$ . In this case, we can compute  $\theta_j = -\delta_j^{-1} \left\{ (\sum_{k=1}^n \theta_k \delta_k) - \theta_j \cdot \delta_j \right\} \bmod q$ . However,  $\theta_j$  is a randomly chosen from  $\mathbb{Z}_q^*$  with  $l$  bit length. Thus,  $\Pr[E] \leq \frac{1}{2^l}$  and therefore advantage of invalid signature is negligible.

**Remark-3:** During the smart grid communication, any entity may try to insert invalid signature to muddle aggregation phase. The invalid signature may be designed for  $SA_{AG}$ , SM or BS. Thus, as per the security of batch verification (Lemma-2.3) the invalid signature can be identified by computing  $\theta_j = -\delta_j^{-1} \left\{ (\sum_{k=1}^n \theta_k \delta_k) - \theta_j \cdot \delta_j \right\} \bmod q$ .

### C. Security Analysis

The attributes satisfied by the CB-DA scheme are as follows:

- **Internal Attack:** As an internal attacker, there are some possible entities;  $SA_{AG}$ , SM, BS and ECC. As  $SA_{AG}$  gets the cipher texts  $(C_{1ij} = g^{r_{ij}}, C_{2ij} = g_t^{m_{ij}} \cdot W^{r_{ij}})$ . Thus, it is not possible to extract consumption data without secret key of ECC. Similarly, SM gets aggregated data  $(C_{1j}, C_{2j})$  and hence it is also unable to decrypt it. Suppose, if BS tries to get  $\sum_{i=1}^m m_{ij}$  from the received  $(\prod_{i=1}^m g^{r_{ij}}, \prod_{i=1}^m g_t^{m_{ij}} \cdot W^{r_{ij}})$ , the  $\text{Decrypt}(\cdot)$  needs the certificate and secret key of ECC. So, to get consumption  $\sum_{i=1}^m m_{ij}$  of a single HAN by BS is infeasible. To get  $\sum_{i=1}^m m_{ij}$  from  $\sum_{i=1}^m \sum_{j=1}^l m_{ij}$  by ECC is also infeasible. Thus, proposed CB-DA withstand against internal attackers.
- **MITM Attack:** In this system model, MITM is comprised of the following attacks:
  - **Impersonation Attack:** Based on the Lemma 2.1 and Lemma 2.2, no adversary can get a forged signature or message. Thus, the CB-DA scheme can withstand against impersonation attack.
  - **Replay Attack:** The execution of the proposed CB-DA scheme requires real time stamps to compute signatures  $((y_j, \sigma_j)$  or  $(y_{sm_j}, \sigma_{sm_j})$  or  $(y_{bs}, \sigma_{bs}))$  by different users ( $SA_{AG}$  or SM or BS). Thus, the freshness of the information can be checked at any instant of time by BS or by ECC. So, CB-DA scheme withstand against this attack.
  - **Modification Attack:** According to the execution of different phases of the proposed CB-DA scheme, any modification to any information will lead to unsuccessful  $\text{SignVerify}(\cdot)$ . For example, suppose

an attacker try to modify the contents communicated from SM to BS. However, BS executes  $\text{SignVerify}(\cdot)$  by checking  $e(\prod_{j=1}^l (\sigma_{sm_j})^{\theta_j}, g) = \prod_{j=1}^l e(H_s(C_{2j} \| y_{sm_j} \| t_{sm_j})^{\theta_j}, y_{sm_j} \cdot pk_{sm_j}) \cdot e(\prod_{j=1}^l H_c(ID_{sm_j} \| pk_{sm_j})^{\theta_j}, pk_{ca})$ . In this process, hashed value  $H_s(C_{2j} \| y_{sm_j} \| t_{sm_j})$  is required. Therefore, the content modification will result a different hashed value. Thus, the execution of  $\text{SignVerify}(\cdot)$  with different hashed value will output reject. Thus, modification attack is also unsuccessful to the CB-DA scheme.

Thus, the proposed CB-DA scheme withstands against MITM attack.

- **Authentication:** Based on the proof of unforgeability (Lemma 2.1 and Lemma 2.2), in the proposed CB-DA scheme only legitimate user ( $SA_{AG}$  or SM or BS) can generate signature  $((y_j, \sigma_j)$  or  $(y_{sm_j}, \sigma_{sm_j})$  or  $(y_{bs}, \sigma_{bs}))$  on the cipher text  $((C_{1j}, C_{2j})$  or  $(C_{1j}, C_{2j})$  or  $(C_{1j}, C_{2j})$  by using its certificate and secret key. SM can verify signatures by checking the equality  $e(\sigma_j, g) = e(H_s(C_{2j} \| y_j \| t_j), y_j \cdot pk_{ag}) e(H_c(ID_{ag} \| pk_{ag}), pk_{ca})$ . Similarly, BS and ECC can also check by suitable equality. However, during  $\text{SignVerify}(\cdot)$ , the genuine public key and ID of user are required to display accept. The proof of correctness of signature verification by concerned entities (Lemma 1.1, Lemma 1.2 and Lemma 1.3) also supports it. Thus, the proposed CB-DA scheme satisfies authentication.
- **Integrity:** As per the discussion, we found that the scheme is secure against MITM attack. Therefore, any attacker is unable to perform *Impersonation Attack*, *Replay Attack* or *Modification Attack* successfully i.e. the message sent from HAN to ECC is unaltered. Therefore, the proposed CB-DA scheme achieves message integrity.
- **Confidentiality:** As per the description of the proposed CB-DA scheme, the decryption is done by computing 
$$g_t^{\sum_{j=1}^l \sum_{i=1}^m m_{ij}} = \frac{C_2}{e(C_1, \text{Cert}_{ecc}, H_c(ID_{ecc} \| pk_{ecc})^{x_{ecc}})}$$
. Thus, secret key and certificate of ECC are key ingredients for execution. Based on the proof of Lemma 2.1 and Lemma 2.2, the success probability of an adversary to get plain text corresponding to challenged ciphertext is negligible. Thus, to get the power usage of a customer by an adversary is infeasible. Thus the proposed CB-DA scheme satisfies confidentiality.
- **Privacy:** From the above discussion, it is observed that the proposed CB-DA scheme is robust against both type of attackers i.e. internal and external attackers. Therefore, if an attacker captures the cipher-text (either of a single HAN or aggregated from several HANs), it is not feasible to get single customer data as plain-text. Without individual customer data, it is not possible to capture life-style related information. Therefore, we can say that the proposed scheme achieves privacy against such attackers.

TABLE III  
COMPARING SECURITY ATTRIBUTES WITH OTHER SCHEMES

Scheme	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9
Fan <i>et al.</i> [2]	×	×	×	✓	×	✓	×	✓	×
Z. Wang [11]	×	✓	✓	×	✓	×	✓	✓	✓
Vahedi <i>et al.</i> [14]	✓	✓	✓	✓	×	✓	×	✓	×
Chen <i>et al.</i> [19]	✓	✓	✓	✓	×	✓	×	✓	×
Proposed Scheme	✓	✓	✓	✓	✓	×	✓	✓	✓

SP1:Key Leakage Resistance, SP2:Authentication Achieved SP3: Integrity Achieved  
SP4:Key Escrow Resistance, SP5:Easy Certificate Revocation, SP6:Secure Channel Required  
SP7:Secure (Impersonation Attack), SP8:Single Point Failure, SP9:Secure (Replay Attack)

TABLE IV  
COMPUTATION COSTS OF DIFFERENT OPERATIONS ON DEVICES

Operation	Notation	SA/SM/ $SA_{AG}$	BS/ECC
Bilinear Pairing	$T_{bp}$	1726.81 ms	31.34 ms
Modular Exponentiation	$T_{ex}$	193.67 ms	2.31 ms
Modular Multiplication	$T_m$	561.93 ms	13.69 ms
Hash	$T_h$	0.62 ms	0.011 ms
Pairing Multiplication	$T_{mp}$	496.89 ms	13.69 ms
Pairing Inversion	$T_{ip}$	216.75 ms	3.42 ms
Scalar Multiplication	$T_{sm}$	409.32 ms	6.42 ms
Map to Point Hash	$T_{ph}$	132.65 ms	2.14 ms
Elliptic Point Addition	$T_{pa}$	126.38 ms	1.69 ms

## V. PERFORMANCE COMPARISON

The purpose of the proposed scheme is to design a secure and lightweight PP-DA solution. Therefore, in this section the performance comparison of the proposed CB-DA scheme is made and analyzed in terms of several imperative security attributes, the computation costs and the communication costs. In this regard, here we consider the literature [2], [11], [14] and [19] for comparison, as these schemes are based on pairing similar to our CB-DA. In Table III, the security attributes satisfied by [2], [11], [14] and [19] are considered. Since, the schemes in [2], [14] and [19] are PKI-based developments, so the certificate management is required and this would increase cost and extra burden on SGI devices. Scheme in [11] is based on IBI and it does not require any certificate management. However, it suffers from the key escrow problem. This would cause the private key leakage issue. On the other hand, the proposed scheme does not require any certificate management and is also robust against key escrow, as secret key is generated by its owner. As in the proposed scheme, certificate works as partial secret key. So, only owner needs its certificate. Thus, certificate revocation can also be done easily. When CA is updated, certificate owner can easily get an updated certificate from the CA. Thus, the proposed CB-DA can be considered more secure than mentioned schemes. Another serious issue with the schemes presented in [2], [11], [14] and [19] is that inside their HAN, no SA encrypts its data. Therefore, an attacker who is able to access HAN network can get the data readings of SAs. Thus, in these schemes the efficiency is achieved by compromising the security.

To discuss the computation cost comparison, we now consider some of the imperative cryptographic operations (as shown in Table IV) used in the proposed scheme and the schemes presented in [2], [11], [14] and [19]. In Table V, the total computation costs on different devices are considered.

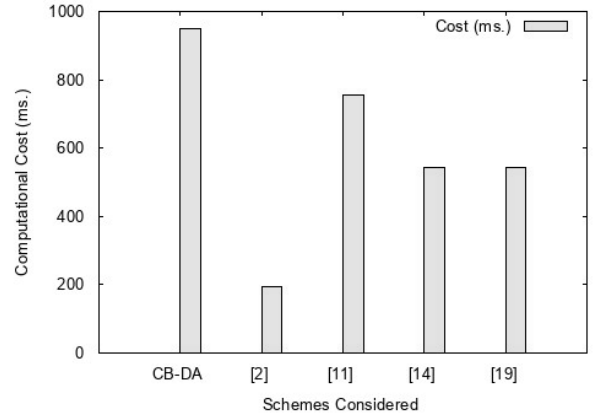


Fig. 4. Comparing computational cost of signing at SM.

Since in the schemes presented in [2], [11], [14] and [19], SAs need not to perform any encryption. Hence, we do not compare these costs at SA. Furthermore, in these schemes aggregation is performed at BS/ECC. Whereas, in the proposed scheme a two step aggregation is performed (within HAN by  $SA_{AG}$  and between ECC-to-BS by BS). In order to show the effectiveness of our proposed scheme, we simulated the cryptographic operations (used in the proposed scheme and [2], [11], [14] and [19]) on a SASEBO-GII board (operating as a SM, SA, and  $SA_{AG}$ ) with a 16-bit MSP430 microcontroller is configured with 128K-Byte of data memory and 32K-Byte of program memory. We implemented this design at a System Clock of 25 MHz to reflect the constrained platform for the devices. The data (as shown in Table IV) indicates that the protocol already fits into a small microcontroller. Besides, a 4300 dual-core 2.6 GHz CPU (operating as BS/ECC as per the scheme) has been used to evaluate their computation time. The simulation also uses the JCE library [33] and JBPC library Pbc-05.14 [34]. The computational costs obtained are given in Table IV.

For a detailed comparison, we assume that the numbers of SAs, SMs and BSs are  $m(= 10)$ ,  $l$  and  $n$  respectively. In schemes [2], [11], [14] and [19] the operations by SAs and  $SA_{AG}$  are not applicable. So, we compare the costs of signing by SM, and the costs of operations by BS and ECC. From Table V, we can note that the computational cost at SM for the schemes [2], [11], [14] and [19] are  $1T_{ex} + 1T_h$  (194.29 ms),  $1T_{ex} + 1T_h + T_m$  (756.22 ms),  $1T_{sm} + 1T_{ph}$  (541.97 ms) and  $1T_{sm} + 1T_{ph}$  (541.97 ms) respectively. While, for the proposed CB-DA the cost at SM is  $2T_{ex} + 1T_h + T_m$  (949.89 ms). Thus, we observe that the schemes [2], [11], [14] and [19]

TABLE V  
COMPUTATION COST COMPARISON

Scheme	SA	SA <sub>AG</sub>	Sign(SM)	BS	ECC
Fan <i>et al.</i> [2]	NA	NA	$1T_{ex} + 1T_h$	NA	$(2ln + 3)T_{ex} + (3ln - 1)T_m + 1T_h + (ln - 1)T_{mp} + (ln + 1)T_{bp}$
Z. Wang [11]	NA	NA	$1T_{ex} + 1T_h + 1T_m$	$(3l + 2)T_{ex} + 2lT_h + lT_{mp} + (4l - 3)T_m + (l + 2)T_{bp}$	$3nT_{bp} + nT_{ip}$
Vahedi <i>et al.</i> [14]	NA	NA	$1T_{ph} + 1T_{sm}$	$(l + 1)T_{bp} + (l + 1)T_{ph} + lT_{pa} + (l - 1)T_{mp} + (l + 1)T_{sm}$	$(n + 1)T_{bp} + nT_{ph} + 1T_{sm} + (n - 1)T_{mp}$
Chen <i>et al.</i> [19]	NA	NA	$1T_{ph} + 1T_{sm}$	$2lT_{bp} + (m - 1)lT_m + 1T_{ph} + 1T_{sm}$	$(2n + 1)T_{bp} + mnT_m$
Proposed Scheme	$3T_{ex}$	$2T_{ex} + 1T_h + (2m - 1)T_m$	$2T_{ex} + 1T_m + 1T_h$	$(2l + 3)T_{ex} + lT_{mp} + 1T_h + (2l - 1)T_m + (l + 2)T_{bp}$	$3nT_{ex} + nT_m + 1T_{ip} + (n + 3)T_{bp}$

TABLE VI  
COMMUNICATION COST COMPARISON

Scheme	SM-to-BS	BS-to-ECC
Fan <i>et al.</i> [2]	1280	NA
Z. Wang [11]	1024	1024
Vahedi <i>et al.</i> [14]	768	768
Chen <i>et al.</i> [19]	$(k+1)256$	$(k+1)256$
Proposed Scheme	1024	1024

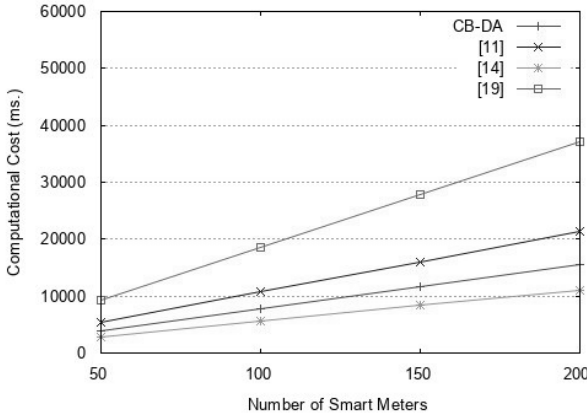


Fig. 5. Comparing computational cost of BS.

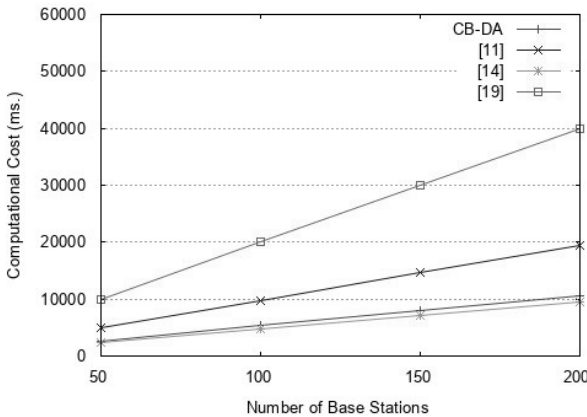


Fig. 6. Comparing computational cost of ECC.

are consuming 20.45%, 79.61%, 57.06% and 57.06% of the proposed CB-DA scheme. Therefore, the computational costs of the schemes [2], [11], [14] and [19] are lower than the proposed CB-DA (Figure 4). However, in the proposed CB-DA scheme, SM generates its secret key and certificate (with secret key) is used for signing. Thus, in the proposed CB-DA scheme, SM is secure against key escrow and no burden of certificate management is there. While, scheme in [11] suffers with key escrow and schemes in [2], [14] and [19] are having burden of certificate management.

In Figure 5 and Figure 6, the costs consumed by BS and ECC are compared. In detail, from Table V we found that cost consumed at BS for different number of SMs (i.e.  $l$ ) for the schemes [11], [14] and [19] are  $\approx 137.26\%$ ,  $\approx 71.58\%$  and  $\approx 238.69\%$  respectively of the proposed CB-DA. Similarly, the cost consumed at ECC for different number of BSs (i.e.  $n$ ) for the schemes [11], [14] and [19] are  $\approx 180.74\%$ ,  $\approx 88.39\%$  and  $\approx 371.36\%$  respectively of the proposed CB-DA. Based on the number of devices, we see that the performance of the proposed scheme is better than [11] and [19]. However, we can also observe that the performance of the scheme presented in [14] is better than CB-DA in terms of computational cost. However, our proposed scheme ensures a better security level as compared to any state-of-the-art solutions. In the scheme presented in [2], BS and ECC are single entity. So, its cost is  $90.72ln + 10.901$  ms, which is quite large than the proposed CB-DA. Therefore, the proposed CB-DA offers best solution to data aggregation in SGI at high security level.

Now, we consider the storage cost at the SM and the communication cost between SM-to-BS and BS-to-ECC. In this regard, we consider the bit sizes of  $\mathcal{G}$ ,  $\mathcal{G}_T$ ,  $N$ ,  $ID$  and elliptic curve points as 256, 256, 1024, 32 and 256 respectively. In the SGI, SM is the device with least storage capacity. In the proposed CB-DA scheme, SM needs to store  $(ID, pk_{sm}, sk_{sm}, Cert_{sm})$  which cost  $32 + 3 \times 256 = 800$  bits. Thus, SM can easily handle such storage cost. Therefore, as far as the storage cost on SM is considered, the proposed CB-DA is suitable for the resource-limited smart meters. Table VI shows the communication cost between SM-to-BS and BS-to-ECC, where we can see that the communication costs (for both phases) of [2], [11], [14], [19] and the proposed CB-DA are 1280, 1024, 768,  $(k+1)256$  and 1024 bits respectively. Thus, the cost of [14] is the least one. However, their scheme requires certificate management. The scheme in [19] is having variable length, which will impact on the communication cost.

From Table VI, we can see that the communication cost is considered, CB-DA is a suitable choice for data aggregation for HAN in SGI.

## VI. CONCLUSION

The data aggregation in SGI is the imperative requirement for many purposes such as demand-response management. However, it uses sensitive and confidential data from the smart grid devices (such as the smart meter) through the insecure wireless communication links. Therefore, with the help of cryptographic algorithms, data aggregation schemes are devised. However, due to PKI or IBI based construction, most of the schemes are not satisfying all security issues and requirements. To overcome such issues, this paper presents a new and efficient data aggregation protocol. The proposed scheme eliminates the security issues (such as certificate revocation or key escrow) of the previously existing schemes. The performance comparison with simulation results demonstrate the effectiveness of the proposed scheme. In the literature, most of the PP-DA schemes are centralized systems (due to certification or key generation authority). Therefore, as a future scope, a data aggregation with good scalable features will be proposed.

## REFERENCES

- [1] M. E. Kantarci and H. T. Mouftah, "Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 179–197, 1st Quart., 2015.
- [2] C. I. Fan, S. Y. Huang, Y. L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, Vol. 10, no. 1, pp. 666–675, 2014.
- [3] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, X. Shen, "EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key Evolution in Smart Grid," *IEEE Trans. Parallel Distrib. Syst.*, 25, 2053–2064, 2014.
- [4] T. Shiobara, P. Palensky, H. Nishi, "Effective metering data aggregation for smart grid communication infrastructure," In *IECON 2015–41st Annual Conference of the IEEE Industrial Electronics Society* 2015 Nov 9 (Yokohama, Japan), pp. 002136–002141, IEEE.
- [5] D. He, N. Kumar, J. H. Lee, "Privacy-preserving data aggregation scheme against internal attackers in smart grids," *Wireless Networks*, vol. 22, no. 2, pp. 491–502, 2016.
- [6] X. F. Wang, Y. Mu, R. M. Chen, "An efficient privacy-preserving aggregation and billing protocol for smart-grid," *Secur. Commun. Netw.*, 9, pp. 4536–4547, 2016.
- [7] H. J. Jo, I. S. Kim and D. H. Lee, "Efficient and Privacy-Preserving Metering Protocols for Smart Grid Systems," in *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1732–1742, 2016.
- [8] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 396–405, 2016.
- [9] C. Melchor and P. Gaborit, "Lattice-based homomorphic encryption of vector spaces," In *proc. IEEE ISIT*, Canada, July 2008.
- [10] H. Bao and R. Lu, "Comment on "Privacy-Enhanced Data Aggregation Scheme Against Internal Attackers in Smart Grid"," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 2–5, 2016.
- [11] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2428–35, 2017.
- [12] D. He, N. Kumar, S. Zeadally, A. Vinel, L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, 2017.
- [13] S. Li, K. Xue, Q. Yang, P. Hong, "PPMA: Privacy-preserving multi-subset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 62–71, 2018.
- [14] E. Vahedi, M. Bayat, M. R. Pakravan, M. R. Aref, "A secure ECC-based privacy preserving data aggregation scheme for smart grids," *Computer Networks*, vol. 129, pp. 28–36, 2017.
- [15] P. Gope and B. Sikdar, "An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3126–35, 2018.
- [16] R. Chaudhary, G. S. Aujla, N. Kumar, A. K. Das, N. Saxena, J. J. Rodrigues, "LaCSys: Lattice-based cryptosystem for secure communication in smart grid environment," In *2018 IEEE International Conference on Communications (ICC)*, 2018 May 20, pp. 1–6, IEEE.
- [17] A. Jindal, B. Bhambu, M. Singh, N. Kumar, S. Naik, "A Heuristic-Based Appliance Scheduling Scheme for Smart Homes," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3242 – 3255, 2020.
- [18] A. Saleem, A. Khan, S. U. Malik, H. Pervaiz, H. Malik, M. Alam, A. Jindal, "FESDA: Fog-Enabled Secure Data Aggregation in Smart Grid IoT Network," *IEEE Internet of Things Journal*, Vol. 7, no. 7, 2019, pp. 6132 – 6142.
- [19] Y. Chen, J. F. Martínez-Ortega, P. Castillejo, L. López, "An Elliptic Curve-Based Scalable Data Aggregation Scheme for Smart Grid," *IEEE Systems Journal*, Vol. 14, no. 2, 2020, pp. 2066–2077.
- [20] K. Xue, B. Zhu, Q. Yang, D. S. Wei, M. Guizani, "An Efficient and Robust Data Aggregation Scheme without A Trusted Authority for Smart Grid," *IEEE Internet of Things Journal*, Vol. 7, no. 3, 2020, pp. 1949 – 1959.
- [21] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Proc. Int. Symp. Privacy Enhancing Technol.*, 2011, pp. 175–191.
- [22] G. Danezis, C. Fournet, M. Kohlweiss, and S. Zanella-Béguelin, "Smart meter aggregation via secret-sharing," in *Proc. 1st ACM Workshop Smart Energy Grid Secur.*, 2013, pp. 75–80.
- [23] F. Knirsch, G. Eibl, and D. Engel "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3351–3361, Jul. 2016.
- [24] H. Mohammed, S. Tonyali, K. Rabieh, M. Mahmoud, and K. Akkaya, "Efficient privacy-preserving data collection scheme for smart grid AMI networks," in *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.
- [25] P. Gope, and B. Sikdar, "Lightweight and Privacy-Friendly Spatial Data Aggregation for Secure Power Supply and Demand Management in Smart Grids," *IEEE Transactions on Information Forensics and Security*, Vol. 14, No. 6, 2019.
- [26] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," In *International conference on the theory and applications of cryptographic techniques*, 1999, May 2, pp. 223–238, Springer, Berlin, Heidelberg.
- [27] C. Gentry, "Certificate-based encryption and the certificate revocation problem," In *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 272–293, Springer, Berlin, Heidelberg, 2003.
- [28] B. G. Kang, J. H. Park, and S. G. Hahn, "A certificate-based signature scheme," In *Cryptographers' Track at the RSA Conference*, pp. 99–111. Springer, Berlin, Heidelberg, 2004.
- [29] J. Li, Z. Wang, and Y. Zhang, "Provably secure certificate-based signature scheme without pairings," *Information Sciences*, vol. 233 (2013), pp. 313–320.
- [30] Y. Lu, Jiguo Li, and Yichen Zhang, "Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks," *IEEE Transactions on Services Computing* (2019).
- [31] L. Chen, J. Li and Yichen Zhang, "Anonymous Certificate-Based Broadcast Encryption With Personalized Messages," *IEEE Transactions on Broadcasting* (2020).
- [32] J. H. Cheon, J. Hong, and M. Kim, "Accelerating Pollard's Rho algorithm on finite fields," *Journal of cryptology*, vol. 25, no. 2, pp. 195–242, 2012.
- [33] Oracle Technology Network. Java Cryptography Architecture (JCA). Accessed: Sept. 20, 2020. [Online]. Available: <http://docs.oracle.com/javase/6/docs/technotes/guides/crypto/CryptoSpec.html>.
- [34] Pbc Library. Accessed: Sept. 20, 2020. [Online]. Available: <http://crypto.stanford.edu/pbc/>
- [35] M. Bellare, P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. In: *First ACM Conference on Computer and Communications Security*. Proceedings, Fairfax, Nov 1993. ACM Press, New York, pp 62–73.
- [36] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC press, 2018.