Secure and Privacy-Preserving Concentration of Metering Data in AMI Networks

Neetesh Saxena[1], Bong Jun Choi[2], and Santiago Grijalva[3]
[1]Department of Computing & Informatics, Bournemouth University, UK
[2]Department of Computer Science, The State University of New York, South Korea & Stony Brook University, USA
[3]School of Electrical & Computer Engineering, Georgia Institute of Technology, USA
Email: nsaxena@ieee.org, bjchoi@sunykorea.ac.kr, sgrijalva@ece.gatech.edu

***Abstract*—The industry has recognized the risk of cyber-attacks targeting to the advanced metering infrastructure (AMI). A potential adversary can modify or inject malicious data, and can perform security attacks over an insecure network. Also, the network operators at intermediate devices can reveal private information, such as the identity of the individual home and metering data units, to the third-party. Existing schemes generate large overheads and also do not ensure the secure delivery of correct and accurate metering data to all AMI entities, including data concentrator at the utility and the billing center. In this paper, we propose a secure and privacy-preserving data aggregation scheme based on additive homomorphic encryption and proxy re-encryption operations in the Paillier cryptosystem. The scheme can aggregate metering data without revealing the actual individual information (identity and energy usage) to intermediate entities or to any third-party, hence, resolves identity and related data theft attacks. Moreover, we propose a scalable algorithm to detect malicious metering data injected by the adversary. The proposed scheme protects the system against man-in-the-middle, replay, and impersonation attacks, and also maintains message integrity and undeniability. Our performance analysis shows that the scheme generates manageable computation, communication, and storage overheads and has efficient execution time suitable for AMI networks.**

I. Introduction

The Smart Grid (SG) is a next-generation power system with intelligent electricity generation, transmission, and distribution [1]. Advanced Metering Infrastructure (AMI) network has two-way communication with the Smart Meter (*SM*), Aggregator (*AG*) with Gateway (*GW*), Communication Server (*CS*), Data Concentrator Unit (*DCU*) at the utility, and Billing Center (*BC*). The smart meters in the AMI network periodically send metering data to the the *DCU* through aggregators. As illustrated in Figure 1, the metering data from a group of smart meters collected by an aggregator is forwarded to the operator at the control center to take necessary actions by monitoring the *DCU*.

Delivering secure and privacy-preserving metering data over the network has become more challenging due to potential cyber-attacks and weak network security [2]. An adversary can inject or modify data over the network, and can also trace behavioral patterns of the household owner to whom the metering data belongs to. The adversary can also modify individual meter readings or intermediate aggregated results computed by the aggregators.

The Open Smart Grid Protocol (OSGP), a family of specifications published by the European Telecommunications Standards Institute (ETSI), is used for smart grid applications along with ISO/IEC 14908 control networking standard. The OSGP aims to provide reliable and efficient delivery of command and control information for different smart grid devices, such as smart meters, direct load control modules, solar panels, and gateways. Over 4 million OSGP-based smart meters and devices have already been deployed worldwide [3]. However, certain weaknesses have been identified in the OSGP protocol, such as the use of a weak digest function that leaks key information and several key recovery attacks [4], [5].

A. Research Problem

In this paper, we address the problem of securely delivering metering data from the smart meters to the utility and the billing center through intermediate devices, such as aggregators and the communication server. The existing schemes generate large overheads and do not secure the communications between all entities in AMI. These schemes also do not address the detection of malicious smart metering data, if any, and its data removal. In addition, if the transmission of metering data over the network is not secure, the adversary can modify or inject malicious data, re-send previous meter reading, and impersonate entities. We also address the privacy problem of revealing the identity of the individual home and metering data units, which may occur at the intermediate devices during periodic data transmission. The network operators operating intermediate devices can reveal such private information and

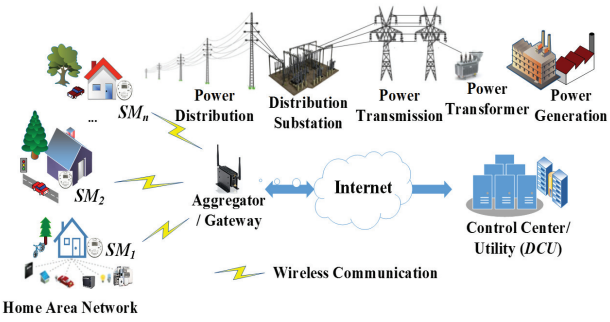


Fig. 1: A scenario of home area network in the smart grid.

pass to the third-party for financial benefits. Therefore, we need a complete secure and privacy-preserved scheme that can work efficiently and accurately, even when a large number of smart meters is deployed in AMI.

B. Our Contribution

In this paper, we present a secure and privacy-preserved scheme for transmitting the metering data from different *SM*s through *AG*s to the *DCU* at utility and the *BC*. Our scheme, based on the homomorphic encryption and proxy re-encryption, allows *AG*s to perform operations over the encrypted data for different smart grid applications. The details of our contribution are as follow. The proposed scheme

1) Provides authentication between all AMI network entities, *i.e.*, *SM*, *AG*, *CS*, *BC*, and *DCU*.
2) Provides privacy-preservation while aggregating metering data from different *SM*s and makes the aggregated data available to the *DCU* at utility. Different from existing schemes, our scheme also securely transmits individual metering data to the *BC* for billing purposes.
3) Protects the system against Man-in-the-Middle (MITM), replay, and impersonation attacks.
4) Generates low and manageable computation and communication overheads, maintains data integrity, and uses far less storage space than what currently deployed smart meters are equipped with, and has lower execution time than the existing schemes.

II. Related Work

In this section, we present existing works on aggregating metering data in the AMI network. F. Li *et al.* [6], [7] presented distributed in-network aggregation approaches to efficiently aggregate smart metering data along a spanning tree. However, these approaches do not consider authentication and integrity protection. Efthymiou *et al.* [8] proposed a third party escrow mechanism for authenticating anonymous meter readings. However, aggregators in the scheme do not perform any operation over the transmitted data. Garcia *et al.* [9] proposed a privacy-preserving protocol to aggregate partial shares of each metering data, but the protocol is not scalable and does not discuss scheme's overhead and efficiency. Rial and Danesiz [10] proposed a privacy preserving protocol using zero knowledge proof that enables the payment without revealing electricity consumption information. F. Li *et al.* [11] introduced an end-to-end signature scheme that supports batch verification of the aggregated results. However, both schemes do not present the scenario of transmitting aggregated data to the billing center and the utility. H. Li *et al.* [12] proposed a demand response scheme to achieve privacy-preserving demand aggregation and efficient response. However, the scheme generates a large number of keys as well as a large overhead. C. Li *et al.* [13] proposed a dual-functional aggregation scheme in which each user reports one data and then multiple statistic values of all users are computed by the data and control center. However, the scheme does not discuss the scenario of transmitting data to the billing center.

III. Design Goals and Preliminaries

In this section, we present our design goals and preliminaries for the proposed scheme.

A. Design Goals

We consider that the *DCU* and the *BC* are trusted by all entities in the network, and it is infeasible for an adversary to compromise them. The aggregators are honest but curious. Specifically, we consider the following design goals to be achieved for security and privacy:

1) Intermediate devices must be authenticated before forwarding the metering data.
2) The metering data must not be revealed to the intermediate devices, such as the *AG* and the *CS*. Even if an adversary can access the messages at the *AG* or the *CS*, it cannot retrieve the actual meter readings.
3) Message integrity must be provided, and generated overheads must be low in order to support a large number of deployed smart meters in AMI.

B. Preliminaries

This section presents a preliminary discussion on bilinear pairing and homomorphic encryption schemes.

1) Bilinear Pairing: Let $\mathbb{G}$ be an additive group and $\mathbb{G}_T$ be a multiplicative group on a symmetric pairing function e. Both groups are of order q, where q is a large prime. Let P be an arbitrary generator of $\mathbb{G}$. Assume that the discrete logarithm problem (*DLP*) is hard in both $\mathbb{G}$ and $\mathbb{G}_T$.

Definition: A bilinear pairing on ($\mathbb{G}$, $\mathbb{G}_T$) is a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ that satisfies the following properties:

(a) *Bilinearity:* $e(aP, bQ) = e(P,\ Q)^{ab}$; $\forall\ a,\ b \in \mathbb{Z}_q^*$, and $\forall\ P,\ Q \in \mathbb{G}$
(b) *Non-degeneracy:* $e(P,\ P) \neq 1$
(c) *Computability:* There exists an efficient algorithm to compute $e(P, Q)$ for $\forall P, Q \in \mathbb{G}$.

Here, given $P, aP, bP, cP \in \mathbb{G}$, and $a, b, c \in \mathbb{Z}_q^*$, it is easy to verify whether $c = ab \mod q$. However, it is difficult to compute abP.

2) Homomorphic Encryption Scheme: In order to perform addition and multiplication operations over the encrypted data, a homomorphic encryption is used to compute the aggregated sum or product of a group of data values. We use a homomorphic encryption scheme [14] to perform additive operations, which we restate for better clarity as follows:

(a) *Key generation:* Here, the public keys and global parameters are generated given a security parameter. Consider two primes as p and q, and $N = pq$. Choose a generator of the group $g \in \mathbb{Z}_{N^2}^*$ having an order (multiple of N). Let $\lambda(N) = lcm(p-1, q-1)$, where $lcm(p-1, q-1)$ represents least common multiple of $p-1$ and $q-1$. Then, public and secret keys of the receiver are generated as $PK = (N, g)$ and $SK = (\lambda(N))$, respectively.

(b) *Encryption:* The sender chooses a message $M \in \mathbb{Z}_N$ and a random number $r \in \mathbb{Z}^*_{N^2}$. Then, the ciphertext C is computed as

$$C = E(M) = g^M r^N \ mod \ N^2,$$

where r^N is used to generate different ciphertexts, even when the same message is encrypted more than once.

(c) *Decryption:* The receiver retrieves the original message from C to by computing

$$M = D(C) = \frac{L(C^{\lambda(N)} \ mod \ N^2)}{L(g^{\lambda(N)} \ mod \ N^2)} \ mod \ N,$$

where input from the set $\{u < N^2 | u = 1 \ mod \ N\}$ is given to the function L to compute $L(u) = (u-1)/N$. In additive homomorphism, two different ciphertexts $C_1 = E(M_1)$ and $C_2 = E(M_2)$ are computed from M_1, $M_2 \in \mathbb{Z}_N$ by the sender and the sum of the plaintexts is retrieved by the receiver as $D(C_1.C_2 \ mod \ N^2) = (M_1 + M_2) \ mod \ N$.

IV. Proposed Scheme

In this section, we propose a scheme that unlike the existing schemes, provides a secure delivery of metering data to the *BC* and the *DCU* using homomorphic and proxy encryptions and also authenticates each entity involved. We also describe an algorithm to detect malicious data at the *AG*.

A. System Architecture

We present a system architecture for secure concentration in the AMI network. Our system architecture, as shown in Figure 2, includes *SM*s, *AG*s, *CS*, *DCU*, and *BC*. The *SM*s are deployed in homes, the *AG*s are located in the wide area network between the homes and the *DCU* along with a *CS*. The AMI geographical area is divided into a number of clusters consisting of homes. Each *SM* sends its metering data to the nearest *AG*, which then transmits metering data to the *CS*. The *CS* processes and further sends the data to the *BC* and the *DCU*. The communication network between the *SM-AG* and the *AG-CS-BC-DCU* can be provided using Zigbee/Wi-Fi, and LTE/WiMAX, respectively.

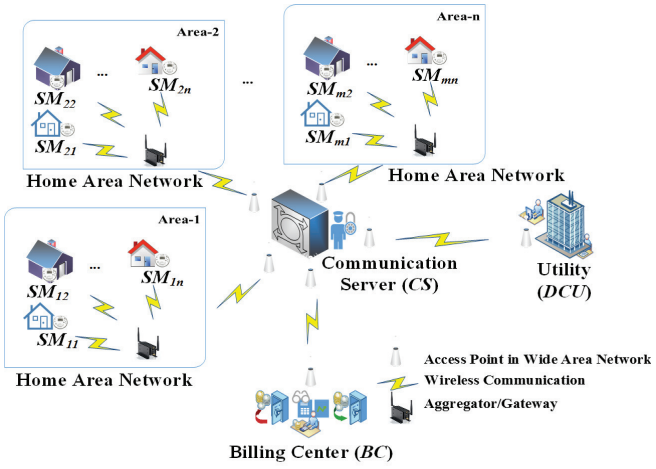


Fig. 2: Proposed system architecture in the AMI network.

B. Proposed Scheme

We present our proposed scheme that provides mutual authentication between different entities in the AMI network, as shown in Figure 3. Different from existing schemes, the proposed scheme ensures the secure periodic delivery of individual metering data to the *BC* for billing purpose as well as an aggregated consumed metering data to the *DCU* for grid control purpose. We describe our scheme in two parts: scheme initialization and scheme execution.

1) Scheme Initialization: The initialization of the proposed scheme consists of keys generation at different entities in the AMI network. The corresponding secret and public keys (SK and PK) are generated as follows:

(a) ***SM$_i$:*** $SK_{SM_i} = s_i \in \mathbb{Z}^*_q$ and $PK_{SM_i} = g^{s_i}$, where i = 1, 2, ...n and n is the total number of *SM*.
(b) ***AG:*** $SK_{AG} = a \in \mathbb{Z}^*_q$ and $PK_{AG} = g^a$.
(c) ***BC:*** $SK_{BC} = b \in \mathbb{Z}^*_q$ and $PK_{BC} = g^b$.
(d) ***DCU:*** $SK_{DCU} = d \in \mathbb{Z}^*_q$ and $PK_{DCU} = g^d$.
(e) ***CS:*** $SK_{CS} = c \in \mathbb{Z}^*_q$ and $PK_{CS} = g^c$.

The *CS* also generates two re-encryption keys as $RK_{CS \to BC} = g^{b/c} \in \mathbb{G}$ and $RK_{CS \to DCU} = g^{d/c} \in \mathbb{G}$ for the *BC* and *DCU*, respectively.

2) Scheme Execution: The scheme execution consists of computations and communications at different entities in the AMI network from *SM*s to *DCU* and *BC*.

(a) ***SM:*** Each SM_i generates encrypted metering data using an additive homomorphic encryption scheme to allow addition over the encrypted data (while preserving data privacy). The scheme uses a public key of the *BC* and its private key for individual data retrieval at the *BC* (preserving data confidentiality). Consider $w = e(g, g)$ and $r_i \in \mathbb{N}^*$ is a random number. Each SM_i sends its encrypted metering data m_i periodically, say 15 minutes, using the *AG*'s public key (N, g) along with meter's identity ID_i by computing

$$C_{1_i} = g^{m_i} \ r_i^N \ mod \ N^2.$$

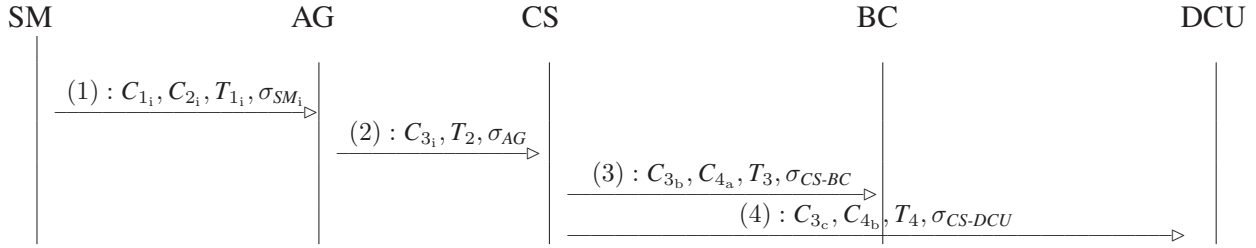
Thereafter, each SM_i computes

$$M_i = m_i \oplus (PK_{BC})^{SK_{SM_i}} || ID_i = m_i \oplus (g^b)^{s_i} || ID_i,$$
$$C_{2_a} = w^{r_i}.M_i,$$
$$C_{2_b} = (PK_{AG})^{r_i} = (g^a)^{r_i} = g^{a.r_i}, \text{and}$$
$$C_{2_i} = (C_{2_a}, C_{2_b}) = (w^{r_i}.M_i, g^{a.r_i}).$$

Each SM_i generates its efficient and short signature

$$\sigma_{SM_i} = H(C_{1_i} || C_{2_i})^{SK_{SM_i}} = H(C_{1_i} || C_{2_i})^{s_i}.$$

Then, each SM_i sends $(C_{1_i}, C_{2_i}, T_{1_i}, \sigma_{SM_i})$ to the *AG*, where T_{1_i} is the timestamp when the SM_i sends metering data.



$C_{1_i} = g^{m_i}\ r_i^N\ mod\ N^2$, $C_{2_a} = w^{r_i}.M_i$, $M_i = m_i \oplus (g^b)^{s_i}||ID_i$, $C_{2_b} = (g^a)^{r_i}$, $C_{2_i} = (C_{2_a}, C_{2_b}) = (w^{r_i}.M_i, g^{a.r_i})$, $\sigma_{SM_i} = H(C_{1_i}||C_{2_i})^{s_i}$, $C_{3_i} = (C_{3_a}, C_{3_b}, C_{3_c})$, $C_{3_a} = g^{c.t}$, $C_{3_b} = w^t.M$, $C_{3_c} = w^t.M_{AG}$, $M = M_1||M_2||M_3||...||M_i$, $C_{4_a} = e(C_{3_a}, g^{b/c})$, $C_{4_b} = e(C_{3_a}, g^{d/c})$, $\sigma_{CS-BC} = H(C_{3_b}||C_{4_a})^c$, $\sigma_{CS-DCU} = H(C_{3_c}||C_{4_b})^c$, $PK = (N, g)$, $SK = (\lambda(N))$, $w = e(g, g)$, $r_i \in \mathbb{N}^*$, m_i: metering data.

Fig. 3: Proposed scheme for the AMI network.

(b) ***AG:*** Upon receiving the message, the *AG* first computes $H^{'}(C_{1_i}||C_{2_i})$ and verifies the signatures in a batch as

$$\begin{aligned} e(g, \sigma_{SM_i}) &\stackrel{?}{=} \prod_{i=1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i})) \\ &\stackrel{?}{=} \prod_{i=1}^{n} e(g^{s_i}, H^{'}(C_{1_i}||C_{2_i})) \\ &\stackrel{?}{=} \prod_{i=1}^{n} e(g, H^{'}(C_{1_i}||C_{2_i})^{s_i}). \end{aligned}$$

Computing only one hash per *SM* and verifying the signatures in a batch improve the overall efficiency of the system. This process ensures the authenticity of each *SM* as well as messages integrity. Similarly, the verification of the signatures at the *CS*, *BC*, and *DCU* can be derived as will be discussed in the following subsections. If the verification is successful, the *AG* collects all the data received during a specific time interval as

$$\begin{aligned} C_{AG} &= \prod_{i=1}^{n} (C_{1_i}) \\ &= g^{m_1+m_2+...+m_n}\ (r_1 r_2 ... r_n)^N\ mod\ N^2, \end{aligned}$$

and applies homomorphic decryption using $\lambda(N)$ key as

$$\begin{aligned} M_{AG} &= D(C_{AG}) \\ &= \frac{L(C_{AG}^{\lambda(N)}\ mod\ N^2)}{L(g^{\lambda(N)}\ mod\ N^2)}\ mod\ N \\ &= \frac{L(g^{(m_1+m_2+...+m_n)\lambda(N)}\ mod\ N^2)}{L(g^{\lambda(N)}\ mod\ N^2)}\ mod\ N \\ &= m_1 + m_2 + ... + m_n. \end{aligned}$$

Thereafter, the *AG* decrypts message M_i using C_{2_i} as

$$\begin{aligned} C_{2_a}/e(C_{2_b}, g^{1/SK_{AG}}) &= w^{r_i}.M_i/e(g^{a.r_i}, g^{1/a}) \\ &= w^{r_i}.M_i/e(g, g)^{r_i} \\ &= w^{r_i}.M_i/w^{r_i} = M_i. \end{aligned}$$

Hence, only the legitimate *AG* can decrypt M_i using one exponential operation and one pairing operation per *SM*. Also, the *AG* chooses $t \in \mathbb{Z}^*$, and computes

$$\begin{aligned} C_{3_a} &= (g^c)^t = g^{c.t}, \\ C_{3_b} &= w^t.M, \text{and} \\ C_{3_c} &= w^t.M_{AG}, \end{aligned}$$

where $M = M_1||M_2||M_3||...||M_i$. Then, the AG computes $\sigma_{AG} = H(C_{3_i})^a$ and sends $(C_{3_i}, T_2, \sigma_{AG})$ to the CS, where $C_{3_i} = (C_{3_a}, C_{3_b}, C_{3_c})$ and T_2 is a timestamp.

(c) ***CS:*** Upon receiving the message, the *CS* computes $H^{'}(C_{3_i})$ and verifies the signature of *AG* as

$$e(g, \sigma_{AG}) \stackrel{?}{=} e(PK_{AG}, H^{'}(C_{3_i})).$$

If the verification is successful, the CS computes re-encryption of data for the *BC* and the *DCU*. Re-encryption performs one exponential and one pairing operations, which remains the system with low computation overhead. Also, only the *BC* and the *DCU* will be able to retrieve actual data from the messages they receive.

- Re-encryption for the BC with (C_{3_a}, C_{3_b})*:*

$$C_{4_a} = e(C_{3_a}, g^{b/c}) = e(g^{c.t}, g^{b/c}) = e(g, g)^{b.t} = w^{b.t}.$$

The *CS* computes $\sigma_{CS-BC} = H(C_{3_b}||C_{4_a})^c$ and sends $(C_{3_b}, C_{4_a}, T_3, \sigma_{CS-BC})$ to the *BC*.

- Re-encryption for the DCU with (C_{3_a}, C_{3_c})*:*

$$C_{4_b} = e(C_{3_a}, g^{d/c}) = e(g^{c.t}, g^{d/c}) = e(g, g)^{d.t} = w^{d.t}.$$

The *CS* computes $\sigma_{CS-DCU} = H(C_{3_c}||C_{4_b})^c$ and sends $(C_{3_c}, C_{4_b}, T_4, \sigma_{CS-DCU})$ to the *DCU*. T_3 and T_4 are timestamps when the data is sent to the *BC* and the *DCU*, respectively.

(d) ***BC:*** Upon receiving the message $(C_{3_b}, C_{4_a}, \sigma_{CS-BC})$, the *BC* computes $H^{'}(C_{3_b}||C_{4_a})$ and verifies the signature of the *CS* as

$$e(g, \sigma_{CS-BC}) \stackrel{?}{=} e(PK_{CS}, H^{'}(C_{3_b}||C_{4_a})).$$

If the verification is successful, the *BC* retrieves M by computing only one exponential operation as

$$C_{3_b}/(C_{4_a})^{1/b} = w_t.M/(w^{b.t})^{1/b} = M.$$

The *BC* retrieves the public key of ID_i, computes $(PK_{SM_i})^{SK_{BC}} = (g^{s_i})^b$ and retrieves message m_i as $m_i = M_i \oplus g^{s_i.b}$. Then, the *BC* uses m_i and ID_i to generate electricity bills. Hence, intermediate devices, such as *AG* and *CS* cannot extract the actual metering data.

(e) ***DCU:*** Upon receiving message $(C_{3_c}, C_{4_b}, \sigma_{CS-DCU})$, the *DCU* computes $H^{'}(C_{3_c}||C_{4_b})$ and verifies the signature of the *CS* as

$$e(g, \sigma_{CS-DCU}) \stackrel{?}{=} e(PK_{CS}, H^{'}(C_{3_c}||C_{4_b})).$$

If the verification is successful, the *DCU* computes M_{AG} by only one exponential operation as

$$C_{3_c}/(C_{4_b})^{1/d} = w_t.M_{AG}/(w^{d.t})^{1/d} = M_{AG}.$$

The *DCU* uses this aggregated demand (M_{AG}) in making decisions to balance the overall power supply-demand of the power.

C. Malicious Smart Metering Data Detection

In a real AMI network scenario, there can be adversaries that try to steal or alter the transmitted data, or inject malicious data to the transmitted packets over the network. Hence, it is an important and required task to detect malicious smart metering data from the aggregated data at the *AG* before forwarding the data further to other entities. In order to detect malicious smart metering data sent to the *AG*, we propose an algorithm based on binary search approach as follows:

Algorithm 1 *Malicious Smart Metering Data Detection*

Input: The *AG* receives a set of n-smart metering data as $SMD = \{MD_1, MD_2, MD_3, ..., MD_n\}$.
Output: Returns a set of malicious metering data MD_i, otherwise return True.

while $(e(g, \sigma_{SM_i}) \neq \prod_{i=1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i})))$ **do**
  $e(g, \sigma_{SM_i}) \stackrel{?}{=} \prod_{i=1}^{\lceil n/2 \rceil} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i}))$
  $e(g, \sigma_{SM_i}) \stackrel{?}{=} \prod_{i=\lceil n/2 \rceil+1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i}))$
  if $(n == 1 \;\&\&\; e(g, \sigma_{SM_i}) \neq \prod_{i=1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i})))$ **then**
  return $SMD = \{MD_i\}$ and malicious $SM = \{SM_i\}$.
if $(e(g, \sigma_{SM_i}) == \prod_{i=1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i})))$ **then**
return True.

The proposed algorithm detects malicious data from the aggregated data by verifying $(n == 1 \;\&\&\; e(g, \sigma_{SM_i}) \neq \prod_{i=1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i})))$. If the condition holds, the algorithm computes $SMD = \{MD_i\}$ and malicious $SM = \{SM_i\}$. At the end, this algorithm returns a set of malicious metering data MD_i if any, in $log\ n$ time. The *AG* removes the malicious data from the aggregated data, and then forward the legitimate and correct data to the other entities.

V. Security and Performance Analysis

In this section, we present security and performance analysis of the proposed scheme.

A. Security Analysis

This section presents the security properties achieved by the proposed scheme.

Property 1. *The proposed scheme provides mutual authentication between the SMs, AG, CS, DCU and BC.*

As presented in Section IV-A, each SM_i, *AG*, and *CS* generate and forward their signatures along with the messages to the *AG*, and *CS*, and (*BC* and *DCU*), respectively. Upon receiving the messages, the signature of the sender is always first verified. The receiving entity proceeds further only if the verification is successful. Hence, all senders are authenticated in the flow of information.

Property 2. *The proposed scheme provides confidentiality of the concentrated data from the users to the DCU.*

The *AG* collects encrypted metering data received from different smart meters and derives aggregated sum of data $M_{AG} = m_1 + m_2 + ... + m_n$ by performing a decryption. However, adversary $\mathcal{A}$ cannot obtain the sum because it does not know the private key $\lambda(N)$. The *AG* and the *CS* compute $C_{3_c} = w^t.M_{AG}$ and $C_{4_b} = w^{d.t}$, respectively, which are sent to the *DCU*. Upon receiving the message, the *DCU* extracts M_{AG} by computing $C_{3_c}/(C_{4_b})^{1/d}$. Adversary $\mathcal{A}$ cannot extract M_{AG}, as it does not have d key of the *DCU*.

Property 3. *The proposed scheme provides undeniability of data sent from the sender to the receiver.*

The signatures at the SM_i, *AG*, and *CS* are generated using their private keys, *i.e.*, s_i, a, and c, which are only known to themselves. Hence, no other entity including the adversary can generate the actual signatures. Therefore, the SM_i, *AG*, and *CS* cannot deny access after the data has been sent to the *AG*, *CS*, and (*BC*, and *DCU*), respectively, as the signatures serve as the undeniable evidence for the sent data.

Property 4. *The proposed scheme defeats MITM, replay and impersonation attacks over the network.*

Each SM_i sends encrypted metering data to the *AG* as $C_{1_i} = (g^{m_i}\ r_i^N\ mod\ N^2)$ and $C_{2_i} = (w^{r_i}.M, g^{a.r_i})$, where $M = m_i \oplus (PK_{BC})^{SK_{SM_i}}||ID_i = m_i \oplus (g^b)^{s_i}||ID_i$. Clearly, the adversary $\mathcal{A}$ performing MITM or the legitimate *AG* cannot retrieve the original individual message, as they do not know the private key of the *BC*. The adversary $\mathcal{A}$ cannot alter the transmitted data over the network, as the hash of each received message is verified (a part of signature verification). Hence, the protocol provides prevention against MITM attack.

Each message in the protocol is transmitted with a timestamp value. If $\mathcal{A}$ resends a previously sent message in the current session, the receiving entity discards the message, as it finds the condition does not hold by verifying $T_{receive} \leq T_{send} + T_{threshold}$, where $T_{receive}$, T_{send}, and $T_{threshold}$ are the receiving, sending, and threshold timestamp values, respectively.

If $\mathcal{A}$ tries to impersonate a sender's entity, it will not be successful as the signature of each sender entity is required

to be verified. The signatures of the SM_i, *AG*, and *CS* are verified as $e(g, \sigma_{SM_i}) = \prod_{j=1}^{n} e(PK_{SM_i}, H^{'}(C_{1_i}||C_{2_i}))$, $e(g, \sigma_{AG}) = e(PK_{AG}, H^{'}(C_{3_i}))$, $e(g, \sigma_{CS-BC}) = e(PK_{CS}, H^{'}(C_{3_b}||C_{4_a}))$, and $e(g, \sigma_{CS-DCU}) = e(PK_{CS}, H^{'}(C_{3_c}||C_{4_b}))$, respectively, at the *AG*, *CS*, *BC*, and *DCU*. Hence, $\mathcal{A}$ cannot successfully perform impersonation attacks.

Table I shows a comparison of security features with existing schemes, where our scheme achieves all four features.

B. Performance Analysis

This section presents the performance evaluation of our scheme in terms of computation, communication, and storage overheads, and execution time.

1) Computation overhead: In each of the smart metering concentration rounds, *SM*s generate their cipher data by performing random numbers generation, exponential and multiplication operations in L encryption function, XOR operations and hash operations, and send their data to the *AG*. The *AG* collects all the encrypted data received from different *SM*s and computes a sum of encrypted data using L decryption function. The *AG* generates random numbers and performs exponential and hash operations to forward the individual and concentrated metering data to the *CS*. The *CS* performs re-encryption over the received data and forwards the data to the *BC* and the *DCU*. Finally, the *BC* extracts individual metering data and generates the electricity bill. The *DCU* retrieves concentrated metering data and computes demand-supply check for making decisions in different smart grid applications, such as vehicle-to-grid, demand-response, load balancing, etc.

Table II summarizes the computations of the proposed scheme in terms of operations performed at different entities in the proposed AMI system architecture. Table III shows that the proposed scheme is more efficient than H. Li's scheme [12] in terms of the computation time for key generation and scheme execution phase. We compared our scheme with H. Li [12], as it is the only scheme that maintains three features listed in Table I. However, our scheme also maintains confidential data delivery at *BC* in addition to three features supported by [12].

2) Execution Time: We simulate the proposed scheme in Java on an Intel Core i3-4005U CPU 1.7GHz with Windows7 and 2GB RAM. The generation of a random number, scalar multiplication, and XOR take 0.69 ms, 0.039 ms, and 0.029 ms, respectively. We implement Java pairing-based cryptography (JPBC) library for a pairing operation, which is performed in 197 ms. Hash function SHA256 takes 4 ms. The exponential function, elliptic addition, and scalar addition took 2.1, 0.604, and 0.033 ms, respectively. The average operation times for 100 runs of Bilinear ElGamal homomorphic encryption and decryption was 3.1 and 8.7 ms, respectively [15]. The RSA encryption and decryption took 40 and 18 ms, respectively. Figure 4(a) shows a comparison of the execution times of our scheme and the scheme in [12] considering the number of smart meters from 1 to 1000. Our scheme outperforms and has execution times between 1.27 to 222.3 s, whereas the scheme in [12] has execution times between 1.78 to 1775.39 s.

3) Storage overhead: A metering data is periodically generated at each *SM* and is sent to the *AG*. In our simulation, we consider that each plaintext, ciphertext, and *SM*'s identity 128 bits long. The *SM* needs a buffer to store the encrypted data for each time instance and generates (C_{1_i}, C_{2_i}) data of (128×n, 384×n) bits for n-*SM*s. The *AG* stores n-*SM*s' data and requires 128×n bits of memory to keep the periodic instant data along with storing $(C_{3_a}, C_{3_b}, C_{3_c})$ data of (128, 256×n, 128) bits. The *CS* stores $(C_{3_a}, C_{3_b}, C_{3_c})$ data of (128, 256×n, 128) bits, whereas the *BC* and the *DCU* store (C_{3_b}, C_{4_a}) and (C_{3_c}, C_{4_b}) of (256×n, 128) and (128, 128) bits, respectively. Each entity also requires to store the received and computed hashes of 64 bits each, except the *SM* that requires only 64 bits of hash to store. Hence, the total number of bits required to be stored is 1344 + 1536×n bits (= 168 + 192×n bytes). In practice, a communication module of a typical *SM* has 4MB RAM and 8MB flash memory [16]. The storage overhead of

TABLE I: Comparison of Security Features

Scheme	Prevention of Attacks	Data Integrity	Privacy-Preserved Data Delivery to DCU	Confidential Data Delivery to BC
F. Li *et al.* [6]	Partial	No	Yes	No
F. Li *et al.* [7]	Yes	No	Yes	No
C. Efthymiou [8]	Partial	No	Partial	No
F. Garcia [9]	Partial	No	Yes	No
F. Li *et al.* [11]	Yes	No	Yes	No
H. Li *et al.* [12]	Yes	Yes	Yes	No
C. Li's PDA [13]	Partial	No	Yes	No
Our Scheme	Yes	Yes	Yes	Yes

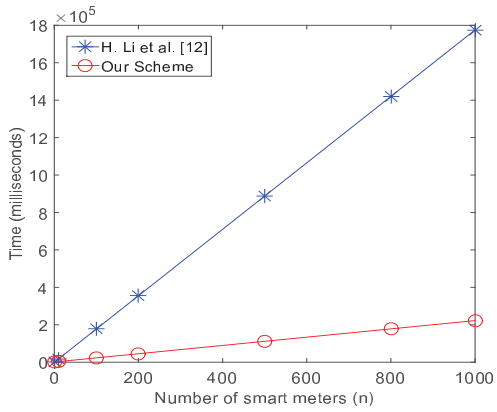
TABLE II: Operations Performed by the Proposed Scheme

Operations	At *SM*	At *AG*	At *CS*	At *BC*	At *DCU*
Random Numbers	$2n$	2	1	1	1
Exponential	$7n$	5	3	3	2
Multiplication	$2n$	$2n$	-	-	-
Pairing	-	n	3	1	1
XOR	n	-	-	n	-
Hash	n	n+1	3	1	1
L Function	n	1	-	-	-

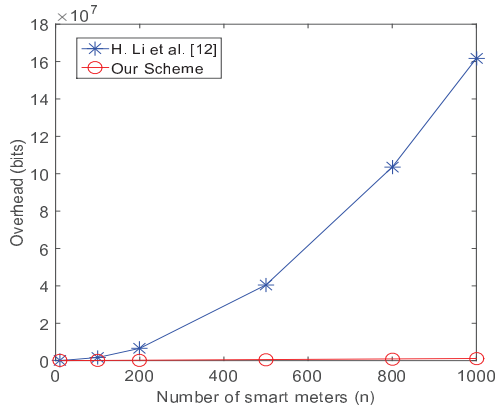
TABLE III: Time Complexity of H. Li *et al.* [12] and Proposed Schemes

Module	H. Li *et al.* [12]	Our Scheme
Key Generation	$(n+1)T_{rn} + (n+1)T_{ECmul}$ + nT_{hash}	$(2n+4)T_{rn}+(n+6)T_{exp}$
Scheme Execution	$(3n+1)T_{rn}+(6n+2)T_{ECmul}$ + $(17n+1)T_{hash}$ + $8nT_{pair}$ + $2nT_{exp}$ + $2nT_{add}$ + $(3n-1)T_{mul}$ + $2nT_{enc}$ + $2nT_{dec}$ + T_{L-fun}	T_{rn} + $(6n+7)T_{exp}$ +$4nT_{mul}$ + $(n+5)T_{pair}$ + $2nT_{xor}$ + $(2n+6)T_{hash}+T_{L-fun}$

T_{rn}: time for generating a random number, T_{exp}: time for an exponential operation, T_{ECmul}: elliptic curve point multiplication time, T_{mul}: time for a multiplication operation, T_{pair}: time for a pairing operation, T_{xor}: time for an XOR, T_{add}: time for an addition, T_{hash}: time for a hash operation, T_{enc}: encryption time, T_{dec}: decryption time, and T_{L-fun}: time to execute a L-function. Here, n is the number of smart meters.



(a) Execution time.



(b) Communication overhead.

Fig. 4: Comparison of execution times and communication overheads.

our scheme is far below the capability of the current *SM*s.

4) Communication overhead: The communication overhead is defined as the total number of bits transmitted over the network during a protocol run. The *SM*s, *AG*, and *CS* generate the communication overhead of $640{\times}n$, $384 + 256{\times}n$, and $640 + 256{\times}n$ bits, respectively. Hence, the total communication overhead generated by our scheme is $1024 + 1152{\times}n$ bits. Figure 4(b) illustrates a graph for generated overheads when the number of *SM*s are 10, 100, 200, 500, 800, and 1000. The figure shows that our scheme is much efficient than the scheme in [12], and lowers the overhead by 43.19% to 99.29% when the number of *SM*s are 10 and 1000, respectively.

VI. Conclusion

We proposed a secure and privacy-preserving scheme for aggregating metering data in the advanced metering infrastructure network. The scheme aggregates the sum of metering data without revealing the actual metering data. Different from existing schemes, the metering data and the identity of the smart meter (and household owner) are only revealed to the billing center for billing purposes, whereas the data concentrator unit receives the aggregated metering data for grid control purposes. The scheme achieves low overheads because of using efficient and short signatures, XOR and hash operations, as well as transmitting less bits compacted by pairing and exponential operations. We also presented an algorithm for detecting malicious metering data that ensures the delivery of correct and accurate metering data in a secure manner. The proposed scheme provides security to the system against man-in-the-middle, replay, and impersonation attacks as well as from deniability. Time and space analysis shows that the scheme is efficient and generates manageable overhead, even when a large number of smart meters are deployed in the network. Therefore, the proposed scheme is suitable to use in the metering infrastructure network.

Acknowledgments

This research was partially funded by the MSIP, Korea, under "ICTCCP" (IITP-2015-R0346-15-1007) supervised by the IITP, by the KEIT, Korea, under "Global Advanced Technology Center" (10053204), and by the NRF, Korea, under "Basic Science Research Program" (NRF-2015R1C1A1A01053788).

References

[1] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18-28, 2010.

[2] J. C. L. Chan and D. S. Wong, "A survey on security assessment of metering infrastructure in smart grid systems," in *Proc. IEEE SoutheastCon*, 2015, pp. 1-4.

[3] H. Crijns, Open Smart Grid Protocol (OSGP), Updated Specification Released, Press Release - Jul. 15, 2015. [Online]. Available: http://www.osgp.org/press-release-15-07-2015.

[4] K. Kursawe and C. Peters, "Structural weaknesses in the open smart grid protocol," *IACR*, Cryptology ePrint Archive: Report 2015/088, pp. 1-19, 2015. [Online]. Available: https://eprint.iacr.org/2015/088.pdf.

[5] P. Jovanovic and S. Neves, "Dumb crypto in smart grids: practical cryptanalysis of the open smart grid protocol," *IACR*, Report 2015/428, pp. 1-20, 2015. [Online]. Available: http://eprint.iacr.org/2015/428.

[6] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proc. IEEE SmartGridComm*, 2010, pp. 327-332.

[7] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *Int. J. Security and Networks*, vol. 6, no. 1, pp. 28-39, 2011.

[8] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proc. SmartGridComm*, 2010, pp. 238-243.

[9] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proc. International Workshop on Security and Trust Management*, 2010, pp. 226-238.

[10] A. Rial and G. Danezis, Privacy-preserving smart metering, Microsoft Research, Tech. Rep. MSR-TR-2010-150, 2010.

[11] F. Li and B. Luo, "Preserving data integrity for smart grid data aggregation," in *Proc. IEEE SmartGridComm*, 2012, pp. 366-371.

[12] H. Li, X. Lin, H. Yang, X. Liang, R. Lu and X. Shen, "EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key Evolution in Smart Grid," *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053-2064, 2014.

[13] C. Li, R. Lu, H. Li, L. Chen, and J. Chen, "PDA: a privacy-preserving dual-functional aggregation scheme for smart grid communications," *Security Comm. Networks*, vol. 8, pp. 2494-2506, 2015.

[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, 1999, pp. 223-238.

[15] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. on Info. and System Security*, vol. 9, no. 1, pp. 1-30, 2006.

[16] Communications Module for Electricity Meters, SilverSpring Networks, 2013. [Online]. Available: http://www.silverspringnet.com/pdfs/SilverSpring-Datasheet-Communications-Modules.pdf.