Appl. Sci. **2015**, *5*, 1503-1527; doi:10.3390/app5041503

OPEN ACCESS

applied sciences

ISSN 2076-3417

www.mdpi.com/journal/applsci

Article

Privacy-Preserving Smart Metering with Authentication in a Smart Grid

Jun Beom Hur [1], Dong Young Koo [2] and Young Joo Shin [3,*]

[1] Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu 02841, Korea; E-Mail: jbhur@korea.ac.kr

[2] Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu 34141, Korea; E-Mail: dykoo@nslab.kaist.ac.kr

[3] National Security Research Institute, 1559 Yuseongdae-ro, Yuseong-gu 34044, Korea

* Author to whom correspondence should be addressed; E-Mail: s.youngjoo@gmail.com; Tel.: +82-42-870-2306.

Academic Editor: Minho Shin

Received: 28 September 2015 / Accepted: 24 November 2015 / Published: 1 December 2015

Abstract: The traditional security objectives of smart grids have been availability, integrity, and confidentiality. However, as the grids incorporate smart metering and load management, user and corporate privacy is increasingly becoming an issue in smart grid networks. Although transmitting current power consumption levels to the supplier or utility from each smart meter at short intervals has an advantage for the electricity supplier's planning and management purposes, it threatens user privacy by disclosing fine-grained consumption data and usage behavior to utility providers. In this study, we propose a distributed incremental data aggregation scheme where all smart meters on an aggregation path are involved in routing the data from the source meter to the collection unit. User privacy is preserved by symmetric homomorphic encryption, which allows smart meters to participate in the aggregation without seeing any intermediate or final result. Aggregated data is further integrated with an aggregate signature to achieve data integrity and smart meter authentication in such a way that dishonest or fake smart meters cannot falsify data en route. Only the collection unit can obtain the aggregated data and verify its integrity while the individual plain data are not exposed to the collection unit. Therefore, user privacy and security are improved for the smart metering in a smart grid network.

Keywords: smart grid; smart metering; privacy; security; data aggregation

1. Introduction

Smart grids are envisioned as a next generation approach to delivering electricity to millions of households by stakeholders. Smart grids have incorporated computation and communication technology into traditional power grids, allowing them to become smart and connected. Processing and storage units are embedded in traditional electricity meters and communicate with electrical appliances at home as well as the generation and management facilities of electric utilities, providing smart grids with great connectivity. With the intelligent, networked smart meters, smart grids enable the instant monitoring of power delivery and consumption information, the subscription of power usage, remote operation, advanced demand and outage management, and usage management.

The rollout of smart meters has already begun, and this new technology has been rapidly adopted in countries throughout the world. For example, the United States and the European Union currently promote the deployment of smart grids. However, despite the numerous advantages offered by smart grids, security and privacy concerns started to arise [1–3]. Smart meters are expected to automatically provide accurate readings at requested time intervals to the utility company or electricity distribution network. Frequent and detailed meter readings can be used to optimize the grid but can also reveal behavioral patterns and leak personal information. Fine-grained electricity consumption profiles differ between devices. Thus, given a sufficient resolution on the time axis, electric utilities could know the daily energy usage patterns of a household and even go so far as to deduct whether the inhabitants are at home, when they get up and go to bed, and what kind of devices are being used at which time, among other things [2–4]. User privacy concerns have already jeopardized the mandatory deployment of smart meters in the Netherlands, leading to a deployment deadlock in April 2009.

Recently, several studies have proposed solutions to this problem in smart grids [3–8]. One promising technical solution to protect user privacy is anonymizing each packet of high-frequency metering data by aggregating them at multiple levels (e.g., neighborhood, subdivision, district, and city) via privacy-protecting cryptographic techniques such as homomorphic encryption [9,10]. In this approach, the data collection unit can obtain sums of the measurements of all the connected smart meters without learning the individual measurements. Additionally, the smart meters involved in the aggregation cannot learn any plain measurements from the other meters. Thus, customer privacy can be protected.

However, most of the well-known homomorphic encryption protocols are malleable [9–11]. Thus, given the ciphertext and public key, an adversary could generate another cipher which decrypts to another meaningful plaintext in the same domain as the original plaintext. As a result, a dishonest or fake smart meter could falsify data, leading to inaccurate aggregation results. Therefore, it is of great importance to provide efficient and secure data aggregation which guarantees user privacy, authentication, and aggregated data integrity in smart grids.

One straightforward solution to this problem is enabling each smart meter to sign its metering data with a private key so that the collection unit can verify the integrity of data from individual meters and authenticate the source meters with their public keys. However, this approach allows the collection unit

to learn the plain meter readings of each smart meter during the verification phase, which violates the privacy of users. On the other hand, when the measurements of smart meters are aggregated using the previous homomorphic encryptions [9–11] in order to protect user privacy, a signature scheme cannot enable the collection unit to verify the authenticity of plain (aggregated or individual) data. Therefore, it is challenging for a verifier, that is, a collection unit, to verify the integrity of aggregated metering data from multiple smart meters in a smart grid network without accessing the individual plain data. This is a problem we will attempt to solve in this study.

1.1. Related Work

The damage that smart meters can cause to user privacy has previously been studied from both a technical and a legal perspective [2,12]. These studies propose the enforcement of privacy based on organizational means, codes of conduct, and regulations, subject to current legislation. Of the technical solutions that have been offered, several recent schemes have suggested protecting user privacy by aggregating individual metering data at a range of levels [3,5,7,8] or by anonymizing high-frequency metering data through the use of pseudonymous identities [4]. However, the importance of data integrity has not drawn a great deal of attention in previous smart grid security literature, despite it being one of the most important security requirements (along with availability and confidentiality) in smart grid cyber security [13].

Bohli *et al.* [8] introduced a smart metering privacy model to measure the degree of privacy that a smart metering application can provide, and proposed a privacy-preserving aggregation scheme. The basic principle of the scheme is that each electricity meter takes its current reading and adds a random value drawn from a known distribution with a known variance σ^2 and expectation μ. The sent value is then distributed around the sum of the actual measurement and the expectation of the distribution. However, since the measurements are randomized, there is a trade-off between the precision of individual and aggregated readings. With small aggregation groups, it is not easy to achieve the desired level of privacy, which is a major drawback with regard to the practical deployment of this scheme.

Garcia *et al.* [3] suggested a multiparty computation protocol that allows multiple smart meters in a neighborhood to compute a partial aggregation of their data without disclosing their individual measurements by taking advantage of Paillier's additive homomorphic encryption [9]. The partially aggregated data of every meter is then sent to the local substation, which is the collection unit in the neighborhood, and fully aggregated. This scheme requires that each meter compute $O(N)$ encryptions, and that additional $O(N^2)$ communications between each meter and the local substation are made before sending the fully aggregated result to the electricity utility, where N is the number of smart meters in the neighborhood. Thus, it lacks efficiency in terms of computation and communication overheads. It is also vulnerable to data forgery at the local substation due to the malleability of homomorphic encryption.

Li *et al.* [5,6] proposed a distributed in-network aggregation scheme for smart grid networks. Data aggregation is performed at all smart meters involved in routing the data from the source meter to the collection unit using Paillier's homomorphic encryption [9]. When $\{m\}_K$ represents the encryption of measurement m under the collection unit's public key K, each intermediate smart meter s_j on the routing path performs $\{m_i\}_K \cdot \{m_j\}_K = \{m_i + m_j\}_K$ and forwards it to the next smart meter,

where m_j is the measurement to be sent and $\{m_i\}_K$ is the received ciphertext from the previous smart meter s_i. The collection unit can then decrypt the final ciphertext with its private key, and obtain the total sum of the measurements from all of the smart meters on the routing path without knowing individual plain measurements. This scheme is efficient in terms of computation and communication cost because the in-network aggregation is integrated into the data routing process. However, this scheme requires additional cryptographic mechanisms to allow the collection unit to verify the integrity of the aggregated data and authenticate the participating meters because intermediate malicious meters or outside adversaries can easily forge intermediate aggregation data en route.

Seo *et al.* [14] proposed a metering data aggregation scheme using Paillier's homomorphic encryption [9]. This scheme suggested an aggregated data verification mechanism using a hash tree, but it is also vulnerable to active attacks by intermediate malicious meters because it grants the task of verification to each meter rather than to the collection unit. Garofalakis *et al.* [15] and Castelluccia *et al.* [16] have also proposed verifiable in-network data aggregation schemes in sensor network literature; however, their schemes have some privacy drawbacks. In Garofalakis *et al.*'s scheme, intermediate aggregators can obtain individual raw data from its child nodes during in-network aggregation. In Castelluccia *et al.*'s scheme, the final data collection unit can generate all of the symmetric keys shared with each node. Thus, if the collection unit captures any of the encrypted data reported from the leaf nodes, it can obtain the raw data in plaintext by deriving the symmetric key and decrypting the ciphertext with that key. Therefore, these two schemes cannot be directly adapted to privacy-preserving smart metering since they violate user privacy.

When multiple metering data are aggregated using homomorphic encryption, it is impossible to verify the integrity of each measurement and aggregated result without violating user privacy because the verification process needs individual metering data which has not been aggregated. In order for the collection unit to verify the aggregated data, every smart meter s_i should send its $\{m_i\}_K$ individually together with a signature for it, where K is the public key of the collection unit. Although this solution ensures the integrity of metering data, it allows the collection unit to obtain the plain metering data, which violates user privacy. Alternatively, a homomorphic signature algorithm [17] is capable of evaluating multivariate polynomials on signed data. Given the public key and a signed data set, it allows users to delegate computations (e.g., produce a signature on the mean, standard deviation, and other statistics of the signed data) to an untrusted third party while ensuring integrity. Even though the homomorphic signature algorithm seems to enable the collection unit to ensure the integrity of the aggregated data by verifying the homomorphic signature generated for it, the homomorphic signature algorithm cannot be adopted by a smart grid environment. This is because, in the homomorphic signature setting, the delegate can produce a signature for the statistics (such as the sum) of a set of data signed by the same signer; while smart metering applications require a solution for the collection unit to ensure the integrity of the statistics of a set of data signed by different signers or smart meters. Therefore, designing an efficient and secure metering scheme that guarantees sender authentication and the integrity of aggregated data while preserving user privacy is essential.

Recently, several other smart metering schemes were proposed [18–24]. However, the schemes are all constructed on the basis of public key based homomorphic encryption such as Paillier's encryption, which may incur much higher computational overhead on resource-limited meters compared to the

symmetric approach in smart grid networks, or do not guarantee data integrity and sender authentication. Fengjun *et al.* [25] and Rottondi *et al.* [26] proposed data aggregation schemes which ensure data integrity, but they cannot preserve user privacy during smart metering.

1.2. Contribution

To fulfill the somewhat contradictory requirement of enabling the collection unit to verify the integrity of aggregated meter readings without revealing any individual measurements, we propose a novel aggregation scheme with authentication capability for smart metering. In the proposed scheme, metering data on the routing path is aggregated by exploiting a symmetric homomorphic encryption in such a way that the collection unit can obtain the aggregated data without individual plain measurements being revealed to it or intermediate smart meters en route. The collection unit can verify the integrity of the aggregated measurements by adopting an identity-based sequential aggregate signature mechanism. The identity-based sequential aggregate signature scheme enables signers (that is, smart meters) to attest to their different readings while permitting savings on bandwidth and storage in comparison to public-key solutions. Signatures are aggregated one-by-one as the aggregate-so-far moves along the path, as is natural in the routing-based data aggregation application we consider. In addition, the verification process does not reveal plain metering data to the collection unit since individual data is obfuscated from the collection unit's view.

Thus, the proposed scheme is secure against any active attacks such as false data injection or data modification attacks launched by outside adversaries or intermediate compromised smart meters on the routing path. In addition, the proposed scheme enables the collection unit to obtain aggregated data, verify its integrity, and authenticate the participating smart meters without any knowledge of individual plain measurements. Therefore, the proposed scheme is also secure against passive attacks such as eavesdropping by a curious collection unit or smart meters in the smart grid network, which preserves user privacy. As opposed to the previous schemes, the proposed scheme enables the collection unit to verify integrity of the aggregated data as a whole (rather than independent verification for each individual measurement) by exploiting the aggregate signature. The efficiency analysis and experimental result show that the proposed scheme is more efficient than the previous schemes [3,5,6,21] while still guaranteeing both user privacy and aggregated data integrity.

1.3. Organization

The remainder of the paper is organized as follows. Section 2 describes the smart grid system architecture and security requirements. Section 3 reviews the cryptographic background and protocols. In Section 4, we propose our construction. We analyze the efficiency and security of the proposed scheme in Sections 5 and 6, respectively. In Section 7, we conclude the paper.

2. Smart Grid Architecture

In this section, we describe the smart grid architecture and define its security model.

2.1. System Description and Assumptions

Figure 1 shows the architecture of the smart grid network. The architecture consists of the following system entities: smart meters, a concentrator, and central management facilities. Smart meters deployed in houses frequently transmit their meter readings to an electric utility for power transmission and distribution network control purposes through the communication network. Smart meters in the neighborhood communicate with each other and a concentrator, which is the data collection unit, through a wireless mesh network. The concentrator further communicates with the central management facilities such as the electric utility and grid operator through wired communication, and reliably reports the aggregated result of the meter readings in the neighborhood to the facilities.

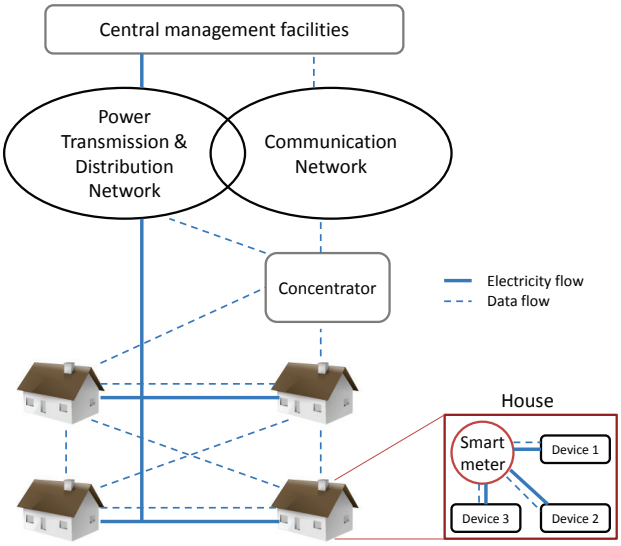

Figure 1. Smart grid architecture.

We assume that the central management facilities do not need to know which smart meter generates specific data. It is only interested in the aggregated measurements (that is, the sum of the current electricity consumption of individual smart meters) in some specified neighborhood over a given time period (e.g., every few minutes or hours) and the smart meters these metering data are associated with [3–5,8]; it is not interested in the current meter readings of any individual customer (Our main target environment is one where the metering data need to be reported frequently over relatively short time periods (e.g., every few minutes or hours) for grid network management or power distribution control rather than real-time pricing for billing purposes. For billing purposes, central management facilities may require the aggregate data from individual meters at comparably longer intervals (e.g., daily or monthly). User privacy would not be violated in this case because outside entities would not have access to any useful information from the measurement on the time axis. Therefore, real-time pricing is not a concern in this study.). As in Li *et al.*'s scheme [5,6], data from individual meters in the neighborhood are delivered to the concentrator following aggregation paths, as shown in Figure 2, and it is assumed that the smart meters do not collude with each other in our data aggregation model. We also assume that the concentrator is honest-but-curious. That is, it will honestly execute the assigned

tasks and will not collude with other smart meters in the system; however, it would like to learn as much information about individual metering data as possible.

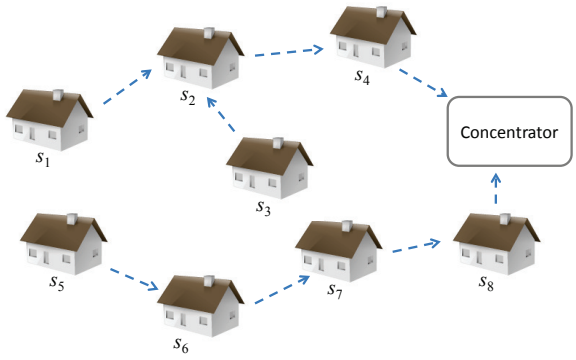

Figure 2. Data aggregation path.

2.2. Threat Model and Security Requirements

In smart grid networks, we consider both passive attacks (e.g., eavesdropping) and active attacks (e.g., data manipulation or false data injection) by inside and outside adversaries. When defending against these attacks, the following security properties are required for secure smart metering.

1. Data confidentiality (user privacy): Individual power usage of a smart meter is considered the private information of the owner. Thus, it should not be revealed to the concentrator or other meters on the routing path during data aggregation and delivery, and passive attacks such as eavesdropping should thus be defended against.
2. Data integrity: Dishonest or compromised smart meters in the network could manipulate the intermediate metering data during aggregation, causing inaccurate aggregation results. Thus, manipulation of the aggregate by active inside attack from the compromised meters should be detected by the concentrator.
3. Sender authentication: Defending against any active outside attacks, such as false data injection attack by outside adversaries, the concentrator should be able to ensure the authenticity of the smart meters' identities on the routing path that have contributed to the data aggregation.

3. Preliminaries and Definition

3.1. Cryptographic Background

3.1.1. Notations

If X and Y are strings, then $X||Y$ denotes the concatenation of X and Y. If S is a set, then $s \xleftarrow{\$} S$ denotes that s is selected uniformly at random from S. For a probabilistic algorithm $\mathcal{A}$, $x \xleftarrow{\$} \mathcal{A}$ assigns the output of $\mathcal{A}$ to the variable x. If $\mathcal{A}$ is deterministic, we drop the dollar sign above the arrow. All algorithms considered in this study are possibly randomized unless indicated otherwise.

3.1.2. Bilinear Pairings

Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be three cyclic groups of prime order p. Let u and v be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Bilinear map e is the map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties.

1. Bilinearity: For all $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p^*$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
2. Non-degeneracy: $e(u, v) \neq 1$.
3. Computability: There is an efficient algorithm to compute $e(g_1, g_2)$ for any $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.

Like many pairing-based cryptographic protocols, our protocol uses a special form of bilinear map called a symmetric pairing where $\mathbb{G}_1 = \mathbb{G}_2$. For the remainder of the paper, all bilinear pairings are symmetric, and we denote $\mathbb{G}_1 = \mathbb{G}_2$ by $\mathbb{G}$.

Weil pairing [27] or Tate pairing [28] on elliptic curves can be used as an efficiently computable non-degenerate bilinear map.

3.1.3. Computational Diffie-Hellman Problem

We first recall the well-known computational Diffie-Hellman (CDH) problem in the group $\mathbb{G}$ of prime order p. For a generator $g \in \mathbb{G}$, we define the CDH-advantage of algorithm $\mathcal{A}$ as

$$\mathrm{Adv}^{CDH}(\mathcal{A}) \stackrel{def}{=} \Pr[C = g^{ab} : a, b \stackrel{\$}{\leftarrow} \mathbb{Z}_p; C \stackrel{\$}{\leftarrow} \mathcal{A}(g, g^a, g^b)].$$

3.1.4. Identity-based Sequential Aggregate Signature (IBSAS) Problem

We introduce the CDH-type computational problem IBSAS by recapitulating the definition in [29], on which the security of the identity-based sequential aggregation scheme is based.

For all $a_1, b_1, a_2, b_2 \in \mathbb{Z}_p$ and generator $g \in \mathbb{G}$, the associated oracle $\mathcal{O}^{IBSAS}_{g,a_1,b_1,a_2,b_2}(\cdot)$ taking as input $m \in \mathbb{Z}_p$ is defined as

$$\begin{aligned} &\text{Oracle } \mathcal{O}^{IBSAS}_{g,a_1,b_1,a_2,b_2}(m) \\ &\quad r, x \stackrel{\$}{\leftarrow} \mathbb{Z}_p \\ &\quad \text{Return } (g^{rx} g^{a_1 b_1} g^{m a_2 b_2}, g^r, g^x). \end{aligned}$$

We then define the $IBSAS$-advantage of an algorithm $\mathcal{A}$ as

$$\begin{aligned} \mathrm{Adv}^{IBSAS}(\mathcal{A}) \stackrel{def}{=} \Pr[C = (m', g^{r'x'} g^{a_1 b_1} g^{m a_2 b_2}, g^{r'}, g^{x'}) : \\ a_1, b_1, a_2, b_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_p; \\ C \stackrel{\$}{\leftarrow} \mathcal{A}^{\mathcal{O}^{IBSAS}_{g,a_1,b_1,a_2,b_2}(\cdot)}(g, g^{a_1}, g^{b_1}, g^{a_2}, g^{b_2})], \end{aligned}$$

where we require that the oracle was not queried about $m' \in \mathbb{Z}_p$ by $\mathcal{A}$. [29] showed that the IBSAS problem is hard in the generic bilinear group model.

3.2. Cryptographic Protocol

3.2.1. Homomorphic Encryption

Homomorphic encryption represents a group of semantically secure encryption functions that allow certain algebraic operations on the plaintext to be performed directly on the ciphertext. Homomorphic encryption is useful in scenarios where someone who does not have a decryption key needs to perform arithmetic operations on a set of ciphertexts. Thus, homomorphic encryption is usually used for privacy-preserving operations (e.g., voting) in which operations are performed but operands are not disclosed. The most common definition is as follows.

Let $\{\cdot\}_k$ denote a probabilistic encryption scheme under key k. Let M denote the message space and C the ciphertext space. An encryption scheme is said to be homomorphic if for given $\{m_1\}_{k1} = c_1$ and $\{m_2\}_{k2} = c_2$, there exists a key k such that

$$\forall m_1, m_2 \in M, c_1 \odot_C c_2 = \{m_1 \odot_M m_2\}_k,$$

for some operators $\odot_M$ in M and $\odot_C$ in C. If $(M, \odot_M)$ and $(C, \odot_C)$ are groups, we have a group homomorphism. We say a scheme is additively homomorphic if we consider addition operators, and multiplicatively homomorphic if we consider multiplication operators.

RSA, El Gamal [30], Paillier [9,11], Naccache-Stern [31], Boneh-Goh-Nissm [10] are the well-known homomorphic encryption schemes. In particular, Paillier's encryption has been adopted in many current metering data aggregation schemes [3,5,6] in smart grid networks due to its additive homomorphic property: $\{m_1\}_k \cdot \{m_2\}_k = \{m_1 + m_2\}_k$.

3.2.2. Aggregate Signatures

Aggregate signatures [29,32–34] allow multiple signers to sign different messages while keeping the total signature size constant. Each signer forwarding a message adds its signature to the label of the next signer on the advertised route, so that route authenticity can be verified upon receipt of the aggregate. Thus, aggregate signatures can be used to prevent unauthorized parties from extending the path and enables signers to sign their own messages.

However, solutions based on public key infrastructure (PKI) [32–34] incur overhead in the distribution of the public key and certificate of each user to all other users. Boldyreva *et al.* [29,35] proposed identity-based sequential aggregate signature schemes for routing-based applications (Recently, Hwang *et al.* [36] has proven that the assumption on which the aggregate signature scheme proposed in [35] was based is false and that the scheme is universally forgeable, *i.e.*, anyone can generate forged signatures on any message of its choice.). In identity-based cryptography, an arbitrary identity string acts as a user's public key and verifying a signature only requires knowledge of a sender's identity. The overhead associated with obtaining and storing identity-based keys is typically much smaller than that of obtaining traditional public keys and certificates for the signers, since the identity-based setting eliminates it. Thus, an identity-based solution can offer a superior alternative to previous PKI-based approaches.

An identity-based sequential aggregate signature suits the verification of metering data aggregates in smart grid networks because signatures are aggregated one-by-one as the aggregate-so-far moves along

the path, which is natural in the routing-based applications we consider. It simplifies key management and reduces storage overhead for smart meters, while providing a way for the concentrator to verify the integrity of the aggregate and authenticate the participating meters on the routing path. Thus, we will adopt an identity-based sequential aggregate signature scheme [29] in our construction.

4. Privacy-Preserving Smart Metering with Authentication Capability

In this section, we propose a privacy-preserving smart metering scheme. The proposed scheme consists of aggregation path generation and path key establishment followed by data encryption (using symmetric homomorphic encryption [37]) and signature generation (using an identity-based sequential aggregate signature algorithm [29]). The proposed scheme allows the concentrator to aggregate metering data from smart meters on the routing path and verify the authenticity of the aggregated result. Thus, both user privacy and data integrity are achieved during the reporting of metering data.

4.1. Primitive Functions

4.1.1. Symmetric Homomorphic Encryption

To tackle the authentication issue introduced by previous aggregation schemes that use Paillier's asymmetric homomorphic encryption [9] in which signature generation and verification coud not be integrated into an aggregation scheme without revealing individual (*i.e.*, not aggregated) metering data to the concentrator, we exploit symmetric additively homomorphic encryption [37]. This encryption algorithm is a slightly modified version of a stream cipher, replacing the exclusive-OR operation with modular addition. It is provably secure and allows the efficient aggregation of encrypted data. It is composed of the following two algorithms: Enc and Dec.

1. **Encryption.** Given integer message m and randomly generated key stream k, for $m, k \in [0, M-1]$ where M is a large integer, the algorithm computes $c = Enc_k(m) = m + k \pmod{M}$.
2. **Decryption.** Given ciphertext c and key k, the algorithm computes $Dec_k(c) = c - k \pmod{M}$.

Additive homomorphism can be achieved simply. Let $c_1 = Enc_{k_1}(m_1)$ and $c_2 = Enc_{k_2}(m_2)$. Then, $c_1 + c_2 = Enc_{k_1+k_2}(m_1 + m_2) \pmod{M}$. In practice, if n different ciphertexts are added, M should be larger than $\sum_{i=1}^{n} c_i$ for correct decryption. It is important to note that each ciphertext that needs to be added can be encrypted under different keys, as opposed to Paillier's encryption.

Theorem 1. *The above encryption scheme is perfectly secure (A cryptosystem is perfect secure if the* a posteriori *probability that the plaintext is x, given that the ciphertext y is observed, is identical to the* a priori *probability that the plaintext is x.).*

Proof. The homomorphic encryption described above is a simple generalization of the stream cipher, where bits are replaced by integers modulo M. In terms of security, it has exactly the same properties as a stream cipher, that is, perfect secrecy if and only if the keystream is random, of the same length as the plaintext, and used only once. Thus, its security can be proven using a similar proof. The security relies

on two features: the key changes from one message to another, and all of the operations are performed modulo integer M.

Intuitively, if we have $\Pr[Enc_k(m_1) = c] = \Pr[Enc_k(m_2) = c]$ for every m_1, m_2 in the plaintext space, the encryption is perfectly secure because every plaintext has the same probability of being encrypted and results in a given ciphertext. The formal security proof is given in [37].

4.1.2. Identity-Based Sequential Aggregate Signature

The aggregate signature scheme consists of the following four algorithms: $Setup$, $KeyDer$, $Sign$, and Vf.

1. **Setup.** The trusted key generation center (KGC) initially runs the $Setup$ algorithm to generate the master public key mpk and master secret key msk.
2. **Key Derivation.** The KGC runs the $KeyDer$ on the input of msk and user identity $ID \in \{0,1\}^*$, and outputs the private key sk_{ID} for the user's ID.
3. **Sign.** A signer with ID runs the $Sign$ algorithm on the input of sk_{ID}, message $m \in \{0,1\}^*$, aggregate-so-far σ, and list of identity-message pairs $L_{i-1} = ((ID_1, m_1), \ldots, (ID_{i-1}, m_{i-1}))$, and returns new aggregate signature σ'; or $\perp$ if any of the inputs are invalid.
4. **Verification.** The verifier runs the Vf algorithm on the input of mpk, list of identity-message pairs $L_n = ((ID_1, m_1), \ldots, (ID_n, m_n))$, and aggregate signature σ, and returns 1 or 0.

For consistency, we require that $Vf(L_n, \sigma_n) = 1$ for all $n \in \mathbb{N}$ and all $\{(ID_i, m_i) | 1 \leq i \leq n, ID_i \in \{0,1\}^*, m_i \in \{0,1\}^*\}$ where the probability is over the experiment

$$
\begin{aligned}
&(mpk, msk) \xleftarrow{\$} Setup \\
&\text{For all } i = 1, \ldots, n \text{ do} \\
&\qquad sk_{ID_i} \xleftarrow{\$} KeyDer(msk, ID_i) \\
&\sigma_0, L_0 \leftarrow \varepsilon \\
&\text{For } i = 1, \ldots, n \text{ do} \\
&\qquad \sigma_i \xleftarrow{\$} Sign(sk_{ID_i}, m_i, L_{i-1}, \sigma_{i-1}) \\
&\qquad L_i \leftarrow ((ID_1, m_1), \ldots, (ID_i, m_i)).
\end{aligned}
$$

The proposed scheme is constructed based on the primitive identity-based aggregate signature protocol in [29]. The following theorem show that the signing and verification algorithms in the proposed scheme are secure in the random oracle model if the IBSAS problem is difficult relative to its associated bilinear-group generator $\mathcal{G}$.

Theorem 2. *Let $\mathcal{G}$ be a bilinear-group generation algorithm and AS be the associated identity-based sequential aggregate signature scheme given by Sections 4.2.3 and 4.2.4. Suppose there exists a forger F against AS in the random oracle model that makes at most q_{h_1} queries to its hash oracles, at most q_k queries to its key derivation oracle, at most q_s queries to its signing oracle, and outputs lists of length at most n_{max}. There is then an algorithm B for the IBSAS problem relative to $\mathcal{G}$ such that*

$$Adv_{AS}^{IBSAS}(F) \leq e(n_{max}(q_s+1)+q_k) \cdot Adv_{\mathcal{G}}^{IBSAS}(B) + q_h^2/l_{min}(\mathcal{G}),$$

where $l_{min}(\mathcal{G})$ is the minimum bit-length of the order p of a bilinear group output by $\mathcal{G}$.

Proof. The proof is given in [29].

4.2. Scheme Construction

In this section, we construct our metering data aggregation scheme with authentication capability.

4.2.1. System Setup

In the initial system setup phase, system security parameters and users' secret keys are set up. Aggregation paths are then set up as follows.

Key Setup

The $Setup$ algorithm chooses random generator $g \in \mathbb{G}$ and random $\alpha_1, \alpha_2 \in \mathbb{Z}_p$, and cryptographic hash functions $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p$. It then returns $(g, g^{\alpha_1}, g^{\alpha_2}, H_1, H_2, H_3)$ as mpk, and (α_1, α_2) as msk.

The KGC then runs the $KeyDer$ algorithm on the input of msk and $ID \in \{0,1\}^*$, and returns secret key $sk_{ID} = (H_1(ID)^{\alpha_1}, H_2(ID)^{\alpha_2})$ to the smart meter associated with identity ID. Henceforth, ID_i denotes the identity of a smart meter s_i.

Aggregation Path Setup

The aggregation path is constructed so that all of the smart meters in a neighborhood are covered. Figure 2 shows an example of the aggregation path constructed in a neighborhood. If we consider the smart meter network as a connected graph where smart meters are vertices and wireless links between any two meters are edges, the aggregation tree is constructed as a spanning tree of the graph rooted at the concentrator. The aggregation task is performed on the basis of each aggregation path. All or a subset of the smart meters on the aggregation path participate in the aggregation task. The construction of an efficient aggregation path is beyond the scope of this study.

Without a loss of generality, we will describe data encryption/decryption and signature generation/verification algorithms below under the assumption that a set of smart meters are deployed on the aggregation path to the concentrator in the order $s_1, \ldots, s_n$.

4.2.2. Data Encryption

After system setup, smart meters and the concentrator establish a path key, which is unique for each aggregation path. Each smart meter then delivers encrypted metering data to the concentrator following the aggregation path. A new path key is established whenever the aggregation path is changed (e.g., in case of the change of a smart grid network topology).

Path Key Establishment

When a set of smart meters are deployed on the aggregation path to the concentrator in the order $s_1, s_2, \ldots, s_n$, the path key is constructed as follows:

1. s_1 randomly selects $k_{init}, k_1 \in [0, M-1]$, where M is a large integer. Then, it securely sends k_{init} to the concentrator via a direct secure connection, and securely sends $K_1 = k_{init} + k_1$ to the next smart meter s_2.
2. For $1 < i \leq n$, s_i randomly selects $k_i \in [0, M-1]$, computes $K_i = K_{i-1} + k_i = k_{init} + \sum_{j=1}^{i} k_j$, and forwards it to s_{i+1} securely. (For simplicity, we assume that s_{n+1} denotes the concentrator.)
3. When the concentrator receives K_n from s_n, it computes $PK = K_n - k_{init} = k_1 + \cdots + k_n$ and stores PK as a path key for the aggregation path.

For instance, in Figure 2, the concentrator would obtain the following path keys: $k_1 + k_2 + k_4$, $k_3 + k_2 + k_4$, and $k_5 + k_6 + k_7 + k_8$.

In the above path key establishment procedure, we assume that the secure channels between each entity can be established using the existing secure protocols such as SSL/TLS. The secure channels between each entity are used only for the secret value transmission at the path key setup phase. After the path key establishment, the secure channels are not used during the metering data aggregation procedure for efficiency. We also assume that every smart meter on the aggregation path participates in the path key computation and delivery process exactly once for each path key establishment.

The security of the path key establishment protocol is guaranteed by the randomly generated initial key k_{init} of the leaf smart meter (s_1 in the above setting) in the aggregation tree. Since k_{init} is added with real secret keys, the possibility of guessing k_{init} and the real secret path keys depends on the length of the secret keys. Thus, it is impossible to determine either k_{init} or the path keys from their composed values when M is large. When an aggregation path consists of n intermediate meters to the concentrator and the attacker can compromise m meters, the attacker cannot guess the path key as long as $m < n$, or the secret keys of non-compromised meters.

Data Encryption

For $1 \leq i \leq n$, s_i encrypts its measurement m_i as $c_i = Enc_{k_i}(m_i)$ and forwards $L_i = (L_{i-1}, (ID_i, c_i))$ (for a first smart meter, *i.e.*, if $i = 1$, L_0 is defined as null) to the subsequent smart meter s_{i+1} together with the aggregate signature σ (which will be generated below) until it reaches the concentrator.

4.2.3. Signing

For $1 \leq i \leq n$, s_i generates the aggregate signature by running the $Sign$ on the input of sk_{ID_i}, c_i, and σ, which is an aggregate-so-far signature (for a first smart meter s_1, σ is defined as $(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})$), and L_{i-1}. s_i first parses σ as $(\sigma_1, \sigma_2, \sigma_3)$ and computes

1. $r_i, x_i \xleftarrow{\$} \mathbb{Z}_p$,
2. $\sigma_3' \leftarrow \sigma_3 \cdot g^{x_i}$,

3. $\sigma_2' \leftarrow \sigma_2 \cdot g^{r_i}$,
4. $\sigma_1' \leftarrow \sigma_1 \cdot (\sigma_3)^{r_i} \cdot (\sigma_2')^{x_i} \cdot H_2(ID_i)^{\alpha_2} \cdot H_1(ID_i)^{\alpha_1 \cdot H_3(ID_i||c_i)}$.

It then sets $\sigma = (\sigma_1', \sigma_2', \sigma_3')$ and forwards it to the next smart meter s_{i+1} together with L_i.

4.2.4. Verification

When the concentrator receives L_n and σ from s_n, the aggregate signature σ has the form

$$\sigma = (g^{xr} \cdot \prod_i H_2(ID_i)^{\alpha_2} \cdot H_1(ID_i)^{\alpha_1 \cdot H_3(ID_i||c_i)}, g^r, g^x),$$

where $r = \sum_i r_i$ and $x = \sum_i x_i$. The concentrator then parses σ as $(\sigma_1, \sigma_2, \sigma_3)$, and runs the Vf algorithm to check if

$$\begin{aligned} e(\sigma_1, g) &\stackrel{?}{=} e(\sigma_2, \sigma_3) \cdot e(\prod_{i=1}^{n} H_2(ID_i), g^{\alpha_2}) \cdot \\ & \quad e(\prod_{i=1}^{n} H_1(ID_i)^{H_3(ID_i||c_i)}, g^{\alpha_1}). \end{aligned}$$

If so, the algorithm returns 1; otherwise, it returns 0.

4.2.5. Data Decryption

If the verification algorithm returns 1, the concentrator runs the Dec algorithm on the input of L_n and $PK(= k_1 + \cdots + k_n)$, and outputs decrypted data by computing

1. $C = \sum_{i=1}^{n} c_i$,
2. $Dec_{PK}(C) = m_1 + \cdots + m_n$.

We can easily check the correctness of the decryption in the following manner:

$$\begin{aligned} Dec_{PK}(C) &= \sum_{i=1}^{n} c_i - PK \\ &= (m_1 + \cdots + m_n) + (k_1 + \cdots + k_n) - PK \\ &= m_1 + \cdots + m_n. \end{aligned}$$

4.3. Discussion

In the proposed scheme, smart meters forward their own encrypted measurements, as well as those received from their child meters in the aggregation tree. Upon receipt of all the encrypted measurements for the aggregation path, the concentrator decrypts the aggregated data with the corresponding path key. In this approach, smart meters that are closer to the concentrator send and receive up to several orders of magnitude more bits than those on the periphery of the spanning tree. Thus, there is an imbalance between the smart meters in terms of the amount of data communicated.

One solution to this is to construct the spanning tree using a breadth-first traversal of the graph, starting at the concentrator. In this way, the height of the tree is the same as the shortest distance from the

furthest meter to the concentrator. It reduces the maximum number of hops for the longest aggregation path, thereby reducing the end-to-end aggregation time and partially relieving (but not perfectly solving) the imbalance problem in terms of communication overhead.

Another solution to the problem is for the smart meters to perform the aggregation of their encrypted measurement with that received from the child meter. That is, smart meter s_i performs $c_j + c_i$, where c_j is the ciphertext received from its child meter s_j, and forwards it, rather than forwarding c_j and c_i individually. This approach completely solves the imbalance in the smart grid network; however, it does not allow the concentrator to verify the measurements due to the malleability property of homomorphic encryption. Therefore, there is a trade-off between communication overhead and data authentication.

5. Efficiency

In this section, we first analyze and compare the efficiency of the proposed scheme with the previous aggregation schemes, that is Garcia *et al.*'s scheme (GJ) [3], Li *et al.*'s scheme (LLL) [5,6], and Li *et al.*'s scheme (LLYLLS) [21] from a theoretical perspective. We also discuss its efficiency when implemented with specific parameters.

5.1. Security Property

Table 1 shows the security properties that each scheme supports. The previous schemes [3,5,6] guarantee data confidentiality in that a passive eavesdropper cannot obtain plain metering data during the aggregation process. Thus, they provide user privacy against curious smart meters or the concentrator. However, they have an inherent weakness against active attacks due to the malleability property of the homomorphic encryption algorithm. Given the ciphertext and public key, an adversary could generate another ciphertext which decrypts to another meaningful plaintext in the same domain as the original plaintext.

Table 1. Security property comparison.

Scheme	Confidentiality (User Privacy)	Data Integrity	Sender Authentication
GJ [3]	yes	no	no
LLL [5,6]	yes	no	no
LLYLLS [21]	yes	yes	yes
Proposed	yes	yes	yes

Unlike the previous schemes [3,5,6], the proposed scheme guarantees aggregated data integrity and smart meter authentication through the aggregate signature mechanism, as well as data confidentiality. Thus, the proposed scheme is secure against active attacks such as data manipulation by a dishonest or fake smart meter, or false data injection by outside adversaries, in addition to passive attacks. Li *et al.*'s scheme [21] also guarantees all the security properties, but their scheme allows the verifier to

check integrity of individual metering data, not aggregated one. This incurs much higher communication and computation cost compared to the proposed scheme as will analyze in the next section. A more formal security analysis of the proposed scheme will be given in Section 6.

5.2. Theoretical Analysis

The theoretical efficiency comparison of the schemes are summarized in Table 2. The notations used in the table are:

$C_{\mathbb{G}}$	bit size of an element in $\mathbb{G}$
C_p	bit size of an element in $\mathbb{Z}_p^*$
C_n	bit size of an element in $\mathbb{Z}_n^*$
	($n = p \cdot q$ where q is a large prime)
C_{n^2}	bit size of an element in $\mathbb{Z}_{n^2}^*$
C_{ID}	bit size of an identity string
C_t	bit size of a timestamp
C_m	bit size of a message (=symmetric key size)
N	the number of smart meters participating
	in the aggregation

Each scheme is analyzed with regard to communication, message size, signature size, private key size, and public key size. Communication represents the number of message exchanges between smart meters and the concentrator, which are required for a metering data aggregation in a neighborhood. Message represents the total size of the ciphertext and the identity information that each smart meter is required to send for an aggregation. Signature represents the size of signatures that are delivered to the concentrator for an aggregation. Private and public key represent the number of private keys each smart meter stores and the number of public keys in the system for data aggregation in a neighborhood, respectively.

Table 2. Efficiency comparison.

System	Communication	Message	Signature	Private Key	Public Key
GJ [3]	$N(N+1)$	$(N-1)(C_{ID}+C_{n^2})+C_m$	-	$2C_p$	$N(C_n+C_{n^2})$
LLL [5,6]	N	$C_{ID}+C_{n^2}$	-	-	$C_n+C_{n^2}$
LLYLLS [21]	$N+1$	$(N+1)(C_{n^2}+2C_{ID}+C_t)$	$(N+1)(C_{\mathbb{G}}+C_p)$	$C_{\mathbb{G}}+C_p+C_m$	$C_n+C_{n^2}$
Proposed	$2N+1$	$[C_{ID}+C_m, N(C_{ID}+C_m)]$	$3C_{\mathbb{G}}$	$2C_{\mathbb{G}}+C_m$	$3C_{\mathbb{G}}$

In Garcia *et al.*'s scheme [3], every smart meter in the neighborhood encrypts its own metering data with the other meters' public keys and performs a three-way handshake with the concentrator in order to aggregate metering data in the neighborhood. Thus, the scheme needs $O(N)$ pairs of public keys of all smart meters participating in the aggregation, and $O(N^2)$ communications. In Li *et al.*'s scheme [5,6], each smart meter encrypts its measurement with the aggregate-so-far message under the concentrator's public key in the hop-by-hop approach following the aggregation tree. Since the smart meters do not need to decrypt messages during aggregation, they are not given private keys. In Li *et al.*'s scheme [21], each meter in the same area network sends its data and signature to the local gateway directly, which is then aggregated with those from the other meters and delivered to the final concentrator.

In the proposed scheme, $N + 1$ more communications are needed than Li *et al.*'s scheme [5,6] in order to establish a path key for an aggregation path. The signature size is constant regardless of the number of signers because each signature is aggregated. Since every smart meter concatenates its data to the list of identity-ciphertext pairs and forwards it to the next smart meter until it reaches the concentrator, the message size a smart meter is required to send increases in proportion to the proximity to the concentrator. As we discussed in Section 4.3, this communication overhead is caused by security requirements that ensure data integrity and sender authentication.

If we do not consider data integrity and sender authentication, as in the previous schemes [3,5,6], the message size and private key size in the proposed scheme would be reduced to $C_{ID} + C_m$ and C_m, respectively, without any signature and public key overhead because each meter s_i can perform $c_i = Enc_{k_i}(m_i)$ and forward the aggregated result $C + c_i$ to the next meter until it reaches the concentrator, where C is the ciphertext aggregated so far. This approach would thus be the most efficient of the schemes and resolve the imbalance problem (but would not be secure against active attacks, as with the previous schemes).

Figures 3 and 4 illustrate the same experimental results on different scales for clear comparison. They show the total message size for metering data aggregations in relation to the number of participating smart meters. (For the proposed scheme, the signature is included in the result.) The message size in Figures 3 and 4 are represented in bits on a linear scale and a log scale, respectively. In the simulation, we set $C_{\mathbb{G}} = 512, C_p = 160, C_n = 160^2, C_{n^2} = 160^4, C_{ID} = 64, C_t = 32$, and $C_m = 32$. Each element size in the groups is chosen for an 80-bit security level. As Figure 3 demonstrates, the proposed scheme requires the least communication overhead since the message size is only determined by the number of meters in the aggregate group and the size of an identity and a message (that is, the metering data), which are much smaller than that of the element in $\mathbb{Z}^*_{n^2}$. Therefore, efficiency is much improved in the proposed scheme in terms of communication overhead.

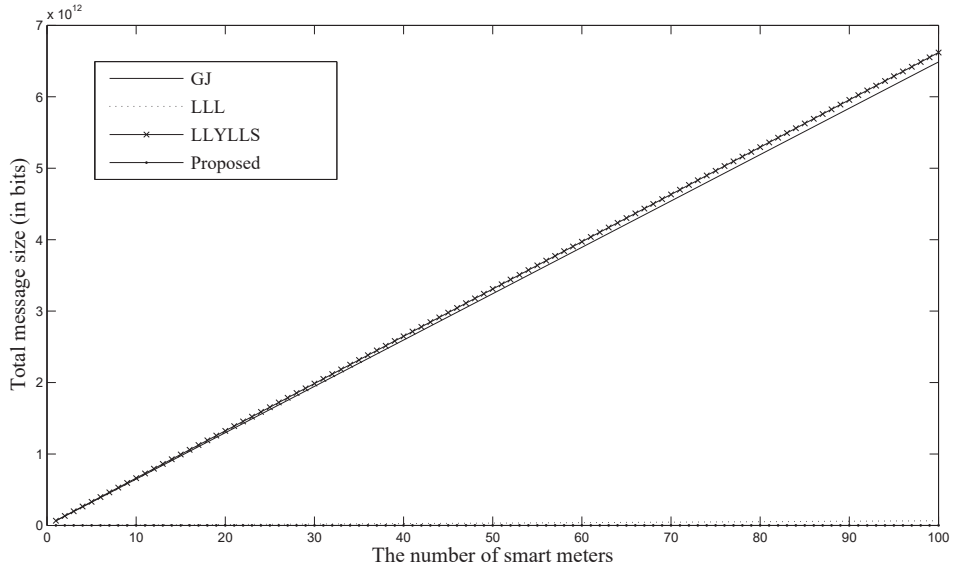

Figure 3. Total message size for an aggregation (linear scale).

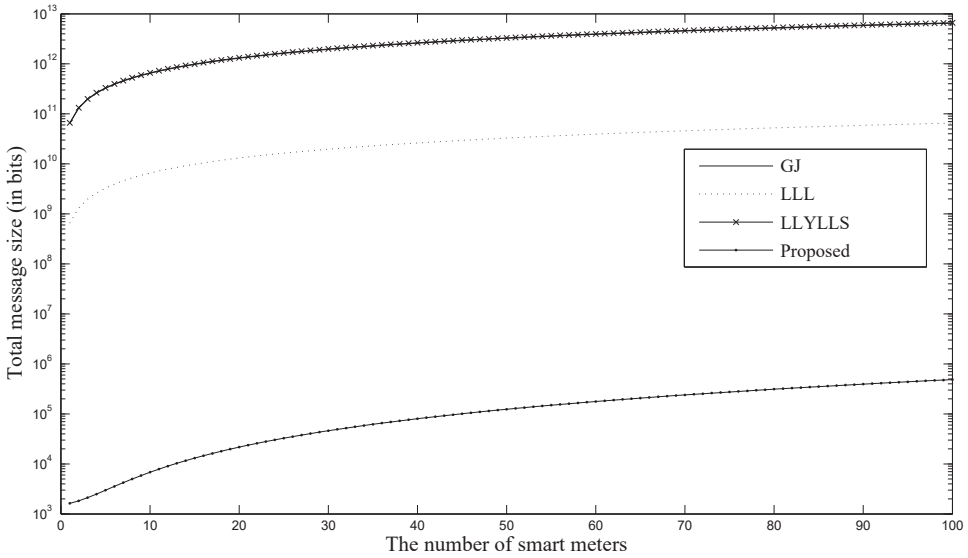

Figure 4. Total message size for an aggregation (log scale).

5.3. Implementation

Next, we analyze the computation cost to aggregate metering data and verify their authenticity in a neighborhood. In the proposed scheme, the stream cipher is used to encrypt metering data from each smart meter and decrypt the aggregated metering data at the concentrator. In general, a block cipher like AES is typically 100 times faster than RSA encryption and 2000 times faster than RSA decryption, with about 60 MB per second on a modest platform [38]. Stream ciphers are even faster, and our symmetric encryption is able to encrypt and decrypt 100 MB per second or more. Therefore, the computation overhead for encryption and decryption operations in the proposed scheme is negligible compared to the previous schemes [3,5,6] using Paillier's asymmetric homomorphic encryption [9,11]. The most time-consuming operations in the proposed scheme are signature generation (by smart meters)

and verification (by the concentrator), which are based on pairing computations. Thus, we measure only the asymmetric computations required for signature generation and verification in the proposed scheme.

We used a Type A curve (in the pairing-based cryptography (PBC) library [39]) providing groups in which bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is defined. Although such curves provide good computational efficiency (especially for pairing computation), the same does not hold from the point of view of the space required to represent group elements. Indeed, each element of $\mathbb{G}$ requires 512 bits at an 80-bit security level and 1536 bits when 128-bit security is chosen.

Table 3 shows the computational times for signature generation by a smart meter and signature verification by a concentrator. For each operation, we include benchmark timing. Each cryptographic operation was implemented using the PBC library version 0.4.18 [39] on a PC with 3.0 GHZ processor. The public key parameters were selected to provide 80-bit security. The implementation uses a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field. The computational cost is analyzed in terms of the pairing and exponentiation operations in $\mathbb{G}$ and $\mathbb{G}_T$. The comparatively negligible hash operations are ignored.

Table 3. Computation cost for signing and verification.

Operation	Time (ms)	Smart Meter	Concentrator
Pairing	2.9	-	4
Exp. in $\mathbb{G}$	1.0	5	-
Exp. in $\mathbb{G}_T$	0.2	-	N
Computation (ms)		5.0	$0.2N + 11.6$

As shown in Table 3, each smart meter needs a constant time period to generate an aggregate signature for its data which is independent of the number of smart meters in the neighborhood. The concentrator, on the other hand, needs a time period which has a linear relationship with the number of meters in order to verify the aggregate signature. However, this cost could be alleviated by reducing the aggregate group size in practice (though the size should be set at more than two to provide user privacy).

In practice, the size of the aggregate group would be determined on the basis of the size of the smart grid system. For example, the aggregate group size can be determined by considering the hop counts (between the smart meter and the collection unit) and the grid network delay [5,6]. When an aggregate group consists of N smart meters and the aggregation tree is constructed with a complete binary tree (the root node is the concentrator in the tree), the maximum hop counts from the meter to the concentrator is $\log(N + 2) - 1$. Since the average hop count is 15, and the network delay for each hop is currently 20 ms for the Internet [14], N could be set to at least $2^{16} - 2 \approx 65500$ in grid networks using the Internet. When it comes to the number of collectors in the grid network, it can be set to $S/N \approx S/65500$ where S is the whole size or total number of nodes in the grid network.

As seen in Table 3, it takes $(0.2N + 11.6)$ms for the concentrator to verify the integrity of the received data while the decryption time is comparably negligible. Since the current network delay for each hop is 20ms and the signature generation by each smart meter requires 5 ms, the total time overhead would be

at most $(20+5)*(\log(N+2)-1)+(0.2N+11.6)$ ms. Thus, as long as the metering frequency is less than $1/(25*(\log(N+2)-1)+(0.2N+11.6))$ ms, the overhead of the proposed security mechanism would be tolerable in a smart grid network.

6. Security

In this section, we prove the security of the proposed scheme with regard to the security requirements discussed in Section 2.

6.1. Data Confidentiality (User Privacy)

Symmetric homomorphic encryption, which is used for metering data aggregation, is unconditionally secure, that is, perfectly secure [40]. Thus, the knowledge of a ciphertext does not provide any information about either the corresponding plaintext or the key. The path key which is used as the decryption key for an aggregated ciphertext on the aggregation path can be obtained only by the concentrator as long as the smart meters and the concentrator do not collude with each other, and every smart meter participates in the path key establishment procedure on aggregation paths it belongs. In addition, under the same assumption, each intermediate smart meter can by no means obtain the other meter's secret key k_i during the path key establishment procedure since the path key aggregation and distribution are done through the secure channel like SSL/TLS. Without k_i, it is computationally infeasible to decrypt $c_i = Enc_{k_i}(m_i)$. Therefore, each c_i does not leak information about m_i without k_i. Thus, any curious smart meters on the path or passive eavesdroppers cannot obtain any information about the plaintext.

In addition, the concentrator also cannot guess individual metering data from each encrypted metering measurement sent from smart meters on the routing path because it cannot determine each individual symmetric key from the combined path key. Specifically, the security of the path key establishment protocol is guaranteed by the randomly generated initial key k_{init} of the leaf smart meter in the aggregation tree. Since the randomly generated initial key is added with real secret keys, the possibility of guessing k_{init} and the real secret path keys depends on the length of the secret keys. Thus, it is impossible to determine either k_{init} or the path keys from their composed values when M is large. When an aggregation path consists of n intermediate meters to the concentrator and the attacker can compromise m meters, the attacker cannot guess the path key as long as $m < n$, or the secret keys of non-compromised meters.

Therefore, user privacy can be preserved in the proposed scheme against passive adversaries.

6.2. Data Integrity and Sender Authentication

We now prove that our scheme guarantees aggregated data integrity and smart meter authentication for those that participate in the aggregation.

Theorem 2 implies that the aggregate signature scheme is secure against active attacks such as message forgery or false data injection. Thus, smart meters can attest to their own encrypted metering

data. The concentrator can verify the integrity of each ciphertext generated from smart meters on the aggregation path, and assure the identities of the meters participating in the aggregation.

Now, we prove that if each ciphertext is authentic and not forged during the aggregation (by Theorem 2), the integrity of the final aggregation result is also guaranteed. Without a loss of generality, we suppose smart meters $s_1, \ldots, s_n$ participate in the aggregation and report $c_1(= Enc_{k_1}(m_1) \pmod M), \ldots, c_n(= Enc_{k_2}(m_n) \pmod M)$ to the concentrator. Let the path key obtained by the concentrator be $PK = k_1 + \cdots + k_n$ for the aggregation path, and $C = c_1 + \cdots + c_n$. If $\forall i$, c_i is not forged, $m_i = c_i - k_i$ is also authentic for a given k_i. Then, we can observe that

$$\begin{aligned} Dec_{PK}(C) &= \sum_{i=1}^{n} c_i - PK = \sum_{i=1}^{n} (c_i - k_i) \\ &= m_1 + \cdots + m_n \end{aligned}$$

is also authentic for a given PK. If any adversaries manipulate the distributed encryption key after the initial path key setup phase and alter a metering data (or inject false data) during the aggregation procedure, it may affect the path key and result in different ciphertext (from the authentic one), which can be detected during the verification procedure by the concentrator. Therefore, aggregated data integrity and sender authentication properties can be achieved in the proposed scheme.

However, if some authenticated meters are compromised subsequently and then launch the same attack using the valid encryption keys and identities (that is, inside adversary model), the proposed scheme cannot detect such an active attack by the compromised meters, because the attack could be done using the valid encryption keys and valid identities. Thus, dealing with the inside attack requires additional security techniques, which is one of the challenging future research issues.

7. Conclusions

Transmitting power consumption levels of individual customers to an electricity supplier or utility over short intervals has advantages for supplier planning and management purposes in smart grids. However, it threatens user privacy by disclosing fine-grained consumption data and usage behavior to the supplier. In this study, we proposed a novel smart metering scheme that features a mechanism that (1) preserves user privacy using symmetric homomorphic encryption so that only the data collection unit can decrypt the aggregated metering data without knowing individual meter measurements; and (2) enables the collection unit to verify the integrity of aggregated data and authenticate the identities of the smart meters participating in the aggregation by exploiting the aggregate signature. Thus, the proposed scheme provides a secure metering solution for smart grids against passive and active attacks. The proposed scheme is also efficient in terms of computation due to symmetric encryption. Because the aggregate signature is constant in size regardless of the number of smart meters in a neighborhood, the additional communication overhead for the signature is also alleviated.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No. 2013R1A2A2A01005559). This work was also supported by

Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea Government (MSIP) (No. B0190-15-2028 and No. R0190-15-2011)

Author Contributions

Junbeom Hur contributed the ideas and wrote the paper; Dongyoung Koo and Youngjoo Shin designed and performed the experiments.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Metke, A.R.; Ekl, R.L. Security Technology for Smart Grid Networks. *IEEE Trans. Smart Grid* **2010**, *1*, 99–107.
2. Khurana, H.; Hadley, M.; Lu, N.; Frincke, D.A. Smarg-Grid Security Issues. *IEEE Secur. Priv.* **2010**, *8*, 81–85.
3. Garcia, F.D.; Jacobs, B. Privacy-friendly Energy-Metering via Homomorphic Encryption. In Proceedings of the International Workshop on Security and Trust Management, Athens, Greece, 23–24 September 2010; pp. 226–238.
4. Efthymiou, C.; Kalogridis, G. Smart Grid Privacy via Anonymization of Smart Metering Data. In Proceedings of the IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 4–6.
5. Li, F.; Luo, B.; Liu, P. Secure Information Aggregation for Smart Grids Using Homomorphic Encryption. In Proceedings of the IEEE International Conference on Smart Grid Communications, Gaithersburg, MD , USA, 4–6 October 2010; pp. 327–332.
6. Li, F.; Luo, B.; Liu, P. Secure and Privacy-Preserving Information Aggregation for Smart Grids. *Int. J. Secur. Netw.* **2011**, *6*, 28–39.
7. Rial, A.; Danezis, G. Privacy-Preserving Smart Metering. In Proceedings of the ACM Workshop on Privacy and the Electronic Society, Chicago, IL, USA, 17 October 2011; pp. 49–60.
8. Bohli, J.M.; Sorge, C.; Ugus, O. A Privacy Model for Smart Metering. In Proceedings of the IEEE International Conference on Communications Workshops, Capetown, South Africa, 23–27 May 2010; pp. 1–5.
9. Paillier, P. Public-Key Cryptosystem Based on Composite Degree Residuosity Classes. In *Advances in Cryptology—EUROCRYPT'99*; Springer Berlin Heidelberg: Heidelberg, Germany, 1999; Volume 1592, pp. 223–238.
10. Boneh, D.; Goh, E.; Nissim, K. Evaluating 2-DNF Formulas on Ciphertexts. In *Theory of Cryptography*; Springer Berlin Heidelberg: Heidelberg, Gemany, 2005; pp. 325–341.
11. Paillier, P.; Pointcheval, D. Efficient Public-Key Cryptosystems Provably Secure Against Active Adversaries. In *Advances in Cryptology-ASIACRYPT'99*; Springer Berlin Heidelberg: Heidelberg, Gemany, 1999; pp. 165–179.

12. McDaniel, P.; McLaughlin, S. Security and Privacy Challenges in the Smart Grid. *IEEE Secur. Priv.* **2009**, *7*, 75–77.
13. Lee, A.; Brewer, T. *The Cyber Security Coordination Task Group, Smart Grid Cyber Security Strategy and Requirements*; NIST Tech. Rep. Draft NISTIR 7628; 2009. Available online: http://www.smartgridinformation.info/pdf/1200_doc_1.pdf (accessed on 10 October 2015).
14. Seo, D.; Lee, H.; Perrig, A. Secure and Efficient Capability-based Power Management in the Smart Grid. In Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops, Busan, Korea, 26–28 May 2011; pp. 119–126.
15. Garofalakis, M.; Hellerstein, J.M.; Maniatis, P. Proof Sketches: Verifiable In-Network Aggregation. In Proceedings of the International Conference on Data Engineering, Istanbul , Turkey, 15–20 April 2007; pp. 996–1005.
16. Castelluccia, C.; Chan, A.C.-F.; Mykletun, E.; Tsudik, G. Efficient and Provably Secure Aggregation of Encrypted Data in Wireless Sensor Networks. *ACM Trans. Sens. Netw.* **2009**, *5*, 1–36.
17. Boneh, D.; Freeman, D.M. Homomorphic Signatures for Polynomial Functions. In *Advances in Cryptology—EUROCRYPT 2011*; Springer Berlin Heidelberg: Heidelberg, Gemany, 2011.
18. Marmol, F.; Sorge, C.; Ugus, O.; Perez, G. Do Not Snoop My Habits: Preserving Privacy in the Smart Grid. *IEEE Commun. Mag.* **2012**, *50*, 166–172.
19. Cho, S.; Li, H.; Choi, B. PALDA: Efficient Privacy-Preserving Authentication for Lossless Data Aggregation in Smarg Grids. In Proceedings of the IEEE International Conference on Smart Grid Communications, Venice, Italy, 3–6 November 2014; pp. 914–919.
20. Alharbi, K.; Lin, X.; Shao, J. A Framework for Privacy-preserving Data Sharing in the Smart Grid. In Proceedings of the IEEE/CIC International Conference on Communications in China, Shanghai, China, 13–15 October 2014; pp. 214–219.
21. Li, H.; Lin, X.; Yang, H.; Liang, X.; Lu, R.; Shen, X. EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key Evolution in Smart Grid. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2053–2064.
22. Borges, F.; Demirel, D.; Bock, L.; Buchmann, J.; Muhlhauser, M. A Privacy-Enhancing Protocol that Provides In-Network Data Aggregation and Verifiable Smart Meter Billing. In Proceedings of the IEEE Symposium on Computers and Communication, Funchal, Portugal, 23–26 June 2014; pp. 1–6.
23. Deng, P.; Yang, L. A Secure and Privacy-preserving communication scheme for Advanced Metering Infrastructure. In Proceedings of the IEEE PES Innovative Smart Grid Technologies, Washington, DC, USA, 16–20 January 2012; pp. 1–5.
24. He, X.; Pun, M.; Kuo, C. Secure and Efficient Cryptosystem for Smart Grid Using Homomorphic Encryption. In Proceedings of the IEEE PES Innovative Smart Grid Technologies, 16–20 January 2012; pp. 1–8.
25. Li, F.; Luo, B. Preserving Data Integrity for Smart Grid Data Aggregation. In Proceedings of the IEEE Third International Conference on SmartGridComm, Tainan, Taiwan, 5–8 November 2012; pp. 366–371.

26. Rottondi, C.; Savi, M.; Verticale, G.; Kraub, C. Mitigation of Peer-to-Peer Overlay Attacks in the Automatic Metering Infrastructure of Smart Grids. *Secur. Commun. Netw.* **2015**, *8*, 343–359.
27. Boneh, D.; Franklin, M.K. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology—CRYPTO 2001*; Springer Berlin Heidelberg: Heidelberg, Gemany, 2001; Volume 2139, pp. 213–229.
28. Galbraith, S.D.; Harrison, K.; Soldera, D. Implementing the Tate Pairing. In *Algorithmic Number Theory*; Springer Berlin Heidelberg: Heidelberg, Gemany, 2002; Volume 2369, pp. 324–337.
29. Boldyreva, A.; Gentry, C.; O'Neill, A.; Yum, D. Ordered Multisignatures and Identity-Based Sequential Aggregate Signatures, with Applications to Secure Routing. A Corrected Version of the Paper [35] 2007. Available online: http://www.cc.gatech.edu/~aboldyre/papers/bgoy.pdf (accessed on 11 October 2013).
30. Gamal, T.E. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology*; Springer Berlin Heidelberg: Heidelberg, Gemany, 1985; pp. 10–18.
31. Naccache, D.; Stern, J. A New Public Key Cryptosystem Based on Higher Residues. In Proceedings of the ACM Conference on Computer and Communications Security, San Francisco , CA, USA, 2–5 November, 1998; pp. 59–66.
32. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Advances in Cryptology—EUROCRYPT 2003*; Springer Berlin Heidelberg: Heidelberg, Gemany, 2003.
33. Lysyanskaya, A.; Micali, S.; Reyzin, L.; Shacham, H. Sequential Aggregate Signatures from Trapdoor Permutations. In *Advances in Cryptology—EUROCRYPT 2004*; Springer Berlin Heidelberg: Heidelberg, Gemany, 2004.
34. Lu, S.; Ostrovsky, R.; Sahai, A.; Shacham, H.; Waters, B. Sequential Aggregate Signatures and Multisignatures without Random Oracles. In *Eurocrypt*; St. Petersburg, Russia, May 28–June 1, 2006; pp. 465–485.
35. Boldyreva, A.; Gentry, C.; O'Neill, A.; Yum, D. Ordered Multisignatures and Identity-Based Sequential Aggregate Signatures, with Applications to Secure Routing. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, October 29 – November 2 2007; pp. 276–285.
36. Hwang, J.Y.; Lee, D.H.; Yung, M. Universal Forgery of the Identity-Based Sequential Aggregate Signature Scheme. In Proceedings of ACM Symposium on Information, Computer and Communications Security, Sydney, Australia, 10–12 March 2009; pp. 157–160.
37. Castelluccia, C.; Mykletun, E.; Tsudik, G. Efficient Aggregation of Encrypted Data in Wireless Sensor Networks. In Proceedings of the The Second Annual International Conference on ACM/IEEE Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous'05), San Diego, CA, USA, 17–21 July 2005; pp. 109–117.
38. Fontaine, C.; Galand, F.; A Survey of Homomorphic Encryption for Nonspecialists. *J. Inf. Secur.* **2009**, *1*, 41–50.
39. The Pairing-Based Cryptography Library. Available online: http://crypto.stanford.edu/pbc/ (accessed on 23 October 2014).

40. Shannon, C. Communication Theory of Secrecy Systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715.