energies

MDPI

Article

A Privacy-Preserving Noise Addition Data Aggregation Scheme for Smart Grid

Yuwen Chen *, José-Fernán Martínez, Pedro Castillejo and Lourdes López

Departamento de Ingeniería Telemática y Electrónica (DTE), Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIST), Universidad Politécnica de Madrid (UPM), C/Nikola Tesla, s/n, 28031 Madrid, Spain; jf.martinez@upm.es (J.-F.M.); pedro.castillejo@upm.es (P.C.); lourdes.lopez@upm.es (L.L.)

* Correspondence: yuwen.chen@upm.es; Tel.: +34-913-365-526

Received: 27 September 2018; Accepted: 24 October 2018; Published: 1 November 2018

Abstract: Smart meters are applied to the smart grid to report instant electricity consumption to servers periodically; these data enable a fine-grained energy supply. However, these regularly reported data may cause some privacy problems. For example, they can reveal whether the house owner is at home, if the television is working, etc. As privacy is becoming a big issue, people are reluctant to disclose this kind of personal information. In this study, we analyzed past studies and found that the traditional method suffers from a meter failure problem and a meter replacement problem, thus we propose a smart meter aggregation scheme based on a noise addition method and the homomorphic encryption algorithm, which can avoid the aforementioned problems. After simulation, the experimental results show that the computation cost on both the aggregator and smart meter side is reduced. A formal security analysis shows that the proposed scheme has semantic security.

Keywords: noise generation methods; bilinear map; smart grid; meter aggregation; homomorphic encryption

1. Introduction

Smart meters are widely applied in Europe. Member states have committed to rolling out close to 200 million smart meters for electricity and 45 million for gas by 2020 [1], and more than 200 million European households will have smart meters in 2023 [2]. According to the European Parliament and the European Council, "Member States are required to ensure the implementation of smart metering systems that assist the active participation of consumers in the electricity supply and gas supply markets" [3].

Smart meters can report instant electricity consumption to servers periodically, making fine-grained energy supply possible. However, these instantly reported data also bring some potential privacy risks. By using advanced power signature analysis tools such as nonintrusive appliance load monitoring (NIALM), an attacker can find out which appliances are working at any time [4], and thus can learn more detailed information about a customer's daily activities. According to Barbosa et al. (2015) [5], "Fine-grained data of electricity usage naturally include personal and privacy-sensitive information regarding which appliances are active." For example, the adversary can tell if there are people in the house or not, when the inhabitants wake up, take a shower, turn off the television, or even if some individual appliances are operating at a desired level of efficiency. There is a great need to protect this kind of personal information from being disclosed. Thus smart meter aggregation schemes have been proposed to protect people's privacy.

Recently, Fan et al. (2014) proposed a smart meter aggregation scheme based on the bilinear map and computationally hard problems of group theory [6]. He et al. (2017) improved the scheme of [6] by importing the homomorphic encryption algorithm [7]. Both of these schemes were claimed to be secure. However, we found that although both schemes can protect a user's personal data from being leaked, they both have scalability problems. Once the system is deployed, it is hard to add a new smart meter to the system, and when one smart meter in the system is broken, the whole system cannot work correctly. In addition, replacing a broken smart meter with a new one is difficult. Moreover, both schemes have higher accuracy requirements for time, which means that all the smart meters in the system have to keep exactly the same time; even a one millisecond error will lead to an incorrect result. We will discuss these problems in Section 3.

To solve these problems, a privacy-preserving data aggregation scheme for the smart grid is proposed, which enables smart meters to report their consumption periodically and at the same time prevents private information from being leaked. The proposed scheme is partly based on the homomorphic encryption algorithm. Our contributions are mainly reflected in two aspects:

1. First, the noise addition method is used to prevent an adversary from obtaining a smart meter's consumption, and the efficiency of the proposed scheme is improved by using this method. We also analyzed different ways of generating noise.
2. Second, the proposed scheme overcomes the problems in related works, such as the scalability problem, and does not have a high accuracy requirement for time.

This study focuses on the security and privacy part of work done under the e-GOTHAM project; the previous work has been published [8]. The paper is organized as follows: Related works are discussed in Section 2. In Section 3, we discuss the problems of the two related works. The proposed scheme is introduced in Section 4. Security analysis is described in Section 5. A comparison with the related schemes is in Section 6. We conclude the paper in Section 7.

2. Related Work

Smart grid privacy and security problems have drawn much attention. There are many ways to protect the privacy of a smart meter when it reports its consumption to the aggregator; for example, homomorphic encryption methods, rechargeable battery methods, noise addition methods, and trusted third party methods.

Noise addition is a promising and efficient way to protect the consumption privacy of a smart meter. Bohli first used this approach [9], and Barbosa et al. (2015) [5] and Wang et al. (2013) [10] analyzed the privacy and utility metric of this problem, both proposing a metric for utility preservation. Wang et al. (2013) masked the data using Gaussian mixture models (GMMs) [10]. Their experimental results show that the accuracy of recovering total electricity consumption can approximate 99%, while the ability to identify an individual's usage pattern is substantially obviated. He et al. (2013) proposed masking the data by adding Gaussian noise [11]. Random noise is purposely introduced to distort the smart meter's consumption so that it is infeasible for an adversary to recover the real consumption. The random noise is chosen according to the power consumption data and other prior knowledge. Jordi and Josep analyzed the optimality of data-independent random noise distributions to achieve ε-differential privacy [12]. They also analyzed the situations for single univariate query and multiple queries. Noise addition methods can significantly reduce the computation and communication costs of smart meters. "Since to preserve privacy the proposed approach just generates a random number, we claim that the proposed approach is lightweight" [5]. However, the lack of authentication between the smart meter and the aggregator makes it possible for an adversary to easily launch an attack.

Some schemes require a trusted third party; we call this the trusted third party model, in which a trusted third party is introduced. He et al. (2017) built their scheme based on elliptic curve cryptography (ECC) [13]. Fan et al. (2014) proposed a scheme based on the bilinear map and computationally hard problems in group theory [6]. He et al. (2017) improved the computation efficiency of the scheme of Fan et al. [7], and their scheme reduced the computation cost.

García and Jacobs [14] were the first to try to apply additive homomorphic encryption to privacy-friendly smart metering architecture. In their architecture, each reporting period requires the transmission of O (n^2) ciphertexts. Lu et al. (2016) proposed an efficient and privacy-preserving aggregation scheme for secure smart grid communication [15]. Their scheme realized a multidimensional data aggregation approach based on the homomorphic Paillier cryptosystem, which satisfies the real-time high-frequency data collection requirements of smart grid communication. Busom et al. [16] built their scheme on the homomorphic encryption method, too. By homomorphically adding all n consumption, the existing link between customers and their consumption values is broken. In this way, detailed information can be sent without leaking individual personal data. Their approach does not require a trusted third party (except a certification authority) or communication among smart meters; the communication complexity is linear O (n). Dimitriou and Awad presented two decentralized privacy-respecting aggregating protocols for smart meters [17]. Their first protocol focuses on honest-but-curious adversaries by using symmetric cryptography primitives. Their second one protects against more aggressive adversaries that not only try to infer individual measurements, but also disrupt protocol execution, which is based on public cryptography primitives.

Besides these ways of protecting the privacy of smart meter consumption, authentication between the smart meter and the aggregator is another factor that should receive attention when thinking about privacy protection. Elliptic curve [18–25] and bilinear map pairing [26–30] are two of the most commonly used encryption methods for authentication schemes. Generally speaking, the bilinear map requires more computation cost than the elliptic curve method, and the elliptic curve method is more efficient.

Ping et al. proposed an elliptic curve cryptography–based authentication scheme with identity protection for smart grids [23]. Adversaries are unable to obtain the real identities because the identities of the smart appliances and substations are encrypted before they are transmitted. Saxena and Choi proposed another authentication protocol for smart grid communication, also based on the elliptic curve. The hierarchy of their scheme is also three-layer [24]. The scheme of Nicanfar and Leung is a multilayer consensus password authenticated key-exchange scheme for the smart grid [25]. Saxena et al. proposed an authentication and authorization scheme for the smart grid; the protocol is based on bilinear map pairing [28]. A bilinear pairing cryptography–based shared secret key is generated between the user and the device, and the key enables the two to communicate securely. Odelu et al. proposed a secure key agreement scheme for the smart grid; they built their scheme on bilinear map pairing [29]. Jo et al. proposed privacy-preserving protocols for the smart grid using the distributed verification method; their encryption scheme is based on bilinear map pairing [30].

3. Problems in the Trusted Third Party Model

In a trusted third party model, three types of entities are in the system: smart meters, an aggregator, and a trusted third party. Figure 1 depicts the system structure.

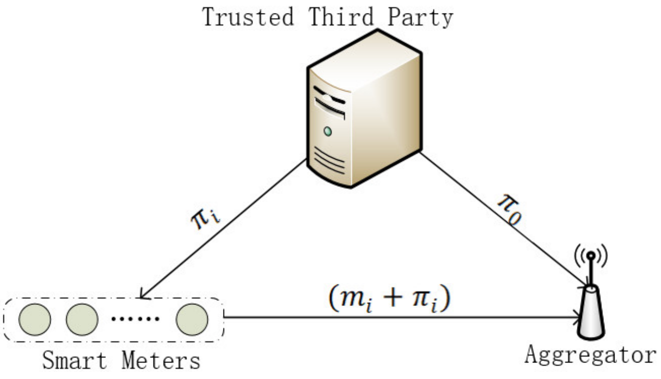

Figure 1. Entities in the trusted third party model.

In this system, during the system initialization phase, the trusted third party will generate a series of random numbers $\pi_0, \pi_1, \pi_1, \ldots, \pi_\mathrm{k}$, and make sure $\pi_0 = -(\pi_1 + \pi_2 +, \ldots, + \pi_k\ ) = -\sum_{i=1}^{k} \pi_i$; these numbers are called blind factors. The blind factor π_0 is sent to the aggregator, and $\pi_1, \pi_2, \ldots, \pi_k$ are sent to the ith smart meter. At the aggregation phase, smart meter M_i sends $(m_i + \pi_i)$ to the aggregator; m_i is the meter's consumption data. The aggregator can recover the total consumption $\sum_{i=1}^{k} m_i$ using π_0.

$$\sum_{i=1}^{k}(m_i + \pi_i) + \pi_0 = \sum_{i=1}^{k} m_i + \sum_{i=1}^{k} \pi_i + \pi_0 = \sum_{i=1}^{k} m_i + \sum_{i=1}^{k} \pi_i - \sum_{i=1}^{k} \pi_i = \sum_{i=1}^{k} m_i$$

In this way, the aggregator can get the total consumption of all the smart meters. However, it is unable to get the consumption of a single smart meter.

3.1. Scalability Problem

One of the drawbacks of the trusted third party model is the scalability problem. After deploying the system, it is difficult to add a new smart meter. If we want to add a smart meter M_{k+1} to the system, we need to assign it a new blind factor, π_{k+1}. However, it is not enough to just assign a new π_{k+1} to the smart meter. We have to update π_0 for the aggregator, otherwise the aggregator is unable to recover the total consumption of the smart meters using the old π_0; π_0 has to be updated to $\pi_0' = -(\pi_1 + \pi_2 +, \ldots, +\pi_k) - \pi_{k+1}$.

However, if π_0' is sent to the aggregator, it can get the blind factor π_{k+1} by computing $\pi_{k+1} = \pi_0' - \pi_0$. If the aggregator knows the blind factor π_{k+1}, it can get the original consumption of smart meter M_{k+1}. One potential solution is to run the system initialization phase again and let the trusted third party assign new blind factors for all smart meters and aggregators; however, it will be a daunting task once the smart meters have been deployed.

Another problem is that the system will fail to work when a smart meter is broken. Suppose M_i is broken and it cannot send $(m_i + \pi_i)$ to the aggregator, then the aggregator is unable to get the total consumption of all the smart meters; what the aggregator gets is $log_{\hat{g}}((H_2(t))^{-\pi_i \cdot q_1} \cdot (g^{q_1})^{\sum_{i=1}^{k} m_i})$, in which $\hat{g} = (g^{q_1})$. The following is an analysis based on the reported data in the research of Fan et al. [6]:

$$\begin{aligned}
\log_{\hat{g}}\left(\prod_{i=1}^{k} c_i\right)^{q_1} &= log_{\hat{g}}\left((H_2(t))^{-\pi_i} \cdot \prod_{i=1}^{k} g_0^{m_i} \cdot h^{r_i' \cdot \pi_i}\ \right)^{q_1} \\
&= log_{\hat{g}}((H_2(t))^{-\pi_i \cdot q_1} \cdot \prod_{i=1}^{k} (g_0^{m_i} \cdot h^{r_i' \cdot \pi_i})^{q_1}) \\
&= log_{\hat{g}}((H_2(t))^{-\pi_i \cdot q_1} \cdot (g^{\sum_{i=1}^{k} m_i})^{q_1} \cdot (h^{\sum_{i=1}^{k} r_i})^{q_1}) \\
&= log_{\hat{g}}((H_2(t))^{-\pi_i \cdot q_1} \cdot (g^{\sum_{i=1}^{k} m_i})^{q_1} \cdot (u^{\sum_{i=1}^{k} r_i' \cdot \pi_i \cdot q_2})^{q_1}) \\
&= log_{\hat{g}}((H_2(t))^{-\pi_i \cdot q_1} \cdot (g^{q_1})^{\sum_{i=1}^{k} m_i} \cdot 1)
\end{aligned}$$

What is worse, it is also difficult to replace a broken smart meter with a new one. If we want to replace the meter, we will encounter the problem of adding a meter to the system. As we have discussed, adding smart meters to the system is difficult.

3.2. Precise Time Requirement

The other problem in the trusted third party model is that it has a high accuracy requirement for time, which means that all of the smart meters have to synchronize their time precisely, because it is a prerequisite of this model that the time of different smart meters must be identical, otherwise the aggregator is unable to recover the original consumption data. The problem becomes worse in Fan's scheme [6], where the aggregator has to synchronize its time with all the smart meters, and even a one millisecond error will lead to a wrong answer.

3.3. Comparison

Finally, we get Table 1, a comparison of the trusted third party model and the proposed scheme. It is clearly shown in the table that the proposed scheme overcomes the problems of the trusted third party model.

Table 1. Comparison of trusted third party model and proposed scheme.

Features	Trusted Third Party Model	Proposed Scheme
Trusted third party	Required	Not Required
Precise time requirement	Required	Not Required
Scalability	Low	High
Adding a new meter	Difficult	Easy
Meter failure problem	×	✓
Replacing a meter	Difficult	Easy

4. Proposed Scheme

The model of the proposed scheme is depicted in Figure 2; there are two types of entities in the system, smart meter and aggregator. All the smart meters in the system have to register at the aggregator first; after registration, they can report their consumption data to the aggregator periodically. The aggregator will only accept the reporting data of the registered smart meters.

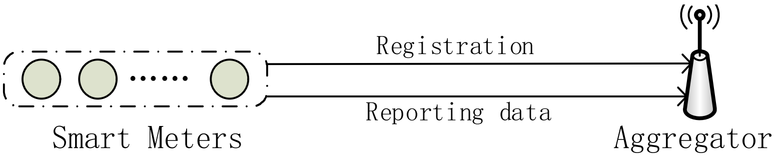

Figure 2. Proposed aggregation model.

To protect the privacy of the users, in every reporting cycle, a smart meter generates a random noise n_i to perturb its consumption, and will send $(n_i + m_i)$ to the aggregator. In this way, the aggregator is unable to get the m_i because it does not know the n_i.

Since the noises are generated following a normal distribution, if we set the average value of the random numbers to be 0, we know $\sum_{i=0}^{k} n_i \approx 0$, k is the number of smart meters in an aggregation system, thus the aggregator can get the total consumption:

$$\sum_{i=0}^{k} m_i = \sum_{i=0}^{k}(n_i + m_i) \approx \sum_{i=0}^{k} m_i$$

We should note here that when k become larger, $\sum_{i=0}^{k} n_i$ will gradually approach 0, and $\sum_{i=0}^{k} n_i$ will not become larger even when k becomes larger.

For example, if we set the tolerable error $\sum_{i=0}^{k} n_i$ to be within the range of [–5, 5] kWh, the probability that $\sum_{i=0}^{k} n_i$ falls into [–5, 5] kWh is set to be: $\Pr(-5 \leq \sum_{i=1}^{k} n_i \leq 5) = 0.98$. If we set $k = 100$, we can get $\sigma_X^2 = 0.0462$. That is, if the noise generated obeys the normal distribution with average $\mu = 0$, and $\sigma_X^2 = 0.0462$, then the sum of $\sum_{i=0}^{k} n_i$ will be within the range of [–5, 5] kWh with a probability of 98%. The noise generated follows other distributions, too, and the results are listed in Table 2 [5].

Table 2. Analytical models obtained for different probability distributions.

Distribution	Model	Comments
Arcsin	$e_0 \sim N(0, \frac{kX^2}{2})$	X is the range of the original distribution
Laplace	$e_0 \sim N(0,\ 2kb^2)$	b is the scale parameter
Normal	$e_0 \sim N(0,\ k\sigma_X^2)$	σ_X^2 is the variance of the original distribution
Uniform	$e_0 \sim N(0, \frac{kX^2}{3})$	X is the range of the original distribution
U-quadratic	$e_0 \sim N(0, \frac{3kX^2}{5})$	X is the range of the original distribution

To find out which distribution model is the best, we use Table 3, which is the distributions of noises when $k = 100$, $\Pr(-5 \leq \sum_{i=1}^{n} n_i \leq 5) = 0.98$.

Table 3. Distribution of noises.

Zone	Arcsin	Laplace	Normal	Uniform	U-quadratic	Arcsin
$-\infty, -0.05$	0%	0.65%	1.52%	13.93%	0%	0%
$-0.05, -0.03$	27.5%	3.03%	08.18%	14.43%	26.34%	27.5%
$-0.03, -0.01$	15.55%	17.28%	23.55%	14.43%	22.78%	15.55%
$-0.01, 0.01$	1.39%	58.09%	33.49%	14.43%	1.75%	13.89%
$0.01, 0.03$	15.55%	17.28%	23.55%	14.43%	22.78%	15.55%
$0.03, 0.05$	27.5%	3.03%	8.18%	14.43%	26.34%	27.5%
$0.05, +\infty$	0%	0.65%	1.52%	13.93%	0%	0%

The noise obeys the normal distribution or the Laplace distribution aggregated too closely around the average value $\mu_e = 0$, which means a large amount of the noise is too small. For noise that obeys the Laplace distribution, 58.09% of the noise is within [–0.01, 0.01], which means more than half of the noise is too small. The range of noises obeying the U-quadratic distribution is [–0.0385, 0.0385] and the range of noises obeying the arcsin distribution is [–0.0462, 0.0462], and both are smaller than the much larger range of noises obeying the uniform distribution, [–0.0693, 0.0693].

Now we can conclude that noises obeying uniform distribution are the best. On the one hand, they are equally distributed within the range; on the other hand, the range of noises obeying uniform distribution is larger.

4.1. Notions Used in the Schemes

The proposed scheme is based on the Boneh–Goh–Nissim homomorphic encryption scheme [31]; Boneh et al. (2005) proposed a probabilistic homomorphic encryption algorithm. The system resembles the Paillier [32] and Okamoto–Uchiyama [33] encryption schemes. This system is additively homomorphic. The proposed scheme consists of three phases, the system initialization phase, the smart meter registration phase, and the meter reporting phase. Some notions are given in Table 4.

Table 4. Symbols used in the scheme.

Symbols	Description
g, u, g_1	Generators of G_1
q_1, q_2	Secret keys of aggregator
k	Number of smart meters in an aggregation system
(M_i, id_i)	ith smart meter and its identity
A_i	ith aggregator
$(x_i,\ X_i)$	Public key pair of smart meter M_i
$(x \leftarrow Z)$	x is randomly picked from set Z
$\vert\,\vert$	String connection
$h()$	General hash SHA256 method
$h_2()$	Hash a string to a big integer

4.2. System Initialization

In this phase, the aggregator initializes and publicizes the parameters; this is a three-step process.

Step 1: For the elliptic curve parameters, the aggregator selects two random τ-bit primes q_1, q_2 and sets $n = q_1 q_2$, and generates a multiplicative group G_1. Let g, u, g_1 be generators of G_1, set $\eta = u^{q_2}$ and $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear map.

Step 2: For the modular exponential group parameters, the aggregator randomly generates two large numbers $\hat{p}, \hat{q}$ ($\hat{p}$ is a 1024-bit prime number and $\hat{q}$ is a 160-bit prime number) and picks a generator $\xi \in Z^*_{\hat{p}}$. In this study, a 1024-bit group with a 160-bit prime order subgroup is chosen.

Step 3: The aggregator publishes the system parameters $\{n,\ g,\ g_1, \eta, \hat{p}, \hat{q}, \xi\}$, and the aggregator keeps its private key (q_1, q_2) secret.

4.3. Smart Meter Registration Phase

The smart meter registration process is depicted in Table 5. In the registration phase, the smart meter generates a registration request and sends it to the aggregator. When the aggregator receives the request, it first checks the correctness of the message; if it is correct, the aggregator will store this message in it memory.

First, smart meter M_i generates a private key $x_i \leftarrow Z^*_{\hat{q}}$, then M_i computes the public key $X_i = \xi^{x_i}\ mod\ \hat{p}$ and a signature $\alpha_i = h(X_i||id_i||T_i)$, where T_i is the current timestamp. M_i sends the registration request $\{X_i, T_i, \alpha_i, id_i\}$ to the aggregator over a secure channel.

When aggregator A_i receives $\{X_i, T_i, \alpha_i, id_i\}$, it checks whether $\alpha_i = h(X_i||id_i||T_i)$. If they are equal, A_i stores $\{X_i, id_i\}$.

Table 5. Registration phase of the proposed scheme.

Smart Meter M_i	**Aggregator A_i**
$x_i \leftarrow Z^*_{\hat{p}}$	
$X_i = \xi^{x_i}\ mod\ \hat{p}$	
$\alpha_i = h(X_i\|\|id_i\|\|T_i)$	
$\{X_i, T_i, \alpha_i, id_i\}$	checks if $\alpha_i = h(X_i\|\|id_i\|\|T_i)$
$\xrightarrow{\qquad\qquad}$	stores $\{X_i, id_i\}$

4.4. Reporting Phase

In the reporting phase, the smart meters extract their consumption data and send the encrypted data to the aggregator. When the aggregator receives the data, it will first authenticate and then decrypt the data using its private key. The reporting process is depicted in Table 6.

At the beginning of a reporting cycle, each smart meter generates a noise n_i to perturb its consumption m_i. Then $(m_i + n_i)$ is encrypted by the homomorphic encryption algorithm. The process is as follows:

1. Meter M_i extracts its consumption data m_i, generates a random element $r_i \leftarrow Z^+$, and picks an element $t_i \leftarrow Z^*_{\hat{q}}$.
2. Meter M_i generates noise n_i, which obeys the uniform distribution.
3. Meter M_i computes $c_i = g^{m_i + n_i} \cdot \eta^{r_i}$.
4. Meter M_i computes $d_i = \xi^{t_i}\ mod\ \hat{p}$.
5. Meter M_i gets the signature of c_i and d_i by computing $\phi_i = h_2(id_i, X_i, c_i, d_i, T_i)$; T_i is the current timestamp.
6. Meter M_i computes $e_i = t_i + \phi_i \cdot x_i\ mod\ \hat{q}$.
7. Meter M_i sends $Message1 = \{c_i, d_i, e_i, T_i\}$ to the aggregator.

After receiving the reporting messages from all smart meters, the aggregator A_i first checks the correctness of the incoming messages, then gets the consumption of all the smart meters using Pollard's lambda method, since the total consumption is not a large number in a regular interval [34].

1. Aggregator A_i gets $\phi_i = h_2(id_i, X_i, c_i, d_i, T_i)$.
2. Aggregator A_i picks $s_1, s_2, \ldots .s_k$ at random.
3. Aggregator A_i gets $e = (\sum_1^k e_i \cdot s_i) mod\ \hat{q}$.
4. Aggregator A_i checks if $\xi^e = \prod_{i=1}^k d_i^{s_i} \cdot \prod_{i=1}^k X_i^{\phi_i \cdot s_i}$.
5. If the upper test holds, aggregator A_i gets the electricity consumption by computing $log_{\hat{g}}(\prod_{i=1}^k c_i)^{q_1}$, where $\hat{g} = g^{q_1}$.

Table 6. Proposed aggregation scheme.

Smart Meter M_i	**Aggregator**
Random numbers $r_i \leftarrow Z^+$, $t_i \leftarrow Z_{\hat{q}}^*$	
Gets m_i, generates noise n_i	
$c_i = g^{m_i+n_i} \cdot \eta^{r_i}$	
$d_i = \xi^{t_i}\ mod\ \hat{p}$	
$\phi_i = h_2(id_i, X_i, c_i, d_i, T_i)$	
$e_i = t_i + \phi_i \cdot x_i\ mod\ \hat{q}$	
$\{c_i, d_i, e_i, id_i, T_i\}$	Picks $s_1, s_2, \ldots .s_k$ at random
$\xrightarrow{\qquad\qquad}$	$\phi_i = h_2(id_i, X_i, c_i, d_i, T_i)$
	$e = \left(\sum_1^k e_i \cdot s_i\right) mod\ \hat{q}$
	checks if $\xi^e = \prod_{i=1}^k d_i^{s_i} \cdot \prod_{i=1}^k X_i^{\phi_i \cdot s_i}$
	gets $log_{\hat{g}}(\prod_{i=1}^n c_i)^{q_1}$

The aggregator is able to get the consumption data of all the smart meters as $\sum_{i=1}^k m_i \approx log_{\hat{g}}(\prod_{i=1}^k c_i)^{q_1}$. The following shows the proof of the correctness of the proposed scheme. As $\sum_{i=1}^k n_i \approx 0$ and $\eta^{q_1} = 1$, we can get the following equations:

$$\begin{aligned}
\left(\prod_{i=1}^k c_i\right)^{q_1} &= \left(\prod_{i=1}^k g^{m_i+n_i} \cdot \eta^{r_i}\right)^{q_1} = \prod_{i=1}^k (g^{m_i} \cdot g^{n_i} \cdot \eta^{r_i})^{q_1} \\
&= \prod_{i=1}^k (g^{m_i})^{q_1} \cdot (g^{n_i})^{q_1} \cdot (\eta^{r_i})^{q_1} \\
&= (g^{\sum_{i=1}^k m_i})^{q_1} \cdot (g^{\sum_{i=1}^k n_i})^{q_1} \cdot (\eta^{\sum_{i=1}^k r_i})^{q_1} \\
&= (g^{\sum_{i=1}^k m_i})^{q_1} \cdot (g^{\sum_{i=1}^k n_i})^{q_1} \cdot (\eta^{q_1})^{\sum_{i=1}^k r_i} \\
&= (g^{\sum_{i=1}^k m_i})^{q_1} \cdot (g^{\sum_{i=1}^k n_i})^{q_1} \cdot (1)^{\sum_{i=1}^k r_i} = (g^{\sum_{i=1}^k m_i})^{q_1} \cdot (g^{\sum_{i=1}^k n_i})^{q_1} \\
&\approx (g^{\sum_{i=1}^k m_i})^{q_1} \cdot (g^0)^{q_1} = (g^{\sum_{i=1}^k m_i})^{q_1} \cdot (1)^{q_1} = \hat{g}^{\sum_{i=1}^k m_i}
\end{aligned}$$

Then we can get $\log_{\hat{g}}(\prod_{i=1}^k c_i)^{q_1} = \log_{\hat{g}}(\hat{g}^{\sum_{i=1}^k m_i}) = \sum_{i=1}^k m_i$. Let $\hat{g} = g^{q_1}$; to compute $\sum_{i=1}^k m_i$, it will take $\widetilde{O}(\sqrt{T})$ using Pollard's lambda method ([35], p. 128).

5. Security Analysis

In this section, we conduct a security analysis of the proposed scheme in terms of security against external and internal adversaries, and security of the signature scheme.

5.1. For External Adversaries

As the Boneh–Goh–Nissim homomorphic encryption algorithm is semantically secure, we can get Theorem 1.

Theorem 1. *The proposed scheme achieves semantic security under the chosen cipher attack if and only if the Boneh–Goh–Nissim homomorphic encryption algorithm achieves semantic security.*

($\Rightarrow$) Suppose there is an efficient algorithm $\mathcal{O}_I$ that could break the Boneh–Goh–Nissim homomorphic encryption algorithm in probabilistic polynomial time, which means for a real consumption pair $\{m_1 + n_1, m_2 + n_2\}$ and a cipher $c_i = g^{m_i+n_i} \cdot \eta^{r_i}$ and public parameter *Paras*, an adversary $\mathcal{A}_{\mathcal{I}}$ is able to judge if $m_i + n_i$ is the cipher of $m_1 + n_1$ or $m_2 + n_2$ with a probability that is higher than 1/2.

Given a cipher $c_i = g^{m_i+n_i} \cdot \eta^{r_i}$ and public parameter *Paras*, the adversary $\mathcal{A}_{\mathcal{I}}$ is able to get $m_i + n_i$ by using algorithm $\mathcal{O}_I$. If $m_1 + n_1 = m_i + n_i$, then $m_i + n_i$ is the cipher of $m_1 + n_1$, and if $m_2 + n_2 = m_i + n_i$, then $m_i + n_i$ is the cipher of $m_2 + n_2$. In both situations, the adversary $\mathcal{A}_{\mathcal{I}}$ is able to judge if $m_i + n_i$ is the cipher of $m_1 + n_1$ or $m_2 + n_2$ with a probability that is higher than 1/2. We can conclude that with algorithm $\mathcal{O}_I$, an adversary can break the semantic security of the proposed scheme with a probability that is higher than 1/2.

($\Leftarrow$) Suppose there is an efficient algorithm $\mathcal{O}_{II}$ that could break the proposed scheme in probabilistic polynomial time. Given a cipher $c_i = g^{m_i+n_i} \cdot \eta^{r_i}$ and public parameter *Paras*, adversary $\mathcal{A}_{\mathcal{II}}$ is able to judge if c_i is the cipher of $m_1 + n_1$ or a random number.

If c_i is the cipher of $m_1 + n_1$, for the Boneh–Goh–Nissim homomorphic encryption algorithm, given $C = (g^m \eta^r)^{q_1} = c_i = g^{m_i+n_i} \cdot \eta^{r_i}$, $\mathcal{A}_{\mathcal{II}}$ can get $m = m_i + n_i$. This means $\mathcal{A}_{\mathcal{II}}$ is able to break the algorithm.

5.2. For Internal Adversaries

In the proposed scheme, the smart meter reports $(m_i + n_i)$ to the aggregator m_i represent the real consumption and a noise n_i is randomly generated by the smart meter. Only the smart meter knows n_i, other entities in the system are unable to get n_i, thus they are unable to get the original consumption m_i. The privacy of a single smart meter is protected, only the smart meter knows the real consumption m_i.

5.3. Security of the Signature Scheme

Now we are going to prove that the signature scheme in the proposed schemes is secure. The proof is based on the computational hardness of the discrete logarithm (DL) problem. The discrete logarithm problem for a group G can be stated as:

Given a group G with order q, for $g \in G$ and $a \in < g >$, find an integer x such that $g^x = a$.

Theorem 2. *The signature scheme in the proposed scheme achieves semantic security under the chosen cipher attack if and only if the discrete logarithm problem is unable to be solved in polynomial time.*

($\Rightarrow$) Suppose there is an efficient algorithm $\mathcal{O}_I$ that could break the DL problem in probabilistic polynomial time. This means that for a message pair $\{c_1, c_2\}$ and a signature $e_i = t_i + \phi_i \cdot x_i$, given $\{d_i, id_i, T_i\}$ and public parameter *Paras* included the public key X_i of id_i, an adversary $\mathcal{A}_{\mathcal{I}}$ is able to get:

$$x_i = \mathcal{O}_I(Paras, X_i)$$

$$t_i = \mathcal{O}_I(Paras, d_i)$$

$$\phi_1 = h_2(id_i, X_i, c_1, d_i, T_i)$$

$$\phi_2 = h_2(id_i, X_i, c_2, d_i, T_i)$$

$$e_1 = t_i + \phi_1 \cdot x_i \ mod \ \hat{q}$$

$$e_2 = t_i + \phi_2 \cdot x_i \ mod \ \hat{q}$$

If $e_1 = e_i$, then e_i is the signature of c_1, and if $e_2 = e_i$, then e_i is the signature of c_2; in both situations, the adversary $\mathcal{A}_{\mathcal{I}}$ is able to judge if e_i is the signature of c_1 or c_2 with a probability that is higher than 1/2. We can get the conclusion that algorithm $\mathcal{O}_I$ can break the semantic security of the signature scheme with a probability that is higher than 1/2.

($\Leftarrow$) Suppose there is an efficient algorithm $\mathcal{O}_{II}$ that could break the signature scheme in the proposed scheme. Given a message c_1, a signature $e_i = t_i + \phi_i{\cdot}x_i$, $\{d_i, id_i, T_i\}$ and public parameter $Paras$ included the public key X_i of id_i, adversary $\mathcal{A}_{\mathcal{I}}$ is able to judge if e_i is the signature of c_1 or a random number. If e_i is the signature of c_1, an adversary $\mathcal{A}_{\mathcal{II}}$ is able to get:

$$\phi_1 = h_2(id_i, X_i, c_1, d_i, T_i)$$

$$d_i{\cdot}(X_i)^{\phi_1} = \xi^{t_i+\phi_i\cdot x_i} mod\ \hat{q}$$

This means that with the help of an algorithm $\mathcal{O}_{II}$, given $\{d_i, id_i, T_i\}$ and public parameter $Paras$ included the public key X_i of id_i, the adversary $\mathcal{A}_{\mathcal{II}}$ can get $d_i{\cdot}(X_i)^{\phi_1} = \xi^{t_i+\phi_i\cdot x_i} mod\ \hat{q}$. As for the DL problem, suppose $a = d_i{\cdot}(X_i)^{\phi_1}$ and $g^x = \xi^{t_i+\phi_i\cdot x_i}$. Given $g^x = a$, $\mathcal{A}_{\mathcal{II}}$ can get $x = t_i + \phi_i{\cdot}x_i$. This means the adversary can break the semantic security of the DL problem.

5.4. Security Analysis Using AVISPA

We ran a security check using the constraint-logic -based model-checker [36] and the on-the-fly model-checker (OFMC) [37,38] of Automated Validation of Internet Security Protocols and Applications (AVISPA). The simulation results shown in Table 7 demonstrate that the proposed scheme is safe.

Table 7. Simulation results of AVISPA.

CL-AtSe(Constraint-Logic-based ATtack SEarcher)	**OFMC**
SUMMARY	% OFMC
SAFE	% Version of 2006/02/13
DETAILS	SUMMARY
BOUNDED_NUMBER_OF_SESSIONS	SAFE
TYPED_MODEL	DETAILS
PROTOCOL	BOUNDED_NUMBER_OF_SESSIONS
/home/iotdev/avispa/avispa- 1.1/testsuite/results/smart.if	PROTOCOL
	/home/iotdev/avispa/avispa- 1.1/testsuite/results/ smart.if
GOAL	GOAL
As Specified	as_specified
	BACKEND
BACKEND	OFMC
CL-AtSe	COMMENTS
	STATISTICS
STATISTICS	parseTime: 0.00s
	searchTime: 0.00s
Analysed: 1 states	visitedNodes: 3 nodes
Reachable: 1 states	depth: 2 plies
Translation: 0.00 seconds	
Computation: 0.00 seconds	

6. Comparison

In this section, we compare the computation times for each scheme. The experimental results of different kinds of operations are shown in Table 8. We use the famous Java Pairing-Based Cryptography Library (JPBC) [39]. Type A1 pairings are constructed on the curve $y^2 = x^3 + x$ over the field F_q for some prime $q = 3\ mod\ 4$, and this pairing is symmetric. The order of the group is some prime factor of $(q + 1)$ for the initiation of the curve, the number of primes is set to 2, and the bit length of each prime is set to 160. The parameters for the elliptic curve are listed in Appendix A. The upper bound of Pollard's lambda is set at 100,000.

We chose a 1024-bit modular exponential group with a 160-bit prime order subgroup. The detailed parameters can be found at RFC 5114 [40], and we have listed the parameters in Appendix B.

The experiment environment is a 64-bit Windows 7 Enterprise operating system with Intel(R) Core(TM) i73370K CPU 3.5 GHz processor and 8 GB memory. The code for testing the computation times of different operations has been uploaded to a public repository at github.com [41]. The meanings of different symbols are given below.

G_{bp} bilinear map pairing operation
G_{h2p} hash to an element operation
G_{mul} element multiplication operation
G_{exp} element exponentiation
G_{pol} Pollard's lambda method
GT_{mul} multiplication operation in G_T
D_{exps} exponentiation operation in a modular group with an exponent of 60 bits
D_{exp} exponentiation operation in a modular group with an exponent of 60–160 bits
D_{mul} multiplication operation in a modular group
H_{2b} hash to a big integer operation
H_{256} SHA256 operation

Table 8. Time cost of basic operations (ms).

Type	G_{pol}	G_{bp}	G_{mul}	G_{exp}	G_{h2p}	GT_{mul}	H_{256}	D_{exp}	D_{exps}	D_{mul}	H_{2b}
Time	1235.04	8.4321	0.0257	10.1560	0.6034	0.0036	0.0041	0.36133	0.14087	0.00363	0.00810

6.1. Computation Performance Analysis

We analyzed the computation cost of different schemes at the smart meter registration and aggregation phases. Suppose there are k smart meters in an aggregation system.

For Fan's scheme, in the registration phase, the smart meter has to conduct two G_{exp} and two H_{2b} operations; the aggregator has to conduct one G_{mul}, one H_{2b}, and two G_{exp} operations. In the aggregation phase, the smart meter has to conduct two G_{mul}, two G_{h2p}, and four G_{exp} operations; the aggregator has to conduct one G_{pol}, $(k+1)$ G_{bp}, $(2k-1)$ G_{mul}, $(2k+2)$ G_{exp}, $(k+1)$ G_{h2p}, and $(k-1)$ GT_{mul} operations.

For He's scheme, in the registration phase, the smart meter has to conduct two D_{exp}, one D_{mul}, and one H_{2b} operations; the aggregator has to conduct two D_{exp}, one D_{mul}, and one H_{2b} operations. In the aggregation phase, the smart meter has to conduct two G_{mul}, one G_{h2b}, three G_{exp}, one D_{exp}, one D_{mul} and one H_{2b} operations; the aggregator has to conduct one G_{pol}, k G_{mul}, two G_{exp}, one G_{h2p}, $(k+1)$ D_{exp}, k D_{exps}, $(2k-1)$ D_{mul}, and k H_{2b} operations.

For the proposed scheme, in the registration phase, the smart meter has to conduct one D_{exp} and one H_{256} operation; the aggregator has to conduct one H_{256} operation. In the aggregation phase, the smart meter has to conduct one G_{mul}, two G_{exp}, one D_{exp}, one D_{mul}, and one H_{2b} operations; the aggregator has to conduct one G_{pol}, $(k-1)$ G_{mul}, one G_{exp}, one D_{exp}, $(122k+120)$ D_{mul}, and k H_{2b} operations.

Table 9 shows the computation cost of the registration phase and Table 10 shows the computation cost of the aggregation phase, in which k stands for the number of smart meters in the aggregation system.

Table 9. Computation cost of the registration phase.

Operation	Meter			Aggregator		
	Fan [6]	**He [7]**	**Ours**	**Fan [6]**	**He [7]**	**Ours**
G_{mul}	2	0	0	1	0	0
G_{exp}	0	0	0	2	2	0
D_{exp}	0	2	1	0	0	0
D_{mul}	0	1	1	0	1	0
H_{2b}	2	1	1	1	1	0
H_{256}	0	1	1	0	0	1

Table 10. Computation cost of aggregation phase.

Operation	Meter			Aggregator		
	Fan [6]	**He [7]**	**Ours**	**Fan [6]**	**He [7] ***	**Ours**
G_{pol}	0	0	0	1	1	1
G_{bp}	0	0	0	$k+1$	0	0
G_{mul}	2	2	1	$2k-1$	k	$k-1$
G_{exp}	4	3	2	$2k+2$	2	1
G_{h2p}	2	1	0	$k+1$	1	0
GT_{mul}	0	0	0	$k-1$	0	0
D_{exp}	0	1	1	0	$k+1$	1
D_{exps}	0	0	0	0	k	0
D_{mul}	0	1	1	0	$2k-1$	$122k+120$
G_{h2b}	0	1	1	0	k	k

* The aggregator's computation cost in the modular group of He's scheme is cited directly from their paper.

Table 11 shows the computation costs of different schemes in the registration phase in milliseconds. It is clearly shown in the table that the cost is minimal. This is because the proposed scheme only needs modular exponential group operations and the general SHA-256 operation. These two kinds of operations are both lightweight.

Table 11. Computation cost of registration phase in milliseconds.

Scheme	Smart Meter Side	Aggregator Side
Fan [6]	20.32823	20.3458
He [7]	0.73439	0.73439
Ours	0.36543	0.00410

Figure 3 shows the computation costs of the smart meter side in the aggregation phase. The horizontal axis of this figure is the computation time, and the unit is a millisecond. It is clearly shown in the figure that the computation cost of the proposed scheme is minimal.

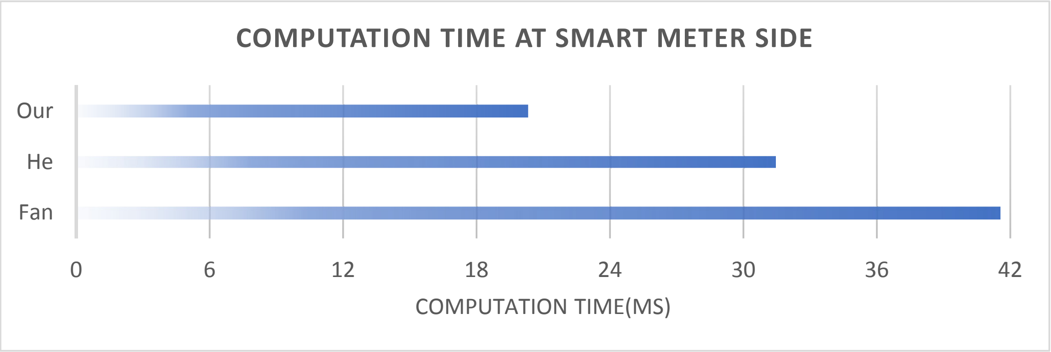

Figure 3. Computation costs at smart meter side in aggregation phase.

Figure 4 shows the computation cost of the aggregator side in the aggregation phase. The vertical axis of this figure indicates the computation time, and the unit is a second; the horizontal axis indicates the number of smart meters. It is clearly shown in the figure that the computation cost of the proposed scheme is minimal under all conditions.

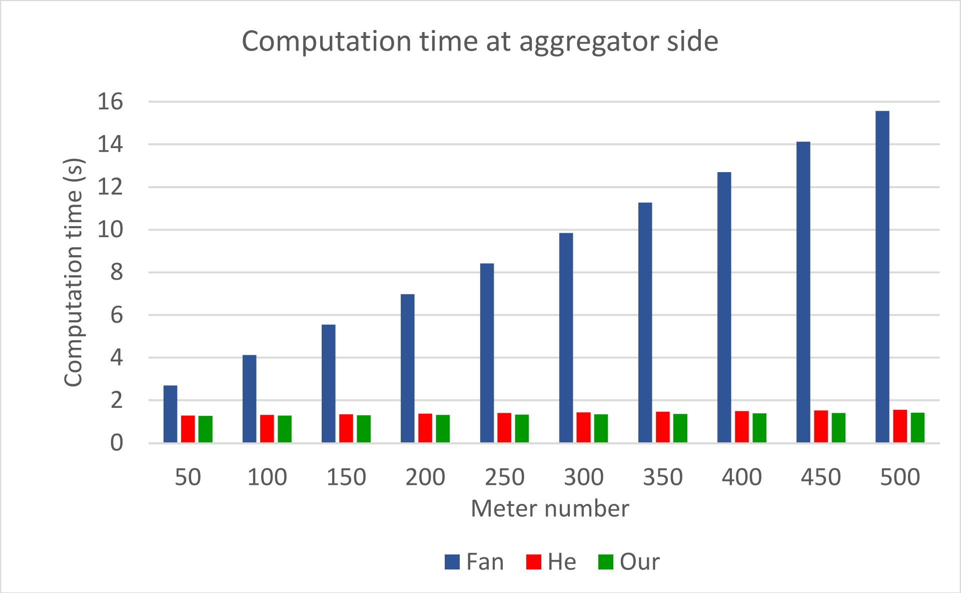

Figure 4. Computation costs at aggregator in aggregation phase.

6.2. Communication Performance Analysis

In this section, we show the communication cost of all the schemes. The lengths of $Z_{\hat{p}}^*, Z_{\hat{q}}^*$ are 1024 bits and 160 bits, respectively. The length of Z^+ is 330 bits, the length of an element of G_1 is 660 bits, the order of the curve is a 320-bit-long number. The size of the timestamp is 32 bits, and the identity is set to be 64 bits long. We analyzed the communication cost of the registration and aggregation phases.

For Fan's scheme, at the registration phase, the smart meter has to send $\{Y_i, \alpha_i, \beta_i, \gamma_i, id_i\}$ to the aggregator, and the bit length of this message is 660 + 660 + 330 + 330 + 64 = 2044. In the aggregation phase, the smart meter has to send $\{id_i, \sigma_i, CT_i\}$ to the aggregator, and the bit length of this message is 64 + 660 + 660 = 1384.

For He's scheme, at the registration phase, the smart meter has to send $\{id_i, X_i, Y_i, \alpha_i\}$ to the aggregator, and the bit length of this message is 64 + 1024 + 1024 + 160 = 2272. In the aggregation phase, the smart meter has to send $\{id_i, c_i, d_i, e_i, T_i\}$ to the aggregator, and the bit length of this message is 64 + 660 + 1024 + 160 + 32 = 1940.

For the proposed scheme, at the registration phase, the smart meter has to send $\{X_i, T_i, \alpha_i, id_i\}$ to the aggregator, and the bit length of this message is 1024 + 32 + 256 + 64 = 1376. In the aggregation phase, the smart meter has to send $\{id_i, c_i, d_i, e_i, T_i\}$ to the aggregator, and the bit length of this message is 64 + 660 + 1024 + 160 + 32 = 1940.

The communication cost of different schemes is shown in Table 12.

Table 12. Communication cost of the schemes.

Scheme	Registration Phase	Aggregation Phase
Fan [6]	2044 bits	1384 bits
He [7]	2272 bits	1940 bits
Ours	1376 bits	1940 bits

6.3. Comparison of All Features

In this section we compare the three schemes in different metrics, and the results are shown in Table 13. As we discussed in Section 3, the schemes of He et al. and Fan et al. have a meter failure problem: when one or more of the smart meters are broken, the scheme fails to work. If we want to add a new smart meter to the system, the whole system needs to be redeployed. Besides, it is a difficult task to replace a broken smart meter with a new one in the other two schemes; if a smart meter is broken, the whole system needs to be redeployed, too. The two schemes also require a higher time accuracy; this means that even if there is only a one millisecond mistake, the aggregator will not get the original data. Moreover, the computation cost of the proposed scheme is the least of the three under all conditions.

Table 13. System comparison.

Comparison	Fan [6]	He [7]	Ours
F1	Difficult	Difficult	Easy
F2	Difficult	Difficult	Easy
F3	×	×	✓
F4	Difficult	Difficult	Easy
F5	Required	Required	Not required
M1	20.3282	0.73439	0.36543
M2	20.3458	0.73439	0.0041
M3	41.8834	31.4965	20.7108
M4	$29.40304k + 1343.6462$	$0.54323k + 1335.6010$	$0.32761k + 1325.3994$
M5	2044	2272	1280
M6	1384	1940	1940

F1: Scalability problem; F2: Adding new smart meters to the system; F3: Meter failure problem; F4: Replacing deployed smart meter with a new one; F5: High accuracy requirement for time; M1: Meter computation cost in registration phase (ms); M2: Aggregator computation cost in registration phase (ms); M3: Meter computation cost in aggregation phase (ms); M4: Aggregator computation cost in aggregation phase (ms); M5: Communication cost of registration phase (bit); M6: Communication cost of aggregation phase (bit).

7. Conclusions

In this study, we first analyzed five noise-generating methods and found that noise obeying uniform distribution is the best for the smart meter privacy protection scenario. We introduced a smart meter aggregation scheme based on the noise addition method and a probabilistic homomorphic encryption algorithm. The proposed scheme can protect the privacy of users and overcome the problems in related works, such as meter replacement problem, meter failure problem, etc. The security analysis shows that the proposed scheme is secure. Besides, by using the noise addition method, we considerably decreased the computation cost of the smart meter side and the aggregator side. Moreover, the authentication process at the aggregator side is accelerated.

Author Contributions: Conceptualization, J.-F.M.; Methodology, Y.C.; Software, Y.C.; Validation, Y.C.; Formal Analysis, Y.C.; Investigation, Y.C., P.C. and L.L.; Resources, J.-F.M.; Data Curation, Y.C.; Writing—Original Draft Preparation, Y.C.; Writing—Review & Editing P.C. and L.L.; Visualization, Y.C.; Supervision, J.-F.M. and L.L.; Project Administration, J.-F.M.; Funding Acquisition, J.-F.M.

Funding: This research was funded by [ARTEMIS-JU] grant number [ARTEMIS-JU 295378], [Ministerio de Industria, Energía y Turismo of Spain] grand number [ART-010000-2012-003] and the [Chinese Scholarship Council (CSC)] grant number [201507040027].

Acknowledgments: The proposal presented in this paper is part of the work made in the European project “Sustainable-Smart Grid Open System for the Aggregated Control, Monitoring and Management of Energy” (e-GOTHAM), funded by ARTEMIS-JU (project code ARTEMIS-JU 295378) and “Ministerio de Industria, Energía y Turismo” of Spain (project code ART-010000-2012-003). This work has also been supported by the Chinese Scholarship Council (CSC) with File No: 201507040027.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The hexadecimal value of the prime is:

p = 17611447540945799965039502289151717146546279638472173362
21820375594574972252514190618931171744649127
n = 143884375334524509518296587329670891720149343451570043
8089722529080535108049439698218081022667197
n_0 = 1349038072991596483339741103753845183283466619883
n_1 = 1066570159991480118361188218992614355059750272759
l = 1224

Appendix B

The hexadecimal value of the prime is:

p = B10B8F96 A080E01D DE92DE5E AE5D54EC 52C99FBC FB06A3C6
9A6A9DCA 52D23B61 6073E286 75A23D18 9838EF1E 2EE652C0
13ECB4AE A9061123 24975C3C D49B83BF ACCBDD7D 90C4BD70
98488E9C 219A7372 4EFFD6FA E5644738 FAA31A4F F55BCCC0
A151AF5F 0DC8B4BD 45BF37DF 365C1A65 E68CFDA7 6D4DA708
DF1FB2BC 2E4A4371

The hexadecimal value of the generator is:

g = A4D1CBD5 C3FD3412 6765A442 EFB99905 F8104DD2 58AC507F
D6406CFF 14266D31 266FEA1E 5C41564B 777E690F 5504F213
160217B4 B01B886A 5E91547F 9E2749F4 D7FBD7D3 B9A92EE1
909D0D22 63F80A76 A6A24C08 7A091F53 1DBF0A01 69B6A28A
D662A4D1 8E73AFA3 2D779D59 18D08BC8 858F4DCE F97C2A24
855E6EEB 22B3B2E5

The generator generates a prime-order subgroup of size:

q = F518AA87 81A8DF27 8ABA4E7D 64B7CB9D 49462353.

References

1. Smart Metering Deployment in the European Union | JRC Smart Electricity Systems and Interoperability. Available online: http://ses.jrc.ec.europa.eu/smart-metering-deployment-european-union (accessed on 15 October 2018).
2. Smart Metering in Europe. Available online: http://www.berginsight.com/ReportPDF/ProductSheet/bi-sm13-ps.pdf (accessed on 15 October 2018).
3. Commission Recommendation of 10 October 2014 on the Data Protection Impact Assessment Template for Smart Grid and Smart Metering Systems', 32014H0724. Available online: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.300.01.0063.01.ENG. (accessed on 19 October 2017).
4. Barbosa, P.; Brito, A.; Almeida, H. Defending Against Load Monitoring in Smart Metering Data through Noise Addition. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 13–17 April 2015; pp. 2218–2224.
5. Barbosa, P.; Brito, A.; Almeida, H. A Technique to provide differential privacy for appliance usage in smart metering. *Inf. Sci.* **2016**, *370–371*, 355–367. [CrossRef]
6. Fan, C.I.; Huang, S.Y.; Lai, Y.L. Privacy-Enhanced Data Aggregation Scheme against Internal Attackers in Smart Grid. *IEEE Trans. Ind. Inf.* **2014**, *10*, 666–675. [CrossRef]
7. He, D.; Kumar, N.; Zeadally, S.; Vinel, A.; Yang, L.T. Efficient and Privacy-Preserving Data Aggregation Scheme for Smart Grid against Internal Adversaries. *IEEE Trans. Smart Grid* **2017**, *8*, 2411–2419. [CrossRef]
8. Chen, Y.; Martínez, J.-F.; Castillejo, P.; López, L. An Anonymous Authentication and Key Establish Scheme for Smart Grid: FAuth. *Energies* **2017**, *10*, 1354. [CrossRef]

9. Bohli, J.M.; Sorge, C.; Ugus, O. A Privacy Model for Smart Metering. In Proceedings of the 2010 IEEE International Conference on Communications Workshops, Cape Town, South Africa, 23–27 May 2010.
10. Wang, S. A Randomized Response Model for Privacy Preserving Smart Metering. *IEEE Trans. Smart Grid* **2012**, *3*, 317–1324. [CrossRef]
11. He, X.; Zhang, X.; Kuo, C.C.J. A Distortion-Based Approach to Privacy-Preserving Metering in Smart Grids. *IEEE Access* **2013**, *1*, 67–78.
12. Soria-Comas, J.; Domingo-Ferrer, J. Optimal data-independent noise for differential privacy. *Inf. Sci.* **2013**, *250*, 200–214. [CrossRef]
13. He, D.; Zeadally, S.; Wang, H.; Liu, Q. Lightweight Data Aggregation Scheme against Internal Attackers in Smart Grid Using Elliptic Curve Cryptography. *Wirel. Commun. Mob. Comput.* **2017**, *2017*. [CrossRef]
14. Garcia, F.D.; Jacobs, B. Privacy-Friendly Energy-Metering via Homomorphic Encryption. In Proceedings of the 6th International Conference on Security and Trust Management, Athens, Greece, 23–24 September 2010.
15. Lu, R.; Liang, X.; Li, X.; Lin, X.; Shen, X. EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications. *IEEE Trans. Parallel Distr. Syst.* **2012**, *23*, 1621–1631. [CrossRef]
16. Busom, N.; Petrlic, R.; Sebé, F.; Sorge, C.; Valls, M. Efficient smart metering based on homomorphic encryption. *Comput. Commun.* **2016**, *82*, 95–101. [CrossRef]
17. Dimitriou, T.; Awad, M.K. Secure and scalable aggregation in the smart grid resilient against malicious entities. *Ad Hoc Netw.* **2016**, *50*, 58–67. [CrossRef]
18. Chatterjee, S.; Das, A.K. An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks. *Secur. Commun. Netw.* **2015**, *8*, 1752–1771. [CrossRef]
19. Jiang, Q.; Wei, F.; Fu, S.; Ma, J.; Li, G.; Alelaiwi, A. Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. *Nonlinear Dyn.* **2016**, *83*, 2085–2101. [CrossRef]
20. Kumari, S.; Chaudhry, S.A.; Wu, F.; Li, X.; Farash, M.S.; Khan, M.K. An improved smart card based authentication scheme for session initiation protocol. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 92–105. [CrossRef]
21. Wu, F.; Xu, L.; Kumari, S.; Li, X. A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. *Comput. Electr. Eng.* **2015**, *45*, 274–285. [CrossRef]
22. Farash, M.S. Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. *Peer-to-Peer Netw. Appl.* **2016**, *9*, 82–91. [CrossRef]
23. Zhang, L.; Tang, S.; Luo, H. Elliptic Curve Cryptography-Based Authentication with Identity Protection for Smart Grids. *PLoS ONE* **2016**. [CrossRef] [PubMed]
24. Saxena, N.; Choi, B.J. Integrated Distributed Authentication Protocol for Smart Grid Communications. *IEEE Syst. J.* **2017**, *12*, 2545–2556. [CrossRef]
25. Nicanfar, H.; Leung, V.C.M. Multilayer Consensus ECC-Based Password Authenticated Key-Exchange (MCEPAK) Protocol for Smart Grid System. *IEEE Trans. Smart Grid* **2013**, *4*, 253–264. [CrossRef]
26. Tsai, J.L.; Lo, N.W. Secure Anonymous Key Distribution Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2016**, *7*, 906–914. [CrossRef]
27. Tsai, J.L.; Lo, N.W. A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services. *IEEE Syst. J.* **2015**, *9*, 805–815. [CrossRef]
28. Saxena, N.; Choi, B.J.; Lu, R. Authentication and Authorization Scheme for Various User Roles and Devices in Smart Grid. *IEEE Trans. Inf. Forensic. Secur.* **2016**, *11*, 907–921. [CrossRef]
29. Odelu, V.; Das, A.K.; Wazid, M.; Conti, M. Provably Secure Authenticated Key Agreement Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2018**, *9*, 1900–1910. [CrossRef]
30. Jo, H.J.; Kim, I.S.; Lee, D.H. Efficient and Privacy-Preserving Metering Protocols for Smart Grid Systems. *IEEE Trans. Smart Grid* **2016**, *7*, 1732–1742. [CrossRef]
31. Boneh, D.; Goh, E.-J.; Nissim, K. Evaluating 2-DNF Formulas on Ciphertexts. In Proceedings of the Theory of Cryptography Conference, Cambridge, MA, USA, 10–12 February.
32. Paillier, P. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999.
33. Okamoto, T.; Uchiyama, S. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology—EUROCRYPT'98*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 308–318.
34. Wiesmann, D.; Lima Azevedo, I.; Ferrão, P.; Fernández, J.E. Residential electricity consumption in Portugal: Findings from top-down and bottom-up models. *Energy Policy* **2011**, *39*, 2772–2779. [CrossRef]

35. Menezes, J.; Van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*, 1st ed.; CRC Press: Boca Raton, FL, USA, 1997; ISBN 0-8493-8523-7.
36. Turuani, M. The CL-Atse Protocol Analyser. In Proceedings of the 17th International Conference on Rewriting Techniques and Applications, RTA, Lecture Notes in Computer Science, Seattle, WA, USA, 12–14 August 2006.
37. Basin, D.; odersheim, S.M.; Vigano, L. *Constraint Differentiation: A New Reduction Technique for Constraint-Based Analysis of Security Protocols*, 1st ed.; ACM Press: New York, NY, USA, 2003; pp. 335–344. Available online: http://www.avispa-project.org (accessed on 26 October 2018).
38. Basin, D.; Mödersheim, S.; Viganò, L. OFMC: A Symbolic Model-Checker for Security Protocols. *Int. J. Inf. Secur.* **2004**, *4*, 181–208. [CrossRef]
39. Available online: http://gas.dia.unisa.it/projects/jpbc/#.Wc0m51uCyUl (accessed on 4 March 2018).
40. Available online: https://www.ietf.org/rfc/rfc5114 (accessed on 4 March 2017).
41. Available online: https://github.com/SevenBruce/JPBC (accessed on 4 March 2018).