energies

MDPI

Article

Privacy-Preserving Electricity Billing System Using Functional Encryption†

Jong-Hyuk Im, Hee-Yong Kwon, Seong-Yun Jeon and Mun-Kyu Lee *

Department of Computer Engineering, Inha University, Incheon 22212, Korea; imjhyuk@gmail.com (J.-H.I.); heeyong.kr@gmail.com (H.-Y.K.); roland.korea@gmail.com (S.-Y.J.)

* Correspondence: mklee@inha.ac.kr; Tel.: +82-32-860-7456

† The present work is an extension of the paper "Jong-Hyuk Im; Yejin Yoon; Mun-Kyu Lee. Secure Electricity Billing Method Using Inner Product Encryption (In Korean)" presented at the 2017 Summer Conference on Information Security and Cryptography, Asan, Korea, 22–23 June 2017.

Received: 16 February 2019; Accepted: 25 March 2019; Published: 1 April 2019

Abstract: The development of smart meters that can frequently measure and report power consumption has enabledelectricity providers to offer various time-varying rates, including time-of-use and real-time pricing plans. High-resolution power consumption data, however, raise serious privacy concerns because sensitive information regarding an individual's lifestyle can be revealed by analyzing these data. Although extensive research has been conducted to address these privacy concerns, previous approaches have reduced the quality of measured data. In this paper, we propose a new privacy-preserving electricity billing method that does not sacrifice data quality for privacy. The proposed method is based on the novel use of functional encryption. Experimental results on a prototype system using a real-world smart meter device and data prove the feasibility of the proposed method.

Keywords: smart meter; electricity billing; privacy; functional encryption; non-intrusive appliance load monitoring (NIALM); disaggregation

1. Introduction

The traditional electricity metering and billing method, in particular for residential consumers, involved installing simple electromechanical meters that could be read manually. Meters were typically read monthly, and customers were charged a rate proportional to their monthly usage. In this situation, the only option an electricity provider could offer was a "flat" or "fixed" rate, with only slight variations such as increasing the unit price as consumption increases over the course of the billing period [1]. A typical implementation of this policy is a "tiered" rate plan [2].

With the development and extensive deployment of smart meters, however, providers are now offering various time-varying rates. The time-of-use (TOU) rate plan divides the day into time periods and sets a different rate for each period [1]. There are other pricing variations as well, such as critical peak pricing (CPP) and peak-time rebates (PTR). The most advanced type of plan is real-time pricing (RTP), which allows price change per hour or even half hour [1,3]. The billing formula for these nonflat rates can be simplified as follows:

$$\sum_{i=0}^{n-1} r_i u_i, \tag{1}$$

where n is the number of unit time periods per billing period, r_i is the electricity price for time period i, and u_i is the amount of electricity consumed in time period i. Alternatively, the electricity charge can be represented as the inner product:

$$\langle u, r \rangle \tag{2}$$

of two n-dimensional vectors, $r = (r_0, \ldots, r_{n-1})$ and $u = (u_0, \ldots, u_{n-1})$.

When smart meters measure and report power consumption data more frequently, consumers can adopt more fine-grained plans regarding their usage. Currently deployed smart meters already deliver load data with very high resolution, e.g., at five-minute intervals [4]. However, these detailed power consumption data raise serious privacy concerns, because personal data can be inferred from the energy use profiles measured by smart meters [5]. Recent advances in non-intrusive appliance load monitoring (NIALM) and disaggregation techniques make it possible to extract the energy consumption statistics of an individual appliance from aggregated data involving many appliances [6–13]. Although the primary goal of these algorithms is to provide the consumer with useful energy feedback, such NIALM analyses might be misused, compromising privacy by monitoring the consumer's appliance usage patterns [14]. Advanced NIALM analysis can reveal significant information regarding the persons in a household, such as their presence, sleep schedule, and meal times [15]. In extreme cases, if the sampling interval is sufficiently small, even the television channel that is being watched can be identified, along with audiovisual content [16]. According to the quantitative analysis in Reference [14], to guarantee that all appliances are privacy-safe, the measurement interval of a smart meter should be at least one hour, which is significantly longer than the time resolution of state-of-the-art smart meters.

Consequently, there has been extensive research to resolve this privacy issue [17–19]. Previous approaches can be classified into the following three categories: (i) Providing only low-resolution data, i.e., data with decreased time granularity [14,17,18,20], (ii) perturbation or obfuscation of the measured data by adding controlled noise [21–23], and (iii) aggregation of data from multiple smart meters using a trusted third party [21], masking [24,25], or additive homomorphic encryption [23,26,27]. However, all of the above methods essentially reduce the quality of the measured data. Consequently, they cannot be used for granular billing that requires high-resolution metering data from a single smart meter. As such, there is a tradeoff between the functionality of smart meters and the privacy of customers.

In this paper, we aim at achieving both functionality and privacy by taking a completely different approach to privacy-preserving billing methods. Our method allows a smart meter to send the provider all measured data with full granularity and without privacy leaks. Our proposal is based on the novel use of a recently developed advanced cryptographic primitive, viz. the functional encryption (FE) algorithm [28–30]. Using the proposed system, a smart meter encrypts the measured consumption data $u = (u_0, \ldots, u_{n-1})$ and sends them to the electricity provider. The provider is provided with a restricted decryption key associated with $r = (r_0, \ldots, r_{n-1})$, with which it cannot directly recover u, and only obtains the weighted sum (Equation (2)). This approach naturally resolves the privacy issue because the provider never sees the individual consumption statistic u_i for each time period. To verify the feasibility of the proposed method, we implemented a prototype billing system composed of a smart meter, a provider's billing server, and a regulatory agency. In particular, we used an off-the-shelf smart meter device to represent a real-world scenario. The proposed system does not require special-purpose hardware. It is realized merely through a software update of the smart meter. Our experimental results with real measurement data prove that the proposed system performs well with currently deployed smart meters, eliminating the need to decrease data granularity. According to the experimental results, only 0.5 s are required for the entire procedure, including the tasks for the smart meter to encrypt the consumption data u and for the provider to compute the weighted sum. It should be noted that the proposed method does not render previous methods obsolete; rather, it can be combined for advanced services. For example, it may be possible to use the new method for billing and either an aggregation or perturbation method for power generation and distribution network control.

2. Preliminaries: Function-Hiding Inner Product Functional Encryption

Functional encryption (FE)is an encryption schemethat supports operations on encrypted data [28,29]. In FE schemes, the owner of the master key can delegate arbitrary secret keys that allow decryptors to learn only specific functions of the data. For example, given a ciphertext of a message x and a secret key restricted to a function f, the decryptor only receives the value $f(x)$, and does not learn anything about x. If the data $\boldsymbol{x}$ of FE are represented as a vector and the function f is defined as the inner product of $\boldsymbol{x}$ with a predefined vector $\boldsymbol{v}$, then a decryptor can recover the inner product $\langle \boldsymbol{v}, \boldsymbol{x} \rangle$ by inputting the ciphertext of vector $\boldsymbol{x}$ and the restricted secret key associated with $\boldsymbol{v}$. This FE scheme is called an inner product encryption (IPE) scheme [30–34].

Some IPE schemes provide an additional property, viz. function hiding. In these schemes, both $\boldsymbol{x}$ and $\boldsymbol{v}$ are kept secret from the decryptor, even though the decryptor possesses the secret key associated with $\boldsymbol{v}$. In 2015, Bishop et al. first proposed function-hiding inner product encryption (FHIPE), which considered these capabilities [31]. Moreover, extensive research is currently under way to improve the performance of the FHIPE [30,32–34]. In this paper, we used the practical scheme Π_{IPE} proposed by Kim et al. in 2018 [30]. The Π_{IPE} scheme consists of four probabilistic polynomial time (PPT) algorithms, Setup, KeyGen, Encrypt and Decrypt, as follows. For more details regarding the operations using bilinear groups, refer to Reference [30].

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{pp}, \mathsf{msk})$: Given a security parameter λ, the setup algorithm Setup outputs the public parameters pp and the master secret key msk corresponding to λ. More precisely, the setup algorithm samples an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two distinct groups of prime order q, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a function that maps two elements from $\mathbb{G}_1$ and $\mathbb{G}_2$ onto a target group $\mathbb{G}_T$, also of prime order q. The setup algorithm then chooses generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. Next, the algorithm samples $\mathbf{B} \leftarrow \mathbb{GL}_n(\mathbb{Z}_q)$, where $\mathbb{GL}_n(\mathbb{Z}_q)$ is the general linear group of $(n \times n)$ matrices over $\mathbb{Z}_q$. Throughout the paper, bold uppercase letters, e.g., $\mathbf{B}$, refer to matrices. Then, the algorithm sets $\mathbf{B}^\star = \det(\mathbf{B}) \cdot (\mathbf{B}^{-1})^\top$, where $\det(\mathbf{B})$ and $(\mathbf{B}^{-1})^\top$ denote the determinant of $\mathbf{B}$ and the transpose matrix of $\mathbf{B}^{-1}$, respectively. Finally, the setup algorithm outputs the public parameters $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ and the master secret key $\mathsf{msk} = (\mathsf{pp}, g_1, g_2, \mathbf{B}, \mathbf{B}^\star)$.
- $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{v}) \rightarrow SK_{\boldsymbol{v}}$: Given the master secret key msk and a vector $\boldsymbol{v} = (v_0, \ldots, v_{n-1})$, the key generation algorithm KeyGen chooses a uniformly random element $\alpha \in \mathbb{Z}_q$ and outputs a secret key:

$$SK_{\boldsymbol{v}} = (K_1, K_2) = (g_1^{\alpha \cdot \det(\mathbf{B})}, g_1^{\alpha \cdot \boldsymbol{v} \cdot \mathbf{B}}).$$

  Note that the second component of $SK_{\boldsymbol{v}}$ is a vector of group elements. That is, according to the notation in Reference [30], for a group element $g \in \mathbb{G}_1$ and a row vector $\boldsymbol{u} = (u_0, \ldots, u_{n-1})$, $g^{\boldsymbol{u}}$ denotes the vector of group elements, $(g^{u_0}, \ldots, g^{u_{n-1}})$.
- $\mathsf{Encrypt}(\mathsf{msk}, \boldsymbol{x}) \rightarrow CT_{\boldsymbol{x}}$: Given the master secret key msk and an input vector $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$, the encryption algorithm Encrypt chooses a uniformly random element $\beta \in \mathbb{Z}_q$ and outputs a ciphertext:

$$CT_{\boldsymbol{x}} = (C_1, C_2) = (g_2^{\beta}, g_2^{\beta \cdot \boldsymbol{x} \cdot \mathbf{B}^\star}).$$

- $\mathsf{Decrypt}(\mathsf{pp}, SK_{\boldsymbol{v}}, CT_{\boldsymbol{x}}) \rightarrow z$: Given the public parameters pp, a secret key $SK_{\boldsymbol{v}} = (K_1, K_2)$, and a ciphertext $CT_{\boldsymbol{x}} = (C_1, C_2)$, the decryption algorithm Decrypt computes:

$$D_1 = e(K_1, C_1) \text{ and } D_2 = e(K_2, C_2).$$

  Then, the algorithm checks whether there exists z, such that $(D_1)^z = D_2$, and either outputs z or an error symbol implying that decryption is impossible. If there is such z, it satisfies $z = \langle \boldsymbol{v}, \boldsymbol{x} \rangle$. In other words, z is the inner product of $\boldsymbol{v}$ and $\boldsymbol{x}$.

An important property of the above FHIPE is that the decryptor computes $z = \langle \boldsymbol{v}, \boldsymbol{x} \rangle$ from $SK_{\boldsymbol{v}}$ and $CT_{\boldsymbol{x}}$ but does not learn anything about either $\boldsymbol{v}$ or $\boldsymbol{x}$. It should be noted that the roles of KeyGen and

Encrypt are symmetric. This symmetry is used to design our system, as explained in the next section (see Section 3.3).

3. Privacy-Preserving Electricity Billing System

3.1. System Model

Figure 1 shows the proposed system model. We considered a system involving three parties: An energy service provider (ESP), a smart meter, and a regulatory agency (RA). The ESP is a company that wants to collect electricity charges in return for the electricity used by smart meter owners. The smart meter is a device that periodically reports the power consumption of its owner, who agrees to pay the electricity charges, yet prefers to hide power usage patterns from the ESP. Because electric power companies are generally monopolies, it is common for RAs to approve prices for electricity [35]. Our model reflects this situation and considers the RA to be a trusted third party with the following roles: (i) Generating the master secret key and public parameters for FHIPE at the registration stage of a smart meter, and (ii) encoding the electricity price for each time period according to the request of the ESP. We remark that the security of the proposed system relies on the trusted RA. That is, if the RA is compromised, the privacy of users may be invaded. Therefore, appropriate technical measures should be provided to protect the RA. In addition, we should consider the possibility that the RA might collude with the ESP. However, as the examples of RA, we considered government agencies such as the Korea Electricity Regulatory Commission or the US Federal Energy Regulatory Commission [36,37]. These government agencies aim to protect consumers' rights and interests [36] and assist consumers in obtaining economically efficient, safe, reliable, and secure energy services at a reasonable cost through appropriate regulatory and market means [37]. Therefore, it is reasonable to assume that they would not collude with the ESP. Because RA is a trusted party, it never deviates from the protocol. We assume that the smart meter is tamper-proof, and accurately measures and reports the electricity usage. Finally, we assume that ESP is honest-but-curious, i.e., it performs the billing protocol honestly and correctly, but might try to extract useful information about the electricity usage patterns if the data from the smart meter are not encrypted.

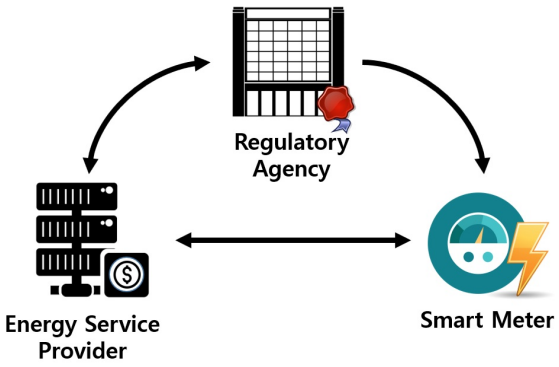

Figure 1. Proposed system model.

3.2. Representation of Power Consumption Data

For our billing system, we generalized the billing Formulas (1) and (2) as follows. We first defined a *reporting period* as the time interval at which the smart meter reports the measured data to the ESP. This is not necessarily the same as the measurement interval. Therefore, let n be the number of measurements of power consumption data in each reporting period. For example, if the length of a reporting period is two hours and the measurement interval is 15 min, then $n = 8$. Let u_i be the electricity consumption measured at the i-th measurement interval in a reporting period. Let r_i be

the electricity rate at the i-th interval. Then, the electricity charge c for this reporting period can be determined by the inner product:

$$c = \langle \boldsymbol{u}, \boldsymbol{r} \rangle, \tag{3}$$

of the rate vector $\boldsymbol{r} = (r_0, \ldots, r_{n-1})$ with the consumption vector $\boldsymbol{u} = (u_0, \ldots, u_{n-1})$.

3.3. Redefining the Roles of KeyGen and Encrypt

In this paper, we used the FHIPE scheme proposed in Reference [30]. As explained in Section 2, the algorithms $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{v})$ and $\mathsf{Encrypt}(\mathsf{msk}, \boldsymbol{x})$ hide the vectors $\boldsymbol{v}$ and $\boldsymbol{x}$, respectively, from the decryptor. Although KeyGen was named "key generation" and its result $SK_{\boldsymbol{v}}$ was defined as a secret key associated with $\boldsymbol{v}$, $SK_{\boldsymbol{v}}$ can also be viewed as a ciphertext of $\boldsymbol{v}$. Then, the decryption algorithm essentially computes the inner product of the two hidden vectors $\boldsymbol{v}$ and $\boldsymbol{x}$ given the two input ciphertexts $SK_{\boldsymbol{v}}$ and $CT_{\boldsymbol{x}}$. This property was already mentioned in Reference [30] as a building block to construct a general two-input functional encryption. In Reference [38] (the full version of [30]), KeyGen and Encrypt were redefined as "left" and "right" encryption algorithms, respectively. Moreover, it was estimated in Reference [30] that the speed of KeyGen was faster than that of Encrypt by 1.8 to 5.3 times. Because it is reasonable to assume that the smart meter is the most resource-constrained among the three parties in Figure 1, we designed our protocol such that the smart meter can hide the measured power consumption data vector by performing the lighter KeyGen algorithm (i.e., the left encryption), instead of Encrypt (i.e., the right encryption).

3.4. Proposed Privacy-Preserving Electricity Billing Protocol

In this section, we present the privacy-preserving electricity billing protocol, Π_{PPB}, for our system. The protocol comprises two stages: The registration stage and the reporting stage.

3.4.1. Registration Stage of Proposed Protocol

Figure 2 shows the registration stage of our protocol. This stage begins with the RA performing the Setup algorithm of FHIPE. That is, the RA generates a master secret key msk and public parameters pp satisfying the security parameter λ for the smart meter. Next, the RA sets an identifier ID and n, the number of measurements of power consumption per reporting period. In addition, the measurement interval, m, is also set. For example, if the measurement interval m is 15 min and $n = 8$, the reporting period will be two hours. This information is also stored in the smart meter. The above process is performed when the smart meter is deployed and is marked with the red dotted box in Figure 2. Next, for every billing period, e.g., a month, the smart meter subscribes to an electricity rate plan P and transmits the tuple $(\mathsf{ID}, \mathsf{pp}, n, m, \mathsf{P})$ to the ESP. Subscriptions to rate plans can be changed as often as desired after deployment. In this paper, we considered the electricity rate plans capable of dividing a day into multiple time periods, as with TOU or RTP. Finally, the ESP stores the tuple $(\mathsf{ID}, \mathsf{pp}, n, m, \mathsf{P})$.

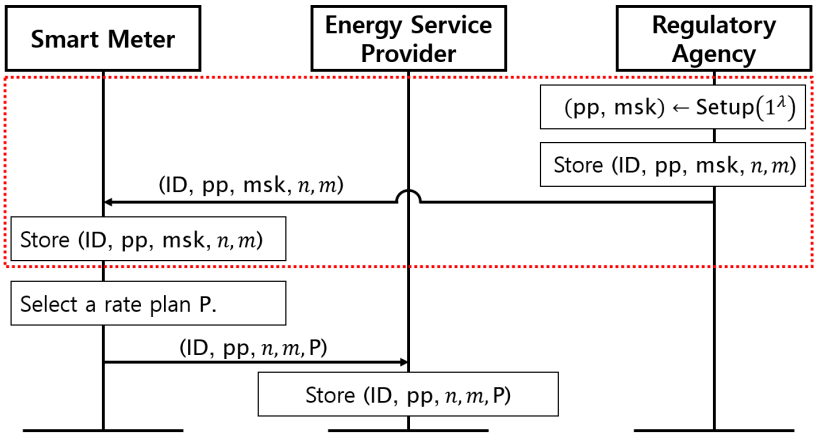

Figure 2. Registration stage of our protocol.

3.4.2. Reporting Stage of Proposed Protocol

Figure 3 shows the reporting stage of our protocol. This stage should only be performed after completing the registration stage. In the reporting stage, the smart meter measures electricity consumption $\boldsymbol{u} = (u_0, \ldots, u_{n-1})$ during a certain reporting period rp. Specifically, the smart meter measures the consumption n times during rp and represents the measured data to vector $\boldsymbol{u}$. The smart meter then encodes this vector to U using the EncodeUsage algorithm, which is defined as $\mathsf{EncodeUsage}(\mathsf{msk}, \boldsymbol{u}) = \mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{u})$. That is, EncodeUsage performs the left encryption algorithm of FHIPE using the master secret key msk. Next, the smart meter transmits the tuple $(\mathsf{ID}, \mathsf{rp}, U)$ to the ESP. Upon receiving the tuple, the ESP generates an n-dimensional rate vector $\boldsymbol{r} = (r_0, \ldots, r_{n-1})$ for rp according to the rate plan P and transmits ID and $\boldsymbol{r}$ to the RA. The RA checks whether the claimed rate $\boldsymbol{r}$ is reasonable and approves $\boldsymbol{r}$ by encoding it using msk associated with ID. For this purpose, the RA performs EncodeRate, which is defined as $\mathsf{EncodeRate}(\mathsf{msk}, \boldsymbol{r}) = \mathsf{Encrypt}(\mathsf{msk}, \boldsymbol{r})$. That is, EncodeRate performs the right encryption algorithm of FHIPE. Note that EncodeRate does not need to be done for every reporting period unless the rate changes frequently. In this case, EncodeRate can be performed once in advance, and the result R can be used for many reporting periods. However, we designed our protocol, as shown in Figure 3, to cover even the most dynamic rate plans (e.g., RTP) where the price changes in real time. Smart meter users can optimize their power usage while continuously monitoring real-time price fluctuations during the reporting period. Immediately after the end of the reporting period, the ESP generates a rate vector $\boldsymbol{r}$ that reflects the tariff for the most recent reporting period. Upon receiving the approval from the RA, i.e., the encoded rate R, the ESP calculates the electricity charge c by performing the FHIPE decryption using pp, U, and R. Finally, the ESP adds the charge c for the reporting period rp to the bill of the smart meter, which has ID as an identifier. This stage is performed once each reporting period.

Importantly, the ESP does not learn anything about individual u_i even though it is given U and can recover the charge (Equation (3)) by decrypting U. That is, our security objective is met according to the nature of FHIPE, which is proven in the next section. The FHIPE scheme [30] also protects r_i from a decryptor through right encryption. However, we do not require this property because $\boldsymbol{r}$ does not need to be secret. In fact, it is generated by the decryptor itself, viz. the ESP, in our protocol.

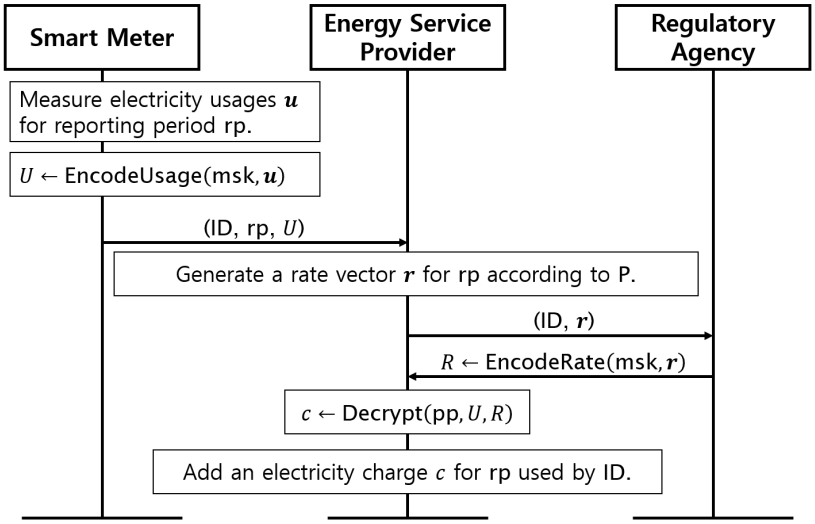

Figure 3. Reporting stage of our protocol.

4. Security Analysis

In this section, we review the security properties of the Π_{IPE} scheme proposed in Reference [30] and prove the security of the proposed method using reduction from Π_{IPE}.

4.1. Review of Security for the FHIPE Scheme Π_{IPE}

Existing inner product encryption schemes, including the Π_{IPE} scheme [30] we use, considered an indistinguishability notion of security [30–34]. Here, we review the security notion for Π_{IPE}. In Reference [30], an experiment between a challenger and an adversary $\mathcal{A}$ that can make key generation and encryption oracle queries is defined as follows:

Definition 1 (Experiment $\mathsf{Expt}_b^{\mathsf{IPE-IND}}$ [30]). *Let $b \in \{0,1\}$. The challenger computes $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, gives pp to the adversary $\mathcal{A}$, and then responds to each oracle query type made by $\mathcal{A}$ in the following manner.*

- **Key generation oracle.** *On inputting a pair of vectors $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{Z}_q^n \setminus \{0\}$, the challenger computes and returns $SK \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{x}_b)$.*
- **Encryption oracle.** *On inputting a pair of vectors $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathbb{Z}_q^n \setminus \{0\}$, the challenger computes and returns $CT \leftarrow \mathsf{Encrypt}(\mathsf{msk}, \boldsymbol{y}_b)$.*

Eventually, $\mathcal{A}$ outputs a bit b', which is also the output of the experiment, denoted by $\mathsf{Expt}_b^{\mathsf{IPE-IND}}(\mathcal{A})$.

Then, the security of an FHIPE scheme is defined using an indistinguishability notion as follows:

Definition 2 (Admissibility of $\mathcal{A}$ [30]). *For an adversary $\mathcal{A}$, let Q_1 and Q_2 be the total number of key generation and encryption oracle queries made by $\mathcal{A}$, respectively. For $b \in \{0,1\}$, let $\boldsymbol{x}_b^{(1)}, \ldots, \boldsymbol{x}_b^{(Q_1)} \in \mathbb{Z}_q^n \setminus \{0\}$ and $\boldsymbol{y}_b^{(1)}, \ldots, \boldsymbol{y}_b^{(Q_2)} \in \mathbb{Z}_q^n \setminus \{0\}$ be the corresponding vectors that $\mathcal{A}$ submits to the key generation and encryption oracles, respectively. We say that $\mathcal{A}$ is admissible if for all $i \in [Q_1]$ and $j \in [Q_2]$, and we have that:*

$$\langle \boldsymbol{x}_0^{(i)}, \boldsymbol{y}_0^{(j)} \rangle = \langle \boldsymbol{x}_1^{(i)}, \boldsymbol{y}_1^{(j)} \rangle.$$

Definition 3 (IND-Security for IPE [30]). *We define an inner product encryption scheme denoted as $\Pi_{IPE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as fully-secure if for all efficient and admissible adversaries $\mathcal{A}$:*

$$\left| \Pr[\mathsf{Expt}_0^{\mathsf{IPE-IND}}(\mathcal{A}) = 1] - \Pr[\mathsf{Expt}_1^{\mathsf{IPE-IND}}(\mathcal{A}) = 1] \right| = \mathsf{negl}(\lambda),$$

where $\mathsf{negl}(\lambda)$ *denotes a negligible function in* λ.

Theorem 1 ([30]). *The inner product encryption scheme* Π_{IPE} *is* IND*-secure in the generic group model.*

Remark 1. *The original statement in Theorem 7 in Reference [30] is that* Π_{IPE} *is* SIM*-secure in the generic group model. It was also remarked in Remark 5 in Reference [30] that a* SIM*-secure scheme is also* IND*-secure. We merged these two statements into the above theorem. For more details regarding the* SIM*-security and a generic group model, refer to Reference [30].*

4.2. Security for the Proposed Privacy-Preserving Electricity Billing Protocol Π_{PPB}

In the system model of the proposed system, we assumed that an ESP is honest-but-curious. That is, the ESP might attempt to extract useful information regarding the consumption vector $\boldsymbol{u}$. As our security goal against the honest-but-curious ESP, we defined an indistinguishability notion of security (IND-security) for Π_{PPB}. Then, we proved the IND-security of Π_{PPB} using that of Π_{IPE}.

We began by defining the following experiment between a challenger and an adversary $\mathcal{A}^*$ that can make usage encoding and rate encoding oracle queries. Although the experiment is designed similarly to that in Definition 1, the adversary provides the rate encoding oracle with only a single vector $\boldsymbol{r}$, instead of a pair of vectors. This definition reflects the situation where the ESP already knows $\boldsymbol{r}$.

Definition 4 (Experiment $\mathsf{Expt}_b^{\mathsf{PPB-IND}}$). *Let* $b \in \{0,1\}$. *The challenger computes* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *using* Π_{IPE}, *gives* pp *to the adversary* $\mathcal{A}^*$, *and then responds to each oracle query type made by* $\mathcal{A}^*$ *in the following manner.*

- **Usage encoding oracle.** *On inputting a pair of vectors* $\boldsymbol{u}_0, \boldsymbol{u}_1 \in \mathbb{Z}_q^n \setminus \{0\}$, *the challenger computes and returns* $U \leftarrow \mathsf{EncodeUsage}(\mathsf{msk}, \boldsymbol{u}_b)$ *using* $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{u}_b)$ *of* Π_{IPE}.
- **Rate encoding oracle.** *On inputting a vector* $\boldsymbol{r} \in \mathbb{Z}_q^n \setminus \{0\}$, *the challenger computes and returns* $R \leftarrow \mathsf{EncodeRate}(\mathsf{msk}, \boldsymbol{r})$ *using* $\mathsf{Encrypt}(\mathsf{msk}, \boldsymbol{r})$ *of* Π_{IPE}.

Eventually, $\mathcal{A}^*$ *outputs a bit* b', *which is also the output of the experiment, denoted by* $\mathsf{Expt}_b^{\mathsf{PPB-IND}}(\mathcal{A}^*)$.

Then, the security of a billing scheme is defined using an indistinguishability notion as follows:

Definition 5 (Admissibility of $\mathcal{A}^*$). *For an adversary* $\mathcal{A}^*$, *let* Q_1 *and* Q_2 *be the total number of usage encoding and rate encoding oracle queries made by* $\mathcal{A}^*$, *respectively. For* $b \in \{0,1\}$, *let* $\boldsymbol{u}_b^{(1)}, \ldots, \boldsymbol{u}_b^{(Q_1)} \in \mathbb{Z}_q^n \setminus \{0\}$ *and* $\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(Q_2)} \in \mathbb{Z}_q^n \setminus \{0\}$ *be the corresponding vectors that* $\mathcal{A}^*$ *submits to the usage encoding and rate encoding oracles, respectively. We say that* $\mathcal{A}^*$ *is admissible if for all* $i \in [Q_1]$ *and* $j \in [Q_2]$, *and we have that:*

$$\langle \boldsymbol{u}_0^{(i)}, \boldsymbol{r}^{(j)} \rangle = \langle \boldsymbol{u}_1^{(i)}, \boldsymbol{r}^{(j)} \rangle.$$

Definition 6 (IND-Security for Π_{PPB}). *We define a privacy-preserving electricity billing protocol* Π_{PPB} *as fully-secure if for all efficient and admissible adversaries* $\mathcal{A}^*$:

$$\left| \Pr[\mathsf{Expt}_0^{\mathsf{PPB-IND}}(\mathcal{A}^*) = 1] - \Pr[\mathsf{Expt}_1^{\mathsf{PPB-IND}}(\mathcal{A}^*) = 1] \right| = \mathsf{negl}(\lambda).$$

Theorem 2. *If* Π_{IPE} *is* IND*-secure (according to Definition 3) in the generic group model, then* Π_{PPB} *that is defined using* Π_{IPE} *is* IND*-secure (according to Definition 6) in the generic group model.*

Proof of Theorem 2. To conduct a reduction proof, assume that there exists an efficient and admissible adversary $\mathcal{A}^*$. We showed that $\mathcal{A}^*$ can be used as a subroutine for the adversary $\mathcal{A}$. We designed $\mathcal{A}$ such that it can simulate usage encoding and rate encoding oracles by forwarding $\mathcal{A}^*$'s corresponding queries to key generation and encryption oracles, respectively.

Algorithm 1 shows our construction of $\mathcal{A}$. It is straightforward to see that if $\mathcal{A}^*$ is an admissible, polynomial time algorithm, Algorithm 1 is so, too. In addition, $\mathcal{A}$'s advantage is the same as that of $\mathcal{A}^*$. That is, $\left|\Pr[\mathsf{Expt}_0^{\mathsf{IPE-IND}}(\mathcal{A}) = 1] - \Pr[\mathsf{Expt}_1^{\mathsf{IPE-IND}}(\mathcal{A}) = 1]\right| = \left|\Pr[\mathsf{Expt}_0^{\mathsf{PPB-IND}}(\mathcal{A}^*) = 1] - \Pr[\mathsf{Expt}_1^{\mathsf{PPB-IND}}(\mathcal{A}^*) = 1]\right|$. However, this construction contradicts Theorem 1, which states that an admissible $\mathcal{A}$ with non-negligible advantage does not exist. This completes the proof. □

Algorithm 1 Construction of $\mathcal{A}$ using $\mathcal{A}^*$

Input: public parameters pp.
Output: a bit b'.

Give pp to $\mathcal{A}^*$.
while true **do**
  if $\mathcal{A}^*$ returns b' **then**
    Return b'.
  Wait until $\mathcal{A}^*$ submits a query $\mathcal{Q}$.
  if $\mathcal{Q} = (\boldsymbol{u}_0, \boldsymbol{u}_1)$ is a usage encoding oracle query **then**
    Set $\boldsymbol{x}_0 \leftarrow \boldsymbol{u}_0$ and $\boldsymbol{x}_1 \leftarrow \boldsymbol{u}_1$.
    Submit $(\boldsymbol{x}_0, \boldsymbol{x}_1)$ to the key generation oracle and receive SK.
    Provide SK to $\mathcal{A}^*$.
  else ▷ $\mathcal{Q} = \boldsymbol{r}$ is a rate encoding oracle query.
    Set $\boldsymbol{y}_0 \leftarrow \boldsymbol{r}$ and $\boldsymbol{y}_1 \leftarrow \boldsymbol{r}$.
    Submit $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ to the encryption oracle and receive CT.
    Provide CT to $\mathcal{A}^*$.

5. Experimental Results

In this section, we verify the feasibility of the proposed privacy-preserving billing system through implementation on a real-world smart meter device. To implement the system presented in Figure 1, we constructed an ESP and RA on separate servers. Both servers were equipped with an Intel Core i7-7700 CPU (3.60 GHz) and 16 GB RAM. For the smart meter, we used *DS-125 Aggregator*, a smart meter device manufactured and deployed by RETIGRID, a company providing smart grid solutions in Korea. This device was equipped with an ARM Cortex-A8 processor and 512 MB RAM. The software for the RA, ESP, and the smart meter was implemented in the C++ programming language. We used the Pairing-Based Cryptography library (PBC) [39] for cryptographic operations involving bilinear map computation e, as well as the GNU Multiple Precision Arithmetic Library (GMP) [40] for big-number arithmetic operations, and the Library for doing Number Theory (NTL) [41] for operations over a finite field, vector, and matrix.

The decryption operation to find z satisfying $(D_1)^z = D_2$ essentially solves a discrete logarithm problem for a small restricted space for z. To accelerate this process, we used the baby-step giant-step method [42].

Figure 4 shows the software and hardware components of the RA, ESP, and the smart meter. The storage and network modules were in common, and DS-125 Aggregator contained a measurement module to measure the amount of consumed electricity. The RA, ESP, and smart meter also differed with regard to their FHIPE modules. Although their subcomponents were common and all used PBC, GMP, and NTL, their high-level functions were different, as explained in Section 3.4. That is, the FHIPE module in RA performed $\mathsf{Setup}(1^\lambda)$ and $\mathsf{EncodeRate}(\mathsf{msk}, \boldsymbol{r}) = \mathsf{Encrypt}(\mathsf{msk}, \boldsymbol{r})$, whereas the FHIPE module in the ESP performed $\mathsf{Decrypt}(\mathsf{pp}, U, R)$, and that in the DS-125 Aggregator performed $\mathsf{EncodeUsage}(\mathsf{msk}, \boldsymbol{u}) = \mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{u})$.

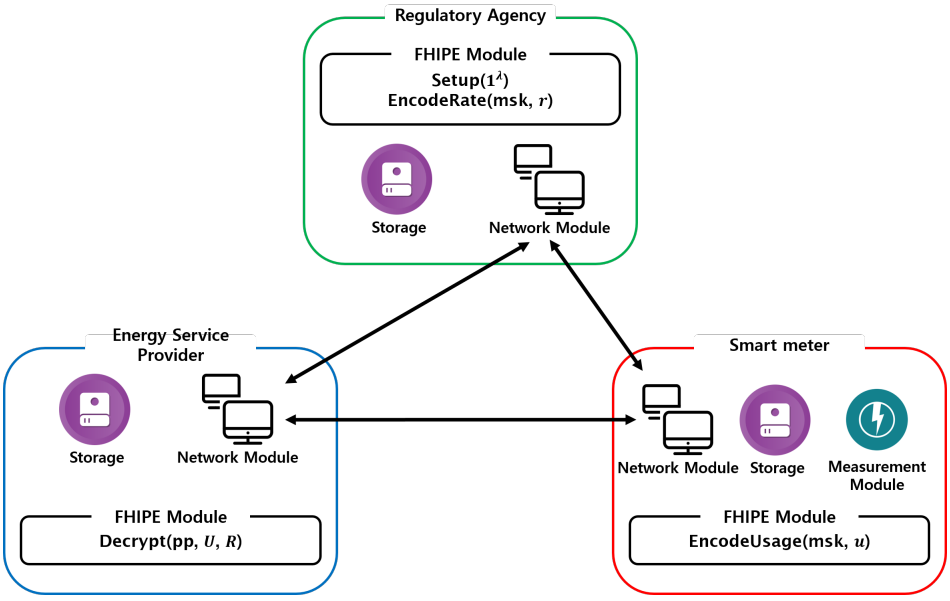

Figure 4. Component modules of proposed system.

To evaluate the proposed system with real-world data, we used measurement data collected by the Korea Electric Power Corporation (KEPCO) [43] from three sources, viz. an apartment building, a commercial building, and a factory. Each record in these datasets represents the daily electricity usage measured at 15-min intervals. The unit of measurement is kWh, and the valid digit of measured values is up to the second decimal place. Table 1 presents more details of the data, including the measured period, the number of records, minimum usage, maximum usage, and average usage.

Table 1. Electricity data details [43].

Dataset	Building Type	Period	#Records	Min. (kWh)	Max. (kWh)	Avg. (kWh)
A	Apartment	January 2018–December 2018	365	0.04	64.84	21.58
C	Commercial	January 2018–December 2018	365	17.47	77.62	40.36
F	Factory	January 2018–December 2018	365	3.90	1096.50	320.18

Along with the above data, we considered the rate plans proposed by KEPCO [44]. The rate plans were TOU rate plans, and the rates were determined by the month $\in$ {summer (June to August), spring/fall (March to May and September to October), winter (November to February)}, the hour in a day $\in$ {the off-peak period of spring/summer/fall/winter (23:00–09:00), the shoulder period of spring/summer/fall (09:00–10:00, 12:00–13:00, 17:00–23:00), the shoulder period of winter (09:00–10:00, 12:00–17:00, 20:00–22:00), the peak period of spring/summer/fall (10:00–12:00, 13:00–17:00), the peak period of winter (10:00–12:00, 17:00–20:00, 22:00–23:00)}, and the purpose of electricity usage $\in$ {general, industrial}. The range of the unit rate varied between 21.6 KRW/kWh and 244.1 KRW/kWh, and the valid digit was up to the first decimal place. We assumed that Datasets A and C used the general rate plan and that Dataset F used the industrial rate plan. We further assumed that the smart meter reported the electricity consumption data every two hours. That is, according to the definition in Section 3.2, the reporting period length was two hours. Because the measurement interval in all datasets was 15 min, the number of measured items per reporting period, i.e., the length of the consumption vector $\boldsymbol{u}$, was $n = 8$. The rationale for setting the reporting period length at two ours was as follows. According to the analysis in Reference [14], the minimum interval to retain privacy is one hour, as mentioned in the introduction above. We thus provided a reasonable security margin by setting the interval as twice the minimum.

The power consumption amount recorded in the datasets listed in Table 1 and the above rates are represented with decimal fractions. Meanwhile, the operations of the FHIPE scheme [30] are only

defined over integers. Therefore, to use the FHIPE operations properly without losing significant digits, $\boldsymbol{u}$ and $\boldsymbol{r}$ need to be quantized. For this purpose, $\boldsymbol{u}$ and $\boldsymbol{r}$ are converted to integer vectors before encoding. Because the valid digits in $\boldsymbol{u}$ are up to the second decimal place, the elements in $\boldsymbol{u}$ are quantized by $\boldsymbol{u} \times 100$. Similarly, $\boldsymbol{r}$ is quantized by performing $\boldsymbol{r} \times 10$, because the valid digits in the original $\boldsymbol{r}$ are up to the first decimal place. Then, the correct electricity charge is recovered by computing $c/1000$, i.e., $z/1000$.

Here, we present our experimental results. The experiments separately measured the execution times and data volume exchanged in the two stages: Registration and reporting. The experimental results for the execution time of the registration stage are presented in Table 2. Note that the performance of this stage depends on the security parameter λ and the vector size n, which also decides the dimension s of the matrices in the master secret key. We used the MNT 159 curve for bilinear group operations. It corresponds to $\lambda = 80$, which implies that the time complexity of the best-known attack algorithm to break the underlying FHIPE is roughly 2^{80}. We set $n = 8$, as explained above. The measured times are the averages over 10,000 executions. The column Setup_{RA} represents the time for the operation $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(1^{\lambda})$ by the RA. The columns Store_{RA}, Store_{SM}, and Store_{ESP} represent the times to perform $\mathsf{Store}(\mathsf{ID}, \mathsf{pp}, \mathsf{msk}, n, m)$ by the RA, $\mathsf{Store}(\mathsf{ID}, \mathsf{pp}, \mathsf{msk}, n, m)$ by the smart meter, and $\mathsf{Store}(\mathsf{ID}, \mathsf{pp}, n, m, \mathsf{P})$ by the ESP, respectively. The Network column represents the overall network overhead for the registration stage. As shown in the table, the registration stage completed quickly, i.e., in less than 50 ms in total.

Table 2. Breakdown of execution time of the registration stage (milliseconds).

Setup_{RA}	Store_{RA}	Store_{SM}	Store_{ESP}	Network	Total Time
11.54	0.53	9.51	0.02	24.39	45.99

Next, Table 3 presents the experimental results for the execution time of the reporting stage. The datasets listed in Table 1 and the above rate plans provide us with the electricity consumption vectors $\boldsymbol{u}$ and the rate vectors $\boldsymbol{r}$, respectively, for the corresponding reporting periods. Because the length of a reporting period is 2 hours, the number of reports per day is 12. Because every record in the datasets corresponds to a single day's usage, the total number of reports for a dataset is calculated by #Records $\times$ 12. The columns Encode_{SM}, Encode_{RA}, and Decrypt_{ESP} represent the time required for the operations $U \leftarrow \mathsf{EncodeUsage}(\mathsf{msk}, \boldsymbol{u})$, $R \leftarrow \mathsf{EncodeRate}(\mathsf{msk}, \boldsymbol{r})$, and $c \leftarrow \mathsf{Decrypt}(\mathsf{pp}, U, R)$ in Figure 3, respectively. The figures in Table 3 were obtained by applying these operations for each reporting period and computing the 10% trimmed means (after eliminating outliers) over each dataset. We did not separately present the time for generating $\boldsymbol{r}$ in the table because it was negligible. That is, its average execution time was less than 0.01 ms. However, it was counted in the total time. As shown in the table, the reporting stage can be completed in less than 1 s. Note that the EncodeUsage operation required more time than EncodeRate, even though the complexity of EncodeUsage, i.e., KeyGen, is lower than that of EncodeRate, i.e., Encrypt, according to Reference [30]. This is because EncodeUsage was performed on a relatively resource-constrained device. This proves that our design strategy of assigning the smart meter KeyGen instead of Encrypt was effective.

Table 3. Breakdown of execution time of the reporting stage (milliseconds).

Dataset	#Reports	Encode_{SM}	Encode_{RA}	Decrypt_{ESP}	Network	Total Time
A	4380	60.23	42.82	397.44	14.25	514.75
C	4380	60.33	42.81	400.39	14.26	517.80
F	4380	60.48	42.79	424.77	14.33	542.37
Average		60.35	42.81	407.53	14.28	524.97

Finally, we evaluated the data volume exchanged among the involved entities in the two stages. Table 4 presents the experimental results for the packet sizes in each stage. In the table, $Packet_{A \to B}$ represents the average size of the packet transmitted from A to B, including the network header and payload. According to the experimental results, all packets require only moderate bandwidth. The packet from the RA to the smart meter in the registration stage, which contains msk with two $(n \times n)$ matrices $\mathbf{B}$ and $\mathbf{B}^{\star}$, consumes the most traffic, but its size is smaller than 8KB. However, we have to examine the reporting stage more carefully because it is expected to occur more frequently than registration. In particular, we should also consider the situation where the ESP receives reports from multiple smart meters. As shown in Table 4, the data volume exchanged between one smart meter, the ESP, and the RA during a reporting stage is approximately 6 KB in total. If there are N smart meters, the amount of data exchanged in a reporting stage will be $6N$ KB. This capacity will be sufficient to cover a reasonable number of smart meters, but it should be verified though an implementation involving multiple smart meters. We leave this issue for future research.

Table 4. Data volume exchanged among the entities in the two stages (bytes).

Registration		Reporting			
$Packet_{RA \to SM}$	$Packet_{SM \to ESP}$	Dataset	$Packet_{SM \to ESP}$	$Packet_{ESP \to RA}$	$Packet_{RA \to ESP}$
7770	860	A	1736	875	3351
		C	1740	878	3351
		F	1735	871	3351

6. Discussion

In this paper, we proposed a privacy-preserving electricity billing method using FHIPE. To our knowledge, this is the first method that does not sacrifice data granularity for privacy. We implemented a prototype billing system composed of a smart meter, an ESP, and an RA. The experimental results with real measurement data show that the power consumption data of a smart meter can be reported to a service provider and accumulated for invoicing in less than 1 s and in a privacy-preserving manner. This proves the feasibility of the proposed system.

We remark that although our experiment was done with TOU, where the rates do not change frequently, the proposed method can be just as effectively applied to RTP, because the performance of our cryptographic operations—e.g., EncodeRate and Decrypt—does not depend on whether the values of r_i are identical or distinct.

We finally remark that for advanced services, the proposed method may be combined with other privacy-preserving protocols. For example, consider a situation where the ESP wants to use the collected data for a real-time load shedding purpose apart from billing. In this case, the ESP requires fine-grained readings, i.e., each u_i, to control power generation and distribution in real time, which is not possible with the proposed method. However, note that for this real-time control purpose, the electricity usage data from each individual smart meter are not necessary, but the aggregate data from multiple smart meters in a certain geographic region are sufficient. Therefore, we may adopt the previous research results aiming at spatially aggregating data from multiple smart meters in a cluster. For example, the method in Reference [23] aggregates spatial consumption of smart meters using a modified version of the Paillier homomorphic encryption [45]. In the spatial consumption aggregation protocol proposed in Reference [23], each smart meter j performs a modified Paillier encryption $\mathcal{E}_{pk}(u_i^j)$ using a public key pk, where u_i^j is the measurement of smart meter j $(0 \le j \le N-1)$ in the i-th interval $(0 \le i \le n-1)$. Then, any smart meter can play the role of an aggregator and aggregates the encrypted data from N smart meters into $\prod_{j=0}^{N-1} \mathcal{E}_{pk}(u_i^j)$. According to the property of the Paillier cryptosystem, $\mathcal{D}_{sk}\left(\prod_{j=0}^{N-1} \mathcal{E}_{pk}(u_i^j)\right) = \sum_{j=0}^{N-1} u_i^j$, where $\mathcal{D}_{sk}$ denotes decryption using the private key sk. This protocol can be combined with ours as follows: The key pair can be set up in the registration stage of the proposed method, and aggregation may be performed in the reporting stage. However, the aggregation should be performed n times in each reporting period. It is also possible to set independent reading

granularity for each purpose, e.g., 15 min for billing and 5 min for real-time control. However, to realize the combination of the proposed method and spatial aggregation protocol, many practical issues, such as optimal parameter-tuning to make the most of the limited resource of a smart meter, should be addressed through implementation and experiments. We leave these issues for future research.

Author Contributions: J.-H.I. designed the protocol and implemented the prototype system. H.-Y.K. collected the datasets and performed experiments. S.-Y.J. implemented the prototype system. M.-K.L. proposed the main idea for this research and organized the entire process.

Funding: This work was supported in part by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20161210200610), and in part by Korea Electric Power Corporation (Grant number: R18XA01).

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

Abbreviations

The following abbreviations are used in this manuscript:

TOU	Time-Of-Use
RTP	Real-Time Pricing
NIALM	Non-Intrusive Appliance Load Monitoring
FHIPE	Function-Hiding Inner Product Encryption
ESP	Energy Service Provider
RA	Regulatory Agency

References

1. Faruqui, A.; Hledik, R.; Palmer, J. Time-Varying and Dynamic Rate Design. 2012. Available online: https://www.raponline.org/knowledge-center/time-varying-and-dynamic-rate-design/ (accessed on 17 January 2019).
2. PG&E. Understand the PG&E Tiered Rate Plan. Available online: https://www.pge.com/en_US/residential/rate-plans/rate-plan-options/tiered-base-plan/tiered-base-plan.page (accessed on 17 January 2019).
3. Wang, Z.; Li, F. Critical peak pricing tariff design for mass consumers in Great Britain. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–29 July 2011; pp. 1–6. [CrossRef]
4. US Department of Energy's Office of Electricity Delivery and Energy Reliability. Advanced Metering Infrastructure and Customer Systems: Results from the Smart Grid Investment Grant Program. Available online: https://www.smartgrid.gov/document/SGIG_Results_for_AMI_and_Customer_Systems_2016.html (accessed on 17 January 2019).
5. McDaniel, P.D.; McLaughlin, S.E. Security and Privacy Challenges in the Smart Grid. *IEEE Secur. Priv.* **2009**, *7*, 75–77. [CrossRef]
6. Hart, G.W. Nonintrusive appliance load monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [CrossRef]
7. Baranski, M.; Voss, J. Genetic algorithm for pattern detection in NIALM systems. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), The Hague, The Netherlands, 10–13 October 2004; pp. 3462–3468. [CrossRef]
8. Zeifman, M.; Roth, K. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Trans. Consum. Electron.* **2011**, *57*, 76–84. [CrossRef]
9. Kim, H.; Marwah, M.; Arlitt, M.F.; Lyon, G.; Han, J. Unsupervised Disaggregation of Low Frequency Power Measurements. In Proceedings of the Eleventh SIAM International Conference on Data Mining, Mesa, AZ, USA, 28–30 April 2011; pp. 747–758. [CrossRef]
10. Kolter, J.Z.; Jaakkola, T.S. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, La Palma, Spain, 21–23 April 2012; pp. 1472–1482.
11. Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors* **2012**, *12*, 16838–16866. [CrossRef] [PubMed]

12. Parson, O.; Ghosh, S.; Weal, M.J.; Rogers, A. Non-Intrusive Load Monitoring Using Prior Models of General Appliance Types. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 356–362.
13. Vogiatzis, E.; Kalogridis, G.; Denic, S.Z. Real-time and low cost energy disaggregation of coarse meter data. In Proceedings of the 4th IEEE PES Innovative Smart Grid Technologies Europe, Lyngby, Denmark, 6–9 October 2013; pp. 1–5. [CrossRef]
14. Eibl, G.; Engel, D. Influence of data granularity on smart meter privacy. *IEEE Trans. Smart Grid* **2015**, *6*, 930–939. [CrossRef]
15. Lisovich, M.A.; Mulligan, D.K.; Wicker, S.B. Inferring Personal Information from Demand-Response Systems. *IEEE Secur. Priv.* **2010**, *8*, 11–20. [CrossRef]
16. Greveler, U.; Justus, B.; Löhr, D. Multimedia Content Identification Through Smart Meter Power Usage Profiles. In Proceedings of the Computers, Privacy and Data Protection (CPDP 2012), Brussels, Belgium, 25–27 January 2012.
17. Engel, D. Wavelet-based load profile representation for smart meter privacy. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference, Washington, DC, USA, 24–27 February 2013; pp. 1–6. [CrossRef]
18. Engel, D.; Eibl, G. Multi-resolution load curve representation with privacy-preserving aggregation. In Proceedings of the 4th IEEE PES Innovative Smart Grid Technologies Europe, Lyngby, Denmark, 6–9 October 2013; pp. 1–5. [CrossRef]
19. Erkin, Z.; Troncoso-pastoriza, J.R.; Lagendijk, R.L.; Perez-Gonzalez, F. Privacy-preserving data aggregation in smart metering systems: An overview. *IEEE Signal Process. Mag.* **2013**, *30*, 75–86. [CrossRef]
20. Efthymiou, C.; Kalogridis, G. Smart Grid Privacy via Anonymization of Smart Metering Data. In Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 238–243. [CrossRef]
21. Bohli, J.; Sorge, C.; Ugus, O. A Privacy Model for Smart Metering. In Proceedings of the 2010 IEEE International Conference on Communications Workshops, Capetown, South Africa, 23–27 May 2010; pp. 1–5. [CrossRef]
22. Kim, Y.; Ngai, E.C.H.; Srivastava, M.B. Cooperative state estimation for preserving privacy of user behaviors in smart grid. In Proceedings of the IEEE Second International Conference on Smart Grid Communications, Brussels, Belgium, 17–20 October 2011; pp. 178–183. [CrossRef]
23. Erkin, Z.; Tsudik, G. Private Computation of Spatial and Temporal Power Consumption with Smart Meters. In Proceedings of the Applied Cryptography and Network Security-10th International Conference, Singapore, 26–29 June 2012; pp. 561–577. [CrossRef]
24. Kursawe, K.; Danezis, G.; Kohlweiss, M. Privacy-Friendly Aggregation for the Smart-Grid. In Proceedings of the Privacy Enhancing Technologies-11th International Symposium, Waterloo, ON, Canada, 27–29 July 2011; pp. 175–191. [CrossRef]
25. Ács, G.; Castelluccia, C. I Have a DREAM! (DiffeRentially privatE smArt Metering). In Proceedings of the 13th International Conference on Information Hiding, Prague, Czech Republic, 18–20 May 2011; pp. 118–132. [CrossRef]
26. Li, F.; Luo, B.; Liu, P. Secure Information Aggregation for Smart Grids Using Homomorphic Encryption. In Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 327–332. [CrossRef]
27. Garcia, F.D.; Jacobs, B. Privacy-Friendly Energy-Metering via Homomorphic Encryption. In Proceedings of the 6th International Workshop on Security and Trust Management, Athens, Greece, 23–24 September 2010; pp. 226–238. [CrossRef]
28. Sahai, A.; Seyalioglu, H. Worry-free Encryption: Functional Encryption with Public Keys. In Proceedings of the 17th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 4–8 October 2010; pp. 463–472.
29. Garg, S.; Gentry, C.; Halevi, S.; Raykova, M.; Sahai, A.; Waters, B. Candidate Indistinguishability Obfuscation and Functional Encryption for All Circuits. In Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 40–49.

30. Kim, S.; Lewi, K.; Mandal, A.; Montgomery, H.; Roy, A.; Wu, D.J. Function-Hiding Inner Product Encryption Is Practical. In Proceedings of the Security and Cryptography for Networks (SCN 2018), Amalfi, Italy, 5–7 September 2018; pp. 544–562.
31. Bishop, A.; Jain, A.; Kowalczyk, L. Function-Hiding Inner Product Encryption. In Proceedings of the 21st International Conference on Advances in Cryptology (ASIACRYPT 2015), Auckland, New Zealand, 29 November–3 December 2015; pp. 470–491.
32. Abdalla, M.; Bourse, F.; De Caro, A.; Pointcheval, D. Simple Functional Encryption Schemes for Inner Products. In Proceedings of the Public-Key Cryptography—PKC 2015, Gaithersburg, MD, USA, 30 March–1 April 2015; pp. 733–751.
33. Datta, P.; Dutta, R.; Mukhopadhyay, S. Functional Encryption for Inner Product with Full Function Privacy. In Proceedings of the Public-Key Cryptography—PKC, Taipei, Taiwan, 6–9 March 2016; pp. 164–195.
34. Kim, S.; Kim, J.; Seo, J.H. A New Approach for Practical Function-Private Inner Product Encryption. Available online: https://eprint.iacr.org/2017/004 (accessed on 29 March 2019).
35. US Department of Energy's Office of Electricity Delivery and Energy Reliability. Time Based Rate Programs. Available online: https://www.smartgrid.gov/recovery_act/time_based_rate_programs.html (accessed on 17 January 2019).
36. Korea Electricity Regulatory Commission. Available online: http://www.korec.go.kr/intrcn/moveObjectiveAndStatus.do (accessed on 18 March 2019).
37. Federal Energy Regulatory Commission. Available online: https://www.ferc.gov/ (accessed on 18 March 2019).
38. Kim, S.; Lewi, K.; Mandal, A.; Montgomery, H.; Roy, A.; Wu, D.J. Function-Hiding Inner Product Encryption Is Practical. Available online: https://eprint.iacr.org/2016/440.pdf (accessed on 29 March 2019).
39. The Pairing-Based Cryptography Library (PBC). Available online: https://crypto.stanford.edu/pbc/ (accessed on 17 January 2019).
40. GNU Multiple Precision Arithmetic Library (GMP). Available online: https://gmplib.org/ (accessed on 17 January 2019).
41. A Library for Doing Number Theory (NTL). Available online: https://www.shoup.net/ntl/ (accessed on 17 January 2019).
42. Shanks, D. Class number, a theory of factorization, and genera. *Proc. Symp. Math. Soc.* **1**971, *20*, 415–440.
43. KEPCO Smart Power Management (iSMART). Available online: https://pccs.kepco.co.kr/iSmart/ (accessed on 15 February 2019).
44. KEPCO Electricity Tariffs. Available online: http://cyber.kepco.co.kr/ckepco/front/jsp/CY/E/E/CYEEHP00101.jsp (accessed on 15 February 2019).
45. Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; pp. 223–238.