Providing Anonymous Communication, Privacy-Preserving Data Aggregation and Dynamic Billing System in Smart Grid Using Permissioned Blockchain

Ozgur Oksuz

Faculty of Engineering, Adiyaman University,
Adiyaman, Turkey

Abstract. This paper proposes an efficient data aggregation and dynamic billing system that it uses anonymous communication for exchanging information between smart meters and CC. Moreover, the given scheme consists of a permissioned blockchain. This blockchain contains the ledger that keeps users' anonymized identities and electricity consumption for predefined time ranges. Using consumption data of users, a billing mechanism can bill the users accordingly. In the construction, since all the parties in the system have the ledger, every party has the aggregated usage of the electricity without using very heavy cryptographic operations such as homomorphic encryption, bilinear pairing, etc. Using the ledger in our model, the aggregation of the users' electricity consumption can be computed by anyone in the system. Moreover, users can verify their bills and check any data using a signature scheme. This results in that the integrity of all data is going to be preserved. The proposed approach mainly uses hash functions to provide the same functionality (aggregation of the users' data consumption, data integrity check, and dynamic pricing and billing) with preserving data privacy of the users.

1 Introduction

A smart grid (SG) uses a two-way digital communication system to supply electricity to consumers. This system provides consumers to monitor, control and analyze their usage and communicate with the entities to improve efficiency, reduce energy consumption and cost. Furthermore, another useful application of the smart grid is to integrate renewable energy resources (solar, wind, wave). This integration makes the energy grid more sustainable and environmentally friendly. Advanced metering infrastructure (AMI) is an important part of SG that it integrates costumers and the energy companies to get involved in utility management. An AMI consists of smart meters (SMs), customers, the control center (CC), and service provider (SP). An SM measures the customer's consumption data and sends it to SP which aggregates the users' data. The data then is used by SP for the real-time energy management systems such as dynamic pricing and billing. Moreover, based on this information, a service provider (SP) can balance and manage bulk generation and consumption for future use of the electricity. A CC provides system parameters and sends it to the parties.

Sending usage consumption data by users to SP introduces some security and privacy problems. Since users' consumption data is used for dynamic pricing and billing purposes, a third party (an attacker/hacker) can inject false consumption data that would unbalance the load management and dynamic pricing. So the user pays much more money than the user is supposed to pay. It

DOI: 10.5121/ijnsa.2020.12202

is possible that an attacker can send the manipulated meters' reading ([18]) and these false information results unbalancing the load management. This causes higher energy generation costs and even causes energy blackouts in a region. Furthermore, a customer can also send incorrect usage information to CC so it gets lowered its bill.

Another privacy issue in the smart grid is to leak individual user's consumption data to third parties. In this case, the attacker can learn users' daily life routine [4], [27]. For example, if a user's consumption is very low or there is no consumption in a certain time interval then the user is not at home. So this information is enough for thieves for getting into user's home to steal valuable things. So the user's data consumption should be sent to an aggregator (SP) that aggregates the users' usage of data in a privacy-preserving way.

The most important problem in smart grid is to keep users' anonymity. It is known that a smart meter sends periodically the users' consumption data to data aggregator (SP) that collects all users' consumption data. Therefore, any message that was sent by a smart meter is going to be known by the aggregator and any outsiders. So, any internal or external adversary sees messages coming from a specific smart meter.

The collection of electricity consumption data is very useful for a bunch of smart-grid applications. To compute dynamic electricity pricing, SP needs to have all the users' aggregated consumption data without learning each user's data consumption. Another application is that users might know their energy usage information in a given period to manage their energy consumption. So, SP needs to collect smart meter readings at arbitrary intervals or periods. This collection should be efficient. Therefore, SP needs to aggregate users' data consumption without performing heavy cryptographic operations such as homomorphic encryption. Although several existing techniques have been proposed for privacy-preserving data aggregation for billing of energy in smart grids, most of the existing schemes are based on computationally expensive cryptographic operations for encrypting user's data or generating a signature on its data, and aggregating all users' consumption data such as Paillier crypto system, lattice-based encryption, ElGamal encryption, functional encryption etc. On the other hand, in the existing masking-based schemes, for verifying the correctness of the masking secrets, they also use the computationally expensive cryptographic operations like Bilinear mapping, or homomorphic hashing. These cryptographic operations are not suitable for resource-limited smart meters.

Furthermore, once SP computes dynamic electricity prices, computes the usage costs based users' electricity consumption. When SP is untrusted, each user might want to learn the correctness of their bills and the electricity prices of the corresponding time interval. Then there should be another transparent mechanism that it allows users to check the integrity of their bills and electricity prices.

To address all the problems above, we introduce privacy-preserving dynamic billing and data aggregation scheme. Our scheme consists of permissioned blockchain that it provides transparent usage consumption of users. The communication anonymity is preserved by using the onion network that is presented in [26]. In this network, messages are encapsulated in multiple layers of encryption and sent through many nodes in the network. This provides communication anonymity since no single node, except the sender and receiver, knows the origin of the messages. Onion routing is also used in some other studies such as in [16]. The study in [16] examines the anonymous energy trading system between the consumers. The blockchain technology is a seminal work that is presented in [21]. A distributed ledger is the main entity in blockchain technology kept by each player in the system. Once a new transaction is issued it is put into the ledger. Then each user can see it. Using the blockchain in smart grid technology comes with some benefits. In our system, a ledger that keeps the user's identity with the user's data usage, a transaction timestamp and a signature that provides integrity. In the ledger, the user's identity is anonymized and the consumption data is in the cleartext. So each entity in the system (users, CC, SP) can see the other users' consumption data in the clear but any malicious entity can not figure out which consumption data belongs to which user. Therefore, the attacker (internal or external) can not map any user identity to any consumption data. Making the all users' consumption in the clear eliminates the heavy use of homomorphic encryption schemes for aggregation of users' data consumption for dynamic pricing

and billing. Another benefit of using ledger in our scheme is to compute billing. Since each user can see all the usage information in the ledger, each user can compute dynamic price of the electricity and its own bill.

Previous Version of This Paper [24]: This paper is the extended version of the paper presented at CRIS 2020 ([24]). In [24], in order to provide communication anonymity between smart meters and CC, the IP addresses of the smart meters should be changed by reconfigurating the smart meters each time when each smart meter sends a consumption message to CC. This can be done by disconnecting and reconnecting the smart meter to achieve this goal. But this option is not going to be practical. It may introduce network congestion and blackout time.

Another method to provide anonymous communication could be the anonymous identities (IP addresses) are embedded to smart meter in advance. When each smart meter sends a message to CC, the smart meter sends one by one of the pre-adjusted identities. This can be adjusted by CC since CC is fully trustable. This is also needs to be adjusted by CC in advance. So this requires a special implementation on smart meters.

In this paper, to have anonymous communication between the smart meters and CC, the onion routing protocol [26] is used. To do this new architecture is designed. Using onion routing protocol which has layered encryption of consumption data also solves the problem in [24] that is mentioned in the paper (the discussion section in [24]). Since the whole message consisting of some parts: a user's consumption data, a time stamp, smart meter's ID, a signature public key and a signature is encrypted so that any internal/external adversary is not able to change any parts at its will when the message is transmitted. Moreover, in this paper, there are given some improvements such as reducing communication complexity between users/consumers and CC.

Contribution: Our scheme has the following properties:

- We use blockchain technology for a transparent privacy preserving data aggregation and dynamic pricing scheme that each player keeps a ledger that has all the messages from the all other players,
- Our scheme provides user anonymity that each user's usage data is private to any internal and external adversaries,
- Our scheme provides data integrity that each message sent from each smart meter has also a signature for verification of the data. Data verification mechanism pretends an external/internal adversary to inject false data consumption,
- Once a user receives its bill, it can verify its bill using the ledger.

Outline. The rest of the paper is organized as follows. In Section 2, we introduce state-of-art privacy preserving data aggregation and dynamic pricing and billing schemes. In section 3, we introduce adversarial model and design goals of our system. Section 4 gives technical preliminaries that are used throughout the paper. Section 5 presents the construction for privacy preserving data aggregation and dynamic pricing and billing scheme. In section 7, we provide some discussions for our protocol. In section 8, we provide the security analysis of our construction. Last, section 9 concludes the paper and outlines future directions.

2 Related Work

There have been several studies about dynamic pricing and billing. The studies in [10], [1], [8], [2], [29], [5], [28], [14], [20], [30] and [19] focus on examining privacy preserving data aggregation problem but do not examine billing mechanism so these studies do not support billing application. They focused on data aggregation. The work in [10] proposed a peer-based privacy for smart grid. The construction in [10] does not hide user identity. The work in [8] use Pallier cryptosystem to encrypt data and use several data aggregators/servers. To extract the aggregated data consumption each server decrypts the aggregated ciphertext. [29] uses identity based encryption scheme to aggregate users consumption. Their scheme has a high computation cost of batch processing. In [5], the authors use Elliptic Curve Diffie Hellman key exchange protocol to aggregate users'

data consumption data. However, their scheme is not secure when there is an internal attacker. The study, [1], uses lattice based cryptography. In [28], the authors achieve privacy using El-Gamal encryption scheme. These studies [1, 28] are computationally expensive.
Some studies [31], [10], [6], [23], [13], [17], [22], have proposed privacy preserving data aggregation and billing. The work in [31] introduced a data aggregation and billing protocol that does not hide user anonymity. In the billing period in [31], each customer sends their total daily load to the utility company for billing purposes. This can be a big problem since the utility company is not trusted and knows each user's consumption data. In [6], the authors use cryptographic commitments for integrity of data and use homomorphic encryption scheme for data privacy. In [23] also uses a homomorphic commitment and an encryption scheme and a digital signature scheme to have privacy. These studies suffer with high computational and communication complexities. The studies, in [6], [23], [9], [17], [22], focus on mainly focuses static billing with standard tariff plans. But some countries such as UK, Norway, etc. use dynamic price and billing systems. In [9], the authors proposed two protocols mainly focus on the privacy-preserving billing but not the data aggregation. Gope et al. [13] proposed an efficient privacy preserving data aggregation and dynamic pricing and billing protocol. Their scheme uses heavily masking technique to randomize the individual measurements of the users. Their scheme has some problems such as it is not clear how the parties reissue their randomness. In their adversarial model, the aggregator is semi-honest adversary which it is not malicious. The work in [7] proposed a privacy preserving data aggregation and dynamic pricing and billing scheme that uses elliptic curve cryptography scheme for secure data aggregation. In their work the trusted party computes the dynamic prices and generates bills to the users and does not provide anonymity. In our scheme, the service provider (a private company) computes dynamic pricing and issues bill to the users. The study in [15] proposed privacy-preserving electricity billing system using functional encryption using bilinear map structure. In their scheme they sacrifice data quality for privacy. Their scheme has some problems since their scheme is based on bilinear map and it is computationally expensive. The work in [3] made a comperative analysis for privacy-preserving data aggregation schemes. The comparisons are based on computation cost, communication cost, privacy, resistance against malicious aggregator. The studies above do not provide anonymity (except [13]). We use blockchain technology for a transparent data aggregation and billing to have user anonymity.

3 System and Adversary Model and Design Goals

Our system model is shown in Fig.1. In our system model, there are five major entities: 1) control center (CC) that sets the system parameters for signature scheme and hash functions, 2) ledger that keeps user identities in the encrypted form, a consumption data of each user, a time-stamp of each transaction (message) and a signature of each transaction (message), 3) smart meters (SM), 4) consumers/users and 5) service provider (SP) that needs aggregated consumption data for generating electricity and also implements dynamic price of each time interval for billing process and issues bills to the users. The service provider also responsible for collecting, processing, and analysing the data. This is because it can adjust the electricity generation, transmission, and distribution to meet the dynamic demands.
The work flow of our system is illustrated in Fig.1:

- CC first sets the system parameters such as it chooses hash functions, sets group and its generator and sends to the all parties.
- Each smart meter chooses public/secret key pair (or CC provides these keys to each smart meter) then each smart sends its usage data to CC via using onion network.
- CC first checks the data if it comes from an authorized smart meter, then CC puts it into a ledger and distributes to the users and SP.
- For each period in a ledger, SP gets aggregated users' consumption data to compute electricity price. SP only computes the sum of all the consumption data in the corresponding ledger. SP

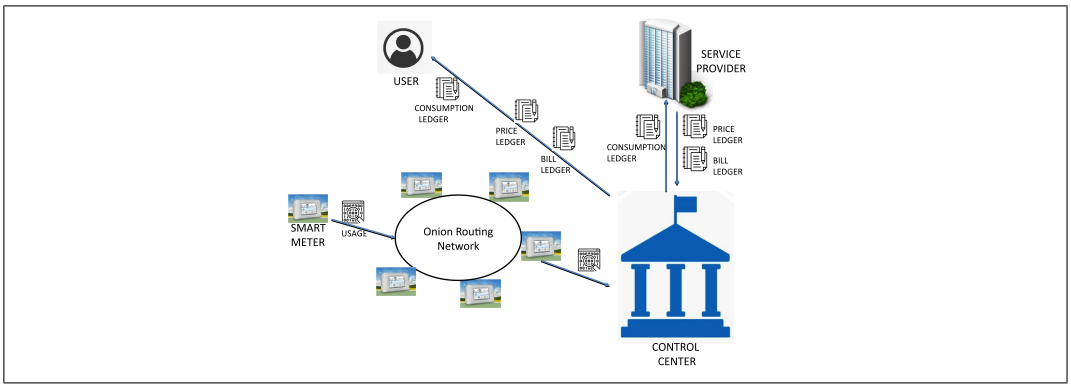

Fig. 1: System architecture and workflow.

sends electricity prices to CC then CC forwards the all users in the ledger. After determination of the price of electricity for the each time range, SP also issues a bill to each user based on the order of smart meters' identities in the ledger. SP also sends this information to CC then CC forwards to the all users.

- Each user can verify its bill by computing its usage in the ledger.

Another good thing is that each user and CC are also able to compute dynamic price of the electricity and so their bills by themselves since each ledger (for each time) is kept by the everyone in the system.

3.1 Adversary Model

In our model, CC is fully trusted (it is owned by a government). SP is malicious that is run by a private company. It can deviate from the protocol execution by sending incorrect values. SP also gets the aggregated consumption data from all the users but also it tries to find out individual users' consumption data. Moreover, SP tries to learn each user's bill. In our system each consumer can also be malicious to reduce its bill. We consider the following attacks in our system:

- Any internal/external attacker tries to learn individual user's consumption and its bill,
- Any external attacker sets a fake smart meter and sends fake data consumption to CC in order to unbalance the load,
- Any internal attacker tries to corrupt a smart meter and sends incorrect consumption data to unbalanced the load,
- Any internal/external adversary tries to corrupt SP to give incorrect prices,
- A consumer/user tries to manipulate its consumption data to reduce its bill.

4 Technical Preliminaries

Definition 1 (Pseudo Random Generator). *A pseudorandom generator (PRG) outputs strings that are computationally indistinguishable from random strings. More precisely, we say that a function* $G : \{0,1\}^n \rightarrow \{0,1\}^m$*, where* $m > n$ *is a* (t, ϵ, q)*-pseudo-random generator if*

1. *G is efficiently computable by a deterministic algorithm,*
2. *for all t time probabilistic algorithm* $\mathcal{A}$ *that makes at most q adaptive queries,*

$$|\Pr[\mathcal{A}(G(s)) = 0 | s \leftarrow \{0,1\}^n] - \Pr[\mathcal{A}(r) = 0 | r \leftarrow \{0,1\}^m]| \leq \epsilon$$

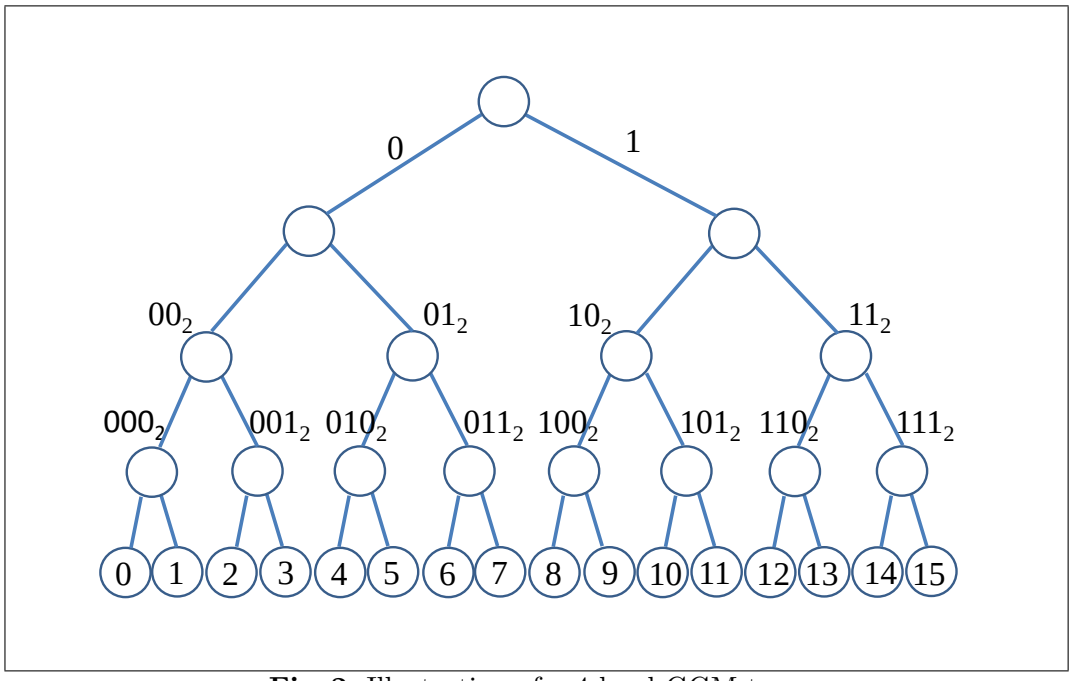

Fig. 2: Illustration of a 4-level GGM tree.

Definition 2 (Pseudo Random Function). *A pseudo-random function is computationally indistinguishable from a random function — given pairs* $(x_1, f_k(x_1)), \ldots, (x_m, f_k(x_n))$*, an adversary cannot predict* $f_k(x_{n+1})$ *for any* x_{n+1}*. More precisely, we say that a function* $f : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^m$ *is a* (t, ϵ, q)*-pseudo-random function if*

1. $f(k, x) = f_k(x)$ *can be computed efficiently from input* $x \in \{0,1\}^n$ *and key* $k \in \{0,1\}^s$*.*
2. *for any* t *time oracle algorithm* $\mathcal{A}$ *that makes at most* q *adaptive queries,*

$$|\Pr[\mathcal{A}^{f_k(.)} = 0 | k \leftarrow \{0,1\}^s] - \Pr[\mathcal{A}^g = 0 | g \leftarrow F : \{0,1\}^s \to \{0,1\}^m]| \leq \epsilon$$

Definition 3. *GGM-Based PRF [11] is built upon the well-known tree-based GGM PRF family [11], proposed by Goldreich, Goldwasser, and Micali. This family defines a PRF that takes a key* k *and a preimage* x*, and assigns it an image* $f_k(x)$*, such that* $f_k(x)$ *(for randomly chosen* k*) is indistinguishable from a uniformly random string of the same length. This PRF is based on the hierarchical application of any length-doubling Pseudorandom Generator (PRG) according to the structure induced by a tree, where input values are uniquely mapped to root-to-leaf paths. Specifically, let* G *be a publicly known PRG that takes a* m*-bit secret string* $k \in \{0,1\}^m$ *as input, and outputs a* $2m$*-bit string,* $G(k)$*. Let* $G_0(k)$ *and* $G_1(k)$ *denote respectively the first and second half of* $G(k)$*. The GGM pseudorandom function family [11] is defined as* $\mathcal{F} = \{f_k : \{0,1\}^m \to \{0,1\}^m\}_{k \in \{0,1\}^m}$ *such that* $f_k(x) = G_{x_0}\left(G_{x_1}\left(\cdots\left(G_{x_{m-1}}(k)\right)\right)\right)$*, where* $(x_{m-1} \ldots x_0)_2$ *is the binary representation of* x*.*

As an example, Fig. 2 depicts a GGM tree with 4 levels. The leaves are labeled with a decimal number from 0 to 15, sorted in ascending order. Every edge is labeled with 0 (resp. 1) if it connects a left (resp. right), child. Every internal node is labeled with the binary string determined by the labels of the edges along the path from the root to this node.

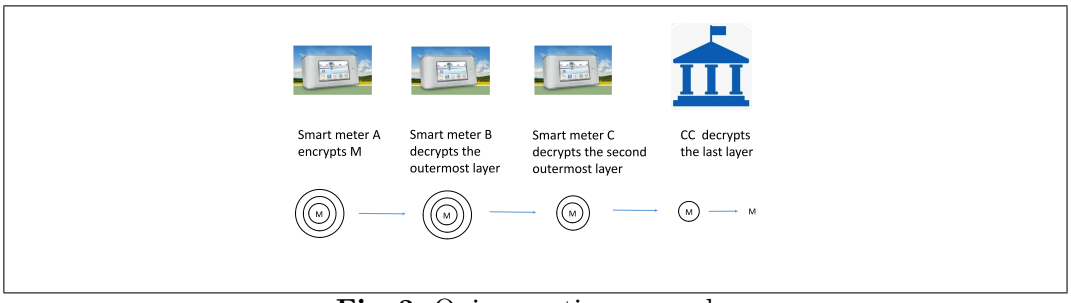

Fig. 3: Onion routing example.

Definition 4 (Computational Diffie-Hellman problem). *Consider G is a cyclic multiplicative group of order q and its generator is g. A probabilistic polynomial time adversary has a negligible probability of computing g^{ab} from given g, g^a, g^b, where $a, b \in Z_q$ are random elements.*

Definition 5 ([12]). *A digital signature scheme, Sig, consists of three algorithms, $Sig = (sigKeyGen; sigSign; sigVerify)$, where $sigKeyGen$ generates public and private keys $sigPk; sigSk$, $sigSign$ generates a signature for a message, and $sigVerify$ determines if a signature is generated under the corresponding message. We say that a digital signature scheme is secure if the signature scheme is existentially unforgeable under adaptive chosen message attack (UF-CMA). UF-CMA means that an adversary who is given signatures for some messages of its choice adaptively should not be able to produce a signature for a new message.*
In our construction we use Schnorr Signature scheme [25]:
We work in group $\mathcal{G}$ of prime order q and g is the generator of this group. A hash function $\mathcal{H}$ is used. $\mathcal{H} : \{0,1\}^* \rightarrow Z_q$.
$sigKeyGen$: Select $z \in Z_q$ as the secret key, $sigSk$ and $Z = g^z$ as the public key, $sigPk$.
$sigSign$: Select $r \in Z_q$, set $R = g^r$ and $c = \mathcal{H}(msg||R)$, where $||$ denotes concatenation and R is represented as a bit string. The signature on $msg \in \{0,1\}^*$ is $\rho = (y, R)$ where $y = r + zc$.
$sigVerify$: Let $c = \mathcal{H}(msg||R)$ and $\rho = (y, R)$, ρ is valid if $g^y = RZ^c$.

Definition 6 (Anonymous Communication ([26])).
In our scheme, we use an onion routing network in order not to reveal each smart meter's identity (IP address) to the public. Only the sender and receiver know the source of the messages. As an example of an onion network is given in Figure 3. In this example, smart meter A sends a message M which is the consumption data of a user to CC, through a network of onion routers. To do this, smart meter A encrypts M multiple times with a bunch of smart meters' public keys that the destination is going to be CC. Each node (smart meter) decrypts the layered ciphertexts with its corresponding secret key and forwards the message to another node (smart meter) so the final ciphertext is going to be decrypted by CC and then CC recovers the message.

5 Construction

Before diving into our construction first we want to explain the design of it. First of all, we use a signature scheme (Schnorr) to provide data integrity and use an onion routing network for anonymous communication (hiding IP addresses of the smart meters) between smart meters and CC. In the network, each smart meter acts as the onion routers. Each smart meter sends its consumption data with a signature as a message to some other smart meters in the network using smart meters public keys. The last node is going to be CC. CC extracts the message and checks if it is an authorized message and comes from an authorized smart meter, CC creates a ledger

and sends to every user and SP. Therefore, any other user can check if the data is intact and not changed by any internal adversary. Secondly, to eliminate internal and external attacks where the attackers can change the value before it is transmitted or any external attacker pretends it is a real smart meter but in the reality, it is not in the system, our scheme includes a fully trusted party which is CC. An external attack can happen when the adversary sets a fake smart meter and sends fake data consumption data. The attack happens as follows: an external attacker can send a consumption value by creating an anonymized identity and signature. If this attack is not eliminated or checked by a trusted party, it results that the load is going to be unbalanced. Also, it results in incorrect billing management. To eliminate these attacks CC checks every message whether the sender of the message is an authorized smart meter. Thirdly, we also introduce a long term key management system between users and CC for keeping the users' identity hidden to any other parties (except CC) and let CC eliminate internal and external attacks. This mechanism comes to play in the signature algorithm. The long term key management means that a key is shared by the users and CC in advance in the beginning of the protocol, then this key is derived by those parties non-interactively time to time when a new message is issued. This can be called as generating multiple secret keys from a single secret key. So for each message, each party does not need to share another key interactively. This approach reduces computation and communication costs. Every message is coded with a different identity and a different signature value by a smart meter (these both hide user's identity) thanks to pseudorandom functions (PRF) and hardness of Computational Diffie Hellman (CDH) assumption that is given in Def. 4. This function is generated from a pseudorandom generator which is going to be explained in section 5.1.

To protect the data privacy of a consumer from any internal/external adversaries we split the consumption of each user based on the average consumption data. The average consumption data of a region can be obtained from a statistical analysis of previous data of this region. According to this statistical analysis, we split the consumption data into 5 categories where the 3rd category is the average consumption category. Using this method, each smart meter first starts sending the data value (user's consumption data) to permissioned ledger via CC when the data consumption value of the user is met with average consumption value which it is the third category consumption data. Then, the smart meter sends its data value (user's consumption value) to permissioned ledger when the data consumption value of the user is met with the second category consumption value. This process continues whenever customer usage is presented. With this approach, the minimum consumption value of a user is less or equal to first category consumption data in the ledger. Each message consists of some public values, a user's consumption data, a timestamp, smart meter's ID, a signature public key and a signature. The user's ID is kept in the permissioned ledger anonymously. Users' consumption data values are in the cleartexts but any insider or external party is not able to map any consumption values to any individual user since each smart meter ID is changing for each message in the ledger. Moreover, there are bunch of the same consumption values that are seen in the ledger and each smart meter sends average consumption data to the ledger via CC. We can also say that this model is a kind of $k-$anonymized database model. That means there are at least $k-1$ values that are the same (identical). Any third party except CC can not retrieve the electricity consumption of a user for a specific time range (ledger). CC and each party (smart meters, SP) have shared secret keys for generating signatures and anonymizing their IDs for each message. Thus, CC is able to retrieve the usage of individual user's data consumption by having the permissioned ledger. As an example, in Tab.1, CC is able to figure out the consumption 1 and 3 belong to the same user which it is user 1, while the consumption 2 and 5 belong to user 2. In the table, each row concatenation of a hash value of a secret key and identity of a smart meter, a timestamp, a consumption data, a signature public key (belongs to the sender) and a signatures of these values for an integrity check by any party to show that the consumption data has not been changed by any untrusted party (internal or external parties).

All messages are kept in the ledgers and we introduce three types of the ledger. The first one is outsourcing consumption data ledger, L_{OCD} as short. The second one is data aggregation and

pricing ledger, L_{DAP} as short and the last ledger is bill calculation and pricing ledger, L_{BC} as short.

5.1 Protocol

Our protocol consists of four steps: Initialization, Outsourcing Consumption Data, Aggregation and Dynamic Pricing and Bill Calculation steps.

Initialization In this step, CC first sets system public parameters. g is a generator of a cyclic group $\mathcal{G}$ prime order q. q is a large prime (m bit). CC chooses four collision resistant hash functions: H, H_1, H_2 and H_3. $H_1 : \mathcal{G} \rightarrow \{0,1\}^m$. $H : \{0,1\}^{2m} \rightarrow \{0,1\}^m$, and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^m$ and $H_3 : \{0,1\}^m \rightarrow Z_q$. These hash functions are one-way cryptographic hash functions that can be implemented as SHA-1, SHA-2 or SHA-3. H is a keyed hash function that can be implemented as a HMAC. PRG is a pseudorandom generator, $PRG : \{0,1\}^m \rightarrow \{0,1\}^{2m}$ and f is a pseudorandom function, $f : \{0,1\}^m \times \{0,1\}^m \rightarrow \{0,1\}^m$. A pseudorandom function (PRF) can be generated from a pseudorandom generator. An implementation of this function can be seen in [32]. Each smart meter i chooses a public key and secret key pk_i, sk_i, where $pk_i = g^{sk_i}$, $sk_i \in Z_q$ (or these keys can be chosen by CC and they are integrated into the smart meter by CC [7]). Smart meter i registers its pk_i to CC as its identity. CC also chooses a pair of public key and secret key (pk_{CC}, sk_{CC}), where $pk_{CC} = g^{sk_{CC}}$, $sk_{CC} \in Z_q$. Then, CC publishes its pk_{CC}. A service provider also chooses its public key and secret key pk_{SP}, sk_{SP}, then the service provider registers its public key pk_{SP} to CC. Each smart meter i chooses a symmetric key for the signature protocol (Schnorr signature) $K_i \in \{0,1\}^m$, encrypts it with CC's public key pk_{CC} and sends to CC. SP chooses a public/secret key pair for the signature protocol (Schnorr signature) $sigSk_{SP} = K_{SP} \in Z_q$, and sends $sigPk_{SP} = g^{K_{SP}}$ to everyone. Also CC chooses a public/secret key pair for the signature protocol (Schnorr signature) $sigSk_{CC} = K_{CC} \in Z_q$, and sends $sigPk_{CC} = g^{K_{CC}}$ to everyone. To eliminate another external attack, $sigPk_{SP}$ should also be registered as SP's signature public key and $sigPk_{CC}$ also be known by every party in the system. In our system the identities of CC and SP can be known by the users. Our focus is to hide users' identity, their individual consumption data and their individual bills from SP, and any other internal and external attackers.

Remark 1. Public key of a signature is also sent by the issuer in the outsourcing computation data step (as a part of a message). Everyone can be able to see this signature public key in the transaction/message and everyone can easily extract this value to use in the signature verification algorithm.

Outsourcing Consumption Data In this step, each smart meter SM_i,

- computes $sk_{i,CC} = (pk_{CC})^{sk_i}$, $ID_i = H_1(pk_i)$, and does the following steps:

1. sends $M = 00||OCD_j||H(H_1(sk_{i,CC})||ID_i)||T_{i,j,1}||c_{i,j,1}||$ $||sigPk_{i,1}||sigSign_{sigSk_{i,1}}(H_2(00||OCD_j||H(H_1(sk_{i,CC})||ID_i)||T_{i,j,1}||c_{i,j,1}))$ to CC via onion routing network. SM_i encrypts the message M many times with public keys of smart meters (These smart meters are chosen by SM_i) and then each node decrypts the layered encryption of M and forwards to next node that the last node is going to be CC. In the message, $sigPk_{i,1}$ is the signature verification key of SM_i, $T_{i,j,1} \in \{0,1\}^{m/2}$ is the time stamp that SM_i sends the first consumption in time interval j, $c_{i,j,1} \in \{0,1\}^{m/2}$ that is generated based on average consumption data category. The definition of the triple index $(i, j, 1)$ is as follows: The first index defines smart meter's identity (user i), the second index is the current time interval which is j (ledger j), and the last index defines the index of the user's consumption data sent from SM_i (1 is first consumption data that SM_i has sent). $OCD_j \in \{0,1\}^{m/2}$ is the

Table 1: Permissioned Ledger L_{OCD_j}: Outsourcing Consumption Data for time interval j

1. $00 \| OCD_j \| H(H_1(sk_{1,CC}) \| ID_1) \| T_{1,j,1} \| c_{1,j,1} \| sigPk_{1,1} \| sigSign_{sigSk_{1,1}}(H_2(00 \| OCD_j \| H(H_1(sk_{1,CC}) \| ID_1) \| T_{1,j,1} \| c_{1,j,1}))$
2. $00 \| OCD_j \| H(H_1(sk_{2,CC}) \| ID_2) \| T_{2,j,1} \| c_{2,j,1} \| sigPk_{2,1} \| sigSign_{sigSk_{2,1}}(H_2(00 \| OCD_j \| H(H_1(sk_{2,CC}) \| ID_2) \| T_{2,j,1} \| c_{2,j,1}))$
3. $00 \| OCD_j \| H(H_1(sk_{1,CC}) \| H(H_1(sk_{1,CC}) \| ID_1)) \| T_{1,j,2} \| c_{1,j,2} \| sigPk_{1,2} \| sigSign_{sigSk_{1,2}}(H_2(00 \| OCD_j \| H(H_1(sk_{1,CC}) \| H(H_1(sk_{1,CC}) \| ID_1)) \| T_{1,j,2} \| c_{1,j,2}))$
4. $00 \| OCD_j \| H(H_1(sk_{3,CC}) \| ID_3) \| T_{3,j,1} \| c_{3,j,1} \| sigPk_{3,1} \| sigSign_{sigSk_{3,1}}(H_2(00 \| OCD_j \| H(H_1(sk_{3,CC}) \| ID_3) \| T_{3,j,1} \| c_{3,j,1}))$
5. $00 \| OCD_j \| H(H_1(sk_{2,CC}) \| H(H_1(sk_{2,CC}) \| ID_2)) \| T_{2,j,2} \| c_{2,j,2} \| sigPk_{2,2} \| sigSign_{sigSk_{2,2}}(H_2(00 \| OCD_j \| H(H_1(sk_{2,CC}) \| H(H_1(sk_{2,CC}) \| ID_2)) \| T_{2,j,2} \| c_{2,j,2}))$
6. $00 \| OCD_j \| H(H_1(sk_{4,CC}) \| ID_4) \| T_{4,j,1} \| c_{4,j,1} \| sigPk_{4,1} \| sigSign_{sigSk_{4,1}}(H_2(00 \| OCD_j \| H(H_1(sk_{4,CC}) \| ID_4) \| T_{4,j,1} \| c_{4,j,1}))$

$\boxed{1.\lfloor 01 \rfloor \lvert\rvert DAP_j \lvert\rvert H(H_1(sk_{SP,CC}) \lvert\rvert ID_{SP}) \lvert\rvert T_{SP,j} \lvert\rvert p_j \lvert\rvert sigPk_{SP} \lvert\rvert sigSign_{sigSk_{SP}}(H_2(01 \lvert\rvert DAP_j \lvert\rvert H(H_1(sk_{SP,CC}) \lvert\rvert ID_{SP}) \lvert\rvert T_{SP,j} \lvert\rvert p_j)).}$

Table 2: Permissioned Ledger L_{DAP_j}: Data Aggregation and Pricing Ledger for time interval j

Table 3: Permissioned Ledger L_{BC_j}: Bill Calculation and Pricing Ledger for time interval j

1. $10 \| BC_j \| H(H_1(sk_{1,CC}) \| ID_1) \| T_{SP,j,1} \| B_{1,j,1} \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(10) \| BC_j \| H(H_1(sk_{1,CC}) \| ID_1) \| T_{SP,j,1} \| B_{1,j,1}))$.
2. $10 \| BC_j \| H(H_1(sk_{2,CC}) \| ID_2) \| T_{SP,j,2} \| B_{2,j,1} \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(10) \| BC_j \| H(H_1(sk_{2,CC}) \| ID_2) \| T_{SP,j,2} \| B_{2,j,1}))$.
3. $10 \| BC_j \| H(H_1(sk_{1,CC}) \| H(H_1(sk_{1,CC}) \| ID_1)) \| T_{SP,j,3} \| B_{1,j,2} \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(10) \| BC_j \| H(H_1(sk_{1,CC}) \| H(H_1(sk_{1,CC}) \| ID_1))) \| T_{SP,j,3} \| B_{1,j,2}))$.
4. $10 \| BC_j \| H(H_1(sk_{3,CC}) \| ID_3) \| T_{SP,j,4} \| B_{3,j,1} \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(10) \| BC_j \| H(H_1(sk_{3,CC}) \| ID_3) \| T_{SP,j,4} \| B_{3,j,1}))$.
5. $10 \| BC_j \| H(H_1(sk_{2,CC}) \| H(H_1(sk_{2,CC}) \| ID_2)) \| T_{SP,j,5} \| B_{2,j,2} \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(10) \| BC_j \| H(H_1(sk_{2,CC}) \| H(H_1(sk_{2,CC}) \| ID_2))) \| T_{SP,j,5} \| B_{2,j,2}))$.
6. $10 \| BC_j \| H(H_1(sk_{4,CC}) \| ID_4) \| T_{SP,j,6} \| B_{4,j,1} \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(10) \| BC_j \| H(H_1(sk_{4,CC}) \| ID_4) \| T_{SP,j,6} \| B_{4,j,1}))$.

ledger's identity, two bits 00 is used for the ledger's type which is outsourcing consumption data ledger. For calculation of a bill, smart meter SM_i sends $H(H_1(sk_{i,CC})||ID_i)$ (or $sigPk_i$) to its user in a private way (using encryption). A consumer can share a secret key with the smart meter in order to send/receive messages from/to the smart meter. The shared secret key can be embedded into the smart meter using USB ([7]). This communication is needed when the user/consumer receives L_{OCD_j} ledger and calculates its bill. Since the user does not know the keys of the smart meter, the user needs the identity of the smart meter for each transaction to extract its bill from L_{OCD_j} ledger.
SM_i shares its secret key K_i with CC, CC and SM_i use GGM tree to derive all other keys non-interactively as follows: For the first message, SM_i computes $sigSign_{sigSk_{i,1}} = H_3(f_{K_i}(000001))$ and $sigPk_{i,1} = g^{H_3(f_{K_i}(000001))}$. PRF f takes a 6-digit input. This six input defines the GGM tree's height. If SM_i wants to generate $sigPk_{i,7}$, it needs to compute $g^{H_3(f_{K_i}(000111))}$. The binary value of $(000111)_2 = 7$. As a note that SM_i uses GGM tree-based PRF [11] to generate its secret key (for signature) based on the index of the message. Furthermore, $sigSk_{i,1}$ and $sigPk_{i,1}$ are the shared keys that SM_i and CC have. Based on the message index, CC and smart meters are able to compute these values. We call this key mechanism as a long term key management mechanism since these keys can be created based on the index of the message and changes every time. $sigSk_i$ and $sigPk_i$ can be computed by CC and SM_i since SM_i sent K_i to CC (or CC chooses it and integrates it into SM_i) in the initialization step.

2. CC checks if the sender is a legitimate smart meter by checking the message's most significant bits which are located between $m/2 + 3$ and $3m/2 + 2$ bits. For the given value $H(H_1(sk_{i,CC})||ID_i) = A$, CC computes $sk_{CC,i} = (pk_i)^{sk_{CC}}$, $ID_i = H_1(pk_i)$ and checks if $H(H_1(sk_{CC,i})||H_1(pk_i)) == A$. CC also checks if the signature is produced by a legitimate smart meter. If SM_i sends $sigPk_{i,1}$ which it is SM_i's first message to the ledger in the message, CC computes $g^{H_3(f_{K_i}(000001))}$ and compares it with $sigPk_{i,1}$. If equality holds, CC also uses signature verification algorithm with the message
$msg = H_2(00||OCD_j||H(H_1(sk_{i,CC})||ID_i)||T_{i,j,1}||c_{i,j,1})$. If the verification passes, CC puts the message into outsourcing consumption data ledger L_{OCD_j} and distributes it. Otherwise, it does not keep it. CC then sends the outsourcing consumption data ledger L_{OCD_j}, a time-stamp and a signature (Schnorr signature) as a message to all the parties. The signature is for the integrity of the outsourcing consumption data ledger L_{OCD_j}. The message is sent by CC to everyone (users,SP):
$H_1(pk_{CC})||L_{OCD_j}||T_{CC,j,OCD_j}||sigPk_{CC}||sigSign_{sigSk_{CC}}(H_2(H_1(pk_{CC})||L_{OCD_j}||T_{CC,j,OCD_j}))$, where $H_1(pk_{CC})$ is the identity of CC, L_{OCD_j} is the ledger in Table 1 that keeps all the transactions, T_{CC,j,OCD_j} is the time stamp, $sigPk_{CC}$ is the public key of the signature, and $sigSign_{sigSk_{CC}}(H_2(H_1(pk_{CC})||L_{OCD_j}||T_{CC,j,OCD_j}))$ is the signature.

Remark 2. This kind of blockchain is kind of permissioned blockchain (or private blockchain) that it is needed for preventing an attacker (outside or inside) to inject false consumption data.

Remark 3. Since smart meters and SP's keys are known by CC, CC can compute all the related shared keys and public keys for the signature verification in advance.

Remark 4. Another way to anonymize a smart meter's identity (it is located between $m/2 + 3$ and $3m/2 + 2$ most significant bits in each message) is to use randomized encryption. A CPA secure AES encryption can be used. When a smart meter sends messages to CC using randomized encryption, smart meter's ID is going to be different for each time. So any attacker can not figure out the source of the message.

Data Aggregation and Dynamic Pricing In this step, the service provider first checks the ledger L_{OCD_j}, its time-stamp and its signature which are sent by CC. If the verification passes,

SP retrieves the each individual consumption $c_{i,j,k}$ value in ledger L_{OCD_j} and adds them together. Then it does some analysis to compute price for each time slot. Since each ledger keeps consumption values for a time range, each ledger has its own electricity price. The service provider sends $01||DAP_j||H(H_1(sk_{SP,CC})||ID_{SP})||T_{SP,j}||p_j||$
$||sigPk_{SP}||sigSign_{sigSk_{SP}}(H_2(01||DAP_j||H(H_1(sk_{SP,CC})||ID_{SP})||T_{SP,j}||p_j)$, where two bits 01 is used for the type of the ledger which is data aggregation and pricing ledger, $DAP_j \in \{0,1\}^{m/2}$ is the identity of the ledger, $ID_{SP} = H_1(pk_{SP})$, p_j is the price for time interval j. $T_{SP,j}$ is the time stamp that the service provider sent the message for time interval j. To preserve the integrity of the data, the service provider also sends $SigSign_{sigSk_{SP}}(H_2(01||DAP_j||H(H_1(sk_{SP,CC})||ID_{SP})||T_{SP,j}||p_j))$ to CC.
Once CC receives a message from SP, CC first checks if the values are correct that the message really comes from SP. If so CC puts it into L_{DAP_j} ledger. The checking process is the same as that in the Outsourcing Consumption Data step.
CC then sends the data aggregation and pricing ledger L_{DAP_j} (Tab.2), a time-stamp and a signature (Schnorr signature) to all the users (or puts into a public bulletin). The signature is for the integrity of the data aggregation. CC sends pricing ledger L_{DAP_j} to everyone as follows:
$H_1(pk_{CC})||L_{DAP_j}||T_{CC,j,DAP_j}||sigPk_{CC}||sigSign_{sigSk_{CC}}(H_2(H_1(pk_{CC})||L_{DAP_j}||T_{CC,j,DAP_j}))$, where $H_1(pk_{CC})$ is the identity of CC, L_{DAP_j} is the ledger in Table 2 that keeps all the transactions, T_{CC,j,DAP_j} is the time stamp, $sigPk_{CC}$ is the public key of the signature, and $sigSign_{sigSk_{CC}}(H_2(H_1(pk_{CC})||L_{DAP_j}||T_{CC,j,DAP_j}))$ is the signature.

Bill Calculation Once the prices are available that are sent by CC, SP checks first L_{DAP_j} ledger, its time-stamp and verifies the signature. It the verification goes well, SP issue the bills to the users using their smart meters' anonymized identities in the ledger L_{OCD_j}. These anonymized identities are located between $m/2+3$ and $3m/2+2$ (most significant bits of each message in the L_{OCD_j} ledger) bits of each smart meter's message in L_{OCD_j} ledger. Then SP sends $10||BC_j||H(H_1(sk_{i,CC})||ID_i)||T_{SP,j,l}||B_{i,j,k}||sigPk_{SP}||$
$||sigSign_{sigSk_{SP}}(H_2(10||BC_j||H(H_1(sk_{i,CC})||ID_i)||T_{SP,j,l}||B_{i,j,k}))$ to CC.
Here, BC_j is the name of the ledger. Two bits 10 is used for the type of the ledger which is the bill calculation ledger, $H(H_1(sk_{i,CC})||ID_i)$ is the identity of the smart meter, $T_{SP,j,l}$ is the time-stamp, where the time stamp is created by SP for ledger j and it is SP's lth message, $B_{i,j,k} = p_j c_{i,j,k}$ (multiplication of electricity price of time interval j with user i's kth consumption) is the kth usage bill of SM_i. SP also sends
$sigSign_{sigSk_{SP}}(H_2(10||BC_j||H(H_1(sk_{i,CC})||ID_i)||T_{SP,j,l}||B_{i,j,k}))$ to CC for the data integrity.
Once CC checks the verification of the message sent by SP for the bill calculation ledger and the signature is valid, CC puts the message in to L_{BC_j} ledger. Then CC sends L_{BC_j} ledger in Tab. 3, a time-stamp and its signature to the users. CC sends bill calculation ledger L_{BC_j} to everyone (or puts into a public bulletin) as follows:
$H_1(pk_{CC})||L_{BC_j}||T_{CC,j,BC_j}||sigPk_{CC}||sigSign_{sigSk_{CC}}(H_2(H_1(pk_{CC})||L_{BC_j}||T_{CC,j,BC_j}))$,
where $H_1(pk_{CC})$ is the identity of CC, L_{BC_j} is the ledger in Table 3 that keeps all the transactions, T_{CC,j,BC_j} is the time stamp, $sigPk_{CC}$ is the public key of the signature, and $sigSign_{sigSk_{CC}}(H_2(H_1(pk_{CC})||L_{BC_j}||T_{CC,j,BC_j}))$ is the signature.
Each user i then checks the signature if it is valid (sent from CC), then extracts its anonymized identities that they are generated from SM_i and computes its total bill for L_{BC_j} ledger as follows: $B_{i,j} = \sum_k B_{i,j,k}$, where k is the number how many times SM_i sent data to a ledger, index j is the time range, and index i is the smart meter's (user's) identity index.
As a note that the user is able to extract its bills from the ledgers since the smart meter's identity is given to the user by the smart meter in the outsourcing data subsection. The smart meter sends each anonymized identity in each message to the user.

Remark 5. It is possible that each user can be able to compute dynamic prices and its bill since all the ledgers are kept by the users. Moreover, each user can check if its bill honestly computed.

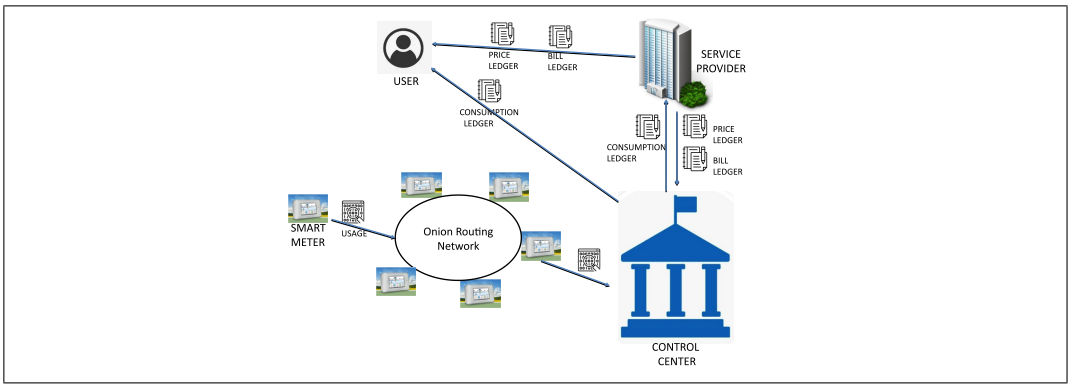

Fig. 4: New System architecture and workflow for Reducing Communication Complexity between Users and CC.

6 Reducing Communication Complexity Between CC and Users

It is possible that SP can send the price of each time period and billing information directly to users instead of sending it to CC in subsection 5.1. This is going to be more realistic scenario in the system. In this case, SP needs to compute, $sk_{SP,U_i} = g^{sk_{SP}sk_{U_i}}$ instead of computing $sk_{SP,CC} = g^{sk_{SP}sk_{CC}}$, where sk_{U_i} is the secret key of user that has smart meter i in its home. Then SP puts
$01||DAP_j||H(H_1(sk_{SP,U_i})||ID_{SP})||T_{SP,j}||p_j||sigPk_{SP}||$
$||sigSign_{sigSk_{SP}}(H_2(01||DAP_j||H(H_1(sk_{SP,U_i})||ID_{SP})||T_{SP,j}||p_j)$, into Ledger L_{DAP_j}, for U_i who has smart meter i in its home. Thus, SP generates L_{DAP_j} for U_i, U_{i+1} and U_{i+2} (user i, user $i+1$ and user $i+2$) and is given in Figure 4. Then SP sends it to the users and CC,
$H_1(pk_{SP})||L_{DAP_j}||T_{SP,j,DAP_j}||sigPk_{SP}||sigSign_{sigSk_{SP}}(H_2(H_1(pk_{SP})||L_{DAP_j}||T_{SP,j,DAP_j}))$, where $H_1(pk_{SP})$ is the identity of SP, L_{DAP_j} is the ledger in Table 4 that keeps all the transactions, T_{SP,j,DAP_j} is the time stamp, $sigPk_{SP}$ is the public key of the signature, and $sigSign_{sigSk_{SP}}(H_2(H_1(pk_{SP})||L_{DAP_j}||T_{SP,j,DAP_j}))$ is the signature.
Since each user i (U_i) can also compute $H(H_1(sk_{U_i,SP})||ID_{SP})$, it can figure out its identity in the ledger. So it can be able retrieve the price in the ledger.
Moreover, SP also sends directly to the users
$H_1(pk_{SP})||L_{BC_j}||T_{SP,j,BC_j}||sigPk_{SP}||sigSign_{sigSk_{SP}}(H_2(H_1(pk_{SP})||L_{BC_j}||T_{SP,j,BC_j}))$,
where $H_1(pk_{SP})$ is the identity of SP, L_{BC_j} is the ledger in Table 3 that keeps all the transactions, T_{SP,j,BC_j} is the time stamp, $sigPk_{SP}$ is the public key of the signature, and $sigSign_{sigSk_{SP}}(H_2(H_1(pk_{SP})||L_{BC_j}||T_{SP,j,BC_j}))$ is the signature.
In this case, instead of CC, SP sends all the pricing and billing information to the users (Fig. 4). Thus, CC does not need to compute and generate any Ledger. Computational complexity is going to be decreased for CC. Moreover, the communication complexity between CC and users are going to be decreased. Users can be able to verify the electricity prices and their bills themselves.

7 Discussion

When a smart meter sends a consumption data to CC, the smart meter also needs to send its identification part ($H(H_1(sk_{i,CC})||ID_i)$) to its consumer/user for allowing to compute its bill. But the sending time of this message somehow needs to be anonymized as well to eliminate leaking privacy of the user. For example, it is going to be a problem if a consumption data is sent to CC

Table 4: New Permissioned Ledger L_{DAP_j}: Data Aggregation and Pricing Ledger for time interval j

1. $01 \| DAP_j \| H(H_1(sk_{SP,U_i}) \| ID_{SP}) \| T_{SP,j,1} \| p_j \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(01 \| DAP_j \| H(H_1(sk_{SP,U_i}) \| ID_{SP}) \| T_{SP,j,1} \| p_j))$.
2. $01 \| DAP_j \| H(H_1(sk_{SP,U_{i+1}}) \| ID_{SP}) \| T_{SP,j,2} \| p_j \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(01 \| DAP_j \| H(H_1(sk_{SP,U_{i+1}}) \| ID_{SP}) \| T_{SP,j,2} \| p_j))$.
3. $01 \| DAP_j \| H(H_1(sk_{SP,U_{i+2}}) \| ID_{SP}) \| T_{SP,j,3} \| p_j \| sigPk_{SP} \| sigSign_{sigSk_{SP}}(H_2(01 \| DAP_j \| H(H_1(sk_{SP,U_{i+2}}) \| ID_{SP}) \| T_{SP,j,3} \| p_j))$.
.

at the time of t_1 and $H(H_1(sk_{i,CC})||ID_i)$ is sent to U_i at the time of $t_1 + \epsilon$, where ϵ is a small number (probably seconds). Since these messages are very closed to each other, any external or internal attacker even the service provider can be able to map usages to users easily. In order to eliminate this attack, users/consumers also need to be included in the onion routing network. In this case, the smart meters and consumers can act as onion routers. Another way to tackle this problem is to program smart meters to send messages to their consumers/users every predefined time ranges (every 5-10 minutes) even there is no consumption is done by a consumer. In this case, there should be a global time clock that every smart meter needs to have.
In this paper, there is a fully trusted party, CC, that is run and control by a government. CC is trusted in order to eliminate fake smart meters to send fake consumption data. Since all the smart meters' identities are anonymized, CC needs to distinguish authorized smart meters from fake smart meters. Otherwise, there is no control on the system and this results in unbalancing loads and blackout on the system.

8 Security Analysis

In this section, we show that our scheme satisfies the following security goals.

8.1 Consumer Privacy

In our protocol, except for CC, no one can gain knowledge of any private information of a user. Thanks to average based usage splitting mechanism, onion routing network protocol and identity anonymizing method, any user's data consumption is going to be private to untrusted parties (external/internal adversary). Any party except CC can not map any energy usage in the ledger to an individual user. The consumer privacy is based on the well-known cryptographic hardness assumption which is Computational Diffie Hellman (CDH) assumption. This assumption assures that when given group elements g, g^a, g^b the adversary can not compute g^{ab} with non-negligible probability ϵ which it is a function of m, where $a, b \in Z_q$ values are random values (secret keys). In our protocol, when SM_i puts the consumption into the ledger, it computes the CDH value of $H_1(g^{sk_{CC}sk_i})$ which it is observed in the ledger. Since $g, g^{sk_{CC}}, g^{sk_i}$ values and H_1 are available in public, untrusted party needs to compute $g^{sk_{CC}sk_i}$ from $g, g^{sk_{CC}}, g^{sk_i}$ to figure out each user's consumption data. Moreover, using a long term key management mechanism between the smart meters and CC in signature scheme provides user anonymity. Each user generates a different signature and a different signature verification key ($SigPk$) for each message, then the smart meter sends them in the message. Any attacker is not able to map a given message to any user. Moreover, the adversary can not infer any useful information by having signature public keys of any smart meter since the signature public keys are always random values. For example, SM_i's signature public keys $sigPk_{i,1} = g^{H_3(f_{K_i}(000001))}$ and $sigPk_{i,2} = g^{H_3(f_{K_i}(000010))}$ are two different random values since $H_3(f_{K_i}(000001))$ and $H_3(f_{K_i}(000010))$ two random values because of pseudorandom generator and pseudorandom function property that is explained in [11].

8.2 Usage Data Integrity

In our protocol, each user's consumption data is sent to the ledger by the corresponding smart meter as a message. The integrity of each message is checked three times. Once the message is a valid message that comes from the authorized smart meter, the message is kept in the ledger. Then anyone can be able to check the integrity of the message by the given signature that the combination of values including time stamp, data usage and hash value of identity of the user (this is a secret value to any untrusted party). If CC receives a message from a smart meter, CC

checks the identity of the smart meter. Checking the most significant bits of the message located between $m/2+3$ and $3m/2+2$ is going to be sufficient (step 1 in the outsourcing consumption subsection). If the message comes from an authorized smart meter, CC checks the signature's public key (verification key $sigPk$) inside the message. $sigPk$ can be computed by CC. If the $sigPk$ is also generated from the same smart meter, then CC checks the signature. The signature verification can fail because the message is changed by the user or an internal attacker corrupts the smart meter. In this case CC asks SM_i to send the correct message so that CC adds this consumption data to the ledger. If the smart meter does not respond or new message is also wrong, the smart meter is going to be excluded in the system by CC. The user is not able to generate a signature with a fake data consumption since it does not know the secret key for the signature ($sigSK_i$). This secret key ($sigSK_i$) for the signature is generated by K_i and only known by SM_i and CC. SP can also give incorrect pricing data and issues an incorrect bill to a user. This attack is also going to be caught by CC and any user in the system. This integrity mechanism prevents our system form any internal/external adversaries. Our system is also resilient to replay attacks since CC has the previous messages from the smart meters. CC can catch these attacks.

8.3 Authentication

In the initialization phase, each party (smart meter, service provider) registers its public key to the control center (CC), then each smart meter's public key is available to all parties except their secret keys. The secret key of a smart meter is kept private to itself. Using this authentication, any untrusted party can not send any message to the ledger. Any fake message/transaction is going to be caught by CC and is not going to be stored in the ledger. This mechanism protects the system from the untrusted party to launch reply attacks. Moreover, any untrusted party can not unbalanced the load by giving different usage than a user consumes.

9 Conclusion and Future Work

We presented a Privacy-Preserving Data Aggregation and Dynamic Billing System in Smart Grid Using Permissioned Blockchain that efficient in terms of computing aggregation data and billing. In our scheme, each user can compute its bill individually that provides integrity of the billing data. Using permissioned blockchain provides a transparent system that each entity can follow the transactions.
As future work, I would like to implement our scheme to check its performance. I would also like to provide an anonymous payment protocol after each user gets its bill. This is also important that it should be anonymous. Moreover, I would like to design a new protocol that provides anonymous communication to improve the computational cost of users and smart meters.

References

1. A. Abdallah and X. Shen. Lightweight security and privacy preserving scheme for smart grid customer-side networks. *IEEE Transactions on Smart Grid*, 8:1–1, 08 2015.
2. A. Agarkar and H. Agrawal. R-lwe based lightweight privacy preserving scheme for smart grid. pages 410–415, 12 2016.
3. I. Ali, E. Khan, and S. Sabir. Privacy-preserving data aggregation in resource-constrained sensor nodes in internet of things: A review. *Future Computing and Informatics Journal*, 3, 12 2017.
4. M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac. Smart meter data privacy: A survey. *IEEE Communications Surveys and Tutorials*, 19:2820–2835, 2017.

5. M. Badra and S. Zeadally. Lightweight and efficient privacy-preserving data aggregation approach for the smart grid. *Ad Hoc Networks*, 64:32–40, 2017.
6. F. Borges, D. Demirel, L. Bock, J. A. Buchmann, and M. Mühlhäuser. A privacy-enhancing protocol that provides in-network data aggregation and verifiable smart meter billing. *2014 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2014.
7. A. Braeken, P. Kumar, and A. J. Martin. Efficient and privacy-preserving data aggregation and dynamic billing in smart grid metering networks. *Energies*, 2018.
8. L. Chen, R. Lu, and Z. Cao. Pdaft: A privacy-preserving data aggregation scheme with fault tolerance for smart grid communications. *Peer-to-Peer Networking and Applications*, 8(6):1122–1132, Nov 2015.
9. T. Eccles and B. Halak. Performance analysis of secure and private billing protocols for smart metering. *Cryptography*, 1:20, 2017.
10. S. Finster and I. Baumgart. Smart-er: Peer-based privacy for smart metering. pages 652–657, 04 2014.
11. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. ACM*, 1986.
12. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, Apr. 1988.
13. P. Gope and B. Sikdar. An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids. *IEEE Internet of Things Journal*, PP:1–1, 05 2018.
14. D. He, S. Zeadally, H. Wang, and Q. Liu. Lightweight data aggregation scheme against internal attackers in smart grid using elliptic curve cryptography. *Wireless Communications and Mobile Computing*, 2017, 2017.
15. J.-H. Im, H.-Y. Kwon, S.-Y. Jeon, and M.-K. Lee. Privacy-preserving electricity billing system using functional encryption. 2019.
16. A. Laszka, A. Dubey, M. Walker, and D. Schmidt. Providing privacy, safety, and security in iot-based transactive energy systems using distributed ledgers. In *Proceedings of the Seventh International Conference on the Internet of Things*, IoT '17, New York, NY, USA, 2017. Association for Computing Machinery.
17. S. Li, K. Xue, Q. Yang, and P. Hong. Ppma: Privacy-preserving multisubset data aggregation in smart grid. *IEEE Transactions on Industrial Informatics*, 14:462–471, 2018.
18. G. Liang, J. Zhao, F. Luo, S. S. R. Weller, and Z. Y. Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 8:1630–1638, 2017.
19. Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng. A practical privacy-preserving data aggregation (3pda) scheme for smart grid. *IEEE Transactions on Industrial Informatics*, 15:1767–1774, 2019.
20. M. A. Mustafa, S. Cleemput, A. Aly, and A. Abidin. A secure and privacy-preserving protocol for smart metering operational data collection. *IACR Cryptology ePrint Archive*, 2018:101, 2018.
21. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*, 03 2009.
22. J. Ni, K. Zhang, X. Lin, and X. Shen. Balancing security and efficiency for smart metering against misbehaving collectors. *IEEE Transactions on Smart Grid*, 10:1225–1236, 2019.
23. K. Ohara, Y. Sakai, F. Yoshida, M. Iwamoto, and K. Ohta. Privacy-preserving smart metering with verifiability for both billing and energy management. In *AsiaPKC@AsiaCCS*, 2014.
24. O. Oksuz. Privacy preserving data aggregation and dynamic billing system in smart grid using permissioned blockchain. In *CRIS*, pages 53–69, 2020.
25. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.
26. P. Syverson, D. Goldschlag, and M. Reed. Anonymous connections and onion routing. pages 44–54, 01 1997.

27. S. Tan, D. De, W.-Z. Song, J. Yang, and S. K. Das. Survey of security advances in smart grid: A data driven approach. *IEEE Communications Surveys and Tutorials*, 19:397–422, 2017.
28. E. Vahedi, M. Bayat, M. R. Pakravan, and M. R. Aref. A secure ecc-based privacy preserving data aggregation scheme for smart grids. *Comput. Netw.*, 129(P1):28–36, Dec. 2017.
29. Z. Wang. An identity-based data aggregation protocol for the smart grid. *IEEE Transactions on Industrial Informatics*, 13:2428–2435, 2017.
30. Z. Wang, H. Xie, and Y. Xu. Security analysis of an identity-based data aggregation protocol for the smart grid. In I. Traore, I. Woungang, S. S. Ahmed, and Y. Malik, editors, *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 63–73, Cham, 2018. Springer International Publishing.
31. S. C. Yip, K. Wong, R. Phan, S.-W. Tan, I. Ku, and W. Hew. A privacy-preserving and cheat-resilient electricity consumption reporting scheme for smart grids. pages 1–5, 07 2014.
32. L. Zhang, O. Oksuz, L. Nazaryan, C. Yue, B. Wang, A. Kiayias, and A. Bamis. Encrypting wireless network traces to protect user privacy: A case study for smart campus. In *2th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2016, New York, NY, USA, October 17-19, 2016*, pages 1–8, 2016.

Author

Ozgur Oksuz received his Ph.D. degree at the Computer Science and Engineering department from the University of Connecticut, USA in 2016. He was a postdoctoral fellow at Washington State University, USA from October, 2016 to December, 2017. He is an assistant professor at the Computer Engineering department at Adiyaman University in TURKEY. His research interests are in the areas of applied and theoretical cryptography, computer security and blockchain technology.