PARK: A Privacy-preserving Aggregation Scheme with Adaptive Key Management for Smart Grid

Kuan Zhang, Rongxing Lu, Xiaohui Liang, Jian Qiao, and Xuemin (Sherman) Shen
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1
Email:{k52zhang, rxlu, x27liang, jqiao, xshen}@bbcr.uwaterloo.ca

Abstract—**Smart Grid, as one kind of promising sustainable power systems, can rely on two-way communications and emerging smart meters to intelligently control the residential electricity usage. However, due to the inherent open communication media as well as the limited communication, computation and storage capabilities of smart meters, security concerns raise and hinder the further flourish of smart grid. In this paper, we propose a privacy-preserving aggregation (PARK) scheme with adaptive key management and revocation, to prevent user's data from being disclosed to untrusted entities in smart grid. Specifically, we first investigate a lightweight aggregation scheme with efficient aggregate authentication, which protects the individual user's data from disclosure to the untrusted aggregator. Furthermore, we propose an adaptive key management mechanism with effective revocation, where users can automatically update their encryption keys if no user joins or departs from the system. The expiry time of the key is determined by user's reputation for the adaptive key management. Finally, the security analysis demonstrates that the PARK can achieve privacy preservation, forward and backward secrecy at the same time, while the performance evaluation shows that the PARK consumes reasonable costs.**

I. Introduction

Smart Grid has emerged as a promising and value-added next generation of sustainable power system, which intelligently controls the residential electricity usage, and manages the power generation, transmission, and distribution [1]. Basically, the intelligence of smart grid relies on two-way communications, where the control center exchanges information with other entities in smart grid and globally balances the power flow, as shown in Fig. 1. Furthermore, smart meters, deployed in each individual residential house, are intended to improve efficiency, reliability, and enable dynamic billing and pricing [2]. Recently, smart grid has attracted ever-increasing attention from not only power and control experts but also the communication society.

In smart grid, due to the open communication media and the limited smart meter's capability, including computation, communication, and storage, some traditional security schemes cannot be directly deployed in smart grid. As a result, a series of attacks perpetrate against smart meters over the past several years. Such attacks cost a single U.S. electric utility hundreds of millions of dollars annually [3]. Some malicious users tamper with the metering data and upload modified fake data to the control center. In addition, since the metering data in smart grid are highly relevant to the individual user's life, the data disclosure might violate user's sensitive or private information, such as the life style, attributes, or preferences. One possible solution is to encrypt the user's data in the ciphertext to protect them from passively eavesdropping [4]. Usually, the statistical data aggregation interval is much shorter than that of the user's electricity usage report, and the volume of aggregated data is considerably large. To efficiently aggregate the local users' data and relay the relevant billing or usage information, some aggregators, shown in Fig. 1, are thus allocated in the residential area to act as local collectors or gateways. However, these local aggregators are possibly compromised by the outside adversaries and might illegally violate legitimate users' private data. To solve this problem, some researchers [4] attempt to confine the aggregator's capabilities so that it can only extract the aggregated statistical data rather than the exact individual user's data.

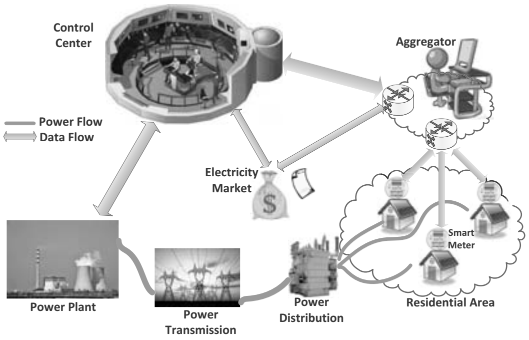

Fig. 1. Smart Grid Architecture

At the same time, how to efficiently manage the encryption keys is still a challenging issue in aggregation for smart grid. The most widely applied schemes [2], [5] need to pre-establish a specific structure, like binary tree, group or ring among users' keys. However, some users might be revoked or depart from the smart grid system, and should not be able to access the system [6]. The pre-established structure might not work as expected, and users have to re-build new structures which consume a large volume of communication overheads. Furthermore, with the existing tons of residential users, the control center has to distribute a giant amount of keys to achieve security requirements. Therefore, an efficient and effective key management with revocation is essential for smart grid, and the cryptography should be lightweight as well.

In this paper, we propose a Privacy-preserving AggRegation scheme with the adaptive Key management and revocation (PARK) for smart grid. The PARK is characterized by privacy-preserving aggregation achieving data confidential, efficient key management, and effective revocation with forward and backward secrecy. Specifically, the major contributions of this paper are three-fold.

- Firstly, we propose a privacy-preserving aggregation scheme to enable the aggregator to learn the statistical information without knowing the data of individual user. We utilize bi-directional Hash chains to produce the encryption keys for each residential user. Furthermore, the aggregator is able to effectively extract the statistical information, such as the sum, from the large volume of residential users' data. We also explore an aggregate authentication scheme which can significantly decrease computational overheads.
- Secondly, we investigate an adaptive key management with revocation to achieve forward and backward secrecy. If no user joins or departs from the system, all users automatically update their keys based on bi-directional Hash chains. The length of Hash chains is determined by user's reputation. The user with a higher reputation value can obtain a key pair with a longer expiry time compared with the misbehaved user. When a user is revoked, the control center updates the revocation list and re-distributes encryption keys.
- Finally, we analyze the security properties of the PARK, and prove that the PARK achieves privacy preservation, forward and backward secrecy at the same time. In additional, the performance evaluation shows that the PARK has a more efficient key management compared with other schemes.

The remainder of this paper is organized as follows: Network model and design goals are presented in Section II. In Section III, we present the details of the PARK, followed by the security analysis and performance evaluation in Sections IV and V, respectively. The related works are reviewed in Section VI. Finally, Section VII concludes the paper.

II. Problem Definition

In this section, we describe the network model and identify our primary design goals to establish an adaptive key management for aggregation in smart grid. Then, we present the security model and illustrate the security requirements.

A. Network model

We consider a smart grid system consisting of a trust control center, a small portion of untrusted aggregators, and N residential users. The details of these entities are presented as follows.

- *Trusted Control Center (TCC)* is a trustable, powerful, and storage-rich entity, and bootstraps the whole system in the initialization phase. When bootstrapping, the TCC generates secret master keys for each legitimate user, and users' certificates for further authentication. Afterwards, the TCC are not involved in the low level data aggregation. Upon receiving attack reports from residential users, the TCC revokes the malicious users and adjusts the users' encryption keys.
- *Aggregator (AG)* is a local gateway, which directly collects the metering data from each individual user in the residential area. Since the AG locates in the residential area where the TCC cannot provide a closed and definitely secure communication network for the AG, it is possible for outside adversaries to compromise the AG. Therefore, the AG is an untrusted entity in smart grid.

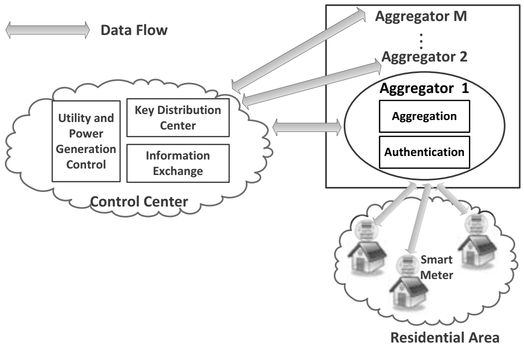

Fig. 2. Network model for Smart Grid

- *Residential users* are denoted by $U = \{u_1, u_2, ..., u_N\}$. Each residential user is equipped with a smart meter which monitors the electricity usage in a real time fashion and periodically reports the usage data to the AG. An individual residential user should first register to the TCC for the profiles (unique identity), certificates and key materials. Then, u_i should securely keep them and generate session keys in each time slot.

B. Security model

We consider an honest-but-curious security model, where all entities honestly follow the protocols, but some of them, especially the aggregator, are curious about other users' data. This is because the aggregators are located in the proximity of the residential area and out of the control of the TCC. The AG might be compromised by the outside adversaries and illegally violate legitimate users' private data.

In addition, some inside users might have misbehaviors, where some existing solutions [7], [8] can efficiently detect them. We mainly focus on the revocation issues. After the detection of such inside attacks, the TCC should timely draw such faulted users into the revocation list and update the encryption keys for the remaining legitimate users.

C. Design goals

Our design goal is to develop a privacy-preserving aggregation scheme with efficient and effective key management and revocation for smart grid.

1) Security goals: Our primary security goal is to protect the individual user's data from being disclosed during the aggregation, while achieving forward and backward secrecy.

- **Privacy Preservation**: The proposed scheme should be able to not only enable the AG to extract the desirable statistical data, for example, the sum, but also disable the AG to decrypt the individual user's data. In other words, the individual user's data cannot be directly revealed to the AG.
- **Forward and Backward Secrecy**: If a user joins the system, it is assigned key materials for further secure communications. These assigned keys should not be able to derive

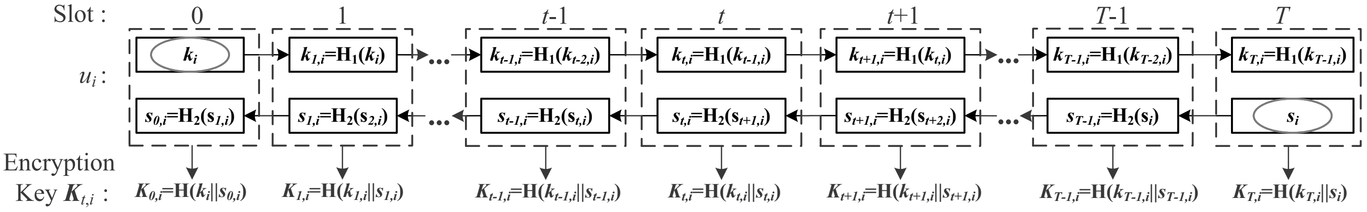

Fig. 3. Hash chain for an individual residential user

the previous keys of other users. Otherwise, the new-joining user might be compromised by the adversaries to recover the previous historical data. On the other hand, if a user departs from the system or is revoked by the TCC, this user should not be able to access the communications in the following slots. In summary, the proposed scheme should be able to achieve forward and backward secrecy.

2) Efficiency goals: We also intend to improve the communication, computation, and storage efficiency, especially during the key management and revocation, for the sustainable smart grid. In specific, the key update should consume the minimized overheads during the revocation.

III. Proposed PARK protocol

In this section, we present the details of our proposed PARK scheme. Firstly, we propose a privacy-preserving aggregation scheme with an efficient aggregate authentication. Then, we explore bi-directional Hash chains for the key management and revocation. If no user joins and departs from the system, all users automatically update their encryption keys. The expiry time of user's key is determined by the user's reputation. A user with a higher reputation value can update its keys in a longer period compared with the misbehaved user. Therefore, the TCC can frequently audit the misbehaved user.

A. Privacy-preserving Aggregation

• *Key Generation and Distribution*: The TCC chooses an additive cyclic group $\mathbb{G}_1$ with the order q, and P is the generator of $\mathbb{G}_1$. $\mathsf{H}_0 : \{0,1\}^* \to \mathbb{G}_1$. Let $\mathbb{G}$ be a cyclic group of prime order q. A bilinear pairing e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}$ [9], [10] between $\mathbb{G}_1$ and $\mathbb{G}$ exists if it can be efficiently calculated. We have i) For random numbers $a, b \in \mathbb{Z}_q^*$, $e(aP, bP) = e(P, P)^{ab}$; ii) Non-degenerated: $e(P, P) \neq 1$. Taking a security parameter κ as input, a probabilistic algorithm outputs a tuple (q, $\mathbb{G}_1$, $\mathbb{G}$, e, P, H_0). The AG selects a random number $x \in \mathbb{Z}_q^*$ as the secret key SK. $\mathsf{PK} = xP$ is the AG's public key.

Let $\mathbb{G}_T$ be a multiplicative cyclic group with the order q'. H_T : $\{0,1\}^* \to \mathbb{G}_T$. H, H_1 and H_2 : $\{0,1\}^* \to \mathbb{Z}_q^*$ are cryptographic Hash functions. The TCC selects N forward seeds $\mathcal{K} = \{k_1, k_2, \cdots, k_N\}$ and N backward seeds $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$, where random numbers k_i and $s_i \in \mathbb{Z}_q^*$. Then, a key pair (k_i, s_i) establishes bi-directional (forward and backward) Hash chains [11] with $\mathsf{H}_1(\cdot)$ and $\mathsf{H}_2(\cdot)$ for individual user u_i where $i \in \{0, 1, \cdots, N\}$, shown in Fig. 3.

The key pair (k_i, s_i) is distributed to u_i and securely kept. In time slot t, u_i computes the forward key $k_{t,i} = \mathsf{H}_1(k_{t-1,i})$ and backward key $s_{t,i} = \mathsf{H}_2(s_{t+1,i})$, respectively. Then, u_i obtains the data encryption key $\mathbf{K}_{t,i} = \mathsf{H}(k_{t,u_i}||s_{t,u_i})$ for time slot t. Every encryption key $\mathbf{K}_{t,i}$ is only valid during time slot t. In the next time slot $t+1$, u_i re-calculates the forward and backward keys for the new encryption key, and releases the previous forward keys. Since the backward keys are useful in computing the encryption key for the coming slots, the backward keys are also securely kept.

The TCC calculates the sum of all the users' encryption keys, $\mathbf{K}_{t,AG} = \sum\limits_{i=1}^{N} \mathbf{K}_{t,i}$, in time slot t. The decryption key $\mathbf{K}_{t,AG}$ is distributed to the AG.

• *Aggregation*: 1) A residential user u_i first selects a random number $x_i \in \mathbb{Z}_q^*$ as the secret key SK_i. x_iP is the public key PK_i of user u_i. When u_i measures the metering data $m_{t,i}$ in time slot t, u_i blinds the plaintext $m_{t,i}$ as $\hat{m}_{t,i} = (m_{t,i} + \mathbf{K}_{t,i}) \bmod q'$. Then, u_i produces the ciphertext $c_{t,i} = \mathsf{H}_T(t)^{\hat{m}_{t,i}}$. Afterwards, u_i generates the signature

$$\mathsf{sign}_i = x_i\mathsf{H}_0(t) + x_i\mathsf{H}(c_{t,i})\mathsf{PK} \tag{1}$$

Finally, u_i sends the tuple $(ID_i||c_{t,i}||\mathsf{sign}_i||t)$ to the AG.

2) When the AG receives N packets in time slot t, the AG first verifies the authenticity. The AG computes the sum of all the signatures $\sum\limits_{i=1}^{N} \mathsf{sign}_i$, and checks

$$e\left(\sum_{i=1}^{N} \mathsf{sign}_i, P\right) \overset{?}{=} e(\mathsf{H}_0(t), \sum_{i=1}^{N} \mathsf{PK}_i) \cdot \prod_{i=1}^{N} e(x\mathsf{H}(c_{t,i})P, \mathsf{PK}_i). \tag{2}$$

If Eqn. 2 holds, all N packets are authenticated. The correctness can be proved as

$$\begin{aligned} e\left(\sum_{i=1}^{N} \mathsf{sign}_i, P\right) &= \prod_{i=1}^{N} e(x_i\mathsf{H}_0(t) + x_i\mathsf{H}(c_{t,i})xP, P) \\ &= \prod_{i=1}^{N} e(\mathsf{H}_0(t), x_iP) \cdot \prod_{i=1}^{N} e(x\mathsf{H}(c_{t,i})P, x_iP) \\ &= e(\mathsf{H}_0(t), \sum_{i=1}^{N} \mathsf{PK}_i) \cdot \prod_{i=1}^{N} e(x\mathsf{H}(c_{t,i})P, \mathsf{PK}_i) \\ &= e(\mathsf{H}_0(t), \sum_{i=1}^{N} \mathsf{PK}_i) \cdot \prod_{i=1}^{N} e(\mathsf{PK}, \mathsf{PK}_i)^{\mathsf{H}(c_{t,i})} \end{aligned} \tag{3}$$

The aggregate authentication [12] can significantly improve the authentication efficiency. The pairing operation

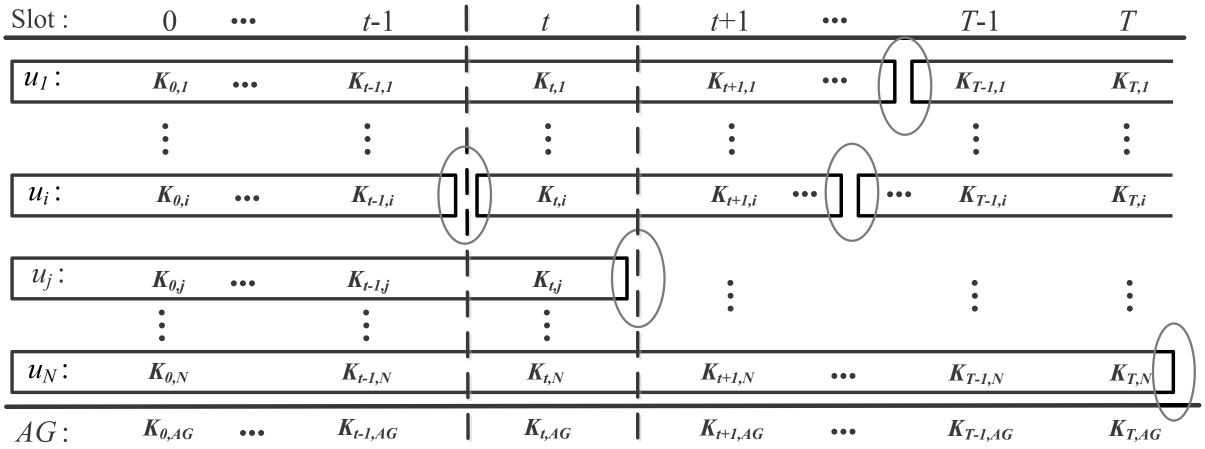

Fig. 4. Hash chains and Adaptive Key Management

$e(\mathsf{H}_0(t), \sum_{i=1}^{N} \mathsf{PK}_i)$ can be pre-computed by the AG, since $\mathsf{H}_0(t)$ is determined by the time slot t, and PK_i is public key of u_i. Furthermore, $e(\mathsf{PK}, \mathsf{PK}_i)$ can also be pre-computed. In every aggregate authentication, only one pairing operation $e(\mathsf{H}_0(t), \sum_{i=1}^{N} \mathsf{PK}_i)$ is required so that the authentication efficiency can be considerably improved in the PARK.

Afterwards, the AG computes $\mathsf{H}_T(t)^{-\mathbf{K}_{t,AG}}$ and multiples it with the products of all $c_{t,i}$s. Finally, the AG extracts the sum of the aggregated data as

$$\mathsf{H}_T(t)^{-\mathbf{K}_{t,AG}} \cdot \prod_{i=1}^{N} \mathsf{H}_T(t)^{\hat{m}_{t,i}} = \mathsf{H}_T(t)^{\left(\sum_{i=1}^{N}(m_{t,i}+\mathbf{K}_{t,i})-\mathbf{K}_{t,AG}\right)} \tag{4}$$

but learns nothing else about each individual user.

Since $\mathbf{K}_{t,AG} = \sum_{i=1}^{N} \mathbf{K}_{t,i}$, the AG has $\mathsf{H}_T(t)^{\sum_{i=1}^{N} m_{t,i}}$. Furthermore, the size of smart metering data is small. The AG adopts brute-force search or Pollard's lambda method [13] to obtain the power of $\mathsf{H}_T(t)$. Therefore, the AG can extract the sum $\sum_{i=1}^{N} m_{t,i}$ of the aggregated data in time slot t.

B. Adaptive Key Management with Revocation

To achieve forward and backward secrecy, we propose an adaptive key management with revocation in the PARK. Intuitively, user u_i has a reputation value R_i related to its historical behaviors including misbehaviors, like false data uploading, and regular behaviors. If u_i uploads more false data, its reputation dwindles correspondingly. When some malicious behaviors are detected on u_i, u_i is promptly revoked by the TCC, and R_i is re-set to 0. On the other hand, the user's reputation could augment when he/she detects some attacks and reports other users' malicious behaviors. Let BM_i^t and BR_i^t be the number of misbehaviors and that of regular behaviors from u_i before time slot t. $R_i = f(BM_i^t, BR_i^t)$, where f is a function satisfying the rules: 1) when BM_i^t increases, R_i decreases; 2) when BR_i^t increases, R_i accordingly increases.

Algorithm 1 Revocation

```
Procedure: Revocation
$r$ users $u_1, \cdots, u_r$ are revoked in time slot $t$. The TCC picks them into
  the revocation list $\mathcal{RL}$.
The TCC distributes $(k_{t,i}, s_i)$ to the AG, where $i \in \{1, \cdots, r\}$.
for $j = t : T$ do
    for $i = 1 : r$ do
        if The expiry time $T_i > j$ then
            The AG computes $\mathbf{K}_{j,i} = \mathsf{H}(k_{j,i}||s_{j,i})$, where $k_{j,i}$ and $s_{j,i}$
              are forward and backward keys for $u_i$ in time slot $j$.
            The AG updates the decryption key $\mathbf{K}_{t,AG} = \mathbf{K}_{t,AG} - \mathbf{K}_{j,i}$.
        end if
    end for
end for
End procedure
```

According to the reputation R_i and $R_{max} = max\{R_i, R_2, \cdots, R_N\}$, the max reputation value of N users, u_i can apply the key pair (k_i, s_i) with the expiry time $T_i = T \times \frac{R_i}{R_{max}}$. Here, T is the max expiry time of key pairs for the users in the system. After the expiry time T_i, u_i cannot produce any valid encryption key based on the previous key pair (k_i, s_i). As a result, u_i has to re-apply to the TCC for a new key pair. At the same time, the TCC estimates u_i's behaviors in the past T_i slots to re-calculate a new reputation value R_i', and assign a new key pair with the new expiry time $T_i' = T \times \frac{R_i'}{R_{max}}$ to u_i. By doing so, the users with higher reputation values can obtain longer Hash chains, while those with lower reputation have shorter Hash chains. The TCC can check the later in a shorter period to detect their malicious behaviors in a real-time fashion. Denote $th > 0$ as a threshold of reputation value. If the TCC detects u_i's reputation value $R_i < th$, the TCC rejects to assign the key pair for u_i.

If a user u_i is revoked or departs from the system in time slot t, the TCC first draws u_i into the revocation list $\mathcal{RL}$ and posts it to the public. Then, the TCC assigns the AG with u_i's remaining Hash chains $(k_{t,i}, s_i)$. In time slot t, the AG calculates u_i's inherent encryption key $\mathbf{K}_{t,i}$ as shown in Fig. 3, and updates the AG's decryption key as $\mathbf{K}_{t,AG} = \mathbf{K}_{t,AG} -$

$\mathbf{K}_{t,i}$. The detailed procedures are demonstrated in Alg. 1. After the revocation, the aggregation can still be proceeded in the coming time slots.

If r new users $u_1, \cdots, u_r$ join the system and apply keys, the TCC assigns their keys at first. Then, the TCC collects their keys and computes the sum $\mathbf{SK}_j = \sum_{i=1}^{r} \mathbf{K}_{j,i}$ in time slot j. The TCC also randomly selects a legitimate user u_d and updates u_d's encryption key as $\mathbf{K}_{j,d} = \mathbf{K}_{j,d} - \mathbf{SK}_j \mod q'$. In the coming time slots, u_d first calculates its encryption key based on the assigned Hash chains, and then adds the new-distributed key $\mathbf{SK}_j$ to $\mathbf{K}_{j,d}$. Therefore, the aggregation can be successfully proceeded. The detailed procedures are similar to Alg. 1. Since the new user's ID is unknown to the residential users, u_d cannot link the keys with the key owners at all.

IV. Security analysis

In this section, we discuss the security properties of our proposed PARK scheme. We focus on the aforementioned security requirements in section II.

• *Privacy Preservation*: First, the outside adversaries cannot passively eavesdrop on the transmitted data between the residential user and the aggregator. The cryptography of data encryption is difficult with the assumption that discrete logarithm problem (DLP) is hard for group $\mathbb{G}_T$. According to Pollard's lambda method, it requires $O(\sqrt{\mathsf{max}(\mathbf{K}_{t,i})})$ time complexity to obtain the power. It is computational infeasible to calculate $\hat{m}_{t,i}$ only given $\mathsf{H}_T(t)^{\hat{m}_{t,i}}$, since $\hat{m}_{t,i}$ is a big data. However, for the aggregation data $\sum_{i=1}^{N} m_{t,i}$, it is a small data compared with $\hat{m}_{t,i}$ so that the AG can extract it. Therefore, the data confidentiality is achieved in the PARK.

Second, each individual user's data cannot be directly revealed to the AG. Recall that $\mathbf{K}_{t,i}$ is added to the metering data $m_{t,i}$, and $\mathbf{K}_{t,i} = \mathsf{H}(k_{t,u_i}||s_{t,u_i})$. As we discussed above, any entity without u_i's encryption key cannot decrypt and obtain $m_{t,i}$ due to the DLP. Furthermore, the AG's capability is merely based on the multiplication of all N ciphertexts $\prod_{i=1}^{N} \mathsf{H}_T(t)^{\hat{m}_{t,i}}$. The only revealed information is the sum $\sum_{i=1}^{N} m_{t,i}$ of N users metering data. Therefore, the AG can obtain the statistical data without knowing any exact data of individual user.

Third, if the AG and multiple residential users are compromised by the adversaries, these compromised users could directly report their own data to the AG. However, the others' data are still protected in the ciphertext form. After the decryption, the AG can only extract the sum of the uncompromising users without learning any individual data from the sum. Therefore, the PARK can resist against the collusion attacks.

• *Forward and Backward Secrecy*: The forward and backward secrecy relies on the security of cryptographic Hash functions H, H_1 and H_2. It is computationally infeasible to execute the reverse Hash function. When a user joins the system, it cannot have the previous keys since $k_{t,i} = \mathsf{H}_1(k_{t-1,i})$. It is computationally infeasible to obtain $k_{t-1,i}$ from $k_{t,i}$. When u_i is revoked, the AG learns u_i's key pair $(k_{t,i}, s_i)$ and can calculate u_i's keys after time slot t. The AG still cannot derive the forward keys due to the computational infeasibility of the reverse Hash function. Thus, $k_i, k_{1,i}, \cdots, k_{t-1,i}$ are unknown to the AG, the AG cannot recover the encryption keys $\mathbf{K}_i, \mathbf{K}_{1,i}, \cdots, \mathbf{K}_{t-1,i}$. The forward secrecy can be achieved.

TABLE I
Security properties among different schemes

Properties	PARK	Shi [4]	Chan [14]	UDP [2]
Privacy	√	√	√	√
Key update	√	×	×	×
Revocation	√	×	√	√

On the other hand, if a user u_i is revoked or departs from the system, u_i is put into the revocation list $\mathcal{RL}$ by the TCC. u_i's key is invalid so that u_i cannot do anything after its revocation. If the key is violated by the outside adversaries, they also cannot derive the keys in the past and coming time slots. We only need to discuss the backward secrecy. Actually, it also holds due to the computational infeasibility of the reverse Hash function H_2^{-1}. Since the backward key $s_{t,i} = \mathsf{H}_2(s_{t+1,i})$, the adversaries are not able to guess the next backward key $s_{t+1,i}$ only based on the current backward key $s_{t,i}$. Therefore, the backward secrecy can be also achieved.

Regarding the bad data and malicious user behavior detections, Esmalifalak et al. [7] and Xiao et al. [8] provide solutions on bad data detection and malicious meter inspection. In this paper, we consider how to adjust the key length according to such detections. The proposed adaptive key management scheme enables the TCC to check the users with lower reputation more frequently. This is because the assigned Hash chain's expiry time $T_i = T \times \frac{R_i}{R_{max}}$, where R_i is the reputation for u_i. If a user uploads some false data or has bad impacts on the system, the TCC adjusts its reputation and reduces the length of this user's Hash chains. Thus, the TCC can monitor the suspicious users more frequently. For the users with high reputation, the TCC assigns longer Hash chains and monitors them infrequently. Therefore, the Hash chain allocation costs for such users dramatically decreases.

As illustrated in Table I, the proposed PARK not only preserves user privacy, but also achieves adaptive key management and revocation at the same time.

V. Performance Evaluation

In this section, we evaluate the performance of the proposed PARK scheme, especially on communication overhead in revocation. We compare the PARK with the UDP [2] and a naive revocation scheme.

As shown in Fig. 5, the revocation overhead is basically the communication costs that the TCC re-assigns the AG's decryption keys. Only one time communication is required in the PARK. Since the TCC assigns the AG with the revoked user u_i's remaining Hash chains $(k_{t,i}, s_i)$. In time slot t, the AG calculates u_i's inherent encryption key $\mathbf{K}_{t,i}$ as shown in Fig. 3, and updates the its own decryption key

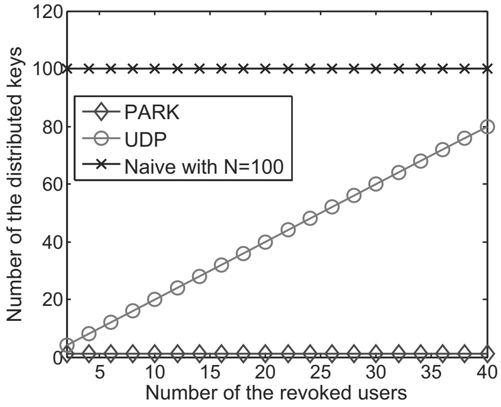

Fig. 5. Communication overhead during revocation

as $\mathbf{K}_{t,AG} = \mathbf{K}_{t,AG} - \mathbf{K}_{t,i}$. The UDP relies on the pre-established ring structure which enables the neighboring users to negotiate the shared keys. However, if a user is revoked, the connecting users are both required to re-establish the group keys. Furthermore, the communication overhead is lineal to the number of the revoked users. Regarding the naive scheme, if any user is revoked, all the users' encryption keys should be re-distributed. The total communication overhead is the total number of the existing users. When the amount of the users is extremely large, for example, 1000, or 10000, the revocation overhead is too large for smart grid. Thus, the PARK has a more efficient key management compared with other schemes.

VI. Related Work

Privacy-preserving aggregation schemes are widely studied in recent years. Shi et al. [4] propose a privacy-preserving aggregation of time series data based on data slicing and mixing to enable the aggregator to decrypt the sum of all users' values without learning anything else. Lu et al. [1] utilize the increasing sequence to merge the user's multi-dimensional data and significantly reduce the communication and computation overhead for smart grid aggregation. Shi et al. [15] propose a privacy-preserving aggregation scheme to support a wide range of statistical additive and non-additive aggregation functions and resist the collusion attacks. Chan et al. [14] upgrade the sophisticated private aggregation scheme with fault tolerance. The trusted authority distributes to the aggregator N capabilities corresponding to the N low level users. They exploit binary tree to build up user groups and then run the traditional block aggregation scheme so that the proposed scheme is fault tolerant. Some of these works in [14] rely on the low-level user grouping and negotiating the group session keys. The dynamic group management and revocation are significantly challenging, since the existing rekeying mechanisms either require all the group members to update their keys, or improve capability of the aggregator.

Regarding key management and revocation, Li et al. [6] propose an efficient demand response scheme to achieve forward secrecy for smart grid. Liang et al. [2] propose a UDP scheme with privacy preservation of each individual user's data from disclosure to the community gateways. The UDP is based on self-organized group secret key sharing. To the best of our knowledge, our paper is the first work to simultaneously achieve privacy preservation, forward and backward secrecy in aggregation for smart grid.

VII. Conclusions

In this paper, we have proposed a privacy-preserving aggregation scheme (PARK) with the efficient and adaptive key management and revocation for smart grid. The PARK enables the aggregator to extract the statistical information from the aggregated data without learning anything else about the individual user. Furthermore, the encryption key for each user can be automatically updated according to the pre-established bi-directional Hash chains. During the revocation, only the aggregators receive update keys from the control center so that the revocation cost is considerably reduced. The security analysis demonstrates that the PARK can extract the aggregated statistical data and preserve user privacy, while achieving forward and backward secrecy. The performance evaluation shows that the PARK has a more efficient key management compared with other schemes. In our future, we intend to investigate fault-tolerant aggregation to resist the smart meter failure.

References

[1] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[2] X. Liang, X. Li, R. Lu, X. Lin, and X. Shen, "UDP: Usage-based dynamic pricing with privacy preservation for smart grid," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 141–150, 2013.

[3] Krebs on Security. [Online]. Available: http://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/

[4] E. Shi, T. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proc. NDSS*, 2011.

[5] H. Lin and Y. Fang, "Privacy-aware profiling and statistical data extraction for smart sustainable energy systems," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 332–340, 2013.

[6] H. Li, X. Liang, R. Lu, X. Lin, and X. S. Shen, "EDR: An efficient demand response scheme for achieving forward secrecy in smart grid," in *Proc. IEEE GLOBECOM*, 2012, pp. 632–640.

[7] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 160–169, 2013.

[8] Z. Xiao, Y. Xiao, and H. Du, "Exploring malicious meter inspection in neighborhood area smart grids," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 214–226, 2013.

[9] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SICOMP: SIAM Journal on Computing*, vol. 32, 2003.

[10] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. EUROCRYPT*, 2003.

[11] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–771, 1981.

[12] R. Lu, X. Lin, Z. Shi, and X. Shen, "EATH: An efficient aggregate authentication protocol for smart grid communications," in *Proc. IEEE WCNC*, 2013.

[13] J. Pollard, "Monte Carlo methods for index computation $(\mathrm{mod} p)$," *Mathematics of Computation*, vol. 32, no. 143, pp. 918–924, 1978.

[14] T. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," *IACR Cryptology ePrint Archive*, vol. 2011, p. 655, 2011.

[15] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-preserving data aggregation in people-centric urban sensing systems," in *Proc. IEEE INFOCOM*, 2010, pp. 758–766.