This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key Evolution in Smart Grid

Hongwei Li, *Member, IEEE,* Xiaodong Lin, *Senior Member, IEEE,* Haomiao Yang, *Member, IEEE,* Xiaohui Liang, *Student Member, IEEE,* Rongxing Lu, *Member, IEEE,* and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Smart grid has recently emerged as the next generation of power grid due to its distinguished features, such as distributed energy control, robust to load fluctuations, and close user-grid interactions. As a vital component of smart grid, demand response can maintain supply-demand balance and reduce users' electricity bills. Furthermore, it is also critical to preserve user privacy and cyber security in smart grid. In this paper, we propose an efficient privacy-preserving demand response (EPPDR) scheme which employs a homomorphic encryption to achieve privacy-preserving demand aggregation and efficient response. In addition, an adaptive key evolution technique is further investigated to ensure the users' session keys to be forward secure. Security analysis indicates that EPPDR can achieve privacy-preservation of electricity demand, forward secrecy of users' session keys, and evolution of users' private keys. In comparison with an existing scheme which also achieves forward secrecy, EPPDR has better efficiency in terms of computation and communication overheads, and can adaptively control the key evolution to balance the trade-off between the communication efficiency and security level.

Index Terms—Smart grid, Demand response, Privacy-preserving, Forward secrecy, Key evolution.

1 Introduction

Recently, smart grid has emerged as a promising solution to the next generation power grid system [1]. It utilizes information and communications technology to gather and act on information, such as information about the behaviors of suppliers and consumers, in an automated fashion to improve the reliability, efficiency, economics, and sustainability of the generation and distribution of electricity [2]. One appealing feature of smart grid is demand response (DR), which can assist users to use energy efficiently and transfer non-emergent power demand from on-peak time to off-peak time [3]. DR can also bring various benefits to users. For example, users can reduce their electricity expenditure by matching the operation time of different electric appliances in their places to the period with the cheapest price. To enable the above characteristics of DR, it often relies on a control center to implement real-time management of users' electricity demand through the communications between the control center and smart meters installed in users' homes.

As smart grid is closely related to people's daily lives, to resolve the security and privacy concerns in smart grid is crucial [4], [5]. Note that in the smart grid networks, adversaries might eavesdrop the communication between users and control center and identify the users' electricity demand. With this information, they are able to track learn about the users' habits or lifestyles [6]. Moreover, adversaries might compromise the smart meters and further obtain stored secret information such as their session keys and private keys [7]. To preserve user privacy and cyber security, DR should not only provide privacy-preservation of electricity demand, but also mitigate the damage caused by the exposure of secret keys stored on the smart meters.

Among many security and privacy requirements for protection of electricity demand and response messages in smart grid, forward secrecy is extremely important since cryptographic computations, e.g. encryption, signature and authentication, are often carried out on the insecure smart meters [8]. In a scheme with forward secrecy, secret keys are evolved at regular time periods. Exposure of a secret key corresponding to a given time period does not enable an adversary to break the scheme for any prior time period [9]. To improve the security level of smart meters, forward secrecy should be considered.

Despite its importance, forward secrecy has not been well studied in smart grid due to the complexity of smart grid communication. Existing schemes mainly focus on achieving confidentiality and integrity of communication, and mutual authentication among different entities [10], [11]. The first attempt to achieve forward secrecy of users' session keys in

- *H. Li, X. Liang, R. Lu and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo,Waterloo, Ontario, Canada (e-mail: hongwei.uestc@gmail.com; x27liang@bbcr.uwaterloo.ca; rxlu.cn@gmail.com; sshen@uwaterloo.ca).*
- *X. Lin is with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada L1H 7K4 (e-mail: xiaodong.lin@uoit.ca).*
- *H. Li and H. Yang are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: hongwei.uestc@gmail.com; haomyang@uestc.edu.cn).*
- *H. Yang is with the School of Computer Engineering, Kyungil University, Kyungsansi, Kyungpook Province, Korea (e-mail: haomyang@uestc.edu.cn).*

Digital Object Indentifier 10.1109/TPDS.2013.124

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

smart grid was studied in [12]. However, since it adopts RSA public key algorithm and Diffie-Hellman exchange protocol to evolve the session key as an effort to ensure forward secrecy, the computation and communication overheads are heavy, thereby making it impractical. In addition, the private keys of users should be evolved since they could also be compromised [8]. However, frequently evolving session keys and private keys will lead to heavy communication overhead; nevertheless, sparsely evolving session keys and private keys will degrade the security level. Therefore, it is challenging to develop a key evolution algorithm which can achieve both efficiency and security level.

In this paper, we propose an Efficient Privacy-Preserving Demand Response scheme with adaptive key evolution, named EPPDR. This work extends our previous research [6] in improving the preliminary demand response scheme for achieving adaptive key evolution. The security and performance of EPPDR are extensively analyzed. Specifically, the contributions of this paper are twofold.

- Firstly, we propose the novel EPPDR scheme that employs the homomorphic encryption [13] to achieve privacy-preserving demand aggregation and efficient response. The security analysis demonstrates that EPPDR can achieve privacy-preservation of electricity demand, forward secrecy of users' session keys, and evolution of users' private keys.
- Secondly, we compare EPPDR with an existing scheme [12] which also achieves forward secrecy. The comparison results demonstrate that EPPDR is more efficient in terms of computation and communication overheads. In addition, EPPDR can adaptively control the key evolution to balance the trade-off between the communication efficiency and security level.

The remainder of the paper is organized as follows. We present the related works in Section 2. In Section 3, we formalize the network model, the security model and the design goal. In Section 4, we review the homomorphic encryption, the bilinear pairing and identity-based signature. Then, we propose the EPPDR scheme in Section 5, followed by the security analysis and the performance evaluation in Section 6 and Section 7, respectively. Finally, we draw our conclusions in Section 8.

2 Related works

In this section, we review and discuss three techniques: homomorphic encryption, forward secrecy and key evolution which are closely related to the proposed scheme in this paper.

Homomorphic encryption can achieve certain algebraic operations on the plaintext to be performed directly on the ciphertext and has been used in many data aggregation schemes [14], [15]. Theses schemes are very promising and have triggered considerable following research work [10], [16], [11] in smart grid. Li et al. [10] present a distributed incremental data aggregation approach. To protect user privacy, homomorphic encryption is used to secure the data en route. Seo et al. [16] propose a secure and efficient power management mechanism leveraging a homomorphic data aggregation and capability-based power distribution. The proposed mechanism enables to gather the power demands of users securely. Lu et al. [11] propose a privacy-preserving aggregation scheme for secure and efficient smart grid communications. It realizes multi-dimensional data aggregation approach based on the homomorphic Paillier cryptosystem. Those schemes assume that the session keys between home area network (HAN) users and building area network gateway (BG) are unchanged. However, once an adversary $\mathcal{A}$ compromises the session keys, $\mathcal{A}$ can decrypt any previous response message.

Forward secrecy is a property that ensures that the messages of prior time period are confidential even if the current time period's key has been compromised [9]. Kate et al. [17] present an improved forward secrecy scheme for onion routing anonymity networks, Its computation and communication overheads are significantly less compared with the previous schemes. Chen et al. [18] propose an efficient approach to establish security links in wireless sensor networks. The proposed scheme only requires small memory size and can achieve forward secrecy. Forward secrecy can be implemented by key evolution technique which generates the new keys based on the old ones. Liu et al. [19] propose a key evolution technique for sensor networks. The technique can ensure forward secrecy and achieve viable trade-offs between security and resource consumption. Libert et al. [20] propose the key evolution systems in untrusted update environments. The systems implement an efficient generic construction and can be extended a forward-secure public key encryption scheme. Although the importance of forward secrecy and key evolution, how to design a data aggregation scheme with forward secrecy of session keys and the evolution of users' private keys in smart grid is a challenging issue. Fouda et al. [12] proposed a lightweight message authentication scheme achieving forward secrecy. Specifically, in the proposed scheme, the HAN users can achieve mutual authentication with BG. Detailed security analysis shows that the proposed scheme can satisfy confidentiality, data integrity, authenticity and forward secrecy. Since the scheme adopts RSA public key signature algorithm and Diffie-Hellman exchange protocol to evolve the session keys between HAN users and BG, the computation and communication overheads are heavy. In this paper, we will design an efficient demand response scheme with forward secrecy of session keys. Based on the non-interactive property of identity-based cryptography, we will propose an adaptive key evolution technique to evolve the session keys in batch mode. Our approach can achieve lower computation and communication overheads compared with the scheme in [12]. In addition, our approach can also achieve the evolution of users' private keys.

3 MODELS AND DESIGN GOAL

3.1 Network Model

As shown in Fig. 1, network model for smart grid is divided into a number of hierarchical networks comprising control center (CC), building area network (BAN) and home area network (HAN). The CC covers n BANs. For the sake of simplicity, we assume each BAN comprises m HANs. Each

HAN is assigned a smart meter enabling an automated, two-way communication between the CC and the HAN user. Meantime, each BAN is equipped with a gateway (GW).

Communication between a HAN user and the BAN GW (BG) is through relatively inexpensive WiFi technology, i.e., within the WiFi coverage of the BG, each HAN user can directly communicate with the BG. For the HAN user who is beyond the coverage of the BG, communication has to be done in multiple hops. However, since the distance between a BG and the CC is far away, the communication between a BG and the CC is through either wired links or any other links with high bandwidth and low delay.

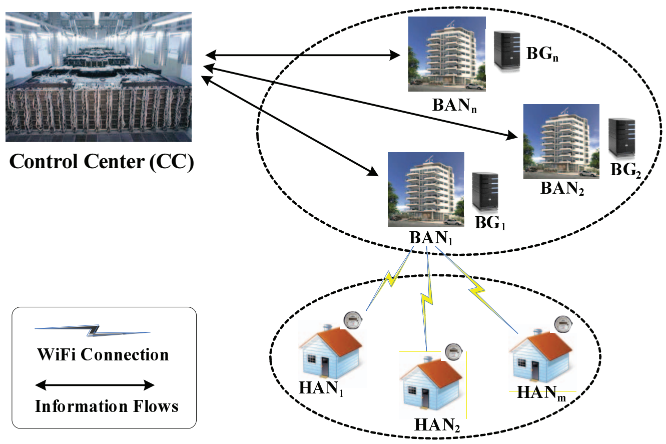


Fig. 1: Network model for smart grid

3.2 Security Model

In our security model, CC and BGs are trusted by all parties in the scheme, and infeasible for any adversary to compromise. In specific, we consider the following security goals needed to be achieved.

- Privacy-preservation of electricity demand: The users' electricity demand should not be disclosed to the unauthorized/untrusted entities. Even if an adversary $\mathcal{A}$ hacks into the database of BGs and CC, it can also not identify the contents of ciphertexts.
- Confidentiality: The electricity response messages should be confidential, i.e., if an adversary $\mathcal{A}$ captures the response messages, it cannot identify the encrypted messages.
- Authenticity and data integrity: BGs and HAN users should be authenticated by CC and BGs, respectively. Meanwhile, if an adversary $\mathcal{A}$ modifies the electricity demand, the malicious operations can be detected.
- Forward secrecy of users' session keys: It should be ensured that the exposure of users' session keys corresponding to a given time period does not enable an adversary to decrypt any prior time period' messages. Specifically, if an adversary $\mathcal{A}$ compromises a HAN user, $\mathcal{A}$ cannot get its previous electricity information. As a result, the forward secrecy can be achieved.
- Evolution of users' private keys: The evolution of users' private keys should be achieved, i.e., if an adversary $\mathcal{A}$ compromises any previous private key of a HAN user, $\mathcal{A}$ cannot use it currently or in the future.

3.3 Design Goal

Under the above models, our design goal is to develop an efficient privacy-preserving demand response scheme with adaptive key evolution. Specifically, the following three desirable objectives will be achieved.

- The proposed scheme should achieve efficient forward secrecy, i.e., evolution of users' session keys should be cost-effective in terms of computation and communication overheads.
- The proposed scheme should achieve adaptively key evolution, i.e, BG can adaptively control the frequency of key evolution by considering the balance of security level and communication efficiency.
- The proposed scheme should achieve the privacy-preservation of electricity demand, the demand's source authentication and data integrity, the confidentiality of the response messages.

4 PRELIMINARIES

In this section, we review the homomorphic encryption, the bilinear pairing and identity-based signature, which will serve as the basis of the proposed EPPDR scheme.

4.1 Homomorphic Encryption

Homomorphic Encryption (HE) allows certain algebraic operations on the plaintext to be performed directly on the ciphertext. HE is usually used for privacy-preserving applications (e.g. data aggregation, e-voting). In this paper, we adopt the Paillier cryptosystem [13]. In the Paillier cryptosystem, the public key is $pk(N, g)$, and the corresponding private key is $sk(\lambda, \mu)$. Let $E(\cdot), m$, and r be the encryption function, a message and a random number, respectively. The ciphertext is

$$c = E(m) = g^m \cdot r^N \bmod N^2 \tag{1}$$

The plaintext is

$$m = D(c) = L(c^{\lambda \bmod N^2}) \cdot \mu \bmod N \tag{2}$$

where the function $L(x) = (x-1)/N$. Then, the additive homomorphic property is as follows:

$$\begin{aligned} E(m_1) \cdot E(m_2) &= (g^{m_1} \cdot r_1^n)(g^{m_2} \cdot r_2^n) \bmod n^2 \\ &= g^{m_1+m_2} \cdot (r_1 r_2)^n \bmod n^2 \\ &= E(m_1 + m_2) \end{aligned} \tag{3}$$

4.2 Bilinear Pairing

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of the same prime order q, and P be a generator of group $\mathbb{G}$. Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are equipped with a pairing, i.e., a non-degenerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $e(aP_1, bQ_1) = e(P_1, Q_1)^{ab} \in \mathbb{G}_T$ for all $a, b \in \mathbb{Z}_q^*$ and any $P_1, Q_1 \in \mathbb{G}$. We refer to [21] for a more comprehensive description of pairing technique, and complexity assumptions.

Definition 1: A bilinear parameter generator $\mathcal{G}en$ is a probabilistic algorithm that takes a security parameter κ as input, and outputs a 5-tuple $(q, P, \mathbb{G}, \mathbb{G}_T, e)$.

4.3 Identity-based signature

Identity-based signature is made of four algorithms that are depicted as follows[22]:

- Setup: Private key generator (PKG) first generates $(q, P, \mathbb{G}, \mathbb{G}_T, e)$ by running $\mathcal{G}en(\kappa)$. Then PKG chooses a random $s \in \mathbb{Z}_q^*$ as the master key and computes the associated public key $P_{pub} = sP$. It also picks two cryptographic hash functions of same domain and range H_1,$H_2 : \{0,1\}^* \rightarrow \dot{\mathbb{G}}$. The system's public parameters are $(q, P, \mathbb{G}, \mathbb{G}_T, e, H_1, H_2)$
- Keygen: Given an user's identity ID, PKG computes Q_{ID}=$H_1(ID)$ and the associated private key d_{ID}=sQ_{ID} that is transmitted to the user.
- Sign: In order to sign message M, the user picks a random number $r \in \mathbb{Z}_q^*$, and computes $U = rP$,$V = d_{ID} + rH_2(ID, M, U)$. The signature on M is the pair $\sigma =< U, V >$.
- Verify: To verify a signature $\sigma =< U, V >$ on a message M for an identity ID, the verifier accepts the signature if $e(P, V)$=$e(P_{pub}, H_1(ID))e(U, H_2(ID, M, U))$ and rejects it otherwise.

5 PROPOSED EPPDR SCHEME

In this section, we propose the EPPDR scheme, which consists of five phases: CC initialization, BAN initialization, demand aggregation, demand processing and response, and key evolution.

5.1 CC Initialization

We assume a control center (CC) will bootstrap the whole system. Specifically, given the security parameter κ, CC first generates the bilinear parameters $(q, P, \mathbb{G}, \mathbb{G}_T, e)$ by running $\mathcal{G}en(\kappa)$, and chooses one secure symmetric encryption algorithm $Enc()$, e.g., AES, and three secure cryptographic hash functions H_1, H_2 and H_3, where $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_3 : \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$. In addition, CC also chooses a random number $\alpha \in \mathbb{Z}_q^*$, and computes $Q = \alpha P$ and $\mathcal{P}_{CC} = \alpha H_1(ID_{CC})$, where ID_{CC} is the identity string of CC. CC also calculates the homomorphic encryption's public key (N, g), and the corresponding private key (λ, μ). Finally, CC publishes the system parameters as

$$\mathsf{pubs} = \left\{ q, P, \mathbb{G}, \mathbb{G}_T, e, Q, H_1, H_2, H_3, N, g, Enc() \right\} \quad (4)$$

and keeps the master key $(\lambda, \mu, \alpha, \mathcal{P}_{CC})$ secretly. When $BG_i(i = 1, 2, \cdots, n)$ registers itself into the system, CC runs the following steps:

- *Step-1:* CC computes the identity-based private key $SK_{BG_i} = \alpha H_1(ID_{CC}||ID_{BG_i})$, where ID_{CC} and ID_{BG_i} are the identity strings of CC and BG_i, respectively.
- *Step-2:* CC grants SK_{BG_i} to BG_i through a secure channel [21].

After receiving SK_{BG_i}, BG_i can non-interactively share a session key K_{BG_i-CC} with CC. BG_i computes $K_{BG_i-CC} = H_3(e(SK_{BG_i}, H_1(ID_{CC})))$, and CC computes $K_{BG_i-CC} = H_3(e(H_1(ID_{CC}||ID_{BG_i}), \mathcal{P}_{CC}))$. The correctness is shown as follows:

$$\begin{aligned} K_{BG_i-CC} &= H_3(e(SK_{BG_i}, H_1(ID_{CC}))) \\ &= H_3(e(\alpha H_1(ID_{CC}||ID_{BG_i}), H_1(ID_{CC}))) \\ &= H_3(e(H_1(ID_{CC}||ID_{BG_i}), H_1(ID_{CC}))^{\alpha}) \\ &= H_3(e(H_1(ID_{CC}||ID_{BG_i}), \alpha H_1(ID_{CC}))) \\ &= H_3(e(H_1(ID_{CC}||ID_{BG_i}), \mathcal{P}_{CC})) \end{aligned} \quad (5)$$

5.2 BAN Initialization

$BG_i(i = 1, 2, \cdots, n)$ chooses a random number $S_{BG_i} \in \mathbb{Z}_q^*$ as its master key and computes

$$Q_{BG_i} = S_{BG_i} P \quad (6)$$

and its private point

$$\mathcal{P}_{BG_i} = S_{BG_i} H_1(ID_{BG_i}) \quad (7)$$

When a HAN user $U_{ij}(j = 1, 2, \cdots, m)$ registers itself into the BG_i, BG_i runs the following steps:

- *Step-1:* BG_i computes the identity-based private key $SK_{U_{ij}} = S_{BG_i} H_1(ID_{BG_i}||ID_{U_{ij}})$, where ID_{BG_i} and $ID_{U_{ij}}$ are the identity strings of BG_i and U_{ij}, respectively.
- *Step-2:* BG_i grants $SK_{U_{ij}}$ to U_{ij} through a secure channel [21].

After receiving $SK_{U_{ij}}$, U_{ij} can non-interactively share a session key $K_{U_{ij}-BG_i}$ with BG_i. U_{ij} computes $K_{U_{ij}-BG_i} = H_3(e(SK_{U_{ij}}, H_1(ID_{BG_i})))$, and BG_i computes $K_{U_{ij}-BG_i} = H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}), \mathcal{P}_{BG_i}))$. Similar to the equation (5), the correctness is shown as follows:

$$\begin{aligned} K_{U_{ij}-BG_i} &= H_3(e(SK_{U_{ij}}, H_1(ID_{BG_i}))) \\ &= H_3(e(S_{BG_i} H_1(ID_{BG_i}||ID_{U_{ij}}), H_1(ID_{BG_i}))) \\ &= H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}), H_1(ID_{BG_i}))^{S_{BG_i}}) \\ &= H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}), S_{BG_i} H_1(ID_{BG_i}))) \\ &= H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}), \mathcal{P}_{BG_i})) \end{aligned} \quad (8)$$

5.3 Demand Aggregation

As shown in Fig. 2, each HAN user $U_{ij} \in BG_i(i = 1, 2, \cdots, n, j = 1, 2, \cdots, m)$ uses the smart meter to collect electricity demand d_{ij}, and performs the following steps:

- *Step-1:* U_{ij} chooses a random number $r_{ij} \in \mathbb{Z}_N^*$ and computes:

$$C_{U_{ij}} = g^{d_{ij}} \cdot r_{ij}^N \bmod N^2 \quad (9)$$

- *Step-2:* U_{ij} uses the private key $SK_{U_{ij}}$ to make an identity-based signature $\sigma_{U_{ij}}$ on M, where $M = C_{U_{ij}}||ID_{BG_i}||ID_{U_{ij}}||TS$, TS is the current timestamp. Firstly, U_{ij} picks a random number $r_{ij} \in \mathbb{Z}_q^*$, and computes $U = r_{ij}P$, $V = SK_{U_{ij}} + r_{ij}H_2(M, U)$. The signature on M is the pair $\sigma_{U_{ij}} =< U, V >$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

U_{ij} $\qquad$ BG_i $\qquad$ CC

$C_{ij} = E(d_{ij})$

$M = C_{U_{ij}} \| ID_{BG_i} \| ID_{U_{ij}} \| TS$

$r_{ij} \in \mathbb{Z}_q^*$

$U = r_{ij} P$

$V = SK_{U_{ij}} + r_{ij} H_2(M, U)$

$\sigma_{U_{ij}} = < U, V >$

$\xrightarrow{M \| \sigma_{U_{ij}}}$

Checks $\sigma_{U_{ij}}$, if it is valid,

$C_{BG_i} = \prod_{j=1}^{m} C_{U_{ij}}$

$M' = C_{BG_i} \| ID_{BG_i} \| ID_{CC} \| TS$

$r_i' \in \mathbb{Z}_q^*$

$U' = r_i' P$

$V' = SK_{BG_i} + r_i' H_2(M', U')$

$\sigma_{BG_i} = < U', V' >$

$\xrightarrow{M' \| \sigma_{BG_i}}$

Fig. 2: Demand aggregation

- *Step-3:* U_{ij} sends the encrypted electricity demand $C_{U_{ij}}||ID_{BG_i}||ID_{U_{ij}}||TS||\sigma_{U_{ij}}$ to the BG_i.

After receiving encrypted electricity demand $C_{U_{ij}}||ID_{BG_i}||ID_{U_{ij}}||TS||\sigma_{U_{ij}}$, BG_i first checks signature $\sigma_{U_{ij}}$ to verify its validity. BG_i accepts the signature if the following equation holds.

$$e(P, V) = e(Q_{BG_i}, H_1(ID_{BG_i}||ID_{U_{ij}}))e(U, H_2(M, U)) \tag{10}$$

where $M = C_{U_{ij}}||ID_{BG_i}||ID_{U_{ij}}||TS$. The correctness is shown as follows:

$$\begin{aligned} e(P, V) &= e(P, SK_{U_{ij}} + r_{ij}H_2(M, U)) \\ &= e(P, SK_{U_{ij}})e(P, r_{ij}H_2(M, U)) \\ &= e(P, S_{BG_i}H_1(ID_{BG_i}||ID_{U_{ij}}))e(r_{ij}P, H_2(M, U)) \\ &= e(S_{BG_i}P, H_1(ID_{BG_i}||ID_{U_{ij}}))e(U, H_2(M, U)) \\ &= e(Q_{BG_i}, H_1(ID_{BG_i}||ID_{U_{ij}}))e(U, H_2(M, U)) \end{aligned} \tag{11}$$

After the validity checking, BG_i performs the following steps based on multiple received $C_{U_{ij}}(j = 1, \cdots, m)$:

- *Step-1:* BG_i computes the aggregated demand on $C_{U_{i1}}, C_{U_{i2}}, \cdots, C_{U_{im}}$ as

$$C_{BG_i} = \prod_{j=1}^{m} C_{U_{ij}} \tag{12}$$

- *Step-2:* BG_i uses the private key SK_{BG_i} to make an identity-based signature σ_{BG_i} on M', where $M' = C_{BG_i}||ID_{BG_i}||ID_{CC}||TS$, TS is the current timestamp. Firstly, BG_i picks a random number $r_i' \in \mathbb{Z}_q^*$, and computes $U' = r_i'P$ and $V' = SK_{BG_i} + r_i'H_2(M', U')$. The signature on M' is the pair $\sigma_{BG_i} =< U', V' >$.
- *Step-3:* BG_i sends the aggregated result $C_{BG_i}||ID_{BG_i}||ID_{CC}||TS||\sigma_{BG_i}$ to the CC.

5.4 Demand Processing and Response

As shown in Fig. 3, upon receiving n encrypted electricity demand $C_{BG_i}||ID_{BG_i}||ID_{CC}||TS||\sigma_{BG_i}(i = 1, 2, \cdots, n)$, CC first checks signature σ_{BG_i} to verify its validity. CC accepts the signature if the following equation holds.

$$e(P, V') = e(Q, H_1(ID_{CC}||ID_{BG_i}))e(U', H_2(M', U')) \tag{13}$$

Where $M' = C_{BG_i}||ID_{BG_i}||ID_{CC}||TS$. The correctness is shown as follows:

$$\begin{aligned} e(P, V') &= e(P, SK_{BG_i} + r_i'H_2(M', U')) \\ &= e(P, SK_{BG_i})e(P, r_i'H_2(M', U')) \\ &= e(P, \alpha H_1(ID_{CC}||ID_{BG_i}))e(r_i'P, H_2(M', U')) \\ &= e(\alpha P, H_1(ID_{CC}||ID_{BG_i}))e(U', H_2(M', U')) \\ &= e(Q, H_1(ID_{CC}||ID_{BG_i}))e(U', H_2(M', U')) \end{aligned} \tag{14}$$

After the validity checking, CC performs the following steps to read the aggregated demand C, where C is implicitly formed by

$$\begin{aligned} C &= \prod_{i=1}^{n} C_{BG_i} \\ &= \prod_{i=1}^{n}(\prod_{j=1}^{m} C_{U_{ij}}) \\ &= \prod_{i=1}^{n}(\prod_{j=1}^{m} g^{d_{ij}} \cdot r_{ij}^N \bmod N^2) \\ &= \prod_{i=1}^{n}(g^{\sum_{j=1}^{m} d_{ij}} \cdot (\prod_{j=1}^{m} r_{ij})^N \bmod N^2) \\ &= g^{\sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})} \cdot (\prod_{i=1}^{n}(\prod_{j=1}^{m} r_{ij}))^N \bmod N^2 \end{aligned} \tag{15}$$

- *Step-1:* By taking $M = \sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})$ and $R = \prod_{i=1}^{n}(\prod_{j=1}^{m} r_{ij})$, the report $C = g^M \cdot R^N mod\, N^2$ is still a ciphertext of Paillier Cryptosystem. Therefore, CC can use the master key (λ, μ) to recover M as $M = \sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})$. Similarly, CC can recover BG_i's aggregated electricity demand as $\sum_{j=1}^{m} d_{ij}$.
- *Step-2:* After analyzing the real-time electricity demand $\sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})$ and $\sum_{j=1}^{m} d_{ij}(i = 1, 2, \cdots, n)$, CC generates the response message $S_i(0 < S_i \leq 1)$ for $BG_i(i = 1, 2, \cdots, n)$, respectively [3], where S_i is a scale coefficient. For example, the electricity demand from BG_i is $\sum_{j=1}^{m} d_{ij}$=20000 kw/h, however, CC would like to provide 16000 kw/h considering the electricity generation and the total electricity demand $\sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})$. Then CC sets $S_i = 0.8$. If electricity consumption from BG_i is more than 16000 kw/h, the electricity tariff will be higher than before.
- *Step-3:* CC sends $C_i||ID_{CC}||ID_{BG_i}||TS$ to $BG_i(i = 1, 2, \cdots, n)$, respectively, where $C_i = Enc_{K_{BG_i-CC}}(S_i||ID_{CC}||ID_{BG_i}||TS)$ and TS is the current timestamp.
- *Step-4:* Upon receiving $C_i||ID_{CC}||ID_{BG_i}||TS$, BG_i decrypts C_i to get S_i. Then BG_i forwards

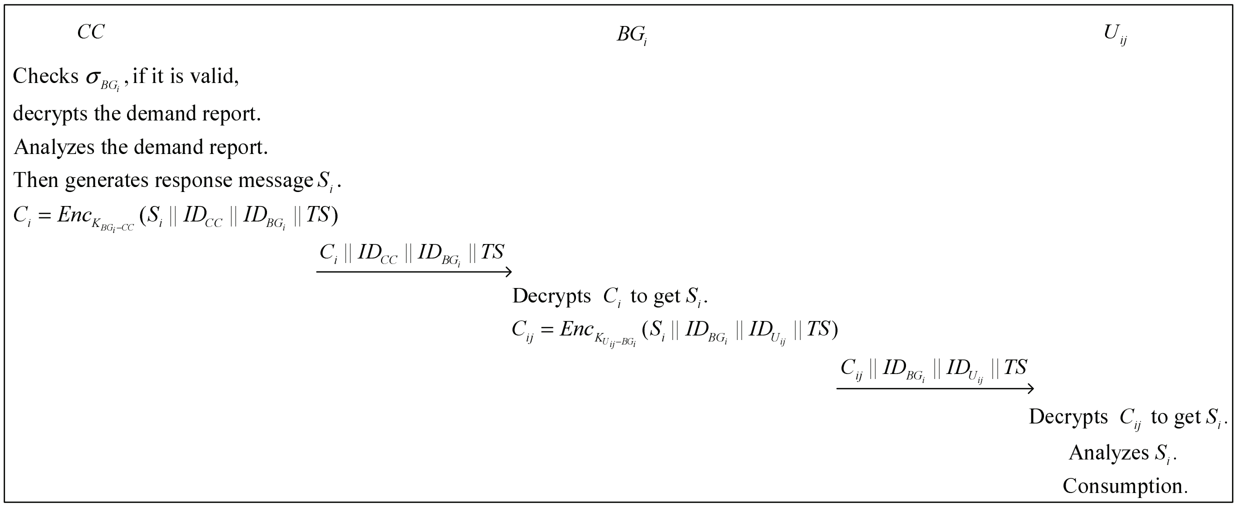


Fig. 3: Demand processing and response

$C_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$ to the HAN user U_{ij}, where $C_{ij} = Enc_{K_{U_{ij}-BG_i}}(S_i||ID_{BG_i}||ID_{U_{ij}}||TS)$ and TS is the current timestamp.

- *Step-5:* After receiving $C_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$, HAN user U_{ij} decrypts C_{ij} to get S_i. Then U_{ij} analyzes S_i and determines to shift power use from peak times to non-peak times for lower electricity bills [3].

5.5 Key evolution

Firstly, we extend the identity string ID to $ID||d$, where d represents the expiry date. Note that the extension does not influence the previous EPPDR scheme. For the HAN user U_{ij}, the identity string $ID_{U_{ij}}||d$ is only valid before the specified expiry date d. After d, the corresponding private key $SK_{U_{ij}||d}$ is automatically revoked if a new private key is not generated by BG_i. If the unit of d is chosen as one day [20], the lifetime of each private key is also that. As shown in Fig. 4, the proposed key evolution mechanism is comprised of many rounds. At the end of $Round_i(i = 1, 2, \cdots)$, $Round_{i+1}$'s keys will be generated by key evolution algorithm as described in Fig. 5. The time interval of each round can be calculated by the number of its keys. For instance, if the number of $Round_{i+1}$'s keys is N, then the time interval of $Round_{i+1}$ is N days since the lifetime of each key is one day. Next, we discuss the key evolution algorithm in detail. Specifically, at the end of $Round_i(i = 1, 2, \cdots)$, a HAN user U_{ij} can generate $Round_{i+1}$'s keys by performing the following steps:

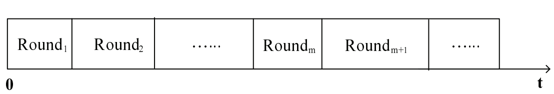


Fig. 4: Proposed key evolution mechanism

- *Step-1:* According to the security and efficiency requirement, U_{ij} chooses an integer l_{ij} as the number of evolving private keys. For example, when the security level is more important than efficiency, U_{ij} can choose a small integer for l_{ij}, e.g., $l_{ij} = 5$. On the other hand, when the efficiency is more important than security level, U_{ij} can choose a bigger integer for l_{ij}, e.g., $l_{ij} = 30$. we will further show the results in Section 7.
- *Step-2:* U_{ij} sends $C_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$ to BG_i, where $C_{ij} = Enc_{K_{U_{ij}-BG_i}}(l_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS)$ and TS is the current timestamp.
- *Step-3:* After receiving $C_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$, BG_i firstly decrypts C_{ij} to get l_{ij} with the session key $K_{U_{ij}-BG_i}$. Then BG_i checks whether $l_{ij} < m_{ij}$, where m_{ij} represents the maximum number of private keys which BG_i can assign to U_{ij}. if it does hold, BG_i does nothing. Otherwise, BG_i lets $l_{ij} = m_{ij}$. And then BG_i generates l_{ij} private keys as

$$\begin{aligned} SK_{U_{ij}||(d+1)} &= S_{BG_i} H_1(ID_{BG_i}||ID_{U_{ij}}||(d+1)) \\ SK_{U_{ij}||(d+2)} &= S_{BG_i} H_1(ID_{BG_i}||ID_{U_{ij}}||(d+2)) \\ &\cdots \\ SK_{U_{ij}||(d+l_{ij})} &= S_{BG_i} H_1(ID_{BG_i}||ID_{U_{ij}}||(d+l_{ij})) \end{aligned} \tag{16}$$

The set of private keys is as follows:

$$\omega = \{SK_{U_{ij}||(d+1)}, SK_{U_{ij}||(d+2)}, \cdots, SK_{U_{ij}||(d+l_{ij})}\} \tag{17}$$

- *Step-4:* BG_i sends $C'_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$ to U_{ij}, where $C'_{ij} = Enc_{K_{U_{ij}-BG_i}}(\omega||ID_{BG_i}||ID_{U_{ij}}||TS)$ and TS is the current timestamp.
- *Step-5:* After receiving $C'_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$ to U_{ij}, U_{ij} firstly recovers ω with the session key $K_{U_{ij}-BG_i}$. Then U_{ij} deletes previous session and processing information and stores ω secretly. Thus, the secure private key evolution is achieved.
- *Step-6:* On the date $d + a(a = 1, \cdots, l_{ij})$, U_{ij} firstly deletes the previous private key. Then U_{ij} can non-interactively share a new session key $K'_{U_{ij}-BG_i}$ with BG_i. U_{ij} computes

$$K'_{U_{ij}-BG_i} = H_3(e(SK_{U_{ij}||(d+a)}, H_1(ID_{BG_i}))) \tag{18}$$

And BG_i computes

$$K'_{U_{ij}-BG_i} = H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}||(d+a)), \mathcal{P}_{BG_i})) \tag{19}$$

The correctness is shown as follows:

$$\begin{aligned} K'_{U_{ij}-BG_i} &= H_3(e(SK_{U_{ij}||(d+a)}, H_1(ID_{BG_i}))) \\ &= H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}||(d+a)), H_1(ID_{BG_i}))^{S_{BG_i}}) \\ &= H_3(e(H_1(ID_{BG_i}||ID_{U_{ij}}||(d+a)), \mathcal{P}_{BG_i})) \end{aligned} \tag{20}$$

After that, U_{ij} deletes the previous session key $K_{U_{ij}-BG_i}$. Thus, even if an adversary $\mathcal{A}$ compromises U_{ij}, it cannot get any previous session key. Therefore, the forward secrecy of session key is achieved.

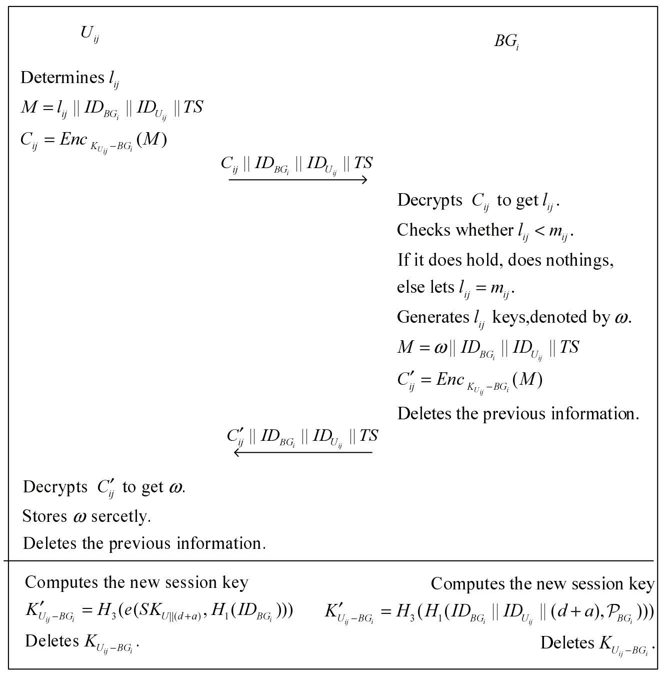


Fig. 5: Key evolution algorithm

6 SECURITY ANALYSIS

In this section, we analyze the security properties of the proposed EPPDR scheme. Especially, following the security model discussed earlier, we are most concerned with how EPPDR can achieve the privacy-preservation of electricity demand, the source authentication and data integrity of electricity demand, and the confidentiality of response messages, the forward secrecy of users' session keys, and the evolution of users' private keys,

6.1 EPPDR provides the privacy-preservation of electricity demand

In the proposed EPPDR scheme, since HAN user's electricity demand is a homomorphic encryption cipthertext [13], an adversary $\mathcal{A}$ cannot identify the corresponding electricity demand even though $\mathcal{A}$ eavesdrops the ciphertext. Moreover, since BG_i only aggregates and does not decrypt the electricity demands, $\mathcal{A}$ cannot get the electricity demand even if $\mathcal{A}$ compromises the BG_i's database. Finally, C-C recovers the aggregated demands $\sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})$ and $\sum_{j=1}^{m} d_{ij} (i = 1, 2, \cdots, n)$. However, since $\sum_{i=1}^{n}(\sum_{j=1}^{m} d_{ij})$ and $\sum_{j=1}^{m} d_{ij} (i = 1, 2, \cdots, n)$ are all aggregated results, even if $\mathcal{A}$ intrudes the CC's database, $\mathcal{A}$ still cannot get the each HAN user's electricity demand. Therefore, the proposed EPPDR scheme preserves the electricity demand privacy.

6.2 EPPDR provides the source authentication and data integrity of electricity demand, and the confidentiality of response messages

In the proposed EPPDR scheme, as mentioned in Section 5.3, each HAN user's electricity demand and the aggregated demand are signed by the identity-based signature [22]. Since the signature is provably secure in the random oracle model [22], the source authentication and data integrity of electricity demand can be guaranteed. In addition, note that the HAN user U_{ij}'s private key $SK_{U_{ij}} = S_{BG_i} H_1(ID_{BG_i}||ID_{U_{ij}})$, namely $SK_{U_{ij}}$ is bound to BG_i. Thus, BG_i can identify whether a HAN user belongs to its administration domain. As mentioned in equation (10), a HAN user cannot pass the signature verification if it is not in BG_i's administration domain. On the other hand, in the proposed EPPDR scheme, when CC sends the response messages to $BG_i (i = 1, 2, \cdots, n)$, CC encrypts them as $C_i = Enc_{K_{BG_i-CC}}(S_i||ID_{CC}||ID_{BG_i}||TS)$. Then, BG_i forwards the response messages to $U_{ij} (j = 1, 2, \cdots, m)$ in the form of $C_{ij} = Enc_{K_{U_{ij}-BG_i}}(S_i||ID_{BG_i}||ID_{U_{ij}}||TS)$. Since C_i and C_{ij} are encrypted by AES [23], the confidentiality of response messages can be guaranteed.

6.3 EPPDR provides the forward secrecy of users' session keys

In the proposed EPPDR scheme, the confidentiality of communication between HAN user and BG is achieved based on the secure session key. In the key evolution phase, after computing the new session key $K'_{U_{ij}-BG_i}$, U_{ij} deletes the previous session key $K_{U_{ij}-BG_i}$. As a result, even if $\mathcal{A}$ compromises the HAN users U_{ij}, $\mathcal{A}$ cannot get any previous session key. Moreover, $\mathcal{A}$ cannot compute any previous session key as mentioned in equation (18) since the corresponding private key has been deleted. Therefore, the forward secrecy of users' session keys is achieved in the proposed EPPDR scheme.

Furthermore, we analyze the information leakage in both scheme without forward secrecy (denoted by NFS) and that with forward secrecy (denoted by FS). The information leakage happens when the encrypted messages are decrypted by an unauthorized adversary [24]. The number of information leakage can be calculated by the time interval in which the encrypted messages cannot be guaranteed to be confidential. As shown in Fig. 6, t_0 and t_1 represent the times when U_{ij} is compromised and when the system detects the attack and revokes U_{ij}, respectively. T_c represents the time interval between $t = 0$ and $t = t_0$. X_i and T_d represent the time interval of $Round_i$ and the time delay of system detection, respectively. In probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with an average rate and independently of the time since the last event [25]. Note that X_i and T_d can be seen as the discrete events in the visual spaces "the time interval of $Round_i$" and "the time delay of system detection", respectively. And "the time interval of $Round_i$" and "the time delay of system detection" both occur with an average rate and independently of the time since the last event. Therefore, we can model that X_i and T_d follow the Poisson distribution with intensity λ_x and λ_d, respectively, then $P(X_i = k) = \frac{e^{-\lambda_x}\lambda_x^k}{k!}$ and $P(T_d = k) = \frac{e^{-\lambda_d}\lambda_d^k}{k!}$. Next, we discuss the number of information leakage in both NFS and FS. For NFS, all messages encrypted before $t = t_1$ can be decrypted since the session key has not been evolved since $t = 0$. Thus, the number of information leakage for NFS is $T_c + \lambda_d$. In comparison, for FS, since a user U_{ij} only stores the current round's keys and has deleted the previous rounds' keys, the previous rounds' messages cannot be decrypted even if U_{ij} is compromised. Therefore, in the worst case of FS when t_0 is at the end of $Round_i(i = 1, 2, \cdots)$, an adversary can decrypt the messages encrypted in both $Round_i$ and the following λ_d time interval. Thus, the number of information leakage is $\lambda_x + \lambda_d(t_0 \geq X_1)$. And in the best case of FS when t_0 is at the beginning of a round, an adversary only can decrypt the messages encrypted in the following λ_d time interval. Thus, the number of information leakage is λ_d.

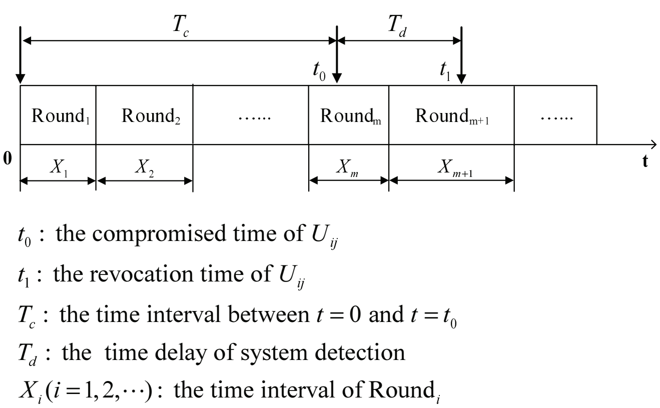
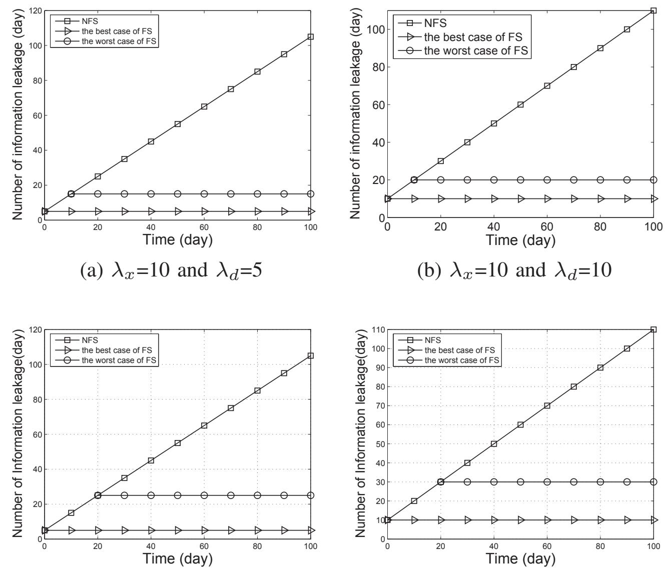


Fig. 6: Information leakage model

The comparison of the number of information leakage between FS and NFS is shown in Fig. 7. It can be seen that FS significantly reduces the number of information leakage compared with NFS. In addition, it is observed that when λ_d is constant, the number of information leakage increases with the increased λ_x. For example, when $\lambda_d = 5$ and $\lambda_x = 10$, as shown in Fig. 7a, the number of information leakage is in the interval [5,15]. However, when $\lambda_d = 5$ and $\lambda_x = 20$, as shown in Fig. 7c, the number of information leakage is in the interval [5,25]. Note that λ_x represents the intensity of X_i, where X_i is the time interval of $Round_i$. As mentioned in Section 5.5, X_i can be controlled by m_{ij} since m_{ij} is the upper bound of X_i. When m_{ij} is set small enough, X_i is also. Further λ_x is also since λ_x is the average value of $X_i(i = 1, 2, \cdots)$ [25]. Thus, the small λ_x enables the number



(a) λ_x=10 and λ_d=5 (b) λ_x=10 and λ_d=10

(c) λ_x=20 and λ_d=5 (d) λ_x=20 and λ_d=10

Fig. 7: Comparison of the number of information leakage

of information leakage to be low. In comparison, when m_{ij} is set bigger, X_i and λ_x can be relatively increased, thereby make the number of information leakage high. Therefore, in our key evolution technique, when m_{ij} is set relatively bigger when the evolution of session keys and private keys is sparse, the number of information leakage is higher. As a result, the security level is degraded. On the other hand, when m_{ij} is set smaller when the evolution of session keys and private keys is frequent, the number of information leakage can be decreased and the security level can be upgraded. However, the smaller m_{ij} will lead to heavy communication overhead. We will discuss that in Section 7.2.

6.4 EPPDR provides the evolution of users' private keys

In the key evolution phase, U_{ij} firstly sends l_{ij} to BG_i by the symmetric encryption algorithm. Then according to l_{ij} provided by U_{ij}, BG_i generates the set of private keys ω and further sends ω to U_{ij} by the symmetric encryption algorithm. Thus, even if an adversary $\mathcal{A}$ eavesdrops the communication between BG_i and U_{ij}, it cannot get any information about ω. On the other hand, even if an adversary $\mathcal{A}$ compromises any previous private key, it cannot deduce current or future private keys since the discrete logarithm problem ensures the private keys' security [21]. Therefore, the secure private key evolution of user is achieved in the proposed EPPDR scheme.

Finally, we present the comparison results of security level in Table 1. It can be seen that the scheme [10] only achieves the confidentiality, the scheme [16] achieves confidentiality and data integrity, the scheme [11] achieves confidentiality, data integrity and authenticity, and the scheme [12] achieves confidentiality, data integrity, authenticity, forward secrecy. The proposed EPPDR scheme achieves additional private key evolution compared with the scheme [12].

TABLE 1: Comparison of Security Level

	[10]	[16]	[11]	[12]	EPPDR
Confidentiality	√	√	√	√	√
Authenticity			√	√	√
Data integrity		√	√	√	√
Forward secrecy				√	√
Private key evolution					√

7 PERFORMANCE EVALUATION

In this section, we evaluate the computation and communication overheads of the session key evolution between user U_{ij} and BG_i in both EPPDR and the scheme [12].

7.1 Computation Overhead

Compared to exponentiation operations in $\mathbb{G}$, pairing operations and RSA encryption/decryption, the computation overhead of AES encryption/decryption and hash operations are negligible[26]. In EPPDR, as described in Section 5.5, BG_i computes l_{ij} new private keys and l_{ij} new session keys with l_{ij} multiplication operations in $\mathbb{G}$ and l_{ij} pairing operations, respectively. U_{ij} computes the new session keys with l_{ij} pairing operations. Note that the above computation overhead is constant even if m_{ij} varies. Therefore, in this section, we do not consider the variants of EPPDR with different m_{ij}. We will discuss the balance between the communication efficiency and security level in the Section 7.2. Denote the computation overhead of a multiplication operation in $\mathbb{G}$ and a pairing operation by C_m and C_p, respectively. Thus, the total computation overhead is $l_{ij} * C_m + 2l_{ij} * C_p$. In comparison, in the scheme [12], for evolving each session key, it requires 1 RSA encryption for HAN user i to generate the request packet. After receiving the ciphertext from HAN user i, BG j decrypts the request packet including 1 RSA decryption and computes the new session key with 1 exponentiation operation in $\mathbb{Z}_q^*$. In addition, BG j sends an encrypted response message including 1 RSA encryption. Then HAN user i decrypts the response message with 1 RSA decryption and computes the new session key including 1 exponentiation operation in $\mathbb{Z}_q^*$. Denote the computation overhead of an exponentiation operation in $\mathbb{Z}_q^*$, a RSA encryption and a RSA decryption by C_e, RSA_e and RSA_d, respectively. Thus, the total computation overhead for evolving a session key is $2*(C_e+RSA_e+RSA_d)$. Therefore, the total computation overhead for evolving l_{ij} session keys is $2 * l_{ij} * (C_e + RSA_e + RSA_d)$.

Experiments were conducted on a 3.0GHz-processor, 1GB memory computing machine with MIRACL [27] and Pbc [28] libraries to study the execution time. For $\mathbb{G}$ over the FST curve, a single multiplication operation costs 1.1 ms and the corresponding paring operation costs 3.1 ms. Meantime, a 1024-RSA decryption and a 1024-RSA encryption cost 3.88ms and 0.02ms, respectively. An exponentiation operation in $\mathbb{Z}_q^*(|q| = 1024)$ costs 0.64ms. The comparison of computation overhead is shown in Fig.8. We can see that EPPDR achieves lower execution times compared to the scheme [12].

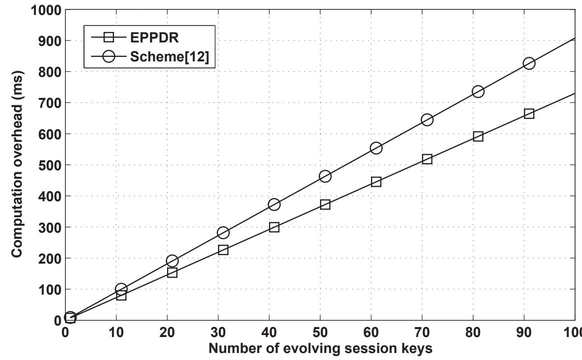


Fig. 8: Comparison of computation overhead

7.2 Communication Overhead

In the scheme [12], for evolving each session key, HAN user i first sends a RSA ciphtertext to BG j in the form of $\{i||j||g^a\}_{\{encr\}Pub_{BAN\_GW_j}}$. Then BAN GW j responds a RSA ciphertext to HAN user i in the form of $\{i||j||g^a||g^b\}_{\{encr\}Pub_{HAN\_GW_i}}$. Finally, HAN user i sends an AES ciphertext to BG j in the form of $\{M_i||T_i||HMAC_{K_i}\}_{\{encr\}K_i}$. The overall communication overhead consists of two RSA ciphertexts and one AES ciphertext. Thus, the overall communication overhead is $2 * 1024 + 256 = 2304$ bits if we choose 1024-bit RSA and 256-bit AES. Therefore, the total communication overhead for evolving l_{ij} session keys is $2304*l_{ij}$ bits. In comparison, in EPPDR, for evolving l_{ij} session keys, U_{ij} firstly sends message to BG_i. The message is in the form of $C_{U_{ij}}||ID_{BG_i}||ID_{U_{ij}}||TS$, where $C_{U_{ij}} = Enc_{K_{U_{ij}-BG_i}}(l_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS)$. If we choose AES ciphertext with 256-bit and set $|BG_i| + |U_{ij}| + |TS|$ as 80-bit length, the whole message size is $256 + 80 = 336$ bits. Then, BG_i checks if $l_{ij} < m_{ij}$, if it does hold, BG_i does nothing else lets $l_{ij} = m_{ij}$. And then BG_i responds a message in the form of $C'_{ij}||ID_{BG_i}||ID_{U_{ij}}||TS$ to U_{ij}, where $C'_{ij} = Enc_{K_{U_{ij}-BG_i}}(\omega||ID_{BG_i}||ID_{U_{ij}}||TS)$, $\omega = \{SK_{U_{ij}||(d+1)}, SK_{U_{ij}||(d+2)}, \cdots, SK_{U_{ij}||(d+l_{ij})}\}$. If we choose $\mathbb{G}$ with 160-bit order and use point compression technique [29], the element in $\mathbb{G}$ is roughly 161-bit. If we choose AES ciphertext with 256-bit, C'_{ij} should be generated based on the 256-bit block encryption [23]. Thus, the size of C'_{ij} is $\lceil(161 * Min(l_{ij}, m_{ij}) + 80)/256\rceil * 256$ bits, where $Min(l_{ij}, m_{ij})$ is the minimum between l_{ij} and m_{ij}. Note that for evolving l_{ij} session keys, the whole evolving process will be run $\lceil l_{ij}/m_{ij}\rceil$ times. So the total communication overhead is $\lceil l_{ij}/m_{ij}\rceil(\lceil(161*Min(l_{ij}, m_{ij})+80)/256\rceil*256+80+336)$ bits.

Fig. 9 shows the communication overhead for different number of evolving session keys. When the number of evolving session keys is small, the communication overhead is low in both EPPDR and the scheme [12]. Then the communication overhead increases with the increased number of keys. However, it should be noted that the increase is much faster in the case of the scheme [12]. EPPDR significantly reduces the communication overhead for the session key evolution. On the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

other hand, among three variants of EPPDR with different m_{ij}, the communication overhead decreases with the increased m_{ij} since the bigger m_{ij} reduces the frequency of key evolution. However, as mentioned in Section 6.3, when m_{ij} is increased, the number of information leakage is also increased and the security level is degraded. In our proposed EPPDR scheme, as mentioned in Section 5.5, BG_i can adaptively adjust m_{ij} to balance the trade-off between the communication efficiency and security level.

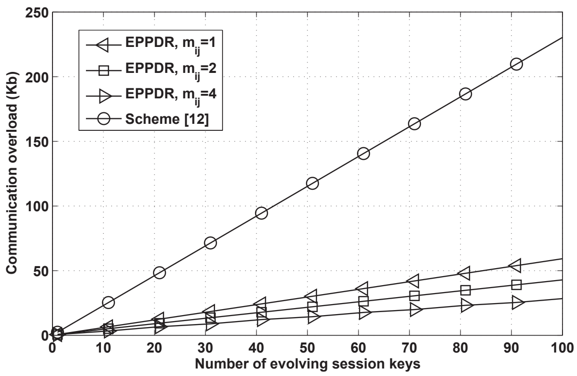


Fig. 9: Comparison of communication overhead

8 CONCLUSIONS

In this paper, we have proposed an efficient privacy-preserving demand response (EPPDR) scheme with adaptive key evolution. It realizes secure and efficient electricity demand aggregation and response based on the homomorphic encryption and the key evolution techniques. Security analysis has demonstrated that EPPDR can achieve privacy-preservation of electricity demand, forward secrecy of users' session keys, evolution of users' private keys. Performance evaluation further demonstrates its efficiency in terms of computation and communication overhead. In addition, EPPDR can adaptively control the key evolution to balance the trade-off between the communication efficiency and security level

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grants 61103207, 61272525, U1233108, 61073106, and 61003232, the Fundamental Research Funds for the Central Universities under Grant ZYGX2011J059, the 2011 Korea-China Young Scientist Exchange Program, and NSERC, Canada. In addition, the first author is very grateful to Mi Wen for helpful discussions and comments.

REFERENCES

[1] H. Liang, B. Choi, W. Zhuang, and X. Shen, "Towards optimal energy store-carry-and-deliver for phevs via v2g system," in *IEEE INFOCOM 2012*, 2012, pp. 1674–1682.

[2] H. Liang, B. Choi, A. Abdrabou, W. Zhuang, and X. Shen, "Decentralized economic dispatch in microgrids via heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1061–1074, 2012.

[3] F. Rahimi and A. Ipakchi, "Demand response as a market resource under the smart grid paradigm," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 82–88, 2010.

[4] X. Li, X. Liang, R. Lu, H. Zhu, X. Lin, and X. Shen, "Securing smart grid: Cyber attacks, countermeasures and challenges," *IEEE Communications Magazine*, vol. 58, no. 8, pp. 38–45, 2012.

[5] 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.

[6] H. Li, X. Liang, R. Lu, X. Lin, and X. Shen, "Edr: An efficient demand response scheme for achieving forward secrecy in smart grid," in *2012 IEEE Global Telecommunications Conference(GLOBECOM 2012)*, Dec. 2012.

[7] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle tree based authentication scheme for smart grid," *IEEE Systems Journal*, to appear.

[8] J. Xia and Y. Wang, "Secure key distribution for the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1437–1443, 2012.

[9] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," *Advances in Eurocrypt 2003*, pp. 646–646, 2003.

[10] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010, pp. 327–332.

[11] 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.

[12] M. Fouda, Z. Fadlullah, N. Kato, R. Lu, and X. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.

[13] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999, pp. 223–238.

[14] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems," in *INFOCOM*, 2010, pp. 1–9.

[15] X. Lin, R. Lu, and X. Shen, "Mdpa: multidimensional privacy-preserving aggregation scheme for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 6, pp. 843–856, 2010.

[16] D. Seo, H. Lee, and A. Perrig, "Secure and efficient capability-based power management in the smart grid," in *IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW)*, 2011, pp. 119–126.

[17] A. Kate, G. Zaverucha, and I. Goldberg, "Pairing-based onion routing with improved forward secrecy," *ACM Transactions on Information and System Security*, vol. 13, no. 4, p. 29, 2010.

[18] C. Chen, S. Huang, and I. Lin, "Providing perfect forward secrecy for location-aware wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 241, 2012.

[19] Z. Liu, J. Ma, Q. Pei, L. Pang, and Y. Park, "Key infection, secrecy transfer, and key evolution for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, pp. 2643–2653, 2010.

[20] B. Libert, J. Quisquater, and M. Yung, "Key evolution systems in untrusted update environments," *ACM Transactions on Information and System Security*, vol. 13, no. 4, p. 37, 2010.

[21] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO*, 2001, pp. 213–229.

[22] B. Libert and J. Quisquater, "The exact security of an identity based signature and its applications," *Preprint available at http://eprint. iacr. org/2004/102*, 2004.

[23] D. Stinson, *Cryptography: theory and practice*. CRC press, 2006.

[24] J. Kelsey, "Compression and information leakage of plaintext," in *Fast Software Encryption 2002*. Springer, 2002, pp. 95–102.

[25] S. Katti and A. Rao, "Handbook of the poisson distribution," *Technometrics*, vol. 10, no. 2, pp. 412–412, 1968.

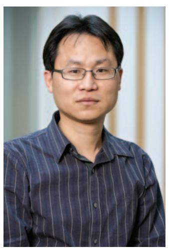
[26] W. Dai, "Crypto++ 5.6.0 benchmarks," *http://www.cryptopp.com/benchmarks.html*, 2009.

[27] "Miracl crypto," *https://certivox.com/solutions/miracl-crypto-sdk/*.

[28] B. Lynn, "Pbc library," *http://crypto.stanford.edu/pbc/*.

[29] I. Blake, G. Seroussi, and N. Smart, "Pairings," *Advances in elliptic curve cryptography*, pp. 183–213, chapter 9, Cambridge University Press, 2005.

Hongwei Li is an Associate Professor at the School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Dr. Li is the Editor of International Journal of Research and Reviews in Wireless Sensor Networks. He also served as the Technical Program Committee Member for the 7th International Conference on Body Area Networks, and the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing. He is an IEEE Member, an IEEE ComSoc Member, an IEEE Computer Society Member, a China Computer Federation Member and a China Association for Cryptologic Research Member.



Xiaodong Lin (S'07-M'09-SM'12) received the Ph.D. degree in information engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1998 and the Ph.D. degree (with Outstanding Achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008. He is currently an assistant professor of information security with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada. His research interests include wireless network security, applied cryptography, computer forensics, software security, and wireless networking and mobile computing. Dr. Lin was the recipient of a Natural Sciences and Engineering Research Council of Canada (NSERC) Canada Graduate Scholarships (CGS) Doctoral and the Best Paper Awards of the 18th International Conference on Computer Communications and Networks (ICCCN 2009), the 5th International Conference on Body Area Networks (BodyNets 2010), and IEEE International Conference on Communications (ICC 2007).

Haomiao Yang received his M.S. degree and Ph.D. degree in Computer Applied Technology from University of Electronic Science and Technology of China (UESTC) in 2004 and 2008 respectively. He is an instructor in Network Security Technology Laboratory in UESTC. Currently, he is doing post-doctoral research in the Research Center of Information Cross-over Security,Kyungil University, Republic of Korea. His research interests include cryptography, cloud security, and the cyber-security for smart grid.

Xiaohui Liang received the B.Sc. degree in Computer Science and Engineering and the M.Sc. degree in Computer Software and Theory from Shanghai Jiao Tong University (SJTU), China, in 2006 and 2009, respectively. He is currently working toward a Ph.D. degree in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include applied cryptography, and security and privacy issues for e-healthcare system, cloud computing, mobile social networks, and smart grid.

Rongxing Lu received the Ph.D.degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2006 and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2012. He is currently a Postdoctoral Fellow with the Broadband Communications Research (BBCR) Group, University of Waterloo. His research interests include wireless network security, applied cryptography, and trusted computing.

Xuemin (Sherman) Shen is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He was the Associate Chair for Graduate Studies from 2004 to 2008. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, vehicular ad hoc and sensor networks. Dr. Shen served as the Technical Program Committee Chair for IEEE VTC'10 Fall and IEEE Globecom'07. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.