SELECTED PAPERS FROM IEEE/CIC ICCC 2017

Privacy-Preserving Prepayment Based Power Request and Trading in Smart Grid

Shaohua Li[1,2], Xiang Zhang[1], Kaiping Xue[1,2,*], Lijie Zhou[3], Hao Yue[3]

[1] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China.
[2] Science and Technology on Communication Networks Laboratory, Shijiazhuang, Hebei, 050081, China.
[3] Department of Computer Science, San Francisco State University, San Francisco, CA 94132, USA.
* The corresponding author, email: kpxue@ustc.edu.cn

Abstract: Demand response has been intensively studied in recent years. It can motivate customers to change their consumption patterns according to the dynamic (time-varying) electricity price, which is considered to be the most cost-effective and reliable solution for smoothing the demand curve. However, many existing schemes, based on users' demand request in each period, require users to consume their requested electricity exactly, which sometimes causes inconvenience and losses to the utility, because customers cannot always be able to consume the accurate electricity demand due to various personal reasons. In this paper, we tackle this problem in a novel approach. Instead of charging after consumption, we adopt the prepayment mechanism to implement power request. Furthermore, we propose a trading market running by the control center to cope with the users' dynamic demand. It is noteworthy that both users' original demand and trading records are protected against potential adversaries including the curious control center. Through the numerical simulation, we demonstrate that our scheme is highly efficient in both computation and communication.
Keywords: Demand response, power request, smart grid, prepayment, privacy preserving.

I. INTRODUCTION

With the development of IT and IoT technologies, smart grid is regarded as the next generation of power grid which utilizes information and communication technologies to achieve a more efficient, reliable, flexible and sustainable system [1-3]. Specifically, smart grid uses two-way flows of electricity and information to create an automated and distributed energy delivery network. In smart grid, the control center can instruct the utility companies to generate electricity based on users' demand requests, i.e., power request [4,5]. This mechanism is useful and desirable because users' electricity demand can always be satisfied. Another benefit is that the redundant power generation can be avoided, which cuts down the cost of the utility and makes less carbon dioxide emission to the environment [6,7].

Power request can be implemented in different ways [7,8]. For example, anonymous credentials are used for privacy preservation in [8]. Users request credentials from the control center in advance, and then submit their power usage demand by sending a certain number of the credentials back. The problem is that the actual consumption of the users is usually

different from the demand they submit, which will seriously disturb the generation plans of the grid. As a common solution, an extra charge will be added to the users' electricity bills to make up the utilities' loss. Similarly, in [7], penalties will be imposed on the user if the submitted request and actual consumption do not match.

An accurate electricity demand estimation is not easy. Many unexpected occurrences, e.g., a business trip or a bad weather, would change the customers' demand dramatically. Since the differences between users' power requests and their actual consumption may lead to great losses and inconvenience, we need a new method more than simply imposing penalties on users. In fact, the changes of users' actual demand lead to the surplus or shortage of their original requests. A basic idea is to transfer the electricity from the users who request too much to the ones who request too little. In this way, the changes of most users' demand will be neutralized and the negative effect on the utility's generation schedule will be minimized.

In order to realize the above idea, in this paper, we introduce a prepayment mechanism [9,10], which requires smart meters equipped with a built-in switch [11,12], and digital credentials called tokens used to represent users' demand. In every requesting phase, users can estimate the amount of electricity they will consume and make their power requests by paying the corresponding electricity bills in advance. At the same time, smart meters receive equivalent tokens for consuming electricity afterward. To adapt to the users' dynamic demand, the control center runs an electricity trading market. If a user's demand changes, he/she can sell or buy electricity tokens in the market before the next requesting phase.

Many studies have shown that users' demand can reveal their private information [13, 14]. Therefore, we need to ensure the privacy of these procedures. For example, if a user sells a lot of electricity tokens in the market, adversaries can infer that no one in the house and then break into it. Hence, users' privacy must be properly preserved. In this paper, we propose a feasible solution in a privacy-preserving way. Specifically, our contributions are summarized as follows:

- We adopt the prepayment mechanism and propose a power request and trading scheme. By power request, the grid utilities can schedule the power generation, distribution, and transmission more economical; by electricity token trading, customers' dynamic demand can be satisfied, thus achieving the win-win between customers and the grid utilities.
- We design two security protocols to protect individual privacy from the malicious adversary or semi-trust grid utilities, such as control center and the gateway. These two protocols can respectively meet two different requirements of the gateway deployment.

The remainder of the paper is organized as follows. Section II discusses the related works. System model and design goals are described in Section III. Our proposed scheme is presented in Section IV. Security and performance analysis is presented in Section V and VI respectively. We draw our conclusions in Section VII.

In this paper, we discuss the users' dynamic demand and propose a power request and trading scheme to deal with it.

II. Related Work

Smart grid has been widely studied in recent years [1, 2], and power request is one of the most fundamental problem. With the help of power request mechanism, the grid is supposed to be achieve outstanding balance between users' demand and the utility's supply [6, 15]. Usually, privacy protection is an essential requirement for an appropriate power request scheme [4, 13] as well as others [16, 17]. The existing schemes for IoT security, such as [18–21], cannot be adopted to meet these requirements, especially privacy preserving. Methods in [7, 22, 23] adopted homomorphic encryption to aggregate users' demand at the gateways so that no one can obtain particular user's demand, including the control center. Chim *et al.* [8] proposed that extra charge

should be imposed on user according to the difference between submitted demand and real consumption. Uludag *et al.* [24] proposed a hierarchical architecture to achieve secure and scalable data collection which can minimize the total collection time. Furthermore, Yang *et al.* [25] proposed a power request scheme which can not only achieve demand aggregation of multiple users but also provide demand traceability of malicious users.

Electricity trading is also an important component of smart grid [26–28]. Tushar *et al.* [26] proposed an energy trading scheme which considered price discrimination using game theory and achieved social and Pareto optimization. Wang. *et al.* [27] designed an incentive mechanism using Nash bargaining theory to encourage proactive energy trading and fair benefit sharing in interconnected microgrids. Specifically, an effective trading approach which adopted double auction mechanism and guaranteed an equilibrium point was provided in [28].

Prepayment mechanism is very necessary to achieve effective electricity trading [29, 30]. Tewari *et al.* reviewed the economics, logistics, and technology underlying the South African experiment of prepaid electricity [11].

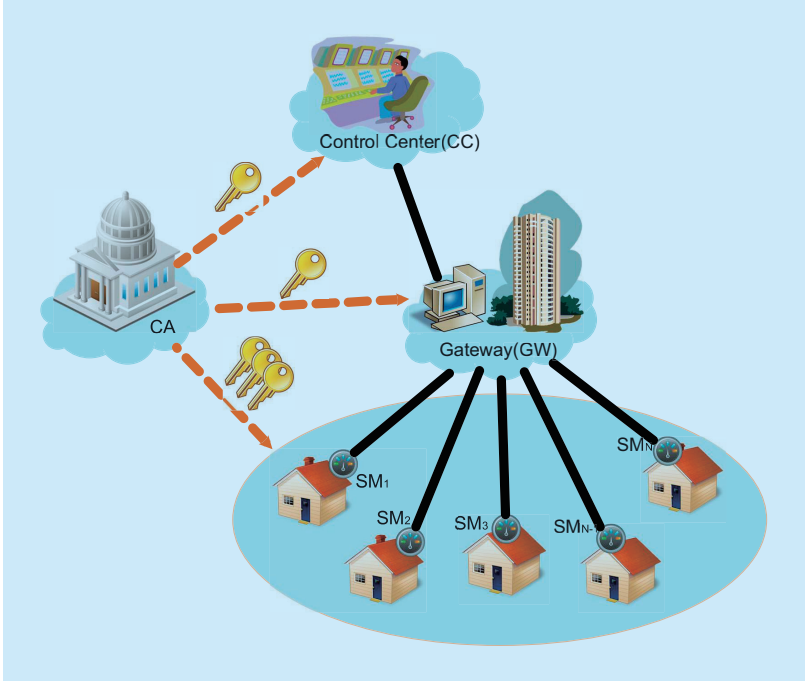


Fig. 1. *System model.*

And Kuzlu *et al.* discussed the communication technology requirements for electric service prepayment [9]. To reduce electricity loss, Jain *et al.* proposed a prepaid smart meter which contains a prepaid card analogous to mobile SIM card [12]. Once the prepaid card is out of balance, the consumer load is disconnected from the grid by the contactor.

III. System Model, Adversary Model, and Design Goals

3.1 System model

As described in figure 1, we consider the following hierarchical smart grid system, which consists of four kinds of entities: a global certificate authority (CA), a control center (CC), a neighborhood area gateway and a number of smart meters (SM) for users.

- **The certificate authority(CA)** is a fully trusted entity in our system. All the other entities in the system apply for key pairs from CA. Especially, every smart meter can apply for multiple key pairs. We assume CA does not collude with control center or gateways.
- **The control center(CC)** is the coordinator of the system, which controls power generation, transmission, distribution, power request, billing and so on. In our scheme, CC also runs an electricity trading market to support buying and selling between users. CC is assumed to be honest but curious. It follows the protocol honestly, but is also curious about users' private information.
- **The gateway(GW)** is located in residential area and is responsible for message collection, forwarding, and possible computing tasks. We assume the GW is honest but curious, just like the CC. Based on the different storage and computation assumption, we consider two scenarios of the GW:
  1) The GW has limited storage and computation capabilities. It only acts as a relay between smart meters and CC, and no complex operations will be executed on it.

2) The GW has sufficient storage and computation capabilities to perform high-cost operations, such as asymmetric encryption and decryption. We call this kind of gateway as an enhanced gateway.

We will introduce our scheme under the above two scenarios in Section IV. Specifically, we will first introduce our scheme under the first scenario in detail, and then briefly describe the main changes of the scheme under the second scenario.

- **Smart meters(SM)** are installed in users' homes and used to generate power requests. Each SM contains a built-in switch controlling whether the user can consume the electricity. To guarantee the protocol runs properly, we assume that the switch cannot be broken from outside and users cannot access the grid without the smart meters.

3.2 Adversary model

We assume that an adversary can either be an outsider or insider. An outside adversary mainly aims at intruding users' privacy, disturbing normal operations or even compromising the grid utilities. We consider that the outside adversary is able to eavesdrop all network communications, replay the transmitted messages, as well as inject bogus messages to the grid. If the adversary is an insider, besides eavesdropping, replaying and injecting messages, it can also commit frauds in electricity payment. For example, when users consume electricity by specialized tokens, the inside adversary may try to consume electricity by forging or double spending tokens.

3.3 Design goals

Our design goal is to develop a privacy-preserving power request scheme which can also enable usage control and electricity trading for users. Specifically, the following three objectives will be achieved.

- An electricity trading market should be provided. Users can buy or sell electricity tokens in this market to decrease the losses caused by the dynamics of their demand.
- A usage control mechanism should be included. It means users cannot consume the electricity they have not requested. It guarantees the effective trading operation.
- Users' privacy should be well preserved. Unauthorized entities cannot know how much electricity a user intends to consume or trade, including GW and CC.

IV. Proposed Scheme

In this section, we present the details of our proposed scheme. We first introduce the general procedures for users to request, consume and trade electricity. Then we describe the details of power request by prepayment. After that, how to consume electricity with tokens is elaborated. Finally, we present the implementation of the electricity trading market.

To be noted, as we introduced in Section III-A, we consider two different assumptions about the storage and computation capacity of the GW. In the section, we first present our detailed scheme under the first assumption, that is, the GW has limited storage and computation resources, and will be used as a relay between the SMs and CC. Then, we will show the changes of the scheme under the second scenario in the last subsection.

4.1 Overview

As shown in figure 2, we define a consumption phase as a period of one week or one month. At the end of every consumption phase, i.e., in requesting phase, users request the electricity they plan to consume in the next consumption phase. Electricity price in requesting phase is normal and fixed. The control center could

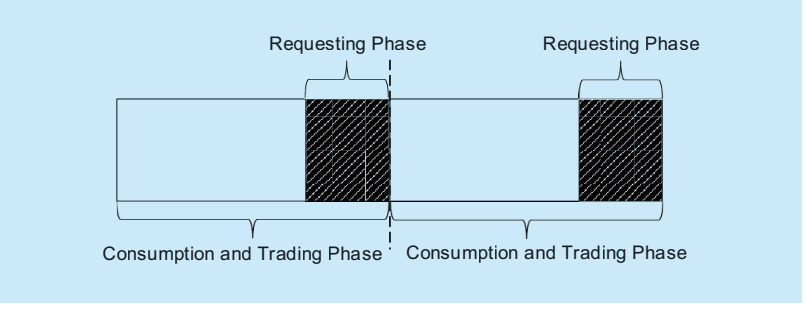


Fig. 2. *General procedure.*

predict the electricity demand of a region, and then make a reasonable distribution of electricity.

To make sure that users estimate their electricity bills as accurate as possible, financial measures are necessary. If users need extra electricity after the requesting phase, they can make new requests at a higher price (e.g., 1.5 times of the original price). If users cannot consume all the electricity they requested by the end of the consumption phase, the tokens will be reclaimed by the control center at a lower price (e.g., half of the original price). Meanwhile, an electricity trading market is provided, where users can declare their desired electricity and price to the control center, and the control center operates the trading according to their bids. The trading phase is overlapped with the consumption phase but it must be completed before the next consumption phase starts.

4.2 System initialization

The CA is responsible for generating secret keys for SMs and CC. For each smart meter SM_i, it generates *m* public/private key pairs, denoted as $(PK_{ij}, SK_{ij}), j = 1,2,\ldots,m$. And for the CC, it generates (PK_{CC}, SK_{CC}).

Then, the CA distributes the secret keys to each SM_i and CC. At the same time, the PK_{CC} and all the PK_{ij} are sent to SM_i and CC, respectively.

4.3 Power request with prepayment

Next, we present our proposed power request procedure with prepayment in detail. In the requesting phase, user $i(i = 1,2,...,N)$ requests d_i kWh of electricity through his/her smart meter SM_i to CC by prepayment. SM_i divides the demand d_i into *m* random parts, $d_{i1}, d_{i2}, ..., d_{im}$, satisfying:

$$d_i = d_{i1} + d_{i2} + \cdots + d_{im},$$

which will be requested with *m* pairs of keys of SM_i respectively. Without losing generality, we take the steps of requesting d_{i1} with the key pair (PK_{i1}, SK_{i1}) as an example, where PK_{i1} is one of SM_i's public keys and SK_{i1} is the corresponding private key. As shown in figure 3, it includes the following steps:

- *Step-1:* SM_i chooses a random number as the secret key K_1 and computes: $C_{i1} = E_{PK_{CC}}(K_1 \| TS)$ where PK_{CC} is the public key of CC, and TS is the current timestamp. The operator "$\|$" denotes the concatenation of two strings or messages. Then SM_i generates a signature on C_{i1} with its private key: $S_{i1} = Sig_{SK_{i1}}(C_{i1})$ and sends $C_{i1} \| S_{i1} \| PK_{i1}$ to CC. CC checks the signature S_{i1} with PK_{i1} and accepts the signature if the following equation holds:

$$E_{PK_{i1}}(S_{i1}) = C_{i1}.$$

After checking the validity, CC decrypts C_{i1} to obtain K_1' and TS'. If TS' is a valid timestamp, CC encrypts K_1' with PK_{i1} and sends it to SM_i.

- *Step-2:* By decrypting $E_{PK_{i1}}(K_1')$ with SK_{i1}, SM_i obtains K_1' and compares it with K_1. If they are equivalent, a secure channel between SM_i and CC has been established and K_1 is the session key. From now on, all the communications between them will be encrypted with K_1. Then, SM_i computes $E_{K_1}(d_{i1})$ and sends it to CC, where d_{i1} is the first part of user i's demand.
- *Step-3:* CC decrypts $E_{K_1}(d_{i1})$ and obtains d_{i1}. Then CC computes $PH_{i1} = H(PK_{i1} \| d_{i1})$, where H is a hash function. Also, a payment message, e.g., a web link of paypal, for SM_i is generated, which is denoted as PM_{i1}. After that, CC sends $E_{K_1}(PH_{i1} \| PM_{i1})$ to SM_i and waits the demand d_{i1} to be paid. SM_i receives and decrypts $E_{K_1}(PH_{i1} \| PM_{i1})$ to extract PH_{i1} and PM_{i1}. If PH_{i1} equals to $H(PK_{i1} \| d_{i1})$, user i will pay for the demand online according to PM_{i1}.
- *Step-4:* After the payment is done, CC

generates corresponding tokens for SM_i. For the purpose of preserving privacy, the number of the generated tokens is fixed no matter how much d_{i1} is. These tokens can be expressed as $t_{11}, t_{12}, ..., t_{1c}$ where c is a constant and d_{i1} kWh of electricity can be consumed using the c tokens. Then CC computes $TH_{i1} = H(t_{11} \| t_{12} \| ... \| t_{1c})$ and sends $E_{K_1}(t_{11} \| t_{12} \| ... \| t_{1c} \| TH_{i1})$ to SM_i. SM_i request the tokens and TH_{i1} by decrypting $E_{K_1}(t_{11} \| t_{12} \| ... \| t_{1c} \| TH_{i1})$. After checking the validity of TH_{i1}, SM_i stores these c tokens.

After m times of the above operations, SM_i submits its demand d_i to the control center successfully. However, CC will not know how much electricity user i requests, because the secure communications of power request are based on different keys, and CC cannot associate any keys to any SM.

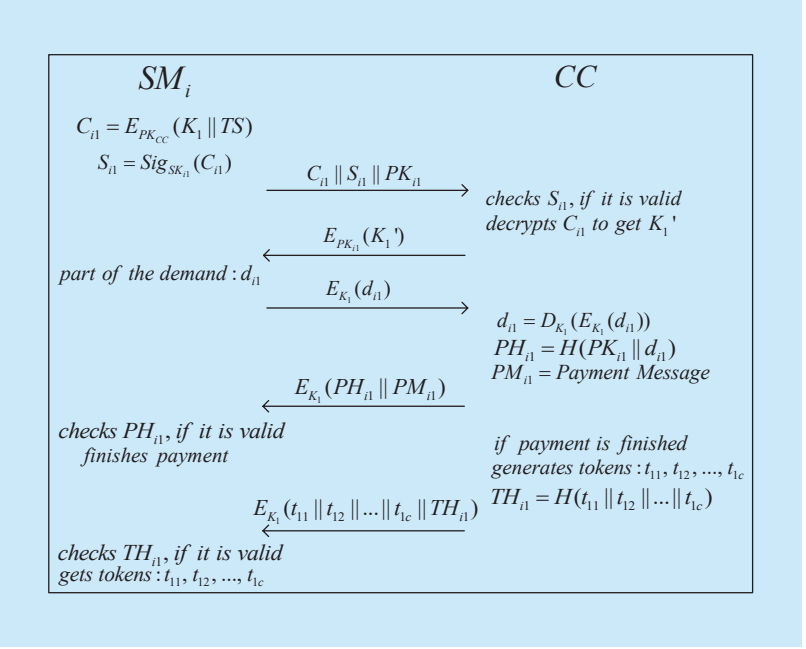


Fig. 3. *Power request by prepayment.*

4.4 Tokens and consumption

Essentially, tokens are the legitimate credentials that users purchase from CC. Every token can be expressed as:

$$token = \left\{ n \| T \| r \| Sig_{SK_{CC}}(n \| T \| r) \right\}$$

Here n is the denomination represented in the token, which means n kWh of electricity is allowed to be consumed. T represents a time point after which the token cannot be used anymore, and r is a random number to avoid two identical tokens. $Sig_{SK_{CC}}(n \| T \| r)$ is CC's signature on $n \| T \| r$. Since no one knows CC's private key, except CC itself, the signature cannot be generated correctly, and hence a valid token cannot be forged.

The switch in SM_i is off by default and SM_i needs to send its tokens to CC to turn the switch on. Once the switch is on, user i can consume electricity from the grid. The detailed steps are:

- *Step-1:* SM_i chooses one of its key pairs (PK_{iv}, SK_{iv}) and establishes a secure channel with CC. In addition to using a different session key, the secure channel is exactly the same as the one established in the requesting phase. So we will not repeat the process. Then, SM_i chooses one of its tokens, denoted by t_{ve}, and uses SK_{iv} to create a signature on t_{ve}: $Sig_{SK_{iv}}(t_{ve})$ and sends $t_{ve} \| Sig_{SK_{iv}}(t_{ve})$ to CC.
- *Step-2:* Upon receiving $t_{ve} \| Sig_{SK_{iv}}(t_{ve})$, CC accepts t_{ve} as a valid token if the signature in t_{ve} is valid. Then, CC checks if t_{ve} is already used. It maintains a database of all the used tokens and users' signatures on them and queries t_{ve} in the database. If t_{ve} is not in the database, CC verifies the validity of $Sig_{SK_{iv}}(t_{ve})$ successfully (CC accepts the signature if the equation holds: $E_{PK_{iv}}(Sig_{SK_{iv}}(t_{ve})) = t_{ve}$). After that, CC adds t_{ve} and $Sig_{SK_{iv}}(t_{ve})$ to the database and sends a verification success message of t_{ve} to the SM_i.
- *Step-3:* SM_i receives the verification success message and checks n_{ve}, which is the denomination represented in t_{ve}. The switch in SM_i stays on until n_{ve} kWh of electricity is consumed completely. By then, another unused token will be sent to CC. When all

the tokens are used, user i will be cut off from the grid.

Note that in the consumption phase, although users send their tokens to the control center, the control center still does not know anything about any user's demand or consumption information.

4.5 Trading market

As we discussed above, users' demand changes dynamically and the electricity they requested may not be the same as they consume actually. However, if users are in shortage of their original requests, they can further request more from the control center, but the price would be much higher than that of the original ones. Also, if they requested too much beyond they actually need, the electricity will be reclaimed by the control center at a very low price. In either situation, users have to take financial losses eventually. Driven by that, users can choose to trade electricity in the market to minimize their losses. Since electricity is provided by tokens, the trading of electricity is actually the trading of tokens.

The trading market is open to users in every consumption phase. When user $i(i \in 1,2,...,N)$ decides to participate in the market, SM_i first establishes a secure channel with CC using one of its key pairs. If user i is a buyer, SM_i can choose a random key pair. But if user i is a seller, SM_i can only use the key pairs with which the tokens to be sold can be obtained. For brevity, we denote the key pair as (PK_i, SK_i). With some interactions, SM_i learns about user i's trading information, including user i wants to buy or sell tokens, the amount of the tokens user i wants to trade, and the price. The trading information can be denoted as BS, Q and P, respectively. SM_i generates a signature on them: $Sig_{SK_i}(BS \| Q \| P)$ and sends $BS \| Q \| P \| Sig_{SK_i}(BS \| Q \| P)$ to CC through the aforementioned secure channel.

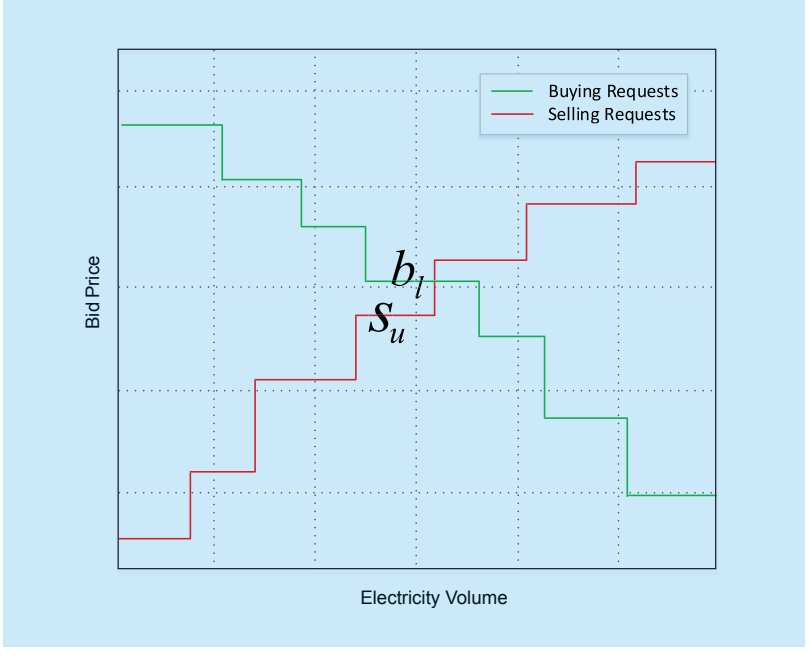


Fig. 4. *Double auction.*

After receiving the user i's trading request message, CC checks the signature with PK_i. If it is valid, CC accepts the request and divides all accepted requests into two categories according to BS: buying bids and selling bids. Because the total amount of the electricity the buyers want to purchase and the sellers want to sell can be different, we need to determine who is allowed to trade and who is not. Besides, the price at which the buyers and sellers make a deal is also important. Since many buyers and sellers are involved, double auction mechanism can be employed as in [28]. CC conducts the double auction and the trading by the following steps:

- *Step-1:* Assume there are X buyers and Y sellers. Denote the buying bids as $(b_j, x_j), j \in \{1,2,...,X\}$ where b_j is buyer j's bid and x_j is the buying amount, and denote the selling bids as $(s_k, y_k), k \in \{1,2,...,Y\}$ where s_k is seller k's bid and y_k is the selling amount.
- *Step-2:* CC sorts the buying bids in a decreasing order as follows: $b_1 \geq b_2 \geq ... \geq b_X$. Also, CC sorts the selling orders in an increasing order such that we have: $s_1 \leq s_2 \leq ... \leq s_Y$. Then, the buying curve (buyers' bids b_j as a function of the buying amount $x_j, j \in 1,2,...,X$) and the selling

curve (sellers' bids s_k as a function of the selling amount $y_k, k \in 1,2,...,Y$) can be generated, as shown in figure 4. These two curves will intersect at a point that corresponds to a buying bid (b_l, x_l) and a selling bid (s_u, y_u) with $b_l \geq s_u$. This intersection point is easily computed using known numerical and graphical techniques.

According to the double auction theory, l buyers and u sellers can participate in the trading process except that buyer l and seller u can only trade a part of their request amount. The trading price should be within the interval $[s_u, b_l]$. Without loss of generality, all buyers $j \leq l$ and sellers $k \leq u$ will trade tokens at a price p_t where $p_t = \frac{b_l + s_u}{2}$. It is possible that buying curve and selling curve have no intersection, which means no buyer or seller can participate in the trading, or all buyers and sellers can participate in the trading and the trading price p_t can be seen as $l = X$ and $u = Y$.

- *Step-3:* The l buyers and u sellers receive a message indicating that they are allowed to proceed trading. The message also includes the trading price and the number of tokens the buyer or seller can trade. According to the message, every buyer pays for his/her buying amount to CC and every seller sends his/her selling tokens and a signature on these tokens to CC. To preserve their privacy, the money buyers pay will not be available to the sellers right away but will be used to offset when the sellers pay for tokens in the next requesting phase. After verifying the tokens and the signatures from the sellers and adding them to the database, CC calculates the difference between the sum of the denominations represented in the tokens and the allowed trading amount of each seller. Then, CC generates new tokens for sellers according to the difference and for buyers according to their payment. Finally, the tokens are sent to the buyers and sellers through secure channels.

Because CC may receive the trading requests at any time, the auction operation will happen at intervals of, e.g., one hour. Every time CC executes a double auction, both the new traders and the ones who failed to trade before are involved. To increase the possibility of trade, the former traders can send fresh bids to override their original bids. They can also stick to their original bids to maximize their benefit.

In the extreme case, there may be still a lot of buyers and sellers remaining in the market who do not give in about their bids so the trading cannot carry on. But this case will not happen or last long. Based on the assumption of rational behavior, For the buyers, if they do not raise their bids and trade in time, the tokens in the market will bid by others at a higher price, and they will purchase tokens from CC, where the tokens are more expensive. For the sellers, it is similar that if they stick to their original bids, their surplus tokens can only be abused or reclaimed by CC at a very low price. In short, compared with being stubborn and no trading, the buyers and sellers will always benefit more from the trading. It is just a matter of buyers and sellers gaming who give in first and how much they give in. We do not care about the detailed process, and know that the market will reach a point eventually, only selling or buying bids will remain.

4.6 Proposed scheme with enhanced gateway

The above procedures assume that the GW is a relay between the SMs and CC, and does not perform any complex computations. But sometimes, there may deploy a GW with strong computing power. We here provide an alternative scheme for this scenario. We will show the main changes to each phase in the following description, and the same procedures will be omitted.

1) System Initialization: In this scenario, each smart meter SM_i only needs one pair of secret keys (PK_i, SK_i), while it is m under the first scenario. And the GW here will be distributed with a pair of secret keys (PK_{GW}, SK_{GW})

by CA. As for CC, it also has (PK_{CC}, SK_{CC}) as its key pairs.

2) Power Request with Prepayment: In the requesting phase, the main change is that every request or response from SMs and CC should be processed by the GW first, which guarantees the individual users' privacy as well as reduces the computation overhead of both SMs and CC. When user $i(i = 1,2,\ldots,N)$ wants to request d_i kWh, instead of dividing d_i into m random parts, SM_i encrypts d_i directly with the session key established with CC. And the session key establishment is based on the only (PK_i, SK_i) of SM_i, which is quite different from the previous version, in which the SM_i has m key pairs. The changes of each steps are as follows:

- *Step-1:* SM_i computes $C_i = E_{PK_{CC}}(K_i \| TS)$, where K_i is a random session key and TS is the current timestamp. Then SM_i generates $S_i = Sig_{SK_i}(C_i)$ and sends $C_i \| S_i \| PK_i$ to the GW, who will accept the signature if $E_{PK_i}(S_i)$ equals to C_i. After verification, the GW generates a new signature on C_i: $S_i^{'} = Sig_{SK_{GW}}(C_i)$, and sends $C_i \| S_i^{'}$ to CC, who will verify and decrypt the C_i to obtain the session key.
- *Step-2:* We cancel this step due to the verification of the session key can be done in the third step, and the GW replaces the CC to verify the public key of SM_i.
- *Step-3 & Step-4:* We replace the PK_{i1} in PH_{i1} with an alias generated by the GW. This alias is unique and will be regenerated for SM_i on each requesting phase. The SM_i can obtain its alias from the GW. The rest procedures are the same as previous, we do ot repeat them here.

3) Tokens and Consumption: Recall that during this phase, the previous procedures that related to SM_i's identity are the signature generation of the tve and its verification in *Step-1 & Step-2*. Now, in this new scenario, the SM_i will send $t_{ve} \| Sig_{SK_i}(t_{ve})$ to GW instead of CC. And the GW will verify the legitimacy of $Sig_{SK_i}(t_{ve})$, then regenerate a new signature $Sig_{SK_{GW}}(t_{ve})$ and forward $t_{ve} \| Sig_{SK_{GW}}(t_{ve})$ to CC to complete the next procedures. The rest of consumption phase is the same as before.

4) Trading Market: Just like the previous consumption phase, the trading information and its signature will be verified by the GW firstly in this scenario, after which the GW will generate a new signature for CC to verify. Specifically, the SM_i generates a signature on the trading information $Sig_{SK_i}(BS \| Q \| P)$, and forward it as well as $BS \| Q \| P$ to the GW, which will verify the information and generate a new signature $Sig_{SK_{GW}}(BS \| Q \| P)$ for CC to verify. Other processes remain no change.

We declare that the powerful GW will reduce the burden of both SMs and CC before. From the above changes, we can conclude the following advantages:

- During the requesting phase, each SM_i no longer needs to generate m requests to protect its privacy. Instead, he/she only generates one request. This is because the non-colluding GW will verify the requests and regenerate a new signature, while only the CC can know the d_i value. At the same time, the CC also only needs to verify once for each SM_i in each requesting phase.
- In every phase, especially the requesting phase, many SM_i may send the messages simultaneously. After verifying each signature, the GW can generate one signature for all the messages, so that the CC only needs to verify once, too. This will significantly reduce the overhead of signature verification.

V. Security Analysis

In this section, we analyze the security properties of our proposed scheme in three phases: requesting phase, consumption phase, and trading phase. To be noted, the analysis below is based on the first scenario, and we omit

the security analysis of the scheme under the second scenario. This is because the latter only changes the verification procedures of the former, the security guarantee of which is straightforward.

5.1 Requesting phase

- Mutual authentication: Smart meter $i(i \in 1,2,...,N)$ with public key PK_i can obtain CC's public key PK_{CC} from CA and generate the ciphertext $C_i = E_{PK_{CC}}(K \| TS)$ which can only be decrypted by CC. Once CC sends the $E_{PK_i}(K')$ back and $K' = K$, the smart meter can believe it is from the CC and CC is authenticated. Using the similar method, CC can also achieve the authentication of every smart meter.
- Privacy-preservation: In our scheme, each smart meter owns multiple keys, which can be regarded as its different identities. The smart meter divides the demand into multiple parts and requests the parts with different identities separately. CC only knows each part of the demand is requested using a legitimate identity, but it cannot learn which keys belong to the same smart meter. Thus, every user's demand information is well preserved.
- Confidentiality: Once CC and the smart meter finish the authentication, they both calculate a secret session key K. Since no one else can obtain K, the confidentiality of the following communications encrypted with the K can be achieved.

5.2 Consumption phase

- Defense against token reusing: When a user wants to consume electricity, the smart meter has to send a token and corresponding signature to CC for verification. CC maintains a database of all used tokens and signatures. Every time CC receives a token, CC checks whether it is a valid token and then queries it in the database. If the token can be found in the token database, the verification will fail and the switch will stay off. As a result, reusing a token is impossible in our scheme.

5.3 Trading phase

During the electricity trading, our scheme can also achieve privacy-preservation and defense against token reusing. Like in the requesting phase, the smart meter participates in the trading with different keys. Since CC cannot associate these keys with a specific smart meter, the privacy of user's trading amount can be preserved. Moreover, when sellers sell their tokens, the tokens and sellers' signatures on them are sent to CC and added into the database. Clearly, the sellers are not able to receive electricity by using these tokens anymore.

VI. Performance Analysis

In this section, we will evaluate the performance of our scheme in terms of computation and communication overhead.

We use RSA, AES, and SHA-1 to evaluate the computation overhead of our scheme. Corresponding parameters are defined as follows:

- TA denotes the time for RSA-1024 encryption/decryption/signature operation.
- TS denotes the time for AES-128 encryption/decryption operation.
- T_H denotes the time for SHA-1 operation.

6.1 Computation overhead

We here consider the computation overhead under the first scenario. Since the gateways only forward the messages between smart meters and the control center, we do not take

Table I. *Computation overhead.*

Operation	Smart Meter	Control Center
Requesting	$m(2T_A + 3T_S + 2T_H)$	$m((2+c)T_A + 3T_S + 2T_H)$
Verifying	$3T_A + 2T_S$	$4T_A + 2T_S$
Trading	$3T_A + 3T_S$	$4T_A + 3T_S$

Table II. *Computation Overhead under Enhanced Gateway.*

Operation	Smart Meter	Gateway	Control Center
Requesting	$2T_A + 2T_S + T_H$	$2T_A$	$(2+c)T_A + 2T_S + T_H$
Verifying	$3T_A + 2T_S$	$2T_A$	$4T_A + 2T_S$
Trading	$3T_A + 3T_S$	$2T_A$	$4T_A + 3T_S$

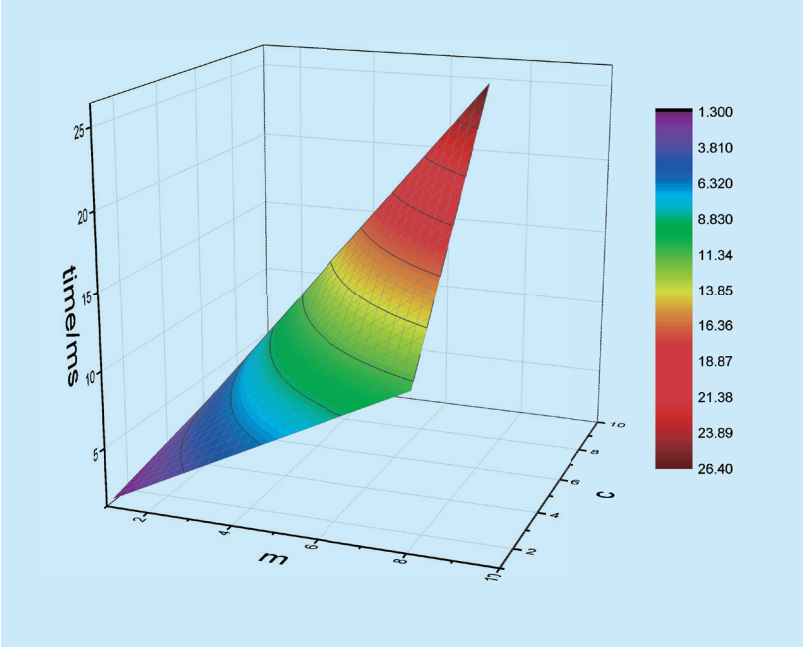


Fig. 5. *Computation overhead of requesting tokens in the first scenario.*

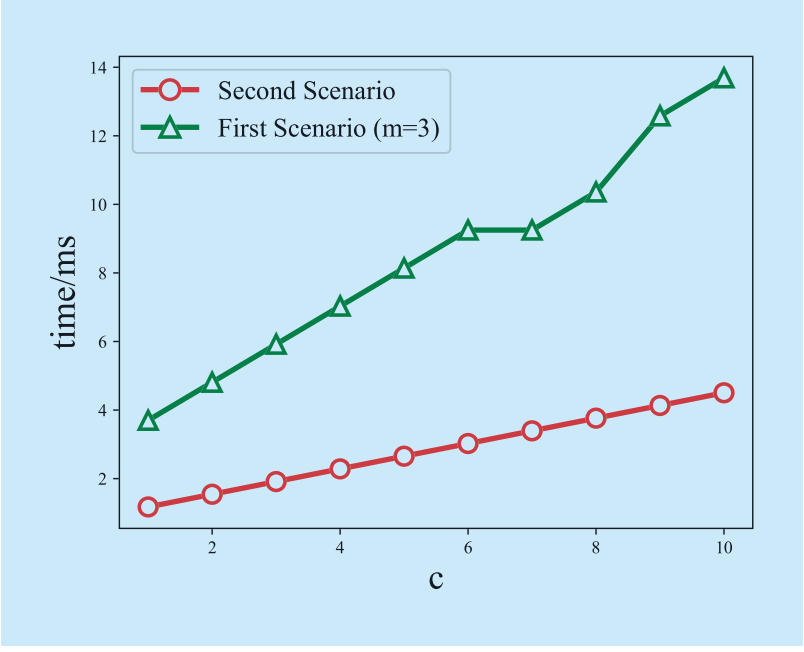


Fig. 6. *Computation overhead of requesting tokens in the second scenario.*

into consideration their computation overhead. Table 1 shows the computation overhead of requesting tokens, verifying a token and trading tokens for smart meters and the control center. Every smart meter has m pairs of keys and requests c tokens with every pair of keys, so the computing time of a complete power request is $m(2T_A + 3T_S + 2T_H)$ and $m(2T_A + cT_A + 3T_S + 2T_H)$ for smart meters and the control center respectively. Compared with requesting tokens, the verification and trading of the token are less frequent, and hence the smart meters and control center have enough computation resources to process them. Thus, we only focus on the computation overhead of requesting tokens.

In our experimental environment (CPU: 2.0 GHz, RAM: 4.0 GB), we run 1000 times to obtain the average result. We find that T_A is approximately 9 times faster than T_S and 25 times faster than H_T (T_A is about 0.37ms on average when using RSA-1024, T_S is nearly 0.041ms on average when using AES-128-cbc, and T_H is about 0.015ms when using SHA-1). Clearly, the overhead of smart meters increases linearly with the increase of m while the overhead of the control center is more complex.

Figure 5 shows the computation overhead variation of the control center with different c and m. Although a larger m improves the security and a larger c makes trading more convenient, the computation overhead also increases as c and m grow. There is a trade-off between security and computation overhead. According to our experiments, we believe that $m = 3$ and $c = 5$ can achieve both high security and low computation cost. In this case, the total time required for requesting tokens for every smart meter is 2.67ms and the total time for control center to serve a smart meter is 8.16ms. Considering that the computation only happens in requesting phases and requesting phases may last for several days, the overhead is affordable for the control center and smart meters.

6.2 Computation overhead under enhanced gateway

Table 2 shows the computation overhead of requesting tokens, verifying a token and trading tokens for smart meters, the gateway, and the control center. Compared with the computation overhead under the first scenario, the gateway has extra computation, that is $2T_A$

for each operation. But the computation overhead of requesting operation is fixed, while the former depends on m. Figure 6 shows the comparison of the communication overhead of the control center in different scenarios. We can see that the computation cost of the second scenario with the enhanced gateway has a significant improvement than the first one with m=3, which is the recommended value of m in Section VI-A. As we described before, there is a particular advantage in the scheme under the enhanced gateway, that is, when some requests arrive at GW simultaneously, the GW could generate one signature for all of them. And the CC also needs to verify only one signature for them. This feature can significantly decrease the computation overhead of both the GW and the CC.

6.3 Communication overhead

In our scheme, the communication overhead mainly comprises of RSA ciphertexts/signatures and AES ciphertexts. When m=1, both two scenarios have almost the same communication overhead. Assume that we use 1024-bit RSA, 128-bit AES and 160-bit SHA-1. The length of a token is 128+128+1024 = 1280 bits. The communication overhead of requesting, verifying and trading tokens once is 6688+1280c bits, 9600 bits, and 10624 bits, respectively. When c=5, the overhead is 1.4KB, 1.2KB and 1.3KB, respectively. Obviously, the communication overhead is considered acceptable.

To be noted, the communication overhead of the second scenario is independent of m, while that of the first scenario increases linearly with m.

VII. Conclusion

In this paper, we discuss the users' dynamic demand and propose a power request and trading scheme to deal with it. We adopt the prepayment mechanism to make electricity trading possible. And the trading market provides an effective way for users to minimize their losses. Furthermore, we show our scheme under the enhanced gateway. Security analysis indicates that the privacy of users' original demand and their trading amounts is well preserved, and performance analysis further demonstrates the efficiency of our scheme in terms of computation and communication.

Acknowledgments

This work is partly supported by the National Key Research and Development Plan of China under Grant No.2016YFB0800301, the Fund of Science and Technology on Communication Networks Laboratory under Grant No. KX162600024, and Youth Innovation Promotion Association CAS under Grant No. 2016394. Shaohua Li and Xiang Zhang contributed equally to this work (co-first authors in alphabetical order). The preliminary results were presented at IEEE ICCC2017.

References

[1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - the new and improved power grid: A survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.

[2] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid." *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 57–64, 2010.

[3] W. Han and Y. Xiao, "Privacy preservation for v2g networks in smart grid: A survey," *Computer Communications*, vol. 91, pp. 17–28, 2016.

[4] S. Zeadally, A.-S. K. Pathan, C. Alcaraz, and M. Badra, "Towards privacy protection in smart grid," *Wireless Personal Communications*, vol. 73, no. 1, pp. 23–50, 2013.

[5] T. Chim, S. Yiu, and L. H. V. Li, "PASS: Privacy-preserving authentication scheme for smart grid network," in *Proceedings of the 2nd IEEE International Conference on Smart Grid Communications*. IEEE, 2011, pp. 196–201.

[6] I. Koutsopoulos and L. Tassiulas, "Optimal control policies for power demand scheduling in the smart grid," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1049–1060, 2012.

[7] T. Chim, S. Yiu, V. Li, L. Hui, and J. Zhong, "PRGA: Privacy-preserving recording & gateway-assisted authentication of power usage information for smart grid," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 85–97, 2015.

[8] T. Chim, S. Yiu, L. Hui, and V. Li, "Privacy-preserving advance power reservation," *IEEE Com-*

munications Mag- azine, vol. 50, no. 8, pp. 18–23, 2012.
[9] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in han, nan and wan," *Computer Networks*, vol. 67, pp. 74–88, 2014.
[10] M. Parvin, S. Kabir *et al.*, "A framework of a smart system for prepaid electric metering scheme," in *Proceedings of the 4th International Conference on Informatics, Electronics & Vision*. IEEE, 2015, pp. 1–5.
[11] D. D. Tewari and T. Shah, "An assessment of south african prepaid electricity experiment, lessons learned, and their policy implications for developing countries," *Energy Policy*, vol. 31, no. 9, pp. 911–927, 2003.
[12] A. Jain and M. Bagree, "A prepaid meter using mobile communication," *International Journal of Engineering, Science and Technology*, vol. 3, no. 3, 2011.
[13] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges," *IEEE Communications Magazine*, vol. 50, no. 8, pp. 38–45, 2012.
[14] S. Tan, D. De, W.-Z. Song, J. Yang, and S. K. Das, "Survey of security advances in smart grid: A data driven approach," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 397–422, 2017.
[15] M. Erol-Kantarci and H. T. Mouftah, "Wireless sensor networks for cost-efficient residential energy management in the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 314–325, 2011.
[16] A. Boustani, A. Maiti, S. Y. Jazi, M. Jadliwala, and V. Namboodiri, "Seer grid: privacy and utility implications of two-level load prediction in smart grids," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 546–557, 2017.
[17] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy- preserving multi-subset aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2018.
[18] X. Yao, X. Han, X. Du, and X. Zhou, "A lightweight multicast authentication mechanism for small scale IoT applications," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3693–3701, 2013.
[19] X. Hei and X. Du, "Biometric-based two-level secure access control for implantable medical devices during emergencies," in *Proceedings of the 30th IEEE International Conference on Computer Communications(INFOCOM)*. IEEE, 2011, pp. 346–350.
[20] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal- credential-based mutual authentication and key agree- ment scheme for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316–323, 2013.
[21] C. Ma, K. Xue, and P. Hong, "Distributed access control with adaptive privacy preserving property for wireless sensor networks," *Security and Communication Networks*, vol. 7, no. 4, pp. 759–773, 2014.
[22] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2014.
[23] A. Abdallah and X. Shen, "Lightweight security and privacy preserving scheme for smart grid customer-side networks," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1064–1074, 2017.
[24] S. Uludag, K.-S. Lui, W. Ren, and K. Nahrstedt, "Secure and scalable data collection with time minimization in the smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 43–54, 2016.
[25] Q. Yang, J. Hong, K. Xue, W. Chen, X. Zhang, and H. Yue, "A privacy-preserving and real-time traceable power request scheme for smart grid," in *Proceedings of 2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.
[26] W. Tushar, C. Yuen, D. B. Smith, and H. V. Poor, "Price discrimination for energy trading in smart grid: A game theoretic approach," *IEEE Transactions on Smart Grid*, 2017.
[27] H. Wang and J. Huang, "Incentivizing energy trading for interconnected microgrids," *IEEE Transactions on Smart Grid*, 2016.
[28] Y. Wang, W. Saad, Z. Han, H. V. Poor, and T. Basar, "A game-theoretic approach to energy trading in the smart grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1439–1450, 2014.
[29] Q.-D. Ho, Y. Gao, and T. Le-Ngoc, "Challenges and research opportunities in wireless communication networks for smart grid," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 89–95, 2013.
[30] R. C. Green, L. Wang, and M. Alam, "Applications and trends of high performance computing for electric power systems: Focusing on smart grid," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 922–931, 2013.

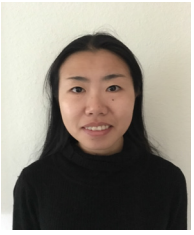
Biographies

Shaohua Li, received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2016. He is currently a graduate student in Communication and Information System from the De-

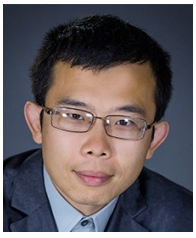
partment of Electronic Engineering and Information Science (EEIS), USTC. His research interests include network security protocol design and analysis.

Xiang Zhang, received the B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2015. He is currently a graduate student in Communication and Information System from the Department of Electronic Engineering and Information Science (EEIS), USTC. His research interests include future Internet and network security.

Kaiping Xue, (M'09-SM'15) received his B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. Currently, he is an Associate Professor in the Department of Information Security and Department of EEIS, USTC. His research interests include next-generation Internet, distributed networks, and network security.



Lijie Zhou, received her B.S. degree in Communication from Qingdao University, China, in 2007, M.A. degree in Linguistics in University of Toledo, OH, USA in 2009. She has been working towards her M.S. degree in the Department of Computer Science at San Francisco State University. Her research interests include data science and cybersecurity.



Hao Yue, received his B.S. degree in Telecommunication Engineering from Xidian University, Xi'an, China, in 2009, and Ph.D degree in Electrical and Computer Engineering from University of Florida, Gainesville, FL, USA, in 2015. He is now an Assistant Processor with the Department of Computer Science, San Francisco State University, San Francisco, CA, USA. His research interests include cyber-physical systems, cybersecurity, wireless networking, and mobile computing.