<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Credential-based privacy-preserving power request scheme for smart grid network</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Cheung, JCL; Chim, TW; Yiu, SM; Hui, LCK; Li, VOK</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>The IEEE Global Communications Conference (GLOBECOM 2011), Houston, TX, USA, 5-9 December 2011. In Proceedings of GLOBECOM, 2011, p. 1-5</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>2011</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/139987">http://hdl.handle.net/10722/139987</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; Globecom. IEEE Conference and Exhibition. Copyright © IEEE.; ©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
Credential-based Privacy-preserving Power Request Scheme for Smart Grid Network

Department of Computer Science
The University of Hong Kong
Email: {ljcheung, twchim, smyiu, hui}@cs.hku.hk

Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Email: vli@eee.hku.hk

Abstract—A smart grid network adjusts power allocation by collecting information about the power usage of the customers in real-time. Authentication and user privacy preservation are the two major concerns on smart grid security. Authentication schemes that preserve users’ privacy from third parties, but not from the power operator, have been proposed. In this paper, we propose a scheme that preserves users’ privacy information, including their daily electricity usage pattern from third parties as well as from the power operator. At the same time, the scheme ensures that authentication can be properly done. These two properties are achieved by using anonymous credential under the principle of blind signature. Basically, a customer generates a set of credentials by himself and asks the control center to blindly sign them. When the customer needs to request more power later on, he presents the signed credential to the control center as proof of his identity. Implementation and analysis show that our scheme is feasible in terms of a number of performance measures such as the signing time and the credential collision rate.

Index Terms—smart grid network, authentication, privacy preservation, anonymous credential, blind signature

I. INTRODUCTION

In this paper, we focus on two major security issues on the communication between the smart meter at the customer premises and the control center maintained by the power operator. Since the requests made by the customers determine the power allocation of the substations, authentication must be done to prevent abuse or attack. Privacy issue has also been raised in earlier discussions of the smart grid system [2] [3]. For example, by observing the power consumption of a family, it is easy for a criminal to know when the family is not at home, and to burglarize it. This concern also applies to commercial customers. A company having a power failure could be at risk if it is revealed. We propose a scheme that can preserve the privacy of the customers not only to third parties, but also to the power operator. At the same time, our scheme provides authentication to prevent hackers from abusing the system. Blind signature is used to achieve privacy preservation.

In our scheme, a customer first generates a set of credentials (analogous to tickets) and blinding factors. Customer “blinds” the credentials and then requests the control center to sign them using the control center’s private key. When the customer needs more power, he will send a credential to the control center anonymously and the control center will adjust the power for the area where the customer is located.

Simulations and mathematical analysis show that our scheme is feasible in terms of credential signing time, communication loading and credential collision chance. 2048-bit credential signing using a laptop takes only 10 minutes. Suppose each customer performs this once per month, the total amount of upload and download per month are only about 10MB and 2.5MB, respectively. We also show that credential collision is extremely unlikely when 128-bit length credential identities are used.

The rest of this paper is organized as follows. Related work is reviewed in Section II. The system model and the security requirements are described in Sections III and IV respectively. Some preliminaries are given in Section V. Our schemes are presented in details in Section VI. Security analysis, discussions and performance analysis are listed in Sections VII, VIII and IX respectively. We conclude our paper in Section X.

II. RELATED WORK

The smart grid project was started by the European Union in 2003 [5]. At about the same time, Electric Power Research Institute of the USA started the IntelliGrid Project [6] and the US DOE also initiated the Grid 2030 project [7]. In 2010, NIST released a report [4] which describes the potential components and cyber security issues of the smart grid system.

A recent work [8] proposes a communication-oriented smart grid framework. New requirements of the communication architecture and possible security problems of the smart grid system are also identified.

Some major security problems have been pointed out and studied [2] [3]. On the communication between the control center and the smart meter, it is proved that statistical analysis approach cannot protect the system from false data injection attack [9]. It would also be infeasible for the smart grid system to adopt this approach since the system will need to handle a large amount of data in real-time, but the control center of the smart grid system only has a few seconds to respond. A solution has been proposed recently [10], which can provide user authentication and some level of user privacy preservation.
III. SYSTEM MODEL

Following [10], we assume that a smart grid network can be simplified into three basic layers to form a hierarchical structure. At the top level, there is a control center maintained by the power operator. At the second level, there are substations inside the distribution network and each substation is responsible for the power supply of an area. At the lowest level, there are smart meters which are placed at the homes of the customers.

Smart meters should send power requests to the control center when they need more power (say when it finds that the current power level is not enough to run all appliances in a customer’s home). The control center can be a single server located inside the power plant or be distributed servers located at different geographical locations for load-basting purposes and to avoid a single point of failure.

In this paper, we focus on the challenge of the trustworthiness of the parties involved. From the power operator’s point of view, its control center and substations are more trusted than the customers since the control center and substations are usually physically protected 1. Smart meters, on the contrary, are not physically protected and can be compromised by dishonest customers or even by hackers who can then abuse the system. From the customers’ point of view, the power operator as well as its control center and substations are only semi-trusted. They perform security operations honestly but they may acquire the electricity usage pattern of the customers and hence deduce the private information of the customers (e.g. when they are at home). As such, customers do not want to reveal too much of their private information to the power operator.

The communication channels from the smart meters to the control center and from the substations to the control center may be the Internet which is public and is always considered as unsafe.

IV. SECURITY REQUIREMENT

We aim at designing a system to resolve the following security problems:

a) Message authentication: Every request message sent by any smart meter should be checked to confirm that it is from a valid user. Authentication is the basis of the system. Without it, anyone can abuse the system easily.

b) Identity privacy preservation: The real identity of the customer during the power requesting phase should be unknown to everyone, including the power operator to protect the privacy of customers.

c) Request message confidentiality: The amount of power requested by any smart meter should not be known by any third party in order to protect the privacy of the customers.

d) Traceability: The total amount of power requested by each customer in a certain period of time should be known by the power operator (i.e. its control center) so that it can correctly charge each customer at the end of the charging period.

VI. PRELIMINARIES

A. Blind signature

Blind signature is a method to allow a first party, Party 1, to sign a message generated by a second party, Party 2, without knowing its actual content. When a third party, Party 3, receives the signed message, it can verify that the message is signed by Party 1. In our scheme, we make use of this technique to allow the control center (Party 1) to sign a credential generated by a customer (Party 2) without knowing its actual content. At a later time, the control center itself (Party 3) can verify that the credential is indeed signed by it without knowing who requested it to sign or when it was signed.

Next let us describe the technique in more details. We assume that the first party, Party 1, possesses a public and private key pair, (e, n) and (d, n) generated under the RSA principle. Party 2 creates a message M for Party 1 to sign. But before that, Party 2 blinds M using a random blinding factor F. The blinded version of M is represented by X = (MF^e) mod n. Party 2 sends X to Party 1 for signature. Next, Party 1 generates its signature on X as SIG(X) = X^d mod n. Party 2 can then retrieve SIG(M) (i.e. Party 1’s signature on the actual message M) by multiplying F^{-1} to SIG(X) where F^{-1} represents the modular multiplicative inverse of F. The resulting signature is represented by SIG(X) = X^{d} = (X^{e} mod n) × F^{-1} = (MF^{e}d mod n × F^{-1} = M^d mod n = SIG(M) because F^{ed} = 1. This is just Party 1’s signature on the actual message M. Hence, when Party 3 receives SIG(M), it can verify using (e, n) that M is indeed signed by Party 1.

The usage of blind signature technique in our scheme is as follows. The customers will prepare a set of credentials, each stating the amount of electricity to request, and ask the control center to sign them blindly so the customer can submit any of these credentials for the request of electricity.

B. Blind signature with message verification

Since Party 1 does not know the actual content of the message sent from Party 2, it cannot verify it in a traditional way. To prevent Party 1 from signing an invalid message or a message that it does not want to sign, there is a method to verify the message before Party 1 signs it. This technique is widely adopted in e-cash schemes like [12]. Party 2 generates n messages using different blinding factors. It then blinds the n messages and sends them to Party 1. Next, Party 1 randomly chooses m messages (m < n) and challenges Party 2 to reveal them by providing the m blinding factors. If the m blinding factors are correct, Party 1 accepts the signature request and signs the remaining (m – n) messages.

Note that in this example, Party 2 can cheat Party 1 successfully with a probability of (n – m) / n. Obviously, a tradeoff in such a scheme is between the chance of cheating and the number of redundant messages required. The more the redundant messages used, the lower the chance that Party 2 can cheat Party 1.

In our scheme, we make use of this technique to ensure that the control center only signs valid credentials.

VI. DETAILS OF OUR SCHEME

In our scheme, we assume that any smart meter can communicate with the control center via a secure
communications channel. That is, every message transmitted is encrypted (say using AES encryption) and third parties cannot read the contents without the key concerned.

The basic idea of our scheme is to make use of the blind signature technique for the control center to sign credentials on behalf of customers. In this way, when a customer presents a credential anonymously (without any information about the customer’s identity provided), the control center cannot tell which customer is making the request, yet it can verify the signature to confirm that it is from a valid customer because only valid customer can request for blind signatures. At the end of each month, each customer sends the unused credentials back to the control center to evaluate the amount of power he has requested so far.

Next let us describe our scheme in details.

A. Setup phase:

During system startup, the control center assigns itself an RSA public and private key pair for signing credentials. The public key is assumed to be known by everyone while the private key is only known by the control center.

Whenever a new smart meter is registered, it will be assigned a unique identity for identification purpose and a secret value for authentication purpose (details of their usage will be discussed later).

B. Registration phase:

At the beginning of each month, the registration phase will be carried out. This phase is not anonymous. Customers need to be authenticated using their real identities in this phase. For this purpose, the smart meter submits its identity and secret value to the control center (via a secure channel) to authenticate itself. This phase continues with the following steps:

Step 1: Each customer (with the help of his/her smart meter) sends credential signing requests to the control center. Each credential \( C_i \) is of the format: \( (CID, date of issuance, value) \). 

\( CID \) is a unique identity for each credential, \( date of issuance \) indicates the date that the credential is issued, \( value \) is the amount of power that the credential holder can request.

Step 2: For credentials of each value, \( n \) credentials with \( n \) different \( CID \)s and blinding factors, where \( n \) is pre-determined by the control center, are generated. Among them, the control center requests the customer to open \( (n – 1) \) of them for verification purpose. Here the message verification scheme mentioned in Section V.B above is used.

Step 3: If the information in all the “opened” credentials is valid, the control center signs the remaining one. Otherwise, it requires the customer to re-submit its request.

Step 4: The customer retrieves the control center’s signature on the actual credential by multiplying the inverse of the blinding factor to the received signature (for details please refer to Section V.A).

Step 5: The customer repeats Steps 2 to 4 for each credential value.

Step 6: The control center calculates and records the number of credentials for each value that it has signed so far into its local database.

Step 7: The smart meter of the customer stores these signed credentials properly for later usage. Since a smart meter can be considered as a tamper-proof device, we assume that the stored signed credentials cannot be modified by an outsider easily.

C. Power requesting phase:

Power requesting phase can be executed at any time during the month when the smart meter of a customer finds that it needs more power to support all the electric appliances in the apartment. This phase is anonymous. Customers do not have to authenticate themselves in this phase and the validity of the customers is represented by the anonymous credentials. The control center cannot determine the identity of each individual customer.

To request for more power, the smart meter of a customer randomly picks a signed credential of the desired value and sends it to the control center. The control center then verifies its own signature on the credential and checks the date of issuance to see whether the credential is up to date. It also checks if the credential number included has been used already. If yes, that credential will be considered as invalid. To facilitate this checking, the control center maintains a list of all used credential numbers. If the above checking is successful, the control center accepts the credential and continues to process the request. Otherwise, it rejects it and asks the smart meter to present another credential in order to complete the power request.

D. Reconciliation phase:

At the end of each month, a reconciliation phase will be carried out. Note that similar to the registration phase, this phase is not anonymous. Customers need to be authenticated using their real identities in this phase. The smart meter of a customer sends all the credentials which have not been used to the control center. The control center then checks the credentials as usual, calculates the sum of their values, \( N \), among all unused credentials and calculates the total requested amount \( R \) by computing \( (T – N) \).

To check against whether a customer requested the same amount of power as he used, we use a traditional kWh meter to collect the monthly power consumption and the system will compare the consumption with \( R \) to see whether the two values are comparable. If there is a big difference between them, an extra charge can be added to the monthly bill of the customer.

VII. Security Analysis

In this section, we evaluate our scheme according to the security requirements listed in Section IV.

A. Message authentication

At the registration phase, a customer needs to authenticate himself/herself using the private key signature before requesting any signing of credentials. So when the customer presents the signed credentials during the power requesting phase, he proves himself/herself authenticated.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

B. Identity privacy preservation

Customers only reveal their identities during the registration and the reconciliation phases. During the power requesting phase, when the customer presents the credentials, the control center only knows whether the credential is from a valid user or not. Due to the properties of the blind signature, the credential identity is only known by the owner. The credentials do not reveal the identity of the customers.

C. Request message confidentiality

As we mentioned earlier, we assume that a smart meter communicates with the control center via a secure channel. Therefore, the amount of power requested by any smart meter cannot be known by any third party. Confidentiality of the request message is preserved.

D. Traceability

In the registration phase, a customer needs to present his/her identity (i.e. not anonymous) to obtain signed anonymous credentials. In the reconciliation phase, a customer again needs to present his/her identity to the control center. Therefore, the total amount of power requested by each customer in a certain period of time (say a month) can be known by the control center. The customer can then be properly charged at the end of the charging period.

VIII. DISCUSSIONS

In our scheme, the smart meter is used to handle the request of the credentials and to send them to the control center when power is needed. It stores its ID and password, a few public keys of the control center and the signed credentials. Even if a hacker compromises his own smart meter, it will not benefit him/her. This is because he/she cannot change the values of the credentials since a valid credential should be signed using the control center’s private key.

The smart meter is not used to protect the benefit of the power operator. It is used to protect the customers’ credentials by insulating the smart grid system from other computer system (such as OS of PC) to avoid malware and hacker attacks. We can assume that as long as the customer does not try to hack the smart meter, it will be safe for him to use the system.

In a nutshell, the benefit of the power operator will not be harmed even when the smart meter is compromised. For the customer, as long as he keeps the smart meter from the hackers physically, it will be safe.

IX. PERFORMANCE ANALYSIS

In this section, we analyze the performance of our scheme in different aspects: credential signing time, communication overhead, and credential collision. 2048-bit credential signing using a laptop takes only 10 minutes. Suppose each customer performs this once per month, the total amount of upload and download per month are only about 10MB and 2.5MB, respectively. We also show that credential collision is extremely unlikely when we use 128-bit credential ID. Thus, our scheme is feasible in terms of performance.

A. Credential signing time

Credential signing is a bottleneck of our scheme. To make sure that our scheme is feasible, we implement this part to analyze its performance.

We have written a test program in Java to measure the time needed for RSA signature on the credentials. We have tested our performance on a laptop computer with an Intel Core 2 Duo CPU, T5870 @ 2.00 GHz. The testing process only ran on a single core. The results are shown in Table 1 and Figure 1. Although 512 bits RSA is not secure nowadays, it is shown here for a better comparison with 1024 and 2048 bits RSA.

<table>
<thead>
<tr>
<th>Key size (bits)</th>
<th>Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>12</td>
</tr>
<tr>
<td>1024</td>
<td>73</td>
</tr>
<tr>
<td>2048</td>
<td>511</td>
</tr>
</tbody>
</table>

Table 1. Signing 10000 credentials with RSA private key of different size

Assume that on the average, each customer uses 10,000 credentials per month, which means the control center needs to sign at least 10,000 credentials for each customer. From the data we obtained from the experiment, signing 10,000 credentials using a consumer PC with 1,024 bits RSA needs about 1 minute, while 2,048 bits RSA needs 10 minutes. Recall that the signing happens on the control center side, which means this process can be done a lot faster by using control center machines or even RSA hardware. It is also straightforward to apply multi-threading on credential signing.

Also, customers need to prepare blinding factors which also require modular exponentiation. However, the exponent part of the public key can be assigned with a small number, so that the exponentiation can be done in a much shorter time, and a cheaper device is required on the customer side.

B. Communication overhead

The amount of data transmission between the control center and the customer should be within a feasible range and it is always better to transmit less. Credentials dominate the data transmission in our scheme. The size of a credential mainly depends on the key size of the RSA signature since the content is usually smaller in size (< 1024 bits). So let us assume that the credential size is equal to the key size.

Assume that each customer uses about 10,000 credentials every month. Then at the beginning of each month, each customer will request the control center to sign 10,000 credentials and the control center will send them back. Assume that we use the message verification in the blind signature and check 2 messages out of 3, thus requiring the
customer to send 20,000 more credentials. Furthermore, let us ignore other control messages since they are relatively small in size. Then the total amount of data being uploaded to the control center is about 10 MB per month while the total amount of data being downloaded from the control center is about 2.5 MB per month. We believe that such a small amount of data transmission should not cause any burden to the network since in the future, a smart grid will probably be built on top of the public Internet.

C. Credential collision

In our protocol, since the control center does not know the credential identity in the registration phase, it cannot avoid two or more credential registrations with the same credential identity (and same timestamp). We call this credential identity collision. The discovery of this possible problem is inspired by the Birthday Paradox – the chance of having two persons among a few dozen of people with the same birthday is extremely high \([11]\). We suspect there might be a similar problem in our scheme. When credential identity collision happens, only one of the credentials will be accepted, and others will be ignored. With a few simple calculations, we will show that the chance of this credential collision is negligible.

One way to decrease the chance is to increase the credential identity size so that any two randomly picked credential identity will have less chance to collide. Assume the credential identity is \(k\) bits in size, and there is a total of \(p\) valid credentials registered in the same period of time.

The chance that an arbitrary credential collides with another random generated credential is \(A = 1/2^k\). The chance for a credential NOT to collide with the other \(p\) credentials is \(B = (1-A)^p = (1 - 1/2^k)^p\). The chance for a credential to collide with any of the other \(p\) credentials is \(C = 1 - B = 1 - (1 - 1/2^k)^p\).

Even when collisions occur, we have two options to resolve the situation. In the first option, the control center ignores the collided credential. The customer concerned does not have to send another credential and can go ahead to consume the power. Since the power represented by each credential is relatively small as compared to those supplied by the substation, and a substation usually supplies a little more than enough power to ensure voltage stability, the power supply will not be affected. In the second option, the control center knows the credential is collided and asks the customer to send another credential. At the end of the month, the amount of power requested by that customer will be a bit higher than his actual power consumption. If we allow a reasonable difference between the requested amount and the actual consumed amount, possible credential collision can be ignored.

X. CONCLUSIONS

This scheme is applicable not only for smart grid. Due to its simplicity and few restrictions, it is also compatible to other systems. This scheme is much more efficient compared with other protocols that use zero-knowledge proof, which usually create a huge load on the communication channel. Implementation details of our proposed scheme such as the communication channel or the protocol are not discussed in depth here. But they must be able to maintain the same security requirement as the scheme does. For example, AES encrypted communication channel should be used in the registration phase and reconciliation phase to prevent message exposure by packet sniffing. Appropriate protocol should be chosen for the power requesting phase to maintain anonymity of the customer. In the future, we will study other security and privacy-preserving problems in the smart grid network.

REFERENCES