Regular Paper

SPaCIS: Secure Payment Protocol for Charging Information over Smart Grid

Hikaru Kishimoto[1,a) Naoto Yanai[1,b) Shingo Okamura[2,c)

Received: March 10, 2016, Accepted: September 6, 2016

Abstract: A use of an electric outlet by a consumer forces the outlet manager to pay for the consumer's power usage in current electrical power systems. Even if a consumer uses an outlet managed by another person, one bill for both indoor and outdoor charging information should be required to the consumer in their contract with the utility company. For this purpose, we define a model for the Smart Grid security and propose a *Secure Payment Protocol for Charging Information over Smart grid*, *SPaCIS* for short, as a protocol satisfying the model. Our model provides for the unlinkability of consumers as well as for the undeniability and unforgeability of billing information using digital signatures and identity federations. SPaCIS is also efficient in the sense that time complexity is constant relatively to a trivial use such as an individual verification for each signatures, unless a verification error happens. We furthermore evaluate performance of SPaCIS via cryptographic implementation, and simulate SPaCIS in a case that one thousand users generate thirty signatures. Then, we show that SPaCIS with ECDSA can be executed within 6.30 msec for signing and 21.04 msec for verification of signatures, and conclude that SPaCIS is fairly practical.

Keywords: electricity charging information, smart grid security, identity federation, digital signature, privacy preserving

1. Introduction

1.1 Motivation

Recently, demand for electricity use outside the home has increased with the spread of mobile device use outside the home. In current electrical power systems, a manager who manages outlets under a smart meter pays the electric bill provided through the outlet. For instance, in public places, e.g., in a restaurant, the manager pays the electric bill even if the electricity was consumed by a visitor. By virtue of developments of IoT technology and smart appliances, large devices such as electric vehicles which are ubiquitous but require a measurable amount of power may be recharged in coming decades. In such a scenario, the use of power for a long period of time may create a financial burden on the manager. Hence, consumers should directly pay the electrical bill issued for the outlet that they used. As described later, the Smart Grid we focus on enables the manager to charge the electric bill to consumers. This means that, from a different standpoint, the consumers are able to unify the management for the electric bill outside/inside the home. Meanwhile, information about the electricity usage and its billing of consumers contains their sensitive private information and therefore protecting these information is a quite important problem.

We describe intuitive use by the Smart Grid in the following discussion. The Smart Grid is an important technology using a power grid. The Smart Grid, rolled out by the National Institute of Standards and Technology (NIST) [1], is a next generation power system which allows consumers and electric utilities to communicate interactively. Compared with legacy power systems, the Smart Grid has new energy management capabilities, such as an advanced metering infrastructure (AMI) [2] and a demand response [3]. AMI is an architecture for automated, two-way communication between a consumer with an IP address and an electric utility. Also, demand response controls the peak of electricity demand for power saving by consumers can stop the generation of electricity from the power utilities when electricity is tight. The Smart Grid is expected to greatly enhance the efficiency and reliability of future power systems with high-speed and two-way communication technologies.

However, according to Khurana et al. [4], security issues of the Smart Grid must first be addressed. More specifically, it must be ensured that the appropriate user is accessing accurate data created by the right device at the expected location at the appropriate time, using the expected protocol, without modifying the data. These requirements are important because user's privacy information can be derived from the data. For example, electricity use patterns could lead to disclosure of not only how much energy customers use but also when they are at home, at work, or traveling. Thus, a protocol guaranteeing both the validity of the data and the privacy of the consumers is crucial.

This paper is the full version of ISITA2014 [5]. In ISITA2014, we proposed the Basic Scheme for electricity charges by utilizing a federated identity. In this work, we found lack of the undenia-

[1] Osaka University, Suita, Osaka 565–0871, Japan
[2] National Institute of Technology, Nara College, Yamatokoriyama, Nara 639–1080, Japan
[a) hikaru.kishimoto@ist.osaka-u.ac.jp
[b) yanai@ist.osaka-u.ac.jp
[c) okamura@info.nara-k.ac.jp

bility of the Basic Scheme and revise the scheme to overcome the weakness. We also improve the efficiency of the scheme in terms of signing and verification of signatures. Moreover, we implement kernel functions of SPaCIS and evaluate their performance.

1.2 Contribution

In this work, we propose a *Secure Payment Protocol for Charging Information over Smart grid* (*SPaCIS*). SPaCIS is a secure consolidation protocol which guarantees the validity of the charging information and the undeniability of the consumer billing. For this purpose, we also define the power grid model and the security requirements for the electric bill required to the consumers. SPaCIS utilizes digital signatures and ID federation services as building blocks. The digital signatures are used for the validity while the ID federation is useful for the outdoor electricity use of consumers to be authenticated. More specifically, in SPaCIS, a manager authenticates each user by using a federated identity, and the signatures are generated by the manager and the consumers to guarantee the trustworthiness of the amount of electricity. We also discuss the security under the model. Our proposed protocol is also more efficient than a trivial use such as an individual verification for each signatures: in particular, whereas time complexity of the trivial construction of signatures is linear with respect to the number of signatures, our protocol decreases that to the constant size as long as a verification algorithm outputs accept. This improvement in efficiency is given by our technical contribution whereby generated signatures are introduced into hash functions. We also evaluate the performance of SPaCIS via cryptographic implementation, and simulate the case where the number of signers is one thousand and each signer generates a single signature in the same time. The case is supposed as an example of the electricity use of a small restaurant during a month and, in this case, SPaCIS is executable within 6.30 msec with ECDSA and within 98.00 msec for signing. Similarly, all of the signatures can be verified within 6.23 msec with RSA and within 21.04 with ECDSA. We believe practical performance is achieved since all operations for the 1,000 signers are finished within 1 second.

The rest of this paper is described as follows. In Section 2.2, we describe related works. The requirements and design principles for SPaCIS are given in Section 3. We present SPaCIS in Section 4. The security and the efficiency of SPaCIS are discussed in Section 5. Finally, our conclusions are given in Section 6.

2. Preliminaries

In this section, we briefly describe ID federation, which is a building block of SPaCIS, and related works about security protocols for the Smart Grid.

2.1 ID Federation

A Web service provider distinguishes between users by using a user identity and password, etc., which provides Web services such as social network service suitable for each user and prevents malicious users. Therefore, a user has to manage many identities and passwords suitable for each service provider. Federated identity management, for example, is a set of technologies and processes that let computer systems dynamically distribute identity information and delegate identification tasks across security domains [6]. There are several types of federated identity protocols, e.g., Security Assertion Markup Language (SAML) [7], the OpenID specification [8], [9]. All these protocols involve the following three components: A *user* is a person who uses several services. The *service provider* (SP) is a party who provides a service but offloads authentication to a third party. The *identity provider* (IdP) is a party who manages identities and the attributes of users. The IdP authenticates users and provides the attributes of the users to the SP. User privacy from the SP is maintained by the IdP assigning a pseudo-random name, called the name identifier, to the user. We utilize these tools as building blocks.

2.2 Related Works

2.2.1 Smart Grid Security Issues

Some solutions are proposed in this paper to the challenges presented by authentication and encryption for a smart grid network. According to Khurana et al. [4], authentication technologies for the Smart Grid networks have strict real-time constraints, e.g., multicast messages must be delivered in less than 4 msec. Wang et al. [10] have developed such an authentication solution by leveraging a one-time signature and one-way hash chain cryptographic constructs. Tsang et al. [11] have developed a low-latency bump-in-the-wire solution for authentication for legacy SCADA devices. They convert random-error detection available on legacy systems into a mechanism that guarantees data authenticity and freshness. A wide-area measurement system uses GPS-clock-synchronized fine-grained power grid measurements to provide increased stability and reliability. It is important to securely share the measurements among power grid entities over wide area networks. Bobba et al. [12] have leveraged the presence of trusted third parties to design a mediated policy-based encryption system that protects the secrecy of data and policies while releasing them to the authorized entities. Moreover, Niwa et al. [13] have developed a portable key generation center for identity-based encryption, and suggest its use to provide users' privacy. Well-known specifications can already be found in the Open Smart Grid Protocol (OSGP) [14] published by the European Telecommunication Standards Institute (ETSI). Jovanovic et al. [15] have analyzed the encryption scheme for OSGP, and show that OSGP has a vulnerable encryption technology. Although these studies have discussed the security for Smart Grid, our work is different from them in the sense of constructing a scheme to guarantee both the privacy and validity of charging information.

2.2.2 Privacy-Preserving Billing Protocol

One of the first privacy-preserving billing protocol (PBP) for smart metering has been proposed by Rial and Denezis [16] and has been enhanced and applied to different contexts. Jawurek et al. [17] have proposed a privacy-preserving billing, PBP for a short, protocol for the charging information of the Smart Grid. Whereas their protocol consists of homomorphic commitments [18] whose cost is heavy, our protocol consists of ordinary digital signatures. The PBP is based on Pedersen Commitments and Zero-Knowledge Proof to provide verification. Also in Ref. [19], Zero-Knowledge Proof is used in a billing protocol specifically optimized for low consumption. Then, Armand et al. [20] analyzed the security of the PBP via a formal method. They have utilized SATMC as a model checker, and we consider that the security of SPaCIS can be analyzed via a similar approach.

3. Threat Model

In this section, we define the *power grid model* for the Smart Grid and its security. This model is an instance of Smart Grid. In this model, Smart Grid provides capabilities for measuring of the amount in charge for each outlet, that for actual time and transition of the amount in charge to other entities over IP communication. Moreover, communication between each entity adopt IP communication. These specifications comply with the guideline published by NIST [1], which declares an investment standard for Smart Grid. First, we describe participants of the model and their ground rules. Then, we describe several assumptions of the participants and define the security requirements.

3.1 Participants and Their Ground Rules

Our model includes the following participants:

Consumer: A consumer is a person who only uses electricity. Let C be a consumer.

Manager: A manager is a person who manages outlets. Here, we define the outlets as interfaces for the use of electricity which is not managed by consumers. Let $\mathcal{M}$ be a manager.

Electric Utility: An electric utility is a utility which provides electricity. While it provides electricity to both the consumer and the manager, there are two types of electric utilities, i.e., for the consumer and for the manager. Let $\mathcal{U}$ be an electric utility, and let $\mathcal{U}_C$, $\mathcal{U}_M$ be the utility which provides electricity to C or $\mathcal{M}$, respectively.

Power Grid: A power grid is an organization that provides an infrastructure for all the entities. Namely, it is a single entity and the other entities are connected via the power grid.

The C has to pay the electric bill if C uses electricity in their home. Similarly, $\mathcal{M}$ has to pay the electric bill if $\mathcal{M}$ uses electricity in their home as a consumer. Suppose that C want to use electricity in the outdoor as well as in their home. Then, $\mathcal{M}$ allows consumer to use their electric resource provided by $\mathcal{U}_M$.

When a consumer C starts to use electricity through an outlet under $\mathcal{M}$, C gives information about its electric utility $\mathcal{U}_C$ to $\mathcal{M}$. This information is stored in an IC card issued by $\mathcal{U}_C$. Then, $\mathcal{M}$ retrieves the electricity utilized by C from $\mathcal{U}_C$. Each record of charging information by C at home and outside is consolidated to $\mathcal{U}_C$. C pays the electric bill only to $\mathcal{U}_C$.

C can utilize $\mathcal{M}$'s outlets by forwarding their identifier given from $\mathcal{U}_C$. $\mathcal{M}$ sends the amount in charge for C's use of outlets. Then, $\mathcal{U}_C$ makes settlements for the amount in charge by C. We call the model described above the power grid model and show its overview in **Fig. 1**. In this study, we deal with the flow of charging information when consumer uses electricity outside in Smart Grid. It is assumed that the flow of billing information conforms to the current specification [1] for assessing the amount in charge and calculating fee.

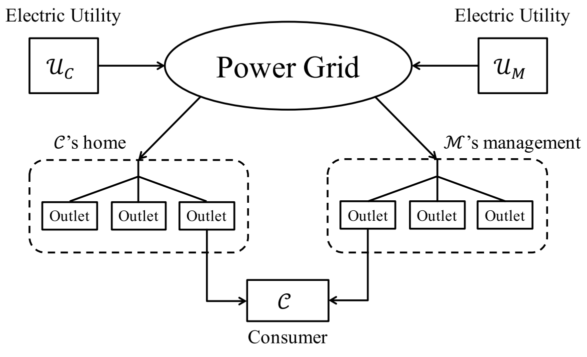


Fig. 1 Power grid model.

3.2 Assumption

We describe several assumptions in our model. A trustworthy utility is registered in the power grid and then is assigned an identifier ID_U from the power grid. We also assume that utilities are trusted: more specifically, we assume that the utilities $\mathcal{U}_C$ and $\mathcal{U}_M$ do not collude with each other and do not fail any operation such as computation of the electricity bill. In general, these utilities are public domains authorized by a government, but used publicly under governmental supervision.

We also assume the existence of a public key infrastructure for the power grid, and let $\mathcal{U}$ and $\mathcal{M}$ each have their own public key and secret key. These keys and their certificates are issued from a certificate authority on the public key infrastructure, and anyone can obtain public keys and certifications through the power grid. $\mathcal{M}$ assigns a unique number to each charging information by C. C is assigned a different pseudo-random identifier for each $\mathcal{M}$ from $\mathcal{U}_C$. This identifier is a unique identifier that $\mathcal{M}$ can identify C, and C does not have any other identifier such as an IP address.

From the perspective of the model, each entity has the following information:

Consumer: A consumer has its own identifier C given from a consumer utility $\mathcal{U}_C$, its secret key SK_C, its corresponding public key PK_C, an IC card with an identifier of $\mathcal{U}_C$, and charging information.

Manager: A manager has its own identifier $\mathcal{M}$ given from an outlet utility $\mathcal{U}_M$, its secret key SK_M, its corresponding public key PK_M, an identifier of $\mathcal{U}_M$, and an identifier of a utility $\mathcal{U}_C$ given from a connecting consumer. $\mathcal{M}$ measures the amount in charge used by C under $\mathcal{M}$. Capability of measurement of the amount in charge depends on capability of devices, and an approach to measuring the charge is out of the scope of this work. In other words, we assume to use devices that can measure small quantities of electricity.

Electric Utility: An electric utility is a trusted entity which is not corrupted and does not falsify any task has its own identifier $\mathcal{U}$, its secret key SK_U, its corresponding public key PK_U, identifiers for registered consumers, identifiers for registered managers, and charging information for each consumer.

Power Grid: The power grid has public information for each entity and their identifiers. In this study, we assume a secure communication channel. A secure communication channel is not eavesdropped and falsified by any third party. Since C and $\mathcal{M}$ can obtain their public keys, they can also authenticate each other to connect securely. Thus, we assume that the man in the middle attack between $\mathcal{C}$ and $\mathcal{M}$ never occurs.

3.3 Requirements

Consumers want to obtain the date, location and amount of consumption for each electricity usage as the charging information, whereas consumers want to conceal the information from $\mathcal{M}$ as much as possible. Moreover, each record of charging information must not be falsified since the electric bill depends on the charging information. Hence, the following requirements are necessary for the power grid model. We also note that it is unavoidable that $\mathcal{M}$, which directly connects with $\mathcal{C}$ to provide the electricity, knows the information of the charges. However, it is necessary to conceal the charging information from other $\mathcal{M}$s.

In this study, we assume an honest-but-curious adversary where each entity complies with the processes of a protocol yet tries to learn something that is outside the bounds of its own information.

Unlinkability of Consumer: An adversary of this requirement is a manager $\tilde{\mathcal{M}}$ who can interact with $\mathcal{C}$ independently of other managers. We assume that $\tilde{\mathcal{M}}$ does not collude with an honest manager $\mathcal{M}$ which $\mathcal{C}$ connects with. Likewise, $\tilde{\mathcal{M}}$ does not collude with utility $\mathcal{U}_C$ since $\mathcal{U}_C$ always knows the charging information of $\mathcal{C}$. $\tilde{\mathcal{M}}$'s advantage is to obtain PK_C, an identifier $\mathcal{U}_C$ of $\mathcal{C}$'s utility, a public key PK_M of the honest manager $\mathcal{M}$, an identifier $\mathcal{U}_M$ of $\mathcal{M}$'s utility and any information given by providing its own outlet to $\mathcal{C}$. Since any other entities cannot obtain more information than $\tilde{\mathcal{M}}$, the main target of the requirement is $\tilde{\mathcal{M}}$. $\tilde{\mathcal{M}}$'s goal is to output a tuple of $\mathcal{C}$'s identifiers under honest manager $\mathcal{M}$, who does not collude with $\tilde{\mathcal{M}}$, and that under $\tilde{\mathcal{M}}$. We say a scheme is unlinkable if $\tilde{\mathcal{M}}$ cannot output the tuple of $\mathcal{C}$'s identifiers under $\mathcal{M}$ and $\tilde{\mathcal{M}}$.

Undeniability of Consumer: An adversary of this requirement is a malicious $\mathcal{C}$ who can interact with $\mathcal{M}$. We say that $\mathcal{C}$ denies the charging information if $\mathcal{C}$ outputs different charging information, which is acceptable in $\mathcal{U}_C$, from that output by $\mathcal{C}$ in the past under interactions with $\mathcal{M}$. We note that how to decide the amount of payment for the charging information is managed by $\mathcal{U}_C$, and $\mathcal{C}$ cannot change the rate by itself. Namely, we assume that $\mathcal{C}$ cannot collude with $\mathcal{U}_C$. Likewise we also suggest that $\mathcal{C}$ do not collude with manager $\mathcal{M}$ since their collusion could allow the possibility that $\mathcal{C}$ may also pay the electric bill for $\mathcal{M}$. Hence, $\mathcal{C}$'s advantage is to obtain its own information, public keys PK_M, PK_{U_C} and PK_{U_M}, and identifiers of $\mathcal{U}_C$ and $\mathcal{U}_M$. $\mathcal{C}$'s goal is to deny the charging information under the above advantage, and we say that a scheme is undeniable for consumer if $\mathcal{C}$ cannot deny the charging information.

Undeniability of Manager: This requirement is almost the same as the previous requirements, but an adversary is a malicious manager $\mathcal{M}$ instead of $\mathcal{C}$. We say that $\mathcal{M}$ denies the charging information if $\mathcal{M}$ outputs different charging information, which is acceptable in $\mathcal{U}_M$, from that output by $\mathcal{M}$ in the past under interactions with $\mathcal{C}$. We assume that $\mathcal{M}$ cannot change the rate in charging information by itself, and that $\mathcal{M}$ cannot collude with $\mathcal{U}_C$. Likewise, we assume that they do not collude with the consumer since their collusion brings the possibility that $\mathcal{M}$ may also pay the electric bill for $\mathcal{C}$. Hence, $\mathcal{M}$'s advantage is to obtain its own information, public keys PK_C, PK_{U_C} and PK_{U_M}, and identifiers of $\mathcal{U}_C$ and $\mathcal{U}_M$. $\mathcal{M}$'s goal is to deny the charging information under the above advantage, and we say that a scheme is undeniable for manager if $\mathcal{M}$ cannot deny the charging information.

Unforgeability: This requirement is for the validity of the charging information of a consumer. The charging information must not be manipulated. An adversary of the requirement is all entities except for the consumer-self, and $\mathcal{U}_C$ and $\mathcal{M}$ can collude with each other only in the requirement. $\mathcal{C}$ utilizes an outlet under its agreement and pays the electric bill. Namely, its main scenario is whether the other entities can generate any charging information of $\mathcal{C}$ without $\mathcal{C}$'s agreement or not.

4. SPaCIS: Secure Payment Protocol for Charging Information over Smart grid

4.1 Design Principle

An outline of SPaCIS is shown in **Fig. 2**. $\mathcal{M}$ has to authenticate each user for billing validity. There is the possibility that $\mathcal{C}$ uses several outlets under different managers. Therefore, we adopt an identity federation scheme where a consumer's requests is redirected to its utilities with an identifier of the consumer. In order to utilize the identify federation scheme, we also assume that a resolver exists who answers the address of a requested electric utility. The resolver is an entity who does not have any authority and cannot provide electricity. Let $\mathcal{R}$ be a resolver. In SPaCIS, we suppose that $\mathcal{C}$ can use any outlet with the same single identity for the identity federation. Here, we assume that, when $\mathcal{M}$ redirects $\mathcal{C}$ to $\mathcal{U}_C$, $\mathcal{R}$ does not leak any information except for utilities' addresses and can be an address server controlled under $\mathcal{M}$. Namely, we recommend the utilization of an identity federation scheme with high-level security to implement our scheme. The roles of user, SP and IdP are assigned to $\mathcal{C}$, $\mathcal{M}$ and $\mathcal{U}$, respectively.

$\mathcal{C}$ and $\mathcal{M}$ generate digital signatures to prevent falsification of charging information. In our previous scheme proposed in ISITA2014 [5], called the Basic Scheme, $\mathcal{C}$ and $\mathcal{M}$ sign each use of electricity. However, this scheme has a problem because the number of signatures increases in proportion to the use of electricity. Hence, a verifier has to verify all the signatures. Namely, its time complexity is linear with respect to the number of signatures. Therefore, we propose a new construction of signatures to reduce time complexity. In the new construction, a hashed digest including all the previously generated signatures as input is given as a part of a message to be signed newly. Thus, the verifier can treat only a single signature for the whole charging information, if its verification is accepted. Moreover, we found a fact that the Basic Scheme is insecure with respect to the undeniability.

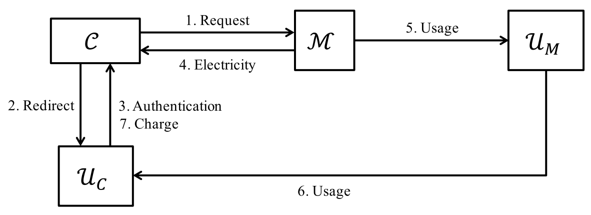


Fig. 2 Outline of SPaCIS.

In particular, C can deny the use of him/her-self because no one guarantees and checks C's signatures. Therefore, C can break the scheme by outputting different signatures from ones to be truly signed. In SPaCIS, we improve the construction to overcome this problem. That is, $\mathcal{M}$ in SPaCIS generates signatures with respect to C's signatures after C signs the C's usage measured by $\mathcal{M}$. Hence, if C tries to deny the use of electricity, $\mathcal{M}$ can prevent the denial by C by outputting the signatures received from C. In this case, $\mathcal{M}$ restarts the protocol from the phase where $\mathcal{M}$ generates signatures, i.e., the beginning of the Closing Phase described later. When multiple Cs try to deny, $\mathcal{M}$ has to execute the same process for each C. $\mathcal{M}$ then needs the computational cost of the order $O(qn)$, where n is the number of Cs who try to deny and q is the total cost for generating signatures from $\mathcal{M}$ to C, verifying the signatures by C and verifying signatures from C to $\mathcal{M}$. On the one hand, the size of required storage is not increased even if Cs try to deny because $\mathcal{M}$ can discard the generated signatures before the restart. Note that C tries to deny the electricity usage even if $\mathcal{M}$'s output is correct, $\mathcal{M}$ is forced to execute the same processes. Hence, in this case, $\mathcal{M}$ should terminate the process at the step (3) in the Closing Phase described later, and does not provide more service to C. Suppose that the payment is then executed outside the scope of the protocol. This construction can preserve the undeniability of $\mathcal{M}$ since C receives $\mathcal{M}$'s signatures in the end of the use and utilizes them as a part of messages in the next phase.

SPaCIS consists of four phases: the Preparation Phase, Start Phase, Closing Phase and the Settlement Phase. The Preparation Phase is performed only once. The Start Phase and the Closing Phase are performed every time C uses outside electricity. These phases are necessary to ensure the undeniability of C. The Start Phase checks whether a user who uses the electricity under the $\mathcal{M}$'s environment is valid or not. On the other hand, the Closing Phase includes verification of signatures generated by C in order for $\mathcal{M}$ to check whether C generates valid signatures or not. Removing these processes potentially enables C to manipulate the electricity usage or send the invalid electricity bill to $\mathcal{U}_C$. The Settlement Phase is performed periodically. A signing function is denoted as $Sign(m, SK)$ which means a signer generates a signature σ on a message m by using their secret key SK. A function of a signature verification is denoted as $F = Verify(m, \sigma, PK)$ which means signature σ on m is verified using a public key PK. $F = accept$ if signature σ is the correct signature for m, $F = false$ if not.

4.2 Construction

In this section, we describe the proposed scheme. Note that the following description assumes a certain $\mathcal{M}$ and C. In other words, each parameter described below is independent for each pair of $\mathcal{M}$ and C which executes the scheme, and its index is a local parameter between them. We also define registration information, which is registered to $\mathcal{U}_C$, as a tuple of $(ID_C, PK_M, ID_P, PK_P, x_i, i)$, where ID_C is an identifier of C, PK_M is a public key of $\mathcal{M}$, and (ID_P, PK_P) is a pair of an identifier and a public key which are assigned from IdP. x_i is a value such that 6th value is max, if a pair of ID_C and PK_M is registered to the registration information described above. If the pair of ID_C and PK_M is not registered, x_i is null. x_i is a hash value generated in the Closing Phase of the C's i-th use at $\mathcal{M}$'s outlets.

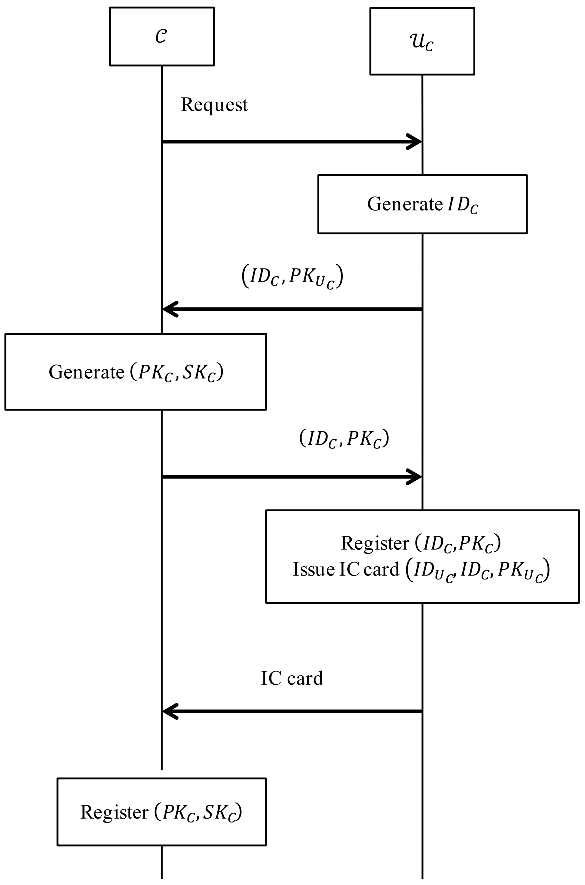


Fig. 3 Preparation phase.

4.2.1 Preparation Phase

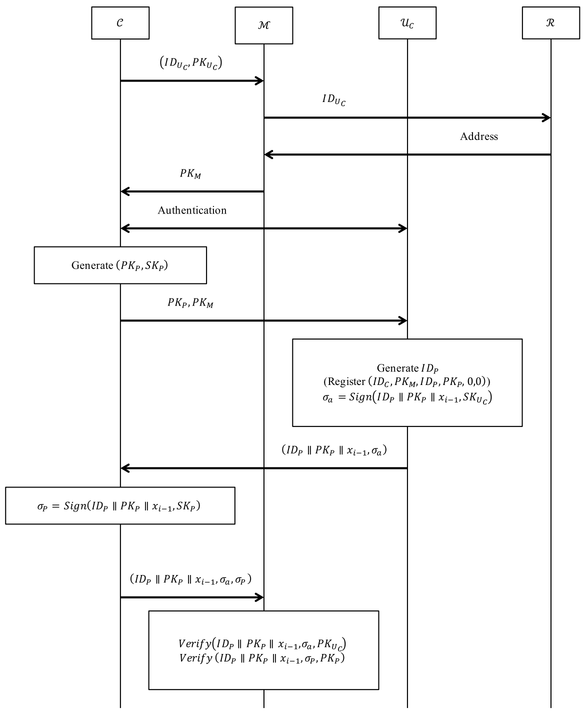
In this phase, an electric utility $\mathcal{U}_C$ issues an IC card which is necessary for user authentication of a consumer C. The Preparation Phase is shown in **Fig. 3**. We describe detail of Preparation Phase below.

(1) C sends a request for registration to $\mathcal{U}_C$.
(2) $\mathcal{U}_C$ generates an identifier ID_C for C and sends (ID_C, PK_{U_C}) to C.
(3) C generates C's public key PK_C and secret key SK_C, and sends (ID_C, PK_C) to $\mathcal{U}_C$.
(4) $\mathcal{U}_C$ stores (ID_C, PK_C) in $\mathcal{U}_C$'s database as a registration.
(5) $\mathcal{U}_C$ stores $(ID_{U_C}, ID_C, PK_{U_C})$ in an IC card and sends the card to C.
(6) C registers its keys, PK_C and SK_C, to the IC card.

4.2.2 Start Phase

This phase is executed when C starts to use electricity through an outlet managed by $\mathcal{M}$. $\mathcal{M}$ authenticates C using an identity federation scheme where $\mathcal{U}_C$ is the IdP and assigns a pseudo-random identifier ID_P as the name identifier to C. Furthermore, C generates a pair of public keys PK_P and secret keys SK_P. The ID_P and these keys are changed for each $\mathcal{M}$. A protocol is depicted for C's i-th use of $\mathcal{M}$'s outlets in **Fig. 4**. We describe details of the Start Phase below.

(1) C sends (ID_{U_C}, PK_{U_C}) to $\mathcal{M}$.
(2) $\mathcal{M}$ sends ID_{U_C} to $\mathcal{R}$ and receives an address of $\mathcal{U}_C$ from $\mathcal{R}$.



$\mathcal{C}$
$\mathcal{M}$
$\mathcal{U}_C$
$\mathcal{R}$
(ID_{U_C}, PK_{U_C})
ID_{U_C}
Address
PK_M
Authentication
Generate (PK_P, SK_P)
PK_P, PK_M
Generate ID_P
(Register $(ID_C, PK_M, ID_P, PK_P, 0, 0)$)
$\sigma_a = Sign(ID_P \parallel PK_P \parallel x_{i-1}, SK_{U_C})$
$(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_a)$
$\sigma_P = Sign(ID_P \parallel PK_P \parallel x_{i-1}, SK_P)$
$(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_a, \sigma_P)$
$Verify(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_a, PK_{U_C})$
$Verify(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_P, PK_P)$

Fig. 4 Start phase.

(3) $\mathcal{M}$ redirects C to $\mathcal{U}_C$ for authentication after $\mathcal{M}$ gives public key PK_M to C.
(4) C and $\mathcal{U}_C$ authenticate mutually.
(5) C generates a pair of public keys PK_P and secret keys SK_P, and sends PK_P and PK_M given by $\mathcal{M}$ to $\mathcal{U}_C$ by encrypting with PK_{U_C}.
(6) $\mathcal{U}_C$ retrieves the registration information $(ID_C, PK_M, ID_P, PK_P, x_{i-1}, i-1)$ at the previous use of the electricity under $\mathcal{M}$. $\mathcal{U}_C$ generates a signature $\sigma_a = Sign(ID_P \parallel PK_P \parallel x_{i-1}, SK_{U_C})$. Here x_{i-1} is included in registration information $(ID_C, PK_M, ID_P, PK_P, x_{i-1}, i-1)$. If C is the first use of the electricity under $\mathcal{M}$'s environment, x_{i-1} is null.
(7) $\mathcal{U}_C$ sends $(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_a)$ to C.
(8) C generates a signature $\sigma_P = Sign(ID_P \parallel PK_P \parallel x_{i-1}, SK_P)$ and sends $(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_a, \sigma_P)$ to $\mathcal{M}$.
(9) $\mathcal{M}$ verifies σ_a and σ_P with $Verify(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_a, PK_{U_C})$ and $Verify(ID_P \parallel PK_P \parallel x_{i-1}, \sigma_P, PK_P)$ respectively. If the verification is passed, $\mathcal{M}$ starts to supply electricity to C.

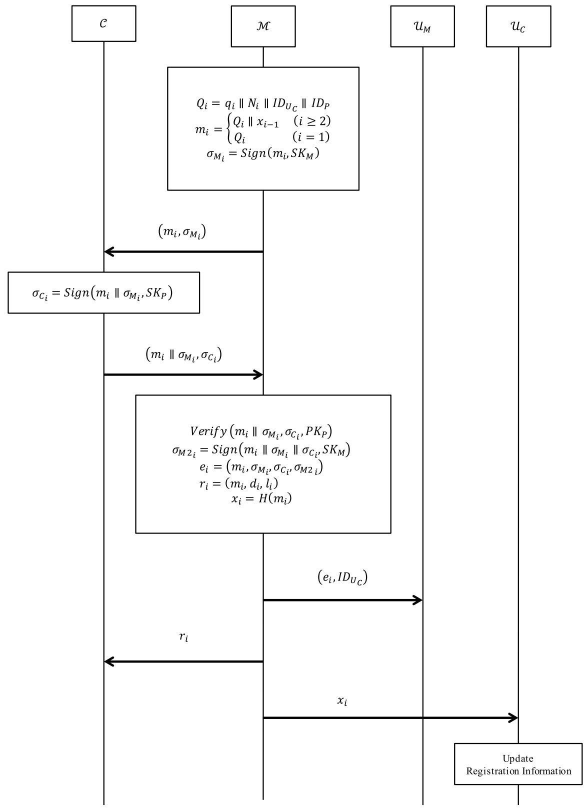
4.2.3 Closing Phase

This phase is executed when C finishes using electricity. In this phase, a message m_i generated by $\mathcal{M}$ at the C's i-th use is defined as follows.

$$m_i = \begin{cases} Q_i \parallel x_{i-1} & (i \geq 2) \\ Q_i & (i = 1) \end{cases} \tag{1}$$

where $Q_i = q_i \parallel N_i \parallel ID_{U_C} \parallel ID_P$, q_i is the amount in charging information, N_i is a unique number, $x_i = H(m_i)$ and H is a hash function[*1]. x_i is calculated and sent to $\mathcal{U}_C$ by $\mathcal{M}$ after i-th use of C. $\mathcal{U}_C$ sends x_i to C as C's attribute information in the $i+1$-th Start Phase. The protocol is illustrated in **Fig. 5**. We describe details of the Closing Phase below.

[*1] $\mathcal{M}$ uses digest of charging information because sending a m_i requires more cost.



$\mathcal{C}$
$\mathcal{M}$
$\mathcal{U}_M$
$\mathcal{U}_C$
$Q_i = q_i \parallel N_i \parallel ID_{U_C} \parallel ID_P$
$m_i = \begin{cases} Q_i \parallel x_{i-1} & (i \geq 2) \\ Q_i & (i = 1) \end{cases}$
$\sigma_{M_i} = Sign(m_i, SK_M)$
(m_i, σ_{M_i})
$\sigma_{C_i} = Sign(m_i \parallel \sigma_{M_i}, SK_P)$
$(m_i \parallel \sigma_{M_i}, \sigma_{C_i})$
$Verify(m_i \parallel \sigma_{M_i}, \sigma_{C_i}, PK_P)$
$\sigma_{M2_i} = Sign(m_i \parallel \sigma_{M_i} \parallel \sigma_{C_i}, SK_M)$
$e_i = (m_i, \sigma_{M_i}, \sigma_{C_i}, \sigma_{M2_i})$
$r_i = (m_i, d_i, l_i)$
$x_i = H(m_i)$
(e_i, ID_{U_C})
r_i
x_i
Update Registration Information

Fig. 5 Closing phase.

(1) $\mathcal{M}$ generates a unique number N_i, a data $Q_i = q_i \parallel N_i \parallel ID_{U_C} \parallel ID_P$ and a message m_i defined in Eq. (1) where q_i is the amount of this charging information, and a signature $\sigma_{M_i} = Sign(m_i, SK_M)$. $\mathcal{M}$ sends (m_i, σ_{M_i}) to C.
(2) C checks the validity of m_i. If the check is passed, C generates a signature $\sigma_{C_i} = Sign(m_i \parallel \sigma_{M_i}, SK_P)$ and sends $(m_i \parallel \sigma_{M_i}, \sigma_{C_i})$ to $\mathcal{M}$.
(3) $\mathcal{M}$ verifies σ_{C_i} with $Verify(m_i \parallel \sigma_{M_i}, \sigma_{C_i}, PK_P)$. If the verification passes, $\mathcal{M}$ generates a signature $\sigma_{M2_i} = Sign(m_i \parallel \sigma_{M_i} \parallel \sigma_{C_i}, SK_M)$ and sends $e_i = (m_i, \sigma_{M_i}, \sigma_{C_i}, \sigma_{M2_i})$ and ID_{U_C} to $\mathcal{U}_M$. Otherwise, $\mathcal{M}$ discards the signatures and then terminates the process or restarts the process from the step (1).
(4) $\mathcal{M}$ generates charging information $r_i = (q_i, d_i, l_i)$ where d_i is the date of the amount and l_i is the location, and sends the r_i to C. Furthermore, $\mathcal{M}$ calculates the hash value of charging information $x_i = H(m_i)$ and sends (PK_M, ID_P, x_i) to $\mathcal{U}_C$.
(5) $\mathcal{U}_C$ registers the registration information $(ID_C, PK_M, ID_P, PK_P, x_i, i)$ to a list.

4.2.4 Settlement Phase

In this phase, C settles the bills for electricity. This phase is executed periodically, e.g., monthly. $\mathcal{U}_M$ sends the set of charging information $(e_1, e_2, ..., e_n)$ to $\mathcal{U}_C$. For $1 \leq i \leq n-1$, $\mathcal{U}_C$ calculates $x'_i = H(m_i)$, where m_i is included in e_i, and checks if $x_i = x'_i$ holds, where x_i is included in e_{i+1}. $\mathcal{U}_C$ also verifies σ_{C_n} and σ_{M2_n} with $Verify(m_n \parallel \sigma_{M_n}, \sigma_{C_n}, PK_P)$ and $Verify(m_n \parallel \sigma_{M_n} \parallel \sigma_{C_n}, \sigma_{M2_n}, PK_M)$. The amount of charging information, the sum of each q_i is regarded as correct if and only if both verification functions output *accept*. If a verification function outputs *false*,

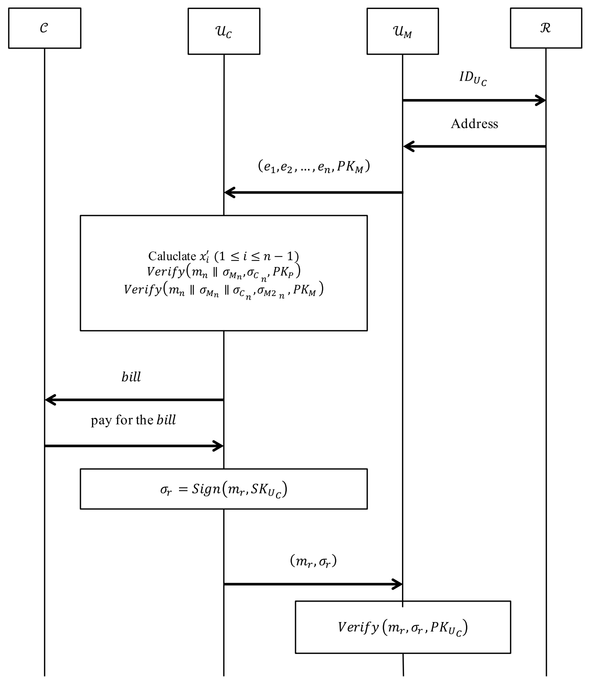

$\mathcal{C}$
$\mathcal{U}_C$
$\mathcal{U}_M$
$\mathcal{R}$
ID_{U_C}
Address
$(e_1, e_2, \ldots, e_n, PK_M)$
Caluclate x'_i $(1 \le i \le n-1)$
$Verify(m_n \parallel \sigma_{M_n}, \sigma_{C_n}, PK_P)$
$Verify(m_n \parallel \sigma_{M_n} \parallel \sigma_{C_n}, \sigma_{M2_n}, PK_M)$
$bill$
pay for the $bill$
$\sigma_r = Sign(m_r, SK_{U_C})$
(m_r, σ_r)
$Verify(m_r, \sigma_r, PK_{U_C})$

Fig. 6 Settlement phase.

$\mathcal{U}_C$ tracks the falsification by verifying all signatures. A Settlement Phase is shown in **Fig. 6**. We describe detail of Settlement Phase below.

(1) $\mathcal{U}_M$ sends ID_{U_C} to $\mathcal{R}$ and receives an address of $\mathcal{U}_C$ from $\mathcal{R}$.
(2) $\mathcal{U}_M$ sends $e_1, e_2, ..., e_n$ and PK_M to $\mathcal{U}_C$.
(3) For each e, $\mathcal{U}_C$ identifies what the number of phases are between C and $\mathcal{M}$ by searching the registered information of $\mathcal{U}_C$ with ID_P and calculates hash value x'_{n-1}. $\mathcal{U}_C$ verifies σ_{C_n} and σ_{M2_n} with $Verify(m_n \parallel \sigma_{M_n}, \sigma_{C_n}, PK_P)$ and $Verify(m_n \parallel \sigma_{M_n} \parallel \sigma_{C_n}, \sigma_{M2_n}, PK_M)$, respectively. $\mathcal{U}_C$ checks N_i in m_i whether N_i is not used in other messages m_i.
(4) $\mathcal{U}_C$ calculates a bill *bill* from each q_i in m_i and sends the *bill* to C.
(5) C pays for the *bill* to $\mathcal{U}_C$.
(6) $\mathcal{U}_C$ generates m_r where is a message that $\mathcal{U}_C$ received the bill for $\sum_{i=1}^{n} q_i$. $\mathcal{U}_C$ also generates a signature $\sigma_r = Sign(m_r, SK_{U_C})$ and sends (m_r, σ_r) to $\mathcal{U}_M$.
(7) $\mathcal{U}_M$ verifies σ_r with $Verify(m_r, \sigma_r, PK_{U_C})$. If the verification is passed, $\mathcal{U}_M$ charges to $\mathcal{M}$ for electricity after deducting amount of C's charges, i.e., $\sum_{i=1}^{n} q_i$, from $\mathcal{M}$'s amount.

5. Discussion

In this section, we analyze the security and the efficiency of SPaCIS. We show that the scheme satisfies the security requirements in Section 3.3, and then we evaluate the computational cost of SPaCIS.

5.1 Analysis of Satisfying Requirements

Unlinkability of Consumer: C has utilized a pseudo-random identifier to connect with $\mathcal{M}$. This identifier is independent for each $\mathcal{M}$ even if C is the same entity. Moreover, $\tilde{\mathcal{M}}$ cannot collude with $\mathcal{M}$ and cannot eavesdrop on communication between C and $\mathcal{M}$ by the definition of the model. Thus, $\tilde{\mathcal{M}}$ has to guess C's identifier under $\mathcal{M}$ by utilizing only $\tilde{\mathcal{M}}$'s own records of the communication with C. Since an identifier of C for each manager is generated via a pseudo-random identifier, the guess is equivalent to guessing the pseudo-randomness generated by $\mathcal{U}_C$. Therefore, C's identifiers are unlinkable for $\tilde{\mathcal{M}}$ unless $\tilde{\mathcal{M}}$ colludes with $\mathcal{M}$.

Undeniability of Consumer: In the proposed scheme, C has generated signatures via interaction with $\mathcal{M}$. In particular, in step (1) in Section 4.2.3, a message is signed by $\mathcal{M}$, and then C signs the signature by $\mathcal{M}$ as a part of a plaintext. This step means that the amount in charge by C can be also guaranteed by $\mathcal{M}$'s signature. Based on these statements, if C wants to deny their signatures, C needs $\mathcal{M}$'s signatures, which were received from $\mathcal{M}$ in the past and whose messages corresponds to topics $\mathcal{M}$ wants to deny, as a part of plaintexts. This process is difficult without colluding with $\mathcal{M}$, and such collusion is not allowed by the definition. Therefore, C has to generate $\mathcal{M}$'s signature by him-/herself. Since a digital signature scheme is unforgeable from the assumption, C cannot forge $\mathcal{M}$'s signatures. Thus, C cannot deny their signatures such that $\mathcal{U}_C$ accepts.

Undeniability of Manager: The discussion of the undeniability of a manager is almost the same as that of a consumer. In the step (2) in Section 4.2.3, C has signed $\mathcal{M}$'s signature as a part of a plaintext. This step means that the agreement on the amount in charge by $\mathcal{M}$ can be guaranteed by C's signature. Therefore, C's signatures, which are output by C in the past, are necessary for $\mathcal{M}$ to deny the charging information. This process is difficult without colluding with C, and such collusion is not allowed by the definition. Therefore, $\mathcal{M}$ has to generate C's signature by him-/herself. Since a digital signature scheme is unforgeable from the assumption, $\mathcal{M}$ cannot forge C's signatures. Thus, $\mathcal{M}$ cannot deny their signatures such that $\mathcal{U}_C$ accepts.

Unforgeability of Consumer: $\mathcal{U}_C$ calculates a bill for charging information from charging information and sends it to C. σ_r and *bill* are generated by $\mathcal{U}_C$ to proof of C's payment. $\mathcal{U}_M$ can check the amount in charges by C from the total amount provided through $\mathcal{M}$'s outlets. Hence, C can guarantee the charging information if the utilizing signature scheme is unforgeable.

5.2 Performance Evaluation

In this section, we evaluate the performance of SPaCIS. For SPaCIS, Preparation Phase, the signature verification and the signature generation in the Start Phase and the signature verification in the Closing Phase are almost the same as that for the Basic Scheme. Meanwhile, while the cost for a single execution of the signature verification and the signature generation is the same as that for the Basic Scheme, and the number of these operations for the Closing Phase and the Settlement Phase for SPaCIS is different from that for the Basic Scheme. Therefore, we implement the signing part of the Closing Phase and the signature-verification part of the Settlement Phase for both the Basic Scheme and SPaCIS, and measure their computational time. In this experiment, we utilize RSA signatures and ECDSA as digital signature schemes. The experimental environment is shown as **Table 1**. Here, digital signature schemes are based on Java SE Development Kit (JDK), and the key length is 3072 bit

Table 1 Experimental environment.

CPU	Intel Core i5 @2.7 GHz
Memory	8 GB
OS	Mac OS 10.11.2
Compiler	java version 1.8.0
Language	Java
Library	Java SE Development Kit

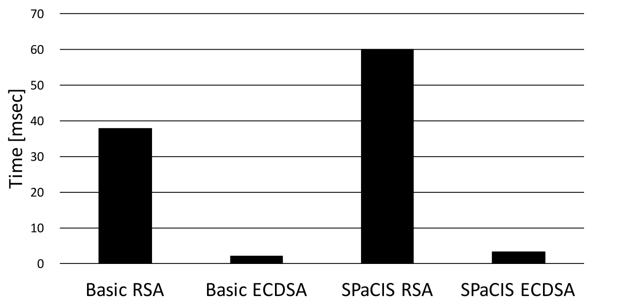


Fig. 7 Evaluation of Closing Phase: We implement the signing step of the Closing Phase. In this experiment, we measure the average time for signature generations at one hundred times.

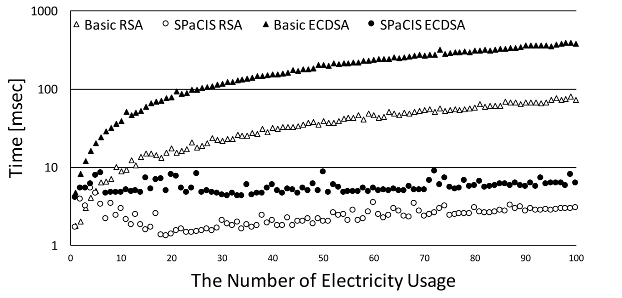


Fig. 8 Evaluation of Settlement Phase: We implement the verification steps of the Settlement Phase for SPaCIS and Basic Construction. In this experiment, we measure the computational time for signature verification. For both schemes, we measure the performance of construction with RSA signatures and that with ECDSA.

for RSA and 256 bit for ECDSA, respectively. Let a hash function be SHA-256.

We show the experimental result for generating signatures in the Closing Phase in **Fig. 7** and that for verifying signatures of RSA and ECDSA in both the Basic Scheme and the SPaCIS in **Fig. 8**.

In the Closing Phase, the Basic Scheme needs 37.83 msec for RSA and 2.12 msec for ECDSA while SPaCIS needs 60.06 msec for RSA and 3.21 msec for ECDSA. In the comparison between the Basic Scheme and the SPaCIS, the time for each scheme is the ratio 2:3 since the number of signing are 2:3 in the Basic Scheme and SPaCIS. Namely, the result measures up to the theoretical analysis. The computational delay of SPaCIS with ECDSA is almost 1 msec from that of the Basic Scheme, and this is negligible in the standpoint of users. Meanwhile, SPaCIS is quite faster in terms of the Settlement Phase as shown in Fig. 8. Namely, the computational cost of SPaCIS is the constant order. Although the number of computations of a hash function is linear, it is quite fast and the computational time is negligible as shown in these figures.

Here, we note the comparison between the signature schemes. ECDSA requires a greater computational cost for verification than generation of signatures in general. The verification of ECDSA is then slower than that of RSA as shown in Fig. 8

Table 2 Performance with Parallelization Case: In this experiment, we suppose that the number of signers are one thousand and each signer generates thirty signatures, i.e., one electricity use per a day during a month. In the Closing Phase, all of the signers singly executes the protocol in the same time. On the other hand, in the Settlement Phase, all of the signers executes the protocol with thirty signatures.

Phase (process)	Basic [msec]		SPaCIS [msec]	
	RSA	ECDSA	RSA	ECDSA
Closing (generation)	67.12	3.86	98.00	6.30
Settlement (verification)	42.11	1,858.00	6.23	21.04

whereas the generation of ECDSA is faster than that of RSA as shown in Fig. 7. The Basic Scheme is affected by this fact, and hence the difference between the signature schemes becomes large, e.g., about 300 msec for 100 times of the electricity usage. On the other hand, the difference between the signature schemes in the SPaCIS is quite small, e.g., about 3 msec in the same case. That is, SPaCIS strengthens an advantage of ECDSA, where the signature verification as a bottleneck can be shortened while the signature generation is still faster.

Moreover, we pick up several data in the both phase and re-implement the protocol with a parallelization case towards a practical use. That is, in the real world, multiple users may access to a single manager at the same time. The result is shown in **Table 2**. Finally, we show that SPaCIS can be executed within 1 sec even if both the Settlement Phase and the Closing Phase are executed.

Thus, SPaCIS with ECDSA is the best practice.

6. Conclusions

In this paper, we proposed SPaCIS as a secure consolidation protocol over the a Smart Grid. In SPaCIS, a manager authenticates each user using a federated identity, and signatures are generated by the manager and consumers in order to guarantee the undeniability and the unforgeability of the amount of electricity. An electric utility verifies these signatures and hence can confirm the validity for the charging information. Moreover, SPaCIS uses a hash function to include the previous signatures, and thus it is efficient for the number of verifications. We also discussed the security and showed the efficiency in comparison with the Basic Scheme. As a result, SPaCIS satisfies the security requirements and the number of verifications in SPaCIS is less than that in the Basic Scheme. Moreover, we partially implemented SPaCIS to evaluate its performance. Then, we showed that the computational delay of SPaCIS in the Closing Phase is quite small in comparison with the Basic Scheme while the computational time in the Settlement Phase is grossly improved. More specifically, we executed the experiment where one thousand signers execute SPaCIS with thirty signatures in the parallel use. In such a situation, the performance in the Closing Phase was 98.00 msec with RSA and 6.30 msec with ECDSA whereas that in the Settlement Phase was 6.23 msec with RSA and 21.04 msec with ECDSA, respectively. Thus, we conclude that SPaCIS is fairly practical. For future research, we plan to suppose a situation in which a consumer and a manager are provided electricity from different power grids. We also plan to analyze the security by the formal methods.

Acknowledgments We would like to thank the anonymous reviewers in ISITA 2014 and professor Toru Fujiwara in Osaka University for their valuable comments. We would also like to thank JSPS KAKENHI grant number 16K16065, which partially supported this work.

References

[1] National Institute of Standards and Technology: *NIST framework and roadmap for smart grid interoperability standards, release 1.0* (2010).

[2] Sui, H., Wang, H., Lu, M.-S. and Lee, W.-J.: An AMI System for the Deregulated Electricity Markets, *IEEE Trans. Industry Applications Society Annual Meeting, IAS '08*, pp.1–5 (2008).

[3] LeMay, M., Nelli, R., Gross, G. and Gunter, C.: An Integrated Architecture for Demand Response Communications and Control, *Hawaii International Conference on System Sciences, Proc. 41st Annual*, pp.174–174 (2008).

[4] Khurana, H., Hadley, M., Lu, N. and Frincke, D.: Smart-grid security issues, *IEEE Security Privacy*, Vol.8, No.1, pp.81–85 (2010).

[5] Kishimoto, H. and Okamura, S.: Secure consolidation of charging information over Smart Grid using ID federation, *2014 International Symposium on Information Theory and Its Applications* (*ISITA*), pp.226–230 (2014).

[6] Maler, E. and Reed, D.: The Venn of Identity: Options and Issues in Federated Identity Management, *IEEE Security Privacy*, Vol.6, No.2, pp.16–23 (2008).

[7] Oasis: Security Assertion Markup Language (SAML) V2.0 (2007), available from ⟨http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf⟩.

[8] OpenID Foundation: OpenID Authentication 2.0 - Final (2007), available from ⟨http://openid.net/specs/openid-authentication-2_0.html⟩.

[9] OpenID Foundation: OpenID Attribute Exchange 1.0 (2007), available from ⟨http://openid.net/specs/openid-attribute-exchange-1_0.html⟩.

[10] Wang, Q., Khurana, H., Huang, Y. and Nahrstedt, K.: Time Valid One-Time Signature for Time-Critical Multicast Data Authentication, *IEEE Conference on Computer Communications 2009*, pp.1233–1241 (2009).

[11] Tsang, P. and Smith, S.: YASIR: A Low-Latency, High-Integrity Security Retrofit for Legacy SCADA Systems, *Proc. IFIP TC 11 23rd International Information Security Conference*, *The International Federation for Information Processing*, Vol.278, pp.445–459, Springer US (2008).

[12] Bobba, R., Khurana, H., AlTurki, M. and Ashraf, F.: PBES: A Policy Based Encryption System with Application to Data Sharing in the Power Grid, *Proc. 4th International Symposium on Information, Computer, and Communications Security*, pp.262–275, New York, NY, USA, ACM (2009).

[13] Niwa, Y., Kanaoka, A. and Okamoto, E.: Development of a Key Generation Center for Identity-Based Encryption on Pocket Server, *Computer Security Symposium 2012*, Vol.2012, No.3, pp.835–842 (2012). (Japanese Only).

[14] ETSI: Open Smart Grid Protocol (OSGP), Refference DGS/OSG-001, European Telecommunications Standards Institute, Sophia Antipolis Cedex (2012).

[15] Jovanovic, P. and Neves, S.: Dumb Crypto in Smart Grids: Practical Cryptanalysis of the Open Smart Grid Protocol, Cryptology ePrint Archive, Report 2015/428 (2015).

[16] Rial, A. and Danezis, G.: Privacy-preserving Smart Metering, *Proc. 10th Annual ACM Workshop on Privacy in the Electronic Society, WPES '11*, pp.49–60, New York, NY, USA, ACM (2011).

[17] Jawurek, M., Johns, M. and Kerschbaum, F.: Plug-In Privacy for Smart Metering Billing, *Privacy Enhancing Technologies*, Vol.6794, pp.192–210 (2011).

[18] Pedersen, T.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing, *Advances in Cryptology — CRYPTO '91*, Feigenbaum, J. (Ed.), Lecture Notes in Computer Science, Vol.576, Springer Berlin Heidelberg, pp.129–140 (1992).

[19] Molina-Markham, A., Danezis, G., Fu, K., Shenoy, P. and Irwin, D.: Designing privacy-preserving smart meters with low-cost microcontrollers, *Financial Cryptography and Data Security*, pp.239–253, Springer Berlin Heidelberg (2012).

[20] Armando, A., Carbone, R. and Merlo, A.: Formal Analysis of a Privacy-Preserving Billing Protocol, *Smart Grid Security*, pp.108–119 (2012).

[21] Bellare, M., Garay, J.A. and Rabin, T.: Fast Batch Verification for Modular Exponentiation and Digital Signatures, *Advances in Cryptology* (*EUROCRYPT 1998*), LNCS, Vol.1403, Springer (1998).

[22] Karati, S., Das, A., Roychowdhury, D., Bellur, B., Bhattacharya, D. and Iyer, A.: Batch Verification of ECDSA Signatures, *Progress in Cryptology* (*AFRICACRYPT 2012*), LNCS, Vol.7374, pp.1–18, Springer (2012).

[23] Camenisch, J., Hohenberger, S. and Pedersen, M.O.: Batch Verification of Short Signatures, *Advance in Cryptology* (*EUROCRYPT 2007*), LNCS, Vol.4515, pp.246–263, Springer (2007).

Appendix

A.1 Batch Verifications

A.1.1 More Constructions with Batch Verifications

When a verification function output *false*, a verifier, i.e., $\mathcal{U}_C$ has to identify which signatures are invalid. In this case, individual verifications for all signatures are inefficient in terms of the computational cost. We can use batch verifications [21] where multiple signatures can be verified via a single verification test. To the best of our knowledge, no generic and practical batch verification scheme for any signature scheme has been proposed; however, we can utilize batch verification schemes for most famous signature schemes. Batch verification schemes for RSA [21] and ECDSA [22] have been proposed. Moreover, batch verification schemes for many pairing-based signature schemes have been proposed in Ref. [23]. These can be useful for identification of invalid signatures. Thus, for a countermeasure against invalid signatures, we can utilize these schemes.

A.1.2 Small Exponents Test

Herein, we describe a method called small exponents test [21]. Let g be a generator of a group $\mathbb{G}$ that a signature scheme is defined with a prime order p. Input of this algorithm is g of $\mathbb{G}$, a security parameter ℓ, $(x_1, y_1), \cdots, (x_i, y_i)$ with $x_j \in \mathbb{Z}_p$ and $y_j \in \mathbb{G}$ for all $j \in [1, i]$. Then, the goal of the algorithm is to check $y_j = g^{x_j}$ for all $j \in [1, i]$. To check them, the algorithm first picks $(s_1, \cdots, s_i) \in \{0, 1\}^{\ell \times i}$ at random, and computes $x = \sum_{j=1}^{i} x_j s_j \bmod p$, and $y = \prod_{j=1}^{i} y_j^{s_j}$. If $g^x = y$ holds, then the algorithm outputs *accept*. Otherwise, it outputs *false*.

A.1.3 Batch Verification for RSA-FDH Signatures

We briefly describe an instantiation of the batch verification scheme described in the previous section below. The following scheme is for RSA-FDH signatures.

Input of this algorithm is a public key (N, e), i pairs $(m_1, x_1) \cdots, (m_i, x_i)$ of messages m_j and signatures $x_j \in \mathbb{Z}_N^*$ for all $j \in [1, i]$, and a hash function H. Then, the algorithm checks if $(\prod_{j=1}^{i} x_j)^e = \prod_{j=1}^{i} H(m_j)$ holds. If so, the algorithm outputs *accept*. Otherwise, it outputs *false*.

Hikaru Kishimoto received B.E. degree in Engineering from The National Institution of Academic Degrees and University Evaluation, Japan, in 2015. He has joined masters course in Graduate School of Information Science and Technology in Osaka University, Japan.

Naoto Yanai received B.E. degree from Ichinoseki National College of Technology, Japan, in 2009, M.S.Eng. from Graduate School of Systems and Information and Engineering, University of Tsukuba, Japan, in 2011, and Dr.E. degree from Graduate School of Systems and Information and Engineering, University of Tsukuba, Japan, in 2014. He is an assistant professor at Osaka University, Japan. His research is in the are of cryptography and information security.

Shingo Okamura received B.E., M.E., and Ph.D. degrees in information science and technology from Osaka University in 2000, 2002, and 2005, respectively. Since 2005, he has worked for Osaka University. In 2008, he joined National Institute of Technology, Nara College. Currently, he is an associate professor at the college. His research interests include cryptographic protocols and cyber security. He is a member of IEICE, IEEJ, ACM, IEEE, and IACR.