A New Payment System for Enhancing Location Privacy of Electric Vehicles

Man Ho Au, *Member, IEEE*, Joseph K. Liu, Junbin Fang, Zoe L. Jiang, Willy Susilo, *Senior Member, IEEE*, and Jianying Zhou

***Abstract*—An electric vehicle (EV) is a promising and futuristic automobile propelled by electric motors, using electrical energy stored in batteries or another energy storage device. Due to the need for the battery to be recharged, the cars will be required to visit a recharging infrastructure very frequently. This may disclose the users' private information, such as their location and, thus, compromise users' privacy. In this paper, we propose a new payment system that is suitable for EVs. Our system not only supports privacy protection (location privacy) but supports traceability in the case where the cars are stolen as well. Our system can further support the future vehicle-to-grid (V2G) paradigm. In addition, we prove the security and produce a proof-of-concept prototype to enumerate our system.**

***Index Terms*—Electric vehicles, location privacy, payment system.**

I. Introduction

An electric vehicle (EV) is a vehicle that does not rely on gasoline or liquefied petroleum gas as fuel but only uses electricity stored inside the car battery as the source of kinetic energy; hence, it offers emission-free urban transportation. It uses an electric motor instead of a gasoline engine to accelerate. The battery can be recharged by the common household electricity for normal charging (slow charging) or by the specifically designed charging station for fast charging. There are many advantages of using EVs. For example, EVs do not emit harmful tailpipe pollutants from the onboard source of power at the point of operation (zero tailpipe emissions). If renewable energy (such as solar or hydroelectricity) is used, the EV becomes a renewable form of transportation.

Future EVs may support a vehicle-to-grid (V2G) system. This concept allows vehicles to provide power (sell electricity) to the grid. This can be done if an EV is equipped with a solar panel and is parked outside in sunshine. In this way, the EV can help to balance loads by "valley filling" (charging at night when the demand is low) and "peak shaving" (sending power back to the grid when the demand is high). It can enable utilizing new ways to provide regulation services (keeping voltage and frequency stable) and spinning reserves (meeting sudden demands for power).

In the U.S., EVs have started to become popular due to the advantages that it can offer. The Department of Energy's eGallon provides a comparison in terms of cost to commute to work or drive across town, by posting at every corner gas station on how much the EV drivers can save on fuel by using electricity instead of gasoline. Essentially, it compares the cost to drive an EV and a traditional gasoline-based car. On average, fueling a gasoline-based car will cost roughly three times more than fueling it with electricity, i.e., in the case of EVs. In addition to saving money, EVs also offer significant environmental benefits, which makes the adoption of EVs very attractive.

Despite their potential benefits, widespread adoption of EVs faces several hurdles and limitations. One of the major problems is the driving range. Most EVs can only go about 100 to 150 km before recharging, whereas gasoline vehicles can go over 500 km before refueling. This may be sufficient for city trips or other short hauls. Nevertheless, people can be concerned that they would run out of energy from their battery before reaching their destination, which is a concern known as the range anxiety.

One of the solutions is to install more fast-charging stations with high-speed charging capability so that consumers could recharge the 100-km battery of their EV to 80% in about 30 min. EV drivers may then charge their vehicles at their homes, offices, shopping malls, or car parks outside restaurants while they are having dinner.

1) Location privacy concern: The luxury of charging EVs at the drivers' comfort also comes with some drawbacks. In practice, EVs need to travel at a certain time between two charging stations. As stated previously, since the distance is relatively much shorter compared with gasoline-based cars, this will lead to some issues that are related to location privacy [23]. These locations include the drivers' homes, places of employment, places of amusement places where they usually go, etc. [34],

Manuscript received October 3, 2012; revised February 21, 2013 and June 20, 2013; accepted July 10, 2013. Date of publication July 23, 2013; date of current version January 13, 2014. The work of Z. L. Jiang was supported by the National Natural Science Foundation of China under Grant 61240011. The work of W. Susilo was supported by the Australian Research Council Future Fellowship under Grant FTP0991397. The work of J. K. Liu and J. Zhou was supported by the Energy Marketing Authority under Project SecSG-EPD090005RFP(D). An earlier version of this paper was presented at the 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10–14, 2012. The review of this paper was coordinated by Prof. Y. Zhang. *(Corresponding author: J. K. Liu.)*

M. H. Au and W. Susilo are with the School of Computer Science and Software Engineering, University of Wollongong, Australia.

J. K. Liu and J. Zhou are with the Institute for Infocomm Research, Singapore 138632.

J. Fang is with the Department of Optoelectronic Engineering, Jinan University, Guangzhou 510632, China.

Z. L. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2013.2274288

[35], [40]. Leaking privacy will directly produce negative impacts [8], [25], [29], such as location-based "spam," which means that the location information could be used by malicious businesses to bombard an individual with unsolicited marketing for products or services related to the location of that individual. Another negative effect is that the location can be used to infer an individual's political views, state of health, or personal preferences. Furthermore, the disclosure of location privacy may also result in safety problems. For example, it may be used by unscrupulous people, such as robbers, for stalking or physical attacks. Therefore, location privacy issues must be carefully addressed before EVs can be adopted everywhere in practice.

It is interesting to note that this location privacy problem does not exist in gasoline cars. Gasoline cars will not require to be refilled within a short distance; therefore, merely tracking the distance between the last gas station and the next gas station will reveal no useful information. In practice, drivers may need to refill the gasoline once a week. There will be many activities within that week that will be untraceable. When drivers pay for the gasoline in the gas station using a credit card, then this information will be known. Nevertheless, many drivers still prefer to pay with cash, which is untraceable. Furthermore, even credit card payment will not reveal too much information since the gasoline refilling activities will not be very frequent, and as highlighted earlier, the activities after the car has been refilled will remain unknown. This is in contrast to EVs. Moreover, since EVs can support V2G charging, the location of the last refill can even be easily recorded. We will examine other payment systems and their impact on location privacy in Section I-B.

2) Revocation of location privacy at the "right" time: It is clear that EVs require protection against location privacy. Nevertheless, caution must be exercise when providing location privacy to EVs as unconditional location privacy is not always desirable in practice. Consider the case when the EV is stolen. The owner will be definitely interested to know the location of the stolen car, and will hence aim to retrieve it back at a later stage. Obviously, some antitheft or thief-tracing devices can be installed in the car (e.g., a Global Positioning System with a GSM communication device) so that if the car is stolen, the device will send a signal to the car owner telling about the current location of the stolen car. Although this kind of devices can be used to trace any stolen car, the installation and running cost are very high. It is fine for a luxury car as the cost of the antitheft device compared with the cost of the car itself is just negligible. However, for some lower end used cars, it is impractical to install these devices where the price is comparable to the value of the used car. Therefore, it is desirable to find an alternative solution that will offer the remedy to this problem while it is still practical. Essentially, it is required that the location privacy can be revoked at the *right* time.

As outlined earlier, the disadvantage of EVs is mainly due to the short driving range. In fact, this is a double-edge sword. This "feature" also provides a cheap alternative and solution to trace the stolen EVs. As the vehicle is required to be recharged very frequently, charging stations can be used to trace any stolen vehicle. If a stolen car is being recharged at a charging station, the charging station can report to the police or the car owner about the location of this stolen car. It may also refuse to provide charging service to any stolen cars; hence, the cars will be stopped.

A. *Summary of System Requirement*

In summary, the unique aspect of a payment system for EVs includes the following.

1) *Support for two-way transactions*. Users in the system could act as the payer and the payee in the transaction.
2) *Privacy preserving*. Users' transactions should be unlinkable.
3) *Voluntary revocation*. Given the user's consent, the user's transactions could be traced.

In the following, we review some existing payment systems and discuss why they cannot satisfy all these requirements simultaneously.

B. *Related Works on Existing Payment Systems*

There are many different forms of existing payment systems. We examine some of the most practical forms and explain why they are not suitable for EVs.

1) **Paper cash**: Unlike gas stations, charging stations for EVs are all machine operated. If they allow cash payment, the installation costs will be very high due to the high security requirement of cash machines [similar to those for automatic teller machines (ATMs)]. Note that, currently, there are many ticketing machines installed in car parks or automatic selling machines (e.g., selling soft drinks), which can accept paper cash or coins. However, as the cost for car park or soft drinks is far less than charging EVs, the physical security requirement can be much lower. Thus, although paper cash can provide anonymity, the high installation and running cost are the main obstacles that are disfavored by suppliers to adopt paper cash as a kind of a payment system in the charging station.
2) **E-cash**: Alternatively, e-cash [21], [24], [26], [44], [45] is the electronic form of paper cash, which also provides anonymity. However, e-cash is mainly used in small-amount transactions (e.g., a few dollars) instead of large-amount transactions (e.g., a few hundred dollars) due to security and efficiency concerns. To support two-way payments, transferable e-cash [4], [19], [37] is needed, and it has been shown that the complexity of transferable e-cash linearly grows in the number of transfers that are supported [22].[1] Apart from that, offline e-cash cannot provide double-spending *prevention*. It can only *detect* double spending and *reveal* the identity of the double

[1]A recent approach has achieved constant-size transferable e-cash, with the drawback that user storage is linear to the number of his spent coins [30].

TABLE I
COMPARISON OF EXISTING PAYMENT SYSTEMS

Scheme	Location privacy	Prevent. of cheating	Sppt. JA	Low impl. cost	Lost protect.	2-ways trans.	Stolen car traceability
Paper cash	✓	✓	×	×	×	✓	×
Prepaid cash card/ Cash coupon	✓	✓	×	✓	×	✗	×
Transferable e-cash	✓	×	✗ [a]	✓	×	✓	×
Micropayment	✗	×	×	✓	×	×	×
Credit card	×	✓	✓	✓	✓	✓	×
PayPal	✗	✓	✓	✓	✓	✓	×
Our system	✓	✓	✓	✓	✓	✓	✓

[a] Most of the existing e-cash systems do not support judge, though some of them (e.g. [12], [16], [9], [20]) do support judge.

spender when the electronic coins are deposited back to the bank. If a cheating user double-spends many times before going bankrupt, the deceived shops cannot get back the money that they deserved to have. Furthermore, different from credit cards, e-cash does not provide lost protection. No one will put a few thousand or even a few hundred dollars in the e-wallet. Thus, e-cash is only suitable for small-amount transactions. Charging for an EV does not definitely belong to the small-amount transaction category.

In addition, we note that e-cash is designed specifically to protect the privacy of the payer. This is different to our system requirement where the users could also take the role of payees in the V2G paradigm.

3) **Micropayment**: Micropayment [27], [36], [39], [41], [48] is another form of an electronic payment system. Compared with e-cash, it is usually more efficient. However, there are some tradeoffs for the advantage of efficiency. The security of micropayment may not be as strong as e-cash. Some of them may require trusted hardware. In addition, privacy is sometimes not considered an essential requirement in a micropayment system. Thus, it is only suitable for payment with a small amount.
4) **Prepaid cash card or cash coupon**: Prepaid cash card or cash coupon (e.g., EZ link in Singapore [1], Octopus in Hong Kong [2], and Oyster in London [3]) is another common way of anonymous e-payment. However, similar to e-cash, it does not support lost protection. Executing a large-amount transaction may bring inconvenience to user, i.e., they may neither want to bring many coupons together nor buy the coupons or top up everyday. In addition, it also does not fully support two-way transactions, which is a necessary requirement for the future V2G system.
5) **Paypal**: Paypal [46] is a kind of the most commonly used electronic prepayment system. However, it requires a third party (the PayPal company). If the authority colludes with the PayPal company (e.g., by telling the PayPal company the exact time and location of a particular transaction), the user can be traced. Thus, we regard PayPal providing *partial location privacy* only.
6) **Credit card**: Credit card is a widely adopted payment system for large-amount transactions. It also supports two-way transactions. Nevertheless, a credit card is not anonymous. Due to the frequent charging requirement for EVs, location privacy will be lost by easily tracing the credit card payment.

We summarize the comparison of our system with some existing payment systems in Table I.

C. Our Contributions

In this paper, we enhance the location privacy of EVs at the right time by proposing a new payment system that provides the following privacy-related features.

1) Two-way anonymous payment: It supports anonymous payment *in both directions*. First, the EV remains anonymous when it recharges at any charging station. It further supports the V2G system. That is, if the car wants to sell back its stored or solar-generated electricity to the grid through the charging station, it will receive its credit anonymously. The location privacy of the car is protected *under normal operations*.
2) Traceability of a stolen car: If the EV is stolen, the owner may provide some secret information to charging stations so that when the stolen car is being recharged again at any charging station, its location will be revealed. Note that if the thief never recharges the car or breaks the car into pieces, there is no way for our system to trace the car.

We argue that our system is practical as it also provides some additional features that can be favored by users or supplier.

1) *Prevention of cheating users*. Unlike e-cash, which cannot prevent users from cheating or double spending (it can only *detect* such behavior), our payment system supports the prevention of any cheating behavior. If any party does not follow the algorithms, the other party can stop providing service immediately. This protects the supplier from being cheated. (The difference between prevention and detection of cheating users is explained in Section I-B.)
2) *Support judging authority (JA)*. In case that there are some disputes between two parties (maybe due to some physical factors such as sudden breakdown of electricity supply), the affected party may submit all transaction information to a JA. The authority can reveal the identity and investigate the situation *after getting the consent from the user*.

3) *Low implementation cost*. Our system does not require any special security device (e.g., different from ATMs). Our security comes from cryptographic algorithms. Our system is also efficient enough to be carried out by current-generation mobile processors. As a proof of concept, we create a prototype that runs on a smartphone with reasonable performance.
4) *Lost protection*. While our system is software based, lost protection is also supported. That is, if the token of the user is stolen, the user can report the incident by supplying some secret information, which prevents the token from being usable. Furthermore, the user can regain his unused credit by providing some authenticated information.

D. Enhancement Over Our Conference Version in [42]

In this paper, we have significantly improved the payment system protocol with the following enhancements.

First we enhance the Judge Open function. In our previous work [42], the judge can open user's transaction at its own will. We redesign the algorithm here so that the judge cannot open any user transaction by itself. Instead, it needs to obtain a kind of "consent" from the user before opening. This is to prevent the judge from having too much power, as in the case of a credit card or PayPal.[2]

The Judge Open algorithm in our previous work [42] is inefficient. It requires composite-order pairing, which is not practical to be implemented in mobile devices. In this paper, the user supplies his consent in the form of some secret value. Given the secret value, the judge can efficiently tell if a transaction is conducted by the user. Here, the judge only needs to carry out one exponentiation for the testing. The whole system can be constructed using prime order pairing, which is at least 50 times faster than composite-order pairing [28].

We also provide a detailed security analysis, which further assumes that the judge could be malicious.

Furthermore, we produce a proof-of-concept prototype to enumerate the computation at both sides.

II. System Architecture

A. Entities

We consider a system that is composed of the following three entities. First, a user refers to an EV, which is implicitly referred to as an in-car-unit device. More details will be described in the following. Second, the supplier refers to the power-grid company. It supplies (sells) electricity to the cars and collects back (buys) electricity from the cars. It is responsible for account opening. Every user needs to obtain an account from it and deposits some money into this account. Finally, the JA is responsible to investigate some disputed transactions between the user and the power company. It has the power to trace all transactions made by a specific user, given the user's consent. It may be the government authority or the court.

[2]In the case for credit card or PayPal, the credit card company or PayPal can open any user/transaction *unconditionally*.

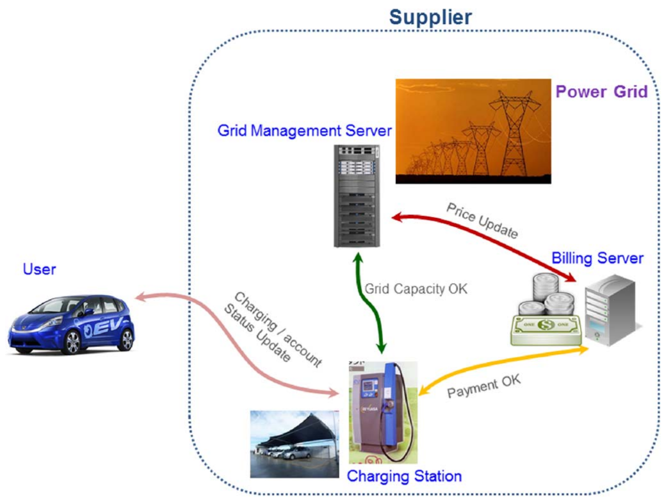

Fig. 1. Charging/top-up scenario.

B. Overall Structure

We briefly describe the overall structure of our system. We assume that each car is associated with an in-car unit in which the computations are executed. In Section IV, we describe the assumption made on this in-car unit in detail. Briefly speaking, we assume the in-car unit consists of a small amount of read-only memory, which will be initialized during the registration process. In our construction, the read-only memory will be used to store the user identity and the user secret value chosen during the registration. The user can freely read the contents of the in-car unit and modify its memory (except the read-only memory).

Our system consists of the following protocols, and the scenario related to charging is shown in Fig. 1.

1) *Registration*. The user contacts the supplier for registration and account opening. He needs to pay a deposit for his account so that the balance should have at least D dollars. At the end of the protocol, the in-car unit is initialized and is ready to be used. Balance D is stored in the in-car unit. In practice, this process may be carried out when the user collects his car from the supplier. The in-car unit may communicate with the supplier through Wi-Fi, Bluetooth, or direct cable connection.
2) *Charging*. The in-car unit carries an interactive protocol with the charging station, which first checks with the grid management server to confirm if the grid capacity is fine. If the price is dynamic (if it is within the peak period, the price may be set higher), it further checks with the grid management server for the updated price. Other than that, ***the charging station works as a front-end terminal, and the major (cryptographic) computation (e.g., those involving a secret key) is done in the supplier's billing server***. If the balance of the user account is larger than the price of the requested service, the charging station starts to charge the car. The balance stored in the in-car unit is updated. In practice, the in-car unit may communicate with the charging station through Wi-Fi or Bluetooth.
3) *Discharging or top up*. The process is similar to charging. The only difference is that, upon completion of the protocol, the balance stored in the in-car unit is increased.

4) *Statement*. At every statement period, the user approaches the supplier to top up the balance to make it D again. In practice, the in-car unit may communicate with the supplier through Wi-Fi or Bluetooth if the parking place contains Wi-Fi or Bluetooth connectivity or by using a GSM data connection directly.
5) *Judge tracing* (stolen car tracing). If the judge thinks that a particular user has performed some illegal activities, it can seek the consent from the user for tracing his activities. Given the user's consent, the judge can trace all the transactions conducted by this user, including those in the future (in the case of stolen car tracing). Meanwhile, the transactions from other users remained anonymous or untraceable.
6) *Report of lost token* (optional). This algorithm serves two purposes, namely, lost protection and lost car tracing. We present this algorithm as optional because both purposes depend on some kind of external assumptions. To enjoy lost protection, we assume that the user will back up the content of his in-car unit in a safe and independent location. In the case that the in-car unit is stolen, the user could present the backup content to the supplier. The supplier checks if the information is correct. If it is, it will block any party from using this content. The user can claim back the remaining balance. If we further assume that each car is equipped with exactly one in-car unit, that the unit is not replaceable, and that the read-only memory is not modifiable, this algorithm allows the user to block any recharging for the lost car. In fact, if the thief tries to recharge a stolen car, it will be identified by the judge tracing algorithm.

C. Threat Model and Research Objectives

In this paper, we consider the following four kinds of attacks, namely, location privacy infringement, statement fraudulent, slandering, and hiding, which are detailed in the following. Note that the adversary in each of the listed attackers could be an insider or collusion of insiders in the system. For instance, in an attempt to breach location privacy of an honest user, the adversary is assumed to control a set of dishonest users, the supplier, and the judge.

1) Location privacy infringement: The attacker tries to track transactions of an honest user. We consider a powerful adversary that can be the collusion of an insider in the supplier and judge, as well as a set of other users. In the formal security model presented in Appendix B, the adversary will be given the secret key of the supplier and the judge. Its goal is to decide if a given payment (or top-up) transaction belongs to the one of the two honest users. We further assume that the transactions are scheduled according to the adversary's wish.[3] That is, the adversary can instruct these two honest users to conduct transactions in an arbitrary sequence chosen by the adversary in an adaptive manner. The practical significance of this threat model is that, even if an attacker physically follows a certain EV, eavesdrops all its transactions, and has access to the private key of the supplier and the judge, the attacker still cannot tell if another payment or top-up transaction belongs to this specific user or another honest user.
2) Fraudulent Statement: The attacker tries to pay less than what he enjoys. The adversary could be an insider with a valid account or a set of colluding users. The adversary is also capable of eavesdropping the payment and top-up transactions of other honest users. Furthermore, the adversary can also collude with the judge. In other words, if the adversary has deposited n dollars to the supplier (n is the balance of all the colluding users), the goal of the adversary is to conduct a set of payment protocol whose total is greater than n.
3) Slandering: The attacker tries to slander an honest user in two ways. It could be a registered but malicious user who releases a piece of tracing information to the judge so that the judge would link transactions from an honest user. As another way to slander an honest user, the attacker, who could be a registered but malicious user, tries to conduct transactions so that, when an honest user releases a piece of tracing information to the judge, the judge would link the transaction conducted by the malicious user and the honest user. We further assume that the attacker is the collusion of a set of registered users, in addition to having access to the judge's secret key. This is to model the case when the adversary is colluding with a set of users together with an insider in the judge organization. The adversary is also capable of eavesdropping payment and top-up transaction of the honest users in the system.
4) Hiding: The attacker, who is a registered user, tries to conduct a transaction that could not be traced in the judge tracing protocol *without being detected*. Of course, a user can always refuse to give the consent to the judge to remain untraceable. This action, however, can be detected, and we discuss in Section V how uncooperative users are to be dealt with. In this attack, however, the malicious user appears to be cooperative, but its goal is to make some of his transactions untraceable. The attacker is supposed to have access to the judge secret key and can eavesdrop transactions conducted by the other users. As a generalized notion, we consider the case when n users are controlled by the adversary. The adversary sends n pieces of valid tracing information to the judge and is considered successful if there exists a transaction conducted by the adversary that is not traceable to any of these n users.

We aim to propose a practical payment system that is secure against the given threats. Since both slandering and hiding are related to the correctness of the tracing algorithm, we say that a payment scheme offers correct tracing if it is secure against both a slandering and a hiding attack. Likewise, we say a payment scheme prevent cheating users if it is secure against a fraudulent statement. In Appendix B, we shall formally define the security and privacy requirements of location privacy, prevention

[3]Looking ahead, our model is even stronger than what is being described. In particular, we assume the attacker will play the role of the supplier and the judge.

of cheating users, and correct tracing using a computational approach, which has been commonly used in the cryptographic community since 1984 [32].

III. Primitives

Here, we first review some cryptographic primitives and number-theoretic assumptions that will be used.

1) Bilinear Pairing: Bilinear pairing (or bilinear map) is a popular building block in public key cryptography. We briefly review its property here. Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic groups of prime order p, where p is of λ bit for some security parameter λ. A function $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is called a bilinear pairing if the following holds: 1) For all g, $h \in \mathbb{G}$, and a, $b \in \mathbb{Z}_p$, $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$ (bilinearity); 2) there exists $g \in \mathbb{G}$ such that $\hat{e}(g, g)$ has order p in $\mathbb{G}_T$ (nondegeneracy); and 3) it is efficient to compute $\hat{e}(g, h)$ for all g, $h \in \mathbb{G}$ (computability).

2) Decisional Diffie–Hellman Assumption: Given a cyclic group $\mathbb{G}_p = \langle \mathfrak{g} \rangle$ of prime order p, we say that the decisional Diffie–Hellman (DDH) assumption holds for group $\mathbb{G}_p$ if it is infeasible to distinguish the two distributions $(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, \mathfrak{g}^{ab})$ and $(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b, T)$, where a, $b \in_R \mathbb{Z}_p$ and $T \in_R \mathbb{G}_p$ are picked uniformly at random. The DDH problem instance consists of (D_1, D_2, D_3, D_4) picked from the former distribution and the latter distribution uniformly at random.

Remarks: The DDH assumption does not hold in group $\mathbb{G}$ equipped with bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ because when a 4-tuple $(D_1, D_2, D_3, D_4) \in \mathbb{G}$ is input, one could test if $\hat{e}(D_2, D_3) = \hat{e}(D_1, D_4)$. If it holds, one could conclude that there exists a, b such that $T_2 = T_1^a$, $T_3 = T_1^b$, and $T_4 = T_1^{ab}$. On the other hand, it is believed to hold in group $\mathbb{G}_T$ for a wide range of pairings.

3) Commitment: Our system uses the well-known commitment scheme, as proposed in [47]. Let $\mathbb{G}$ be a cyclic group of prime order p and g, h be the generators of $\mathbb{G}$. When value $x \in \mathbb{Z}_p$ is input, the committer randomly chooses $r \in \mathbb{Z}_p$ and computes and outputs $C = g^x h^r$ as a commitment of value x. To reveal the value committed in C, the committer outputs (x, r). Everyone can test if $C = g^x h^r$. Sometimes, we say r is the opening of C with respect to x. One could extend the commitment scheme to allow committing a tuple of elements $(x_1, \ldots, x_n)$ at the same time by setting $C = g_1^{x_1} \ldots g_n^{x_n} h^r$, where g_i are independent generators of $\mathbb{G}$.

We use $\mathsf{CMT}(x)$ (resp. $\mathsf{CMT}(x_1, \ldots, x_n)$) to denote a Pedersen commitment of value x (resp. $(x_1, \ldots, x_n)$). Note that this commitment scheme is homomorphic, i.e., $\mathsf{CMT}(a) * \mathsf{CMT}(b)$ gives $\mathsf{CMT}(a + b)$, and the opening of the latter is the sum of that of the former.

4) BBS + Signature: We employ the signature scheme proposed by Au *et al.* [7], which is based on the schemes in [15] and [11]. Their scheme, which is called BBS + signature, is briefly reviewed here. Let $g, g_0, g_1, g_2, g_3 \in \mathbb{G}$ be the generators of $\mathbb{G}$. Let $\hat{e}$ be a pairing defined over $(\mathbb{G}, \mathbb{G}_T)$.

The signer's secret is value $\gamma \in \mathbb{Z}_p$, and the public key is $(w = g^\gamma, g_0, g_1, g_2)$. To create a signature over a tuple of messages (m_1, m_2, m_3), the signer randomly picks $e, y \in_R \mathbb{Z}_p$ and computes $A = (g g_0^y g_1^{m_1} g_2^{m_2} g_3^{m_3})^{(1/\gamma + e)}$. The signer outputs (A, e, y) as the signature on message (m_1, m_2, m_3). Anyone can verify the signature by testing if the following verification equation holds, i.e., $\hat{e}(A, wg^e) \overset{?}{=} \hat{e}(g g_0^y g_1^{m_1} g_2^{m_2} g_3^{m_3}, g)$.

The BBS + signature allows the signer to produce signature in a partially blinded way. That is, it allows the signer to sign a tuple of the messages (m_1, m_2, m_3) in a commitment $\mathsf{CMT}(m_1, m_2, m_3)$ without knowing the values.

Zero-Knowledge Proof: A zero-knowledge proof [33] is an interactive protocol for one party, i.e., the prover, to prove to another party, i.e., the verifier, that some statement is true, without revealing anything other than the veracity of the statement. In [31], it has been shown that, assuming the existence of a one-way function, one can create a zero-knowledge proof system for the NP-complete graph coloring problem with three colors. Since every problem in NP can be efficiently reduced to this problem, it means that all problems in NP have zero-knowledge proofs. In practice, various efficient constructions of zero-knowledge proof for statements regarding the relationship about discrete logarithms in a cyclic group of known order have been proposed [13]. We follow the notation introduced in [18]. For example, $\mathcal{PK}\{(x) : y = g^x\}$ denotes a zero-knowledge proof that the prover knows an integer x such that the statement $y = g^x$ holds. Symbols appearing on the left of the colon denote values, whose knowledge is being proven, whereas symbols appearing on the right, but not the left, of the colon denote public values.

IV. Our Proposed System

A. Hardware Assumptions and Limitations

As discussed, our system is constructed using cryptographic techniques; hence, it does not depend on any proprietary hardware. Next, we describe the relationship between the system requirements and the hardware assumptions. Note that the major security concern of the supplier, namely, prevention of cheating users, *does not* depend on any assumption on the hardware. That is, the supplier is always guaranteed that the money it receives will be equivalent to the total amount of the electricity sold based on the hardness of some number-theoretic problems even if the malicious user is able to modify the content of the in-car unit. For the user, his location privacy is guaranteed as long as the attacker do not have access to the content of his in-car unit and the hardness of some number-theoretic problems. We would also like to stress that, since we are proposing a cryptographic solution, an attacker having access to the user's secret can spend the money of the user's account. Thus, this is of the user's interest to keep the content of the in-car unit safe. Preventing the attacker from accessing the content of the in-car unit can be achieved quite easily (e.g., the content of the in-car unit can be password protected, or the unit itself is kept physically away from the attacker). Finally, lost protection depends on the following hardware-related assumptions: 1) Each car is associated with an irreplaceable in-car unit, and 2) the read-only memory of the in-car unit is not modifiable. Table II summarizes the security requirements and the corresponding assumptions.

We remark that the scope of this paper is to deal with the location privacy issues related to the payment system and does

TABLE II
SECURITY REQUIREMENTS AND HARDWARE ASSUMPTIONS

Party / Requirement	Assumptions
Supplier / Prevention of Cheating Users	Hardness of some standard Number-Theoretic Problems
User / Location Privacy	Hardness of some standard Number-Theoretic Problems; Confidentiality of Memory-Content of the In-Car-Unit
User / Unauthorized Use of the User's Account	Confidentiality of Memory-Content of the In-Car-Unit
User / Lost Protection	User backups the memory-content of the In-Car-Unit
User / Lost Car Tracing	In-Car-Unit cannot be replaced; Read-only memory of the In-Car-Unit cannot be modified
User, Supplier / Correct Tracing	In-Car-Unit cannot be replaced; Read-only memory of the In-Car-Unit cannot be modified Hardness of some standard Number-Theoretic Problems

not cover the physical aspect of possible privacy breach. For instance, if a physical camera is installed in each charging station and it records the physical identifier of the vehicle (e.g., the registration plate number), therefore, it is obvious that location privacy cannot be maintained. This is analogous to the use of physical money. If the cash register records the image of the payer, then it is always possible to link the payment from the user across different locations; therefore, anonymity is no longer preserved.

We further assume that all communication channels are authenticated. Attacks on the communication channels, including Internet Protocol hijacking, distributed denial-of-service attack and man-in-the-middle attack, are out of the scope of this paper.

B. Detailed Description

1) *System setup.* Let $\hat{e}:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ be a bilinear map as discussed. In practice, we could use asymmetric pairing (such as type D pairing) for better space efficiency. $\mathbb{G}$ will be chosen so that it is of prime order p, where p is of length λ, i.e., the security parameter. Let $g, g_0, g_1, g_2, g_3, g_4 \in_R \mathbb{G}$. The supplier randomly picks $\gamma \in_R \mathbb{Z}_p$ and computes $w = g^\gamma$. The system parameter is $\mathsf{param} = (\mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_0, g_1, g_2, g_3, g_4, w)$, and the secret key of the supplier is γ.
2) *Judge setup.* In our construction, the judge does not have any public/private key.
3) *Registration.* Each user is assigned a unique identity I in the system. In practice, this could be his driver license number. Let D be the deposit. The user engages with the supplier and enrolls into the system as follows.
    1) The user randomly picks y', s, $t' \in_R \mathbb{Z}_p$, and computes and sends $C = g_0^{y'} g_3^s g_4^{t'}$ to the supplier, along with the following proof:

    $$\mathcal{PK}_1\left\{(y', s, t'):\quad C = g_0^{y'} g_3^s g_4^{t'}\right\}.$$

    $\mathcal{PK}_1$ assures the supplier that the value C is computed correctly. A precise description of the proof (and subsequent proofs) will be given in Appendix A.
    2) The supplier randomly picks y'', e, $t'' \in_R \mathbb{Z}_p$, computes $A = (Cgg_0^{y''} g_1^I g_2^D g_4^{t''})^{(1/e+\gamma)}$, and returns (A, y'', e, t'') to the user.
    3) The user computes $y = y' + y''$, $t = t' + t''$ and checks if $\hat{e}(A, wg^e) \overset{?}{=} \hat{e}(gg_0^y g_1^I g_2^D g_3^s g_4^t, g)$. Note that value t remains unknown to the supplier. We assume that, by the end of the protocol, the values (A, e, y, I, D, s, t) are stored in the in-car unit. In particular, the values I, t are stored in the read-only memory. We remark that $\sigma_s := (A, e, y)$ is a BBS+ signature on the tuple (I, D, s, t).
    4) The supplier stores the communication transcript (which includes C, and t'') while the user stores (y', s, t'), which will be useful in the judge tracing algorithm.

    The registration protocol is shown in Fig. 2.
4) *Charging.* Let v be the value of the transaction. Let $(\widetilde{\sigma_s} := (\tilde{A}, \tilde{e}, \tilde{y}), I, \tilde{B}, \tilde{s}, t)$ be the content stored in the in-car unit. It checks if $\tilde{B} - v \geq 0$. Next, they engage in the following protocol.
    1) The in-car unit randomly picks y', $s \in_R \mathbb{Z}_p$, $\mathfrak{R} \in_R \mathbb{G}_T$, and computes and sends $C = g_0^{y'} g_1^I g_2^{\tilde{B}} g_3^s g_4^t$, $E = \mathfrak{R}^t$, $\mathfrak{R}$, and $\tilde{s}$ to the supplier, along with the following proof:

    $$\mathcal{PK}_2\left\{\begin{array}{ll}(\tilde{A}, \tilde{e}, \tilde{y}, I, \tilde{B}, y', s, t): & \\ & C = g_0^{y'} g_1^I g_2^{\tilde{B}} g_3^s g_4^t \\ \wedge & E = \mathfrak{R}^t \\ \wedge\ \hat{e}(\tilde{A}, wg^{\tilde{e}}) = \hat{e}(gg_0^{\tilde{y}} g_1^I g_2^{\tilde{B}} g_3^{\tilde{s}} g_4^t, g) & \\ \wedge & D \geq \tilde{B} - v \geq 0\end{array}\right\}.$$

    2) Note that tuple $(E, \mathfrak{R})$ contains information about the user. If the user wishes to be traced, he or she could send the value t to the judge. For each transaction, the judge can checks if $E \overset{?}{=} \mathfrak{R}^t$. This technique is borrowed from the traceable signatures due to [38]. The supplier checks if $\tilde{s}$ has never been used and randomly picks y'', $e \in_R \mathbb{Z}_p$, computes $A = (Cgg_0^{y''} g_2^{-v})^{(1/e+\gamma)}$, and returns (A, y'', e).
    3) The user computes $y = y' + y''$ and $B = \tilde{B} - v$ and checks if $\hat{e}(A, wg^e) \overset{?}{=} \hat{e}(gg_0^y g_1^I g_2^B g_3^s g_4^t, g)$. The in-car unit parses $\sigma_s = (A, e, y)$ and stores the tuple (σ_s, B, s). Note that σ_s is a BBS+ signature on the tuple (I, B, s, t). In addition, note that the content of the read-only memory (I, t) remains unchanged.

    The charging protocol is shown in Fig. 3.
5) *Top up.* Let v be the top-up value. Let $(\widetilde{\sigma_s} := (\tilde{A}, \tilde{e}, \tilde{y}), I, \tilde{B}, \tilde{s}, t)$ be the content stored in the in-car

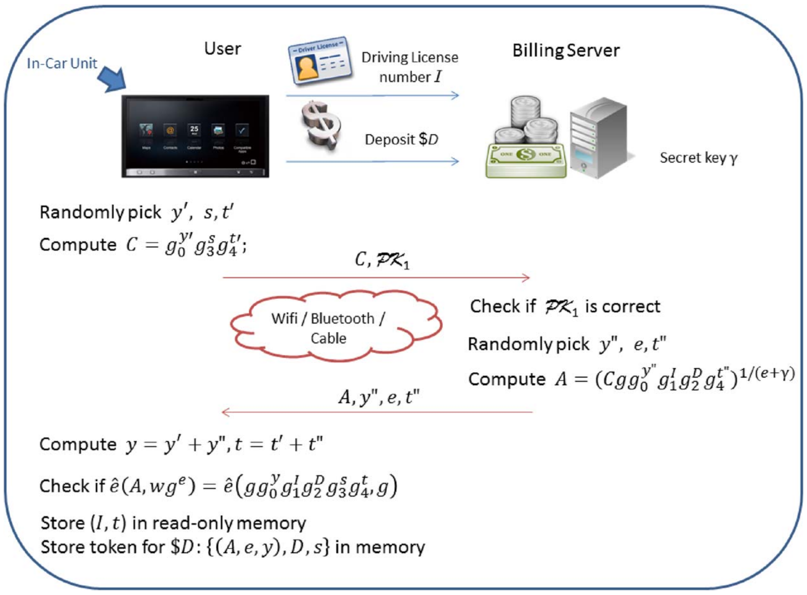
In-Car Unit
User
Driving License number I
Deposit $D
Billing Server
Secret key γ
Randomly pick y', s, t'
Compute $C = g_0^{y'} g_3^s g_4^{t'}$;
$C, \mathcal{PK}_1$
Wifi / Bluetooth / Cable
Check if $\mathcal{PK}_1$ is correct
Randomly pick y'', e, t''
Compute $A = (Cgg_0^{y''} g_1^I g_2^D g_4^{t''})^{1/(e+\gamma)}$
A, y'', e, t''
Compute $y = y' + y'', t = t' + t''$
Check if $\hat{e}(A, wg^e) = \hat{e}(gg_0^y g_1^I g_2^D g_3^s g_4^t, g)$
Store (I, t) in read-only memory
Store token for $D: $\{(A, e, y), D, s\}$ in memory

Fig. 2. Registration.

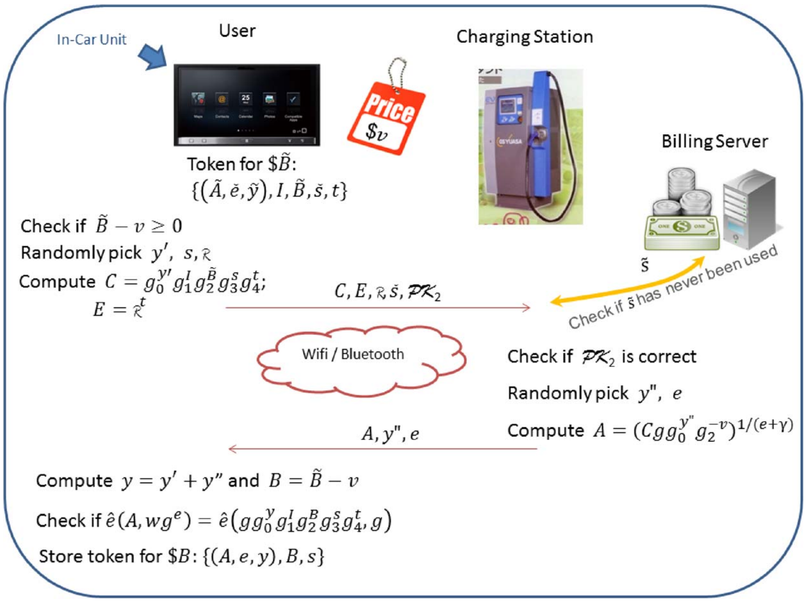
In-Car Unit
User
Charging Station
Price $v
Billing Server
Token for $\tilde{B}$:
$\{(\tilde{A}, \tilde{e}, \tilde{y}), I, \tilde{B}, \tilde{s}, t\}$
Check if $\tilde{B} - v \geq 0$
Randomly pick y', $s, \mathfrak{R}$
Compute $C = g_0^{y'} g_1^I g_2^{\tilde{B}} g_3^s g_4^t$;
$E = \mathfrak{R}^t$
$C, E, \mathfrak{R}, \tilde{s}, \mathcal{PK}_2$
$\tilde{s}$
Check if $\tilde{s}$ has never been used
Wifi / Bluetooth
Check if $\mathcal{PK}_2$ is correct
Randomly pick y'', e
Compute $A = (Cgg_0^{y''} g_2^{-v})^{1/(e+\gamma)}$
A, y'', e
Compute $y = y' + y''$ and $B = \tilde{B} - v$
Check if $\hat{e}(A, wg^e) = \hat{e}(gg_0^y g_1^I g_2^B g_3^s g_4^t, g)$
Store token for $B: $\{(A, e, y), B, s\}$

Fig. 3. Charging.

unit. We assume that D is the maximum account balance. Next, they engage in the following protocol.

1) The in-car unit randomly picks y', $s \in_R \mathbb{Z}_p$, and $\mathfrak{R} \in_R \mathbb{G}_T$ and computes and sends $C = g_0^{y'} g_1^I g_2^{\tilde{B}} g_3^s g_4^t$, $E = \mathfrak{R}^t$, $\mathfrak{R}$, as well as $\tilde{s}$, to the supplier, along with the following proof:

$$\mathcal{PK}_3 \left\{ \begin{array}{ll} (\tilde{A}, \tilde{e}, \tilde{y}, I, \tilde{B}, y', s, t) : & \\ & C = g_0^{y'} g_1^I g_2^{\tilde{B}} g_3^s g_4^t \\ \wedge & E = \mathfrak{R}^t \\ \wedge\, \hat{e}(\tilde{A}, wg^{\tilde{e}}) = \hat{e}(gg_0^{\tilde{y}} g_1^I g_2^{\tilde{B}} g_3^{\tilde{s}} g_4^t, g) & \\ \wedge & D \geq \tilde{B} + v \geq 0 \end{array} \right\}.$$

2) The supplier checks that $\tilde{s}$ has never been used and randomly picks y'', $e \in_R \mathbb{Z}_p$, computes $A = (Cgg_0^{y''} g_2^{v})^{(1/e+\gamma)}$, and returns (A, y'', e).

3) The in-car unit computes $y = y' + y''$ and $B = \tilde{B} + v$ and checks if $\hat{e}(A, wg^e) \overset{?}{=} \hat{e}(gg_0^y g_1^I g_2^B g_3^s g_4^t, g)$. It parses $\sigma_s = (A, e, y)$ and stores the tuple (σ_s, B, s). Note that σ_s is a BBS+ signature on the tuple (I, B, s, t). In addition, note that the content of the read-only memory (I, t) remains unchanged.

6) *Statement*. Let $(\tilde{\sigma_s} := (\tilde{A}, \tilde{e}, \tilde{y}), I, \tilde{B}, \tilde{s}, t)$ be the contents of the in-car unit. The user pays $v = D - \tilde{B}$ to settle his account. Next, they engage in the following protocol.

1) The in-car unit randomly picks y', $s \in_R \mathbb{Z}_p$ and computes and sends $C = g_0^{y'} g_3^s g_4^t$, $\tilde{s}$, I, and $\tilde{B}$ to the supplier, along with the following proof:

$$\mathcal{PK}_4 \left\{ \begin{array}{ll} (\tilde{A}, \tilde{e}, \tilde{y}, y', s, t) : & \\ & C = g_0^{y'} g_3^s g_4^t \\ \wedge & \hat{e}(\tilde{A}, wg^{\tilde{e}}) = \hat{e}(gg_0^{\tilde{y}} g_1^I g_2^{\tilde{B}} g_3^{\tilde{s}} g_4^t, g) \end{array} \right\}.$$

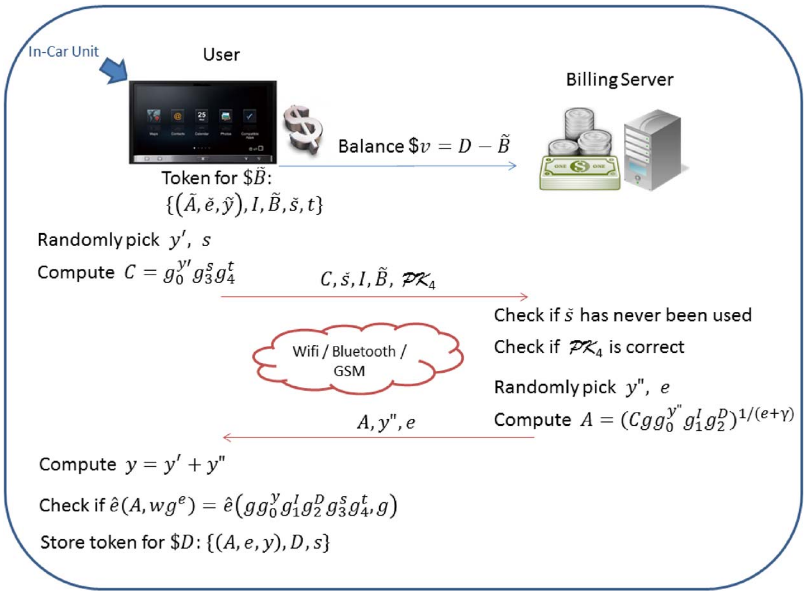
In-Car Unit
User
Billing Server
Balance $\$v = D - \tilde{B}$
Token for $\$\tilde{B}$: $\{(\tilde{A}, \check{e}, \tilde{y}), I, \tilde{B}, \check{s}, t\}$
Randomly pick y', s
Compute $C = g_0^{y'} g_3^s g_4^t$
$C, \check{s}, I, \tilde{B}, \mathcal{PK}_4$
Check if $\check{s}$ has never been used
Check if $\mathcal{PK}_4$ is correct
Wifi / Bluetooth / GSM
Randomly pick y'', e
Compute $A = (Cgg_0^{y''} g_1^I g_2^D)^{1/(e+\gamma)}$
A, y'', e
Compute $y = y' + y''$
Check if $\hat{e}(A, wg^e) = \hat{e}(gg_0^y g_1^I g_2^D g_3^s g_4^t, g)$
Store token for $\$D$: $\{(A, e, y), D, s\}$

Fig. 4. Statement.

2) The supplier checks that $\tilde{s}$ has never been used and randomly picks y'', $e \in_R \mathbb{Z}_p$, computes $A = (Cgg_0^{y''} g_1^I g_2^D)^{(1/e+\gamma)}$, and returns (A, y'', e).
3) The in-car unit computes $y = y' + y''$ and checks if $\hat{e}(A, wg^e) \overset{?}{=} \hat{e}(gg_0^y g_1^I g_2^D g_3^s g_4^t, g)$. It parses $\sigma_s = (A, e, y)$ and stores the tuple (σ_s, I, D, s, t). Note that the content of the read-only memory remains unchanged.

The statement protocol is shown in Fig. 4.

7) *Judge tracing*. In the case that the judge would like to trace a particular car, it has to obtain the consent from the user. Specifically, the user sends t to the judge. To prevent a cheating user from submitting a random t value, the judge shall obtain the registration transcript, which includes C and t'', from the supplier for this user. The user would need to prove that he is in possession of a tuple (y', s, t') such that $C/g_4^{t-t''-t'} = g_0^{y'} g_3^s$. This could be done with a zero-knowledge proof of y' and s after releasing t and t'. Of course, since the user is going to obtain a new credential after the completion of tracing, the user can simply submit y', s, and t, and the supplier can compute $t = t' + t''$ and check if $C = g_0^{y'} g_3^s g_4^t$.

After ensuring the correctness of t, for each transaction, the judge can start tracing the user by checking if the following equation holds, i.e., $E \overset{?}{=} \mathfrak{R}^t$.

For maximum privacy protection, we assume that the user will register again and obtain a fresh credential, after completion of the tracing, so that the judge cannot trace the future transaction of this user.

V. Extensions

We discuss some useful extensions for our system.

A. Incorporating Credit Expiry

In our previous construction, the supplier needs to store all the s forever. An expiration mechanism can be easily incorporated so that the supplier only needs to store the s that have not expired. Let $H : \{0,1\}^* \to \mathbb{G}$ be a collision-resistant hash function. Let $T \in \{0,1\}^*$ be the identifier of the current time period. In practice, T could be the bit string `Jan2012`, `Feb2012`, etc. The public parameter g_1, g_2, g_3, g_4 in param is replaced with hash function H.

Let T_j be the current period and T_{j+1} be the next period. For example, T_j = `Jan2012` and T_{j+1} = `Feb2012`. In the protocols, the value g_i will be replaced with $H(T, i)$ for $i = 1–4$. At the end of period T_j, all users will contact the supplier in the statement protocol. During the execution of the protocol, $g_i = H(T_{j+1}, i)$ will be used in the computation of the value A. Thus, in period T_{j+1}, the user will be using $g_i = H(T_{j+1}, i)$ for charging and topping up, and the content issued in the previous period will not be usable.

To accommodate the user who executes the statement protocol before the end of T_j, both $g_i = H(T_j, i)$ and $g_i = H(T_{j+1}, i)$ will be accepted at the end of period T_j (e.g., 25th–30th of each month). This extension does not alter the efficiency of our system.

B. Dealing with Uncooperative Users in Tracing

Although users in our payment system enjoys a very high degree of privacy, sometimes, it is necessary for them to be traceable. Indeed, our system incorporates an efficient mechanism for user tracing with the user's consent. It is natural to ask what to do if the user refuses to cooperate. One possible solution is to introduce another powerful entity, which can be used to disclose the identity of the underlying user in all transactions. This can be quite easily done by requiring the user to verifiably encrypt [17] his identity under this powerful entity's public key. However, this solution is undesirable from the users' point of view since the existence of this powerful entity essentially means that the location privacy of the system is always subjected to the mercy of the entity.

Our scheme can be easily modified to strike a balance between user privacy and accountability based on the extension of credit expiry. For the user who refuses to release his tracing information upon the judge's request, the supplier would have the right to refuse to conduct a statement protocol with the user.[4] The user's remaining credit will expire in the succeeding period; thus, his account will no longer be usable. We believe that this is one possible compromise. If the user chooses to refuse to give his consent for tracing, the supplier has the right to refuse serving this user in the future. The user's privacy is still preserved and that he would also have the time to use up all the remaining balance within the current period. The consequence of refusing to give consent for tracing would merely be the termination of the service contract without comprising user privacy.

C. Report of Lost Token

Remember that this is the only feature that depends on specific hardware assumption. This feature also assumes that the user always backs up his latest memory content (A, e, y, I, B, s, t). In the case that the memory content has been stolen (including the case when his car was physically stolen), he could report the case to the supplier, along with the backup content. The supplier checks that the token is valid, and value s has never been used. Then, the supplier marks s as used, and the user can claim back the unused credit of value B.

If s has been used, it means that the thief has used the memory content for another recharge. In this case, the supplier can use the value t to trace all the transactions of the thief.

D. Incorporating Other Payment Systems

In reality, the charging station may also support other payment methods (e.g., credit cards). In this case, we require the user to run the charging protocol of our system and pay nothing to the charging station in addition to the other payment method. Specifically, upon completion of the charging protocol of our system, the user may use other form of payment to pay the balance. This is to ensure the traceability of the stolen car can still be executed while using other payment systems. However, we note that the other payment systems that are used may leak information about the user.

VI. Practicality Analysis

Here, we show that our scheme is practical by giving analysis data in two aspects: efficiency and security.

A. Efficiency Analysis

We implement a prototype that supports the user-side computation for protocol registration, charging/top up, and statement. In the following, we show the empirical numbers measured for the computational time at the user device. Note that the number only corresponds to the time taken for the computation and, thus, does not measure network latency. For the user side, we use NEXUS 4 with 1.5-GHz Quad-Core Qualcomm Snapdragon S4 Pro with 2-GB RAM. For the supplier side, we use a notebook with 2.6-GHz Core i5-3320 and 8-GB RAM. The parameter is the same as those used on the crypto library jPBC [43]. The result is summarized in Table III.

TABLE III
IMPLEMENTED RUNNING TIME (IN SECONDS)

	Registration		Charging/Topup		Statement	
	User	Supplier	User	Supplier	User	Supplier
Overall Running time	3.566	0.164	10.094	0.801	6.246	0.421

B. Security Analysis

1) Security Requirements: We first state the security requirements of a payment system that is suitable for EVs.

1) *Prevention of cheating users*. If a user does not follow the designated algorithm to modify his credential (he may intend to do so to obtain more than what he deserves or pay less than what he should pay), he cannot pass the authentication process.
2) *Location privacy*. Without the consent from the user, no party (including the judge) is able to trace the identity of user for a particular transaction. Thus, the location privacy of user is preserved.
3) *Correct tracing*. With the consent from the user (who may reveal some secret information), the Judge should be able to trace all his previous and future possible transactions.

2) Analysis: We analyze the security of our scheme using a game-based approach. Each security requirement is modeled as a game played between probabilistic polynomial time adversary $\mathcal{A}$ and challenger $\mathcal{C}$. The games are defined so that they capture the capabilities and behavior of an adversary. The adversary winning the game would imply that it is possible to break a security requirement. Using reduction argument, we would then show any adversary winning the game could be used to break some hardness assumptions.

The details of the analysis are presented in Appendix B.

VII. Conclusion

In this paper, we have presented a mechanism to enhance location privacy for EVs. Our proposed solution provides an anonymous payment system with privacy protection support. In the case where traceability is required, such as when the EV is stolen, this feature can be also provided. Hence, our solution provides location privacy enhancement at the right time, which will make the adoption of EVs practical.

Our system provides an option to incorporate a judge who can open all transactions in the case of any dispute with the consent from the user. Given this user's consent, which is a piece of secret information, the judge can open a particular transaction for investigation. It can also trace a user for all his previous and future transactions while keeping the transactions from other users unopened.

[4]Recall that the statement protocol is not anonymous; thus, this is feasible if it is clearly written in the service contract.

We also note that the scheme described in this paper is specifically designed for EVs. However, we do not eliminate the possibility to apply our scheme (or modified version) in other environments whenever it is deemed suitable.

APPENDIX A
DETAILS OF $\mathcal{PK}_1$–$\mathcal{PK}_4$

$\mathcal{PK}_2$ and $\mathcal{PK}_3$ require the prover to demonstrate that he knows a value B, such that $\tilde{B}+v$ (or $\tilde{B}-v$ in $\mathcal{PK}_3$) is within the interval 0 and D. While zero-knowledge range proof exists, we observe that the interval $[0, D]$ is fixed and is relatively small compared with the security parameter. Thus, we can make use of the efficient interval proof due to [14]. Specifically, the supplier publishes a set of "digital signatures" on the messages 0, …, D, which is denoted $\sigma_0, \ldots, \sigma_D$. To prove that $\tilde{B}+v$ lies in the interval $[0, D]$ for a known value v, the user proves that he is in possession of signature $\sigma_{\tilde{B}+v}$ on message $\tilde{B}$. The proof is of constant size. This trick has been used in reputation-based anonymous authentication [5], [6]. We instantiate the interval proof with the weakly secured signature scheme in [10]. Looking ahead, ς_i is a signature on i under public key (f, h). Thus, in the instantiation of $\mathcal{PK}_2$ or $\mathcal{PK}_3$, the user is not directly proving that $\tilde{B}+v$ (or $-v$) is within 0 to D. Rather, the user demonstrates that he knows value ς_i, which is a valid signature on message $\tilde{B}+v$ (or $\tilde{B}-v$).

To efficiently instantiate the zero-knowledge proof $\mathcal{PK}_1$ to $\mathcal{PK}_4$, the supplier adds the following auxiliary parameters: h, $h_1, h_2 \in_R \mathbb{G}$, $f = h^{\delta}$ for some randomly generated $\delta \in_R \mathbb{Z}_p$. For $i = 0$ to D, $\varsigma_i = h^{(1/\delta+I)}$. For efficiency considerations, set $\hat{E} = \hat{e}(g, g)$ and $\hat{E}_i = \hat{e}(g_i, g)$ for $i = 0$–4, $\hat{H} = \hat{e}(h, h)$, $\hat{H}_0 = \hat{e}(h_1, w)$, $\hat{H}_1 = \hat{e}(h_1, g)$, $\hat{H}_2 = \hat{e}(h_1, h)$, and $\hat{H}_3 = \hat{e}(h_1, f)$. They will be included in the public parameter to speed up the protocol. Set $\mathsf{param} := \mathsf{param} \cup \{h, h_1, h_2, f, \varsigma_0, \ldots, \varsigma_D, \hat{E}, \hat{E}_0, \hat{E}_1, \hat{E}_2, \hat{E}_3, \hat{E}_4, \hat{H}, \hat{H}_0, \hat{H}_1, \hat{H}_2, \hat{H}_3\}$, In addition, the value of δ should be deleted or be kept secret. To reduce the number of rounds and for better space efficiency, we use the well-known Fiat–Shamir transformation where the function H is modeled as a random oracle.

$\mathcal{PK}_1$:

1) The supplier sends a random challenge R.
2) The user randomly chooses $\rho_{y'}$, ρ_s, $\rho_t \in_R \mathbb{Z}_p$ and computes $T = g_0^{\rho_{y'}} g_3^{\rho_s} g_4^{\rho_t}$.
3) The user computes $c = H(T, R) \in_R \mathbb{Z}_p$.
4) The user computes $z_{y'} = \rho_{y'} - cy'$, $z_s = \rho_s - cs$, and $z_t = \rho_t - ct$ and sends c, $z_{y'}$, z_s, and z_t to the supplier.
5) The supplier computes $T = C^c g_0^{z_{y'}} g_3^{z_s} g_4^{z_t}$ and accepts the proof if and only if $c \stackrel{?}{=} H(T, R)$.

$\mathcal{PK}_2$:

1) The supplier sends a random challenge R.
2) The user randomly chooses k_1, k_2, k_3, $k_4 \in_R \mathbb{Z}_p$, and computes $F_1 = h_1^{k_1} h_2^{k_2}$, $F_2 = \tilde{A} h_1^{k_2}$, $F_3 = h_1^{k_3} h_2^{k_4}$, $F_4 = \varsigma_{\tilde{B}-v} h_1^{k_4}$. Next, the user randomly chooses ρ_{k_1}, ρ_{k_2}, ρ_{k_3}, ρ_{k_4}, $\rho_{y'}$, ρ_I, $\rho_{\tilde{B}}$, ρ_s, $\rho_{\tilde{e}}$, $\rho_{\tilde{y}}$, ρ_{β_1}, ρ_{β_2}, ρ_{β_3}, and ρ_{β_4}, $\rho_t \in_R \mathbb{Z}_p$ and computes $T_1 = g_0^{\rho_{y'}} g_1^{\rho_I} g_2^{\rho_{\tilde{B}}} g_3^{\rho_s} g_4^{\rho_t}$, $T_2 = h_1^{\rho_{k_1}} h_2^{\rho_{k_2}}$, $T_3 = F_1^{-\rho_{\tilde{e}}} h_1^{\rho_{\beta_1}} h_2^{\rho_{\beta_2}}$, $T_4 = \hat{H}_0^{\rho_{k_2}} \hat{H}_1^{\rho_{\beta_2}} \hat{E}_0^{\rho_{\tilde{y}}} \hat{E}_1^{\rho_I} \hat{E}_2^{\rho_{\tilde{B}}} \hat{E}_4^{\rho_t} \hat{e}(F_2, g)^{-\rho_{\tilde{e}}}$, $T_5 = h_1^{\rho_{k_3}} h_2^{\rho_{k_4}}$, $T_6 = F_3^{-\rho_{\tilde{B}}} h_1^{\rho_{\beta_3}} h_2^{\rho_{\beta_4}}$, $T_7 = \hat{H}_2^{\rho_{\beta_4}} \hat{H}_3^{\rho_{k_4}} \hat{e}(F_4, h)^{-\rho_{\tilde{B}}}$, and $T_8 = \mathfrak{R}^{\rho_t}$.
3) The user computes $c = H(\{F_i\}_{i=1}^{4}, \{T_i\}_{i=1}^{8}, R, \mathfrak{R}) \in_R \mathbb{Z}_p$.
4) The user computes and sends c, F_1, F_2, F_3, F_4, $z_{k_1} = \rho_{k_1} - ck_1$, $z_{k_2} = \rho_{k_2} - ck_2$, $z_{k_3} = \rho_{k_3} - ck_3$, $z_{k_4} = \rho_{k_4} - ck_4$, $z_{y'} = \rho_{y'} - cy'$, $z_I = \rho_I - cI$, $z_{\tilde{B}} = \rho_{\tilde{B}} - c\tilde{B}$, $z_s = \rho_s - cs$, $z_{\tilde{e}} = \rho_{\tilde{e}} - c\tilde{e}$, $z_{\tilde{y}} = \rho_{\tilde{y}} - c\tilde{y}$, $z_{\beta_1} = \rho_{\beta_1} - ck_1\tilde{e}$, $z_{\beta_2} = \rho_{\beta_2} - ck_2\tilde{e}$, $z_{\beta_3} = \rho_{\beta_3} - c(\tilde{B} - v)k_3$, $z_{\beta_4} = \rho_{\beta_4} - c(\tilde{B} - v)k_4$, and $z_t = \rho_t - ct$ to the supplier.
5) The supplier computes T_1–T_8 as follows: $T_1 = C^c g_0^{z_{y'}} g_1^{z_I} g_2^{z_{\tilde{B}}} g_3^{z_s} g_4^{z_t}$, $T_2 = F_1^c h_1^{z_{k_1}} h_2^{z_{k_2}}$, $T_3 = F_1^{-z_{\tilde{e}}} h_1^{z_{\beta_1}} h_2^{z_{\beta_2}}$, $T_4 = (\hat{e}(F_2, w)\hat{E}^{-1}\hat{E}_3^{-\tilde{s}})^c \cdot \hat{H}_0^{z_{k_2}} \hat{H}_1^{z_{\beta_2}} \hat{E}_0^{z_{\tilde{y}}} \hat{E}_1^{z_I} \hat{E}_2^{z_{\tilde{B}}} \hat{E}_4^{z_t} \hat{e}(F_2, g)^{-z_{\tilde{e}}}$, $T_5 = F_3^c h_1^{z_{k_3}} h_2^{z_{k_4}}$, $T_6 = F_3^{-vc} F_3^{-z_{\tilde{B}}} h_1^{z_{\beta_3}} h_2^{z_{\beta_4}}$, $T_7 = (\hat{e}(F_4, fh^{-v})\hat{H}^{-1})^c \hat{H}_2^{z_{\beta_4}} \hat{H}_3^{z_{k_4}} \hat{e}(F_4, h)^{-z_{\tilde{B}}}$, and $T_8 = E^c \mathfrak{R}^{z_t}$. Then, it accepts the proof if and only if $c \stackrel{?}{=} H(\{F_i\}_{i=1}^{4}, \{T_i\}_{i=1}^{8}, R, \mathfrak{R})$.

$\mathcal{PK}_3$ is the same as $\mathcal{PK}_2$, except the value of $-v$ is replaced with $+v$.

$\mathcal{PK}_4$:

1) The supplier sends a random challenge R.
2) The user randomly chooses $k_1, k_2 \in_R \mathbb{Z}_p$ and computes $F_1 = h_1^{k_1} h_2^{k_2}$, $F_2 = \tilde{A} h_1^{k_2}$. Next, the user randomly chooses ρ_{k_1}, ρ_{k_2}, $\rho_{y'}$, ρ_s, ρ_t, $\rho_{\tilde{e}}$, $\rho_{\tilde{y}}$, ρ_{β_1}, and $\rho_{\beta_2} \in_R \mathbb{Z}_p$ and computes $T_1 = g_0^{\rho_{y'}} g_3^{\rho_s} g_4^{\rho_t}$, $T_2 = h_1^{\rho_{k_1}} h_2^{\rho_{k_2}}$, $T_3 = F_1^{-\rho_{\tilde{e}}} h_1^{\rho_{\beta_1}} h_2^{\rho_{\beta_2}}$, $T_4 = \hat{H}_0^{\rho_{k_2}} \hat{H}_1^{\rho_{\beta_2}} \hat{E}_0^{\rho_{\tilde{y}}} \hat{E}_4^{\rho_t} \hat{e}(F_2, g)^{-\rho_{\tilde{e}}}$.
3) The user computes $c = H(\{F_i\}_{i=1}^{2}, \{T_i\}_{i=1}^{4}, R) \in_R \mathbb{Z}_p$.
4) The user computes and sends c, F_1, F_2, $z_{k_1} = \rho_{k_1} - ck_1$, $z_{k_2} = \rho_{k_2} - ck_2$, $z_{y'} = \rho_{y'} - cy'$, $z_s = \rho_s - cs$, $z_t = \rho_t - ct$, $z_{\tilde{e}} = \rho_{\tilde{e}} - c\tilde{e}$, $z_{\tilde{y}} = \rho_{\tilde{y}} - c\tilde{y}$, $z_{\beta_1} = \rho_{\beta_1} - ck_1\tilde{e}$, and $z_{\beta_2} = \rho_{\beta_2} - ck_2\tilde{e}$ to the supplier.
5) The supplier computes T_1–T_4 as follows: $T_1 = C^c g_0^{z_{y'}} g_3^{z_s} g_4^{z_t}$, $T_2 = F_1^c h_1^{z_{k_1}} h_2^{z_{k_2}}$, $T_3 = F_1^{-z_{\tilde{e}}} h_1^{z_{\beta_1}} h_2^{z_{\beta_2}}$, and $T_4 = (\hat{e}(F_2, w)\hat{E}^{-1}\hat{E}_1^{-I}\hat{E}_2^{-\tilde{B}}\hat{E}_3^{-\tilde{s}})^c \cdot \hat{H}_0^{z_{k_2}} \hat{H}_1^{z_{\beta_2}} \hat{E}_0^{z_{\tilde{y}}} \hat{E}_4^{z_t} \hat{e}(F_2, g)^{-z_{\tilde{e}}}$. Then, it accepts the proof if and only if $c \stackrel{?}{=} H(\{F_i\}_{i=1}^{2}, \{T_i\}_{i=1}^{4}, R)$.

APPENDIX B
SECURITY ANALYSIS

Prevention of Cheating Users: The following game between attacker $\mathcal{A}$ and challenger $\mathcal{C}$ defines the requirement *prevention of cheating users*. $\mathcal{A}$ plays the role of a set of cheating users, whereas $\mathcal{C}$ plays the role of an honest supplier. In the game, $\mathcal{C}$ keeps a running balance W possessed by $\mathcal{A}$. $\mathcal{A}$ wins the game if it can make W negative. Note that, in this game, we allow $\mathcal{A}$ to register multiple times. This models the situation when several users collude together. Note that we do not assume that the judge is trusted and the secret key of the judge is also given to the attacker.[5]

1) System Parameter: $\mathcal{C}$ creates and publishes the system parameter param and keeps the secret key private. $\mathcal{C}$ also creates the public/secret key of on behalf of the judge. The secret key of the judge is also given to $\mathcal{A}$. $\mathcal{C}$ initializes counter W, which is 0.

2) Interactions: $\mathcal{A}$ can make the following four types of interaction freely with $\mathcal{C}$.

1) *In the registration process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the registration protocol. Upon successful completion of the protocol, W is increased by the value D.
2) *In the charging process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the charging protocol of value v. Upon successful completion of the protocol, W is decreased by the value v.
3) *In the discharging process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the top-up protocol of value v. Upon successful completion of the protocol, W is increased by the value v.
4) *In the statement process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the statement protocol of value d. Upon successful completion of the protocol, W is increased by the value d.

3) Eavesdropping: $\mathcal{A}$ can also observe the four types of interaction between honest users with $\mathcal{C}$. We further assume that the user to be observed in each of these transactions is to be specified by $\mathcal{A}$.

4) Winning: $\mathcal{A}$ wins the game if there exists a sequence of interaction query so that W becomes negative.

Proof: Our security proof is by reduction. Specifically, assuming that there exists $\mathcal{A}$, we show how to construct a forgery attack against the underlying BBS+ signature [7]. Since BBS+ signature is known to be unforgeable, this means that no PPT attacker $\mathcal{A}$ can win in the given game. That is, our system supports prevention of cheating users.

Before stating our proof, let us assume that the zero-knowledge proof $\mathcal{PK}_1$, $\mathcal{PK}_2$, $\mathcal{PK}_3$, and $\mathcal{PK}_4$ are *sound*. That is, given blackbox access to the prover that makes these zero-knowledge proofs, there exists extractor algorithms $\mathcal{EX}_1$, $\mathcal{EX}_2$, $\mathcal{EX}_3$, and $\mathcal{EX}_4$, which are capable of outputting the witnesses used by the prover. Indeed, the protocols described in Appendix A are *sound* in the random oracle model.

Next, we describe an algorithm called simulator $\mathcal{S}$, which provides the view to $\mathcal{A}$ as the challenger and, at the same time, forges a BBS+ signature based on the interaction with $\mathcal{A}$. $\mathcal{S}$ is given the public key of the BBS+ signature in the form of $(\hat{e}, \mathbb{G}, \mathbb{G}_T, g, g_0, g_1, g_2, g_3, g_4, w)$, together with a blackbox $\mathcal{SO}$, which is normally referred to as the *signing oracle*. $\mathcal{SO}$ outputs BBS+ signature (A, e, y) on input (m_1, m_2, m_3, m_4). $\mathcal{S}$ successfully forges a BBS+ signature if it can output valid signature (A^*, e^*, y^*) on message $(m_1^*, m_2^*, m_3^*, m_4^*)$ such that the former is not the output of $\mathcal{SO}$.[6]

Now, we describe the behavior of $\mathcal{S}$. It sets $\mathsf{param} = (\hat{e}, \mathbb{G}, \mathbb{G}_T, g, g_0, g_1, g_2, g_3, g_4, w)$. The value param is given to $\mathcal{A}$. Note that $\mathcal{S}$ does not know the secret key of the supplier, but param is distributed correctly. In the following, we show how $\mathcal{S}$ interacts with $\mathcal{A}$ in each of the possible interactions. The value W is set to 0.

1) *Registration.* Upon executing $\mathcal{PK}_1$ with $\mathcal{A}$, $\mathcal{S}$ uses $\mathcal{EX}_1$ to extract the witness $(y', s, t, \alpha, \beta)$. $\mathcal{S}$ assigns the unique identity I to this user and issues a signature query with input (I, D, s, t) to $\mathcal{SO}$. $\mathcal{S}$ receives (A, e, y) and computes $y'' = y - y'$. It returns (A, y'', e) to $\mathcal{A}$. $\mathcal{S}$ sets $W = W + D$.
2) *Charging.* Upon executing $\mathcal{PK}_2$ with $\mathcal{A}$, $\mathcal{S}$ uses $\mathcal{EX}_2$ to extracts the witness $(\tilde{A}, \tilde{e}, \tilde{y}, I, \tilde{B}, y', s, t)$. If $\tilde{A}$, $\tilde{e}$, $\tilde{y}$ is not the output of $\mathcal{SO}$, $\mathcal{S}$ outputs them as the forgery on $(I, \tilde{B}, \tilde{s}, t)$ and aborts. Otherwise, it checks if $\tilde{s}$ is fresh. If not, it rejects the request. Otherwise, $\mathcal{S}$ issues a signature query with input $(I, \tilde{B} - v, s, t)$ to $\mathcal{SO}$. $\mathcal{S}$ receives (A, e, y) and computes $y'' = y - y'$. It returns (A, y'', e) to $\mathcal{A}$. $\mathcal{S}$ sets $W = W - v$.
3) *Discharging.* Upon executing $\mathcal{PK}_3$ with $\mathcal{A}$, $\mathcal{S}$ uses $\mathcal{EX}_3$ to extracts the witness $(\tilde{A}, \tilde{e}, \tilde{y}, I, \tilde{B}, y', s, t)$. If $\tilde{A}, \tilde{e}, \tilde{y}$ is not the output of $\mathcal{SO}$, $\mathcal{S}$ outputs them as the forgery on $(I, \tilde{B}, \tilde{s}, t)$ and aborts. Otherwise, it checks if $\tilde{s}$ is fresh. If not, it rejects the request. Otherwise, $\mathcal{S}$ issues a signature query with input $(I, \tilde{B} + v, s, t)$ to $\mathcal{SO}$. $\mathcal{S}$ receives (A, e, y) and computes $y'' = y - y'$. It returns (A, y'', e) to $\mathcal{A}$. $\mathcal{S}$ sets $W = W + v$.
4) *Statement.* Upon executing $\mathcal{PK}_4$ with $\mathcal{A}$, $\mathcal{S}$ uses $\mathcal{EX}_4$ to extracts the witness $(\tilde{A}, \tilde{e}, \tilde{y}, y', s, t)$. If $\tilde{A}, \tilde{e}, \tilde{y}$ is not the output of $\mathcal{SO}$, $\mathcal{S}$ outputs them as the forgery on $(I, \tilde{B}, \tilde{s}, t)$ and aborts. Otherwise, it checks if $\tilde{s}$ is fresh. If not, it rejects the request. Otherwise, $\mathcal{S}$ issues a signature query with input (I, D, s, t) to $\mathcal{SO}$. $\mathcal{S}$ receives (A, e, y) and computes $y'' = y - y'$. It returns (A, y'', e) to $\mathcal{A}$. $\mathcal{S}$ sets $W = W + D - \tilde{B}$.

Due to the setting of the game, the value W remains positive if $\mathcal{S}$ never aborts. This is because, to reduce the value of W, $\mathcal{A}$ has to interact with $\mathcal{S}$ in the discharge protocol, the number of signatures given to $\mathcal{A}$ via $\mathcal{S}$ is limited, and $\mathcal{PK}_3$ assures that $\mathcal{S}$ will not accept on message of the form $(\cdot, B, \cdot)$ with $B > v$. Thus, in order for $\mathcal{A}$ to win the game, $\mathcal{S}$ will abort and obtain a forgery to the underlying BBS+ signature. ■

Location Privacy: Location privacy is defined via the following game. The rationale is that the malicious supplier cannot tell if a particular interaction is due to one out of two possible honest users under the extreme condition that all other interaction sequences are specified by the malicious supplier. Of course, the particular interaction could only be charging or discharging since the identity of the actual user is to be known in registration and statement. Our definition also guarantees that the charging or discharging interactions are not linkable. We *do not* assume that the judge can be trusted; thus, the attacker also plays the role of the judge. That is, the location privacy guarantee is strong. For a particular interaction, a malicious supplier, with the help of the judge, cannot distinguish if it is from one of the two possible honest users, even if the previous interactions of these two users are all scheduled by the attacker. Of course, these two honest users have not participated in the

[5]In our actual construction, the judge has no public/secret key.

[6]Note that this is formally called strong existential forgery under an adaptive chosen message attack, which is one of the strongest possible attacks on the digital signature of which BBS+ has been proven to be immune.

voluntary tracing since the attacker could trace their transaction if their consent is given.

5) System Parameter: $\mathcal{C}$ creates and publishes system parameter param and secret key γ. $\mathcal{C}$ also creates the public/secret key on behalf of the judge. The secret key of the supplier and the judge are given to $\mathcal{A}$.

6) Interactions: $\mathcal{A}$ can make the following four types of interaction freely with $\mathcal{C}$, which acts on behalf of two honest users.

1) *In the registration process* ($b \in \{0,1\}$). $\mathcal{A}$ interacts with $\mathcal{C}$, who acts on behalf of U_b in the registration protocol. The value b is specified by $\mathcal{A}$.
2) *In the charging process* ($b \in \{0,1\}$). $\mathcal{A}$ interacts with $\mathcal{C}$ who acts on behalf of U_b in the charging protocol of value v for the user. The value b is specified by $\mathcal{A}$.
3) *In the discharging process* ($b \in \{0,1\}$). $\mathcal{A}$ interacts with $\mathcal{C}$ who acts on behalf of U_b in the top-up protocol of value v. Value b is specified by $\mathcal{A}$.
4) *In the statement process* ($b \in \{0,1\}$), $\mathcal{A}$ interacts with $\mathcal{C}$, who acts on behalf of U_b in the statement protocol of value d. Value b is specified by $\mathcal{A}$.

7) Challenge: $\mathcal{A}$ chooses a type of interaction, either charging or discharging, if both U_0 and U_1 have a sufficient balance in the case when it is charging. $\mathcal{C}$ flips a fair coin $\hat{b} \in \{0,1\}$ and interacts with $\mathcal{A}$ on behalf of user $U_{\hat{b}}$.

Winning $\mathcal{A}$ outputs a guess bit b and wins the game if $b = \hat{b}$.

Proof: Our security proof is by reduction to the DDH assumption in the group $\mathbb{G}_T$. That is, if $\mathcal{A}$ can distinguish the action of two honest users, we show how to construct simulator $\mathcal{S}$, which solves an instance of the DDH problem. $\mathcal{S}$ is given as a 4-tuple $(D_1, D_2, D_3, D_4) \in \mathbb{G}_T$, and its goal is to tell if there exists a, b, such that $D_2 = D_1^a$, $D_3 = D_1^b$, $D_4 = D_1^{ab}$. The view of $\mathcal{A}$ is provided by simulator $\mathcal{S}$ who has control over the random oracle used. Next, we describe the behavior of $\mathcal{S}$.

1) *Registration.* $\mathcal{S}$ acts on behalf of user U_0 honestly. For U_1, $\mathcal{S}$ randomly picks $C \in_R \mathbb{G}$ and uses the zero-knowledge simulator to simulator the proof $\mathcal{PK}_1$.
2) *Charging.* $\mathcal{S}$ honestly acts on behalf of user U_0. For U_1, the pair of $\mathfrak{R}$ and E supplied to $\mathcal{A}$ is not correctly formed. Specifically, for the jth query, $\mathcal{S}$ randomly picks $\mu_j \in_R \mathbb{Z}_p$ and computes $\mathfrak{R} = D_1^{\mu_j}$ and $E = D_3^{\mu_j}$. $\mathcal{S}$ also randomly picks at random $\tilde{s} \in_R \mathbb{Z}_p$, $C \in_R \mathbb{G}$. Then, $\mathcal{S}$ sends C, E, $\mathfrak{R}$, $\tilde{s}$ to $\mathcal{A}$ and invokes the zero-knowledge simulator to simulate the proof $\mathcal{PK}_2$.
3) *Discharging.* $\mathcal{S}$ acts on behalf of user U_0 honestly. For U_1, the pair of $\mathfrak{R}$ and E supplied to $\mathcal{A}$ is not correctly formed. Specifically, for the jth query, $\mathcal{S}$ randomly picks $\mu_j \in_R \mathbb{Z}_p$ and computes $\mathfrak{R} = D_1^{\mu_j}$ and $E = D_3^{\mu_j}$. $\mathcal{S}$ also randomly picks at random $\tilde{s} \in_R \mathbb{Z}_p$ and $C \in_R \mathbb{G}$. Then, $\mathcal{S}$ sends C, E, $\mathfrak{R}$, and $\tilde{s}$ to $\mathcal{A}$ and invokes the zero-knowledge simulator to simulate the proof $\mathcal{PK}_3$.
4) *Statement.* $\mathcal{S}$ acts on behalf of the user U_0 honestly. For U_1, $\mathcal{S}$ picks $C \in_R \mathbb{G}$ at random and uses the zero-knowledge simulator to simulator the proof $\mathcal{PK}_4$.

Note that, although the proof for user U_1 are not computed following the protocol, the view to $\mathcal{A}$ is perfect as if user U_1 is using a value $t = \log_{D_1} D_3$.

In the *challenge phase*, $\mathcal{S}$ first checks if both users are eligible to participate in the transaction. That is, they are having a sufficient balance if the interaction is charging. Then, $\mathcal{S}$ flips a fair coin $\hat{b} \in \{0,1\}$. If $\hat{b} = 0$, $\mathcal{S}$ following the protocol honestly and acts on behalf of user U_0. Otherwise, it sets $\mathfrak{R} = D_2$ and $E = D_4$ and uses the zero-knowledge simulator to simulate the protocol $\mathcal{PK}_2$ or $\mathcal{PK}_3$.

Note that the simulated proof is perfect if $\log_{D_1} D_4 = \log_{D_1} D_2 \cdot \log_{D_1} D_3$ and that the whole transcript contains no information about U_0 or U_1 if the relation does not hold.

Finally, $\mathcal{A}$ outputs guess bit b. If $b = \hat{b}$, $\mathcal{S}$ confirms that there exists a, b, such that $D_2 = D_1^a$, $D_3 = D_1^b$, and $D_4 = D_1^{ab}$. Otherwise, it confirms that no such pair of (a, b) exists.

If $\mathcal{A}$ wins the game with probability $1/2 + \epsilon$, we show that $\mathcal{S}$ solves the DDH problem with probability $1/2 + \epsilon/4$.

If there exists (a, b) such that $D_2 = D_1^a$, $D_3 = D_1^b$, and $D_4 = D_1^{ab}$. In this case, the probability that $\mathcal{S}$ answers correctly is $\Pr[\mathcal{A}\text{ wins}|\hat{b} = 0] + \Pr[\mathcal{A}\text{ wins}|\hat{b} = 1]$. Since, in this case, the simulation is perfect, the probability is $1/2(1/2 + \epsilon) + 1/2(1/2 + \epsilon) = 1/2 + \epsilon$.

On the other hand, if there is no (a, b) such that $D_2 = D_1^a$, $D_3 = D_1^b$, and $D_4 = D_1^{ab}$. In this case, the probability that $\mathcal{S}$ answers correctly is $\Pr[\mathcal{A}\text{ loses}|\hat{b} = 0] + \Pr[\mathcal{A}\text{ loses}|\hat{b} = 1]$. In this case, the simulation is perfect when $\hat{b} = 0$; thus, the former probability is $1/2(1/2 - \epsilon)$. On the other hand, when $\hat{b} = 1$, the probability that $\mathcal{A}$ wins is exactly $1/2$ since the challenge contains no information on the bit $\hat{b}$. Thus, the probability for the latter is $1/2 \cdot 1/2$. Thus, in this case, the probability is $1/2(1/2 - \epsilon) + 1/2 \cdot 1/2$, which is $1/2 - \epsilon/2$.

In summary, the probability of $\mathcal{S}$ answering correctly (recall that, in the DDH problem specification, the problem instance that comes from each distribution with probability $1/2$) is $1/2(1/2 + \epsilon) + 1/2(1/2 - \epsilon/2) = 1/2 + \epsilon/4$.

In other words, if $\mathcal{A}$ can win the game with probability ϵ better than random guessing, $\mathcal{S}$ can solve the DDH problem with probability $\epsilon/4$ better than random guessing. This completes the proof. ■

C) Correct Tracing: Correct tracing consists of two aspects, namely, slandering and hiding. We first define slandering through the following game between attacker $\mathcal{A}$ and challenger $\mathcal{C}$.

1) System Parameter: $\mathcal{C}$ creates and publishes the system parameter $\mathtt{param}$ and the secret key γ. $\mathcal{C}$ also creates the public/secret key of on behalf of the judge. The secret key of the supplier and the judge are given to $\mathcal{A}$.

2) Interactions: $\mathcal{A}$ can make the following four types of interaction freely with $\mathcal{C}$, who acts on behalf of the honest users. $\mathcal{C}$ keeps tracks of the honest user with set $\mathcal{J}$, which is empty initially.

1) *In the registration process* ($j \in \{0,1\}^*$). $\mathcal{A}$ interacts with $\mathcal{C}$, who acts on behalf of U_j in the registration protocol. The index j is specified by $\mathcal{A}$ and will be used to refer to this user in the future. $\mathcal{C}$ conducts $\mathcal{J} = \{j\} \cup \mathcal{J}$. We assume that j is unique for each registration query.[7]

[7]This can be enforced by rejecting subsequent registration request on index J.

2) *In the charging process* (j). $\mathcal{A}$ interacts with $\mathcal{C}$, who acts on behalf of U_j in the charging protocol of value v for user $j \in \mathcal{J}$ specified by $\mathcal{A}$.
3) *In the discharging process* (j). $\mathcal{A}$ interacts with $\mathcal{C}$, who acts on behalf of U_j in the top-up protocol of value v for user $j \in \mathcal{J}$ specified by $\mathcal{A}$.
4) *In the statement process* (j). $\mathcal{A}$ interacts with $\mathcal{C}$, who acts on behalf of U_j in the statement protocol of value d. The value $j \in \mathcal{J}$ is specified by $\mathcal{A}$.

3) Challenge: $\mathcal{A}$ chooses a type of interaction, either tracing, charging, or discharging, and conducts the interaction as a malicious user. If the interaction is charging or discharging, $\mathcal{A}$ further outputs an index $j \in \mathcal{J}$.

4) Winning: If the interaction in the challenge phase is tracing, $\mathcal{A}$ wins if the challenger, playing the role of an honest judge, outputs any of the transactions conducted by an honest user. If the interaction is charging or discharging, $\mathcal{A}$ further outputs $j \in \mathcal{J}$. $\mathcal{C}$ releases the tracing information on user U_j, and $\mathcal{A}$ wins if the tracing information would like to conduct the charging or discharging transaction in the challenge phase. Next, we define hiding through the following game between attacker $\mathcal{A}$ and challenger $\mathcal{C}$.

5) System Parameter: $\mathcal{C}$ creates and publishes the system parameter param and keeps the secret key private. $\mathcal{C}$ also creates the public/secret key of on behalf of the judge. The secret key of the judge is also given to $\mathcal{A}$.

6) Interactions: $\mathcal{A}$, acting as a malicious user, can make the following four types of interactions freely with $\mathcal{C}$.

1) *In the registration process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the registration protocol.
2) *In the charging process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the charging protocol of value v. Upon successful completion of the protocol, W is decreased by the value v.
3) *In the discharging process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the top-up protocol of value v. Upon successful completion of the protocol, W is increased by the value v.
4) *In the statement process*. $\mathcal{A}$ interacts with $\mathcal{C}$ in the statement protocol of value d. Upon successful completion of the protocol, W is increased by the value d.

7) Eavesdropping: $\mathcal{A}$ can also observe the four types of interactions between honest users with $\mathcal{C}$. We further assume that the user index j to be observed is to be specified by $\mathcal{A}$.

8) Winning: Suppose $\mathcal{A}$ has made n registration query. $\mathcal{A}$ wins the game if it can conduct one charging or discharging with $\mathcal{C}$ and n trace transaction with $\mathcal{C}$ so that the n pieces of tracing information obtained by $\mathcal{C}$ in the trace transaction do not link to the charging or discharging transaction conducted by $\mathcal{A}$.

Proof: Below, we outline how this is infeasible for the attacker to slander an honest user or to hide according to the given definitions.

1) Slandering. The attacker releases a piece of tracing information to an honest judge so that it would link to the transactions conducted by an honest user. It is straightforward to show that the attacker has to compute the discrete logarithm of the value E from a transaction conducted by an honest user, which is infeasible under the discrete logarithm assumption.
2) Slandering. The attacker conducts a transaction so that when an honest user releases his tracing information, it would be linked to the transaction conducted by the attacker. This requires the attacker to be able to conduct a zero-knowledge proof of the tracing information in $\mathcal{PK}_2$ or $\mathcal{PK}_3$. However, it is infeasible for an attacker to compute the tracing information on an honest user under the discrete logarithm assumption.
3) Hiding. The attacker controlling n users have at most n different "valid" pieces of tracing information. To conduct a transaction that cannot be traced, the attacker has to produce a forged account (which has a different tracing information). This is not feasible under the assumption that the underlying BBS+ signature is unforgeable, with the soundness of $\mathcal{PK}_2$, $\mathcal{PK}_3$, and $\mathcal{PK}_4$. ■

References

[1] EZ-link. [Online]. Available: http://www.ezlink.com.sg

[2] Octopus Hong Kong. [Online]. Available: http://www.octopus.com.hk

[3] Oyster online. [Online]. Available: https://oyster.tfl.gov.uk/oyster/entry.do

[4] R. S. Anand and C. E. V. Madhavan, "An online, transferable E-cash payment system," in *Proc. INDOCRYPT*, vol. 1977, *Lecture Notes in Computer Science*, 2000, pp. 93–103.

[5] M. H. Au and A. Kapadia, "PERM: Practical reputation-based blacklisting without TTPS," in *Proc. ACM Conf. Comput. Commun. Security*, T. Yu, G. Danezis, and V. D. Gligor, Eds., 2012, pp. 929–940.

[6] M. H. Au, A. Kapadia, and W. Susilo, "BLACR: TTP-free blacklistable anonymous credentials with reputation," in *Proc. NDSS*, 2012.

[7] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *Proc. SCN*, vol. 4116, *Lecture Notes in Computer Science*, R. D. Prisco and M. Yung, Eds., 2006, pp. 111–125.

[8] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy in mobile computing for location-sharing-based services," in *Proc. PETS*, vol. 6794, *Lecture Notes in Computer Science*, 2011, pp. 77–96.

[9] O. Blazy, S. Canard, G. Fuchsbauer, A. Gouget, H. Sibert, and J. Traoré, "Achieving optimal anonymity in transferable E-cash with a judge," in *Proc. AFRICACRYPT*, vol. 6737, *Lecture Notes in Computer Science*, 2011, pp. 206–223.

[10] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Proc. EUROCRYPT*, C. Cachin and J. Camenisch, Eds., 2004, pp. 56–73.

[11] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. CRYPTO*, vol. 3152, *Lecture Notes in Computer Science*, 2004, pp. 41–55.

[12] E. F. Brickell, P. Gemmell, and D. W. Kravitz, "Trustee-based tracing extensions to anonymous cash and the making of anonymous change," in *Proc. SODA*, 1995, pp. 457–466.

[13] J. Camenisch, "Group signature schemes and payment systems based on the discrete logarithm problem," Ph.D. dissertation, Swiss Fed. Inst. Technol. Zurich, Zurich, Switzerland, 1998.

[14] J. Camenisch, R. Chaabouni, and A. Shelat, "Efficient protocols for set membership and range proofs," in *Proc. ASIACRYPT*, vol. 5350, *Lecture Notes in Computer Science*, J. Pieprzyk, Ed., 2008, pp. 234–252.

[15] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proc. CRYPTO*, vol. 3152, *Lecture Notes in Computer Science*, 2004, pp. 56–72.

[16] J. Camenisch, J.-M. Piveteau, and M. Stadler, "An efficient fair payment system," in *Proc. ACM Conf. Comput. Commun. Security*, 1996, pp. 88–94.

[17] J. Camenisch and V. Shoup, "Practical verifiable encryption and decryption of discrete logarithms," in *Proc. CRYPTO*, vol. 2729, *Lecture Notes in Computer Science*, D. Boneh, Ed., 2003, pp. 126–144.

[18] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups (Extended Abstract)," in *Proc. CRYPTO*, vol. 1294, *Lecture Notes in Computer Science*, 1997, pp. 410–424.

[19] S. Canard, A. Gouget, and J. Traoré, "Improvement of efficiency in (unconditional) anonymous transferable e-cash," in *Proc. Financial Cryptogr.*, vol. 5143, *Lecture Notes in Computer Science*, 2008, pp. 202–214.

[20] B. Carbunar, W. Shi, and R. Sion, "Conditional e-payments with transferability," *J. Parallel Distrib. Comput.*, vol. 71, no. 1, pp. 16–26, Jan. 2011.
[21] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," in *Proc. CRYPTO*, vol. 403, *Lecture Notes in Computer Science*, 1988, pp. 319–327.
[22] D. Chaum and T. P. Pedersen, "Transferred cash grows in size," in *Proc. EUROCRYPT*, vol. 658, *Lecture Notes in Computer Science*, 1992, pp. 390–407.
[23] M. Chia, S. Krishnan, and J. Zhou, "Challenges and opportunities in infrastructure support for electric vehicles and smart grid in a dense urban environment," in *Proc. IEEE Int. Elect. Veh. Conf.*, 2012, pp. 1–6.
[24] A. die Solages and J. Traoré, "An efficient fair off-line electronic cash system with extensions to checks and wallets with observers," in *Proc. Financial Cryptogr.*, vol. 1465, *Lecture Notes in Computer Science*, 1998, pp. 275–295.
[25] M. Duckham, "Moving forward: Location privacy and location awareness," in *Proc. SPRINGL*, 2010, pp. 1–3.
[26] C.-I. Fan and V. S.-M. Huang, "Provably secure integrated on/off-line electronic cash for flexible and efficient payment," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 5, pp. 567–579, Sep. 2010.
[27] S. N. Foley, "Using trust management to support transferable hash-based micropayments," in *Proc. Financial Cryptogr.*, vol. 2742, *Lecture Notes in Computer Science*, 2003, pp. 1–14.
[28] D. M. Freeman, "Converting pairing-based cryptosystems from composite-order groups to prime-order groups," in *Proc. EUROCRYPT*, vol. 6110, *Lecture Notes in Computer Science*, 2010, pp. 44–61.
[29] J. Freudiger, R. Shokri, and J.-P. Hubaux, "Evaluating the privacy risk of location-based services," in *Proc. Financial Cryptogr.*, vol. 7035, *Lecture Notes in Computer Science*, 2012, pp. 31–46.
[30] G. Fuchsbauer, D. Pointcheval, and D. Vergnaud, "Transferable constant-size fair e-cash," in *Proc. CANS*, vol. 5888, *Lecture Notes in Computer Science*, 2009, pp. 226–247.
[31] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *J. ACM*, vol. 38, no. 3, pp. 690–728, Jul. 1991.
[32] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, Apr. 1984.
[33] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM J. Comput.*, vol. 18, no. 1, pp. 186–208, Feb. 1989.
[34] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Proc. Pervasive*, vol. 5538, *Lecture Notes in Computer Science*, 2009, pp. 390–397.
[35] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," *IEEE Pervasive Comput.*, vol. 5, no. 4, pp. 38–46, Oct./Dec. 2006.
[36] S. Jarecki and A. M. Odlyzko, "An efficient micropayment system based on probabilistic polling," in *Proc. Financial Cryptogr.*, vol. 1318, *Lecture Notes in Computer Science*, 1997, pp. 173–192.
[37] I. R. Jeong, D. H. Lee, and J. I. Lim, "Efficient transferable cash with group signatures," in *Proc. ISC*, vol. 2200, *Lecture Notes in Computer Science*, 2001, pp. 462–474.
[38] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable signatures," in *Proc. EUROCRYPT*, C. Cachin and J. Camenisch, Eds., 2004, pp. 571–589.
[39] M. Lesk, "Micropayments: An idea whose time has passed twice?" *IEEE Security Privacy*, vol. 2, no. 1, pp. 61–63, Jan./Feb. 2004.
[40] L. Liao, D. J. Patterson, D. Fox, and H. A. Kautz, "Learning and inferring transportation routines," *Artif. Intell.*, vol. 171, no. 5/6, pp. 311–331, Apr. 2007.
[41] R. J. Lipton and R. Ostrovsky, "Micropayments via efficient coin-flipping," in *Proc. Financial Cryptogr.*, vol. 1465, *Lecture Notes in Computer Science*, 1998, pp. 1–15.
[42] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Enhancing location privacy for electric vehicles (at the right time)," in *Proc. ESORICS*, vol. 7459, *Lecture Notes in Computer Science*, 2012, pp. 397–414.
[43] B. Lynn, The Java Pairing Based Cryptography Library (jPBC) 2010. [Online]. Available: http://libeccio.dia.unisa.it/projects/jpbc/
[44] T. Okamoto, "An efficient divisible electronic cash scheme," in *Proc. CRYPTO*, vol. 963, *Lecture Notes in Computer Science*, 1995, pp. 438–451.
[45] T. Okamoto and K. Ohta, "Universal electronic cash," in *Proc. CRYPTO*, vol. 576, *Lecture Notes in Computer Science*, 1992, pp. 324–337.
[46] Send Mondy, Pay Online and Merchant Accounts. [Online]. Available: https://www.paypal.com
[47] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. CRYPTO*, vol. 576, *Lecture Notes in Computer Science*, J. Feigenbaum, Ed., 1992, pp. 129–140.
[48] S.-M. Yen, K.-Z. Chiou, J. Zhang, and P.-H. Lee, "A new peer-to-peer micropayment protocol based on transferable debt token," in *Transactions on Computational Science X*. New York, NY, USA: Springer-Verlag, 2010, pp. 352–363.

Man Ho Au (M'12) received the Ph.D. degree from the University of Wollongong, Wollongong, Australia, in 2009.

He is currently an Associate Lecturer with the School of Computer Science and Software Engineering, University of Wollongong. He is the author of over 50 referred research papers presented at international conferences. His research interests include information security, privacy, and cryptography.

Dr. Au has served as a Workshop Chair, a Publication Chair, and a member of program and organizing committees of around ten international conferences.

Joseph K. Liu received the Ph.D. degree in information engineering with specialization on cryptographic protocols for securing wireless networks, privacy, authentication, and provable security from the Chinese University of Hong Kong, Shatin, Hong Kong, in July 2004.

He is currently a Research Scientist with the Infocomm Security Department, Institute for Infocomm Research, Singapore. His current research interests include lightweight cryptography, wireless security, and security in smart-grid systems and cloud computing environments.

Junbin Fang received the Ph.D. degree from South China Normal University, China, in 2008.

He is currently an Associate Professor with the Department of Optoelectronic Engineering, Jinan University, Guangzhou, China. His research interests include information security and forensics and quantum cryptography.

Zoe L. Jiang received the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2010.

She is currently an Assistant Researcher with the Computer Application Research Center, School of Computer Science and Technology, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. Her research interests include applied cryptography and digital forensics.

Willy Susilo (SM'01) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia.

He is currently the Director of the Center for Computer and Information Security Research and a Professor with the School of Computer Science and Software Engineering, University of Wollongong. He is the author of numerous papers in the area of digital signature and encryption schemes. His main research interests include cryptography and information security and digital signature schemes.

Dr. Susilo has served as a member of the program committees of several international conferences. He received the prestigious Australian Research Council Future Fellow Award.

Jianying Zhou received the Ph.D. degree in information security from the University of London, London, U.K., in 1997.

He is a Senior Scientist and the Head of the Infocomm Security Department with the Institute for Infocomm Research, Singapore. His research interests include computer and network security and mobile and wireless communications security.

Dr. Zhou is a Founder and a member of the Steering Committee of the International Conference on Applied Cryptography and Network Security.