1-1-2014

A new payment system for enhancing location privacy of electric vehicles

Man Ho Au
University of Wollongong, aau@uow.edu.au

Joseph K. Liu
Institute for Infocomm Research, Singapore

Junbin Fang
Jinan University

Zoe L. Jiang
Harbin Institute of Technology

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

See next page for additional authors

Recommended Citation

Au, Man Ho; Liu, Joseph K.; Fang, Junbin; Jiang, Zoe L.; Susilo, Willy; and Zhou, Jianying, "A new payment system for enhancing location privacy of electric vehicles" (2014). *Faculty of Engineering and Information Sciences - Papers: Part A*. 1913.
https://ro.uow.edu.au/eispapers/1913

A new payment system for enhancing location privacy of electric vehicles

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.

Keywords

electric, privacy, vehicles, location, payment, enhancing, system

Disciplines

Engineering | Science and Technology Studies

Publication Details

Au, M. Ho., Liu, J. K., Fang, J., Jiang, Z. L., Susilo, W. & Zhou, J. (2014). A new payment system for enhancing location privacy of electric vehicles. IEEE Transactions on Vehicular Technology, 63 (1), 3-18.

Authors

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

A New Payment System for Enhancing Location Privacy of Electric Vehicles[*]

Man Ho Au, Joseph K. Liu[△], Junbin Fang, Zoe L. Jiang[◇], Willy Susilo[♮], Jianying Zhou

Abstract

An electric vehicle 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 of battery recharging, the cars will be required to visit 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 electric vehicles. Our system not only supports privacy protection (location privacy), but also supports traceability in the case where the cars are stolen. Our system can further support the future vehicle-to-grid paradigm. In addition, we prove the security and produce a proof-of-concept prototype to enumerate our system.

I. Introduction

An Electric Vehicle (also known as EV) is a vehicle that does not rely on gasoline or liquid petroleum gas (LPG) as fuel, but only uses electricity stored inside the car battery as the source of kinetic energy, and hence, it offers emission-free urban transportation. It uses electric motor instead of 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 EV. For example, EV does not emit harmful tailpipe pollutants from the onboard source of power at the point of operation (zero tail pipe emissions). If the country uses renewable energy (such as solar or hydroelectricity), EV becomes a renewable form of transportation.

Future EV may support Vehicle-to-grid (V2G) system. This concept allows vehicle to provide power (sell electricity) to the grid. This can be done if the the EV is equipped with a solar panel and parked outside with sunshine. In this way, EV can help to balance loads by "valley filling" (charging at night when the demand is low)

[△] Corresponding Author

[◇] Zoe L. Jiang is supported by The National Natural Science Foundation of China (NSFC)(61240011) and Shenzhen Strategic Emerging Industries Program (ZDSY20120613125016389).

[♮] Willy Susilo is supported by the ARC Future Fellowship (FT0991397).

[*] The extended abstract of this version was published in [30]. Joseph K. Liu and Jianying Zhou are with Institute for Infocomm Research, Singapore; Man Ho Au and Willy Susilo are with University of Wollongong, Australia; Junbin Fang is with Jinan University, Guangzhou, China; Zoe L. Jiang is with Harbin Institute of Technology Shenzhen Graduate School, and Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen, China.

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 provide spinning reserves (meet sudden demands for power).

In the US, EV has 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 traditonal gasoline-based car. On average, fueling a gasoline-based car wll cost roughly 3 times more than fueling with electricity, in the case of EV. In addition to saving money, EV also offers significant environmental benefits, which makes the adoption of EV become very attractive.

Despite their potential benefits, widespread adoption of electric vehicles faces several hurdles and limitations. One of the major problems is the driving range. Most electric vehicles can only go about 100 to 150 km before recharging, while 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, a worry 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 electric vehicle to 80 percent in about 30 minutes. Electric vehicle drivers may then charge their vehicles at their homes, offices, shopping malls or car parks outside restaurants when they are having dinner.

LOCATION PRIVACY CONCERN. The luxury of charging electric vehicles at the drivers’ comfort also comes with some drawbacks. In practice, EV needs to travel at a certain time between two charging stations. As mentioned earlier, since the distance is relatively much shorter compared to gasoline-based cars, this will lead to some issues, related to location privacies [18]. These locations include the drivers’ living places, working companies, the amusement places they usually go, and so forth [27], [29], [26]. Leaking privacy will directly produce negative impacts [19], [4], [21], 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 that individuals location. 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 persons such as the robbers for stalking or physical attacks. Therefore, location privacy issues must be addressed carefully before EV 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 re-filled within a short distance and therefore, by merely tracking the distance between the last charging station and the next one will not reveal any useful information. In practice, drivers may need to re-fill 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 re-fill activities will not be very frequent, and as highlighted earlier, the activities after the car has been re-filled will remain unknown. This is in contrast to EV. Moreover, since EV can support V2G charging, the location of the last re-fill can even be easily recorded.

We will examine other payment systems and their impact on location privacy in Section I-B.

REVOCATION OF LOCATION PRIVACY AT THE "RIGHT" TIME. It is clear that EV requires protection against location privacy. Nevertheless, cautions 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 car is stolen. The owner will definitely be interested to know the location of the stolen car, and hence with the aim to retrieve it back at a later stage. Obviously, some anti-theft or thief-tracing devices can be installed in the car (e.g. GPS with 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 these 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 anti-theft device compared to the cost of the car itself is just negligible. However, for some lower-end used cars, it is impractical to install such 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. Essetially, it is required that the location privacy can be revoked at the *right* time.

As outlined earlier, the disadvantage of EV 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 EV. As the vehicle is required to be re-charged very frequently, charging stations can be used to trace any stolen vehicle. If a stolen car is being re-charged 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, and hence, the cars will be stopped.

A. *Summary of System Requirement*

In summary, the unique aspect of a payment system for electric vehicles includes the following:

1) Support two-way transactions. Users in the system could act as the payer as well as 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.

Below 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 ones and explain why they are not suitable for electric vehicles.

- **Paper cash**: Different from gas stations, charging stations for electric vehicles are all machine operated. If they allow cash payment, the installation costs will be very high due to high security requirement of cash machine (similar to those for ATM). Note that currently there are many ticketing machine installed in car parks or automatic selling machines (e.g. selling soft drink) which can accept paper cash or coins. However, as the cost for car park or soft drink is far less than charging electric vehicles, 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 disfavoured by supplier to adopt paper cash as a kind of payment system in the charging station.
- **E-cash**: Alternatively, e-cash is the electronic form of paper cash which also provides anonymity. However, e-cash is mainly used in small amount transaction (e.g. a few dollars) instead of large amount transaction (e.g. a few hundred dollars) due to security and efficiency concerns. In order to support two-way payments, transferable e-cash is needed and it has been shown complexity of transferable e-cash grows linearly in the number of transfer supported [17][1]. Apart from that, off-line e-cash cannot provide double-spending *prevention*. It can only *detect* double-spending and *reveal* the identity of the double-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 card, 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 transaction. Charging for an electric vehicle definitely does not 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 take the role of payees as well in the V2G paradigm.
- **Prepaid cash card or cash coupon**: Prepaid cash card or cash coupon is another common way of anonymous e-payment. However, similar to e-cash, it does not support lost protection. Executing large amount transaction may bring inconvenience to user: They may neither want to bring many coupons together, nor buy the coupons or

[1] A recent approach achieve constant size transferable e-cash, at the expense that the user storage is linear to the number of his spent coins [22].

topup everyday. In addition, it also does not fully support 2-ways transactions, which is a necessary requirement for the future Vehicle-to-grid system.

- **Paypal**: Paypal is a kind of most commonly used electronic prepaid system. However, it requires a third party (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.
- **Credit card**: Credit card is a widely adopted payment system for large amount transaction instead. It also supports 2-ways transaction. Nevertheless, credit card is not anonymous. Due to the frequent charging requirement for electric vehicles, location privacy will be lost by tracing the credit card payment easily.

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

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]	✓	×	✓	×
Credit card	×	✓	✓	✓	✓	✓	×
PayPal	✓×	✓	✓	✓	✓	✓	×
Our system	✓	✓	✓	✓	✓	✓	✓

[a] Most of the existing e-cash systems do not support judge, though some of them (e.g. [8], [13], [5], [16]) do support judge.

C. Our Contributions

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

TWO-WAY ANONYMOUS PAYMENT: It supports anonymous payment *in both directions*. First, the electric vehicle remains anonymous when it re-charges at any charging station. It further supports 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 *in normal operations*.

TRACEABILITY OF STOLEN CAR: If the electric vehicle is stolen, the owner may provide some secret information to charging stations so that next time when the stolen car is being re-charged at any charging station, its location will be revealed. Note that if the thief never re-charges 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 favoured by users or supplier:

1) **Prevention of Cheating User**: Different from e-cash which cannot prevent users from cheating or double-spending (it can only *detect* such behaviour), our payment system supports prevention of any cheating behaviour. 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 user is explained in Section I-B.)
2) **Support Judging Authority (JA)**: In case 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 Judging Authority. 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 ATM). 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 which runs on a smart phone with reasonable performance.
4) **Lost Protection**: While our system is software-based, lost protection is also supported. That is, if the secret of the user is stolen, the user can report the incident by supplying some secret information, which prevents the secret from being usable. Furthermore, the user can regain his un-used credit by providing some authenticated information.

D. Enhancement over our work appeared at the conference in [30]

In this version, we have significantly improved the payment system protocol with the following enhancements, summarized as below:

1) We enhance the Judge Open function. In our previous work [30], the judge can open user's transaction at its own will. We re-design the algorithm here so that the Judge cannot open any user transaction by itself. Instead, it needs to get a kind of "consent" from user before opening. This is to prevent the Judge from having too much power, as the case of credit card or PayPal[2].
2) The Judge Open algorithm in our previous work [30] is inefficient. It requires composite order pairing which is not practical to be implemented in mobile device. In this version, the user supplies his/her consent in the form of some secret value. Given the secret value, the judge can tell efficiently if a transaction is conducted by the user. Here the judge only needs to carried out one exponentiation for the testing. The whole system

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

can be constructed using prime order pairing, which is at least 50 times faster than composite order pairing [20].

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

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

II. System Architecture

A. Entities

We consider a system which is composed of the following entities:

1) **User**: A user refers to an electric vehicle, which is implicitly referred to an In-Car-Unit device. More details will be described in the next sub-section.
2) **Supplier**: It refers to the power grid company. It supplies (sells) electricity to the cars, and also collects (buys) electricity back 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.
3) **Judging Authority (JA)**: It is responsible to investigate into some disputed transactions between user and power company. It has the power to trace all transactions made by a specific user given the user's consent. It maybe the government authority or the court.

B. Overall Structure

We briefly describe the overall structure of our system. We assume 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 as well as modify its memory (except the read-only memory).

Our system consists of the following protocols, and the scenario related to charging is depicted in Fig. 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. The 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 wifi, bluetooth or direct cable connection.
- Charging: The In-Car-Unit carries an interactive protocol with the charging station, which first checks with the grid management server to confirm the grid capacity is fine. If the price is dynamic (if it is within peak period

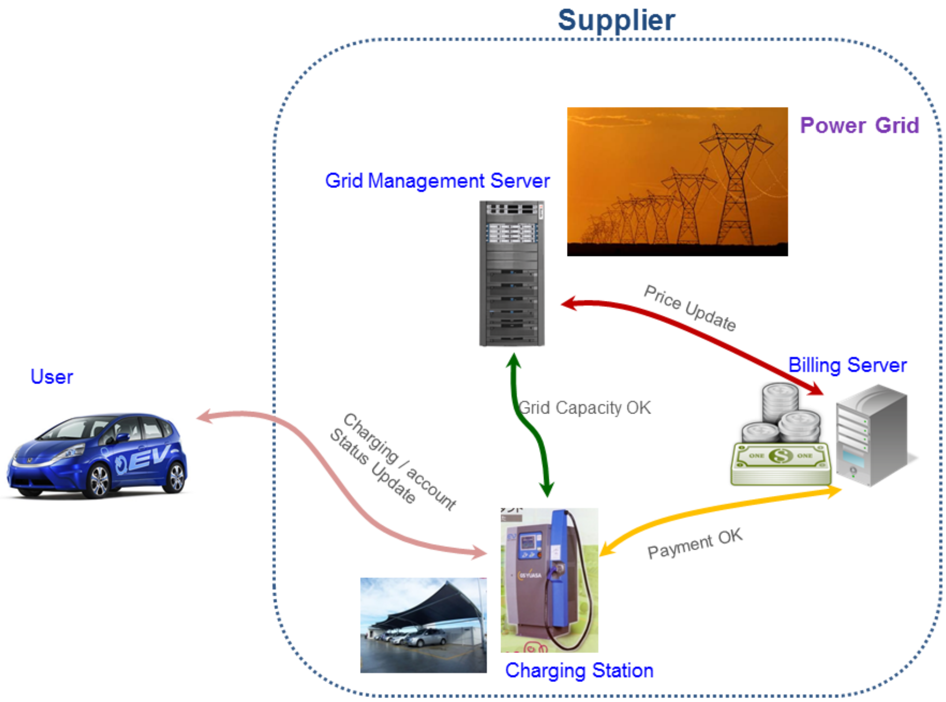

Fig. 1: Charging/Topup Scenario

the price maybe 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 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 wifi or bluetooth.

- Discharging or Topup: 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.
- Statement: Every statement period, the user approaches the supplier to topup the balance to make it D again. In practice, the In-Car-Unit may communicate with the supplier through wifi or bluetooth, if the parking place contains wifi or bluetooth connectivity; or using GSM data connection directly.
- 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.
- 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 depends on some kind of external

assumptions. In order to enjoy lost protection, we assume the user will backup the content of his In-Car-Unit in a safe and independent location. In case 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 yes, it will block any party from using this content. The user can claim back the remaining balance. If we further assume each car is equipped with exactly one In-Car-Unit, the unit is not replaceable and that the read-only memory is not modifiable, this algorithm allows the user to block any re-charging for the lost car. In fact, should the thief tried to re-charging 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 is detailed below. Note the the adversary in each of the listed attackers could be an insider, or a collusion of insiders in the system. For instance, in an attempt to breach location privacy of an honest user, the adversary is assume to be controlling a set of dishonest users, the supplier as well as the judge.

1) *Location Privacy Infringement.* The attacker tries to track transactions of an honest users. We consider a powerful adversary which 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 as well as the judge and its goal is to decide if a a given payment (or top-up) transaction belongs to one of the two honest users. We further assume 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 follows physically a certain electric vehicle, eavesdrops all its transactions and have access to the private key of the supplier as well as the judge, the attacker still cannot tell if another payment (or top-up) transaction belongs to this specific user or another honest user.
2) *Statement Fraudulent.* 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, suppose the adversary have deposited n dollars to the supplier, where 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] 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.

3) *Slandering.* The attacker tries to slander an honest user in two ways. It could be a registered but malicious user and who releases a piece of tracing information to the judge so that the judge would link transactions from an honest user. As an other 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 un-cooperative users are to be deal with. In this attack, however, the malicious user appears to be cooperative yet its goal is to make some of his/her transactions untraceable. The attacker is supposed to have access to the judge secret key and can eavesdrops transactions conducted by the other users. As a generalized notion, we consider the case when n users are controlled by the adversary. The adversary send 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 above threats. Since both slandering and hiding are related to the correctness of the tracing algorithm, we say a payment scheme offers correct tracing if it is secure against both slandering and hiding attack. Likewise, we say a payment scheme prevent cheating users if it is secure against statement fraudulent. In Appendix B, we shall formally define the security and privacy requirements of location privacy, prevention of cheating users and correct tracing using a computational approach which have been commonly used in the cryptographic community since 1984 [24].

III. Primitives

In this section we first review some cryptographic primitives and number-theoretic assumptions that will be used.

a) 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) *Bilinearity*: For all $g, h \in \mathbb{G}$, and $a, b \in \mathbb{Z}_p$, $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$.
2) *Non-degeneracy*: There exists $g \in \mathbb{G}$ such that $\hat{e}(g, g)$ has order p in $\mathbb{G}_T$.

3) *Computability*: It is efficient to compute $\hat{e}(g,h)$ for all $g,h \in \mathbb{G}$.

b) Decisional Diffie-Hellman (DDH) Assumption: Given a cyclic group $\mathbb{G}_p = \langle \mathfrak{g} \rangle$ of prime order p, we say that the 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) randomly picked from the former distribution and 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 on input a four tuple $(D_1, D_2, D_3, D_4) \in \mathbb{G}$, 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.

c) Commitment: Our system uses the well known commitment scheme due to Pedersen [32]. Let $\mathbb{G}$ be a cyclic group of prime order p and g, h be generators of $\mathbb{G}$. On input a value $x \in \mathbb{Z}_p$, the committer randomly chooses $r \in \mathbb{Z}_p$, 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 $C = g_1^{x_1} \cdots g_n^{x_n} h^r$, 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 a value x (resp. $(x_1, \ldots, x_n)$). Note that this commitment scheme is homomorphic: $\mathsf{CMT}(a) * \mathsf{CMT}(b)$ gives $\mathsf{CMT}(a+b)$ and the opening of the later is the sum of that of the formers.

d) BBS+ Signature: We employ the signature scheme proposed by Au et al. [3], which is based on the schemes of Camenisch and Lysyanskaya [12] and of Boneh et al. [7]. Their scheme, called BBS+ signature, is briefly reviewed here. Let $g, g_0, g_1, g_2, g_3 \in \mathbb{G}$ be generators of $\mathbb{G}$. Let $\hat{e}$ be a pairing defined over $(\mathbb{G}, \mathbb{G}_T)$.

The signer's secret is a 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 $e, y \in_R \mathbb{Z}_p$, computes $A = (g g_0^y g_1^{m_1} g_2^{m_2} g_3^{m_3})^{\frac{1}{\gamma + e}}$. 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:

$$\hat{e}(A, w g^e) \overset{?}{=} \hat{e}(g g_0^y g_1^{m_1} g_2^{m_2} g_3^{m_3}, g)$$

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.

e) Zero-knowledge Proof: A zero-knowledge proof [25] is an interactive protocol for one party, the prover, to prove to another party, the verifier, that some statement is true, without revealing anything other than the veracity of the statement. In [23], it has been shown that, assuming the existence of 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 construction of zero-knowledge proof for statements regarding relationship about discrete logarithms in cyclic group of known order has been proposed [10]. We follow the notation introduced by Camenisch and Stadler [15]. 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 are being proved while 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 and hence, it does not depend on any proprietary hardware. Next, we describe the relationship between the system requirements and the hardware assumption. Note that the major security concern of the supplier, namely, prevention of cheating users, *does not* depends 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/her 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 physical 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.
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 not cover the physical aspect of possible privacy breach. For instance, suppose a physical camera is installed in each charging station and it records the physical identifier of the vehicle (e.g. registration plate number), and therefore, it is obvious that location privacy cannot be maintained. This is analogous to the use of physical money. Suppose the cash register records the image of the payer, then it is always possible to link the payment from the user across different locations, and therefore anonymity is no longer preserved.

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

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

B. Detailed Description

- 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, 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.
- Judge Setup: In our construction, the judge does not have any public/private key.
- 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 the supplier and enrolls into the system as follow.
  1) The user randomly picks $y', s, t' \in_R \mathbb{Z}_p$, computes and sends $C = g_0^{y'} g_3^s g_4^{t'}$ to the supplier, along with the following proof:

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

     $\mathcal{PK}_1$ assures the supplier that the value C is computed correctly. 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 $A = (C g g_0^{y''} g_1^I g_2^D g_4^{t''})^{\frac{1}{e+\gamma}}$ 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 the value t remains unknown to the supplier. We assume 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 $\sigma_s := (A, e, s)$ 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 figure 2.

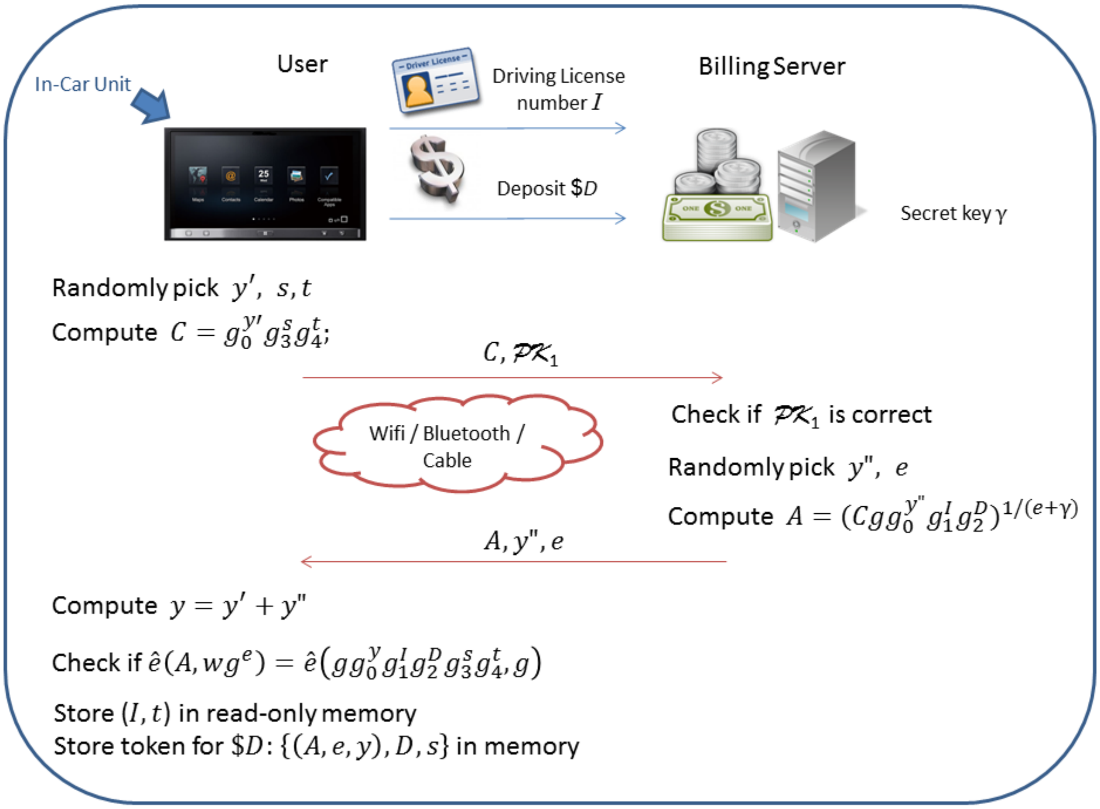
In-Car Unit
User
Driving License number I
Deposit $$D$
Billing Server
Secret key γ
Randomly pick y', s, t
Compute $C = g_0^{y'} g_3^s g_4^t$;
$C, \mathcal{PK}_1$
Check if $\mathcal{PK}_1$ is correct
Wifi / Bluetooth / Cable
Randomly pick y'', e
Compute $A = (Cgg_0^{y''} g_1^I g_2^D)^{1/(e+\gamma)}$
A,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

Fig. 2: Registration

- Charging: Let v be the value of the transaction. Let $\left(\tilde{\sigma_s} := (\tilde{A}, \tilde{e}, \tilde{y}), I, \tilde{B}, \tilde{s}, t\right)$ be the content stored in the In-Car-Unit. It checks if $\tilde{B} - v \geq 0$. Next, they engages in the following protocol.

1) The In-Car-Unit randomly picks $y', s \in_R \mathbb{Z}_p$, $\mathfrak{R} \in_R \mathbb{G}_T$, 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:

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

2) Note that the 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 [28]. The supplier checks that $\tilde{s}$ has never been used[4] and randomly picks $y'',e \in_R \mathbb{Z}_p$, computes $A = (Cgg_0^{y''}g_2^{-v})^{\frac{1}{e+\gamma}}$ and returns (A,y'',e).
3) The user computes $y = y' + y''$, $B = \tilde{B} - v$ 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). Also note that the content of the read-only memory (I,t) remains unchanged.

The charging protocol is shown in figure 3.

- <u>Topup:</u> Let v be the topup value. Let $\big(\tilde{\sigma_s} := (\tilde{A},\tilde{e},\tilde{y}),I,\tilde{B},\tilde{s},t\big)$ be the content stored in the In-Car-Unit. We assume D is the maximum account balance. Next, they engages in the following protocol.

  The In-Car-Unit randomly picks $y',s \in_R \mathbb{Z}_p$, $\mathfrak{R} \in_R \mathbb{G}_T$, 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:

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

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

$$\hat{e}(A,wg^{e}) \overset{?}{=} \hat{e}(gg_0^{y}g_1^{I}g_2^{B}g_3^{s}g_4^{t},g).$$

[4] The practical issue of the checking process will be described in Section V-A.

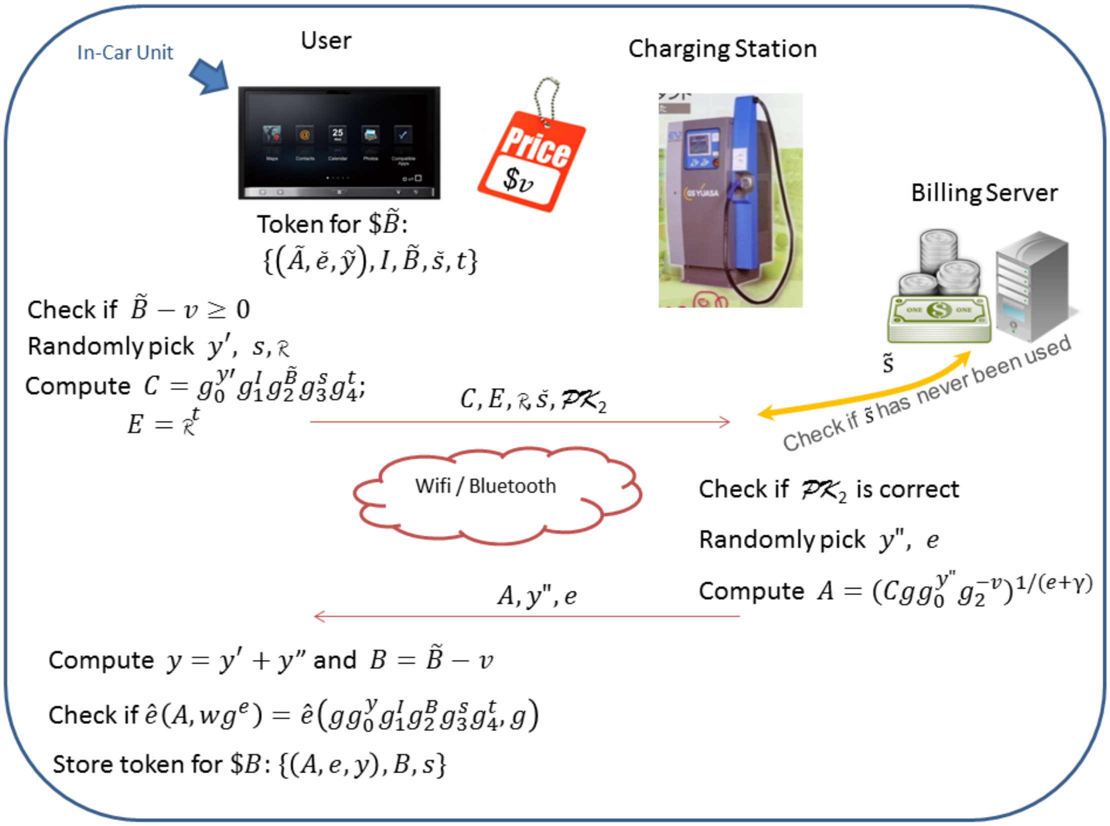
In-Car Unit
User
Charging Station
Price
$v
Token for $\$\tilde{B}$:
$\{(\tilde{A}, \breve{e}, \tilde{y}), I, \tilde{B}, \check{s}, t\}$
Billing Server
Check if $\tilde{B} - v \geq 0$
Randomly pick $y', s, \mathcal{R}$
Compute $C = g_0^{y'} g_1^I g_2^{\tilde{B}} g_3^s g_4^t$;
$E = \mathcal{R}^t$
$\tilde{s}$
$C, E, \mathcal{R}, \check{s}, \mathcal{PK}_2$
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

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). Also note that the content of the read-only memory (I, t) remains unchanged.

- Statement: Let $\left(\tilde{\sigma_s} := (\tilde{A}, \tilde{e}, \tilde{y}), I, \tilde{B}, \tilde{s}, t\right)$ be the contents of the In-Car-Unit. The users pays $v = D - \tilde{B}$ to settle his account. Next, they engages in the following protocol.

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

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

  2) The supplier checks that $\tilde{s}$ has never been used and randomly picks $y'', e \in_R \mathbb{Z}_p$, computes $A = (Cgg_0^{y''} g_1^I g_2^D)^{\frac{1}{e+\gamma}}$ 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 figure 4.

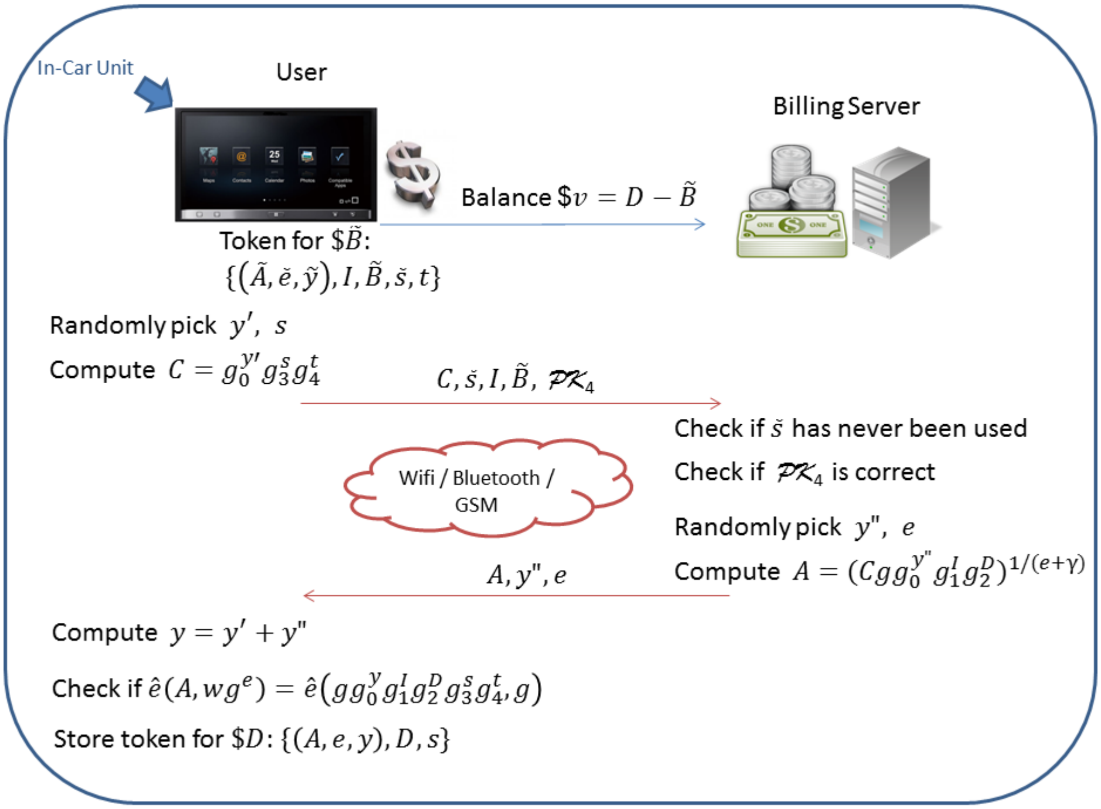
In-Car Unit
User
Billing Server
Balance $\$v = D - \tilde{B}$
Token for $\$\tilde{B}$:
$\{(\tilde{A}, \check{e}, \tilde{y}), I, \tilde{B}, \check{s}, t\}$
Randomly pick y', s
Compute $C = g_0^{y'} g_3^s g_4^t$
$C, \check{s}, I, \tilde{B}, \mathcal{PK}_4$
Check if $\check{s}$ has never been used
Check 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

- Judge Tracing In case 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. In order to present a cheating user from submitting a random t value, the judge shall obtain the registration transcript, which include 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-knowledge 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 submits y', s and t, and the supplier can compute $t = t' + t''$ and checks 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:

$$E \overset{?}{=} \mathfrak{R}^t.$$

For maximum privacy protection, we assume 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 incorporated easily so that the supplier only needs to store the s that have not expired. Let $H : \{0,1\}^* \rightarrow \mathbb{G}$ 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 in param is replaced with the 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 $i = 1$ to 3. 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 topup and the content issued in the previous period will not be usable.

Of course, 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. Fig.5 illustrates our idea. This extension does not alter the efficiency of our system.

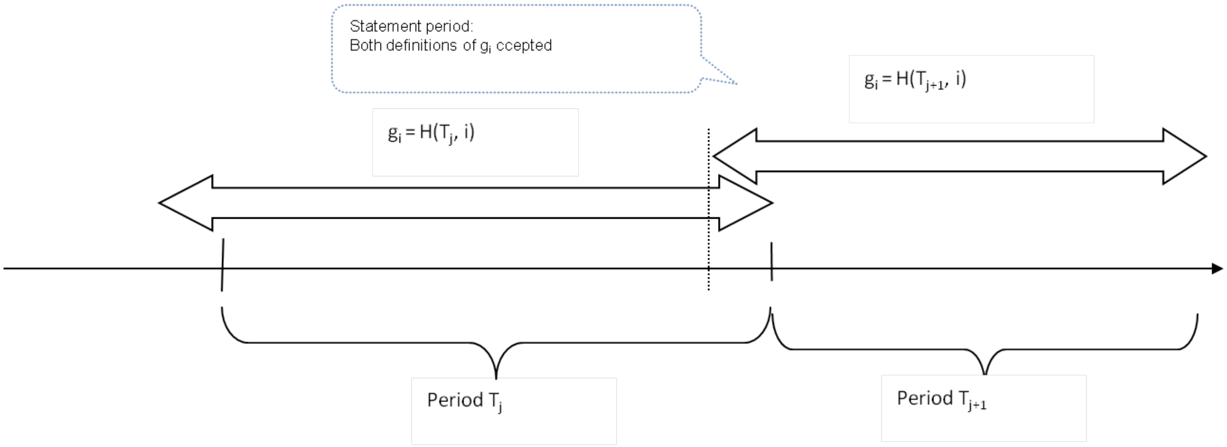

Fig. 5: Timeline demonstrating the expiration mechanism.

B. Dealing with Uncooperative Users in Tracting

While users in our payment system enjoys a very high degree of privacy, sometimes it is necessary for them to be traceable. Indeed, our system incorporate an efficient mechanism for user tracing with the user's consent. It is natural to ask what if the user refuse 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 transaction. This can be done quite easily by requiring the user to verifiably encrypt [14] his/her identity under this powerful entity's public key. However, this solution is undesirable from the users' point of view since the existence of such a powerful entity essentially means that the location privacy of the system is always subjected to the mercy of the entity.

Our scheme can be modified easily to strike a balance between user privacy and accountability based on the extension of credit expiry. For the user who refuse to release his/her tracing information upon the judge's request, the supplier would have the right to refuse to conduct a statement protocol with the user [5]. The user's remaining credit will expiry in the next period and thus his account will no longer be useable. We believe this is one possible compromise. If user choose 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/she 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 the user always backup his latest memory content (A, e, y, I, B, s, t). In case 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 the 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 theft 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 theft.

D. *Incorporating Other Payment Systems*

In the reality, the charging station may also support other payment methods (e.g. credit card). In this case, we require the user to run the charging protocol of our system and pay $0 to the charging station in addition to the other payment method. Specifically, upon completion of the charing protocol of our system, the user may use other form of payment to pay the balance. This is to ensure the traceability of stolen car can still be executed while using other payment systems. However, we note that the other form payment used may leak information about the user.

VI. Practicality Analysis

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

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

A. *Efficiency Analysis*

We analyze the efficiency of our scheme in two ways. The first one is an estimation based on the benchmark from jPBC [31] on the timing of various mathematical operations required in the system implementation. The second one is the actual implementation of various protocols on a current smart phone and a notebook computer.

1) Estimation Based on Benchmark on Various Operations: We first count the number of basic operations required in various protocol and provide an estimation based on the benchmark of the jPBC library for the following devices:

- HTC Desire HD (running Android 2.2, equipped with Qualcomm Snapdragon^{TM} QSD8255 1GHz as the CPU and with 1.5GB ROM). This is used to emulate the In-Car-Unit.
- Desktop (running Ubuntu 10.04, equipped with Intel(R) Core(TM)2 Quad CPU Q6600 2.40GHz, 3 GB RAM). This is used to emulate the server of the supplier.

We only count the time required for exponentiation and pairing. Other operations such as hashing, group addition, integer addition/multiplication etc. are insignificant compared with exponentiation and pairing.

For exponentiation, we further optimize for those bases which are constant. It allows the use of some pre-processed data for faster computation. For pairing, we also optimize for those such that one of the pairing elements is a constant. We put our analyzed result in table III:

TABLE III: Operations required for user and supplier

	Registration		Charging/Topup		Statement	
	User	Supplier	User	Supplier	User	Supplier
Group $\mathbb{G}$ exponentiation (pre-processed)	10	6	30	15	19	10
Group $\mathbb{G}$ exponentiation (no pre-processed)	0	2	4	9	1	4
Group $\mathbb{G}_T$ exponentiation (pre-processed)	0	0	8	9	4	6
Group $\mathbb{G}_T$ exponentiation (no pre-processed)	0	0	2	4	1	2
Pairing (1 element is a constant)	1	0	3	3	2	2
Pairing (both elements are not constant)	1	0	1	1	1	0

We estimate the time based on the number of different operations, using the benchmark result from jPBC. In the benchmark result, supersingular curve $y^2 = x^3 + x$ has been used. The based field size is 512 bits, which is equivalent to 1024 bits discrete logarithm security. The overall result is summarized in table IV.

For storage, the supplier may need to store all transactions for the charging and top processes within a time period. For each process, it may take about 2k bytes. Assume the user charges or topup twice a day. Let each period last for a month. Therefore, the supplier may need to store about 120k bytes for every user in each period.

TABLE IV: Estimated Running Time

	Registration		Charging/Topup		Statement	
	User	Supplier	User	Supplier	User	Supplier
Overall Running Time	1.094 s	0.054 s	3.642 s	0.259 s	2.076 s	0.124 s

Assume there are 1 million users in the system. There will be about 120G bytes transaction data. These stored data are mainly for the judge operation. When the next period comes, the data from the previous period can be deleted. 120G data should be easily stored within a normal harddisk.

During the normal charging/topup operation, the supplier (at the backend) needs to search for a number (selected by the user) in the database to check whether it has already been used within a time period. Using the above assumption (1 month as a time period, there are 1 million users in the system and each user charges/topups twice per day), there will be at most 60 million entries in the database. In practice, this operation will be delegated to the database server (e.g. MySQL) and it is equivalent to a single database query. Modern database server supports 100 Transaction Per Second (TPS) and thus it is reasonable to assume this checking requires less than 0.1 second.

The charging station and the backend server can be connected through Internet connection. As the data transmitted per transaction is just a few thousand bytes, the cost is very low and the transmission time should be very fast.

2) Experimental Results from Our Prototype Implementation: We implement a prototype which supports the user side computation for protocol Registration, Charging/Topup and Statement. Below 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 Qualcomm Snapdragon$^{\text{TM}}$ S4 Pro CPU 1.5 GHz Quad Core 2 GB RAM. For the supplier side, we use a notebook with Core i5-3320 running 2.6GHz and 8GB RAM. The parameter is the same as those specified in our estimation based on the benchmark on jPBC. The result is summarized in Table V.

Note that experimental results is several times slower than the analytical estimation. The main reason is that we do not employ any optimization techniques including pre-generation of random numbers and pre-processing of exponentiations. Nonetheless, the prototype shows that our proposal is practical.

TABLE V: Implemented Running Time

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

B. Security Analysis

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

- **Prevention of Cheating User**: If a user does not follow the designated algorithm to modify his credential (he may intend to do so in order to get more than what he deserves, or pay less than what he should pay), he cannot pass the authentication process.
- **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.
- **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 a probabilistic polynomial time adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The game are defined so that it captures the capabilities and behavior of an adversary. The adversary winning the game would imply 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 presented a mechanism to enhance location privacy for electric vehicles. Our proposed solution provides an anonymous payment system with privacy protection support. In the case where traceability is required, such as when the electric vehicle is stolen, this feature can also be provided. Hence, our solution provides location privacy enhancement at the right time, which will make the adoption of electric vehicles practical.

Our system provides an option to incorporate a judge who can open all transactions in 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.

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

References

[1] M. H. Au and A. Kapadia. Perm: practical reputation-based blacklisting without ttps. In T. Yu, G. Danezis, and V. D. Gligor, editors, *ACM Conference on Computer and Communications Security*, pages 929–940. ACM, 2012.

[2] M. H. Au, A. Kapadia, and W. Susilo. BLACR: TTP-Free Blacklistable Anonymous Credentials with Reputation. In *NDSS*. The Internet Society, 2012.

[3] M. H. Au, W. Susilo, and Y. Mu. Constant-Size Dynamic *k*-TAA. In R. D. Prisco and M. Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2006.

[4] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad. Privacy in mobile computing for location-sharing-based services. In *PETS*, volume 6794 of *Lecture Notes in Computer Science*, pages 77–96. Springer, 2011.

[5] O. Blazy, S. Canard, G. Fuchsbauer, A. Gouget, H. Sibert, and J. Traoré. Achieving optimal anonymity in transferable e-cash with a judge. In *AFRICACRYPT*, volume 6737 of *Lecture Notes in Computer Science*, pages 206–223. Springer, 2011.

[6] D. Boneh and X. Boyen. Short signatures without random oracles. In Cachin and Camenisch [9], pages 56–73.

[7] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[8] E. F. Brickell, P. Gemmell, and D. W. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *SODA*, pages 457–466. ACM/SIAM, 1995.

[9] C. Cachin and J. Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.

[10] J. Camenisch. Group signature schemes and payment systems based on the discrete logarithm problem. *PhD thesis*, 1998.

[11] J. Camenisch, R. Chaabouni, and A. Shelat. Efficient protocols for set membership and range proofs. In J. Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 234–252. Springer, 2008.

[12] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004.

[13] J. Camenisch, J.-M. Piveteau, and M. Stadler. An efficient fair payment system. In *ACM Conference on Computer and Communications Security*, pages 88–94. ACM, 1996.

[14] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003.

[15] J. Camenisch and M. Stadler. Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[16] B. Carbunar, W. Shi, and R. Sion. Conditional e-payments with transferability. *J. Parallel Distrib. Comput.*, 71(1):16–26, 2011.

[17] D. Chaum and T. P. Pedersen. Transferred cash grows in size. In *EUROCRYPT*, volume 658 of *Lecture Notes in Computer Science*, pages 390–407. Springer, 1992.

[18] M. Chia, S. Krishnan, and J. Zhou. Challenges and Opportunities in Infrastructure Support for Electric Vehicles and Smart Grid in a Dense Urban Environment. To appear in IEEE International Electric Vehicle Conference 2012, 2012.

[19] M. Duckham. Moving forward: location privacy and location awareness. In *SPRINGL*, pages 1–3. ACM, 2010.

[20] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61. Springer, 2010.

[21] J. Freudiger, R. Shokri, and J.-P. Hubaux. Evaluating the privacy risk of location-based services. In *Financial Cryptography*, volume 7035 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2011.

[22] G. Fuchsbauer, D. Pointcheval, and D. Vergnaud. Transferable constant-size fair e-cash. In *CANS*, volume 5888 of *Lecture Notes in Computer Science*, pages 226–247. Springer, 2009.

[23] 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*, 38:690–728, July 1991.

[24] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[25] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[26] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive*, volume 5538 of *Lecture Notes in Computer Science*, pages 390–397. Springer, 2009.

[27] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.

[28] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In Cachin and Camenisch [9], pages 571–589.

[29] L. Liao, D. J. Patterson, D. Fox, and H. A. Kautz. Learning and inferring transportation routines. *Artif. Intell.*, 171(5-6):311–331, 2007.

[30] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Enhancing location privacy for electric vehicles (at the right time). In *ESORICS*, volume 7459 of *Lecture Notes in Computer Science*, pages 397–414. Springer, 2012.

[31] B. Lynn. The Java Pairing Based Cryptography Library (jPBC), 2010. http://http://libeccio.dia.unisa.it/projects/jpbc/.

[32] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In J. Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

Appendix A
Details of $\mathcal{PK}_1$ - $\mathcal{PK}_4$

$\mathcal{PK}_2$ and $\mathcal{PK}_3$ requires the prover to demonstrate 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 relative small compared with the security parameter. Thus, we can make use of the efficient interval proof due to [11]. Specifically, the supplier publishes a set of 'digital signatures' on the messages $0, \ldots, D$, denotes as $\sigma_0, \ldots, \sigma_D$. To prove that $\tilde{B}+v$ lies in the interval $[0, D]$ for a known value v, the user proves that he/she is in possession of a 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 [1], [2]. We instantiate the interval proof with the weakly-secure signature scheme in [6]. 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 proving directly $\tilde{B}+v$ (or $-v$) is within 0 to D. Rather, the user demonstrates that he/she knows a 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, $\varsigma_i = h^{\frac{1}{\delta+i}}$. For efficiency considerations, set $\hat{E} = \hat{e}(g, g)$, $\hat{E}_i = \hat{e}(g_i, g)$ for $i = 0$ to 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 $\mathsf{param} := \mathsf{param} \cup \{h,\ h_1,\ h_2,\ f,\ \varsigma_0,\ \ldots,\ \varsigma_D,\ \hat{E},\ \hat{E}_0,\ \hat{E}_1,\ \hat{E}_2,\ \hat{E}_3,\ \hat{E}_4,\ \hat{H},\ \hat{H}_0,\ \hat{H}_1,\ \hat{H}_2,\ \hat{H}_3\}$ and the value of δ should be be deleted or 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.

A. $\mathcal{PK}_1$

1) The supplier sends a random challenge R.

2) The user randomly chooses $\rho_{y'}, \rho_s, \rho_t \in_R \mathbb{Z}_p$, 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 $z_{y'} = \rho_{y'} - cy'$, $z_s = \rho_s - cs$, $z_t = \rho_t - ct$ and sends c, $z_{y'}$, z_s, z_t to the supplier.

5) The supplier computes

$$T \quad = \quad C^c g_0^{z_{y'}} g_3^{z_s} g_4^{z_t}$$

accepts the proof if and only if

$$c \overset{?}{=} H(T, R).$$

B. $\mathcal{PK}_2$

1) The supplier sends a random challenge R.

2) The user randomly chooses $k_1, k_2, k_3, k_4 \in_R \mathbb{Z}_p$, 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}, $\rho_{\beta_4}, \rho_t \in_R \mathbb{Z}_p$ and computes the following:

$$\begin{aligned}
T_1 &= g_0^{\rho_{y'}} g_1^{\rho_I} g_2^{\rho_{\tilde{B}}} g_3^{\rho_s} g_4^{\rho_t} \\
T_2 &= h_1^{\rho_{k_1}} h_2^{\rho_{k_2}} \\
T_3 &= F_1^{-\rho_{\tilde{e}}} h_1^{\rho_{\beta_1}} h_2^{\rho_{\beta_2}} \\
T_4 &= \hat{H}_0^{\rho_{k_2}} \hat{H}_1^{\rho_{\beta_2}} \hat{E}_0^{\rho_{\tilde{y}}} \hat{E}_1^{\rho_I} \hat{E}_2^{\rho_{\tilde{B}}} \hat{E}_4^{\rho_t} \hat{e}(F_2, g)^{-\rho_{\tilde{e}}} \\
T_5 &= h_1^{\rho_{k_3}} h_2^{\rho_{k_4}} \\
T_6 &= F_3^{-\rho_{\tilde{B}}} h_1^{\rho_{\beta_3}} h_2^{\rho_{\beta_4}} \\
T_7 &= \hat{H}_2^{\rho_{\beta_4}} \hat{H}_3^{\rho_{k_4}} \hat{e}(F_4, h)^{-\rho_{\tilde{B}}} \\
T_8 &= \mathfrak{R}^{\rho_t}
\end{aligned}$$

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 to T_8 as follows:

$$\begin{aligned}
T_1 &= C^c g_0^{z_{y'}} g_1^{z_I} g_2^{z_{\tilde{B}}} g_3^{z_s} g_4^{z_t} \\
T_2 &= F_1^c h_1^{z_{k_1}} h_2^{z_{k_2}} \\
T_3 &= F_1^{-z_{\tilde{e}}} h_1^{z_{\beta_1}} h_2^{z_{\beta_2}} \\
T_4 &= \left(\hat{e}(F_2, w)\hat{E}^{-1}\hat{E}_3^{-\tilde{s}}\right)^c \cdot \\
&\quad \hat{H}_0^{z_{k_2}} \hat{H}_1^{z_{\beta_2}} \hat{E}_0^{z_{\tilde{y}}} \hat{E}_1^{z_I} \hat{E}_2^{z_{\tilde{B}}} \hat{E}_4^{z_t} \hat{e}(F_2, g)^{-z_{\tilde{e}}} \\
T_5 &= F_3^c h_1^{z_{k_3}} h_2^{z_{k_4}} \\
T_6 &= F_3^{-vc} F_3^{-z_{\tilde{B}}} h_1^{z_{\beta_3}} h_2^{z_{\beta_4}} \\
T_7 &= \left(\hat{e}(F_4, fh^{-v})\hat{H}^{-1}\right)^c \hat{H}_2^{z_{\beta_4}} \hat{H}_3^{z_{k_4}} \hat{e}(F_4, h)^{-z_{\tilde{B}}} \\
T_8 &= E^c \mathfrak{R}^{z_t}
\end{aligned}$$

and accepts the proof if and only if

$$c \overset{?}{=} H(\{F_i\}_{i=1}^4, \{T_i\}_{i=1}^8, R, \mathfrak{R})$$

C. $\mathcal{PK}_3$

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

D. $\mathcal{PK}_4$

1) The supplier sends a random challenge R.
2) The user randomly chooses $k_1, k_2 \in_R \mathbb{Z}_p$, 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}, $\rho_{\beta_2} \in_R \mathbb{Z}_p$ and computes the following:

$$\begin{aligned}
T_1 &= g_0^{\rho_{y'}} g_3^{\rho_s} g_4^{\rho_t} \\
T_2 &= h_1^{\rho_{k_1}} h_2^{\rho_{k_2}} \\
T_3 &= F_1^{-\rho_{\tilde{e}}} h_1^{\rho_{\beta_1}} h_2^{\rho_{\beta_2}} \\
T_4 &= \hat{H}_0^{\rho_{k_2}} \hat{H}_1^{\rho_{\beta_2}} \hat{E}_0^{\rho_{\tilde{y}}} \hat{E}_4^{\rho_t} \hat{e}(F_2, g)^{-\rho_{\tilde{e}}}
\end{aligned}$$

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, $z_{k_1} = \rho_{k_1} - ck_1$, $z_{k_2} = \rho_{k_2} - ck_2$, $z_{y'} = \rho_{y'} - cy'$, $z_s = \rho_s - cs$, $z_t = \rho_t - ct$, $z_{\tilde{e}} = \rho_{\tilde{e}} - c\tilde{e}$, $z_{\tilde{y}} = \rho_{\tilde{y}} - c\tilde{y}$, $z_{\beta_1} = \rho_{\beta_1} - ck_1\tilde{e}$, $z_{\beta_2} = \rho_{\beta_2} - ck_2\tilde{e}$ to the supplier.

5) The supplier computes T_1 to T_4 as follows:

$$\begin{aligned}
T_1 &= C^c g_0^{z_{y'}} g_3^{z_s} g_4^{z_t} \\
T_2 &= F_1^c h_1^{z_{k_1}} h_2^{z_{k_2}} \\
T_3 &= F_1^{-z_{\tilde{e}}} h_1^{z_{\beta_1}} h_2^{z_{\beta_2}} \\
T_4 &= \left(\hat{e}(F_2, w)\hat{E}^{-1}\hat{E}_1^{-I}\hat{E}_2^{-\tilde{B}}\hat{E}_3^{-\tilde{s}}\right)^c \cdot \\
&\quad \hat{H}_0^{z_{k_2}} \hat{H}_1^{z_{\beta_2}} \hat{E}_0^{z_{\tilde{y}}} \hat{E}_4^{z_t} \hat{e}(F_2, g)^{-z_{\tilde{e}}}
\end{aligned}$$

and accepts the proof if and only if

$$c \overset{?}{=} H(\{F_i\}_{i=1}^2, \{T_i\}_{i=1}^4, R)$$

Appendix B
Security Analysis

A. Prevention of Cheating User

The following game specifies the interaction between an attacker $\mathcal{A}$ and the challenger $\mathcal{C}$ defines the requirement *Prevention of Cheating User*. $\mathcal{A}$ plays the role of a set of cheating user, while $\mathcal{C}$ plays the role of an honest supplier. In the game, $\mathcal{C}$ keeps a running balance W possesses by $\mathcal{A}$. $\mathcal{A}$ wins the game if it can make W to be 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 the judge is trusted and the secret key of the judge is also given to the attacker[6].

- *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. Secret key of the judge is also given to $\mathcal{A}$. $\mathcal{C}$ initializes a counter W which is 0.
- *Interactions.* $\mathcal{A}$ can make the following four types of interaction freely with $\mathcal{C}$.
  1) *Registration.* $\mathcal{A}$ interacts with $\mathcal{C}$ in the registration protocol. Upon successful completion of the protocol, W is increased by the value D.
  2) *Charging.* $\mathcal{A}$ 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) *Discharging.* $\mathcal{A}$ interacts with $\mathcal{C}$ in the topup protocol of value v. Upon successful completion of the protocol, W is increased by the value v.
  4) *Statement.* $\mathcal{A}$ interacts with $\mathcal{C}$ in the statement protocol of value d. Upon successful completion of the protocol, W is increased by the value d.

[6] In our actual construction, the judge does not have any public/secret key.

- *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, assume there exists $\mathcal{A}$, we show how to construct a forgery attack against the underlying BBS+ signature [3]. Since BBS+ signature is known to be unforgeable, this means no PPT adversary $\mathcal{A}$ can win in the above game. That is, our system supports prevention of cheating user.

Before stating our proof, let us assume the zero-knowledge proof-of-knowledge $\mathcal{PK}_1$, $\mathcal{PK}_2$, $\mathcal{PK}_3$, $\mathcal{PK}_4$ are *sound*. That is, given blackbox access to the prover that makes these zero-knowledge proofs, there exists extractor algorithms $\mathcal{EX}_1$, $\mathcal{EX}_2$, $\mathcal{EX}_3$, $\mathcal{EX}_4$ 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 black-box $\mathcal{SO}$, normally referred to as *signing oracle*. $\mathcal{SO}$ outputs a 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 a valid signature (A^*, e^*, y^*) on message $(m_1^*, m_2^*, m_3^*, m_4^*)$ such that the former is not the output of $\mathcal{SO}$[7].

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. Below 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 extracts 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

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

(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 in order to reduce the value of W $\mathcal{A}$ has to interact with $\mathcal{S}$ in the discharge protocol and the number of signatures given to $\mathcal{A}$ via $\mathcal{S}$ is limited and that $\mathcal{PK}_3$ assures $\mathcal{S}$ 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.

■

B. 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 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 the judge is trusted and 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 two possible honest users, even if the previous interactions of these two users are all scheduled by the adversary. Of course, these two honest users have not participate in the voluntary tracing since the attacker could trace their transaction if their consent are given.

- *System Parameter.* $\mathcal{C}$ creates and publishes the system parameter param and the secret key γ. $\mathcal{C}$ also creates the public/secret key of on behalf of the judge. Secret key of the supplier and the judge judge are given to $\mathcal{A}$. The challenger The malicious adversary $\mathcal{A}$ creates and publishes the system parameter param as well as the public key of the judge.
- *Interactions.* $\mathcal{A}$ can make the following four types of interaction freely with $\mathcal{C}$, who acts on behalf of two honest users.
  1) *Registration($b \in \{0, 1\}$).* $\mathcal{A}$ interacts with $\mathcal{C}$ who acts on behalf of U_b in the registration protocol. The value b is specified by $\mathcal{A}$.
  2) *Charging($b \in \{0, 1\}$).* $\mathcal{A}$ interacts with $\mathcal{C}$ who acts on behalf of U_b in the charging protocol of value v for user . The value b is specified by $\mathcal{A}$.
  3) *Discharging($b \in \{0, 1\}$).* $\mathcal{A}$ interacts with $\mathcal{C}$ who acts on behalf of U_b in the topup protocol of value v. The value b is specified by $\mathcal{A}$.
  4) *Statement($b \in \{0, 1\}$).* $\mathcal{A}$ interacts with $\mathcal{C}$ who acts on behalf of U_b in the statement protocol of value d. The value b is specified by $\mathcal{A}$.

- *Challenge.* $\mathcal{A}$ chooses a type of interaction, either charging or discharging provided that both U_0 and U_1 has sufficient balance in case 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 a simulator $\mathcal{S}$ which solves an instance of the DDH problem. $\mathcal{S}$ is given 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 a 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.* $\mathcal{S}$ acts on behalf of user U_0 honestly. For U_1, the pair $\mathfrak{R}$, E supplied to $\mathcal{A}$ are not correctly formed. Specifically, for the j-th query, $\mathcal{S}$ randomly picks $\mu_j \in_R \mathbb{Z}_p$, 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 $\mathfrak{R}$, E supplied to $\mathcal{A}$ are not correctly formed. Specifically, for the j-th query, $\mathcal{S}$ randomly picks $\mu_j \in_R \mathbb{Z}_p$, 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 $\mathcal{PK}_3$.
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 while 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 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 simulator 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 a guess bit b. If $b = \hat{b}$, $\mathcal{S}$ confirms that there exists a, b such that $D_2 = D_1^a$, $D_3 = D_1^b$, $D_4 = D_1^{ab}$. Otherwise, it confirms that no such pair of (a, b) exists.

Suppose $\mathcal{A}$ 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 $D_2 = D_1^a$, $D_3 = D_1^b$, $D_4 = D_1^{ab}$. In this case, 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 $D_2 = D_1^a$, $D_3 = D_1^b$, $D_4 = D_1^{ab}$. In this case, 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 $\hat{b} = 1$, 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 $1/2 - \epsilon/2$.

Summing up, the probability of $\mathcal{S}$ answering correctly (recall that in the DDH problem specification, the problem instance 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

The last security requirement is the correctness of tracing. This is due to the soundness of $\mathcal{PK}_2$ or $\mathcal{PK}_3$ that the value $E = \mathfrak{R}^t$.