P6V2G: a privacy-preserving V2G scheme for two-way payments and reputation

Rebecca Schwerdt[*], Matthias Nagel, Valerie Fetzer, Tobias Gräf and Andy Rupp

*Correspondence:
schwerdt@kit.edu
KIT Department of Informatics, Am Fasanengarten 5, 76131 Karlsruhe, Germany

Abstract

The number of electric vehicles (EVs) is steadily growing. This provides a promising opportunity for balancing the smart grid of the future, because vehicle-to-grid (V2G) systems can utilize the batteries of plugged-in EVs as much needed distributed energy storage: In times of high production and low demand the excess energy in the grid is stored in the EVs' batteries, while peaks in demand are mitigated by EVs feeding electricity back to the grid. But the data needed for managing individual V2G charging sessions as well as for billing and rewards is of a highly personal and therefore sensitive nature. This causes V2G systems to pose a significant threat to the privacy of their users. Existing cryptographic protocols for this scenario either do not offer adequate privacy protection or fail to provide key features necessary to obtain a practical system.

Based on the recent cryptographic toll collection framework P4TC, this work introduces a privacy-preserving but efficient V2G payment and reward system called P6V2G. Our system facilitates two-way transactions in a semi online and post-payment setting. It provides double-spending detection, an integrated reputation system, contingency traceability and blacklisting features, and is portable between EVs. The aforementioned properties are holistically captured within an established cryptographic security framework. In contrast to existing protocols, this formal model of a V2G payment and reward system allows us to assert all properties through a comprehensive formal proof.

Keywords: Vehicle-to-grid, Location privacy, Provable security, Electronic payments, Reward system, Double-spending detection

Introduction

In an effort to counter global warming and fossil fuel depletion, energy transitions from fossil fuels to renewable energy sources are expedited on a global scale. This change from time-independent, fully controllable power plants to highly volatile, seasonal and distributed renewable power generation poses many challenges. Foremost among them is balancing generation and demand to maintain stability of the energy distribution system and provide uniform power quality. This requires increased storage capacities and load flexibility. One aspect of the energy transition—which has the potential to solve at least part of this problem—is the introduction of electric vehicles (EVs) as a means of private

and public transportation. EVs do not require fossil fuels and can significantly reduce air and noise pollution, especially in urban areas. Most industrial nations have started to subsidize EVs as well as set concrete objectives on how many EVs they want to deploy by a certain year (Foley et al. 2010). Target dates for complete bans on the sale of new cars with internal combustion engines are as early as 2020 in some countries—with many others planning to have phased them out by 2030 (Burch and Gilchrist 2018; Löfven 2019; Department of Public Expenditure and Reform 2019). While EVs themselves pose a substantial and growing demand on the system, they also have the capacity to contribute to a solution. Privately owned vehicles remain parked for more than 95% of the time (Kempton and Letendre 1997). During this time the EV's battery can be utilized in a vehicle-to-grid (V2G) scheme, providing storage capacity and flexibility in the recharging process to help balance power generation and demand in the grid (Kempton and Letendre 1997): Surpluses from volatile renewable power generation can be stored in parked EV's batteries, which collectively provide substantial distributed energy storage. Conversely, energy stored in EVs can be used to mitigate peaks in demand and drops in generation. In contrast to the benefits such a system provides it can also pose a significant threat to the privacy of its users (Krumm 2007; Golle and Partridge 2009). The individual location data that may be gathered from identifying or linkable transactions is highly sensitive and can be maliciously exploited. To further participation in a voluntary V2G system, privacy guarantees can be just as important as low effort usability, as concerns arising from loss of personal information as well as any effort required from the user could discourage its use. Facilitating effortless payment for EV charging at a public charging spot (EVSE) as well as rewards for ancillary services during the charging session while keeping transactions privacy-preserving, unidentifying, and unlinkable poses a big challenge.

This paper introduces a novel and privacy-preserving V2G payment system called P6V2G. It was developed by adapting the recently proposed cryptographic toll collection system P4TC (Hoffmann et al. 2018) to a V2G setting where multiple charging point operators provide publicly accessible EVSEs for EVs. P6V2G facilitates two-way transactions for EV charging, allowing both payment from the user to the operator for having their vehicle charged as well as rewards for providing the EV's battery as a buffer to the grid. P6V2G is a post-payment system where a user accumulates debt and rewards on a wallet and gets periodic, e.g., monthly, bills from their e-mobility operator (eMO). Therefore, P6V2G offers convenient low effort usability, leading more people to make their EV's batteries available for ancillary services. Our protocols are designed to be efficient even when performed on low performance hardware that is realistically present in EVs, EVSEs and wallets[1]. To further take realistic technology restrictions into account, EVSEs are not required to have permanent online capabilities; tasks can be conducted offline without loss of any functionality. An integrated reputation scheme records how reliable the user is, e.g., in leaving their EV plugged in as long as they predicted at the start of a charging session. This enables EVSEs to better plan the charging of vehicles and, e.g., offer cheaper/higher rewarded long term charging to reliable users. To ensure security, P6V2G

---

[1]Wallets might be realized on smart phones, but a separate device like a smart card might be more desirable as it can be left in the car. To handle the low performance of currently available smart cards, we would assume wallets to act as a secure store of the users secrets only and be able to outsource more costly computation to the EV.

provides a fraud detection mechanism that not only detects fraudulent behavior and identifies the misbehaving user, but outputs a publicly verifiable proof of guilt to avert the possibility of false accusations. With consent of a designated authority, misbehaving users can have their wallets traced (*contingency traceability*), blacklisted, or their debt recalculated—for instance if the user fails to present their wallet for billing at the end of a billing period. In addition, our system implements a mechanism to blacklist EVs (independently of blacklisted wallets), so that stolen vehicles can be traced or even detained at an EVSE. Since EVs and wallets are modeled as two different entities in P6V2G, they may be combined arbitrarily, resulting in a portable system. This is particularly useful for users with multiple cars as well as for renting or sharing EVs. The whole system is formalized in a comprehensive security model and validated by a rigorous proof asserting system security as well as user privacy.

Related work

In this section we firstly give a summary of different payment and reward systems that were proposed for, or can be adapted to a V2G setting. This overview shows that while all of them have their respective merits, none so far provides solutions for *all* the challenges and restrictions posed by realistic V2G interactions. In a second part, we give a more detailed introduction of the two cryptographic systems (BBA+ (Hartung et al. 2017) and P4TC (Hoffmann et al. 2018)) that our own V2G system P6V2G is based on.

Other (V2G) payment and reward systems There are several previously published payment and reward systems that can be deployed in a V2G scenario. Reward-only systems (without two-way payments) include TPP (Wang et al. 2015), and PRAC (Wan et al. 2016). Both systems are not able to aggregate rewards. This results in linkability unless rewards are sufficiently uniform and redeemed individually. Furthermore both systems require permanent online capabilities and very few properties are proven in any formal way. For the remainder of this paragraph we will focus on payment systems that offer two-way transactions. A terse comparison of the discussed systems, the common options of (Chip-and-PIN) *Debit/Credit Card* and *Paper Cash*, and our system is depicted in Table 1. In a survey by Han and Xiao (2016) another overview of several V2G payment and reward systems, as well as a general summary of challenges and methods associated with privacy in the V2G environment can be found.

In 2012, Liu et al. (2012) proposed a system designed to provide privacy at the "right" time (PRT), which has been further improved by Au et al. (2014). PRT facilitates two-way transactions between an EV and a single operator running multiple EVSEs. Further parties representing the operator include a grid management and a billing server. User's accounts are pre-paid and have to be topped up regularly. This excludes the possibility of problems arising from users who do not pay their bills (on time) but requires more effort on the user side and leads to users being stuck at EVSEs if they do not have sufficient balance on their account to charge an empty EV. Each EVSE needs a permanent connection to the operator's other parties as PRT is designed as an online system. In particular, all-time access to the central billing server is essential to prevent double-spending. Further assumptions are some read-only memory on the EV that can not be removed from the vehicle. PRT provides traceability that is facilitated by a judge authority and only possible with the user's consent. Via this traceability the system implements lost

Table 1 Comparison of two-way payment systems

System	Location privacy	Post-payment	Semi online	Double-spending detection	Reputation	Contingency traceability	Blacklisting	Portable	Formal proof
PRT (Au et al. 2014)	✓	✗	✗	n/a[1]	✗	✓	✗	✓	✓/✗
PnC (ISO 15118) (DIN-Normenausschuss Automobiltechnik 2015; 2016)	✗	✓	✓	n/a[2]	✗	n/a[2]	✓	✗	✗
E-Cash (Baldimtsi et al. 2015)	✓	✗	✓	✓	✗	✗	✗	✓	✓/✗
Debit/Credit Card	✗	✗/✓	✓	n/a[2]	✗	n/a[2]	✓	✓	✗
Paper Cash	✓	✗	✓	n/a	✗	✗	✗	✓	✗
P6V2G	✓	✓	✓	✓	✓	✓	✓	✓	✓

[1]Double-spending detection is not applicable in online systems.
[2]Double-spending detection/traceability are not applicable in identifying systems

protection, dispute resolution and tracing of stolen vehicles. While neither (Liu et al. 2012) nor (Au et al. 2014) provide any reputation system, (Liu et al. 2012) describes the possibility of facilitating a portable mode where the account is managed on some mobile device instead of the EV. This, however, excludes other features like correct tracing and the system still relies on a read-only memory that now has to be available on this mobile device. Au et al. (2014) includes two game-based proofs for cheating prevention and location privacy. Unfortunately these proofs only assert the impossibility of very specific attacks on the system and do not give any guarantees for attacks that were not explicitly considered.

The International Electrotechnical Commission's V2G standard ISO 15118 (DIN-Normenausschuss Automobiltechnik 2015; 2016) allows for contract based charging called "Plug & Charge" (PnC). In this system the EV shows a valid contract certificate from its eMO to the EVSE at the start of a charging session. At the end of each charging session, the EV signs the service detail record (SDR) of this session and sends it to the EVSE. In an online phase, the EVSE sends the SDR to the respective eMO who collects and uses them to bill the contract owner at the end of each billing period. Since the certificate includes the contract ID, this system is fully identifying. Furthermore—since the system is not portable between EVs—driver specific billing within EV fleets (like car sharing or company fleets) is not possible while simultaneously preserving the drivers privacy from the operator of this fleet: the corresponding eMO has to provide the fleet operator with a detailed account of all the EV's charging sessions so they can match them to specific drivers. The PnC system facilitates semi online post-payments and provides blacklisting. Please also note that only minimal security is guaranteed in ISO 15118. While off-the-shelf cryptographic methods like encryption and signatures are employed for EV authentication and to guarantee privacy from eavesdropping but non-involved parties, EVSEs are essentially assumed to be trustworthy and very little user security is offered. Furthermore, no resulting privacy or security guarantees are formalized, formally discussed or proven. In addition to the PnC payment method, the ISO 15118 allows for alternative payment methods (like P6V2G) to be added in future. Although the ISO 15118 only deals with direct communication between the EV and SDR—this inherently represents only a small part of our system, namely Part I of Debt Accumulation (cf. Figs. 1 and 2 in the next section)—much of our protocol can be integrated into this standard in an intuitive way. Details of how this might be done can be found in the full version (Schwerdt et al. 2019) of this paper.

A privacy increasing modification of PnC was proposed under the name POPCORN (Höfer et al. 2013). POPCORN employs anonymous group signatures and three different trusted third parties to achieve anonymous one-way payments as long as no parties collude. It does not hide individual transactions from the eMO, is not portable, its proof-of-concept implementation is inefficient and no security or privacy proof is given. Some of POPCORN's privacy properties were later verified in ProVerif (Blanchet 2016) by Fazouane et al. (2015), who simultaneously identified several weaknesses.

Lastly, we consider the option of anonymous e-cash for V2G payments. One general drawback of e-cash is that it is a prepaid and primarily one-way system. Hence parties need to always have coins of exactly the right amount at hand for any transaction they might want to participate in—which limits either pricing flexibility or efficiency.

When trying to employ e-cash for two-way payments in a V2G scenario there are several technical roadblocks. In standard offline e-cash (e.g., (Camenisch et al. 2005)), coins are not transferable without consulting the bank. When the user receives e-coins from the EVSE—which could be done by letting the EVSE participate in the spending protocol as the payer and the user as the payee—they first need to deposit those coins with the operator who originally issued them. In case this operator and the EVSEs collude they learn all the locations a user got rewards at. We also can not assume that a user receives freshly issued coins directly from the operator as rewards, as the corresponding e-cash withdrawal protocol is not anonymous; only spending a coin protects the privacy of the payer. Hence, the privacy guarantees that standard offline e-cash provides do not fit our needs. Transferable e-cash such as (Baldimtsi et al. 2015) does not achieve our goals either. Transferable e-cash schemes allow to anonymously transfer an e-coin multiple times between parties without consulting the bank. Thus, in our scenario an EVSE could transfer a coin received from the operator or another user to a user who is eligible for a reward. Unfortunately, there is an impossibility result regarding privacy that applies to this scenario. Canard and Gouget's work (Canard and Gouget 2008) implies, that if the parties representing the V2G payment infrastructure collude, payment and reward transactions of a user can be linked.

BBA+ and P4TC At its core, our proposed scheme P6V2G facilitates a black-box accumulator (BBA+) (Hartung et al. 2017). This building block implements a personal wallet for anonymous, unlinkable point collection and redemption. It ensures that a wallet can only be used by its legitimate owner, protects against manipulation of the wallet's true value, provides double-spending detection and comes with a mix of game-based and simulation-based proofs for user security and privacy as well as system security. Hoffmann et al. (2018) utilize BBA+ to build a system called P4TC which applies BBA+ to a toll collection scenario and offers anonymous, unlinkable tolling with post-payments. In order to fulfill all desired properties of a toll collection scenario, the authors augmented BBA+ with additional features like blacklisting, selective tracing of misbehaving users and recalculation of debt in case of a dispute. Moreover, P4TC comes with a full simulation-based security and privacy proof.

Our contribution

This work transfers, adjusts and expands P4TC to fit the V2G scenario. The resulting system P6V2G allows for unlinkable and efficient V2G payments, rewards and reputation scores. It facilitates post-payment two-way transactions in a semi online and portable setting and includes double-spending detection as well as features for blacklisting (of both EVs and users), recalculation of debt and contingency traceability. To the best of our knowledge it is the first system offering all these properties simultaneously and for some properties it is the first to provide them at all. Furthermore, P6V2G is privacy-preserving and secure against malicious adversaries. This was asserted in a comprehensive formal model and proof within an established cryptographic framework, the Universal-Composability (UC) setting (Canetti 2001; Canetti et al. 2007)[2].

To achieve the above qualities we build upon the toll collection scheme P4TC (Hoffmann et al. 2018). Transferring this to the V2G scenario already provides several of the
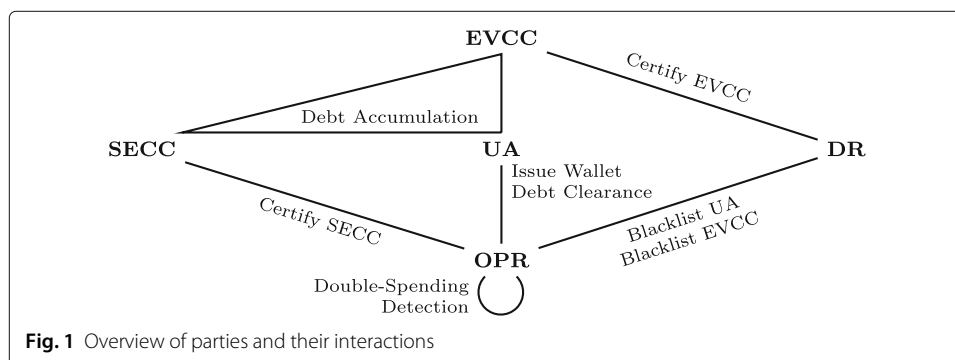
---

[2]Although formal security proofs are not widely appreciated in application-oriented communities yet, we consider it vital for security (especially of complex and necessarily convoluted systems) that this changes in future.

required functions and properties, e.g., privacy-preserving accumulation of debt, a verifiable fraud detection feature and blacklisting for wallets. Some requirements of our system, like two-way payments, are not present in the toll collection scenario where users only have to pay toll and clear their wallet once, but do not collect rewards. Fortunately this important feature is still achieved by the employed black-box accumulator without any further modifications. There are, however, several characteristics of our setting with either much simpler or no equivalent aspects at all in P4TC, which call for novel solutions. First and most challenging among them is the presence of multiple operators instead of just one operating party in P4TC. This introduces considerable complexity as it yields additional potential for security threats in the system which our protocols have to preclude. Similar challenges arise from the user side of the system. Other than in P4TC—where a user is represented by their car's fixed on-board unit—our system is designed to be portable. Hence, the user's side is divided into two separate parties: user accounts, containing the users' wallets, and EVs represented by their communication controller. While user accounts have a similar role to on-board units in P4TC, EV communication controllers introduce additional complexity. Among other challenges, they require an additional but likewise privacy-preserving blacklisting mechanism to deal with stolen cars. Furthermore, P4TC does not include any reputation scheme for the reliability of users. While our formal model and proof largely follow the structure of (Hoffmann et al. 2018), they are more complex due to the existence of arbitrarily many operators (instead of just one in P4TC) and EV communication controllers as entirely new parties.

System definition

This section introduces the general type of V2G system we consider. In addition to the overall setting this includes details on participating parties, required functions and desired features. An overview can be found in Fig. 1.

   We consider a setting in which multiple charging point operators each provide publicly accessible EVSEs for EVs. Users can charge their EVs at EVSEs as well as offer their EV's battery capacity to provide ancillary services to the grid. For these bidirectional services each user gets an accumulated bill from their own eMO at the end of each billing period. We assume that charging point operators and eMOs cooperate in a way that every user may use any of the EVSEs. While the necessary information to manage operator accounting is included in our system, the financial settlement between operators lies outside the scope of our system.

Fig. 1 Overview of parties and their interactions

Schwerdt *et al. Energy Informatics* (2019), 2(Suppl 1): 32

Page 8 of 21

Parties

We now give a detailed description of the different parties and their roles. An *operator* (*OPR*) $\mathcal{O}$ is either the eMO of an EV, a charging point operator maintaining EVSEs, or possibly both. Each EVSE is represented by their *supply equipment communication controller* (*SECC*) $\mathcal{C}$ and is assigned to one specific (charging point) OPR but may be used by any (e-mobility) OPR's customers. It is assumed to have the capabilities for occasional database synchronization with its OPR but does not require any permanent internet connection to other parties. Different OPRs are assumed to occasionally synchronize their databases as well in order to catch double-spenders. An EV is represented by its *electric vehicle communication controller* (*EVCC*) $\mathcal{E}$. It needs to be registered to be able to participate in the protocols but is not associated with any other party, enabling multiple users to share the same car or one person using several different cars. A *user account* (*UA*) $\mathcal{A}$ is a low performance electronic device (we tend to think of it as a smart card) that is issued to a user upon registration with their OPR. It contains the user's wallet which is used to collect debt and reputation during charging sessions and is not bound to a specific EV. In addition to these mandatory parties we assume existence of some regulatory *dispute resolver* (*DR*) $\mathcal{D}$. This party mainly handles disputes and gives permission for any exceptional measures that either limit a party's access to the system (like being black-listed) or detract from their privacy (e.g., recalculating the debt on an uncooperative UA). A *user* is a physical person using an EV, deciding where and when to charge it, owning a UA, picking their eMO and paying the bills. They do not, however, participate as a separate party in our protocols. The only input needed from a user are the charge targets at the start of a charging session. As we assume this choice to be indicated via the EVSE's human machine interface, it is formally made by the SECC in our protocols. Other than this input the user is represented by a UA and an EVCC for the duration of a charging session.

Functions and features

The remainder of this section gives an overview of the functions and features we require of a V2G system. In contrast to more cumbersome prepaid and cash options, our aim is a post-payment system where the user is able to charge their vehicle over the course of a billing period and gets a combined bill for all their charging sessions afterwards. Basic functions of a post-payment V2G system include the accumulation of debt on a personal wallet and clearance of this wallet's debt. Both should be conducted in a privacy preserving way, keeping transactions anonymous as well as unlinkable. As an additional feature, we want users to be able to gain (or lose) reputation for, e.g., adhering to their predicted behavior. This way EVSEs might offer special tariffs to reliable users while not trusting the predictions of users who regularly end their charging sessions sooner than promised. All other features of the system deal with fraud and misbehaving users: As explained in "Inherent privacy and security limitations" section, not all kinds of fraud can be prevented in the given setting. But any fraud needs to at least be detected after the fact. To protect users from false accusations, this fraud detection mechanism has to provide a publicly verifiable proof of the user's guilt. To further protect the system from misbehaving users, we require the possibility to blacklist UAs as well as EVs (independently of each other) and a mechanism to recalculate the debt accumulated by an uncooperative UA or lost wallet with the consent of the

DR. In extreme circumstances we also want the charging sessions of a UA to become traceable.

Protocol description

This section describes the main features and protocols of the P6V2G system on a high level. To make these explanations more accessible, we first give an intuition behind the basic cryptographic building blocks utilized therein. But readers with a cryptographic background may want to skip "Cryptographic background" section. After going into detail on how UAs work in "Wallets" and "Tasks" sections gives an overview of the different tasks P6V2G is composed of. Lastly, "Efficiency" section illustrates the efficiency of our system. For the complete protocol we refer the reader to the full version (Schwerdt et al. 2019) of this paper.

Cryptographic background

In order to easily understand the following protocol descriptions, familiarity with a few cryptographic primitives is essential.

Public key encryption enables two parties to exchange messages such that no other party can read them. For public key encryption, the party who wants to receive messages first has to generate a *public key* and a *secret key*. The public key is then distributed to any prospective senders, while the secret key is kept secret. When a sender wants to send a confidential message, they *encrypt* the message with the public key of the recipient and send the resulting *ciphertext* instead. The recipient uses their secret key to *decrypt* the ciphertext and recover the message.

A *digital signature* is a means to verify the authenticity and integrity of a received message. Just like public key encryption, digital signatures use a public key infrastructure where two keys correspond to each party: a secret *signing key* and a public *verification key*. The sender uses their signing key to *sign* the message and sends the resulting *signature* along with the message to the receiver, who *verifies* the signature using the sender's public verification key.

A *commitment scheme* enables one party to commit themselves to a value towards another (receiving) party. The receiver is sent a *commitment*, inside which the value itself is safely hidden. At a time of their choosing, the sending party can reveal the value to the receiver by *opening* the commitment. But the sender can only open the commitment to reveal the value they initially committed to and not alter the content in retrospect.

NIZK stands for *Non-Interactive Zero-Knowledge*. As with any proof, a *NIZK proof* enables one party (called the prover) to convince a verifying party that a specific statement is true. The *zero-knowledge* property asserts that the receiver does not learn *why* the statement is true, only that it is. In fact, the receiver can infer nothing from the proof except the validity of the claim. *Non-interactive* means only one message has to be send from the prover to the verifier—namely the proof itself. No further (back or forth) interaction between the parties is necessary.

A *pseudo-random function*, intuitively, is a function that is indistinguishable from one with truly random output. At the same time it is efficiently computable for anyone with knowledge of the right input information.

The central idea of *cryptographic accumulators* is the representation of a set of elements in a single *accumulation value* of fixed size. On one hand this accumulation value securely

Schwerdt *et al. Energy Informatics* (2019), 2(Suppl 1): 32

Page 10 of 21

hides the elements within it. On the other hand it must be possible to efficiently prove that a given element is indeed contained in the accumulator. Cryptographic accumulators are called *dynamic* if elements can efficiently be added to and removed from the set over time.

Wallets

The concepts of *wallets* and *wallet states* are central for our P6V2G system. At the beginning of each billing period, the UA and corresponding OPR create a new wallet for the UA. This wallet is used for one billing period and then cleared and exchanged for a new one at the beginning of the next billing period.

A wallet is identified by its wallet ID λ and its current status described by a wallet state τ. This wallet state is altered with each transaction. It does not only store the current balance and reputation of the wallet, but rather all information needed to conduct a successful update after the next charging session. To be more formal, a (simplified[3]) wallet state τ is of the form $\tau := \left(s \mid b, r \mid x^{\text{next}} \mid \text{cert}_{\mathcal{C}} \mid \lambda, a_\lambda, \text{pk}_{\mathcal{O}_\lambda}\right)$ It consists of an *updatable part* $\left(s \mid b, r \mid x^{\text{next}} \mid \text{cert}_{\mathcal{C}}\right)$ that changes with each transaction and a *fixed part* $\left(\lambda, a_\lambda, \text{pk}_{\mathcal{O}_\lambda}\right)$ that is created once along with the wallet and stays the same for the whole life cycle of this wallet. The components of the updatable part are the last used serial number s, the current balance b and reputation r, the transaction counter x^{next} for the upcoming transaction and the certificate $\text{cert}_{\mathcal{C}}$ of the last SECC that updated the wallet (containing the SECC's attributes and OPR ID). The components that pertain to the fixed part are the wallet ID λ, the wallet attributes a_λ and the public key $\text{pk}_{\mathcal{O}_\lambda}$ of the OPR that issued this wallet.

At the beginning of a new billing period each UA $\mathcal{A}$ and its OPR $\mathcal{O}$ create a new wallet by performing the task Issue Wallet (see "Tasks" section). The wallet's initial state is $\tau := \left(s \mid 0, r \mid 1 \mid \text{cert}_{\mathcal{C}_{\mathcal{O}}} \mid \lambda, a_\lambda, \text{pk}_{\mathcal{O}}\right)$. The certificate $\text{cert}_{\mathcal{C}_{\mathcal{O}}}$ and public key $\text{pk}_{\mathcal{O}}$ correspond to the issuing OPR, who also assigns the initial reputation r (probably choosing the reputation the UA had accumulated on their last wallet) and wallet attributes a_λ (for details about the choice of wallet attributes see "Proven privacy properties" section). The wallet ID λ and serial number *s* are jointly computed randomness. Note, however, that λ is only known to the UA and the OPR does not learn it.

Each time the user charges their EV, the wallet is updated via the task Debt Accumulation (see "Tasks" section). Suppose at the beginning of a charging session with an SECC $\mathcal{C}$ the wallet's state was $[* := \left(s^* \mid b^*, r^* \mid x \mid \text{cert}_{\mathcal{C}^*} \mid \lambda, a_\lambda, \text{pk}_{\mathcal{O}_\lambda}\right)$. Then a fresh random serial number *s* is jointly computed by the UA and SECC and the transaction counter increased by one. Balance and reputation are updated by respectively adding the total cost (price minus rewards) *p* of the conducted charging and reputation gain *d*. This forms the new wallet state $\tau := \left(s \mid b^* + p, r^* + d \mid x + 1 \mid \text{cert}_{\mathcal{C}} \mid \lambda, a_\lambda, \text{pk}_{\mathcal{O}_\lambda}\right)$, which the UA saves at the end of the charging process.

At the end of each billing period, the wallet is turned over to the UA's OPR for billing purposes. This is handled via the task Debt Clearance (see "Tasks" section). The OPR learns the final balance and reputation of the wallet and can use this information to bill the user accordingly. When this task is finished, the UA discards the wallet and initiates the task Issue Wallet again to get a new one.

---

[3]For a cleaner presentation we omit some variables from the wallet state at this point. The left out variables are only relevant for cryptographic details like creating an anonymous NIZK proof for the validity of the wallet state which we do not explain in detail here. The complete wallet state can be found in the full version (Schwerdt et al. 2019) of this paper.

Schwerdt *et al. Energy Informatics* (2019), 2(Suppl 1): 32

Page 11 of 21

So far we only described the wallet's state and life cycle, but not how they are used to guarantee several of our system's properties. On one hand, whenever a UA communicates with an SECC to update a wallet, they do not want to be identified or disclose the content of their wallet. On the other hand, the SECC has to be assured that the wallet they help updating is actually valid and does not, e.g., contain a balance or attributes that were illicitly manipulated by the user. This is achieved by utilizing commitments, digital signatures and NIZK proofs. Although the actual process is a little more complicated, it generally works like this: Instead of disclosing the old wallet state to the SECC, the UA proves the (hidden) state's validity by showing that the fixed part was created and signed by its OPR (and could therefore not have been altered since the wallet was issued) and that the updatable part was updated and signed by a properly certified SECC. Note that this previous SECC is not identified in the process. After checking these proofs, the SECC provides the UA with the necessary information to update the wallet in exactly the right way and signs the (still hidden) new wallet state so the UA can assure the next SECC of its validity. This way, both privacy and system security can be achieved.

Tasks

This section illustrates the different tasks (like registering parties or issuing wallets) our V2G system is composed of. While most of our tasks have similar counterparts in P4TC, some are entirely new (like Certify EVCC and Blacklist EVCC), and some tasks (like Debt Accumulation) contain a core part that is similar to the corresponding task in the toll collection setting but have been modified and extended to encompass the additional requirements of our scenario. Due to space restrictions, we describe the main task of Debt Accumulation in detail but sketch the other tasks on a high level only. Figure 1 depicts an overview of the tasks and the parties involved in them. The only tasks missing in Fig. 1 are the registration task every party conducts on their own before participating in the system and the task Guilt Verification, which can be performed by anyone (even from outside the system) as our double-spending proofs are publicly verifiable.

Registration Before participating in the system, every party has to register. This registration entails the generation of their respective cryptographic keys (used for encryption, digital signatures, etc.).

Certification Each SECC has to get a certificate from its OPR. Main part of this certificate are a digital signature on the public key and the attributes of the SECC. It is renewed each billing period and used to ensure that the SECC is not corrupted. In addition to the actual SECCs, each OPR certifies themselves as an SECC as well[4].

EVCCs have to also be certified, but by the DR. The core part of their certificate is a revocation value *rev*. This value is drawn randomly and stored by the DR. In case the EV containing the EVCC gets, for example, stolen, the DR adds the corresponding revocation value to a blacklist bl_{EVCC} that each SECC has access to. Apart from the revocation value, the certificate contains a digital signature from the DR on the EVCC's revocation value, attributes and public key.

Issue Wallet The task Issue Wallet is executed by a UA and its OPR at the start of each billing period. The UA obtains a new wallet λ, allowing it to take part in the task Debt

---

[4]They need an SECC certificate so wallets that are freshly issued by the OPR can not be distinguished from wallets updated by an SECC.

Schwerdt *et al. Energy Informatics* (2019), 2(Suppl 1): 32

Page 12 of 21

Accumulation, and the OPR obtains a hidden trapdoor htd_λ, allowing the DR to blacklist the wallet and recover its debt if necessary.

Debt Accumulation The Debt Accumulation task is special in that it requires the interaction of three different parties (all other require at most two). An SECC, UA and an EV represented by its EVCC interact to realize the charging process of the EV. Note that the task is only identifying for the SECC while the identities of the UA and EVCC remain hidden from the SECC. Since Debt Accumulation is the main task and in many ways representative of our system, we describe it in some detail.

The task can be split into two parts: The first part, prior to charging the EV, consists of the SECC interacting with the EVCC to determine its authenticity, battery characteristics and the user's charging choices. In the second part—reminiscent of P4TC's Debt Accumulation—the SECC and UA facilitate the billing after charging took place. Overviews of both parts can be found in Figs. 2 and 3 respectively.

The first part starts with the SECC sending a list bl_{EVCC} of blacklisted EVCC's revocation values to the participating EVCC. The EVCC computes a NIZK proof that shows its own revocation value is not contained in this list. Additionally, it sends its battery's characteristics β, its attributes $a_{\mathcal{E}}$, and a NIZK proof that these attributes are correct. The SECC verifies both proofs and selects charge targets[5] based on this information. For bookkeeping purposes and dispute cases, information about the selected targets is sent to the EVCC, which marks the end of the first part.

The second part starts with the SECC authenticating itself to the UA by sending its certificate $\text{cert}_{\mathcal{C}}$. The UA checks this and calculates the proper next fraud detection ID φ. Fraud detection IDs are similar to a transaction's serial number and essential for the detection of double-spending in the task Double-Spending Detection. They are computed by applying a pseudo-random function PRF to the UA's wallet ID λ and current transaction counter x, taken from the latest recorded wallet state τ^*, i.e., $\varphi := \text{PRF}(\lambda, x)$. This assures that fraud detection IDs are random and unique if and only if no double-spending is committed. The UA sends φ over to the SECC, together with its wallet attributes a_λ and a NIZK proof that the wallet state which all this information was taken from was valid. The SECC verifies that the UA is not blacklisted by checking φ against the UA blacklist bl_{UA} and verifies the NIZK proof. Note that since the UA has to stay anonymous, it can not simply send its wallet state over in the clear. Instead, it proves the wallet state's validity without revealing anything about the state itself. In particular, the so far accumulated debt b^* and reputation r^* stay secret, as this information could infringe upon the UA's anonymity and privacy. Then both parties jointly choose a fresh random serial number s using a Blum coin toss[6].

A double-spending tag t (see task Double-Spending Detection) is jointly computed as well. The SECC then sends all information that the UA needs to correctly update its wallet. This update information includes the price p and reputation gain d. The UA ensures that its new wallet state τ is valid and saves it. The SECC creates two database entries $\omega_{\text{bl}} := (\Phi, p, d)$ and $\omega_{\text{ds}} := (\Phi, t)$, where ω_{bl} is used again in the task Blacklist UA to recompute the overall debt and reputation of a blacklisted and uncooperative UA and ω_{ds}

---

[5]The possibilities for target setting are intended to be communicated via the human machine interface of the SECC and selected by the (physical) user. Since we do not model the user as an explicit party, the targets are formally set by the SECC in this protocol.

[6]A Blum coin toss is a two-party protocol that—using commitments—results in a mutually known truly random value as long as at least one party is honest.
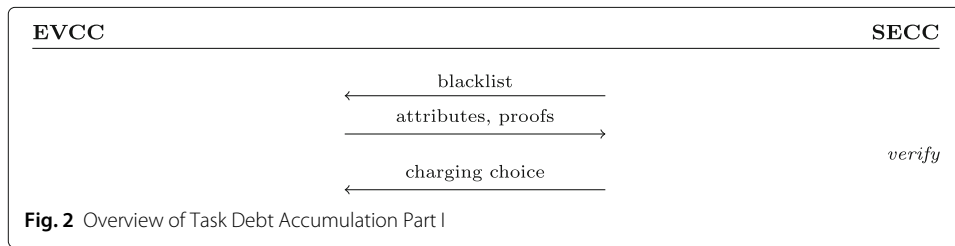
```
EVCC                                                                    SECC

                              blacklist
                        ←───────────────────

                          attributes, proofs
                        ───────────────────→
                                                                     verify
                           charging choice
                        ←───────────────────
```
Fig. 2 Overview of Task Debt Accumulation Part I

is used again in the task Double-Spending Detection to convict a UA of double-spending. We assume that each SECC periodically sends these database entries to the corresponding OPRs.

Debt Clearance The task Debt Clearance is executed between a UA and its OPR at the end of each billing period. It is similar to the interaction of an SECC and a UA in Debt Accumulation. The main differences are that the UA is not anonymous in Debt Clearance, no new wallet state is created and the OPR learns the balance and reputation accumulated on the UAs wallet. The goal of this task is for both parties to calculate the debt the user owes its OPR, so that they can be billed out-of-band.

Double-Spending Detection The task Debt Accumulation and Debt Clearance are always conducted with the assumption that the UA presents its most recent wallet and wallet state. Due to the semi online setting of the SECCs, this can not be enforced during the tasks themselves, but any misbehavior has to be detected afterwards. This kind of fraud that consists of the UA reusing and old wallet state is called *double-spending* (see also the paragraph on security in "[Inherent privacy and security limitations](#)" section). During the Debt Accumulation task, the SECC learns a unique fraud detection ID φ. If an OPR detects two entries $\omega_{ds}, \omega_{ds}{}^*$ in his database (which the SECCs regularly send their collected data to) featuring the same fraud detection ID φ, double-spending occurred. The OPR can now calculate the identity of the UA by combining the two (otherwise non-identifying) database entries that contain the same fraud detection ID φ, because they will contain different double-spending tags t, t^*. The output of this algorithm is the identity of the UA along with a proof of guilt π that shows the UA is indeed guilty of double-spending. After learning the identity of the fraudulent UA, appropriate measures can be taken by the OPR out-of-band.

Guilt Verification Whether a UA is guilty of double-spending can be verified using the Guilt Verification task. This algorithm may be run by any party. Essentially, the algorithm checks if a proof of guilt π asserts the guilt of a given UA's identity.

```
UA                                                                      SECC

                              certificate
                        ←───────────────────
 verify
                          attributes, proof
                        ───────────────────→
                                                                     verify
                      jointly compute parameters
                        ←───────────────────→

                          update information
                        ←───────────────────
 update wallet
```
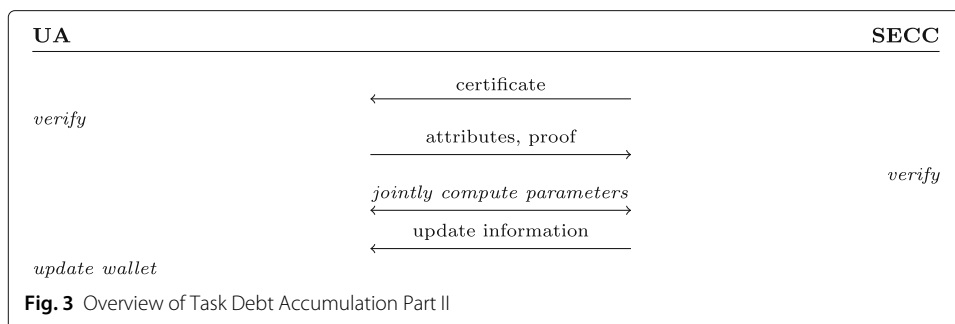**Fig. 3** Overview of Task Debt Accumulation Part II

Blacklist UA An OPR and the DR cooperate in the scope of the task Blacklist UA to provide the OPR with the data necessary to blacklist a certain UA. The OPR loads the set HTD_λ of hidden trapdoors from its storage and selects the entries htd_λ for all wallets of the UA that is to be blacklisted. The DR is the only party able to decrypt the ciphertexts contained in hidden trapdoors and is assumed to only cooperate in blacklisting tasks if the OPR has valid reason (such as proven fraud) to suspend the users privacy. The DR verifies that the htd_λ originate from the UA it wants to be blacklisted to ensure the OPR handed over the correct hidden trapdoors. Using htd_λ, the DR can calculate the (as of yet secret) wallet ID λ. Since the UA uses a pseudo-random function PRF to calculate the fraud detection IDs φ from the wallet ID λ in each run of Debt Accumulation, the DR can use its knowledge of λ to precompute all fraud detection IDs the UA might use in the current billing period. This yields fraud detection IDs $\{\varphi_0, \ldots, \varphi_{x_{bl}}\} = \{\text{PRF}(\lambda, 0), \ldots, \text{PRF}(\lambda, x_{bl})\}$ for each wallet which are collected in a set $\Phi_{\mathcal{A}}$. By adding $\Phi_{\mathcal{A}}$ to the blacklist bl_{UA}, this wallet is not able to partake in a successful run of Debt Accumulation again. The OPR can use the set $\Phi_{\mathcal{A}}$ to identify the corresponding entries of his database Ω_{bl} and recompute the current overall debt and reputation of the blacklisted UA.

Blacklist EVCC With the help from the DR an OPR is able to revoke the access of a specific EVCC. To this end they run the task Blacklist EVCC. Note that we require our system to be able to blacklist EVCCs independently from any UAs they might conjointly be used with and that the Blacklist UA method—disclosing otherwise unpredictable fraud detection IDs—is not applicable to EVCCs. Hence the design of a separate and different mechanism was required to complete our P6V2G system. Using cryptographic accumulators we were able to achieve this in a way that makes adding EVCCs to the existing blacklist very efficient: The OPR sends the EVCCs public key $\text{pk}_{\mathcal{E}}$ to the DR who assures itself that this EVCC should legitimately be banned from the system. The DR then uses $\text{pk}_{\mathcal{E}}$ to look up the EVCC's revocation value *rev* and returns it to the OPR. This value can now be added to the blacklist bl_{EVCC} on which the cryptographic accumulator is computed during Debt Accumulation.

Efficiency

To obtain a practical real world system it is paramount protocols perform efficiently under the hardware restrictions of the given setting. In this section we show this to be the case for P6V2G. Instead of discussing each task of the system in turn, we present the runtime analysis for the task Debt Accumulation in detail. Not only is this by far the computationally most extensive task of our system, but it is (partly) conducted in the presence of a waiting user and therefore time-sensitive.

The efficiency of our protocols is heavily determined by the cost of cryptographic operations, i.e., creating commitments, signing messages, computing NIZK proofs and so on. Any other factors only contribute to a negligible fraction of the overall runtime and are therefore not considered in our analysis. The individual cryptographic operations are constructed from atomic operations in the underlying pairing group setting. More precisely, every cryptographic operation is a combination of $\mathbb{G}_1/\mathbb{G}_2$ exponentiations and pairing evaluations. This enables us to computationally determine a reliable upper bound on the runtime of our protocols: We calculate an upper bound on the number of respective atomic operations performed by each party, measure runtimes of single operations

on the type of hardware parties would realistically employ, and combine those values to attain an overall runtime estimation. Another relevant aspect for the runtimes of our system are precomputations. From the detailed protocol description in the full version (Schwerdt et al. 2019) of this paper, it is apparent that many of the more complicated operations can easily be precomputed before the start of the actual protocol. We therefore distinguish between online and offline operations: Operations that can be conducted before the start of the actual protocol are denoted as *offline operations*. For Part I of Debt Accumulation, these may be performed at any point before the EV is plugged into an SECC; for Part II the parties compute them during the charging process itself, before the user returns to retrieve their car. All operations that depend on inputs of the protocol and can therefore only be computed at the proper time are denoted as *online operations*. By precomputing as much as possible the online runtime of the actual protocol can be significantly shortened and offline runtimes flexibly scheduled to convenient time slots. Table 2 shows the number of atomic operations in the Debt Accumulation task, differentiated by parties and offline and online computations.

For the verification of NIZK proofs, we assume batch verification techniques from (Herold et al. 2017) to significantly speed up the verification process. Also note, that the number of operations needed for verification was generously estimated. Therefore, the values in Table 2 are upper bounds for the computational costs rather than exact figures.

To get a more tangible measurement of runtimes, we need to make assumptions about the hardware used in a realization of our system. For an EVCC we consider a conventional on-board unit, like the Savari MobiWAVE-1000 (Savari.net 2017), to be a realistic choice. We therefore measured the runtime of $\mathbb{G}_1/\mathbb{G}_2$ exponentiations and pairing evaluations on the type of processor present in this on-board unit: An i.MX6 Dual-Core processor running at 800MHz with 1GB DDR3 RAM and 4GB eMMC Flash. The processor runs an embedded Linux and is ARM Cortex-A9 based (32-bit). For the bilinear group setting, we use the Barreto-Naehrig curves Fp254BNb and Fp254n2BNb (Barreto and Naehrig 2006; Kawahara et al. 2016) (with 254-bit order) and the optimal Ate pairing (Moody et al. 2015). The resulting benchmarks for single exponentiations in $\mathbb{G}_1$ and $\mathbb{G}_2$ and for pairing evaluations are 6.07ms, 15.52ms and 32.19ms respectively.

For use in SECCs the Sitara AM335x processor—based on an ARM Cortex-A8 (32-bit) running at up to 1GHz—has been suggested (Instruments 2017). Since this processor has a similar core to the ones of the i.MX6, we use the same benchmarks to estimate runtimes for the SECC[7]. To realize UAs we propose to use either smart phones or a smart card plugged into the EVCC to utilize its computing power. Since the latter yields slower runtimes, we choose this option to obtain an upper bound on the realistic runtime. We therefore use the above benchmarks to calculate the runtimes for the EVCC and SECC, as well as the UA. Combining the figures from Table 2 with the benchmarks we can now calculate upper bounds on the runtime of both parts of the task Debt Accumulation. The results are depicted in Table 3.

These results show that the only time-critical part—the online part of Part II—takes significantly less than one second to complete. All other parts do not compel the user to

---

[7]Note that they were indeed measured on a single thread only.

Schwerdt *et al. Energy Informatics* (2019), 2(Suppl 1): 32

Page 16 of 21

Table 2 Upper bound on $\mathbb{G}_1/\mathbb{G}_2$ exponentiations and pairing evaluations in debt accumulation

	Party	Offline			Online		
		$\mathbb{G}_1$	$\mathbb{G}_2$	Pairing	$\mathbb{G}_1$	$\mathbb{G}_2$	Pairing
Part I	EVCC	91	82	0	$42+2v$	33	0
	SECC	v	0	0	0	0	$140+y$
Part II	UA	240	228	0	4	0	$z+9$
	SECC	4	0	$252+2j$	3	9	10

Here, $j := |a_\lambda|, y := |a_{\mathcal{E}}|, z := |a_{\mathcal{C}}|$ and $v := |bl_{EVCC}|$

wait for completion and with roughly 2, 6 and 13 s they lie well within acceptable time-frames. Although these times are already sufficient to not impede a user's experience with our system, we note that they are still upper bounds in several ways: They were obtained assuming a very naive implementation without any computational optimizations, the number of group operations for proof verification was generously overestimated and runtimes were calculated using a single core only, when realistically multiple cores would be available. Hence, we assume real runtimes to be even better than the figures given in this section.

Security and privacy

This section discusses the security and privacy properties of our P6V2G protocol. After reviewing the inherent limitations implied by the underlying scenario, we individually detail the security and privacy properties our protocol was proven to provide. Finally we sketch how the formal proof was conducted.

Inherent privacy and security limitations

To assess the level of security and privacy our protocol provides, we first discuss which security limitations and privacy losses are inherent in the chosen setting and what levels of security and privacy an optimal solution might be able to yield.

   Privacy Although ideally we would like OPRs to learn nothing at all, there are things we can not hope to hide from them due to their control over the EVSEs. Information they will always obtain for a charging session are the ID and location of the EVSE, time and duration of charging, charge targets put in by the (anonymous) user, the EV's battery properties, and the actual SDR[8] of the session. A privacy-optimal solution would not give away anything more than those parameters.

   Security Some security limitations are imposed by post-payments paired with the semi online setting of SECCs. Facilitating this in a privacy preserving manner requires users to collect their debt on a wallet instead of trusting the OPR with their information. In this setting it is always possible for a user to reuse an old wallet state for upcoming charging sessions. Without instant synchronization the participating SECC has no way of knowing the old wallet state has already been used before. This kind of misbehavior is called *double-spending* and is a widely accepted drawback of semi online payment systems like bitcoin or e-cash—regardless of application. An optimal solution will detect this unavoidable fraud after the fact, identify the misbehaving user if required, and be able to recalculate the debt missing from the user's wallet. This feature is also required in case a

---

[8]Of course, this SDR should only contain details of the conducted charging, but no identifying information.

Table 3 Upper Bound on Runtimes of Debt Accumulation

	Party	Party runtimes		Total runtime	
		offline	online	offline	online
Part I	EVCC	1825 ms	1374 ms	2129 ms	5945 ms
	SECC	304 ms	4571 ms		
Part II	UA	4995 ms	346 ms	13389 ms	826 ms
	SECC	8394 ms	480 ms		

Here, $j := |a_\lambda| = 4, y := |a_{\mathcal{E}}| = 1, z := |a_{\mathcal{C}}| = 1$ and $v := |bl_{EVCC}| = 50$

fraudulent user refuses to present their wallet for billing or claims to have lost it, which can not be prevented either. Another inherent limitation is that a maliciously colluding user and SECC are always able to agree on updating the wallet in a way that is not reflective of any charging session that physically took place. Due to this gap between the digital communication captured by our model and the real, physical world, we had to omit the case of collusions between the user side (UA, EVCC) and operator side (OPR, SECC) from our security proof. We suggest mitigating the effects of corrupted SECCs by adding timestamps to their attributes so that keys of corrupted SECCs expire. Note also that the corrupt behavior of a colluding user and SECC (or OPR) has no more consequences for the security and privacy of *honest* users than if only the SECC/OPR were corrupted, which is covered by our proof.

Proven privacy properties

After listing some scenario specific impossibilities, let us discuss how P6V2G can yield nearly optimal privacy in this setting. Our proof of user privacy asserts that in addition to the necessary information listed in "Inherent privacy and security limitations" section, the only thing SECCs (and therefore OPRs) learn from each charging session are the wallet's, EV's and previous SECC's attributes as well as the certifying OPRs. Content of these attributes is a subject of choice in implementation and fully determines the level of privacy provided by the system. While empty attributes yield complete privacy, it would also be possible to, e.g., include the users/EVs/EVSEs identities in their respective attributes, and hence implement a fully identifying system. The important point is that this level of privacy is explicit and easy to check upon registration and charging[9]. Realistically, wallet's attributes might contain the billing period (e.g., month) they are valid for and information on whether the user has a business or private contract with his operator. EV's attributes might differentiate between standard cars and busses. An SECC's attributes should consist of the current billing period only, leaving all honest SECCs with the same attributes. Simple properties like these yield that any recorded transaction can only be attributed to a group of k different and indistinguishable pairs of UAs/EVs—with k being the number of UAs with the same wallet attributes and OPR multiplied by the number of EVs of the same type—but not to any specific user in this group.

Assuming this kind of attributes we formally prove the following privacy properties, even under participation of malicious OPRs/SECCs or UAs/EVs in the system:

---

[9]Once the system is implemented with a specific form of attributes, checking the content of attributes is easily automated (and we would advise to do so), saving users the effort of checking the system's privacy level themselves.

(P1)	Charging sessions of honest UAs/EVs are anonymous and unlinkable.
(P2)	Tracing of an honest UA and recalculation of its accumulated debt are impossible without cooperation of the DR.
(P3)	Tracing of a misbehaving UA has no more implications for the privacy of all other UAs/EVs than if the traced UA had never participated in the system at all.

Proven security properties

We consider static corruption (and collusion) of an arbitrary number of either UAs and EVCCs or of SECCs and OPRs (for an explanation on UAs/EVCCs colluding with SECCs/OPRs, see "Inherent privacy and security limitations" section). Any corrupted party may maliciously deviate from the protocol instead of just acting in an honest-but-curious manner.

Under this kind of corruption we prove the following security properties:

(S1)	A UA can only use wallets that were legitimately issued to this UA.
(S2)	If a UA commits double-spending (see "Inherent privacy and security limitations" section), it will be identified.
(S3)	A UA which does not commit double-spending can not lie about or modify the debt or reputation on its wallet.
(S4)	An OPR is able to blacklist a specific UA with cooperation from the DR. An SECC will know that a UA is blacklisted before its wallet is used for payment.
(S5)	An OPR is able to recalculate a specific UA's debt with cooperation from the DR.
(S6)	A UA that does not clear its wallet's debt will be detected.
(S7)	Only registered EVs can take part in Debt Accumulation.
(S8)	An OPR is able to blacklist specific EVs with cooperation from the DR. An SECC will know that an EV is blacklisted at the start of a charging session.
(S9)	No party can lie about or modify their attributes.

Proof sketch

We conducted a simulation-based proof following the real/ideal paradigm. For this type of proof the execution of the real world protocol is compared to the execution of an ideal model where all computation is done by an uncorruptible ideal functionality. The idea is for the ideal functionality to be designed in a way that it obviously guarantees the desired security and privacy properties. A proof of indistinguishability of the real world and ideal model yields that the protocol provides these properties as well. The advantage over game-based approaches is that all security properties and restrictions are explicit. However, the traditional notion of simulation-based security only captures security requirements in a standalone setting, where a single protocol instance runs in isolation. If multiple protocol instances run concurrently, this approach fails to guarantee security. The Universal Composability (UC) framework (Canetti 2001) on the other hand does not only capture parallel execution of multiple protocol instances, but also the use of protocols in an arbitrary context.

For our protocol [$\mathcal{F}_{\mathrm{CRS}}, \overline{\mathcal{G}}_{\mathrm{BB}}$]-hybrid model (cp. (Canetti and Fischlin 2001; Canetti et al. 2016)) that it securely realizes the ideal V2G functionality $\mathcal{F}_{\mathrm{V2G}}$ under common hardness assumptions for cryptographic building blocks. Further information on the ideal

functionality and additional details of the proof can be found in the full version (Schwerdt et al. 2019) of this paper.

Conclusion and future work

Based on the cryptographic toll collection framework P4TC (Hoffmann et al. 2018) we were able to develop the privacy-preserving V2G payment and reputation scheme P6V2G. Our system facilitates two-way transactions between EV users and EVSEs as well as reputation scores. In offering post-payment functionality—with, e.g., monthly billing—P6V2G is a highly convenient system with low effort usability. We achieve real world practicality by offering double-spending detection and guilt verification features, contingency traceability (with the consent of a designated authority) as well as blacklisting of EVs and users. All the systems functions are subject to explicit and formally proven privacy and security guarantees. Furthermore SECCs do not require permanent online capabilities, as P6V2G is designed to be operated in a semi online setting. Our runtime analysis demonstrates the P6V2G protocols to be efficient enough for the low performance hardware that is realistically present in the V2G setting. With EVs and users modeled as two separate entities, the system is fully portable between EVs and hence supports users with multiple cars as well as car sharing between multiple users.

Although the P6V2G system is a complete and practical V2G payment scheme, there are several directions where further expansion might prove beneficial. Modelling the EV and the actual charging process in more detail could yield additional features, e.g., an automated pricing check and continuous monitoring of the SECC's charging instructions. Another direction with potential for further development would be the financial settlement between charging point and e-mobility OPRs. At the moment each SECC needs to learn the user's eMO during a charging session to realistically provide OPRs with the means for financial settlement between them. Appending P6V2G with a cryptographic protocol for financial settlement between OPRs could remove the need for this information and thus further improve privacy.

Authors' contributions
Under supervision of RS and MN, TG developed a preliminary version of this work in the scope of his master's thesis. RS advanced this first draft to its final version and wrote most of the manuscript. MN contributed to this revision by providing key insights into the finer details of the UC-model and corresponding proofs, as well as by refining the EVCC blacklisting mechanism of the protocol. VF discussed the results, analyzed the protocol's efficiency, and wrote the protocol descriptions for the final manuscript. AR discussed the results and reviewed the manuscript. All authors have read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

References
Au MH, Liu JK, Fang J, Jiang ZL, Susilo W, Zhou J (2014) A new payment system for enhancing location privacy of electric vehicles. IEEE Trans Veh Technol 63(1):3–18
Baldimtsi F, Chase M, Fuchsbauer G, Kohlweiss M (2015) Anonymous transferable e-cash. In: Katz J (ed). Public-Key Cryptography – PKC 2015. Springer, Berlin. pp 101–124
Barreto PSLM, Naehrig M (2006) Pairing-friendly elliptic curves of prime order. In: Preneel B, Tavares S (eds). SAC 2005 Annual International Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 3897. Springer, Berlin. pp 319–331
Blanchet B (2016) Modeling and verifying security protocols with the applied pi calculus and proverif. Found Trends Priv Secur 1(1-2):1–135
Burch I, Gilchrist J (2018) Survey of Global Activity to Phase Out Internal Combustion Engine Vehicles. Cent Clim Prot
Camenisch J, Hohenberger S, Lysyanskaya A (2005) Compact e-cash. In: Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. pp 302–321
Canard S, Gouget A (2008) Anonymity in transferable e-cash. In: Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings. pp 207–223
Canetti R (2001) Universally composable security: A new paradigm for cryptographic protocols. In: 42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA. pp 136–145
Canetti R, Dodis Y, Pass R, Walfish S (2007) Universally composable security with global setup. In: Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings. pp 61–85
Canetti R, Fischlin M (2001) Universally composable commitments. In: Kilian J (ed). Advances in Cryptology — CRYPTO 2001. Springer, Berlin. pp 19–40
Canetti R, Shahaf D, Vald M (2016) Universally composable authentication and key-exchange with global pki. In: IACR International Workshop on Public Key Cryptography. Springer. pp 265–296
Department of Public Expenditure and Reform (2019) Project Ireland 2040: National Development Plan 2018-2027. Gov Declaration
DIN-Normenausschuss Automobiltechnik (2015) Road Vehicles – Vehicle to Grid Communication Interface – Part 1: General Information and Use-case Definition. Deutsches Institut für Normung e.V., Berlin. Deutsches Institut für Normung e.V.. DIN EN ISO 15118-1
DIN-Normenausschuss Automobiltechnik (2016) Road Vehicles – Vehicle to Grid Communication Interface – Part 2: Network and Application Protocol Requirements. Deutsches Institut für Normung e.V., Berlin. Deutsches Institut für Normung e.V.. DIN EN ISO 15118-2
Fazouane M, Kopp H, van der Heijden RW, Métayer DL, Kargl F (2015) Formal verification of privacy properties in electric vehicle charging. In: ESSoS. Lecture Notes in Computer Science, vol. 8978. Springer, Cham. pp 17–33
Foley A, Winning I, O Gallachoir B (2010) State-of-the-art in electric vehicle charging infrastructure. In: 2010 IEEE Vehicle Power and Propulsion Conference (VPPC). IEEE. pp 1–6
Golle P, Partridge K (2009) On the anonymity of home/work location pairs. In: Tokuda H, Beigl M, Friday A, Brush AJB, Tobe Y (eds). Pervasive Computing. Springer, Berlin. pp 390–397
Han W, Xiao Y (2016) Privacy preservation for v2g networks in smart grid: A survey. Comput Commun 91:17–28
Hartung G, Cryptographic AccumulatorsNg Matthias Nagel MH, Rupp A (2017) BBA+: improving the security and applicability of privacy-preserving point collection. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. pp 1925–1942
Herold G, Hoffmann M, Klooß M, Ràfols C, Rupp A (2017) New techniques for structural batch verification in bilinear groups with applications to groth-sahai proofs. In: ACM Conference on Computer and Communications Security. pp 1547–1564
Höfer C, Petit J, Schmidt RK, Kargl F (2013) POPCORN: privacy-preserving charging for emobility. In: CyCAR@CCS. ACM, New York. pp 37–48
Hoffmann M, Fetzer V, Nagel M, Rupp A, Schwerdt R (2018) P4TC—Provably-Secure yet Practical Privacy-Preserving Toll Collection. Cryptology ePrint Archive, Report 2018/1106. https://eprint.iacr.org/2018/1106. Accessed Feb 2019
Instruments T (2017) TI Designs: TIDEP-0087. Human Machine Interface (HMI) for EV Charging Infrastructure Reference Design. http://www.ti.com/lit/ug/tidude7/tidude7.pdf. [Online; accessed 09-Jan-2019]
Kawahara Y, Kobayashi T, Scott M, Kato A (2016) Barreto-naehrig curves. Internet draft, Internet Engineering Task Force. Work in Progress
Kempton W, Letendre SE (1997) Electric vehicles as a new power source for electric utilities. Transp Res A D Transp Environ 2(3):157–175
Krumm J (2007) Inference attacks on location tracks. In: LaMarca A, Langheinrich M, Truong KN (eds). Pervasive Computing. Springer, Berlin. pp 127–143
Liu JK, Au MH, Susilo W, Zhou J (2012) Enhancing location privacy for electric vehicles (at the right time). In: European Symposium on Research in Computer Security. Springer. pp 397–414
Löfven S (2019) Regeringsförklaring. Gov Declaration
Moody D, Peralta RC, Perlner RA, Regenscheid AR, Roginsky AL, Chen L (2015) Report on pairing-based cryptography. In: Journal of Research of the National Institute of Standards and Technology, vol. 120. National Insititute of Standards and Technology, Gaithersburg. pp 11–27
Savari.net (2017) MobiWAVE On-Board-Unit (OBU). http://savari.net/wp-content/uploads/2017/05/MW-1000_April2017.pdf. [Online; accessed 05-Feb-2018]

Schwerdt *et al. Energy Informatics* (2019), 2(Suppl 1): 32

Page 21 of 21

Schwerdt R, Nagel M, Fetzer V, Gräf T, Rupp A (2019) P6V2G: A Privacy-Preserving V2G Scheme for Two-Way Payments and Reputation. Cryptology ePrint Archive, Report 2019/786. https://eprint.iacr.org/2019/786

Wan Z, Zhu W-T, Wang G (2016) Prac: Efficient privacy protection for vehicle-to-grid communications in the smart grid. Comput Secur 62:246–256

Wang H, Qin B, Wu Q, Xu L, Domingo-Ferrer J (2015) Tpp: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids. IEEE Trans Inf Forensic Secur 10(11):2340–2351

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.