Janus: Privacy-Preserving Billing for Dynamic Charging of Electric Vehicles

Hongyang Li
University of Illinois Urbana-Champaign
hli52@illinois.edu

György Dán
KTH Royal Institute of Technology
gyuri@kth.se

Nikita Borisov
University of Illinois Urbana-Champaign
nikita@illinois.edu

Klara Nahrstedt
University of Illinois Urbana-Champaign
klara@illinois.edu

Carl A. Gunter
University of Illinois Urbana-Champaign
cgunter@illinois.edu

***Abstract*—Dynamic charging is an emerging technology that allows an electric vehicle (EV) to charge its battery while moving along the road. Dynamic charging charges the EV's battery through magnetic induction between the receiving coils attached to the EV's battery and the wireless charging pads embedded under the roadbed and operated by Pad Owners (POs). A key challenge in dynamic charging is billing, which must consider the fact that the charging service happens while the EV is moving on the road, and should allow for flexible usage plans. A promising candidate could be the subscription-based billing model, in which an EV subscribes to an electric utility that has a business relationship with various POs that operate charging sections. The POs report charging information to the utility of the EV, and at the end of each billing cycle, the EV receives a single bill for all its dynamic charging sessions from the utility. Overshadowing its advantages, a major shortcoming of such a solution is that the utility gets access to the EVs' mobility information, invading thus the location privacy of the EVs.**

To enable subscription based billing for dynamic charging, in this paper we propose Janus, a privacy-preserving billing protocol for dynamic EV charging. Janus uses homomorphic commitment and blind signatures with attributes to construct a cryptographic proof on the charging fee of each individual dynamic charging session, and allows the utility to verify the correctness of the EV's total bill without learning the time, the location, or the charging fee of each individual charging session of the EV. Our Python-based implementation shows that the real-time computational overhead of Janus is less than 0.6 seconds, which is well within the delay constraint of the subscription-based billing model, and makes Janus an appealing solution for future dynamic charging applications.

I. Introduction

Dynamic charging is an emerging technology that enables charging electric vehicles' (EVs) batteries on the go. Dynamic charging works by installing a series of short charging pads under the road, which together form a charging section. While a charging pad would be in the order of tens of centimeters long, a typical charging sections could be several kilometers long. The EV has receiving coils attached to its battery, and as it drives through the charging section, the magnetic induction between the coil on the EV and the coils in the charging pad enable to charge the EV's battery. Dynamic charging provides several benefits including increasing the driving range of the EV and reducing its battery size, as the EV can charge its battery while moving [13].

In lack of physical contact and due to mobility, a significant challenge in dynamic charging is billing. In static charging where the EV stops at a charging station for minutes, the driver can make the payment for the charging session at the point-of-sale [2]. For dynamic charging such a billing model is impractical, as the EV is on the move while the battery charging happens. A more appealing billing model for dynamic charging is the subscription-based billing model, which is similar in spirit to today's cellular service: the EV subscribes to a utility who has business relationship with various Pad Owners (POs) that operate charging sections. At the end of each billing cycle, the EV receives a single bill for all its dynamic charge sessions in the past billing cycle from its utility. The EV then settles the bill with its utility and the utility settles its bills with the POs.

The subscription-based billing model provides several advantages for the EV owner: (i) with a single entity having a complete view of all the EV's charging activities, including charging at home, at parking lots, at commercial charging stations, and dynamic charging on the road, it enables value added services such as battery health monitoring; (ii) it allows to the EV to be treated as one of the user's home appliances; and (iii) it enables receiving a flexible pricing plan. Having a complete view of the EV's charging activities allows the utility to better manage the charging demands and to reduce peak load, while the ability to treat the EV as a home appliance enables flexibility in pricing plans. For instance, if the user chooses to join the vehicle-to-grid (V2G) program that helps the utility to reduce peak load, the utility could apply discounts to the EV's dynamic charging bill. The subscription-based billing model also enables a flexible pricing plan similar to the data plan model in today's cellular service. For example, the EV could purchase a plan of 1000 miles from the utility, and the EV can use dynamic charging anywhere anytime to recharge its battery up to 1000 miles of total driving distance.

Subscription-based billing does, however, pose a significant threat to the EVs' location privacy. The naive approach where the EV or the PO submits to the utility the location and time of each charging session in order for the utility to calculate the total bill would allow the utility to learn the exact time and location of each dynamic charging session. This information could possibly reveal interests, driving habits, health conditions and economic status, which clearly invades

the EV owner's privacy. Thus, to make subscription-based billing models applicable for dynamic EV charging, it is essential for the billing protocol to preserve the EV's location privacy. Existing solutions developed for privacy-preserving toll pricing [4], [9], [15], [18], public transportation [19] and static charging [2] cannot be applied to the use case of dynamic EV charging, partly due to the different nature of the service and due to mobility.

In this paper we propose Janus, a privacy-preserving billing protocol for dynamic charging of electric vehicles. A key feature of Janus is that in the bill calculation process the utility is not able to learn when and where an EV has used dynamic charging. Janus achieves this by embedding a homomorphic commitment of the price for each charging session as attributes in blind signatures signed by the utility. The cryptographic constructions allow the EV and the PO to compute their total bill locally, without involving any trusted third party, and prove the correctness of the total bill to the utility. We implement Janus in Python based on petlib [1] and evaluate its execution time on a combination of Raspberry Pi and Macbook hardware and software. Our results show that all billing process computations can be done within 0.6 seconds, which is well within the delay constraint for the considered billing model.

Our main contributions can be summarized as follows: (i) we propose a privacy-preserving billing protocol, Janus, for dynamic charging of electric vehicles under the subscription-based billing model [1]; to our best knowledge this is the first privacy-preserving billing protocol proposed for dynamic charging; (ii) we provide experimental results from a Python implementation that demonstrate the practical feasibility of Janus, (iii) we provide an analysis of the location privacy provided by Janus and, in general, by any protocol revealing aggregate information; and (iv) we provide a comparison of Janus with other privacy-preserving billing solutions such as electronic toll pricing and digital cash.

The rest of our paper is organized as follows. In Section II we describe the system model and assumptions. In Section III we introduce key security building blocks used in Janus. In Section IV we summarize the notations, and in Section V we describe the Janus protocol. In Section VI we evaluate the running time and message overhead of Janus. In Section VII we analyze the security and privacy properties of Janus. We discuss related solutions in Section VIII, and in Section IX we review related work. We conclude the paper in Section X.

II. Background and System Model

A. Dynamic EV Charging

We define a *dynamic charging section* to be a road segment under which the wireless charging pads are installed. To facilitate power management, a charging section is typically a few kilometers long [8]. Within the charging section, short wireless charging pads (e.g., 40 cm long) are placed consecutively under the roadbed, with some tens of centimeters between adjacent pads. For efficiency, each charging pad can be individually switched on and off, independent of other charging pads, and a charging pad should only be switched on when there is an EV above it, and should switch off immediately after the EV moves away.

[1]Although Janus is designed for dynamic charging, it can also be applied to static charging of EVs, i.e., one could think of each static charging station as a dynamic charging section.

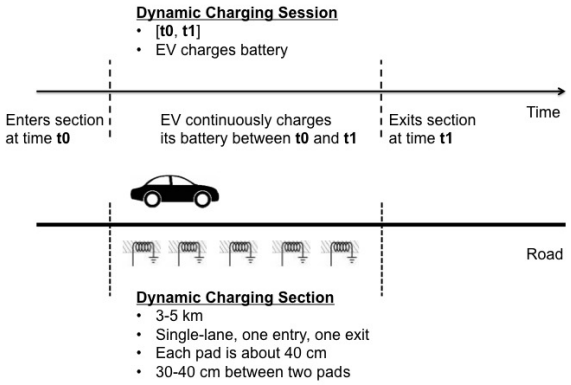

Fig. 1: Illustration of Dynamic Charging Section and Dynamic Charging Session.

We consider that each dynamic charging section is operated by some *Pad Owner* (PO). The PO may either produce electricity itself, or may purchase electricity from some electricity provider. A PO can own multiple dynamic charging sections, and there can exist multiple POs operating different dynamic charging sections in the same area.

We define a *dynamic charging session* between an EV e and a PO p to be the contiguous time period starting from the moment that EV e starts charging its battery from a dynamic charging section operated by PO p, until the moment that EV e stops charging from that charging section. During a single dynamic charging session the EV continuously charges its battery by moving over a series of wireless charging pads in the dynamic charging section. Figure 1 illustrates a dynamic charging section and a dynamic charging session.

B. Subscription-based Billing Model

The subscription-based billing model we consider is similar to the billing model used in today's cellular services. The billing model involves three types of entities: utilities, EVs, and POs. Following this model an EV pays a single bill to its contracted utility once every billing cycle rather than making individual payments for each dynamic charging session to the PO that owns the charging section.

The billing model consists of two operations: fee negotiation and fee aggregation.

- Fee negotiation happens prior to each dynamic charging session. The EV and the PO negotiate and agree on the charging fee that the EV should pay for the upcoming dynamic charging session.
- Fee aggregation happens once at the end of each billing cycle. The EV calculates and submits to its utility its total fee that it should pay to its utility, and the PO calculates and submits to the utilities the total fee that it should receive from them for charging their contracted EVs.

We illustrate the proposed billing model in Figure 2. EV 1 receives dynamic charging only once from PO A, and the

charging fee for that charging session is \$3. EV 2 is involved in one dynamic charging session with PO A for \$4, and another charging session with PO B for \$5. From the utility's perspective, the total bill for EV 1 would be \$3, and the total bill for EV 2 would be \$9 (= 4 + 5). Since PO A provided dynamic charging to both EV 1 and EV 2, the total fee that the utility should pay to PO A is \$7 (= 3 + 4). PO B only provided dynamic charging to EV 2, and receives \$5 from the utility.

C. Communication Model

We make the reasonable assumption that the EV can communicate wirelessly with its utility and with each PO through WiFi, DSRC, or a cellular network, and there is a secure connection (e.g., TLS/SSL) between the utilities and each PO through a broadband network infrastructures. We assume that prior to each charging session, the EV establishes a secure communication channel with the PO, e.g., by negotiating a temporary session key with the PO.

D. Security Model

We assume that the utilities are honest but curious, in that they faithfully follow the protocol but are interested to infer the location information of individual EVs. We assume that in the fee negotiation phase of the subscription-based billing model, the EV and the PO are able to agree on the fee for each individual charging session. However, in the fee aggregation phase at the end of each billing cycle, the EV may have an incentive to underclaim its total fee that should be paid to its utility, and the PO may have an incentive to overclaim the total fees that it should receive from the utilities.

III. Security Building Blocks

In this section we describe the security building blocks used in the construction of Janus.

A. Homomorphic Commitment

A commitment scheme allows a user to bind a secret value x to a commitment C. The commitment C itself is information-hiding in that C does not reveal any information about x. Nonetheless, the user can reveal the secret value x and can prove that C is indeed a commitment of x. One example of perfect information-hiding commitment is the Pedersen commitment scheme [17]: to commit a value x, the user chooses a random secret r and computes $C = g^x h^r$, where g and h are public. In order to prove that x is indeed committed in C, the user reveals x and r, and the verifier computes $C' = g^x h^r$ and checks that $C = C'$.

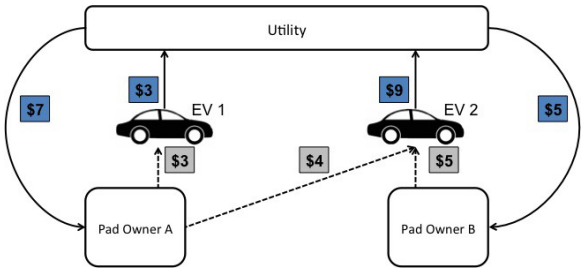

Fig. 2: Illustration of subscription-based billing model for dynamic charging. The fees in light-colored boxes beneath the EVs (\$3, \$4, \$5) are the results of fee negotiation. The fees in dark-colored boxes above the EVs (\$7, \$3, \$9, \$5) are the results of fee aggregation.

A homomorphic commitment scheme additionally allows one to obtain useful information by operating directly on commitments, without knowing the secret values in the commitments. A Pedersen commitment can be extended to a homomorphic commitment as follows: given $C_1 = g^{x_1} h^{r_1}$ and $C_2 = g^{x_2} h^{r_2}$, we define

$$C_1 \boxplus C_2 = C_1 C_2 = g^{x_1+x_2} h^{r_1+r_2} \quad (1)$$

Note that anyone can compute $C_1 \boxplus C_2$, given only C_1 and C_2. Later the user can prove that $x_1 + x_2$ equals to the claimed value, without revealing x_1 and x_2 themselves, by revealing $r_1 + r_2$. In Janus we use $Cmt(x_1, \ldots, x_n; r)$ to denote a Pedersen commitment with secret values $x_1, \ldots, x_n$ and commitment opener r.

B. Zero-Knowledge Proof

In a Zero-Knowledge Proof (ZKP) the prover proves to the verifier possession of certain secret values that satisfy certain relations. The proof is zero-knowledge in the sense that the proof itself does not reveal any additional information about the secret values. We use notation $ZK\{x : P(x)\}$ to denote a zero-knowledge proof of secret value x such that x satisfies the relation $P(x)$. Every variable that appears to the left of the colon is a secret value only known to the prover, and every variable that only appears to the right of the colon is public. For example, the following notation $ZK\{x, r : g^x h^r = C\}$ represents a zero-knowledge proof that the prover knows the secret value x and the commitment opener r that is used to form the Pedersen commitment C.

A zero-knowledge proof is interactive if it involves real-time interaction (e.g., message exchanges) between the prover and the verifier. One example is the Schnorr Identification scheme [5], which, given public values g and h, allows the prover to prove knowledge of value x such that $g^x = h$ without revealing x. One can transform an interactive zero-knowledge proof into a non-interactive zero-knowledge proof (NIZKP) using the Fiat-Shamir Heuristic [12].

C. Blind Signature with Attributes

A blind signature scheme allows the user to obtain a signature from the signer while the signer does not learn the content of the message to be signed. Baldimtsi and Lysyanskaya extended the definition of blind signature and introduced the concept of blind signature with attributes [5]. In addition to the message m, the user possesses certain secret attributes $L_1, \ldots, L_n$ and commits the attributes in a commitment $C = Cmt(L_1, \ldots, L_n; R)$ with commitment opener R. The commitment C is public while the message m and the attributes $L_1, \ldots, L_n$ are only known to the user. A blind signature with attributes would allow the user to obtain a signature σ on $(m, \tilde{C})$, where $\tilde{C}$ is a new commitment to the same attributes $L_1, \ldots, L_n$ but with a different opening secret $\tilde{R}$, i.e., $\tilde{C} = Cmt(L_1, \ldots, L_n; \tilde{R})$. The commitment opener $\tilde{R}$ of the new commitment $\tilde{C}$ is only known to the user. In

Janus we use Anonymous Credential Light (ACL) [5] as the implementation of the blind signature with attributes scheme.

D. Single-Use Anonymous Credentials

An anonymous credential allows the user to prove possession of the credential without revealing the user's true identity. A single-use anonymous credential further guarantees that the credential can be used at most once. If the user attempts to spend a single-use credential more than once, the user's true identity can be revealed. One way of constructing a single-use anonymous credential is to use blind signatures with attributes as described in [5]. To obtain a single-use anonymous credential, the user first generates a random secret L_0, and constructs a commitment $C = Cmt(L_0, L_1; R)$. The user then proves to the signer knowledge of L_0, L_1, R with respect to C and obtains a blind signature σ on $(m, \tilde{C})$, where m is a random message and $\tilde{C} = Cmt(L_0, L_1; \tilde{R})$. To spend the credential, the user reveals $m, \tilde{C}, \sigma$, receives a challenge c from the verifier, and then reveals the double-spending factor $d = cL_1 + L_0$. The verifier verifies that σ is a valid signature on $(m, \tilde{C})$. Note that if the user attempts to spend the same credential twice, the verifier would know $d = cL_1 + L_0$ and $d' = c'L_1 + L_0$, from which the user's true identity can be inferred as $L_1 = \frac{d-d'}{c-c'}$.

IV. Notation

Before we introduce the Janus protocol, we summarize the key notations.

- H: one-way hash function.
- e: EV's true identity.
- p: PO's true identity.
- u: Utility's true identity.
- (e, p, u, i, j): index of the considered charging session, which is the i-th charging session of EV e, and the j-th charging session of PO p with any EV subscribed to utility u.
- $P = P_i^e = P_j^{p,u}$: the three variables all denote the same charging fee for a particular charging session agreed by EV e and PO p.
- $Cmt(L_0, L_1, \ldots, L_n; R)$: Pedersen commitment of attributes $(L_0, L_1, \ldots, L_n)$ with commitment opener R.
- $\pi = NIZK\{(x_1, \ldots, x_n) : P(x_1, \ldots, x_n, y_1, \ldots, y_m)\}$: Non-Interactive Zero-Knowledge proof of knowledge. The prover proves knowledge of secret values $x_1, \ldots, x_n$ that satisfy the relation $P(x_1, \ldots, x_n, y_1, \ldots, y_m)$, where $y_1, \ldots, y_m$ are public values.
- $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$: single-use anonymous credential issued by the utility to EV e during the registration phase.
- sn_i^e: a random secret generated by EV e.
- $\hat{C}_i^e$: $\hat{C}_i^e = Cmt(L_i^e, e; \hat{R}_i^e)$ is a Pedersen commitment with random secret L_i^e and EV's identity e as committed values.
- L_i^e: random secret generated by EV e.
- c: challenge used in the double-spending equation $d_{i,j}^{e,p,u}$.
- $d_{i,j}^{e,p,u}$: $d_{i,j}^{e,p,u} = c \cdot e + L_i^e$ is the double-spending equation that allows identification of the EV's identity e if the same single-use credential τ_i^e is spent twice.
- $\hat{\sigma}_i^e$: utility's ACL signature on $(sn_i^e, \hat{C}_i^e)$.
- $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$: the receipt of EV e.
- σ_i^e: utility's ACL signature on $(m_i^e, \tilde{C}_i^e)$.
- m_i^e: $m_i^e = Cmt(e; z_i^e)$ is a Pedersen commitment with the EV's identity e as the committed value, and commitment opener z_i^e that is only known to EV e.
- $\tilde{C}_i^e$: $\tilde{C}_i^e = Cmt(P_i^e; \tilde{R}_i^e)$ is a Pedersen commitment with the charging fee P_i^e as the committed value, and commitment opener $\tilde{R}_i^e$ known to both PO p and EV e.
- $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$: the receipt of PO p.
- $\sigma_j^{p,u}$: utility's ACL signature on $(m_j^{p,u}, \tilde{C}_j^{p,u})$
- $m_j^{p,u}$: $m_j^{p,u} = Cmt(p; z_j^{p,u})$ is a Pedersen commitment with the PO's identity p as the committed value, and commitment opener $z_j^{p,u}$ that is only known to PO p.
- $\tilde{C}_j^{p,u}$: $\tilde{C}_j^{p,u} = Cmt(P_j^{p,u}; \tilde{R}_j^{p,u})$ is a Pedersen commitment with the charging fee $P_j^{p,u}$ as the committed value, and commitment opener $\tilde{R}_j^{p,u}$ only known to PO p.

V. Janus Protocol

A. Design Goals and Protocol Overview

Janus is designed to enable privacy-preserving dynamic EV charging, and as such it has two major design goals: correctness and privacy.

- Correctness: Janus must allow the utility to verify that the total fee submitted by the EV and the PO to the utility in the fee aggregation phase of the billing model is consistent with the charging fees of each individual charging session. In particular, the EV should be able to prove to the utility that it does not underclaim the total fee, and the PO should be able to prove to the utility that it does not overclaim the total fee.
- Privacy: Janus should only provide information to the utility that is needed for settlement, and the information should not be sufficient to infer location information of individual EVs. In particular, for each individual charging session, the time of the charging, the identity of the PO, and the charging fee for this individual charging session should be infeasible to infer for the utility.

To be compatible with subscription-based billing, at a high level Janus consists of three phases: registration, price validation, and reconciliation. The purpose and operation of these phases is as follows.

- The registration phase happens once at the beginning of each billing cycle between each EV e and its utility u, where the utility issues N single-use credentials $\tau_1^e, \ldots, \tau_N^e$ to EV e. For each dynamic charging session the EV will have to spend one unused credential.
- The price validation phase happens at the beginning of each dynamic charging session (after the EV and

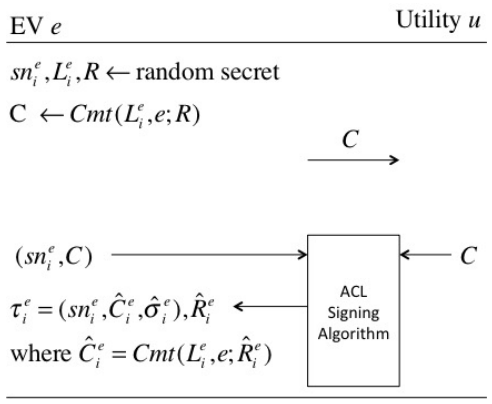

Fig. 3: Overview of Registration phase of Janus. The above protocol shows how EV e obtains one single-use anonymous credential $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$ from its utility. To obtain a total of N credentials, the EV repeats the above protocol N times at the beginning of the billing cycle.

the PO have completed fee negotiation and agreed on the charging fee for the upcoming charging session, and before the actual charging happens), where EV e, PO p and utility u run the price validation protocol described in Section V-C. As a result, EV e obtains the receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and PO p obtains the receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ on the price $P_i^e = P_j^{p,u} = P$ from the utility u, where σ_i^e is the utility's signature on the pair $(m_i^e, \tilde{C}_i^e)$, and $\sigma_j^{p,u}$ is the utility's signature on the pair $(m_j^{p,u}, \tilde{C}_j^{p,u})$. The price validation protocol uses blind signature so that the utility does not learn either $m_j^{p,u}$ or $\tilde{C}_j^{p,u}$ during the signing process. The receipt proves that both the EV and the PO agreed on the price P, and the utility's signature prevents the EV and the PO from modifying the receipt.

- The reconciliation phase happens once at the end of the billing cycle. During this phase EV e submits the total price $P^e = \sum_{i=1}^{M^e} P_i^e$ and all its receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ to the utility, where M^e is the number of charging sessions of EV e during the billing cycle. PO p also submits the total price $P^{p,u} = \sum_{i=1}^{M^{p,u}} P_j^{p,u}$ and the validation tokens $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$, where $M^{p,u}$ is the number charging sessions provided by PO p to EVs of utility u. In addition each EV e also needs to reveal any unused credentials $\tau_{M^e+1}^e, \ldots, \tau_N^e$ to its utility u.

In what follows we explain these three phases in detail.

B. Registration

The registration phase happens once at the beginning of each billing cycle. The main purpose of this phase is for the utility to issue anonymous credentials to the EV that will be used in the price validation phase. The EV can authenticate with the utility using its true identity (e.g., its long-term public key) at the beginning of the registration phase, and thus the utility knows the EV's identity during the communication of the registration phase (but the utility does not learn the anonymous credentials issued to the EV until the EV spends them). During the registration phase, each EV e is issued N credentials for the billing cycle, where N is large enough for all charging sessions of an arbitrary EV. In practice, to make sure that an EV will not run out of credentials during the billing cycle, the number N can be estimated using traffic statistics and study of EV charging behavior. Note that, since Janus requires each EV to submit all unused credentials at the end of the billing cycle (explained later in Section V-D), it is important that all EVs are issued the same number of credentials; otherwise the utility can identify the EV according to the total number of credentials it is issued at the beginning of the billing cycle.

In Figure 3 we illustrate how EV e obtains a single-use credential $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$ from the utility as described in [5]. EV e obtains N credentials by repeating the protocol N times. The protocol itself consists of three steps.

- Step 1: to obtain the i-th credential, the EV first generates random secrets sn_i^e, L_i^e, and R. sn_i^e is a serial number that serves as the message to be signed, and L_i^e together with the EV's true identity e serve as the attribute, as described in Section III-C.
- Step 2: the EV commits L_i^e and its identity e in the commitment $C = Cmt(L_i^e, e; R)$ using secret R.
- Step 3: the EV runs the ACL signing protocol with the utility as the signer on the message sn_i^e and attribute commitment C. As a result, EV e obtains $\hat{C}_i^e, \hat{R}_i^e, \hat{\sigma}_i^e$, where $\hat{C}_i^e$ is a commitment to the same attributes (L_i^e, e) but with a different secret $\hat{R}_i^e$, i.e., $\hat{C}_i^e = Cmt(L_i^e, e; \hat{R}_i^e)$, and $\hat{\sigma}_i^e$ is the utility's signature on the pair $(sn_i^e, \hat{C}_i^e)$.

Note that during the signing process the utility does not learn the value of L_i^e, e or the new commitment $\hat{C}_i^e$, and the output signature $\hat{\sigma}_i^e$ cannot be linked to this signing session.

C. Price Validation

Price validation happens between the EV and the PO. Note that when the EV uses the dynamic charging service provided by some PO, the PO can physically observe the EV at its charging section, and thus there is no reason to hide the EV's location information from the PO. A malicious PO can indeed disclose the identity of the observed EV to any utility or other entities. Such malicious PO behavior is out of the scope of Janus, and thus we consider that the EV trusts the PO not to disclose the EV's identity and location to any third party. As a consequence, we can rely on the PO to act as a proxy and relay messages between the EV and its utility, and thus the EV does not communicate directly with the utility during the price validation phase.

In Figure 4 we illustrate the price validation protocol through showing the message exchange for the i-th charging session of EV e in the current billing cycle, which happens to be the j-th charging session of PO p with any EV subscribed to utility u. We refer to this dynamic charging session as session (e, p, u, i, j). Once EV e and PO p have agreed on the price P for this charging session, EV e records the price $P_i^e = P$ and the PO records the price $P_j^{p,u} = P$. The goal of the price validation phase is to let EV e obtain receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$

and PO p to obtain receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ on the price $P_i^e = P_j^{p,u} = P$ from the utility u. To preserve the EV's privacy, we rely on the ACL blind signature with attributes [5]. This way, during the signing process, the utility does not learn the value of the price P, the EV's identity e, and cannot link the produced receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e), (m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ with the signing session.

The price validation protocol consists of four major steps: PO preparation, EV preparation, receipt generation, and EV validation.

1) PO Preparation: In this step, the PO prepares two commitments of the charging fee $P = P_i^e = P_j^{p,u}$.

- Step 1: PO receives a random nonce n generated by the utility.
- Step 2: PO generates a random secret $z_j^{p,u}$ and commits its identity p in $m_j^{p,u} = Cmt(p; z_j^{p,u})$.
- Step 3: PO generates two secrets $R_i^e, R_j^{p,u}$ and commits the price $P = P_i^e = P_j^{p,u}$ in the commitments $C_i^e = Cmt(P_i^e; R_i^e)$ and $C_j^{p,u} = Cmt(P_j^{p,u}; R_j^{p,u})$.
- Step 4: PO sends $C_i^e, R_i^e, C_j^{p,u}, R_j^{p,u}, n$ to the EV.

2) EV Preparation: In this step, the EV spends an anonymous credential τ_i^e and constructs the non-interactive zero-knowledge proof accordingly.

- Step 1: EV verifies that the two commitments $C_i^e, C_j^{p,u}$ are formed correctly.
- Step 2: EV binds the nonce n generated by the utility with the two commitments into $c = H(n, C_i^e, C_j^{p,u})$ using a one-way function H, and constructs the double-spending equation $d_{i,j}^{e,p,u} = c \cdot e + L_i^e$. The EV is essentially using c as the challenge to spend its single-use credential $\tau_i^e = (sn_i^e, \hat{C}_i^e, \hat{\sigma}_i^e)$.
- Step 3: EV then generates a random secret z_i^e and commits its identity e in $m_i^e = Cmt(e; z_i^e)$.
- Step 4: EV constructs a non-interactive zero-knowledge proof that (i) it knows the openings to C_i^e and $C_j^{p,u}$; and (ii) $d_{i,j}^{e,p,u}$ is correctly formed; and (iii) the single-use credential τ_i^e is correctly spent. The proof-of-knowledge equation is illustrated in eqn. (2).

$$\begin{aligned}\pi_{i,j}^{e,p,u} = NIZK\{(P_e, P_p, R_e, R_p, \hat{R}, e, L) : \\ C_i^e = Cmt(P_e; R_e) \wedge \\ C_j^{p,u} = Cmt(P_p; R_p) \wedge \\ \hat{C}_i^e = Cmt(L, e; \hat{R}) \wedge \\ d_{i,j}^{e,p,u} = H(n, C_i^e, C_j^{p,u}) \cdot e + L\}\end{aligned} \tag{2}$$

- Step 5: EV sends the $m_i^e, \tau_i^e, d_{i,j}^{e,p,u}, \pi_i^e$ to the PO.

3) Receipt Generation: In this step, the PO obtains two receipts by running two instances of ACL signature generation algorithm with the utility.

- Step 1: PO verifies the validity of τ_i^e and π_i^e, and sends to the utility $C_i^e, C_j^{p,u}, \tau_i^e, d_{i,j}^{e,p,u}, \pi_{i,j}^{e,p,u}$.
- Step 2: utility verifies that the credential τ_i^e is valid, and $\pi_{i,j}^{e,p,u}$ is correct.
- Step 3: utility runs the ACL signing algorithm as the signer with the PO on (m_i^e, C_i^e) and $(m_j^{p,u}, C_j^{p,u})$ respectively. In the end the PO obtains receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, $\tilde{R}_i^e$, receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$, and $\tilde{R}_j^{p,u}$, where $\tilde{C}_i^e = Cmt(P_i^e; \tilde{R}_i^e)$ is a commitment to the same attributes P_i^e as C_i^e, but with a different secret $\tilde{R}_i^e$, and σ_i^e is the utility's ACL signature on $(m_i^e, \tilde{C}_i^e)$. Similarly, $\tilde{C}_j^{p,u} = Cmt(P_j^{p,u}; \tilde{R}_j^{p,u})$ is a commitment to the same attributes as $C_j^{p,u}$, and $\sigma_j^{p,u}$ is the utility's ACL signature on $(m_j^{p,u}, \tilde{C}_j^{p,u})$.
- Step 4: PO verifies that $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ are formed correctly, and the commitment openers $\tilde{R}_i^e$, $\tilde{R}_j^{p,u}$ are valid.
- Step 5: PO stores its own receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ together with $\tilde{R}_j^{p,u}$
- Step 6: PO sends the other receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and the corresponding commitment opener $\tilde{R}_i^e$ to the EV.

Note that during the signing process the utility does not learn the value of $m_i^e, \tilde{C}_i^e, m_j^{p,u}, \tilde{C}_j^{p,u}$.

4) EV Validation: In the validation step the EV receives $(m_i^e, \tilde{C}_i^e, \sigma_i^e), \tilde{R}_i^e$ from the PO, and verifies that σ_i^e is indeed the utility's signature on $(m_i^e, \tilde{C}_i^e)$, and that $\tilde{R}_i^e$ opens the commitment $\tilde{C}_i^e = Cmt(m_i^e; \tilde{R}_i^e)$.

D. Reconciliation

The reconciliation phase happens at the end of the billing cycle. Each EV e computes the total sum that it should pay to its utility u, and each PO p also computes the total sum that it should receive each utility u.

Let us consider that in the current billing cycle EV e has participated in a total of M^e dynamic charging sessions (with any PO). For the i^{th} charging session EV e has the price P_i^e, the price receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, the secret $\tilde{R}_i^e$ that opens $\tilde{C}_i^e$, and z_i^e that opens m_i^e, for all $1 \leq i \leq M^e$. EV e computes the total price

$$P^e = \sum_{i=1}^{M^e} P_i^e \tag{3}$$

and the commitment opener for the homomorphic commitment

$$R^e = \sum_{i=1}^{M^e} \tilde{R}_i^e \tag{4}$$

EV e then sends P^e, R^e together with the receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and z_i^e for all $1 \leq i \leq M^e$ to utility u. The utility checks that

- $Cmt(P^e; R^e) = \Pi_{i=1}^{M^e} \tilde{C}_i^e$
- $\forall i, m_i^e = Cmt(e; z_i^e)$
- σ_i^e is a valid ACL signature on $(m_i^e, \tilde{C}_i^e)$.

As a second step, EV e proves to the utility that it does not omit any payment by revealing the remaining $N - M^e$ unused

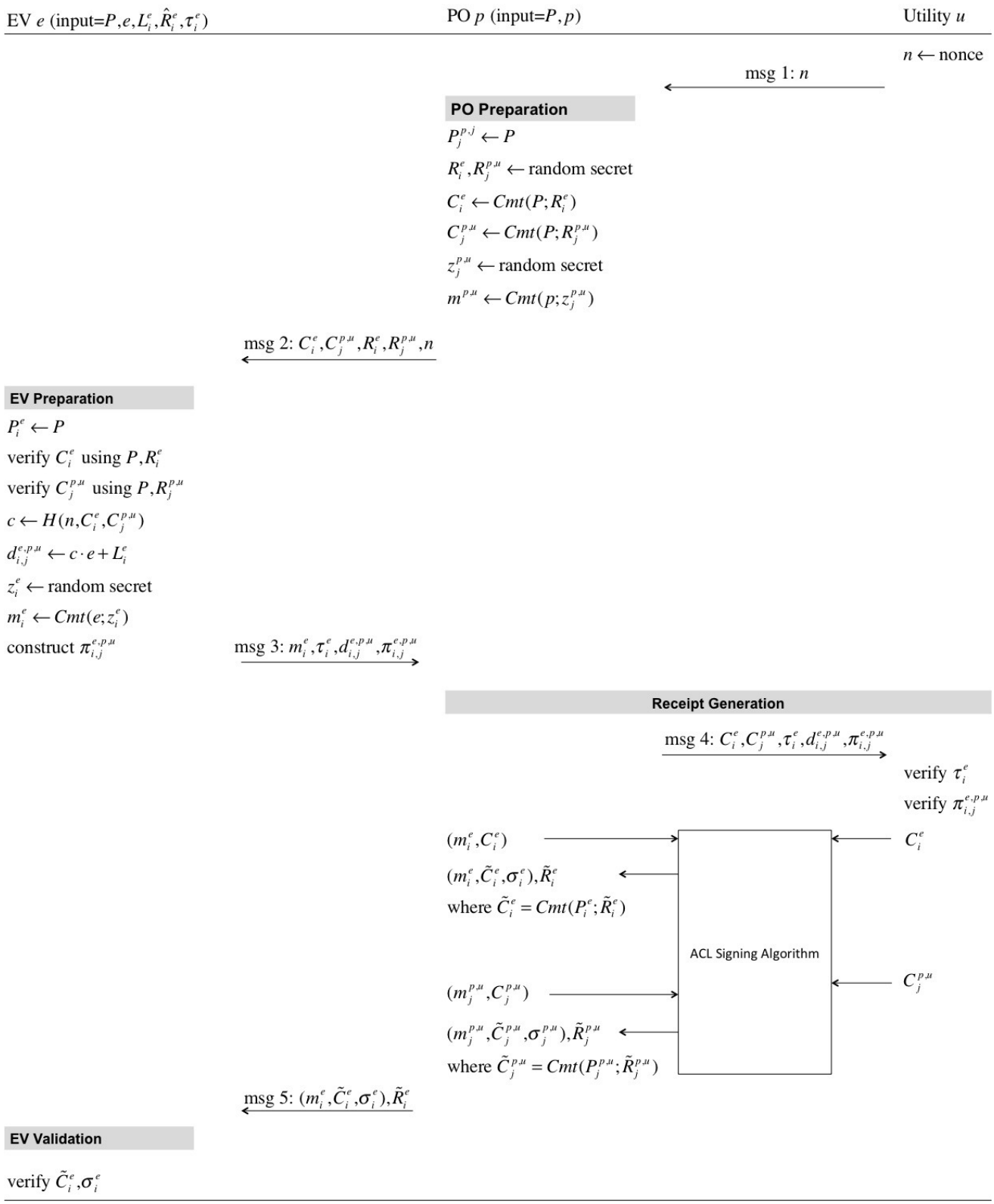

Fig. 4: Overview of the Price Validation phase of Janus. The figure shows the message exchange for the i-th charging session of EV e, which happens at a charging section of PO p, and happens to be the j-th time that PO p provides dynamic charging service to any EV subscribing to utility u. As a result, EV e obtains receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and PO p obtains receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ on the price $P_i^e = P_j^{p,u} = P$ from the utility u.

credentials $\tau^e_{M^e+1}, \ldots, \tau^e_N$. If all the above verifications succeed, the utility accepts P^e as the correct total sum that EV e owes the utility for the billing cycle.

The reconciliation phase for the PO is analogous. Consider that in the current billing cycle PO p has engaged in a total of $M^{p,u}$ dynamic charging sessions with any EV subscribed to utility u, and the fee of the j^{th} session was $P^{p,u}_j$. PO p constructs $P^{p,u} = \sum_{j=1}^{M^{p,u}} P^{p,u}_j$ and $R^{p,u} = \sum_{j=1}^{M^{p,u}} \tilde{R}^{p,u}_j$, and sends $P^{p,u}, R^{p,u}$ together with $(m^{p,u}_j, \tilde{C}^{p,u}_j, \sigma^{p,u}_j), z^{p,u}_j$ for all $1 \leq j \leq M^{p,u}$ to utility u. The utility checks that

- $Cmt(P^{p,u}; R^{p,u}) = \Pi_{j=1}^{M^{p,u}} \tilde{C}^{p,u}_j$
- $\forall i, m^{p,u}_j = Cmt(p; z^{p,u}_j)$
- $\sigma^{p,u}_j$ is a valid ACL signature on $(m^{p,u}_j, \tilde{C}^{p,u}_j)$.

If all verifications succeed, the utility accepts the value $P^{p,u}$ as the total sum it owes PO p. Since in the considered billing model the utility should pay the PO, the PO has no economic incentive to omit a price value in computing the total sum $P^{p,u}$. Therefore, the protocol does not require the PO to prove to the utility that no price value $p^{p,u}_j$ is omitted.

Note that, although the receipts are both generated and verified by the utility, the utility cannot link the EV's receipt $(m^e_i, \tilde{C}^e_i, \sigma^e_i)$ or the PO's receipt $(m^{p,u}_j, \tilde{C}^{p,u}_j, \sigma^{p,u}_j)$ to the price validation session (e, p, u, i, j) where the receipts were constructed. This is because the receipts are constructed using blind signatures, the verification of which will not reveal the signing session nor the identity of the signature requester (See section III-C).

VI. Evaluation

A. Implementation

We have implemented Janus in Python using the petlib library [1], which includes an implementation of the ACL signature scheme. The implementation uses Pederson commitment as the homomorphic commitment scheme and P-224 elliptic curve group. Since the implementation is primarily meant to evaluate the execution time rather than the communication delay, we implemented the protocols as a single process and simplified message passing between different entities as function calls.

B. Execution Time

The Janus protocol consists of 3 phases, but the registration and the reconciliation phases are only executed once per billing cycle, at the beginning and at the end, and rely only on ACL signatures and homomorphic commitments. The most complex part of the Janus protocol is the price validation phase shown in Figure 4, which is executed at the beginning of each dynamic charging session, and thus we focus on the execution time of this phase. For the evaluation we consider a reasonable scenario in which the PO and the utility have more computational power than the EV, and thus we run the EV preparation and the EV validation steps on a Raspberry Pi 2 Model B, which has a 900MHz quad-core ARM Cortex-A7 CPU and 1GB RAM, and run the PO preparation and the receipt generation steps on a macbook pro with a 2.7GHz Intel Core i5 and 8 GB RAM. We repeat the execution for 20 times, and we show the average execution time of the various steps in Figure 5.

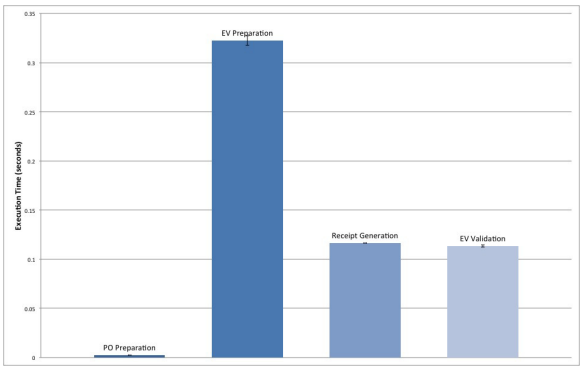

Fig. 5: Execution time of the price validation protocol. EV preparation and EV validation are run on a Raspberry Pi 2 Model B. PO preparation and signature generation are run on a Macbook. Error bars indicate 95% confidence intervals.

Clearly, the most time-consuming operations are included in the EV preparation step, where the EV constructs the non-interactive zero-knowledge proof $\pi^{e,p,u}_{i,j}$. The signature generation step, although involving more cryptographic operations, takes less time to execute since the utility and the PO have more computational power than the EV. Overall, the price validation protocol takes less than 0.6 seconds to execute, which makes it practical for the subscription-based billing model, even if communication delays are considered.

C. Communication Overhead

Recall that Janus consists of three phases: registration, price validation, and reconciliation. Both the registration and the reconciliation phase happen only once per billing cycle, and the delay constraint on these two phases are very loose. We therefore are more interested in the communication overhead of the price validation phase, which happens once per charging session.

The price validation phase involves two instances of ACL signature generation, each of which involves 5 message exchanges [5]. Nonetheless, the ACL signature generation algorithm is run by the PO and the utility, hence the time needed for 5 message exchanges is not significant. We thus treat ACL signature generation as a blackbox, and refer to [5] for its analysis.

Besides the message exchanges used in the ACL algorithm, the price validation phase requires only 5 messages. In Table I, we show the sizes of the messages exchanged in Figure 4[2]. The numbers correspond to the sizes of the actual python objects in our implementation. The two largest messages are msg 3 and msg 4, which include the non-interactive zero-knowledge proof $\pi^{e,p,u}_{i,j}$ and the EV's credential τ^e_i. Recall that the price validation phase during which these message exchanges occur happens once per charging session, their transmission over any reasonable wireless communication system would incur a small transmission time. These experimental results show

[2]We omitted the message exchanges in the standard ACL signature signing process.

that the computational and communication complexity of Janus make it practical for dynamic EV charging.

Message	Size (bytes)
msg 1	28
msg 2	180
msg 3	1117
msg 4	1165
msg 5	833

TABLE I: Message sizes

D. Scalability

In Janus, the EV obtains all its single-use anonymous credentials τ_i^e from the utility during the registration phase. Issuing one anonymous credential requires running the signing algorithm of ACL signature, which is time consuming (e.g., taking around 1 sec to complete). However, this does not affect the scalability of the protocol because the registration phase happens once at the beginning of the billing cycle, where the EV has several hours or even days to obtain all the anonymous credentials it needs. Similarly, the reconciliation phase also happens once every billing cycle, and does not affect the real-time scalability of Janus.

VII. Analysis

In this section we prove the correctness of Janus and analyze the location privacy it provides. Throughout the section we focus on EVs that contracted a particular utility u, and we denote by $\mathcal{E}$ the set of EVs that contracted the considered utility and by $\mathcal{P}$ the set of POs.

A. Security and Privacy Analysis

In this section we review the three phases of Janus, and analyze the use of security building blocks as well as discuss what information can be learned in each phase.

During the registration phase, the utility issues anonymous credentials to EVs. The EV must authenticate itself using its true identity with the utility at the beginning of the registration phase, e.g., using its long-term public key. This guarantees that the Utility will issue the correct number of credentials to each EV. Although the EV authenticates with the utility using its true identity during this phase (e.g., using a long-term public key), the ACL signing algorithm guarantees that the anonymous credential τ_i^e issued to EV e cannot be linked to the signing session. Therefore, the only information that the utility learns about EV e during the registration phase is that the EV has obtained a total of N credentials. Since each EV obtains the same number of credentials during the registration phase, the number N will not provide any useful information to the utility for inferring the EV's identity. Finally, credentials cannot be forged due to the construction based on the ACL signature.

During the price validation phase, the utility collects the price commitments $C_i^e, C_j^{p,u}$, and issues EV's receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ and the PO's receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ for session (e, p, u, i, j). The price $P_i^e = P_j^{p,u}$ committed is never revealed to the utility. Note that the utility learns neither the messages m_i^e, $m_j^{p,u}$, nor the commitments $\tilde{C}_i^e$, $\tilde{C}_j^{p,u}$ in the receipts at this time, due to the particular construction of the ACL blind signature (See Section III-C). During this phase, EV e also spends one single-use anonymous credentials τ_i^e. The utility is only able to verify that the credential is valid and has not been used before, but cannot infer the EV's identity from the anonymous credential. Finally, note that during this phase EV e does not directly communicate with the utility, and therefore network-level anonymity (e.g., Tor) is not necessary. Note that only EV e knows the secret value z_i^e that opens the commitment $m_i^e = Cmt(e; z_i^e)$. Since the true identity e of the EV is committed in m_i^e, and since σ_i^e is a signature on m_i^e with attributes committed in $\tilde{C}_i^e$, the receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ can only be used by EV e and is not useful to an attacker that steals the receipt. Similarly, the receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ can only be used by PO p. The signatures σ_i^e and $\sigma_j^{p,u}$ used in the receipts cannot be forged by a third party due to the construction of ACL.

During the reconciliation phase, the utility learns the total fee P^e and the total number of charging sessions M^e of each contracted EV e, as well as the total fee $P^{p,u}$ and the total number of charging sessions $M^{p,u}$ about each PO p. It also collects the corresponding receipts from each EV e and each PO p. Janus does not reveal to the utility the fee for each individual dynamic charging session, neither the charging sections that an EV has charged its battery at.

B. Correctness

To prove correctness, we have to show that the utility is able to verify that for each EV e the total charging fee P^e submitted by the EV is indeed the sum of the charging fees of all charging sessions that EV e participated in, i.e., $P^e = \sum_{i=1}^{M^e} P_i^e$. Similarly, we have to show that utility u is able to verify that the total charging fee $P^{p,u}$ claimed by PO p is not more than the sum of the charging fees of all charging sessions provided by the PO to EVs with a contract with u, i.e., $P^{p,u} \leq \sum_{j=1}^{M^{p,u}} P_j^{p,u}$.

1) EV Cannot Omit Payment: In Janus, the utility does not compute the total fee owed by the EV, it only verifies that the total fee is consistent with the receipts, both of which are submitted by the EV. Naturally a malicious EV may attempt to underclaim the total fee by intentionally withholding submitting one or multiple receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$. Janus mitigates this by requiring each EV to spend *all* of its single-use anonymous credentials τ_i^e. Recall that during the registration phase which happens once at the beginning of each billing cycle, the utility issues a total of N single-use credentials $\tau_1^e, \ldots, \tau_N^e$ to each EV e. In the price validation phase, in order to receive a receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$, EV e must spend one of its unused credentials. Therefore, if during the reconciliation phase the EV submits a total of M receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e), 1 \leq i \leq M^e$, the EV must prove to the utility that it still possesses a total of $N - M^e$ unused credentials. The EV can prove this to the utility by repeating a simple challenge-response protocol for $N - M^e$ times: each time the EV receives a fresh challenge c from the utility it has to spend an unused credential by binding the credential to the value of c, in a way similar to how the EV binds the price commitments C_i^e and $C_j^{p,u}$ in the double-spending equation $d_{i,j}^{e,p,u} = H(n, C_i^e, C_j^{p,u}) \cdot e + L$ as we described in Section V-C. If the EV fails to prove possession

of $N - M^e$ unspent credentials, the utility knows that the EV is trying to omit payments, and can thus levy a fine on the EV. How high of a fine the utility levies is outside of the scope of the protocol.

2) EV and PO Can Only Claim Correct Totals: Before we prove correctness of Janus in the sense formulated at the beginning of the section, we establish an important relationship between receipts obtained by an EV and a PO upon charging. For convenience, let us define the function Fee that extracts the charging fee from the commitment in a receipt, i.e.,

$$Fee((m, \tilde{C}, \sigma)) = P \text{ where } \tilde{C} = Cmt(P; \tilde{R}). \quad (5)$$

Proposition VII.1 *Consider the charging fees P_i^e committed in $\tilde{C}_i^e$, and the charging fees $P_j^{p,u}$ committed in $\tilde{C}_j^{p,u}$. There exists a bijective mapping f between the set $\{(m_i^e, \tilde{C}_i^e, \sigma_i^e) : \forall e \in \mathcal{E}, 1 \leq i \leq M^e\}$ and the set $\{(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u}) : \forall p \in \mathcal{P}, 1 \leq j \leq M^{p,u}\}$ such that $Fee((m_i^e, \tilde{C}_i^e, \sigma_i^e)) = Fee(f((m_i^e, \tilde{C}_i^e, \sigma_i^e)))$.*

Proof: Consider an arbitrary receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ of an arbitrary EV e. Without loss of generality, assume EV e obtained this receipt in session (e, p, u, i, j). Observe that the price validation phase for charging session (e, p, u, i, j) starts with $P_i^e = P_j^{p,u}$. EV e then binds its credential τ_i to the two commitments $C_i^e = Cmt(P_i^e; R_i^e)$ and $C_j^{p,u} = Cmt(P_j^{p,u}; R_j^{p,u})$. Since PO j also knows the commitment opener R_i^e and $R_j^{p,u}$, it can verify that the EV indeed binds its credential to the two commitments C_i^e and $C_j^{p,u}$ instead of to some other commitments of different values. Therefore, when the PO relays the EV's zero-knowledge proof π to the utility, it is guaranteed that both PO p and EV e agree on the price $P_i^e = P_j^{p,u}$ committed in C_i^e and $C_j^{p,u}$. Since the PO must relay the zero-knowledge proof π in order for the price validation protocol to complete, and since the EV and the PO can only obtain their receipts when the price validation completes, we are guaranteed that for the charging session (e, p, u, i, j) the price committed in $\tilde{C}_i^e$ of the EV's receipt $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ must be equal to the price committed in $\tilde{C}_j^{p,u}$ of the PO's receipt $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$. By defining $f((m_i^e, \tilde{C}_i^e, \sigma_i^e)) = (m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$ for each charging session (e, p, u, i, j), we obtain the desired bijective mapping. ■

We can use the existence of a bijective mapping between receipts and commitments to prove the correctness of Janus.

Proposition VII.2 *Janus is correct in the sense that it allows the utility to verify that $P^e = \sum_{i=1}^{M^e} P_i^e$ for each EV e and $P^{p,u} \leq \sum_{j=1}^{M^{p,u}} P_j^{p,u}$ for each PO $p \in \mathcal{P}$.*

Proof: We give an indirect proof for the correctness of Janus, which it achieves using homomorphic commitments that are signed by the utility. Consider an EV e that during the reconciliation phase submits a total of $M^{e'}$ receipts $(m_i^e, \tilde{C}_i^e, \sigma_i^e)$ where $1 \leq i \leq M^{e'}$. From the reconciliation phase it is clear that EV e can only claim a total fee that is equal to the sum of the charging fees committed in the commitments $\{\tilde{C}_i^e : 1 \leq i \leq M\}$. If the EV attempts to replace $\tilde{C}_i^e$ with another commitment $C' = Cmt(P'; R')$ with a different charging fee $P' \neq P_i^e$, it must forge the utility's signature σ' on (m_i^e, C'), which is infeasible given the security of the ACL signature [5]. Since $\tilde{C}_i^e$ is a commitment of P_i^e, this guarantees that the claimed total is $P^e = \sum_{i=1}^{M^{e'}} P_i^e$. Note that, so far there is no guarantee that the number of receipts $M^{e'}$ submitted by EV e is equal to the number of charging sessions M^e that the EV actually participated in. To guarantee that $M = M^e$, Janus requires the EV to reveal all unused credentials during the reconciliation phase. As discussed in Section VII-B1, this prevents EV e from intentionally omitting one or multiple individual charging fees in the calculation of the total fee. Given that $P^e = \sum_{i=1}^{M^{e'}} P_i^e$ and that $M^{e'} = M^e$, we have $P^e = \sum_{i=1}^{M^e} P_i^e$, which proves the first part.

The correctness of PO p's total fee follows a similar argument. PO p can only claim a total fee that is the sum of a subset of the charging fees committed in the commitments $\{\tilde{C}_j^{p,u} : 1 \leq j \leq M^{p,u}\}$. The PO cannot modify the value $P_j^{p,u}$ committed in $\tilde{C}_j^{p,u}$ without invalidating the signature $\sigma_j^{p,u}$ on the pair $(m_j^{p,u}, \tilde{C}_j^{p,u})$. The PO has to support its claimed total fee using a series of receipts $(m_j^{p,u}, \tilde{C}_j^{p,u}, \sigma_j^{p,u})$. Each receipt proves to the utility that some EV, whose true identity is unknown to the utility, has agreed on the fee that is committed in $\tilde{C}_j^{p,u}$. The utility can verify that the total fee claimed by the PO is indeed the sum of all of the fees committed in the homomorphic commitments $\tilde{C}_j^{p,u}$ that the PO submits, without learning the value of the individual fees themselves, which proves the second part. PO p can thus claim at most an amount of $P^{p,u} = \sum_{j=1}^{M^{p,u}} P_j^{p,u}$. ■

Note that Janus does not prevent a PO from underclaiming the total fee, e.g., by intentionally omitting one or more receipts in the calculation of the total fee. Nonetheless, since in our billing model the PO does not receive payment directly from the EV but from the utility that aggregates the dynamic charging activities of the EVs during the past billing cycle, underlaiming the charging fee would only cause financial damage to the PO. A rational PO would thus not attempt to underclaim the total charging fee.

C. Location Privacy

If a utility has a single EV as customer then the aggregate information is clearly enough for the utility to invade the location privacy of the EV, i.e., to infer the charging sections the EV visited. In what follows we are interested in whether the utility can infer the set of charging sections that a particular EV visited, and possibly the fee for each possible charging session of an EV, in more likely scenarios.

For the analysis recall that $\sum_{e \in \mathcal{E}} M^e = \sum_{p \in \mathcal{P}} M^{p,u} = M$, and let us consider a particular outcome of charging sessions of EVs $e \in \mathcal{E}$ at POs $p \in \mathcal{P}$. We can model this by a bipartite multigraph $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \mathcal{S})$, where $\mathcal{S}$ is the set of edges, which contains an edge (e, p) for every charging session of EV e at PO p. Observe that $\mathcal{G}$ has parallel edges if any EV had multiple charging sessions at the same PO.

Without a priori information about the charging fees, inferring the EVs' location can be formulated as finding the

(number of) bipartite multigraphs $(\mathcal{E}, \mathcal{P}, \mathcal{S})$ that satisfy

$$\sum_{e\in\mathcal{E}} \mathbf{1}_{\mathcal{S}}((e,p)) = M^{p,u} \quad \forall p \in \mathcal{P} \tag{6}$$

$$\sum_{p\in\mathcal{P}} \mathbf{1}_{\mathcal{S}}((e,p)) = M^{e} \quad \forall e \in \mathcal{E}, \tag{7}$$

and that allow a feasible vector of payments $(P_{(e,p)})$ to the problem

$$\sum_{p\in\mathcal{P}} P_{(e,p)} = P^{e}, \quad \forall e \in \mathcal{E} \tag{8}$$

$$\sum_{(e,p)\in\mathcal{S}} P_{(e,p)} = P^{p,u}. \tag{9}$$

Constraint (6) corresponds to the number of charging sessions of PO p, (7) to the number of charging sessions of EV e, (8) ensures that the total fee of each EV is allocated, and (8) enforces a feasible allocation of fees between EVs and POs.

In the worst case every bipartite multigraph that satisfies (6) and (7) allows a feasible vector of payments. The following result shows that if each PO provides charging sessions to sufficiently many EVs then the number of feasible bipartite graphs grows exponentially.

Proposition VII.3 *The number of bipartite graphs that satisfy (6)-(7) is lower bounded by*

$$\left(\frac{M}{|\mathcal{E}||\mathcal{P}|}\right)^{\Omega(|\mathcal{E}||\mathcal{P}|)}. \tag{10}$$

Proof: To obtain the number of bipartite multigraphs satisfying the above constraints, let us consider the biadjacency matrix of a bipartite multigraph, i.e., the non-negative integer valued matrix of size $|\mathcal{E}| \times |\mathcal{P}|$ whose entry (e,p) is the number of parallel edges between vertices e and p. Every bipartite multigraph satisfying (6) and (7) has a biadjacency matrix whose row sum for row e is M^e and column sum for column p is $M^{p,u}$. Finding the feasible bipartite multigraphs is thus equivalent to finding the non-negative integer valued matrices of size $|\mathcal{E}| \times |\mathcal{P}|$ with row sum sequence $(M^e)_{e\in\mathcal{E}}$ and column sum sequence $(M^{p,u})_{p\in\mathcal{P}}$. While it is known that a feasible matrix can be found in polynomial time, counting the number of feasible matrices is known to be #P-hard even for $|\mathcal{E}| = 2$ [11]. Furthermore, if the matrix is dense, i.e., $min_{e\in\mathcal{E}}\{M^e\} = \Omega(|\mathcal{P}|)$ and $min_{p\in\mathcal{P}}\{M^{p,u}\} = \Omega(|\mathcal{E}|)$, then the number of bipartite graphs can be lower bounded by [6]

$$(\frac{M}{|\mathcal{E}||\mathcal{P}|})^{\Omega(|\mathcal{E}||\mathcal{P}|)}. \tag{11}$$

■

The above result shows that the search space grows exponentially with both the number of EVs and the number of POs. For example, suppose there are a total of 10^4 EVs and 10 POs, and each EV participates in 2 charging sessions on average per day. Suppose the billing cycle is 30 days, then there will be a total of $M = 6 * 10^5$ charging sessions in the billing cycle. To consider all feasible bipartite graphs, i.e., which EV visited which PO for how many times, the utility needs to search in a space of $(\frac{6*10^5}{10^4*10})^{\Omega(10^4*10)} \sim 6^{10^5}$. Thus, without a priori information about the charging fees Janus provides a high level of location privacy when there are sufficiently many EVs and POs.

To evaluate the case when the utility does have a priori information we now consider the case when a PO charges the same amount for every charging session. To consider this case it suffices to introduce the additional constraint

$$P_{(e,p)} = P^{p,u}/M^{p,u} \quad \forall (e,p) \in \mathcal{S}, \tag{12}$$

i.e., the fee of a charging session at PO p is always $P^{p,u}/M^{p,u}$.

Proposition VII.4 *The problem of finding a bipartite graph and a feasible vector of payments that satisfy (6)-(9) and (12) is NP-hard.*

Proof: It is easy to see that the problem defined by (6)-(9) and (12) corresponds to the sized multiple subset sum problem, which is a generalization of the sized subset sum problem [10]. The sized subset sum problem consists of a list of positive integers (the charging fees of the POs, one integer per charging session), a positive integer P^e, and a positive integer parameter M^e. The objective is to decide whethere there is a sublist of size M^e of the integers that sums to P^e. The sized subset sum problem is W[1]-hard [10], i.e., its complexity increases exponentially in the parameter M^e, and is exactly the problem of assigning charging sessions that satisfy (8) to an EV e. Since the problem has to be solved for all EVs simultaneously, the problem (6)-(9) and (12) is a generalization of the sized subset sum problem and is thus NP-hard. ■

To summarize, without a priori information it is the number of feasible bipartite graphs that makes privacy invasion infeasible, while with a priori information it is the computational complexity. An analysis of the complexity of identifying charging sessions under different priors is a topic on its own right, and is beyond the scope of this paper.

VIII. Discussion

To put Janus into a context, we continue with a discussion of topics related to the design of Janus.

A. Comparison with Electronic Toll Pricing

If we regard each dynamic charging section as a toll road, then electronic toll pricing protocols [4], [15], [18] can be used in dynamic charging. However, to the best of our knowledge, most electronic toll pricing protocols require random spot checks to combat the malicious behavior where the vehicle drives through the toll road without running the protocol, e.g., by switching off the vehicle's on-board communication device. Random spot check incur additional maintenance cost, e.g., deployment of patrol cars, and may also raise fairness issues in certain cases, e.g., short-term rental car service. Janus does not rely on random spot check at all to detect payment omission. One important difference between the scenario of dynamic charging and that of electronic toll pricing is that, in electronic toll pricing, even if the vehicle does not own the proper authorization and does not authenticate itself, it can still drive on the toll road segment (unless there is a physical gate enforcing proper payment before entry). To combat driving on toll roads without proper authentication and payment, plate-reading cameras could capture the plate number of violating

vehicles and the driver would be responsible for paying a fine. However, in the dynamic charging scenario, authentication between the EV and the charging pads must complete before the EV's battery can be charged. If the EV chooses to turn off its communication device and does not authenticate with the charging pads, but simply drives through the dynamic charging section, it is not a violation because the charging pads will simply not charge the EV's battery, and there is no loss for either the pad owners or the utility. Janus utilizes the fact that the EV must authenticate itself before battery charging, and effectively binds authentication with payment: the EV must first prove to the pad owner its authorization by spending the single-use anonymous credential obtained from the utility. The anonymous credentials serve as both an authentication token and a binding of the dynamic charging fee to the payment token.

B. Comparison with Direct Billing Model using Digital Cash

An alternative billing model for dynamic charging might be for the EV to directly pay the PO using digital cash for each dynamic charging session. Anonymous digital cash such as Zerocoin [16] or Zerocash [20] can be used to implement the financial transaction between the EV and the PO. One might argue that, given the possibility of digital cash, it is not necessary to have billing protocol for the subscription-based billing model.

One drawback of using digital cash under the direct billing model is that the EV must pre-load funds into its account and make sure that the account has sufficient balance before entering the charging section, and the EV may thus run out of funds at inconvenient times. This problem does not exist in the subscription-based billing model that Janus is designed for, where the EV does not need to pre-load funds, and makes the actual payment only at the end of the billing cycle.

Note that the subscription-based billing model and the direct billing model described above are not mutually exclusive. Just like a store may accept both credit card and cash payment, a PO may accept both direct payment using digital cash and indirect payment using Janus. It is thus important to clarify the distinct advantages provided by the subscription-based billing model. The subscription-based billing model, however, enables flexible pricing plans that are not provided by the direct billing model, e.g., the EV can purchase a plan of 1000 miles from the utility and use dynamic charging anywhere anytime to recharge its battery up to 1000 miles of total driving distance.

We also note that Janus is designed specifically for dynamic charging scenarios, whereas digital cash is designed for more general scenarios, and thus there are certain features that a digital cash scheme can provide but Janus cannot. For example, a digital coin can be transferred multiple times among different entities, whereas in Janus the EV can only spend a credential τ once, and can only spend it with the utility. Nonetheless, because of the generality that digital cash aims to provide, it generally involves more complex designs than Janus. Anonymous digital cash designs may require even more complex cryptographic operations or zero-knowledge proofs to guarantee anonymity during spending. This may result in longer execution time of digital cash operations. For example, the Zerocash [20] design involves a zero-knowledge succinct non-interactive argument of knowledge (zk-SNARKs) that takes more than 2 minutes to generate on a platform with Intel Core i7-2620M 2.7GHz CPU and 12GB RAM. In comparison, the zero-knowledge proof π used by Janus takes less than 0.4 seconds to generate on the portable Raspberry Pi 2 platform with significantly less computational power (900MHz CPU and 1GB RAM).

C. Trust relationship between EV, Pad Owner, and the Utility

We finally discuss our assumption concerning the trust relationship between the EV, the PO, and the utility. Our billing model is a subscription-based model, in that the EV subscribes to the utility and receives a monthly bill aggregating all its dynamic charging usage from the utility. This requires the EV to trust the utility to make the correct calculation. In Janus, since the EV knows the ground truth of the dynamic charging fee for each charging session, it can easily verify if the total price in the monthly bill is correct. The utility can verify that the EV does not omit any payment token, and each payment token is authorized by some PO. However, the trust relationship between the EV and the utility does not automatically imply that the EV should give up its location privacy to the utility. Janus protects the EV's location privacy by not allowing the utility to infer the fee of a particular individual charging session from the payment tokens submitted by the EV.

To a certain extent, Janus assumes more trust relationship between the EV and the PO. This is reflected in the design where the PO effectively acts as a proxy between the EV and the utility: the EV does not directly communicate with the utility during the price validation phase; instead, the PO relays the credentials of the EV to the utility and the payment token from the utility to the EV. We justify this design choice by the observation that the PO is able to physically observe the EV. By our definition, the PO operates the dynamic charging section, and if the EV drives over the dynamic charging section, the corresponding PO inevitably observes the EV. We thus argue that the billing protocol, which works within the cyber space, cannot prevent the PO from revealing the EV's location and identity to any other entity in the physical space, and the billing protocol must assume that the PO will not reveal the EV's location information. As a consequence, mechanisms are needed to discourage a PO from revealing the EV's identity and location to a third party, whether or not Janus is used. The design of such mechanisms is outside of the scope of the paper.

D. Charging Fee Negotiation

Janus assumes that the EV and the PO are able to negotiate charging fees for each dynamic charging session. The problem of charging fee negotiation is orthogonal to the problem space of Janus, whose major goal is to allow verifiable aggregation of per-session charging fee into total charging fee without compromising the EV's location privacy. The EV and the PO may negotiate charging fee for the incoming dynamic charging session according to various pricing policies, e.g., fixed-rate pricing, day-ahead pricing, real-time pricing, etc., and the exact policy and negotiation protocol used by the EV and the PO is not our concern in this paper.

Janus requires the EV and the PO to negotiate charging fee and complete the price validation phase *prior* to the charging.

Note that once the price validation phase completes, the PO already obtains the receipt that it can use to claim money from the utility. In particular, a malicious PO may complete price validation with the EV but refuses to charge the EV. One might suggest that the PO should first charge the EV's battery, and only after the dynamic charging finishes do they run the price validation phase. However, this alternative design choice would allow a malicious EV to freeride the dynamic charging service by simply not running the price validation phase after it has received electricity from the PO. The question of whether price validation should complete before or after the actual charging is thus related to the question whether the EV or the PO is more likely to behave maliciously. We made the design choice where the EV and the PO complete price validation before the actual charging for the following reasons: (i) this is consistent with our assumption that the EV fully trusts the PO; (ii) a malicious PO is more likely to be caught, since the dynamic charging section is a physical road segment that does not move, and if the PO is reported of behaving maliciously, e.g., not charging the EV's battery after price validation completes, the utility or some other authority could send their own EVs to collect evidence; and (iii) in the scenario where multiple POs co-exist in the area and compete with each other, a PO not honoring the negotiated charging amount is likely to lose customers.

Another advantage of having the EV complete price validation phase with the PO prior to charging is authentication. Recall that in the price validation phase, EV e must spend a single-use anonymous credential τ_i^e. The PO is able to verify that the credential is spent correctly, which in turn tells the PO that this EV is valid and indeed subscribes to the utility. If the EV fails to spend an unused anonymous credential, the PO considers the EV as unauthenticated and will simply not charge the EV's battery at all.

IX. Related Work

Several recent works proposed to improve the privacy of electronic toll pricing services using on modern cryptographic solutions [4], [15], [18]. A common feature shared by these designs is that the vehicle is required to periodically broadcast certain information that can be used later by the authority to calculate its bill. The challenge is to make sure that the information disclosed during the periodic broadcast and the bill calculation process do not violate the vehicle's location privacy. In VPriv [18] the vehicle periodically broadcasts tags whose commitment the vehicle has registered with the authority. To calculate the bill, the authority sends to the vehicle the prices associated with each tag that the authority has received from any vehicle, and the vehicle calculates the total price using the prices corresponding to its own tags. The authority and the vehicle then engage in a two-party protocol where the vehicle proves either of the following: i) the tags used in the calculation are valid; or ii) the total price is calculated correctly with respect to the tags. The two-party protocol can be executed multiple times to improve the authority's confidence that the vehicle's bill is calculated correctly. Unlike VPriv, PrETP [4] does not use two-party computations. Instead, the vehicle periodically broadcasts a homomorphic commitment of the price corresponding to the current road segment. During the bill calculation phase, the authority combines all the individual commitments of the vehicle to obtian a commitment of the total fee; the vehicle calculates the total fee and proves to the authority that it knows the opening to the commitment of the total fee. To guarantee that an EV would faithfully broadcast the information required by the protocol, both VPriv and PrETP rely on random spot checks, and if the vehicle is physically observed at certain time and location, the vehicle is responsible for providing the proof that the toll price corresponding to that time and location is included in the total fee correctly. Milo [15] improves PrETP by considering the possibility that multiple vehicles collude and share the location of random spot checks with each other, and uses blind identity-based encryption to guarantee that the vehicles would not be able to learn these random spot check locations. In Spectre [9], the vehicle is given certain amount of Chaum's e-cash [7] as tokens, and periodically broadcasts tokens while driving (each token can be spent only once). At the end of the billing cycle, the vehicle submits all unused tokens, and pays for the tokens spent on the road.

As explained in Section VIII-A, the common disadvantage of VPriv, PrETP, Milo, and Spectre is that they all rely on random spot checks to guarantee that the vehicle is following the protocol faithfully. The random spot check incurs additional cost for the authority, and to some degree violates the vehicle's location privacy as well. Janus differs from existing toll pricing protocols in that it does not require random spot checks to enforce proper payment. Instead, it requires the EV to spend one single-use anonymous credentials per-session, and the PO will only charge the EV's battery after the EV has spent the credential and completed the price validation phase.

A handful of other recent works apply modern cryptographic tools to provide privacy-preserving payment for other transportation scenarios. Au et al. [2] proposed a privacy-preserving payment scheme with revocation for static charging of electric vehicles based on the BBS+ signature [3]. The solution proposed in [2] assumes that the EV maintains a balance and pays directly to the charging station, whereas Janus is designed for the subscription-based billing model, where the EV only makes the actual payment, indirectly through the utility to the POs, at the end of the billing cycle. Kerschbaum et al. [14] proposed a privacy-preserving billing mechanism for public transportation. However, the application scenario in [14] assumes that the user must tap a special cash card at the gate before entering or exiting the transportation system (e.g., the user must tap a card to enter the turnstile at train station), and that the user will add value to the cash card at a designated machine from time to time. Such assumptions make it difficult to apply the solution directly to our case, where the EV does not stop to tap a card before entering or exiting a charging section.

X. Conclusion

In this paper we have presented Janus, a privacy-preserving billing protocol for dynamic charging of electric vehicles. By using blind signatures with attributes and homomorphic commitments, Janus allows the utility to verify that the total payment of the individual EVs and the total fee that the PO should receive are calculated correctly, but it does not allow the utility to learn the dynamic charging fees and locations of individual charging sessions. Our Python-based implementation indicates that the computational and communication complexity of Janus is low enough to make it a

practical solution for privacy-preserving billing for dynamic charging. Our analysis of location privacy based on aggregate information indicates that in practical scenarios it will be a combination of computational complexity and the cardinality of the set of candidate solutions that helps preserve location privacy. Nonetheless, it is open question how much information a utility could learn over several billing cycles using an intersection attack. This question is of general interest as it concerns all privacy-preserving protocols that allow a party to learn aggregates about a set of individuals.

REFERENCES

[1] "Petlib (https://github.com/gdanezis/petlib)." [Online]. Available: https://github.com/gdanezis/petlib

[2] M. H. Au, J. Liu, J. Fang, Z. Jiang, W. Susilo, and J. Zhou, "A New Payment System for Enhancing Location Privacy of Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 3–18, 2014.

[3] M. H. Au, W. Susilo, and Y. Mu, "Constant-Size Dynamic k-TAA," in *Proc. SCN*, 2006.

[4] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens, "PrETP: Privacy-preserving Electronic Toll Pricing," in *Proc. USENIX Security*, 2010.

[5] F. Baldimtsi and A. Lysyanskaya, "Anonymous Credentials Light," in *Proc. ACM CCS*, 2013.

[6] A. Barvinok, "Asymptotic estimates for the number of contingency tables, integer flows, and volumes of transportation polytopes," *Int. Math. Res. Notices*, 2009.

[7] D. Chaum, "Blind Signatures for Untraceable Payments," in *Proc. CRYPTO*, 1983.

[8] G. Covic and J. Boys, "Modern Trends in Inductive Power Transfer for Transportation Applications," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 1, no. 1, pp. 28–41, 2013.

[9] J. Day, Y. Huang, E. Knapp, and I. Goldberg, "SPEcTRe: Spot-checked Private Ecash Tolling at Roadside," in *Proc. ACM WPES*, 2011.

[10] R. G. Downey and M. R. Fellows, "Fixed-parameter tractability and completeness II: On completeness for W[1]," *Theor. Comput. Sci.*, vol. 141, no. 1-2, pp. 109–131, 1995.

[11] M. Dyer, R. Kannan, and J. Mount, "Sampling contingency tables," *Random Struct. Alg.*, vol. 10, pp. 487–506, 1997.

[12] A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," in *Proc. CRYPTO*, 1986.

[13] S. Jeong, Y. J. Jang, and D. Kum, "Economic Analysis of the Dynamic Charging Electric Vehicle," *IEEE Trans. Power Electr.*, vol. 30, no. 11, pp. 6368–6377, 2015.

[14] F. Kerschbaum, H. W. Lim, and I. Gudymenko, "Privacy-preserving billing for e-ticketing systems in public transportation," in *Proc. ACM WPES*, 2013.

[15] S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham, "The Phantom Tollbooth: Privacy-preserving Electronic Toll Collection in the Presence of Driver Collusion," in *Proc. USENIX Security*, 2011.

[16] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," in *Proc. IEEE S&P*, 2013.

[17] T. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," in *Proc. CRYPTO*, 1991.

[18] R. A. Popa, H. Balakrishnan, and A. J. Blumberg, "VPriv: Protecting Privacy in Location-Based Vehicular Services," in *Proc. USENIX Security*, 2009.

[19] A. Rupp, F. Baldimtsi, G. Hinterwälder, and C. Paar, "Cryptographic Theory Meets Practice: Efficient and Privacy-Preserving Payments for Public Transport," *ACM TISSEC*, vol. 17, no. 3, pp. 10:1–10:31, 2015.

[20] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized Anonymous Payments from Bitcoin," in *Proc. IEEE S&P*, 2014.