Robust Revocable Anonymous Authentication for Vehicle to Grid Communications

Vishnu Teja Kilari, *Student Member, IEEE,* Ruozhou Yu, *Member, IEEE,* Satyajayant Misra, *Member, IEEE,* and Guoliang Xue, *Fellow, IEEE*

***Abstract*—Electric vehicles can place a significant load on the power grid due to their unscheduled charging events. One way of improving power grid stability is to schedule electric vehicle charging in advance. Before a charging visit, the electric vehicle provides necessary information to request for charging at a charging station, which prepares and reserves the energy before the visit. However, the reported information can cause privacy leakage of the electric vehicle user. Anonymous information reporting can protect user privacy, but also enables attacks on the charging station by unauthorized users. An anonymous authentication system can address these issues, but cannot detect misbehaviors by authenticated users. One remedy to this is revocable anonymity-based authentication, which can revoke the anonymity of malicious users after their misbehaviors. However, we show that such a system is still vulnerable to application-level Denial of Service attacks, where a malicious user requests for large amounts of energy simultaneously from many charging stations, preventing these stations from serving other users. To address this, we improve upon an existing revocable anonymity-based authentication framework. We propose a permit-based mechanism, where each electric vehicle is only issued with one blind signature-based permit at a time. A request is valid only if it contains a valid and unused permit, which protects the system from the application-level Denial of Service attacks. Security analysis and experiments demonstrate that our framework, while ensuring user anonymity and being robust to the aforementioned attack, is also scalable and lightweight.**

***Index Terms*—Smart grid, V2G communications, anonymous authentication, revocable anonymity**

I. Introduction

Electric vehicles (EVs) are becoming popular due to their competitive pricing and low cost in the long term, as well as the high availability of charging stations. While more than $402,000$ plug-in vehicles were sold in the US from 2011–2015 [9], $150,000$ plug-in hybrid and all-electric vehicles were sold in the US in 2016, and around $40,000$ charging stations existed in the US in 2016 [10]. Tesla's Model 3 sedan, a prominent EV model, received $455,000$ orders by August 2017 [8]. Meanwhile, China, one of the major markets for EVs, mandated 10% of the conventional passenger vehicle market in 2019 and 12% in 2020 to consist of EVs, which would result in the production of more than two million EVs by 2020 [25].

Yet, one problem caused by the rapid increase of EVs is the excessive load of EV charging posed on the power grid, which may cause grid instability and inefficiency. The Smart Grid [13] can resolve this issue by employing proper scheduling and coordination of the charging services in advance. To schedule a charging service, the EV reports critical information, such as expected time of arrival and amount of charge needed, to the charging station. Such information is used by the charging station and the grid for load prediction, distribution network congestion avoidance, and price management. After receiving this information, the charging station replies with its decision of acceptance or rejection of service to the EV based on the availability of charge, and reserves the energy if the request is accepted.

The scheduling process involves the EV reporting private information, such as duration and location, to the possibly untrusted charging station. Other private information, such as travel patterns of the EV as well as its occupants and identity information, can also be inferred from the reported information. For example, a service provider managing a network of charging stations can correlate this information to track an EV and its occupants, constituting a major privacy breach. To ensure the privacy of the EV and its users, the information reported by the EV must be anonymized.

On the other hand, a system based solely on anonymous information is susceptible to impersonation and active attacks from external attackers. For this reason, communications between EVs and charging stations must also be authenticated. Both goals can be achieved using anonymous authentication. For example, pseudonym based approaches have been proposed to provide anonymity [14], [20]. While these approaches allow at least one entity to know the EV's identity, a completely anonymous authentication framework is proposed in [21] for real-time EV reporting.

A significant drawback of a completely anonymous authentication system is that it is vulnerable to insider attacks from authenticated EVs. When an EV sends a charging request to a charging station, the station will reserve the charge from the grid and wait for the EV to arrive. A malicious EV can utilize this feature, by continuously sending large-amount charging requests to the charging station, without showing up for the charge. This will prevent the

Kilari and Xue ({vkilari, xue}@asu.edu) are with Arizona State University, Tempe, AZ 85287. Yu (ryu5@ncsu.edu) is with North Carolina State University, Raleigh, NC 27606. Misra (misra@nmsu.edu) is with New Mexico State University, Las Cruces, NM 88003. This research was supported in part by NSF grants 1704092, 1717197, 1345232, 1719342, and 1757207, and Intel grant 34627535.

charging station from serving other EVs, affecting both the charging station and other EVs in the same area. Since the communications are completely anonymous, identity of the malicious EV is hidden from the charging station, and hence the station has no way of stopping future attacks from the same EV. This example elucidates just one of many possible scenarios in which a malicious EV can attack a charging station and remain undetected.

In our preliminary work [17], we proposed an authentication framework with revocable anonymity, which can revoke the anonymity of a user if it acts maliciously. In this framework, the charging station can submit its complaint to a set of Federated Trust Entities (FTEs) regarding a malicious EV. The EV is given a chance to prove its honesty. If it fails, the FTEs will act together to revoke the EV's anonymity. Revealing the identity of a malicious EV serves two main purposes. First, EVs will be discouraged from being malicious due to the fear of being detected and penalized. Second, the charging stations can employ network-wide blacklisting to protect themselves from future attacks by the same EV.

Unfortunately, this framework cannot fully eliminate attacks from malicious EVs. Specifically, we show that this framework is vulnerable to the *application-level Denial of Service* (DoS) attack. Consider a scenario in which a malicious EV simultaneously submits a large number of charging requests to all the charging stations in the same area. Since all the requests are anonymous, there is no limit to the requests each EV can place to any of the charging stations. A charging station considers these requests and make reservations for the possible charging event. A charging station holds a finite amount of charge and thus can only handle a finite number of requests. These malicious requests will saturate the charge budgets at these stations, forcing them to deny future requests from other EVs. This constitutes an application-level DoS attack.

With the framework in [17], although the identity of malicious EV(s) orchestrating such attacks is eventually revealed, this anonymity revocation takes some time. A charging station needs to wait for the malicious EV to show-up for charging at the appointed time; when the EV does not show-up, the station complains to the FTEs. The time between the charging request and the scheduled charging event is the attack window. The malicious EV is free to place any number of charging requests to any number of charging stations in this attack window, thus blocking resources, i.e., launching an application-level Denial of Service (DoS) attack. Many EVs can also coordinate their reservations to result in an application-level Distributed Denial of Service (DDoS) attack.

Such application-level DoS attacks can impact charging performance, causing delays for EV passengers, and can result in eventual erosion of public trust in safety and viability of EVs. In this paper, we improve our previous framework to defend against this new type of attack. The protocols in our original framework were carefully constructed to provide revocable anonymity. Our aim is to improve our original framework to provide defense against these application-level DoS/DDoS attacks without sacrificing the existing security and privacy guarantees. To achieve this goal, we propose a novel throttling-based framework that limits the ability of an EV to launch such an attack.

Our improved framework is based on distributing a permit to each EV. A permit is a blind token issued by a third party authority. When requesting for charging, an EV submits its permit to the charging station. If the charging station accepts the request, it retains the permit until the EV visits and completes the transaction. Once the transaction is completed, the charging station provides a receipt to the EV, which can be used by the EV to acquire another permit. The permits are issued and used with complete anonymity, thus they do not violate the anonymity of the EV.

The revocable anonymity feature of the framework does not rely on the permit, and hence is not affected either. Moreover, an EV must submit an initial deposit to get its first permit, which should be high enough to economically prevent permit accumulation attacks. We note that the scheduling process requires real-time communications between the EV and the charging station, hence the proposed framework must be both lightweight and scalable. Our framework not only provides revocable anonymity to EVs and robustness against application-level DoS attacks, but is also scalable and efficient in terms of computation and communication overheads as demonstrated in our experiments and analysis.

The **main contributions** of our paper are:

- To the best of our knowledge, we are the first to uncover and address the application-level DoS attack for anonymously authenticated EV charging scheduling systems.
- We propose a novel permit-based mechanism, which ensures the anonymity of EV users if they behave honestly, enables anonymity revocation in the event of malicious behaviors, and additionally defends against the application-level DoS attacks on the charging infrastructure.
- We perform detailed security and privacy analysis of our framework, and evaluate its scalability and efficiency through analysis and implementations.

The rest of this paper is organized as follows. In Section II, we present our system and threat models, and an overview of our framework. In Section III, we explain the cryptographic concepts needed to understand our framework and its protocols. In Section IV, we describe our framework in detail. In Section V, we present the privacy and security analysis of our framework. In Section VI, we summarize the experiments, analysis, and results. In Section VII, we discuss related work. In Section VIII, we conclude this paper.

II. Models and Overview

A. System Model

We consider a system that consists of EVs, charging stations, and three Federated Trust Entities (FTEs): a Financial Authority (FA), a Certificate Authority (CA), and a Department of Motor Vehicles (DMV). We assume that each of these parties has its own unique public/private key pair. The charging stations could belong to a few well-known (centralized) entities, e.g., Tesla has its network of stations. In the real-world, Symantec, DigiCert, and Verisign are examples of entities that can act as the CA. A separate entity, such as a bank or a financial institution, can act as the FA. The DMV is a government agency that administers vehicle registration and regulations and we assume it has no covert financial interest. Such an agency exists in almost every country.

For ease of illustration, we focus on the interactions among one EV, one charging station, and the FTEs. We illustrate our protocols in the context of a specific reservation involving one pseudonym only. We note that an EV may obtain and possess multiple valid pseudonyms concurrently in our framework.

All communications are carried over anonymous channels such as Tor [12], which prevents traffic analysis attacks. Each EV can carry out offline computation for computationally intensive tasks during its idle time. A typical EV is assumed to have an idle time period before it needs to be charged, *e.g.*, during the night time before charging. We assume that the charging stations and the FTEs are always online, and communication channels exist between all parties. If a charging station accepts a request from an EV, we assume that the station has reserved enough energy for the EV's charging request.

B. Threat Model

The FA is a new FTE compared to the framework in [17]. The FTEs are assumed to be honest-but-curious: they follow all protocols honestly, but may try to infer sensitive information regarding the EVs. First, we assume that none of the FTEs (the CA, the FA, and the DMV) collude with the charging stations. This is a conservative assumption because there is a good chance that the malicious FTE is caught in the act eventually. Certificate Authorities and Financial Authorities depend on public trust in them to function. The risk of public exposure and reputation loss coupled with financial costs (from fines levied by legal or government oversight agencies) act as a strong deterrent against such collusions.

Second, we assume that the CA, the FA, and the DMV do not collude with each other (for the same trust and financial reasons). In the real world, the DMV is a government entity entrusted in preserving users interest and it gains nothing from colluding with the CA, FA, or the charging stations. We also assume that two or more of the FTEs do not become compromised at any time. A charging station is assumed to be honest-but-curious, i.e., it can benefit from identifying an EV and/or its patterns.

All the attackers (both internal and external) are assumed to be computationally bounded. Some of the EVs and/or some of the charging stations can be compromised and their secrets can be exposed. We consider the following threats against the framework: privacy violation, DoS/DDoS attacks, forgery attacks, collusion attacks, Man-in-the-Middle (MITM) attacks, and replay attacks. We explain the first four threats in the following, while the last two are common attacks.

Privacy violation. The CA, the FA, the DMV, and the charging stations may try to violate the privacy of uncompromised EVs by trying to link multiple charging requests to a specific EV, which in turn will give them access to private data such as the EV's traffic patterns (without violating the EV's identity).

DoS/DDoS attacks. Malicious EVs either by themselves or in collusion with other EVs can launch DoS and/or DDoS attacks on the charging station(s) by issuing either repeated requests to a single station, or a large number of requests to multiple stations in a specific area.

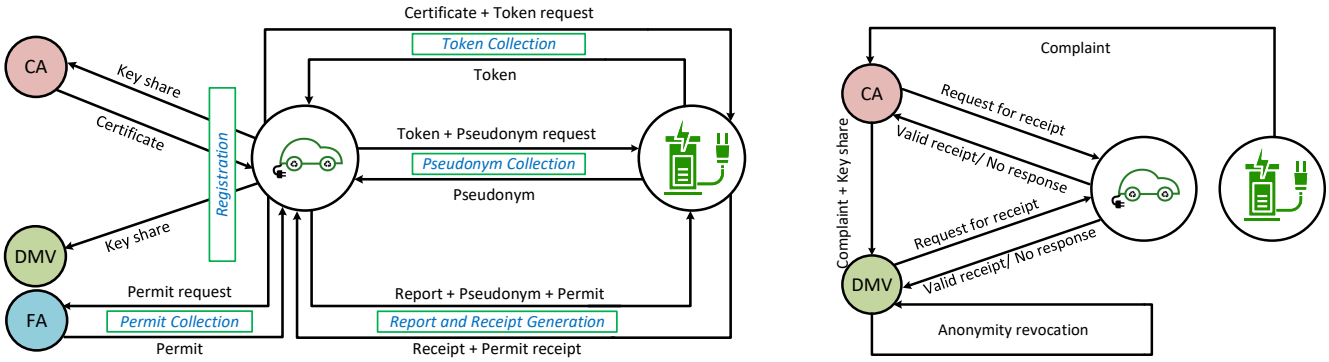
Forgery attacks. Forgery attacks involve acquisition of tokens, pseudonyms, and receipts without providing prerequisite inputs and following the protocols. An adversary can try to trick a charging station to issue tokens, pseudonyms, and receipts without the charging station identifying it properly.

Collusion attacks. We assume arbitrary collusion among EVs and charging stations. For example, an EV may collude with a charging station to try to violate the privacy of another EV. As aforementioned, we assume that FTEs do not participate in collusion due to the significant legal and economic consequences involved.

C. Intuition and Framework Overview

Fig. 1 provides an illustration of our framework. Our framework consists of a registration protocol and four essential everyday operational protocols as shown in Fig. 1(a): permit collection, token collection, pseudonym collection, and report and receipt generation. It also has an optional protocol for revoking a malicious EV's anonymity as shown in Fig. 1(b). Here, we explain the intuition behind our protocol design and then give a brief overview of each protocol.

Registration. The registration protocol enables the EV to get a certificate to prove its identity with the private information used in certificate generation split between multiple FTEs to preserve the EV's identity privacy, while at the same time enabling anonymity revocation of an EV that turns malicious. Although several FTEs can be involved, in our illustration, we assume that the DMV and the CA work together and the protocol enables an EV to collect the certificate from the CA, in the process securely registering with both the DMV and the CA. The EV uses a secret sharing scheme [29] to divide the key used to encrypt its



(a) Anonymous registration and service request protocols when the EV is benign.

(b) Anonymity revocation of a malicious EV.

Fig. 1. Overview of our framework: (a) five essential everyday protocols and (b) one optional protocol for anonymity revocation of malicious EVs.

identity into three shares, giving two of them along with the encrypted identity to the CA and the DMV. In return, the EV receives a certificate from the CA to use as its proof of identity. We assume that each EV is uncompromised at the time of registration. As per the design, the DMV and the CA working together (on proof of malicious behavior from a charging station) can revoke an EV's anonymity.

Pseudonym Collection. To interact with a charging station, an EV needs a unique unused credential. Since EV's privacy needs to be preserved, the credential is issued blindly by a charging station to the EV; this credential is a pseudonym. Before issuing this pseudonym, a charging station needs to verify that the requesting EV is legitimate. As the charging station cannot be trusted with the EV's identity, this verification is done using a token, blindly signed by the charging station itself (obtained in the token collection protocol described next). In the pseudonym collection protocol, the EV uses an unused token to collect a pseudonym and associated session keys from the station. These session keys are used to establish secure anonymous authenticated communications with the station. The tokens cannot be reused, hence multiple pseudonyms cannot be linked to an EV.

Token Collection. In the token collection protocol an EV proves that it is authentic and requests and collects a blindly-signed token (to ensure identity unlinkability). This blindly signed token and its components ensure the validity of EV, enable anonymity revocation of a malicious EV, and are used to obtain a pseudonym and establish session keys during the pseudonym collection protocol. A token request consists of the certificate of the EV, which is subsequently included in the token.

Permit Collection. Our framework allows each EV to collect multiple pseudonyms. To prevent application-level DoS attacks, an EV should not be able to use all the pseudonyms at once (technically an EV cannot charge at two places at the same time or charge at same place twice at the same time). To enforce this, in our framework an EV requires a permit to make a charging reservation. To preserve the privacy of an EV, this permit is blindly signed by an FA. A new permit is only issued by the FA after the EV makes a monetary deposit or on the receipt of the old permit by the FA. The request and issuance of the permit are done anonymously, hence the FA cannot link an EV to a permit. When the EV sends the permit to the station to reserve a slot to charge itself, the station interacts with the FA to ensure that the permit is not already in use before reserving a slot for the EV.

Report and Receipt Generation. In the report and receipt generation protocol, the EV establishes secure communications with the charging station using an unused pseudonym and its corresponding session keys. The EV uses a permit to make a charging reservation and provides required information to the charging station. In the event of a successful transaction between the charging station and the EV, the charging station issues a receipt to the EV. The EV can then use this receipt to apply for a new permit with another round of permit collection. The EV can also use this receipt to counter false complaints by a charging station.

Anonymity Revocation. If an EV is deemed malicious at some point in its life, the anonymity revocation mechanism illustrated in Fig. 1(b) is executed. A charging station complains about the malicious behavior of an EV to the CA and sends it the corresponding charging request. The CA verifies the certificate accompanying the pseudonym and decrypts the private key used to sign it. The CA places the certificate on the blacklist, and then sends to the DMV its share of the key for the encrypted ID associated with the certificate. The DMV uses its own share and the share from the CA to recover the key, which is used to decrypt the ID of the malicious EV and impose punishment on it. The CA and the DMV broadcast a request for the receipt of the charging request from the corresponding EV after receiving the complaint and only proceed to subsequent steps if the receipt is not produced in a timely manner.

Among the above, the registration protocol is performed when the vehicle is brought into service. The token and pseudonym collection protocols are assumed to be executed during the time when the EV is idle (*e.g.*, at night), since they are computationally intensive. The report and receipt generation protocol is executed in real-time. The permit collection protocol is executed either when the EV makes

its initial deposit, or after a random amount of time from the completion of the report and receipt generation protocol. We discuss the details of our framework in Section IV.

III. Cryptographic Concepts

We explain the cryptographic concepts needed to understand our framework and protocols in this section.

Definition 3.1 (Blind Signature [5]): A *blind signature scheme* allows one party (user) v to get a signature $Sig_E(m)$ on a message m from another party (signer) E without revealing any information about m to E. A blind signature scheme typically works as follows. User v blinds the message m with a blinding factor b, resulting in a blinded message $b(m)$. User v sends the blinded message $b(m)$ to signer E. Signer E signs the blinded message and sends the signed blinded message $Sig_E(b(m))$ to user v. User v unblinds the signed blinded message by applying the unblinding factor u, $u(Sig_E(b(m)))$. This unblinding reduces the signature of E on the blinded message to a signature of E on the original message m, $Sig_E(m)$. Since E does not know the message m, it cannot link its signature $Sig_E(b(m))$ on the blinded message to its signature $Sig_E(m)$ on the original message. □

Definition 3.2 (Partial Message Proof [18]): A *partial message proof* is used such that a signer E signs a message m blindly only if it can verify a part p_m of the message m prior to signing the message. This can be accomplished by utilizing a cut-and-choose protocol [26] or by selecting a blind signature scheme that incorporates zero-knowledge proofs on p_m. A cut-and-choose protocol works as follows. User v sends to E many blinded versions of the message m that must all contain a valid p_m. Signer E selects all but one of the messages which v has to unblind so that E can access them. Signer E signs the remaining blinded message $b(m)$ if all the unblinded messages contain a valid p_m. □

Definition 3.3 (Implicitly Authenticated Diffie-Hellman Key Exchange [11]): *Diffie-Hellman Key Exchange (DHKE)* is a method of establishing a shared secret key over an insecure channel between two parties with no prior knowledge of each other. However, DHKE does not provide authentication and thus is vulnerable to the MITM attack. *Implicitly Authenticated DHKE (IADHKE)* thwarts the MITM attacks by using digital signatures to authenticate the communicating parties. □

Definition 3.4 ((t,n)-threshold secret sharing scheme [29]): A *(t,n)-threshold secret sharing scheme* is a method of sharing a secret S among n participants in such a way that any group of t or more participants can collectively compute the secret, but any group of fewer than t participants cannot compute the secret. □

IV. Framework Description

In this section, we describe the details of the protocols in our framework. For illustration, we use v and $\mathcal{U}$ to denote the EV and the charging station, respectively. In Table I, we summarize the notations used in this paper. Messages in all the protocols have timestamps, which are omitted in the protocol description for ease of illustration.

TABLE I
List of Notations

$K(m)$	Message m encrypted with a key K
σ_E	Entity E's digital signature on the entire message
$Cert^v_{\text{CA}}$	Certificate issued by the CA to EV v at registration
p	A permit request from EV v to the FA
t	A token request from EV v to a charging station
ρ	A charging request from EV v to a charging station
ϕ^v	A permit issued by the FA to EV v
τ^v	A token issued to EV v by a charging station
δ^v	A pseudonym issued to EV v by a charging station
ID_v	Identity of EV v, *e.g.*, its VIN number
$c(m,z)$	Commitment of message m with opening secret as z
$b_e(m)$	Blinding factor e applied to message m
$u_e(m)$	Unblinding factor e applied to message m
$BS_E(m)$	Blind signature of entity E on message m
$ETM^{K^A}_{K^E}(m)$	Message m encrypted with key K^E, and then attached with a Message Authentication Code (MAC) computed on the encrypted message with key K^A. Both keys are derived from a session key K.
$PK_{\text{CA}}, SK_{\text{CA}}$	CA's public/private key pair
$PK_{\text{DMV}}, SK_{\text{DMV}}$	DMV's public/private key pair
$PK_{\text{FA}}, SK_{\text{FA}}$	FA's public/private key pair
PK_v, SK_v	Ev's public/private key pair

A. Registration

Protocol 1 explains the procedure followed by EV v to register with the DMV and the CA to obtain a certificate from the CA. We assume that the EV is uncompromised during registration and physically present at the DMV. To achieve revocable anonymity, the EV's identity ID_v is encrypted with a secret key K_v, where the key is shared among three entities using a secret sharing scheme. Specifically, K_v is divided into three shares K_v^{DMV}, K_v^{CA}, and K_v^v using a $(2,3)$-threshold secret sharing scheme. The $(2,3)$-threshold secret sharing scheme requires at least 2 out of the 3 shares to reconstruct the key K_v which can then be used to decrypt ID_v. Our framework uses Blakley's secret sharing scheme [3].

Using the IADHKE protocol, EV v establishes secure communication channels with the DMV and the CA using symmetric keys $K_{v\text{DMV}}$ and $K_{v\text{CA}}$, respectively. In Step 1, the EV sends to the DMV the share K_v^{DMV} of the key and the encrypted ID $K_v(ID_v)$, both encrypted with the key $K_{v\text{DMV}}$. In Step 2, the DMV creates a message, tkt, containing the encrypted ID $K_v(ID_v)$ and a random nonce n, signs tkt using its private key SK_{DMV}, and then encrypts the message and its signature using the public key PK_{CA} of the CA. It sends this signed encrypted message to the EV. In Step 3, the EV sends to the CA the share K_v^{CA} of the key, the encrypted ID $K_v(ID_v)$, a certificate request ($Cert$), and the signed encrypted message given by the DMV, all of them encrypted with the key $K_{v\text{CA}}$. The certificate request, $Cert$,

contains the public key of the EV, PK_v. In Step 4, the CA decrypts the message of the DMV using its private key SK_{CA} and verifies the signature of the DMV using DMV's public key PK_{DMV}.

If the signature is valid and the encrypted identity in the message tkt matches the encrypted identity sent by the EV, the CA signs the decrypted message tkt with its private key SK_{CA}, encrypts it with the public key PK_{DMV} of the DMV, and sends this signed encrypted message to the DMV. The DMV decrypts the message with its private key SK_{DMV}, verifies the signature of the CA using its public key PK_{CA}, matches the message it sent to the EV with the message it received from the CA, and notifies the CA that the EV is currently executing the registration protocol if the messages match and the EV is physically present at the DMV. If the CA receives the notification from the DMV that the EV with the encrypted identity $K_v(ID_v)$ is currently executing the registration protocol, the CA signs the certificate request and sends this certificate $Cert^v_{\text{CA}}$ to v.

The EV retains the third share of the key, K^v_v. This way, unless both the DMV and the CA are in agreement regarding the maliciousness of the EV, neither can obtain v's encryption key K_v and reveal its identity. The DMV and the CA both store their respective key shares along with the encrypted ID. The CA also stores the corresponding certificate.

To ensure that the EV is uncompromised during the registration protocol, the EV must be physically present at the DMV. In this case, the integrity of the software/firmware can be easily verified by the DMV. For example, the DMV can compute the checksum of the EV's software/firmware and compare it against the checksum provided by the vehicle's manufacturer. Recent advances in secure enclaves (e.g., Intel Software Guard Extensions (SGX) or ARM Trustzone) can be used to ensure that software/firmware of the vehicles can be attested inside the enclave in a vehicle to confirm that the process is not tampered. Through this process, the DMV can ensure that the EV follows the secret sharing scheme in the registration protocol, which is crucial for enabling anonymity revocation. We make no assumptions regarding the integrity of the software/firmware of EV after the registration protocol.

B. Permit Collection

Before an EV attempts to make a charging request, it needs to execute Protocol 2 to obtain a permit anonymously from the FA. This is to ensure that each EV makes only one reservation at a time. When an EV v communicates with the FA to acquire a permit for the first time, it makes a deposit anonymously based on the amount required by the FA. To ensure anonymity during this step, the deposit must be anonymous and untraceable. This can be done using, for example, cash, prepaid cash cards, or privacy-preserving cryptocurrencies. In all the subsequent times, the EV presents a receipt for a previously issued permit to acquire a new permit anonymously. To defend against application-level DoS attacks, the initial deposit must be high enough to discourage the same EV from obtaining many permits that can be used to launch a application-level DoS attack. As long as the EV acts benignly, it will not lose anything.

Protocol 1 Registration Protocol

Input: Encrypted identity $K_v(ID_v)$, key shares $K^{\text{DMV}}_v, K^{\text{CA}}_v$, symmetric keys $K_{v\text{DMV}}, K_{v\text{CA}}$.

Output: Certificate $Cert^v_{\text{CA}}$ from the CA.

1: EV v sends encrypted identity and key share K^{DMV}_v to the DMV, encrypted using symmetric key $K_{v\text{DMV}}$:

$$v \rightarrow \text{DMV}: K_{v\text{DMV}}(K^{\text{DMV}}_v || K_v(ID_v)).$$

2: The DMV decrypts using $K_{v\text{DMV}}$, and stores the encrypted identity of v along with K^{DMV}_v. The DMV creates $tkt = K_v(ID_v)||n$:

$$\text{DMV} \rightarrow v: PK_{CA}(SK_{DMV}(tkt)).$$

3: EV v sends its certificate request ($Cert$), signed encrypted message from the DMV, encrypted identity and key share K^{CA}_v to the CA, encrypted using symmetric key $K_{v\text{CA}}$:

$$v \rightarrow \text{CA}:$$

$$K_{v\text{CA}}(Cert||PK_{CA}(SK_{DMV}(tkt))||K^{\text{CA}}_v||K_v(ID_v)).$$

4: The CA decrypts using $K_{v\text{CA}}$, and then decrypts $PK_{CA}(SK_{DMV}(tkt))$ using SK_{CA}, verifies the signature using PK_{DMV} and if $K_v(ID_v)$ in tkt matches $K_v(ID_v)$ sent by v, the CA signs tkt with its private key SK_{CA} and encrypts it with the public key of the DMV PK_{DMV} and sends this signed encrypted message to the DMV.

$$\text{CA} \rightarrow \text{DMV}: PK_{DMV}(SK_{CA}(tkt)).$$

5: The DMV decrypts $PK_{DMV}(SK_{CA}(tkt))$ using SK_{DMV}, verifies the signature using PK_{CA}, and if tkt from the CA matches the tkt it sent to v and if the EV is physically present at the DMV, the DMV notifies the CA that the EV v with encrypted identity $K_v(ID_v)$ is currently executing the registration protocol.

6: If the CA receives the notification from the DMV that the EV v is currently executing the registration protocol, the CA signs the certificate request and sends this certificate to v and stores the encrypted identity of v along with K^{CA}_v:

$$\text{CA} \rightarrow v: K_{v\text{CA}}(Cert^v_{\text{CA}}).$$

To request a permit anonymously, the EV creates a permit request p by combining E and r, where E is a public number published by the FA, and r is a secret random number. The EV blinds the permit request p using a blinding factor b_f, resulting in a blinded permit request $b_f(p)$. Then, v sends this blinded permit request along with the initial deposit or a permit receipt to the FA. The FA verifies the

deposit or the permit receipt, and signs the blinded permit request if it is valid. The EV applies the unblinding factor u_f to the blindly signed permit request, resulting in a blind signature of the FA on the permit request. Signature of the FA on the permit request p is considered as a permit. The EV can prove the ownership of the permit by revealing the r since only v knows about it.

Protocol 2 Permit Collection Protocol

Input: Deposit or permit receipt, blinding & unblinding factors (b_f, u_f).

Output: Permit request p, permit ϕ^v.

1: v creates permit request $p = E||r$ where E is publicly known and r is a random secret.
2: v computes blinded message $b_f(p)$.
3: $v \rightarrow$ FA: (deposit / receipt, $b_f(p)$).
4: The FA verifies deposit or receipt and signs the blinded message: $BS_{\text{FA}}(b_f(p))$.
5: $FA \rightarrow v$: $BS_{\text{FA}}(b_f(p))$.
6: v obtains the permit ϕ^v by unblinding the signature on the message: $\phi^v = u_f(BS_{\text{FA}}(b_f(p))) = BS_{\text{FA}}(p)$.

C. Token Collection

The EV executes Protocol 3 to obtain a token anonymously from the charging station $\mathcal{U}$. This token will be used in the pseudonym collection protocol to obtain a pseudonym anonymously. In Step 1, the EV v and the charging station agree upon a base g as the base for DHKE; v then generates two random numbers x and z, and computes g^x and a commitment $c(g^x, z)$. The commitment is used to commit z to open g^x. EV v also creates a certificate $Cert^v$. To ensure its authenticity, it is signed with the public key corresponding to the certificate $Cert^v_{\text{CA}}$ issued to v by the CA during the registration protocol. The EV creates a token request t containing the commitment and the certificate encrypted with the public key of the CA, $PK_{\text{CA}}(Cert^v)$, to be signed by the charging station.

The EV blinds the token request t as $b_e(t)$ with a blinding factor e, and creates its signature σ_v upon that blind token request using its private key SK_v, in Step 2. In Step 3, v sends the blind token request along with the signature to the charging station. For the cut-and-choose protocol, v computes a specific number (determined by the implementation) of blind token requests, and sends them to the charging station $\mathcal{U}$. Each blind token request has a unique certificate $Cert^v$ in it. To verify that $Cert^v$ is a valid certificate, the charging station $\mathcal{U}$ uses the partial message proof through the cut-and-choose protocol as follows. From all the blind token requests received by the charging station from EV v, the charging station selects a random blind token request and asks v to unblind all the blind token requests except the selected one.

With the exception of the selected blind token request, the EV provides all the other token requests and their blinding factors. The charging station verifies all the token requests against the corresponding blind token requests and sends all the encrypted certificates in the token requests to the CA. Since the certificates are encrypted using the public key of the CA, the CA decrypts them using its private key and verifies that they are valid certificates (signed by the public key corresponding to one of the certificates issued by the CA, in this case $Cert^v_{\text{CA}}$). After ensuring that all of the certificates are valid, the CA notifies the charging station $\mathcal{U}$ regarding their validity. Since the selection is random, the charging station knows that the selected blind token request has a valid certificate in it. The cut-and-choose protocol have little communication overhead, because all the blind token requests are sent to $\mathcal{U}$ from v in a single message.

In Step 4, after ensuring that the blind token request actually contains a valid certificate and a valid digital signature (σ_v is verified using the public key of the EV, PK_v), the charging station issues a digital signature on the blind token request and then issues a signature on the whole message ($\sigma_{\mathcal{U}}$). The charging station sends the blindly signed token request and the digital signature to EV v. After v receives the blindly signed token request from $\mathcal{U}$, it verifies that the digital signature on the blindly signed token request is a valid signature (from the charging station), and applies the unblinding factor to retrieve the charging station's signature on the token request in Step 6. The charging station's signature on the token request t is considered a token, τ^v.

Protocol 3 Token Collection Protocol

Input: Valid certificate $Cert^v_{\text{CA}}$ from the CA, blinding & unblinding factors (b_e, u_e).

Output: Token τ^v to interact with the charging station $\mathcal{U}$.

1: v generates random numbers x and z, and computes a commitment $c(g^x, z)$, a certificate $Cert^v$ signed by the public key certified in $Cert^v_{\text{CA}}$, and creates a token request $t = (c(g^x, z)||PK_{\text{CA}}(Cert^v))$.
2: v computes $b_e(t)$, and the signature σ_v.
3: $v \rightarrow \mathcal{U}$: $(b_e(t)||\sigma_v)$.
4: $\mathcal{U}$ verifies σ_v, verifies the certificate in t through a cut-and-choose protocol, signs the blind token request, and then signs the whole message: $(BS_{\mathcal{U}}(b_e(t))||\sigma_{\mathcal{U}})$.
5: $\mathcal{U} \rightarrow v$: $(BS_{\mathcal{U}}(b_e(t))||\sigma_{\mathcal{U}})$.
6: v verifies $\sigma_{\mathcal{U}}$ and obtains the token by unblinding the signature on t: $\tau^v = u_e(BS_{\mathcal{U}}(b_e(t))) = BS_{\mathcal{U}}(t)$.

D. Pseudonym Collection

Protocol 4 illustrates the pseudonym collection protocol. In this protocol, EV v uses an unused token τ^v to acquire a pseudonym and a session key to be used during the report and receipt generation protocol. The EV sends a part of the commitment, g^x, to the charging station $\mathcal{U}$ in Step 1. As mentioned before, g is the agreed base of the DHKE protocol by both v and $\mathcal{U}$. After receiving g^x, the charging station generates a random number y, computes g^y, and calculates the session key $K_{v\mathcal{U}} = g^{yx}$ in Step 2. In Step 3,

the charging station creates a message containing g^y and $(g^x||g^y||\sigma_{\mathcal{U}})$ encrypted with the key $K_{v\mathcal{U}}$, where $\sigma_{\mathcal{U}}$ is a signature of $\mathcal{U}$ on $(g^x||g^y)$. The charging station sends this message to v in Step 4.

In Step 5, after receiving the message from $\mathcal{U}$, v computes $K_{v\mathcal{U}} = g^{xy}$ by using g^y from the message. After computing $K_{v\mathcal{U}}$, v proceeds to decrypt the rest of the message. The EV verifies the validity of digital signature $\sigma_{\mathcal{U}}$ on $(g^x||g^y)$ to ensure that it is from $\mathcal{U}$. If it is valid, then $\mathcal{U}$ has authenticated itself to v. In Step 6, v sends $\mathcal{U}$ an encrypted message containing g^x, g^y, $c(g^x, z)$, z, τ^v, and $PK_{\text{CA}}(Cert^v)$. After receiving this message, $\mathcal{U}$ decrypts it and verifies that τ^v is its signature on $c(g^x, z)||PK_{\text{CA}}(Cert^v)$ in Step 7. Since v provided it along with the commitment $c(g^x, z)$ and the secret z that opens the commitment, v has authenticated itself anonymously to $\mathcal{U}$. In Step 8, $\mathcal{U}$ creates a pseudonym δ^v, and derives two keys from the key $K_{v\mathcal{U}}$, namely ${K_{v\mathcal{U}}}^E$ and ${K_{v\mathcal{U}}}^A$, using a Key Derivation Function (KDF).

We use Encrypt-then-MAC (ETM) to ensure that only untampered messages are read by both the EV and the charging station. To provide general security of the protocols, we use the KDF to generate two keys for the ETM. The key ${K_{v\mathcal{U}}}^E$ is used for encryption while the key ${K_{v\mathcal{U}}}^A$ is used for Message Authentication Code (MAC). It is a strongly recommended cryptogaphic practice to not use one key for encryption and MAC (use two different keys, one for encryption and one for MAC) [15]. This ensures that if the key in either one of the encryption or the authentication scheme is compromised, then the other scheme is not automatically compromised. In Step 9, $\mathcal{U}$ sends a message to v containing the pseudonym and $PK_{\text{CA}}(Cert^v)$. After receiving the message, v verifies that its MAC is correct, decrypts it, and stores the pseudonym δ^v, in Step 10.

Protocol 4 Pseudonym Collection Protocol

Input: Token τ^v, commitment $c(g^x, z)$, proof of commitment (g^x,z), encrypted certificate $PK_{\text{CA}}(Cert^v)$.
Output: Pseudonym δ^v for reporting to $\mathcal{U}$.

1: $v \rightarrow \mathcal{U}$: g^x.
2: $\mathcal{U}$ generates random y, computes g^y and $K_{v\mathcal{U}} = g^{yx}$.
3: $\mathcal{U}$ generates a message containing $(g^x||g^y)$ and its signature $\sigma_{\mathcal{U}}$ on $(g^x||g^y)$, and encrypts it with $K_{v\mathcal{U}}$.
4: $\mathcal{U} \rightarrow v$: $(g^y, K_{v\mathcal{U}}(g^x||g^y||\sigma_{\mathcal{U}}))$.
5: v uses g^y and calculates $K_{v\mathcal{U}} = g^{xy}$ and decrypts the rest of the message using $K_{v\mathcal{U}}$ and verifies $\sigma_{\mathcal{U}}$.
6: $v \rightarrow \mathcal{U}$: $K_{v\mathcal{U}}(g^x||g^y||c(g^x, z)||PK_{\text{CA}}(Cert^v)||\tau^v||z)$.
7: $\mathcal{U}$ decrypts the message using $K_{v\mathcal{U}}$, opens the commitment using z and then checks that τ^v is a valid token.
8: $\mathcal{U}$ creates a pseudonym δ^v and derives two keys ${K_{v\mathcal{U}}}^E$ and ${K_{v\mathcal{U}}}^A$ from key $K_{v\mathcal{U}}$ using a KDF.
9: $\mathcal{U} \rightarrow v$: $ETM^{{K_{v\mathcal{U}}}^A}_{{K_{v\mathcal{U}}}^E}(\delta^v||PK_{\text{CA}}(Cert^v))$.
10: v decrypts the message and stores δ^v and $PK_{\text{CA}}(Cert^v)$.

E. Report and Receipt Generation

Protocol 5 Report and Receipt Generation Protocol

Input: Permit ϕ^v, permit request p, pseudonym δ^v, encrypted certificate $PK_{\text{CA}}(Cert^v)$, session key $({K_{v\mathcal{U}}}^E, {K_{v\mathcal{U}}}^A)$, charging request information $info$.
Output: Transaction receipt $\sigma^{\rho}_{\mathcal{U}}$, permit receipt $\sigma^{\phi^v}_{\mathcal{U}}$.

1: v creates message $cred = ((info||\phi^v)||PK_{\text{FA}}(p))$, and charging request $\rho = (\delta^v||PK_{\text{CA}}(Cert^v)||cred||\sigma^{cred}_{Cert^v})$.
2: $v \rightarrow \mathcal{U}$: $ETM^{{K_{v\mathcal{U}}}^A}_{{K_{v\mathcal{U}}}^E}(\rho)$.
3: $\mathcal{U}$ decrypts the message and verifies that δ^v is unused.
4: $\mathcal{U} \rightarrow$ DMV: $(cred||\sigma^{cred}_{Cert^v})$.
5: $\mathcal{U} \rightarrow$ CA: $PK_{\text{CA}}(Cert^v)$.
6: CA $\rightarrow$ DMV: $Cert^v$.
7: The DMV verifies that $\sigma^{cred}_{Cert^v}$ is the signature of $Cert^v$ on $cred$, and notifies $\mathcal{U}$.
8: $\mathcal{U} \rightarrow$ FA: $(\phi^v||PK_{\text{FA}}(p))$.
9: The FA decrypts $PK_{\text{FA}}(p)$, verifies that the permit ϕ^v is valid and unused, and notifies $\mathcal{U}$.
10: If both $\sigma^{cred}_{Cert^v}$ and the permit are valid, $\mathcal{U}$ reserves the charge and waits for EV arrival.
11: After v finishes the transaction, $\mathcal{U}$ issues v the receipts:

$$\mathcal{U} \rightarrow v\text{: } ETM^{{K_{v\mathcal{U}}}^A}_{{K_{v\mathcal{U}}}^E}(\rho||\sigma^{\rho}_{\mathcal{U}}||\sigma^{\phi^v}_{\mathcal{U}}),$$

where $\sigma^{\rho}_{\mathcal{U}}, \sigma^{\phi^v}_{\mathcal{U}}$ are $\mathcal{U}$'s signatures on ρ, ϕ^v respectively.

The report and receipt generation protocol is shown in Protocol 5. When v wants to communicate with $\mathcal{U}$ to schedule a charging service, it crafts a message $cred$ containing the request information, the permit, and the permit request encrypted with the FA's public key. Note that to make a reservation, the EV must use a unique and unused pseudonym each time. Otherwise, the anonymity of the EV may be broken if the adversary can link multiple reservations using the same pseudonym. It then creates a charging request ρ using the pseudonym δ^v, the certificate $Cert^v$ encrypted with the CA's public key, the message $cred$, and the signature of $cred$ using the private key corresponding to $Cert^v$. The request is then sent to $\mathcal{U}$, encrypted and authenticated with MAC.

The charging station decrypts the request and verifies that the included pseudonym is unused, and that $\sigma^{cred}_{Cert^v}$ is a valid signature on $cred$ by sending $cred$ and $\sigma^{cred}_{Cert^v}$ to the DMV and $PK_{\text{CA}}(Cert^v)$ to the CA. The CA retrieves the corresponding certificate and sends it to the DMV, which then verifies the signature and notifies $\mathcal{U}$. The charging station then extracts the permit and the encrypted permit request, and sends them to the FA securely. The FA decrypts the permit request, verifies its own signature on the permit as well as that the permit has never been used, and notifies $\mathcal{U}$. Upon receiving positive verification from all FTEs, the charging station will then proceed to preparing for the arrival of v for charging.

If v arrives for the reservation and completes the transac-

tion, $\mathcal{U}$ issues a receipt $\sigma_{\mathcal{U}}^{\rho}$ to v by signing the entire request ρ. It also issues a permit receipt $\sigma_{\mathcal{U}}^{\phi^v}$ by signing only on the permit ϕ^v. $\mathcal{U}$ only signs the request and the permit after the transaction is complete. This ensures that the EV will not leave the charging station before the transaction completes.

F. Anonymity Revocation

If v does not show up for charging according to the request ρ, the anonymity revocation protocol (Protocol 6) will be executed. First, $\mathcal{U}$ sends a complaint to the CA containing the request of v and the pseudonym used in the request. Since the permit is not signed, the malicious EV will not have a permit receipt, and hence it cannot acquire another permit without paying another deposit. The CA decrypts the message and then retrieves $Cert^v$. The CA decodes the private key of $Cert^v_{\text{CA}}$ used to sign $Cert^v$, and sends out a message requesting for the receipt of this request. If v does not provide the receipt within a time limit, the CA blacklists $Cert^v_{\text{CA}}$.

The CA then establishes a secure channel with DMV, and sends the key share K_v^{CA} corresponding to $Cert^v_{\text{CA}}$ to the DMV. Upon receiving this information, the DMV will use its own share K_v^{DMV} and the share K_v^{CA} given by the CA to reconstruct K_v, and proceed to decrypt the encrypted ID of the vehicle, $K_v(ID_v)$. If the charging station generates a false complaint and the EV possesses a receipt for the transaction, the EV provides the receipt, which is actually the charging station's signature on the request for reservation, to the CA:

$$v \rightarrow \text{CA}\colon (\rho || \sigma_{\mathcal{U}}^{\rho}).$$

The CA proceeds to verify if it is indeed a valid signature of the charging station in the receipt. If the signature is valid, the CA ignores the complaint of the charging station $\mathcal{U}$ on the EV. A separate procedure can later be executed to check if the charging station is compromised or malfunctioning, which is out of the scope of this paper.

Protocol 6 Anonymity Revocation Protocol

Input: Malicious charging request ρ.
Output: ID_v of v which issued ρ.

1: $\mathcal{U}$ complains to the CA with the unattended request ρ:

$$\mathcal{U} \rightarrow \text{CA}\colon \rho.$$

2: The CA gets $Cert^v$ and decodes which $Cert^v_{\text{CA}}$ signed it.
3: The CA broadcasts a request for receipt of the transaction pertaining to $Cert^v$. If the CA does not receive the receipt in a given amount of time, the CA blacklists $Cert^v_{\text{CA}}$.
4: $\text{CA} \rightarrow \text{DMV}\colon (Cert^v_{\text{CA}} || K_v^{\text{CA}})$.
5: The DMV retrieves K_v from $(K_v^{\text{CA}}, K_v^{\text{DMV}})$ and uses the key K_v to decrypt the identity of ID_v.
6: If v provides the receipt, then the CA verifies the validity of the receipt and abort the complaint from $\mathcal{U}$.

G. Making Concurrent Reservations

In practice, an EV user may occasionally have the demand to make multiple reservations in advance before fulfilling any of them, for example, when planning for a long trip. Because the main focus of our work is to defend against the application-level DoS attack by a malicious EV, we require that each EV can only obtain and possess one permit with one deposit, each permit redeemable for one reservation at any time.

To facilitate multiple charging events during a long travel, however, we allow an EV to obtain multiple permits for making multiple concurrent reservations, but the EV must make multiple deposits as well. To prevent an EV with multiple deposits from launching the application-level DoS attack, punishment can be enforced after the EV's malicious behavior is confirmed and its identity is revealed through anonymity revocation. For example, all the deposits associated with the permits that were used to launch an attack are forfeited to the FA.

V. Privacy and Security Analysis

In this section, we perform detailed privacy analysis of our framework, and discuss the security of the framework in face of the attacks described in Section II.

A. Privacy Analysis

Privacy from the charging station. The charging station cannot link a pseudonym or a permit used by an EV v to its original identity ID_v. Unlinkability of a token and a permit with the identity of the EV is ensured by the blind signature scheme used during the token collection protocol and the permit collection protocol.

In the pseudonym collection protocol, the EV produces a token and receives a pseudonym anonymously. Since a token cannot be tied to an identity, it is not possible to link the pseudonym to an identity. If each token is used only once to acquire a pseudonym, the charging station cannot link any two pseudonyms belonging to the same EV. Further, the blind signatures used in our framework ensure that the pseudonyms of any EV are unlinkable and indistinguishable from the pseudonyms of other EVs. Hence, tracking of multiple pseudonyms of a single EV or multiple EVs cannot compromise an EV's privacy either.

Privacy from FTEs. First, since the CA and the DMV only verify the validity of the encrypted certificates sent by the charging station, they do not know about the permits, tokens, or pseudonyms issued to an EV from the normal protocols. Specifically, the cut-and-choose protocols will prevent the CA from acquiring the specific certificate in the actual token used to acquire a pseudonym, as each token contains a different certificate generated by the EV itself.

The FA verifies the validity of permits, which are anonymized at issuance, so it cannot violate the privacy of the EV. The secret sharing scheme prevents either the DMV or the CA from unilaterally obtaining the EV's identity without the agreement from the other due to confirmed

maliciousness. Note that the registration protocol also employs anonymized communications. Since it happens before any permit or charging request, it does not contain any information regarding the permits, tokens and pseudonyms of an EV.

B. Security Analysis

DoS/DDoS attacks. A malicious EV may launch DoS attacks either on a single charging station, or on all charging stations in an area, which can prevent other EVs from receiving charging services. In our framework, however, launching such an attack requires the user to possess a large number of valid permits at the same time. This can be done through accumulating permits using one or multiple EVs.

However, since each permit requires a deposit of a high-enough amount, such an attack can only be launched by an attacker with huge financial resources, making the attack economically infeasible for most common attackers. Moreover, once an attack is detected, the user's initial deposit can be lost (due to inability to obtain another permit), and the CA can blacklist the certificate(s) of one or multiple involved EVs and broadcast to the whole network, so that future requests by the EVs will not be accepted network-wide. Both punishments make an attack extremely costly for the attacker, economically preventing the attack from happening.

Forgery attacks. Our system model assumes that an attacker cannot forge digital signatures. Unless an external attacker can forge digital signatures, she cannot forge tokens or pseudonyms. An EV may try to get the signature of a charging station on a real time report so as to forge a receipt. To do so, it needs to send the real time report as a blinded message in the token collection protocol, upon which the charging station can issue a blind signature. This is prevented by the cut-and-choose protocol, which checks for the commitment inside the message and sends the encrypted certificate to the CA for verification. It can also be prevented by having the charging station sign the receipts with a different public/private key pair.

Man-in-the-Middle attacks. In order to prevent the MITM attacks, our framework uses authentication in every protocol. For example, authentication between the EV and the charging station during the token collection protocol is achieved using their digital signatures on their messages. The charging station authenticates itself using the digital signature on (g^x, g^y) and the EV authenticates itself using the commitment in the pseudonym collection protocol. The EV authenticates to the charging station by signing the message with the certificate $Cert^v$ while reporting. During the receipt generation, the charging station authenticates through its signature on the message.

Replay attacks. Since all the messages are associated with timestamps, the framework is immune to replay attacks.

Collusion attacks. Collusion between malicious EVs and charging stations cannot affect the anonymity of the honest EVs. If there are at least two benign EVs, the charging station cannot differentiate between their pseudonyms and by extension cannot infer their private information.

VI. Performance Evaluation

In this section, we analyze and evaluate the performance of our framework.

A. Computation Overhead

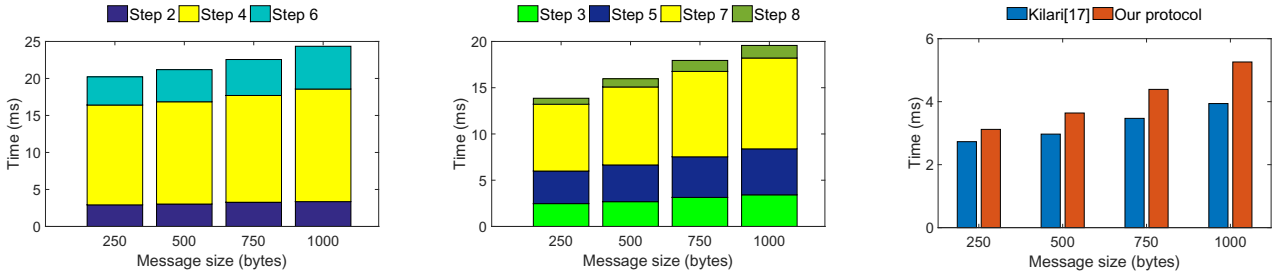
We first show the computation overhead of our protocols through real-world implementations. Our framework can be implemented without any special hardware. We implemented our protocols using standard crypto libraries in Java. The EV parts of the protocols were implemented on an Intel Core i5-2450M (2.5 Ghz) machine with 8 GB RAM. The charging station parts of the protocols were implemented on an Intel Core i7-6700K (4.0 Ghz) machine with 32 GB RAM. We measured the time taken by each computation-intensive step in the protocols in order to evaluate the scalability of the protocols. All the results were averaged over 1000 runs.

We used the following implementations of the cryptographic primitives in our protocols. For asymmetric encryption and Blind signatures, we used RSA with 2048-bit keys. For digital signatures, we used SHA256 with RSA. For symmetric encryption and ETM (Encrypt-Then-MAC), we used AES-CBC (Cipher Block Chaining) with 256-bit key and AES-CBC-MAC for MAC with 256-bit key. All the above implementations are in line with the National Institute of Standards and Technologies (NIST), USA recommendations [2].

Besides evaluating our proposed framework alone, we compared our framework to the framework in our preliminary work [17]. To the best of our knowledge, our framework in [17] is the state-of-the-art anonymous authentication solution that supports revocable anonymity for EV information reporting. Since the major difference between [17] and this paper is the report and receipt generation protocol, the comparison was done regarding this protocol alone.

We compared the time taken by computation-intensive steps in the token collection protocol, the pseudonym collection protocol, and the report and receipt generation protocol. Fig. 2(a) shows the time taken by the computation intensive steps in the token collection protocol. Step 2 corresponds to the time taken by an EV for a blinding and a digital signature operations. Step 4 corresponds to the time taken by a charging station for a signature verification, a blind signature, and a digital signature operations. Step 6 corresponds to the time taken by an EV for a digital signature verification and an unblinding operations. Time taken by the token collection protocol for a 1000 byte message was approximately 25 ms.

Fig. 2(b) shows the time taken by the computation intensive steps in the pseudonym collection protocol. Step 3 corresponds to the time taken by a charging station for a digital signature and an encryption operations. Step 5 corresponds to the time taken by an EV for a decryption



(a) In token collection protocol (b) In pseudonym collection protocol (c) During reporting by an EV

Fig. 2. Time taken by various steps in the protocols of our framework.

and a signature verification operations. Step 7 corresponds to the time taken by a charging station for a decryption, a commitment verification and a digital signature verification operations. Step 8 corresponds to the time taken by a charging station for an encryption operation. Time taken by the pseudonym collection protocol for a 1000 byte message was approximately 24 ms.

Fig. 2(c) compares the time taken by our report and receipt generation protocol against the time taken by the same protocol in [17]. Since receipt generation took approximately the same amount of time as reporting, we did not show it in the figures. As we can see, our protocol took slightly more time than Kilari *et al.* [17]. This was due to the encryption of the permit requests. During receipt generation, the additional time was due to computing the signature of the charging station on the permit. In our experiments, the maximum time taken in each protocol was in the order of 10 ms. The token collection and the pseudonym collection protocols can be executed when an EV is idle. Only the report and receipt generation protocol, which took less than 10 ms in experiments, is executed in real-time.

Step 4 in token collection and Step 7 in pseudonym collection took the longest time, due to operations involving digital signatures. Time taken by the steps increased with the message size, due to the dependence of cryptographic operations on the message size. The token collection and the pseudonym collection protocols took a combined 49 ms $(25 + 24)$ per EV. This means that a charging station can serve a million EVs in 14 hours. Even including the time required for the cut-and-choose protocols, the time required for our framework is less than an average idling time of an EV (15 mins, for every 24 hours), demonstrating the scalability of our framework.

B. Communication Overhead

Below, we further analyze the communication overhead of our framework in terms of the messages sent and received by the EV and the charging station during the execution of our normal operation protocols. Since Protocol 1 (Registration) and Protocol 6 (Anonymity Revocation) are executed infrequently, we omit their communication overhead in the analysis. Unlike the above experiments, we do not assume specific implementations of our protocols for generality. We summarized the communication overhead of our normal operation protocols in Table II. For simplicity, we omit the sizes of the payload message and the MAC.

We can see that the token collection protocol (Protocol 3) has the highest communication overhead. This is due to the use of the cut-and-choose protocol to verify the validity of EV's token requests. Assuming $M = 4096$ bits for the prime q of the group, $R = 10$ for the cut-and-choose protocol, $P = 4096$ bits for the size of the pseudonym, and $N = 1000$ EVs in the system, the worst-case size for the charging station is Protocol 3 with total size 18.55 MB while receiving and Protocol 5 with total size 6.35 MB while sending. The worst-case size for the EV is Protocol 3 with total size 19 KB while sending and Protocol 5 with total size 3.5 KB while receiving. Alternatively, we can use Zero Knowledge Proofs (ZKPs) to replace the cut-and-choose protocol, which may reduce the communication overhead but may lead to excessive computational overhead.

Only Protocol 5 must be executed in real-time during the time of reservation, while all other protocols (including Protocols 1 and 6) can be executed offline. Protocol 5 has the worst-case send message size of 6.35 MB and receive message size of 2.44 MB for a charging station. For an EV, Protocol 5 has the worst-case send message size of 2.5 KB and receive message size of 3.5 KB. Note that it is highly unlikely that all EVs will be interacting with the charging station at the same time, hence the worst-case is very rare. The worst-case communication overhead for the EV for the real-time protocol is low. Also, since the only protocol that presents a significant communication overhead to the EV and/or the charging station is Protocol 3 which is executed offline, and the communication overhead of both the EV and the charging station for the only real-time protocol (Protocol 5) is sufficiently low, our framework is easily scalable.

VII. Related Work

In this section, we briefly discuss the related work on anonymous authentication frameworks and revocable anonymity mechanisms. Anonymous authentication frameworks can be achieved by using Group signatures [6]. The drawback of Group signatures is that the group manager knows the identity of users. Users can be deanonymized if the manager is compromised or malicious. Camenish *et al.* [4] proposed

TABLE II

COMMUNICATION OVERHEAD: MEASURED IN TERMS OF THE NUMBERS AND SIZES OF MESSAGES SENT AND RECEIVED, WHERE ALL ASYMMETRIC OPERATIONS ARE PERFORMED ON A GROUP OF A PRIME ORDER OF M BITS. *CS* REPRESENTS A CHARGING STATION SERVING N EVS SIMULTANEOUSLY. R IS THE NUMBER OF MESSAGES USED IN THE CUT-AND-CHOOSE PROTOCOL. D IS THE SIZE OF DEPOSIT DURING INITIAL PERMIT COLLECTION IN PROTOCOL 2. P IS THE SIZE OF THE PSEUDONYM DURING PROTOCOL 4. C IS THE SIZE OF THE COMMITMENT KEY DURING PROTOCOL 4. PROTOCOLS 1 AND 6 ARE OMITTED BECAUSE THEY ARE EXECUTED INFREQUENTLY.

Protocol	**Entity**	**# Messages**		**Total Size**	
		Send	**Recv**	**Send**	**Recv**
Protocol 2	EV	1	1	$\max(D, 2M) + M$	M
Protocol 3	EV	2	2	$(4R-2) \cdot M$	$R + 2M$
	CS	$2N$	$2N$	$(R + 2M) \cdot N$	$(4R-2) \cdot MN$
Protocol 4	EV	2	2	$7M + C$	$5M + P$
	CS	$2N$	$2N$	$(5M + P) \cdot N$	$(7M + C) \cdot N$
Protocol 5	EV	1	1	$4M + P$	$6M + P$
	CS	$4N$	N	$(12M + P) \cdot N$	$(4M + P) \cdot N$

a token based credential system that lets a user anonymously authenticate itself at most n times using n one-time tokens. This method used online-zero knowledge proofs to verify the token, which is computationally expensive.

PACP [14] is a framework in which vehicles interact with the Road Side Units (RSUs) to generate pseudonyms for anonymous communications. However, this protocol contains a trusted third party (DMV) which knows the true identity of each user. Köpsell *et al.* [18] proposed a revocable anonymity framework based on threshold group signatures and blind signatures. It uses a user's self-generated identity certificate as the basis, which is verified by an intermediary before the start of the protocol. This framework is not applicable in Vehicle to Grid (V2G) communications, due to the lack of an intermediary which can verify an EV's identity.

Chowdury *et al.* [7] proposed a method for anonymous pseudonym-renewal and pseudonymous authentication for vehicular ad-hoc networks over a Named Data Networking architecture. However, it is not applicable to IP networks, while most current networks are IP based. Traceable signatures [16] are another cryptographic tool to provide conditional anonymity. However, their drawback is that they are too computationally expensive to be used for communications between EVs and charging stations.

Lu *et al.* [24] proposed a conditional privacy preservation scheme in vehicular ad hoc networks (VANETs) which divides privacy into three levels. This scheme contains a Trusted Authority which is trusted by all the parties in the system and is assumed to be immune to compromise by an attacker. Also, this scheme is proposed for VANETs (which have specific properties and requirements) and has scalability issues. Regarding V2G communications specifically, various efforts in the literature concentrated on privacy preserving V2G communications [32], [23], [31]. These solutions however do not address the authentication problem, hence they are vulnerable to external attacks from unauthorized parties.

Li *et al.* [20] proposed a pseudonym-based authentication method, which allows at least one party to know an EV's identity. Li *et. al.* [21] further proposed a method with complete anonymity for real-time reporting of EVs, using partially blind signatures. Saxena *et al.* [27], [28] proposed mutual authentication methods that provide forward privacy, identity anonymity, and untraceability. Afrin *et al.* [1] proposed an authentication method to achieve anonymity and time-flexibility in EV charging scheduling. However, all these solutions are vulnerable to insider attacks by malicious EVs.

Kilari *et al.* [17] proposed a revocable anonymous authentication framework, which provides anonymity revocation in case of EV misbehavior. Yet, the framework is vulnerable to application-level DoS attacks. Other related work to security and privacy in EV charging and communications includes privacy preserving auctions [19], blockchain-based EV charging scheduling [30], privacy preserving reputation evaluation [22], etc. They concern different problems from ours, and are therefore orthogonal to our work.

VIII. CONCLUSION

In this paper, we proposed a revocable anonymous authentication framework that is robust against application-level DoS attacks. Our framework allows EVs to authenticate themselves anonymously to a charging station for reporting real-time information. To prevent malicious EVs from anonymously attacking the system, we equipped our framework with the ability to revoke the anonymity of an EV with verified maliciousness. To further protect charging stations and benign EVs from application-level DoS attacks by malicious EVs, we designed a permit-based mechanism to enforce excessive cost and penalization for launching an attack, economically preventing the attacks from happening. We thoroughly analyzed the security and privacy properties of our framework, and evaluated its performance through analysis and implementations in real-world settings. The analysis and experiments showed that our framework is both efficient and scalable.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers, the Associate Editor, and the Editor for their insights.

REFERENCES

[1] S. Afrin and A. Kwasinski, "A Privacy-Preserving Method with Flexible Charging Schedules for Electric Vehicles in the Smart Grid," in *Proc. IEEE ANTS*, 2017, pp. 1–6.

[2] E. Barker and A. Roginsky, "Transitioning the use of cryptographic algorithms and key lengths," National Institute of Standards and Technology, Tech. Rep., 2018.

[3] G. R. Blakley, "Safeguarding Cryptographic Keys," in *Proc. AFIPS National Computer Conference*, 1979, pp. 313–317.

[4] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to Win the Clonewars: Efficient Periodic N-Times Anonymous Authentication," in *Proc. ACM CCS*, 2006, pp. 201–210.

[5] D. Chaum, "Blind Signatures for Untraceable Payments," in *Proc. Crypto*, 1983, pp. 199–203.

[6] D. Chaum and E. Van Heyst, "Group Signatures," in *Proc. EUROCRYPT*, 1991, pp. 257–265.

[7] M. Chowdhury, A. Gawande, and L. Wang, "Anonymous Authentication and Pseudonym-renewal for VANET in NDN," in *Proc. ACM ICN*, 2017, pp. 222–223.

[8] J. Crosbie, "Elon Musk Finally Reveals the Number of Tesla Model 3 Reservations," 2017.

[9] S. Davis, S. Williams, R. Boundy, and S. Moore, "2015 Vehicle Technologies Market Report," *Oak Ridge National Laboratory*, 2015.

[10] ——, "2016 Vehicle Technologies Market Report," *Oak Ridge National Laboratory*, 2017.

[11] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and Authenticated Key Exchanges," *Designs, Codes and Cryptography*, vol. 2, pp. 107–125, 1992.

[12] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proc. USENIX Security*, 2004, pp. 303–320.

[13] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart gridthe new and improved power grid: A survey," *IEEE communications surveys & tutorials*, vol. 14, no. 4, pp. 944–980, 2011.

[14] D. Huang, S. Misra, M. Verma, and G. Xue, "PACP: An Efficient Pseudonymous Authentication-Based Conditional Privacy Protocol for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 736–746, 2011.

[15] J. Katz, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[16] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable Signatures," in *Proc. EUROCRYPT*, 2004, pp. 571–589.

[17] V. T. Kilari, S. Misra, and G. Xue, "Revocable Anonymity Based Authentication for Vehicle to Grid (V2G) Communications," in *Proc. IEEE SmartGridComm*, 2016, pp. 351–356.

[18] S. Köpsell, R. Wendolsky, and H. Federrath, "Revocable Anonymity," in *Proc. ETRICS*, 2006, pp. 206–220.

[19] D. Li, Q. Yang, D. An, W. Yu, X. Yang, and X. Fu, "On Location Privacy-Preserving Online Double Auction for Electric Vehicles in Microgrids," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[20] H. Li, G. Dán, and K. Nahrstedt, "Portunes: Privacy-preserving Fast Authentication for Dynamic Electric Vehicle Charging," in *Proc. IEEE SmartGridComm*, 2014, pp. 920–925.

[21] ——, "Lynx: Authenticated Anonymous Real-Time Reporting of Electric Vehicle Information," in *Proc. IEEE SmartGridComm*, 2015, pp. 599–604.

[22] Z. Li and C. T. Chigan, "On Joint Privacy and Reputation Assurance for Vehicular Ad Hoc Networks," *IEEE Trans. Mob. Comput.*, vol. 13, no. 10, pp. 2334–2344, 2014.

[23] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, "Aggregated-proofs Based Privacy-Preserving Authentication for V2G Networks in the Smart Grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1722–1733, 2012.

[24] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 1229–1237.

[25] J. Perkowski, "What China's Shifting Subsidies Could Mean For Its Electric Vehicle Industry." [Online]. Available: https://www.forbes.com/sites/jackperkowski/2018/07/13/china-shifts-subsidies-for-electric-vehicles/

[26] M. O. Rabin, "Digitalized Signatures," *Foundations of Secure Computation*, vol. 78, pp. 155–166, 1978.

[27] N. Saxena and B. J. Choi, "Authentication Scheme for Flexible Charging and Discharging of Mobile Vehicles in the V2G Networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1438–1452, 2016.

[28] N. Saxena, S. Grijalva, V. Chukwuka, and A. V. Vasilakos, "Network Security and Privacy Challenges in Smart Vehicle-To-Grid," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 88–98, 2017.

[29] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, pp. 612–613, 1979.

[30] Z. Su, Y. Wang, Q. Xu, M. Fei, Y. Tian, and N. Zhang, "A Secure Charging Scheme for Electric Vehicles with Smart Communities in Energy Blockchain," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[31] H.-R. Tseng, "A Secure and Privacy-Preserving Communication Protocol for V2G Networks," in *Proc. IEEE WCNC*, 2012, pp. 2706–2711.

[32] Z. Yang, S. Yu, W. Lou, and C. Liu, "P2: Privacy-preserving Communication and Precise Reward Architecture for V2G Networks in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 697–706, 2011.

Vishnu Teja Kilari (STM'13) received his M.S. degree from Arizona State University, Tempe, Arizona, U.S.A in 2013. Currently he is a Ph.D student in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. His research interests include Botnets, Smart Grid security and hardware assisted security.

Ruozhou Yu (STM'13, M'19) is an Assistant Professor of Computer Science at North Carolina State University. He received his Ph.D degree (2019) in Computer Science from Arizona State University, Tempe, Arizona, USA. His research interests include internet-of-things, cloud/edge computing, smart networking, algorithms and optimization, security and privacy, blockchain, etc.

Satyajayant Misra (STM'05, M'09) is an Associate Professor in Computer Science at New Mexico State University. He completed his M.Sc. in Physics and Information Systems from BITS, Pilani, India in 2003 and his Ph.D. in Computer Science from Arizona State University, Tempe, USA. His research interests include security, privacy, and resilience in wireless networks, the Internet, supercomputing, and in IoT/CPS. He has served on several IEEE journal editorial boards and IEEE/ACM conference executive committees.

Guoliang Xue (M'96, SM'99, F'11) is a Professor of Computer Science and Engineering at Arizona State University. He received the Ph.D degree (1991) in computer science from the University of Minnesota, Minneapolis, USA. His research interests include survivability, security, and resource allocation issues in networks. He is the Area Editor of *IEEE Transactions on Wireless Communications* for the area of Wireless Networking.