Authentication Scheme for Flexible Charging and Discharging of Mobile Vehicles in the V2G Networks

Neetesh Saxena, *Member, IEEE,* and Bong Jun Choi, *Member, IEEE*

Abstract*—Navigating security and privacy challenges is one of the crucial requirements in the Vehicle-to-Grid (*V2G*) network. Since Electric Vehicles (*EV*) need to provide their private information to aggregators/servers when charging/discharging at different charging stations, privacy of the vehicle owners can be compromised if the information is misused, traced, or revealed. In a wide *V2G* network, where vehicles can move outside of their home network to visiting networks, security and privacy becomes even more challenging due to untrusted entities in the visiting networks. Although some privacy-preserving solutions were proposed in literature to tackle this problem, they do not protect against well-known security attacks and generate a huge overhead. Therefore, we propose a mutual authentication scheme to preserve privacy of the *EV*'s information from aggregators/servers in the home as well as distributed visiting *V2G* networks. Our scheme, based on a bilinear pairing technique with an accumulator performing batch verification, yields higher system efficiency, defeats various security attacks, and maintains untraceability, forward privacy, and identity anonymity. Performance analysis shows that our scheme, in comparison with existing solutions, generates significantly lower communication and computation overheads in the home and centralized *V2G* networks, and comparable overheads in the distributed visiting *V2G* networks.

Keywords*—authentication, bilinear pairing, privacy-preserving, security attacks, V2G;

I. INTRODUCTION

In the future, Vehicle-to-Grid (*V2G*) system is expected to be one of the most powerful systems in the smart grid by integrating with renewable energy sources to provide ancillary services, and keeps track of the power demand utilized by the Electric Vehicles (*EV*)/Battery Vehicles (*BV*). These vehicles can communicate with the smart grid under distributed and/or centralized *V2G* networks for charging/discharging their batteries from/to the grid. To support *V2G* communications, a Dedicated Short Range Communication (*DSRC*) standard protocol is specifically designed for Vehicle-to-Vehicle (*V2V*) and Vehicle-to-Infrastructure (*V2I*) that includes *IEEE 802.11p* and *IEEE 1609* Wireless Access in Vehicular Environments (*WAVE*) [1].

Several parties have significant interest in exploring the possibilities of the *V2G* operations, such as vehicle supplier, vehicle battery supplier, vehicle owner, Electric Vehicle Supply Equipment (*EVSE*) owner, business/home users, aggregation service provider, and electrical utility [2]. In addition, US Department of Defense also has significant interest in *V2G*. Regulatory and governmental agencies also have particular motivations for investigating *V2G*. It is expected that in the future, *V2G* will assist both, Plug-in Electric Vehicle (*PEV*) and renewable energy to increase market penetration. Furthermore, *V2G* can also provide peak power and cost-benefit, as currently meeting the demands of peak power is a very expensive obligation for utilities. It can also provide the operating reserve, which is available online within a short time in case of any disruption to the electricity supply.

V2G communication system is different from other existing communication systems in several aspects, such as vehicle mobility, geographical location of the vehicle, charge and discharge operations, driving pattern, and limited communication range. Non-cooperative (individual benefit of selfish *EV*) and cooperative (overall benefit of the connected *EV*) optimization approaches are used to optimize charging of *EV*'s battery under uncertain demand [3]. It takes almost 10 hours to charge a 15-kWh battery using a standard 120-volt outlet [4]. In terms of security, authentication in the *V2G* network needs to be fast and efficient in order to support a large number of *EV*s expected to participate in dynamic charging/discharging [5]. Also, confidential information like vehicle identity, vehicle type, charging and discharging time, and Charging Station Identity (*CSID*) needs to be protected.

A. Research Challenges

*EV*s perform charge and discharge operations in order to meet their energy demand and to balance the power in the grid. In the centralized *V2G* network, where the power is directly supplied to the grid, vehicles can only perform discharge operation [6], [7], [8], [9]. This power is absorbed by the smart grid and is supplied to the areas where balancing of power-demand is required. In the distributed *V2G* network, vehicles perform local charge and discharge operations, and the power is only used within the local area to fulfill the power demand. The local area where a vehicle is registered

N. Saxena and B. J. Choi are with the Computer Science Department, The State University of New York (SUNY) Korea, S. Korea and also with the Department of Computer Science, Stony Brook University, USA. e-mail: mr.neetesh.saxena@ieee.org, bjchoi@sunykorea.ac.kr.

is commonly referred as its home area, and outside this area is referred to as its visiting area. A distributed *V2G* network consists of home as well as visiting *V2G* networks, while a centralized *V2G* network is considered as visiting network for all the vehicles [6], [10].

In the *V2G* network, an *EV* is connected to a Charging Station (*CS*) for charging/discharging its battery. The *EV* provides variety of information to the *CS*, such as its identity, State of Charge (*SoC*), desired *SoC*, and charging/discharging preferences [11]. The Local Aggregator (*LAG*) collects information from the connected vehicles and forwards some of the information to the Control Center (*CC*) for monitoring and billing purposes. However, it is possible for the *LAG* to reveal and misuse this information for its benefit [6], [11].

There exist various security and privacy challenges in the *V2G* system that can massively affect practical usage of this next generation technology [12]. The information shared by the *EV*s and other *V2G* entities, such as *LAG*, Certification/Registration Authority (*CA/RA*), and *CC* must be secured over the network, and privacy of the personal and confidential information must be maintained. According to *IEC 15118-2* [13], only unilateral authentication (server side authentication) is mandatory and mutual authentication (both server and *EV* authentication) is optional. However, unilateral authentication is not considered secure, as it may result in redirection and impersonation attacks. It is risky to assume that all the *LAG*s and/or *CA/RA*s are trusted entities. As a point of strong security in the future generation *V2G* network, we strongly emphasize that the *V2G* system must provide mutual authentications between all *EV*s and their respective *LAG*s or *CA/RA*s in order to ensure the communication involvement only by the legitimate entities. Furthermore, the *LAG* must not be able to recognize and keep track any *EV* by its information and behavioral pattern. Otherwise, the *LAG* can misuse the information resulting in insider attacks.

The existing protocols/schemes do not discuss some of the possible attacks in the *V2G* network, such as Man-in-the-Middle (*MITM*), replay, impersonation, redirection, known key, and repudiation attacks. Furthermore, there is always a possibility of insider attacks. Moreover, protection of private information of the vehicles and resistance against security attacks are more challenging outside of its home *V2G* network, as the vehicles may also interact with untrusted *LAG*s and/or *CA/RA*s. Also, since a huge number of entities would be involved in the future distributed and centralized *V2G* networks, the generated overheads must be kept as low as possible. These overheads have direct impact on the optimal performance-security trade-off. Therefore, a secure, lightweight, and privacy-preserved authentication scheme for the home, visiting, and centralized *V2G* networks is needed.

B. Security Goals and Requirements in the V2G Networks

There are various security goals and requirements in the *V2G* network, such as authentication, forward secrecy, information confidentiality, and message integrity. The *V2G* network also suffers from various security attacks due to its connectivity with Internet. We define the security properties in the *V2G* network as follows:

1) Authentication: Authentication is one of the mandatory requirements that enables communication between legitimate entities, and defeats impersonation attacks.

Definition 1: A mutual authentication holds if (i) the *EV* successfully verifies the *LAG* and/or *CA/RA*, and (ii) the *EV* is also verified by the *LAG* and/or *CA/RA* before the actual communication starts. The computed secret parameters must be verified by the involved entities.

$$EV| \equiv LAG| \equiv CA/RA \rightarrow EV| \equiv LAG \wedge EV| \equiv CA/RA.$$

2) Perfect Forward Secrecy (PFS): At any stage of the scheme, adversary $\mathcal{A}$ is allowed making a query to learn information about an unexpired secret key. $\mathcal{A}$ guesses whether the learned challenge is a true session key or a random key.

Definition 2: A scheme maintains *PFS* if no adversary $\mathcal{A}$ in time t can retrieve the past session keys k, even the long term keys *LTK* (*i.e.*, the private key of the vehicle or a session key) are compromised, when (i) entities involved in the session compute same key, and (ii) $\mathcal{A}$ wins if its output bit $b^{'}$ is equal to a randomly chosen bit b selected in query. $\mathcal{A}$, running against the scheme, has negligible advantage as

$$Adv_{k,LTK}^{pfs}(\mathcal{A}) = Pr[b = b^{'}] - 1/2.$$

3) Information Confidentiality: Each encrypted message in the *V2G* scheme must provide enough security to be indistinguishable from a randomly generated message, considering adversary $\mathcal{A}$ has access to an encryption oracle that encrypts messages chosen by $\mathcal{A}$'s without knowing the secret key. In other words, the scheme must support Indistinguishability under the Chosen Message Attack (*IND-CMA*).

Definition 3: A scheme is *IND-CMA* secure if no adversary $\mathcal{A}^{Enc_k(.)}$ in time t can distinguish between two chosen messages msg_0 and msg_1, and has no or negligible advantage.

$$Adv_{Enc_k}^{ind-cma}(\mathcal{A}) = Pr[\mathcal{A}^{Enc_k(.)}(msg_0) = 1] - \\ Pr[\mathcal{A}^{Enc_k(.)}(msg_1) = 1] \leq \epsilon.$$

4) Message Integrity: Integrity of each message can be achieved using a well known Collision-Resistant Hash Functions (*CRHF*).

Definition 4: $H : \{0,1\}^n \rightarrow \{0,1\}^m$ $(m<n)$ is a collision-resistant hash function if it there exists a negligible function ϵ such that for all security parameters $n \in \mathbb{N}$,

$$Pr[(msg_0, msg_1) \leftarrow \mathcal{A}(1^n, h) : msg_0 \neq msg_1 \wedge \\ h(msg_0) = h(msg_1)] \leq \epsilon(n).$$

The advantage of $\mathcal{A}$ in breaking H under security notion $bhf \in \{$*collision*, *preimage*, *second-preimage*$\}$ is given by $Adv_H^{bhf}(\mathcal{A}) = Pr[msg_0 \neq msg_1$ and $H(msg_0) = H(msg_1)]$.

C. Privacy Requirements in the V2G Networks

The privacy of the *EV*s must be preserved whenever *EV*s access charging stations. Revealing vehicle's identity to $\mathcal{A}$ results in privacy breaches, in which $\mathcal{A}$ can track the behavior of the victim vehicle, and performs some unwanted activities, such as linking messages to extract partial secret information of the victim, and retrieving personal and location information.

The vehicle's private information, such as its location and battery status should not be revealed during the authentication. For example, the *LAG* should not be able to retrieve the location of the *EV* making a request. Similarly, the *LAG* must not be able to track the *EV* based on its battery status. The required information should directly be sent to the intended recipient in a secure manner. Furthermore, the *LAG* must be unaware of the *EV*'s timing selection and the choice of operation. Similarly, the *LAG* must not be able to track the *EV* based on other operational information like *CSID*. We define the following privacy properties in the *V2G* networks:

Definition 5: (Vehicle Untraceability and Information Privacy): Vehicle untraceability is maintained if $\mathcal{A}$ cannot distinguish whether two generated messages (with pseudo-identity and/or vehicle's location, battery status, selection of charging/discharging, and expected time information), say msg_0 and msg_1, correspond to same or two different identities of the vehicles, say ID_0 and ID_1. The game is over once $\mathcal{A}$ announces its guess of the selected message. A scheme satisfies untraceability if $\mathcal{A}$ cannot select the correct message with probability higher than that of random guessing. We also define forward and backward vehicle untraceability.

- Forward untraceability is maintained if $\mathcal{A}$ cannot determine whether a vehicle at time t_{frw} ($t_{frw} > t_{current}$) will be involved in communication based on current derived information.
- Similarly, backward untraceability is maintained if $\mathcal{A}$ cannot determine whether a vehicle at time t_{brd} ($t_{brd} < t_{current}$) was previously involved in communication.

Definition 6: (Forward Privacy): Forward privacy is similar to untraceability with additional capability that one of two Pseudo-Identity (*PID*) messages and/or {vehicle location, battery status, selection of charge/discharge, expected time} messages information is given to $\mathcal{A}$. Clearly, now $\mathcal{A}$ can trace the vehicle's identity and/or other information. However, forward privacy is maintained if $\mathcal{A}$ is still unable to trace previous sessions (without giving a secret or session key).

Definition 7: (Vehicle Identity Anonymity): In the *V2G* network, anonymity is maintained if only the sender (vehicle) and the intended receiver (registration authority) can know the actual identity of the vehicle, *i.e.*, *EV(ID)* $\overset{PID}{\leftrightarrow}$ *CA/RA(ID)*.

Note that vehicle's (i) anonymity and (ii) untraceability guarantee that besides the vehicle and the registration authority $\in$ {home, centralized} *V2G* networks, no one including the aggregator $\in$ {home, visiting, centralized} *V2G* networks: (i) can figure out the identity of the vehicle, and (ii) is able to identify previous sessions involving that vehicle, respectively.

D. Our Contribution

We make the following main contributions by extending our work in [10] (considers only home *V2G* network) by including visiting and centralized *V2G* networks for charging/discharging. Our scheme:

- Provides mutual authentications between the *EV* and the *LAG* (in the home and visiting *V2G* networks), and between the *EV* and the *CA/RA* (in the home, visiting, and centralized *V2G* networks) so that no malicious entity can participate over the communicated network.
- Preserves privacy of the *EV*'s identity, location, charge/discharge selection, expected time, battery status, and other personal information in the home as well as visiting *V2G* networks. Each *EV*'s identity is well protected in all three networks. It also ensures that the *LAG* and adversary $\mathcal{A}$ cannot trace and extract information regarding *EV*'s behavior pattern.
- Generates lower communication overhead (by transmitting limited information) and computation overhead (by reducing the pairing, exponential, and scalar multiplication operations) than existing schemes in [16] and [7] in the home as well as centralized *V2G* networks. In the visiting *V2G* network, computation overhead of our scheme is also better than the scheme in [16], but is slightly large than the scheme in [7]. For subsequent authentications, our scheme achieves lower communication overhead than these schemes.
- Defeats various security attacks, such as *MITM*, replay, impersonation, redirection, known key, and repudiation attacks, and maintains information confidentiality, perfect forward secrecy, and message integrity. Our two-factor authentication scheme also defeats insider attacks, when a rogue device is installed in the network, and when a friend of the user tries to connect the vehicle to the *CS* for a misconduct on behalf of the user.
- Also maintains vehicle untraceability, forward privacy, vehicle identity anonymity, and information privacy.

This paper is organized as follows. Section II presents related work with *V2G* security and privacy issues. Section III discusses system and attack models in the distributed *V2G* network (home and visiting *V2G* networks) and centralized *V2G* network. Our authentication scheme under the distributed as well as centralized networks is presented in detail in section IV. The security and performance analysis of the proposed scheme is evaluated in section V. Finally, section VI concludes the work. Table I shows various symbols and acronyms used in the paper with their descriptions and sizes.

II. Related Work

Recently, many research works have been performed on authentication protocols/schemes for the *V2G* network [5], [8], [9], privacy preserving authentication [7] and threshold credit-based incentive mechanism [14], privacy-enhanced data

TABLE I: Symbols and Abbreviations

Symbol	Description	Size (bits)
EV	Electric vehicle	-
LAG	Local aggregator	-
CA/RA	Certification/registration authority	-
$H()$	One-way hash function	-
ID	Identity of the *EV*	128
PID	Pseudo-identity of the *EV*	128
CSID	Charging station identity	128
Γ	A public key at *CA/RA*	128
μ	A variable for a product of identities	256
λ	A random number generated by *CA/RA*	128
γ	A random number generated by *EV*	128
ξ	A variable for a product of identity	256
δ	Signature of the *EV*	128
r	A random number for key label	16
Option_request	A variable to store selected option	1
$Expected\_time$	Time duration for charging/discharging	64
Decision	Decision to conduct operation	1
k	Shared secret key between *EV* and *LAG*	128
H/hash	Hash value	64
T	Timestamp	64

aggregation [15], privacy preserving communication [16], and virtual ring architecture for smart grid privacy [17]. However, various possible attacks in the *V2G* network are still not well investigated. One of the main reasons that the *V2G* network is vulnerable to several security attacks due to enabling *IP*-based communications [18]. Also, due to the introduction of Internet of Things (*IoT*), where the inter-network traffic flow is allowed, security of the *V2G* network will become a critical issue [19], [20]. In order to keep the system protected against such attacks, various security and privacy requirements, such as authentication, secure key management, confidentiality, message integrity, anonymity, and untraceability need to be maintained [21], [22], [23].

A threshold anonymous authentication protocol for *VANET* is presented in [24], while a threshold anonymous announcement service using direct anonymous attestation and one-time anonymous authentication protocol is proposed in [25]. However, both protocols are not suitable for the *V2G* network due to the dynamic involvement of the *EV*s. Furthermore, an Energy Management Framework (*EMF*) is proposed in [26] to collect the real-time power consumption status and demand from the *EV*s and charging stations. However, the framework does not discuss its prevention against security attacks. An authentication scheme in [23] is not comparable to our scheme, as it neither considers important parameters in the *V2G* network, such as battery status and time to charge/discharge, nor discusses its prevention against security attacks. A study was performed for making a reservation on charging stations via *VANET* [27]. However, the drawback is to include a trusted authority that verifies the vehicle's identity. The user privacy may not be maintained in such a scenario, if the entity is malicious or compromised. Also the scheme only provides unidirectional authentication to verify the vehicles by the authority, which may lead to an impersonation attack. Further, a secure and privacy-aware fair billing framework is proposed for an online *EV* to move through charging plates installed under the road [28]. However, the scheme does not consider discharging of the vehicle. In addition, the idea of installing charging plates under the road is in the early phase and requires a huge setup cost.

A role-dependent privacy-preservation scheme (*ROPS*) [8] uses three *BV* roles, *i.e.*, energy demand, energy storage, and energy supply. Similarly, a battery status-aware authentication scheme [9] uses charging, fully charged, and discharging status of the battery. Further, an aggregated-proofs based privacy-preserving authentication scheme (*AP3A*) [7] achieves secure identification of the *BV* by verifying a group of *EV*s and establishing an aggregated-proof. However, all protocols/schemes generate huge overheads and do not entirely fit in the *V2G* network where a fast and efficient authentication is required.

A batch authentication protocol (*UBAPV2G*) [5] takes into account the vehicle communication in order to provide authentication in the *V2G* network. However, the scheme is just a variant of standard *DSA* algorithm and does not consider the important aspects in the *V2G* network, such as privacy-preservation of vehicle's sensitive information, prevention against various security attacks, and key management for secure communication. Privacy of the users and communication security of the smart grid are studied in [15] where a batch-oriented power-usage data aggregation scheme for the smart grid is proposed. However, the scheme discusses a generic adversary model without any security attack scenario, and does not provide mutual authentication. Further, a precise reward scheme for the *V2G* network [16] provides privacy protection by verifying the generated permit and rewarding the *BV*s later when they wish to disconnect. However, the scheme generates a large overhead. Moreover, to protect sensitive energy usage information of the user, a privacy protection scheme is proposed in [17]. However, the scheme is only for the smart grid, not directly applicable to the *V2G* network. In summary, with the best of our knowledge, one of the major limitations of the existing schemes/protocols is that they do not present security attacks scenarios in the *V2G* networks and most of them generate a huge overhead.

III. System, Security, and Privacy Models

This section presents an overview of our *V2G* system model, and the Strand Space model, and discusses security and privacy attacks model.

A. System Model

Consider a *V2G* system model as shown in Figure 1, which includes distributed as well as centralized *V2G* networks. In the distributed network, a vehicle can also move to visiting *V2G* network for charging/discharging. On the other hand, a vehicle can only discharge its battery in the centralized *V2G* network. Our system has three main entities: *EV*s, *LAG*s, and *CA/RA*s. An *EV* can charge/discharge its battery any time at any *CS*. An *LAG* is an entity located between the *CC* or *CA/RA* and the *CS*. We call them, *Home-LAG*, *Visiting-LAG*, and *Central-LAG*, respectively, in the home, visiting, and centralized *V2G* networks. A *CA/RA* is a trusted certification/registration authority

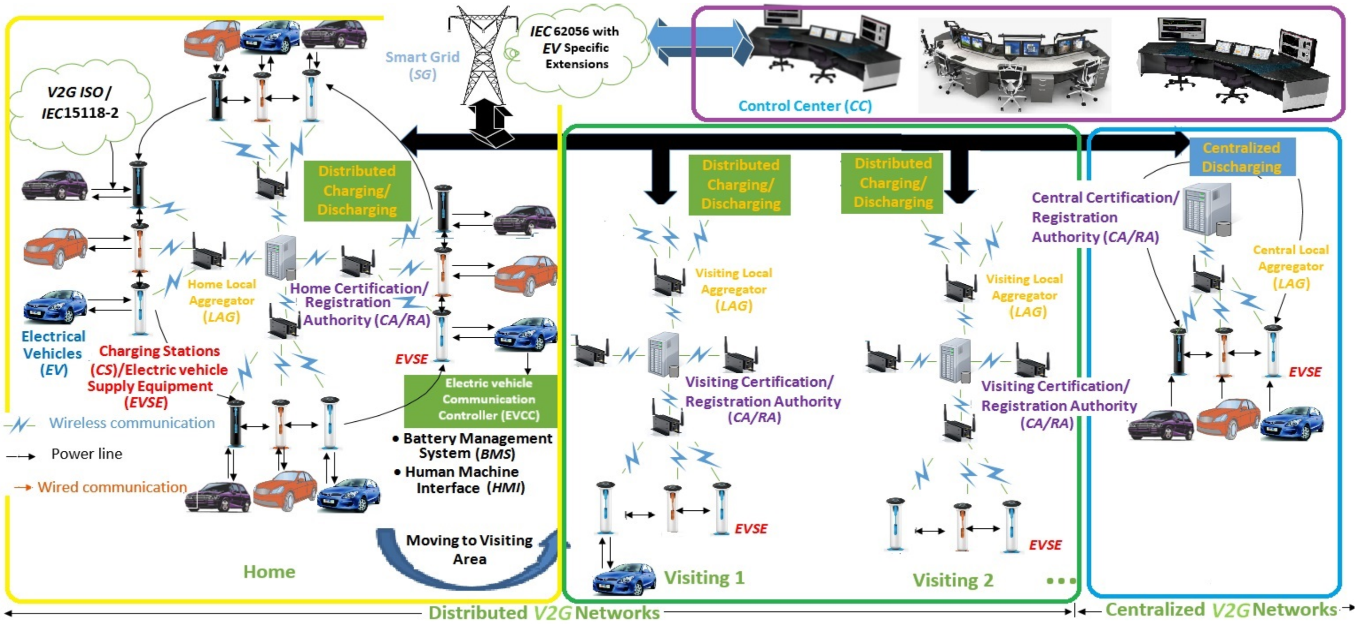


Fig. 1: *V2G* system model: a common architecture for the distributed *V2G* network (consisting of one home *V2G* network and multiple visiting *V2G* networks), and centralized *V2G* networks.

that maintains a database containing information of various *EV*s and *LAG*s. We call them *Home-CA/RA*, *Visiting-CA/RA*, and *Central-CA/RA* in the home, visiting, and centralized *V2G* networks, respectively. We assume that the *LAG* is curious about the *EV* related information. Each *EV* first registers itself to its *Home-CA/RA* by specifying the *Home-LAG* of its area.

An *EV* charges/discharges its battery from a group of charging stations connected to a *LAG*, and a number of *LAG*s are connected to a single *CA/RA* based on the capacity of *CA/RA* to handle *EV*s' requests. As an example, we consider that a single *Home/Visiting-CA/RA* is responsible for four *Home/Visiting-LAG*s in the home as well as visiting *V2G* networks. For the centralized *V2G* network, one *Central-LAG* is connected to the *Central-CA/RA* as shown in Figure 1, but in reality, there can be a number of *Central-LAG*s connected to the *Central-CA/RA*. All the *CA/RA* are further connected to the *CC* through wireless/wired communication. The communications between the *CS* and a *LAG*, and between a *LAG* and a *CA/RA* are enabled through wireless networks. The communication between the *EVSE* and the Electric Vehicle Communication Controller (*EVCC*), and between the *EVSE* and the load balance controller at *CC* are governed by *ISO/IEC 15118-2* [29] and *IEC62056* with *EV* extensions [30], respectively.

B. Adversary Model: Security and Privacy Attacks

The strength of adversary $\mathcal{A}$ is defined by the set of oracles that it can access and is allowed to query. A weak adversary never corrupts the message, while a destructive adversary may corrupt the message at any time. In addition, a strong adversary may corrupt the message at any time without destroying the message. We consider a strong adversary in our attack model. In our attack model, an outsider attacker may perform *MITM attack* between an *EV* and a *LAG*. The attack is successful if $\mathcal{A}$ retrieves message information using $Enc_k(.)/Dec_k(.)$ over the unencrypted or weak encrypted network. $\mathcal{A}$ can also perform a *replay attack* if it delays ($T_{receive} = T_{send} + T_{prop\_time} + T_{attack}$) or repeats the transmitted message ($msg \in OLD\_MSG$) to the *EV*/*LAG* over the network, where $T_{current} > T_{old\_msg}$. $\mathcal{A}$ can perform *integrity violations* if it modifies the transmitted messages over the network, such that $msg_{receive} \neq msg_{send}$. $\mathcal{A}$ can also initiate an *impersonation attack* if it pretends itself as one of the *EV*s/*LAG*s involved in the communication, such that *EV* = $\mathcal{A}$–*EV* or *LAG* = $\mathcal{A}$–*LAG*, and *EV* $\overset{k}{\Leftrightarrow}$ $\mathcal{A}$–*LAG* or $\mathcal{A}$–*LAG* $\overset{k}{\Leftrightarrow}$ *EV*. In addition, $\mathcal{A}$ can execute *repudiation attack*, in which acting as an *EV* it denies after sending a message (*msg*) to the *LAG* such that the *EV* either owns a valid message (*msg*) proof received by *LAG*, or the *LAG* has received *msg* by the *EV* and owns a valid proof. Also, $\mathcal{A}$ may generate future session keys based on the current session key ($key_{future} = f(key_{current})$), resulting in a *known key attack*.

In our privacy model, original identity of *EV*s must be protected. Otherwise, $\mathcal{A}$ (or even the *LAG*) can extract personal information of the users by tracing their behaviors. In a computational environment, privacy properties are typically defined by means of games. We consider untraceability, forward privacy, and anonymity properties in our privacy model. We assume that $\mathcal{A}$ can eavesdrop communications and can also query all the messages in the beginning. Then, $\mathcal{A}$ chooses a message *msg* randomly from a set and makes a query. $\mathcal{A}$

can break untraceability if it can detect the selected message *msg* with probability $\{\Pr < \text{guess}_{random}\}$. Furthermore, $\mathcal{A}$ may perform backward and forward untraceability, in which the internal state of the *EV* (such as identity) is known to $\mathcal{A}$, and based on the information derived in current session, it can determine whether a particular *EV* was involved in the past and future subsequent communications, respectively. Moreover, $\mathcal{A}$ can also break forward privacy and anonymity by tracing the previous sessions and actual identity of the *EV*, respectively.

C. Strand Space Model for Protocol Security

We define the protocol as a sequence of events for each role of the *EV*, *LAG*, and *CA/RA* using Strand Space model [31]. A strand s represents a sequence of actions of an instance of a role. $\mathbb{A}$ is a set of the elements $terms$, which are possible messages that can be exchanged between the *EV*, *LAG*, and *CA* in the protocol. A strand space is a set Σ with a trace mapping $tr : \Sigma \rightarrow (\pm\mathbb{A})^*$, where $\langle send, a\rangle$ and $\langle receive, a\rangle$ are signed terms as $\langle +a\rangle$ and $\langle -a\rangle$, respectively, and $(\pm\mathbb{A})^*$ is a set of finite sequence of signed terms. Furthermore, a bundle $C = (N_C, E)$, which is a subgraph of N, represents the protocol execution under some configuration, where $E \subseteq (\rightarrow \cup \Rightarrow)$ is a set of the edges and $N_C \subseteq N$ is a set of nodes incident with the edges in E. A node is a pair $\langle s, i\rangle$ with $s \in \Sigma$ and i an integer satisfying $length(tr(s))$, and is denoted as $n \in s$. Also, assume $T \in \mathbb{A}$ is the set of atomic messages, $m \in M$ is a *Text* term, $k \in K$ is a *Key* term, inverse of symmetric key k is k^{-1}, and $\mathcal{K}_\mathcal{A}$ is a key space of the keys known to adversary $\mathcal{A}$.

IV. Proposed Authentication Scheme

This section presents a preliminary discussion on bilinear pairing as well as proposes an authentication scheme in the home, visiting, and centralized *V2G* networks. We assume that each *EV* has a tamper-proof device that is responsible for all cryptographic-related computations, such as storage of secret keys and algorithms, generation of Pseudo-Identities (*PID*s) of *EV*s, and encrypting and signing the messages.

A. Preliminaries

Preliminaries include our bilinear pairing technique and dynamic accumulator.

1) Bilinear Pairing: We define the bilinear pairing of our system as follows:

Definition 8: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic multiplicative groups of prime order p generated by g_1 and g_2 for which there exists an isomorphism $\varphi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\varphi(g_2) = g_1$. Consider $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$. Let $\mathbb{G}_T$ be a cyclic multiplicative group with the same order p where e: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear pairing with the following properties:
Properties: (i) Bilinearity: $e(P^a, Q^b) = e(P, Q)^{ab}$, $\forall P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ and $a,b \in \mathbb{Z}_p^*$.
(ii) Non-degeneracy: $e(g_1, g_2) \neq 1$
(iii) Computability: There exists an efficient algorithm to compute $e(P, Q)$, $\forall P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$.

Domain of hash functions are as follows: $H_1 : \mathbb{G}_1 \times \{0,1\}^* \times \mathbb{Z}_p^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_T \times \{0,1\}^* \times \mathbb{Z}_p^* \rightarrow \mathbb{G}_T$, $H_3 = H(f_1) = H(f_2) : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, $H_4 = H(Q^S) : \mathbb{Z}_p^* \times \mathbb{G}_2 \rightarrow \mathbb{G}_2$. Various input parameters of each hash function (including integer modulo prime p and elliptic group) are converted in bit-string, and then it produces 256-bit output by *SHA256* [32]. Further, we define a bilinear pairing instance generator that takes a security parameter l as input and returns a uniformly random tuple $t = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ of bilinear pairing parameters such that p grows exponentially with l.

2) Accumulator from Bilinear Pairing: An accumulator is a one-way function that verifies whether a candidate is a member of a given set without revealing the identity of other members in a set. We define a dynamic accumulator for our system. Let $\mathbb{N}$ be the set of positive integers.

Definition 9: An accumulator is a tuple $(\{X_l\}_{l\in\mathbb{N}}, \{F_l\}_{l\in\mathbb{N}})$, where $\{X_l\}_{l\in\mathbb{N}}$ is the value domain of the accumulator, and $\{F_l\}_{l\in\mathbb{N}}$ is a sequence of the families of pairs of functions such that each $(f, g) \in F_l$ is defined as $f : U_f \times X_f^{ext} \rightarrow U_f$ for some $X_f^{ext} \supseteq X_l$, and $g : U_f \rightarrow U_g$ is a bijective function [33], [34]. The following properties are satisfied:
Properties: (i) Efficient Generation: There exists an efficient algorithm that takes a security parameter l as input and outputs a random element $(f, g) \in_R F_l$ with auxiliary information β.
(ii) Quasi Commutativity: For every $l \in \mathbb{N}$, $(x_1, x_2) \in X_l$, $u \in U_f$: $f(f(u, x_1), x_2) = f(f(u, x_2), x_1)$. The $g(f(u, X))$ is computable in polynomial time in l, even without the knowledge of β, where $X = \{x_1, ..., x_q\} \subset X_l$.

B. Our Scheme in the Home V2G Network

We present the details of our scheme including initial setup, *EV* registration, and scheme execution, as shown in Figure 2. In our scheme, a dynamic accumulator is used by the *Home-LAG* and the *Home-CA/RA* in order to verify whether an *EV* belongs to a set of all registered *EV*s at that point of time. Further, a bilinear pairing map is used to generate a shared secret key between the *EV* and the *Home-LAG*. This key is used for all subsequent authentications within a session. In addition, a hash of signatures are used to provide non-repudiation and confidentiality of the transmitted messages.

1) ***Initial Setup*****:** All *EV*s, *LAG*s, and *CA/RA* (in all three networks) randomly generate their private keys as S_{EV}, S_{LAG}, $S_{CA} \in_R \mathbb{Z}_p^*$, and further compute their public keys as $Q_{EV} = g_2^{S_{EV}}$, $Q_{LAG} = g_2^{S_{LAG}}$, and $Q_{CA} = g_2^{S_{CA}}$, respectively, where $g_2 \in \mathbb{G}_2$. These public keys are stored in an off-line key repository. Further, we define $(f, g) \in F_l$ as $g(f(g_2, PID))$ where $PID = \{PID_1, PID_2, ..., PID_q\}$ is a set of pseudo-identities of the *EV*s. Consider $f : \mathbb{Z}_p \times \mathbb{G}_2 \rightarrow \mathbb{G}_2$, $g : \mathbb{G}_2 \rightarrow \mathbb{Z}_p$, $f : (g_2, PID) \mapsto PID.H(\sigma_{CA\text{-}LAG})$, $g : g_2 \mapsto g_2/H(\sigma_{LAG\text{-}CA})$, where signature $\sigma_{CA\text{-}LAG}$ is computed at *CA/RA* as $(Q_{LAG})^{S_{CA}}$, while $\sigma_{LAG\text{-}CA}$ is computed at *LAG* as $(Q_{CA})^{S_{LAG}}$.

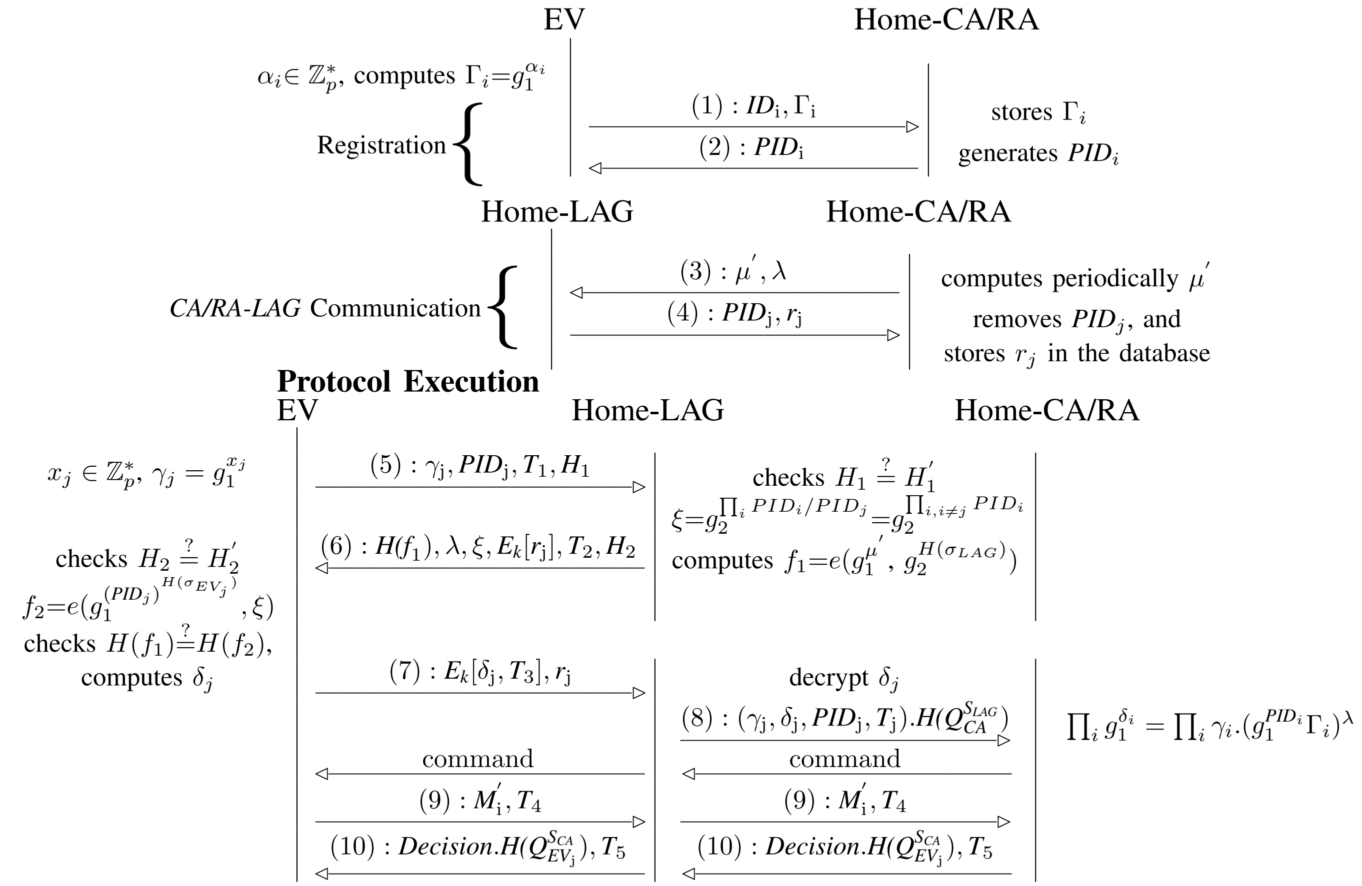


Fig. 2: Proposed scheme for the *V2G* smart grid network.

*2) **EV Registration**:* First of all, each EV_i has to register itself with the *Home-CA/RA* of its home area. This registration can be done either by physically reaching *Home-CA/RA* or remotely using a pre-shared login credentials. Each EV_i generates a random secret $\alpha_i \in_R \mathbb{Z}_p^*$ and computes $\Gamma_i = g_1^{\alpha_i} \in \mathbb{G}_1$. Thereafter, the EV_i submits its original identity ID_i to the *Home-CA/RA* along with Γ_i. This α_i is used by the EV_i for its signature generation during its request to the *Home-LAG*.

Msg-(1): $EV_i \rightarrow$ Home-CA/RA: $\{ID_i, \Gamma_i\}$

The *Home-CA/RA* stores its ID_i, generates a pseudo-identity of the EV_i, *i.e.*, PID_i, using a pseudo-random function [35], and sends it to the EV_i.

Msg-(2): Home-CA/RA $\rightarrow EV_i$: $\{PID_i\}$

After each successful registration of a new EV_j, the *Home-CA/RA* computes $\mu = (\prod_{i\neq j} PID_i).H(\sigma_{CA\text{-}LAG}).PID_j$, where $\prod PID_i$ is the product of all *PID*s of registered *EV*s. Similarly, once the session expires for an EV_i, its PID_i is removed from the database at *Home-CA/RA* and then the *Home-CA/RA* recomputes μ. Hence, registration process creates a dynamic accumulator that supports efficient evaluation, efficient addition, and efficient deletion of an EV_i. We define our dynamic collision resistant accumulator with the following properties:

Definition 10: EV's Evaluation: Consider a set of pseudo-identities *PID*s of various registered *EV*s as $\{PID_1, PID_2, ..., PID_i\}$. The *Home-CA/RA* computes $\mu = \prod_i PID_i$ that maps $g(f(g_2, PID))$ as $\prod_i PID_i$.

Definition 11: EV's Addition: The *Home-CA/RA* computes $\mu = g(f(g_2, PID))$ considering $PID_i \in PID$, $PID_j \notin PID$, and $g(f(g^{-1}(\xi), PID_i)) = \mu$. When a new EV_j is registered, the updated μ is computed as $\mu' = g(f(g_2, PID \bigcup \{PID_j\}))$ $= \mu.PID_j$. Here, the value ξ' is such that $\mu' = g(f(g^{-1}(\xi'), PID_i))$ where $\xi' = \xi.PID_j$. The ξ is a witness for the fact that $PID_i \in PID$ has been accumulated in μ whenever $g(f(g^{-1}(\xi), PID_i)) = \mu$.

Definition 12: EV's Deletion: The *Home-CA/RA* computes $\mu = g(f(g_2, PID))$ considering PID_i, $PID_j \in PID$, $PID_i \neq PID_j$, and $g(f(g^{-1}(\xi), PID_i)) = \mu$. After performing operations by the EV_j within a session, its PID_j must be deleted and the updated μ is computed as $\mu' = g(f(g_2, PID \backslash \{PID_j\})) = \mu/PID_j$. Here, the value ξ' is such that $\mu' = g(f(g^{-1}(\xi'), PID_i))$, where $\xi' = \xi/PID_j$.

*3) **Home-CA/RA and Home-LAG Communication**:* Whenever a new EV_j is registered at *Home-CA/RA*, the *Home-CA/RA* updates the *Home-LAG* by transmitting updated μ, *i.e.*,

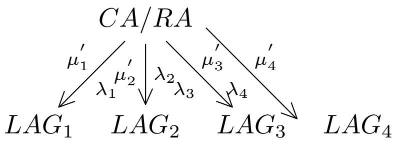


Fig. 3: The *Home-CA/RA* periodically transmits $\mu^{'}$ and λ to different *LAG*s associated with it.

$\mu^{'}$ as $\mu^{'}.H(\sigma_{CA\text{-}LAG})$ where a signature $\sigma_{CA\text{-}LAG} = (Q_{LAG})^{S_{CA}}$. The *Home-CA/RA* also generates a random $\lambda \in \mathbb{Z}_p^*$ for each associated *LAG* and sends it (only first time) to the respective *LAG*. On receiving, the *Home-LAG* extracts $\mu^{'}$ using its signature's hash as $\mu^{'}/H(\sigma_{LAG\text{-}CA})$, where $\sigma_{LAG\text{-}CA} = (Q_{CA})^{S_{LAG}}$.

Msg-(3): Home-CA/RA $\rightarrow$ *Home-LAG:* $\{\mu^{'}, \lambda\}$

As shown in Figure 3, the *Home-CA/RA* sends a unique $\lambda_i \in \mathbb{Z}_p^*$ to each associated *LAG* along with updated $\mu^{'}$ of the registered *EV*s served by the respective *LAG*s. During first authentication, a shared secret key k is generated at EV_j and *Home-LAG*. For all the subsequent authentication requests, the EV_j sends $E_k[PID_j, T_j]$ to the *Home-LAG* as message (7) in our scheme (discussed in the next subsection). After expiry of session time, the *Home-LAG* discards session key k and sends corresponding PID_j and recently received r_j to the *Home-CA/RA*. On receiving, the *Home-CA/RA* deletes PID_j of the EV_j from its database and stores r_j to the database.

Msg-(4): Home-LAG $\rightarrow$ *Home-CA/RA:* $\{PID_j,\ r_j\}$

4) ***Scheme Execution*****:** Whenever an EV_j wishes to charge/discharge its vehicle's battery, it generates a random $x_j \in_R \mathbb{Z}_p^*$ and computes $\gamma_j = g_1^{x_j} \in \mathbb{G}_1$. Thereafter, the EV_j sends γ_j to the *Home-LAG* along with its PID_j, a timestamp T_1, and a hash value $H_1 = H(\gamma_j, T_1, PID_j)$.

Msg-(5): $EV_j \rightarrow$ *Home-LAG:*$\{\gamma_j, PID_j, T_1, H_1\}$

On receiving message (5), the *Home-LAG* verifies $H_1 \stackrel{?}{=} H_1^{'}$ and extracts $\mu^{'}$ as $\mu^{'}.H(\sigma_{CA\text{-}LAG})/H(\sigma_{LAG\text{-}CA})$, where signature $\sigma_{CA\text{-}LAG} = (Q_{LAG})^{S_{CA}}$ and $\sigma_{LAG\text{-}CA} = (Q_{CA})^{S_{LAG}}$. Thereafter, the *Home-LAG* computes ξ as $\xi = g_2^{(\prod_i PID_i)/PID_j}$. It is worth to note that PID_i also includes PID_j, as it is a registered *EV*. Hence, $\xi = g_2^{\prod_{i,i \neq j} PID_i}$, which ensures that the EV_j belongs to $\mu^{'}$ and thereby the EV_j is authenticated by the *Home-LAG*. This process can be achieved in a batch of multiple *EV*s that send their *PID*s to the respective *LAG*. Next, the *Home-LAG* computes f_1 as $f_1 = e(g_1^{\mu^{'}}, g_2^{H(\sigma_{LAG})})$ and sends $(H(f_1), \lambda, \xi, T_2, H_2)$ to the EV_j where λ was received from the *Home-CA/RA*, $H(\sigma_{LAG}) = H(Q_{EV_j}^{S_{LAG}})$, $r_j \in \mathbb{Z}_p^*$ is a random number, and $H_2 = H(f_1, \lambda, \xi, r_j, T_2)$.

Msg-(6): Home-LAG $\rightarrow EV_j : \{H(f_1), \lambda, \xi, E_k[r_j], T_2, H_2\}$

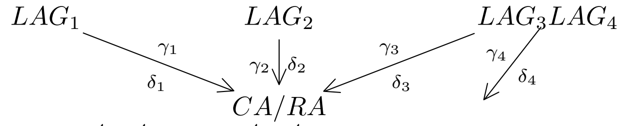
On receiving message (6), the EV_j verifies $H_2 \stackrel{?}{=} H_2^{'}$, computes f_2 as $f_2 = e(g_1^{(PID_j)^{H(\sigma_{EV_j})}}, \xi)$ where $H(\sigma_{EV_j}) = H(Q_{LAG}^{S_{EV_j}})$, and checks whether $H(f_1) \stackrel{?}{=} H(f_2)$. If $f_1 = f_2 = k$ (shared secret key) holds, the *Home-LAG* is authenticated by the EV_j. The r_j is associated with session key k at *Home-LAG* and this key can be used for further communications within a session. The *Home-LAG* keeps PID_j in its database until the expiry (session time) of the key k. Further, the EV_j computes $\delta_j = x_j + \lambda(\alpha_j + PID_j)$ and sends it to the *Home-LAG*.

Msg-(7): $EV_j \rightarrow$ *Home-LAG:*$\{E_k[\delta_j, T_3], r_j\}$

After receiving message (7), the *Home-LAG* sends $(\gamma_j, \delta_j, PID_j)$ to the *Home-CA/RA* signed by $H(Q_{CA}^{S_{LAG}})$.

Msg-(8): LAG $\rightarrow$ *CA/RA:*$\{(\gamma_j, \delta_j, PID_j, T_j).H(Q_{CA}^{S_{LAG}})\}$

Message (8) may contain information of multiple EV_i associated with that *Home-LAG*. It may also be the case where different *LAG*s send message (8) simultaneously (or in a very short time) to the *Home-CA/RA*. Hence, it is recommended that the *Home-CA/RA* authenticates these requests in a batch for better efficiency. First, the *Home-CA/RA* separates out requests that belong to each *LAG* using λ and $H(Q_{LAG}^{S_{CA}})$, and then verifies all the *EV*s in a batch corresponding to each *LAG* by verifying $\prod_i g_1^{\delta_i} = \prod_i \gamma_i.(g_1^{PID_i}\Gamma_i)^{\lambda}$.



$$\sum_{i=1}^{4}\prod_{j=1}^{4} g_1^{\delta_j} = \sum_{i=1}^{4}\prod_{j=1}^{4} \gamma_j.(g_1^{PID_i}\Gamma_j)^{\lambda_i}$$

Fig. 4: Verification of the *EV*s at *Home-CA/RA* by the information received from different *LAG*s.

If it holds, all EV_i are successfully verified. Otherwise, one or more EV_i are invalid. In such a case, invalid requests need to be located and removed from a batch. Then, a re-batch verification is performed. The detection of invalid requests can be performed using a divide and conquer approach described in [36]. Similarly, different *LAG*s connected to a *Home-CA/RA* send the received EV_i's information to the respective *Home-CA/RA*, and the *Home-CA/RA* verifies all the requests

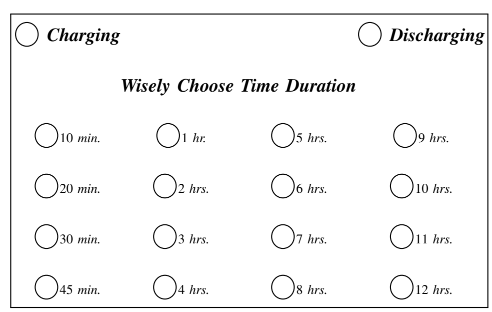


Fig. 5: Charging and discharging time selection window.

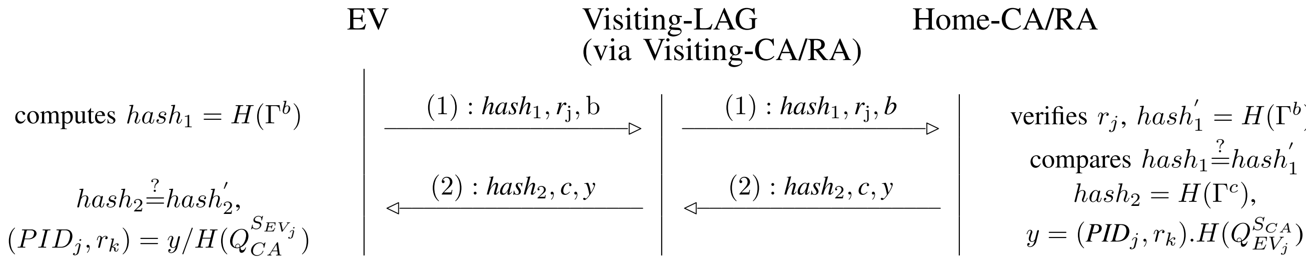


Fig. 6: Pre-phase of our scheme for vehicle mobility in the *V2G* visiting area network via *Visiting-LAG* and *Visiting-CA/RA*.

as shown in Figure 4. After successful authentication, the *Home-CA/RA* sends a command that opens a window for all EV_i to select charging/discharging duration as illustrated in Figure 5. The smart grid's *CC* computes power supply and demand load based on the operation selected by all EV_i. All the EV_i preferences are captured in a message M_i where $M_i = (PID_i, CSID, Option\_request, Expected\_time)$. Here, *Option_request* has two options, one is request for charging and other is request for discharging. Each EV_i computes M_i' as $M_i.H(Q_{CA}^{S_{EV_i}})$ and sends it to the *Home-CA/RA*.

Msg-(9): $EV_i \rightarrow$ Home-CA/RA:$\{M_i', T_4\}$

On receiving message M_i', the *Home-CA/RA* retrieves the original message M_i from the received message as $M_i = M_i'/H(Q_{EV_i}^{S_{CA}})$. The *Home-CA/RA* sends a One Time Password (*OTP*) to the EV_i for its identity verification. Thereafter, the *Home-CA/RA* asks to the *CC* to compute the power based on charging/discharging request by the EV_i. Further, it computes dynamic power load and announces its decision of allowing charging/discharging decision, *i.e.*, $Decision$, to the EV_i.

Msg-(10): Home-CA/RA $\rightarrow EV_i$: $\{Decision.H(Q_{EV_i}^{S_{CA}}), T_5\}$

Finally, the EV_i performs required operation based on the decision received from the *Home-CA/RA* as $Decision/H(Q_{CA}^{S_{EV_i}})$. We have shown $Decision.H(Q_{EV_j}^{S_{CA}})$ for EV_j in Figure 2. After completion of the desired operation by the EV_i, the *Home-CA/RA* sends required information to the *CC* for billing purposes.

For all subsequent requests within a valid session of the key k, the EV_j sends message (7) as $\{E_k[PID_j, T_j], r_j\}$ to the *Home-LAG*. On receiving the message, the *Home-LAG* decrypts the message using a session secret key k identified by r_j and verifies PID_j. If it is valid within a session, the *LAG* sends a verification command with $H(Q_{CA}^{S_{LAG}})$ to the *Home-CA/RA*. In addition, the *Home-LAG* sends a new random $r_j^{'}$ as $E_k[r_j^{'}]$ to the respective EV_j. The EV_j sends next authentication request along with this number so that the *Home-LAG* can extract respective session key of the EV_j. The *Home-CA/RA* extracts verification command using $H(Q_{LAG}^{S_{CA}})$ and sends a command to open a selection window for the EV_j. Thereafter, the scheme executes message (9) and message (10) as it is. After session expiration of the key k, the *Home-LAG* discards k and sends its related PID_j and r_j to the *Home-CA/RA*, which then removes PID_j and stores r_j to the database.

C. Our Scheme in the Visiting V2G Network

In a more realistic *V2G* scenario, the vehicle may also move outside of its registered home *V2G* network to a visiting *V2G* network. Hence, we extended our scheme by considering the visiting *V2G* network. As shown in Figure 6, the vehicle has to execute a pre-phase before being mutually authenticated with the *Visiting-LAG*. In detail, mutual authentications between the *EV* and the *Visiting-LAG*, and between the *EV* and the *Visiting-CA/RA* are achieved by carrying out the following steps:

Step-1: Pre-phase: First, the EV_j selects a random number $b \in_R \mathbb{Z}_q^*$ and computes a hash $hash_1 = H(\Gamma^b)$. Thereafter, it sends $hash_1, r_j$, b, and its *Home-CA/RA* to the *Visiting-LAG*, which is then transmitted to the *Visiting-CA/RA* in a secure manner using their public and private keys. This exposes the identity of *Home-CA/RA* to which the EV_j belongs to. However, it is still very difficult to recognize the actual identity of the EV_j. Furthermore, the *Visiting-CA/RA* transmits $hash_1, r_j$, and b to the respective *Home-CA/RA*. Here, r_j is the latest number stored in the database with respect to corresponding EV_j (message (4) in Figure 2) that helps the *Home-CA/RA* to retrieve the *EV*'s identity and public key Q_{EV_j}.

Msg-(1): $EV_j \rightarrow$ Home-CA/RA:$\{hash_1, r_j, b\}$
On receiving message (1), the *Home-CA/RA* verifies the received r_j, retrieves Γ_j, computes $hash_1^{'} = H(\Gamma^b)$, and compares $hash_1 \stackrel{?}{=} hash_1^{'}$. If it is true, the *Home-CA/RA* computes $hash_2 = H(\Gamma^c)$ and $y = (PID_j, r_k).H(Q_{EV_j}^{S_{CA}})$, where $c, r_k \in_R \mathbb{Z}_q^*$ and PID_j is a new random pseudo-identity generated by the *Home-CA/RA*. Finally, it sends $hash_2, c$, and y to the EV_j.

Msg-(2): Home-CA/RA $\rightarrow EV_j$: $\{hash_2, c, y\}$
When message (2) is received by the EV_j, it computes $hash_2^{'} = H(\Gamma^c)$ and compares $hash_2 \stackrel{?}{=} hash_2^{'}$. If it is true, the EV_j retrieves PID_j and r_k from y as $PID_j = y/H(Q_{CA}^{S_{EV_j}})$.

Step-2: The *Home-CA/RA* sends PID_j and $\Gamma_k = g_1^{r_k}$ to the *Visiting-CA/RA*. Thereafter, the *Visiting-CA/RA* sends updated $\mu^{'} = \prod_i PID_i$ to the *Visiting-LAG*, where PID_i includes PID_j.

Step-3: Once PID_j is received by the EV_j, the scheme continues from message (5) in Figure 2. Following points highlight differences with the home *V2G* network scheme:
1. The public and private keys of the *Visiting-LAG* are used (instead of *Home-LAG*) for computing signatures σ_{LAG} and

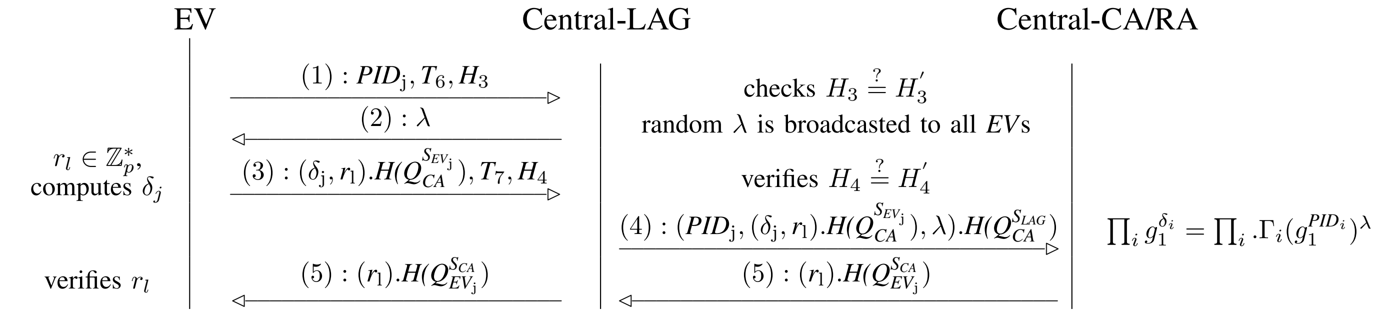


Fig. 7: Proposed scheme for *EV*'s discharging in the *V2G* centralized network.

σ_{EV_j} between the *Visiting-LAG* and the EV_j.
2. The EV_j computes $\delta_j = x_j + \lambda(r_k + PID_j)$ in message (7).
3. The public and private keys of the *Visiting-CA/RA* are used (instead of *Home-CA/RA*) for computing $Q_{CA}^{S_{LAG}}$ in message (8), and $Q_{CA}^{S_{EV_j}}$ and $Q_{EV_j}^{S_{CA}}$ in messages (9) and message (10).
4. The *Visiting-CA/RA* uses Γ_k received from the *Home-CA/RA* for the respective EV_j by verifying all EV_i in a batch.
5. When the scheme run is over, the *Visiting-LAG* sends recent r_j to the *Home-CA/RA* via *Visiting-CA/RA* to store it.

D. Our Scheme in the Centralized V2G Network

We consider that a centralized *V2G* network covers multiple geographical locations, and the *EV*s can discharge their batteries to the grid, but cannot charge from it. Having such a centralized *V2G* network enables transmission of power directly and immediately to the smart grid. In a later time, this stored power can be supplied to other locations where power is urgently needed. In order to attract the *EV*s for discharging their batteries, a reward scheme can be considered. In fact, an *EV* can be paid relatively better while discharging via centralized *V2G* network rather than via distributed *V2G* network. Since only discharging can be performed in the centralized *V2G* network, the *EV*s can be paid on the spot without generating their bills at later stage. Mutual authentication between the *EV* and the *Home-CA/RA* via *Central-CA/RA*, and retrieving PID_j and r_k are achieved through a similar process described in Figure 6. Furthermore, a session key does not need to be provided to the *EV*s for discharging their batteries. This will save time and power of the system when connecting and disconnecting vehicles frequently. The verification of all EV_i signatures are performed in a batch.

As shown in Figure 7, the *EV*'s discharging scheme under the centralized *V2G* network executes the following steps:

Step-1: After receiving PID_j from the *Home-CA/RA*, the EV_j sends its PID_j, T_6, and H_3 to the *Central-LAG*, where $H_3 = H(PID_j, T_6)$.

Msg-(1): EV_j → Central-LAG:$\{PID_j, T_6, H_3\}$

Step-2: On receiving message (1), the *Central-LAG* computes $H_3^{'}$ and checks whether $H_3 \stackrel{?}{=} H_3^{'}$. If it is true, it broadcasts a random number $\lambda \in \mathbb{Z}_p^*$ to all *EV*s that are requesting for discharge within a very short period of time so that all requests can be verified in a batch at *Central-CA/RA*.

Msg-(2): Central-LAG $\rightarrow EV_j : \{\lambda\}$

Step-3: After receiving λ, each EV_j generates its signature δ_j as $\delta_j = r_k + \lambda.PID_j$ and a random number $r_l \in \mathbb{Z}_p^*$. Thereafter, the EV_j sends $(\delta_j, r_l).H(Q_{CA}^{S_{EV_j}})$, T_7, and H_4 to the *Central-LAG*, where $H_4 = H((\delta_j, r_l).H(Q_{CA}^{S_{EV_j}}), T_7)$.

Msg-(3): EV_j → Central-LAG:$\{(\delta_j, r_l).H(Q_{CA}^{S_{EV_j}}), T_7, H_4\}$

Step-4: The *Central-LAG* computes and compares $H_4 \stackrel{?}{=} H_4^{'}$. If it is true, it sends $((\delta_j, r_l).H(Q_{CA}^{S_{EV_j}}), PID_j, \lambda).H(Q_{CA}^{S_{LAG}})$ to the *Central-CA/RA*.

Msg-(4): Central-LAG → Central-CA/RA: $\{(PID_j, \lambda, (\delta_j, r_l).\ H(Q_{CA}^{S_{EV_j}})).H(Q_{CA}^{S_{LAG}})\}$

Step-5: On receiving message (4), the *Central-CA/RA* computes and compares $\prod_i g_1^{\delta_i} = \prod_i \Gamma_i.(g_1^{PID_i})^{\lambda}$. Here, Γ_i includes $\Gamma_k = g_1^{r_k}$ that was sent from the *Home-CA/RA* to the *Central-CA/RA* for each EV_j.

Step-6: The *CA/RA* sends an acknowledgment message to each authenticated EV_j as $(r_l).H(Q_{EV_j}^{S_{CA}})$.

Msg-(5): Central-CA/RA $\rightarrow EV_j : \{(r_l).H(Q_{EV_j}^{S_{CA}})\}$

On receiving, each EV_j retrieves and verifies r_j by computing $H(Q_{CA}^{S_{EV_j}})$. Once it is verified, the EV_j starts discharging its battery to the smart grid. After process completion, the *Central-CA/RA* sends r_l to the *Home-CA/RA* to store it.

V. Security and Performance Analysis

This section presents computation proofs, as well as security and performance analysis of our scheme.

A. Computation Proofs

Theorem 1. The proposed scheme in the home and visiting *V2G* networks generates a shared secret key k between the EV_j and the *Home/Visiting-LAG*.
Proof: *Generation of a shared secret key k:*

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIFS.2016.2532840

Key k_{LAG} at *Home/Visiting-LAG:* $f_1 = e(g_1^{\mu'}, g_2^{H(\sigma_{LAG})})$,
$=e(g_1^{\prod_i PID_i}, g_2^{H(\sigma_{LAG})})$
$=e(g_1, g_2^{H(\sigma_{LAG})})^{\prod_i PID_i}$
Key k_{EV_j} at EV_j: $f_2 = e((g_1)^{PID_j^{H(\sigma_{EV_j})}}, \xi)$,
where $H(\sigma_{EV_j}) = H(Q_{LAG}^{S_{EV_j}})$
$=e((g_1)^{PID_j^{H(\sigma_{EV_j})}}, g_2^{\prod_{i,i\neq j} PID_i})$
$=e(g_1, g_2^{H(\sigma_{EV_j})})^{\prod_i PID_i}$,
since $e(P^a, Q)$=$e(P, Q^a)$=$e(P, Q)^a$, and $H(\sigma_{LAG})$=$H(\sigma_{EV_j})$.

In a similar way, the other *EV*s can generate a shared secret key with their respective *Home/Visiting-LAG*.

Theorem 2. If all the requests are made by the legitimate *EV*s to the respective *LAG*, the *CA/RA* verifies all the requests correctly. The *LAG* refers all types of local aggregators, *i.e.*, *Home-LAG*, *Visiting-LAG*, and *Central-LAG*.
Proof: *Batch verification at Home/Visiting-CA/RA in the home and visiting V2G networks:*
R.H.S. = $\prod_i (g_1^{PID_i})^{\lambda}(\gamma_i)(\Gamma_i)^{\lambda}$ = $\prod_i (g_1^{\lambda PID_i})(g_1^{x_i})(g_1^{\alpha_i})^{\lambda}$
= $\prod_i g_1^{x_i+\lambda(\alpha_i+PID_i)}$
L.H.S.= $\prod_i g_1^{\delta_i}$ = $\prod_i g_1^{x_i+\lambda(\alpha_i+PID_i)}$
Hence, $\prod_i g_1^{\delta_i} = \prod_i \gamma_i.(g_1^{PID_i}\Gamma_i)^{\lambda}$ is true. The proof for $\prod_i g_1^{\delta_i} = \prod_i \Gamma_i.(g_1^{PID_i})^{\lambda}$ batch verification at *Central-CA/RA* in the centralized *V2G* network can be similarly derived.

B. Security Analysis

In this subsection, authentication, session key establishment, and privacy-preservation of the proposed scheme are discussed along with prevention against different security attacks.

Property 1. *The proposed scheme provides mutual authentications between the EV_j and the LAG in the home and visiting V2G networks. It is also provided between the EV_j and the Central-CA/RA in the centralized V2G network.*

In the home and visiting *V2G* networks, the *LAG* authenticates the EV_j by verifying $\xi = g_2^{(\prod_i PID_i)/PID_j}$, and each EV_j authenticates the *LAG* by comparing $H(f_1) \stackrel{?}{=} H(f_2)$. Further, the original message M can only be extracted by the *CA/RA* with Q_{EV_j} and S_{CA}. Similarly, *Decision* can only be retrieved by the EV_j with S_{EV_j} and Q_{CA}. In the centralized *V2G* network, *Central-CA/RA* authenticates the EV_j by verifying its signature. Also, the EV_j verifies r_l and confirms authentication with the *Central-CA/RA*. In other words,

$$\forall C.LAG(\overrightarrow{x}) \in C \Rightarrow EV_j(\overrightarrow{x}) \in C, \text{and}$$
$$\forall C.CA(\overrightarrow{y}) \in C \Rightarrow EV_j(\overrightarrow{y}) \in C,$$

where $\overrightarrow{x}$ and $\overrightarrow{y}$ are bindings (verification information in messages (6) and (10)) to complete the protocol run by $EV_j - LAG$ and $EV_j - CA$, respectively.

Property 2. *Adversary $\mathcal{A}$ cannot extract secret session key k over the network. Furthermore, perfect forward secrecy is maintained by our scheme in the V2G network.*

Each session secret key is used for authentications between the EV_j and the *LAG* in the home as well as visiting *V2G* networks, and is actually never sent over the network. There is no such requirement in the centralized *V2G* network. Furthermore, even if $\mathcal{A}$ is allowed to access private key of the *EV*, it cannot generate past session keys, as these keys are generated using (μ', S_{LAG}, Q_{EV_j}) and (PID_i, ξ, S_{EV_j}, Q_{LAG}). Clearly, past PID_i are not valid in the current session. Moreover, old PID_i are no longer part of μ' and ξ. Therefore, $\mathcal{A}$ cannot retrieve the past session keys. Furthermore, if $k_{EV_j}^{-1} \notin \mathcal{K}_{\mathcal{A}}$, $k_{LAG}^{-1} \notin \mathcal{K}_{\mathcal{A}}$, and T_i is uniquely originated in Σ, then for all $m \in C$, $T_i \neq term(m)$.

$$\neg\exists C.(LAG(\overrightarrow{x}) \wedge CA(\overrightarrow{x}) \in C \wedge EV(+SoC) \in C),$$

where $\overrightarrow{x}$ is the response received by the EV_j to complete the protocol run (messages (6) & (10) in Figure 2 and messages (2) & (5) in Figure 7).

Property 3. *Adversary $\mathcal{A}$ cannot gain non-negligible advantage by performing chosen message attack in the V2G network. Also, $\mathcal{A}$ cannot compromise message integrity.*

In our scheme, encrypted message (6) and message (7) generate different ciphertexts even by using same session key. The *LAG* generates each *PID* using a secure and efficient pseudo-random function [35]. Moreover, encryption of unique r_j in message (6) and signature in message (7) are performed by the *LAG* and *EV*, respectively, using *AES-CTR* that encrypts and decrypts the successive values of a counter (ctr) with *AES* as $C[0] \leftarrow ctr$; $P[i] \leftarrow F_k(ctr+i)$; $C[i] \leftarrow P[i] \oplus M[i]$; and $ctr \leftarrow C[0]$; $P[i] \leftarrow F_k(ctr+i)$; $M[i] \leftarrow P[i] \oplus C[i]$, where $C[i]$, $P[i]$, $M[i]$, and $F_k()$ are ciphertext, plaintext, message block to be processed, and a function to process ctr, respectively. $\mathcal{A}$ cannot distinguish between such streams of equal lengths. In fact, encrypting two distinct ctr using *AES-CTR* obtain two distinct values, and hence, it is indistinguishable.

Our scheme provides integrity protection by using hash values with each transmitted message over the network. Here, hash function $H : \mathbb{A} \rightarrow \mathbb{A}$, and $H(M) = H(M') \Leftrightarrow M = M'$, $M, M' \in \mathbb{A}$, where strand with trace for EV_j, *LAG*, and *CA* are $\langle +H_1, -H_2, +hash_1, -hash_2, +H_3, +H_4 \rangle$, $\langle -H_1, +H_2, -H_3, -H_4 \rangle$, and $\langle -hash_1, +hash_2 \rangle$, respectively. If $\mathcal{A}$ intentionally changes any message, the received and computed hash values will not match at the receiver, and the connection will be terminated. Furthermore, a hash of the key, instead of the actual session key, is sent over the network. We use *SHA256* hash function (where possible hash codes $m = 2^{256}$ with n-bit message), which is still considered collision resistant. The probability of successful attack on *SHA256* is as follows:

$$Pr \approx 1 - exp\left(-\frac{n(n-1)/2}{2^{256}}\right) \approx 1 - exp\left(\frac{1}{2}\left(\frac{n}{2^{128}}\right)^2\right)$$

Hence, the probability of successful attack is negligible as long as 2^{128} values of hash are used.

Property 4. *Our scheme defeats impersonation, MITM, replay and injection, and redirection attacks over the network.*

Our scheme defeats the following security attacks:

a. Impersonation Attack: Adversary $\mathcal{A}$ must know ID_j and/or session key of the victim EV_j in order to perform this attack. However, $\mathcal{A}$ cannot obtain secret shared key. There are two possible cases of this attack as follows:

- *Case-1: $\mathcal{A}$ impersonates the EV_j:* $\mathcal{A}$ uses a fake *PID* as PID_l with a hash $\mathcal{A}$-H_1. Obviously, $PID_l \neq PID_j$, and *LAG* rejects the request and terminates the connection.
- *Case-2: $\mathcal{A}$ impersonates the LAG:* The rogue $\mathcal{A}$-*LAG* would not be able to retrieve correct μ', as $H(Q_{CA}^{S_{\mathcal{A}\text{-}LAG}}) \neq H(Q_{CA}^{S_{LAG}})$. Furthermore, $H(f_1) \neq H(f_2)$ at EV_j. Hence, the EV_j terminates the connection. A similar case exists in the centralized *V2G* network where the private key of the *Central-LAG* is different than $\mathcal{A}$-*LAG*'s key.

b. MITM Attack: $\mathcal{A}$ may try to secretly build a connection between two communicated parties with the following cases:

- *Case-1: Key-exchange by $\mathcal{A}$:* $\mathcal{A}$ cannot establish a connection with the EV_j and the *LAG*, as it cannot compute $H(Q_{EV_j}^{S_{LAG}})$ or $H(Q_{LAG}^{S_{EV_j}})$. Also, it cannot compute correct $H(Q_{EV_j}^{S_{CA/RA}})$ or $H(Q_{CA/RA}^{S_{EV_j}})$ between the EV_j and the *CA/RA*. Further, $\mathcal{A}$ cannot generate correct f_1 or f_2.
- *Case-2: $\mathcal{A}$ as a rogue LAG or a friend:* $\mathcal{A}$ may install a fake $\mathcal{A}$-*LAG*, extracts information provided by the EV_j and later uses it to access the system from a valid *LAG*. Moreover, a friend who has an access to the vehicle and knows security key may perform various unintended tasks. In order to prevent such access, after receiving the message from the *LAG*, the *CA/RA* sends an *OTP* to the EV_j's owner for identity verification. Hence, two-factor authentication prevents the system against a rogue *LAG*: one by sending an *OTP* and other by verifying PID_j.
- *Case-3: $\mathcal{A}$ tries to extract secret information:* $\mathcal{A}$ may also try to extract information from message (7). $\mathcal{A}$ neither can decrypt the message as it cannot generate k, nor can retrieve the private keys of the EV_j, *LAG*, and *CA/RA*. Hence, $\mathcal{A}$ cannot perform *MITM* attack.

c. Replay and Injection Attacks: $\mathcal{A}$ can intercept, inject, or re-send messages in order to perform replay attacks. Our scheme resists replay attacks by using timestamp values in all transmitted messages between the EV_j and the *LAG*. If $\mathcal{A}$ replays a previous message or injects information to a message at T_i, legitimate *LAG*, *CA/RA*, and *EV* discard the message if $T_i + T_{threshold} \leq T_{current}$, where $T_{threshold}$ is the threshold value of the propagation time between two entities.

d. Redirection Attack: In the home and visiting *V2G* networks, each EV_j sends *CSID* to the *Home/Visiting-CA/RA*. *Home/Visiting-CA/RA* verifies the location of each EV_j by matching received information from the EV_j with the stored information. If they do not match, *Home/Visiting-CA/RA* discards the connection. Furthermore, there is no such requirement for the centralized *V2G* network, as it allows only discharging of the battery, and each EV_j is paid on the spot.

TABLE II: Comparison of Security and Privacy Goals

Goals	[16]	[7]	[8]	[9]	[15]	[5]	Our Scheme
Mutual authentication	Yes	Yes	Yes	Yes	No	Yes	Yes
Identity protection	Yes	Yes	Yes	Yes	No	No	Yes
Message integrity	Yes	No	No	No	No	No	Yes
Replay attack	Yes	No	No	No	No	Yes	Yes
MITM attack	Yes	No	Yes	No	No	Yes	Yes
Redirection attack	Yes	Yes	Yes	Yes	No	No	Yes
Impersonation attack	Partial	No	No	No	No	No	Yes

e. Other Attacks: Our scheme also prevents *Known Key* attack in the home and visiting *V2G* networks, as each secret key k is different and is newly generated for each session between the EV_j and the *Home/Visiting-LAG*. Also, the identity and hash-signature verification prevents *Repudiation* attack.

Table II summarizes security and privacy goals achieved by various schemes, and our scheme fulfills all such goals.

Property 5. *Adversary $\mathcal{A}$ cannot compromise privacy of the vehicle, as our scheme maintains anonymity, untraceability, and forward privacy.*

Privacy of each EV_j is protected during authentications over the network. Each EV_j's PID_j, which is initially generated by the *Home-CA/RA*, is actually sent only once over the network. After each session, the EV_j requests for a new PID_j to the *Home-CA/RA*. Similarly, in the visiting and centralized *V2G* networks, the actual identity of each EV_j is well protected. We quantify the anonymity provided by PID_i in terms of the advantage of $\mathcal{A}$ for correctly guessing the challenge bit.

Definition 13: (*Indistinguishability under Anonymous Identity (IND-ANO)*): Our scheme is *IND-ANO* as no adversary $\mathcal{A}$ at time t can distinguish between the two chosen identity PID_1 and PID_2 with negligible ϵ advantage.

$$Pr[\mathcal{A}(PID_1) = 1] - Pr[\mathcal{A}(PID_2) = 1] \leq \epsilon.$$

If EV_j is a vehicle of *EV* strand s and *fun(C) = EV*, where *fun* is an evaluation function of bundle C.

$$\text{For } \forall u \in U, \text{ if } fun(C) = u, \text{ then } \forall EV \in U/\{u\},$$

where U is an anonymity set. Also,

$$\text{if } \exists C_{EV_j} \text{ satisfies } fun(C_{EV_j}) \text{ and } C_{EV_j} \cong C,$$

the protocol of bundle C preserves sender anonymity. Our scheme maintains anonymity, as the actual identity is only known to EV_j and *CA/RA*. The intermediate *LAG*s believe on only the facts (identity set) provided by the *CA/RA*.

Definition 14: (*Untraceability*): Our scheme satisfies untraceability as $\mathcal{A}$ cannot distinguish whether two *PID*s correspond to the same *EV* or two different *EV*s.

$$\begin{aligned}&Verif(publicChannel)[(ID_1, ID_2)|PID_i|EV|CA/RA]\\ \approx\ &Verif(publicChannel)[ID_1|ID_2|PID_i|EV|CA/RA].\end{aligned}$$

Our scheme transmits PID_j instead of the original identity over the network. Even if $\mathcal{A}$ retrieves a PID_j and makes a query from random oracle to generate several PID_j from ID_j, $\mathcal{A}$ cannot conclude which ID_j matches with the retrieved PID_j, as a unique PID_j is generated using a pseudo-random function. Furthermore, our scheme generates a new key for each session based on the unique PID_j. Therefore, linkability to previous sessions is not possible. Also, by holding the generated identities and messages during the current session (say time t), $\mathcal{A}$ cannot determine whether these messages belong to a particular vehicle after t_{frd}, as each PID_j is independent and is deleted by *CA/RA* after each authentication. Similarly, $\mathcal{A}$ cannot know whether these messages were generated by a particular vehicle before t_{brd}, as each session's identities, keys, and messages are independent.

Definition 15: (*Forward Privacy*): Our scheme satisfies forward privacy as $\mathcal{A}$ is allowed to trace the *EV* in the current session, but it cannot trace information related to the previous scheme sessions. In other words,

$$Verif(publicChannel)[(ID_1, PID_2)|PID_i|EV|CA/RA] \approx Verif(publicChannel)[PID_1|PID_2|PID_i|EV|CA/RA].$$

We also consider forward privacy scenario, where even if $\mathcal{A}$ is given a breakable ID_1, $\mathcal{A}$ cannot trace PID_1, as the identity is generated by a secure pseudo-random function. Also, the location of each vehicle is unknown to the *LAG*s, as it can only access PID_i, and not ID_i. Furthermore, each session protects secret keys by *PFS*. Therefore, our scheme maintains anonymity, untraceability, and forward privacy properties.

C. Performance Analysis

The performance of our scheme is evaluated in terms of communication and computation overheads. We compare our scheme (home, visiting, and centralized *V2G* networks) only with the schemes presented in [16] and [7]. Others are not comparable since the scheme presented in [15] does not provide mutual authentication and is vulnerable to attacks. The scheme in [5] is not fit to the *V2G* network, as it does not focus on vehicle behavior and *V2G* security and privacy features. The schemes proposed in [8] and [9] are the extended works of the scheme in [7], which has a huge overhead. We did not consider the overhead generated by the schemes in [8] and [9] since they generate even greater overheads.

i) Communication Overhead: Communication overhead is the total number of bits transmitted over the network during the scheme execution. As shown in Table III, the overhead of our scheme in the home *V2G* network for initial and subsequent authentications are 2993 bits and 737 bits, respectively, which is lower than the existing schemes [16] and [7]. In detail, if we assume that there are n number of *EV*s that are requesting for authentications simultaneously, the total communication cost (for the first attempt) of our scheme in the home *V2G* network would be $2993 \times n$. Also, if we assume that r number of authentication requests are allowed by each *EV* to the *CA/RA* within a session. For subsequent authentications, our scheme generates $737 \times r$ communication overhead while the existing schemes (with no session) in [16] and [7] generate $3392 \times r$ and $3264 \times r$, respectively in the home *V2G* network. Our scheme is efficient in terms of communication overhead, as fewer parameters are required to send over the network. We also compute the communication overhead for our scheme under the visiting and centralized *V2G* networks. We analyzed that our scheme in the visiting *V2G* network generates 3649 and 737 bits for initial and subsequent authentications, respectively, while it is 1728 bits for an authentication in the centralized *V2G* network. Our scheme is always better in the home as well as centralized *V2G* networks in comparison with other schemes. Moreover, our scheme is also efficient in all the networks for subsequent authentications.

ii) Computation Overhead: We compute the computation overhead as presented in Table IV for n-*EV*s simultaneously requesting for authentications. In the home *V2G* network, the total computation cost for schemes in [16], [7], and our scheme are $76 \times n$, $49 \times n$+5, and $39 \times n$+16, respectively. Assuming a unit value for each operation, our scheme is more efficient than the existing schemes. The actual computation time by each scheme depends on the actual time taken by each operation. The lower overhead is achieved by reducing pairing, exponential and scalar multiplication operations, and utilizing hash-based signatures. Furthermore, the computation overhead incurred by our scheme under the visiting and centralized *V2G* networks are $53 \times$n+16 and $38 \times$n+5, respectively, outperforming the scheme in [16]. Although, it generates a slightly

TABLE III: Communication Overhead (in bits)

Overhead	P^2 [16]	*AP3A* [7]	Our Scheme Home	Visiting	Centralized
Initial authentication	3392	3264	2993	3649	1728
Subsequent authentication	3392	3263	737	737	1728

TABLE IV: Computation Overhead

Operations	P^2 [16]	*AP3A* [7]	Our Scheme Home	Visiting	Central.
Pairing	$19\times n$	–	$2\times n$	$2\times n$	–
Exponential	$14\times n$	$12\times n$	$9+11\times n$	$9+17\times n$	$3+13\times n$
Scalar multiplication	$28\times n$	n	$1+8\times n$	$1+10\times n$	$10\times n$
Addition	$11\times n$	$-3+3\times n$	$2\times n$	$2\times n$	n
Invertible	n	$2+2\times n$	–	–	–
Hash (H)	$6\times n$	$1+8\times n$	$6+12\times n$	$6+18\times n$	$2+14\times n$
Auth. code (*HMAC*)	$7\times n$	$2\times n$	–	–	–
Encryption/ decryption	–	$4\times n$	$4\times n$	$4\times n$	–
XOR	–	$5+17\times n$	–	–	–
Total	$76\times n$	$5+49\times n$	$16+39\times n$	$16+53\times n$	$5+38\times n$

higher computation overhead in the visiting *V2G* network as compared to [7]. Our scheme always outperforms in the home as well as centralized *V2G* networks.

iii) Experimental Setup: We consider a *V2G* network scenario with an authentication server *CA/RA* remotely connected with various *LAG*s. The specification of our system is 1.70 GHz Core i3-4005U CPU with 4GB RAM and 500 GB drive. We performed simulation in Java with JDK1.7. We implemented H_1 as *SHA256* function, which took 20 milliseconds (*ms*). Further, a pairing function (J-pairing) took 197 *ms*, while modular exponentiation and scalar multiplication were executed in 2.1 *ms* and 0.8 *ms*, respectively. Moreover, addition operation, *MAC* function (*HMACSHA256*), and *AES* with Counter (*AES-CTR*) encryption and decryption took 0.03 *ms*, 246 *ms*, and 0.23 *ms* and 0.13 *ms*, respectively. Also, a simple invertible function executed in 0.8 *ms*. For a single authentication, P^2 scheme [16] and *AP3A* scheme [7] took 5637.93 *ms* and 680.41 *ms*, respectively, while our scheme in home, visiting, and centralized *V2G* networks took 680.28 *ms*, 814.48 *ms*, and 320.33 *ms*, respectively. Currently, the average mobile broadband download speed on *4G* Long Term Evolution (*LTE*) is 10 *Mbps* [37]. The connection establishment time for each scheme is about 3000 *ms*, and the total transmission times for all the messages in each scheme's initial and subsequent authentications are computed to be (3000.34, 3000.34) *ms*, (3000.32, 3000.32) *ms*, (3000.3, 3000.07) *ms*, (3000.36, 3000.07) *ms*, and (3000.17, 3000.17) *ms* for P^2 scheme [16], *AP3A* scheme [7], our scheme in home, visiting, and centralized *V2G* networks, respectively. Overall, the total execution times for a single authentication in P^2 scheme [16], *AP3A* scheme [7], and our scheme in home, visiting, and centralized *V2G* networks are 8.63, 3.68, 3.68, 3.81, and 3.32 *sec.*, respectively. It is clear that P^2 scheme [16] has a larger execution time, while our scheme in visiting *V2G* network is slightly slow than *AP3A* scheme [7]. However, our scheme in home and centralized *V2G* networks outperforms other schemes.

D. Security Proof of Our Scheme

We prove the correctness of our scheme using automatic tool named *Proverif*. Following are the input and output observed from *Proverif*:

free *pubChannel* : channel.
free *secureChannel* : channel [private].
type key. type msgHdr. type bitstring. type skey. type pkey.
const *MSG*$_1$, *MSG*$_2$, *MSG*$_3$, *MSG*$_4$, *MSG*$_5$, *MSG*$_6$, *MSG*$_7$, *MSG*$_8$, *MSG*$_9$, *MSG*$_{10}$, *CMC*, *MSG*: msgHdr.
fun sha256 (bitstring): bitstring.
fun sha2561 (bitstring,bitstring,ident): bitstring.
fun sha2562 (bitstring,bitstring,bitstring,bitstring,bitstring): bitstring.
fun sencrypt (bitstring,key): bitstring.
reduc forall m: bitstring, k: key;
sdecrypt(sencrypt(m,k),k) = m.
fun pk(skey): pkey.
fun aenc(bitstring,pkey): bitstring.
reduc forall m: bitstring, k: skey; adec(aenc(m, pk(k)),k) = m.
fun msg1(bitstring,bitstring,ident,bitstring,bitstring): bitstring.
fun msg2(bitstring,bitstring): bitstring.
fun mul(bitstring,bitstring): bitstring.
fun sign(bitstring,skey): bitstring.
fun tempid (ident,bitstring): bitstring.
fun e(bitstring,bitstring): key.
fun exp1(bitstring,bitstring): bitstring.
fun del(bitstring,bitstring,bitstring,bitstring):bitstring.
fun div(bitstring,bitstring):bitstring.
free s: bitstring [private].
query attacker(s).
free K_i: key [private].
query attacker(K_i).
not attacker(new *ID*$_{ev}$).
event begLAG(bitstring,key). event endLAG(bitstring,key).
event begEV(bitstring,key). event endEV(bitstring,key).
event begLAG(msgHdr). event endLAG(msgHdr).
event begEV(msgHdr). event endEV(msgHdr).
query x_1: bitstring, x_2: key;
event(endLAG(x_1,x_2)) ==> event(begLAG(x_1,x_2)).
event(endEV(x_1,x_2)) ==> event(begEV(x_1,x_2)).
event(endLAG(*MSG*)) ==> event(begLAG(*MSG*)).
event(endEV(*MSG*)) ==> event(begEV(*MSG*)).
event enableEnc.
query attacker(s) ==> event(enableEnc).

let processEV =
(* The identity and pre-shared key of the *EV* *)
new S_{ev}: skey, Q_{ev}: pkey, Q_{lag}: pkey, Q_{ca}: pkey;
new *ID*$_{ev}$: bitstring, x_{ev}: bitstring, α_{ev}: bitstring, $g1_{ev}$: bitstring, $g2_{ev}$:bitstring, $T1_{ev}$: bitstring, $T3_{ev}$:bitstring, $T4_{ev}$:bitstring, $T5_{ev}$:bitstring, M_{ev}:bitstring;
let σ_{ev-lag}:bitstring=sign(Q_{lag},S_{ev}) in
let σ_{ev-ca}:bitstring=sign(Q_{ca},S_{ev}) in
let Γ_{ev}: bitstring=exp1(α_{ev},$g1_{ev}$) in
out($secureChannel$,(*MSG*$_1$,*ID*$_{ev}$,Γ_{ev}));
in($secureChannel$,(=*MSG*$_2$,*PID*$_{ev}$:bitstring));
let γ_{ev}: bitstring=exp1(x_{ev},$g1_{ev}$) in
let $H1_{ev}$:bitstring=sha2561(γ_{ev},*PID*$_{ev}$,$T1_{ev}$) in
out($pubChannel$,(*MSG*$_5$,γ_{ev},*PID*$_{ev}$,$T1_{ev}$,$H1_{ev}$));
event begLAG(*MSG*$_6$);
in($pubChannel$,(=*MSG*$_6$,*HK*$_{lag}$:bitstring,λ_{lag}:bitstring,ξ_{lag}: bitstring, *EK*$_{lag}$:bitstring,$T2_{lag}$:bitstring,$H2_{lag}$:bitstring));
let $H2_{ev}$:bitstring=sha2562(*HK*$_{lag}$,λ_{lag},ξ_{lag},*EK*$_{lag}$,$T2_{lag}$) in
if $H2_{ev}$ = $H2_{lag}$ then
event endLAG(*MSG*$_6$);
let $tmp1_{ev}$:bitstring=exp1(*PID*$_{ev}$,$g1_{ev}$) in
let $tmp2_{ev}$:bitstring=sha256(σ_{ev-lag}) in
let $tmp3_{ev}$:bitstring=exp1($temp1_{ev}$,$tmp2_{ev}$) in
let k_{ev}:key=e($tmp3_{ev}$,ξ_{ev}) in
let HK_{ev}:bitstring=sha256(k_{ev}) in
if HK_{ev} = *HK*$_{lag}$ then
let r_{ev}:bitstring=sdecrypt(sencrypt(*EK*$_{lag}$,k_{ev}),k_{ev}) in
let δ_{ev}:bitstring=del(x_{ev},α_{ev},λ_{ev},*PID*$_{ev}$) in
let *EK*$_{ev}$:bitstring=sencrypt((δ_{ev},$T3_{ev}$),k_{ev}) in
out($pubChannel$,(*MSG*$_7$,*EK*$_{ev}$,r_{ev}));
(* Command and Operation Selection Window appeared *)
if enableEnc = true then
out($secureChannel$,(*MSG*$_9$,M_{ev},$T4_{ev}$));
(* Receive *Decision* from *CA* *)
in($secureChannel$,(=*MSG*$_{10}$,$msg2_{ca}$:bitstring,$T5_{ev}$:bitstring));
let $hmsg2_{ev}$:bitstring=sha256(σ_{ev-ca}) in
let dec_{ev}:bitstring=div($msg2_{ca}$,$hmsg2_{ev}$) in
event endLAG(*PID*$_{ev}$,k_{ev});

let processLAG =
new S_{lag}: skey, Q_{lag}: pkey, Q_{ev}: pkey, Q_{ca}: pkey;
new $g1_{lag}$: bitstring, $g2_{lag}$: bitstring, r_{lag}:bitstring, *PID*$_{ev}$:bitstring, γ_{lag}:bitstring, Γ_{ev}:bitstring;
let σ_{lag-ev}:bitstring=sign(Q_{ev},S_{lag}) in
let σ_{lag-ca}:bitstring=sign(Q_{ca},S_{lag}) in
in($secureChannel$,(=*MSG*$_3$, μ_{lag}:bitstring, λ_{lag}:bitstring));
event begEV(*MSG*$_5$);
in($secureChannel$,(=*MSG*$_5$,γ_{ev}: bitstring, *PID*$_{ev}$:bitstring, $T1_{ev}$:bitstring, $H1_{ev}$:bitstring));
let $H1_{lag}$:bitstring=sha2561(γ_{ev},*PID*$_{ev}$,$T1_{ev}$) in
if $H1_{ev}$ = $H1_{lag}$ then
event endEV(*MSG*$_5$);
let ξ_{lag}:bitstring=exp1(*PID*$_{ev}$,$g2_{lag}$) in
let $tmp1_{lag}$:bitstring=exp1($g1_{lag}$,μ_{lag}) in
let $tmp2_{lah}$:bitstring=sha256(σ_{lag-ev}) in
let k_{lag}:key=e($tmp1_{lag}$,$tmp2_{lag}$) in
let *HK*$_{lag}$:bitstring=sha256(k_{lag}) in
let *EK*$_{lag}$:bitstring=sencrypt(r_{lag},k_{lag}) in

let $H2_{lag}$:bitstring=sha2562(HK_{lag},λ_{lag},ξ_{lag},EK_{lag},$T2_{lag}$) in
out($pubChannel$,(MSG_6,HK_{lag},λ_{lag},ξ_{lag},EK_{lag},$T2_{lag}$,$H2_{lag}$));
event beginEV(PID_{ev},k_{lag});
in($pubChannel$,(=MSG_7,EK_{ev}:bitstring,r_{ev}:bitstring));
let δ_{lag}, T_3:bitstring=sdecrypt(sencrypt(EK_{ev},k_{ev}),k_{lag}) in 0.
let H_{lag-ca}:bitstring=sha256(σ_{lag-ca}) in
let $msg1_{lag}$:bitstring=msg1(γ_{lag},δ_{lag},PID_{ev},T_j,H_{lag-ca}) in
out($pubChannel$,(MSG_8,$msg1_{lag}$));
event endEV(PID_{ev},k_{lag});
out($secureChannel$,(MSG_4,PID_{ev},r_{lag}));

let processCA =
new S_{ca}: skey, Q_{lag}: pkey, Q_{ev}: pkey, Q_{ca}: pkey;
new $g1_{ca}$: bitstring, Γ_{ev}:bitstring, PID_{ev}:bitstring, γ_{ca}:bitstring, μ_{ca}:bitstring, ID_{ev}:bitstring, λ_{ca}:bitstring, r_{ca}:bitstring, $msg1_{lag}$:bitstring, $T4_{ca}$:bitstring, M_{ca}:bitstring, $decision_{ca}$:bitstring, $T5_{ca}$:bitstring, r_{ca}:bitstring;
let σ_{ca-lag}:bitstring=sign(Q_{lag},S_{ca}) in
let σ_{ca-ev}:bitstring=sign(Q_{ev},S_{ca}) in
in($secureChannel$,(=MSG_1,ID_{ev}:bitstring,Γ_{ev}:bitstring));
let PID_{ev}: bitstring=tempid(ID_{ev},Γ_{ev}) in
out($secureChannel$,(MSG_2,PID_{ev}));
out($secureChannel$,(MSG_3,μ_{ca},λ_{ca}));
in($pubChannel$,(=MSG_8,$msg1_{lag}$:bitstring));
let H_{ca-lag}:bitstring=sha256(σ_{ca-lag}) in
let $msg1_{ca}$:bitstring=div($msg1_{lag}$,H_{ca-lag}) in
(* Extract γ_{lag}:bitstring,δ_{lag}:bitstring,PID_{ev}:bitstring, T_j:bitstring *)
let δ_{ca}:bitstring=exp1($g1_{ca}$,δ_{lag}) in
let PID_{ca}:bitstring=exp1($g1_{ca}$,PID_{ev}) in
let $PIDmul$:bitstring=mul(δ_{ca},Γ_{ev}) in
let $PIDmul_{ca}$:bitstring=exp1($PIDmul$,λ_{ca}) in
let mul_{ca}:bitstring=mul(γ_{ca},$PIDmul_{ca}$) in
if mul_{ca} = δ_{ca} then
(* Send Command Window to *EV* *)
in($secureChannel$,(=MSG_9,M_{lag}:bitstring,$T4_{lag}$:bitstring));
(* Compute Supply-Demand and Make a *Decision* *)
let H_{ca-ev}:bitstring=sha256(σ_{ca-ev}) in
let $msg2_{ca}$:bitstring=msg2($decision_{ca}$,H_{ca-ev}) in
out($pubChannel$,(MSG_{10},$msg2_{ca}$));
in($secureChannel$,(=MSG_4,PID_{ev}:bitstring,r_{ca}:bitstring));
(* *CA* Removes PID_{ev} and Stores r_{ca} *)

process
((! processEV) | processLAG | processCA)

Output: `Neetesh@Neetesh-PC /proverif1.88`
`$ ./proverif examples/v2g.pv`
– *Query attacker(s[]) ==> event(enableEnc)*
Completing... ok, secrecy assumption verified: fact unreachable attacker (ID_{ev}[!1 = v_763])
RESULT attacker(s[]) ==> event(enableEnc) is true.
– *Query event(endEV(x_1,x_2)) ==> event(begEV(x_1,x_2))*
Completing... ok, secrecy assumption verified: fact unreachable attacker (ID_{ev}[!1 = v_1651])
RESULT event(endEV(x_1,x_2)) ==> event(begEV(x_1,x_2)) is true.
– *Query event(endLAG(x_1_1791,x_2_1792)) ==> event(begLAG(x_1_1791,x_2_1792))*
Completing... ok, secrecy assumption verified: fact unreachable attacker (ID_{ev}[!1 = v_2542])
RESULT event(endLAG(x_1_1791,x_2_1792)) ==> event(begLAG(x_1_1791,x_2_1792)) is true.
– *Query not attacker(K_i[])*
Completing... ok, secrecy assumption verified: fact unreachable attacker (ID_{ev}[!1 = v_3345])
RESULT not attacker(K_i[]) is true.
– *Query not attacker(s[])*
Completing... ok, secrecy assumption verified: fact unreachable attacker (ID_{ev}[!1 = v_4140])
RESULT not attacker(s[]) is true.

VI. Conclusion

In this work, we presented an authentication scheme for charging/discharging of electric vehicles considering mobility of the vehicles in distributed as well as centralized *V2G* networks. Specifically, our scheme, based on a two-factor authentication, provides mutual authentications between the *EV*s and the *CA/RA*s (and/or *LAG*s) in all three networks, *i.e.*, home *V2G* network, visiting *V2G* network, and centralized *V2G* network. Our scheme is shown to defeat various security attacks, including insider attacks, and preserves privacy of the vehicles by establishing a secure connection to charging stations. Through comprehensive security analysis, we prove that our scheme provides resistance against various security attacks, such as *MITM*, replay, redirection, impersonation, known key, and repudiation attacks in the *V2G* network. Moreover, our scheme provides perfect forward secrecy, indistinguishability under the chosen message attack, message confidentiality and integrity, untraceability, forward privacy, and identity and location anonymity. Analytic results show that our scheme generates lower communication and computation overheads in comparison with the existing schemes in the home and centralized *V2G* networks, and comparable overheads in the visiting *V2G* network by sending limited information over the network. Experimental results show that our scheme in the home and centralized *V2G* networks outperform, while our scheme in the visiting *V2G* network is slightly slow than *AP3A* [7], but is better than P^2 [16].

Acknowledgment

This research was funded by the Ministry of Science, ICT and Future Planning, Korea, under the "ICT Consilience Creative Program" (IITP-2015-R0346-15-1007) supervised by Institute for Information & Comm. Tech. Promotion and under the "Basic Science Research Program" (2013R1A1A1010489, 2015R1C1A1A01053788) through NRF.

References

[1] IEEE 1609, Family of Standards for Wireless Access in Vehicular Environments. http://www.standards.its.dot.gov/factsheets/factsheet/80.

[2] A. Briones, J. Francfort, and P. Heitmann, "Vehicle-to-Grid (V2G) power flow regulations and building codes review by the AVTA," INL US Department of Energy National Laboratory, Sep. 2012, 98 pages. [Online]. http://energy.gov/sites/prod/files/2014/02/f8/v2g_power_flow_rpt.pdf.

[3] H. Yang, X. Xie, and A. V. Vasilakos, "Non-cooperative and cooperative optimization of electric vehicles charging under demand uncertainty: a robust stackelberg game," *IEEE Transactions on Vehicular Technology*, Oct. 2015. [Online]. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber= 7297873.

[4] Batteries for Electric Cars: Challenges, Opportunities, and the Outlook to 2020, The Boston Consulting Group. [Online]. www.bcg.co.kr/documents/file36675.pdf.

[5] H. Guo, Y. Wu, F. Bao, H. Chen, and M. Ma, "UBAPV2G: a unique batch authentication protocol for vehicle-to-grid communications," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 707-714, Dec. 2011.

[6] Y. Zhang, S. Gjessing, L. T. Yang, S. F. Xavier, and M. Guizani, "Securing vehicle-to-grid communications in the smart grid," *IEEE Wireless Communication*, vol. 20, no. 6, pp. 66-73, Dec. 2013.

[7] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, "Aggregated-proof based privacy-preserving authentication for V2G networks in the smart grid," *IEEE Trans. on Smart Grid*, vol. 3, no. 4, pp. 1722-1733, Dec. 2012.

[8] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, "Role-dependent privacy preservation for secure V2G networks in the smart grid," *IEEE Trans. on Information Forensics & Security*, vol. 9, no. 2, pp. 208-220, Feb. 2014.

[9] H. Liu, H. Ning, Y. Zhang, and M. Guizani, "Battery status-aware authentication scheme for V2G networks in smart grid," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 99-110, Mar. 2013.

[10] N. Saxena, B. J. Choi, and S. Cho, "Lightweight privacy preserving authentication scheme for V2G networks in smart grid," in *Proc. IEEE International Conference TrustCom*, Aug. 2015, pp. 604-611.

[11] C. Guille and G. Gross, "A conceptual framework for the vehicle-to-grid (V2G) implementation," *Energy Policy*, vol. 37, pp. 4379-4390, 2009.

[12] N. Saxena and B. J. Choi, "State of the art authentication, access control, and secure integration in smart grid," *Energies*, vol. 8, no. 10, pp. 11883-11915, 2015.

[13] R. Schmidt and A. Caldevilla, "V2G interface specifications between the electric vehicle, the local smart meter, and its service providers," 7^{th} Framework Programme, 2012. [Online]. http://www.power-up.org/wp-content/uploads/2012/07/PowerUp_D4.1_final.pdf.

[14] J. Zhou, X. Dong, and A. V. Vasilakos, "Secure and privacy preserving protocol for cloud-based vehicular DTNs," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 6, pp. 1299-1314, Jun. 2015.

[15] C. I. Fan, S. Y. Huang, and Y. L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 666-675, Feb. 2014.

[16] Z. Yang, S. Yu, W. Lou, and C. Liu, "P^2: 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, Dec. 2011.

[17] M. Badra and S. Zeadally, "Design and performance analysis of a virtual ring architecture for smart grid privacy," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 321-329, Feb. 2014.

[18] Z. M. Fadlullah, T. Taleb, and A. V. Vasilakos, "DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis," *IEEE/ACM Tr. on Networking*, vol. 18, no. 4, pp. 1234-1247, 2010.

[19] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, pp. 2481-2501, 2014.

[20] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," *Journal of Network and Computer Applications*, vol. 42, pp. 120-134, 2014.

[21] H. Yang, Y. Zhang, Y. Zhou, X. Fu, H. Liu, and A. V. Vasilakos, "Provably secure three-party authenticated key agreement protocol using smart cards," *Computer Networks*, vol. 58, pp. 29-38, 2014.

[22] T. Wang, Y. Liu, and A. V. Vasilakos, "Survey on channel reciprocity based key establishment techniques for wireless systems," *Wireless Networks*, vol. 21, pp. 1835-1846, 2015.

[23] C. Jie, Z. Yueyu, and S. Wencong, "An anonymous authentication scheme for plug-in electric vehicles joining to charging/discharging station in vehicle-to-grid (V2G) networks," *China Communications*, vol. 12, no. 3, pp. 9-19, Mar. 2015.

[24] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," *IEEE Trans. on Vehicular Technology*, Feb. 2015. [Online]. ieeexplore.ieee.org/iel7/25/4356907/07047924.pdf?arnumber=7047924.

[25] G. Wang, L. Chen, and S. L. Ng, "Threshold anonymous announcement in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605-615, 2011.

[26] J. Wan, H. Yan, D. Li, K. Zhou, and L. Zeng, "Cyber-physical systems for optimal energy management scheme of autonomous electric vehicle," *The Computer Journal*, vol. 56, no. 8, pp. 947-956, 2013.

[27] T. Chim, J. Cheung, S. Yiu, L. Hui, and V. Li, "SPCS: secure and privacy-preserving charging-station searching using VANET," *Journal of Information Security*, vol. 3, no. 1, pp. 59-67, 2012.

[28] R. Hussain, D. Kim, M. Nogueira, J. Son, A. O. Tokuta, and H. Oh, "PBF: a new privacy-aware billing framework for online electric vehicles with bidirectional auditability," Cornell University Library, Apr. 2015. [Online]. http://arxiv.org/pdf/1504.05276.pdf.

[29] M. A. R. Ortega, A. P. Ortega, and Z. Jako, "V2G conformance test specifications," 7^{th} Framework Programme, INFSO-ICT 285285, 2013. [Online]. http://www.power-up.org/wp-content/uploads/2013/11/ISO-IEC-15118-2_Conformance_Test_Descriptions_PowerUp.pdf.

[30] A. Amditis, E. Portouli, and A. Caldevilla, "Powerup preliminary V2G architecture," 7^{th} Framework Programme, INFSO-ICT 285285, 2012. [Online]. http://www.power-up.org/wp-content/uploads/2012/02/D3.1_PowerUp_Preliminary_ Architecture.pdf.

[31] F. Thayer, J. C. Herzog, and J. D. Guttman, "Strand spaces: Why is a security protocol correct?," in *Proc. IEEE Symposium on Security and Privacy*, May 1998, pp. 160-171.

[32] "SEC1: elliptic curve cryptography," Standards for Efficient Cryptography, Certicom, 2000. [Online]. www.secg.org/SEC1-Ver-1.0.pdf.

[33] E. Tremel, "Real-world performance of cryptographic accumulators," 2013. https://cs.brown.edu/research/pubs/theses/ugrad/2013/ tremel.pdf.

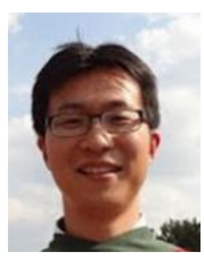
[34] L. Nguyen, "Accumulators from bilinear pairings and applications to ID-based ring signatures and group membership revocation," *Topics in Cryptology (CT-RSA)*, LNCS vol. 3376, Feb. 2005, pp. 275-292.

[35] D. Boneh, H. Montgomery, and A. Raghunathan, "Algebraic pseudorandom functions with improved efficiency from the augmented cascade, in *Proc. ACM CCS*, 2010, pp. 131-140. [Online]. http://dl.acm.org/citation.cfm?id=1866323.

[36] N. Saxena and N. S. Chaudhari, "VAS-AKA: an efficient batch verification protocol for value added services," in *Proc. IEEE International Conference on System, Man, and Cybernetics*, Oct. 2013, pp. 1560-1565.

[37] P. Goldstein, "T-Mobile offers fastest average LTE speeds," OpenSignal Report Finds, March 12, 2015. [Online] http://www.fiercewireless.com/story/t-mobile-offers-fastest-average-lte-speeds-opensignal-report-finds/2015-03-12.

Neetesh Saxena (S'10-M'14) received the Ph.D. degree in computer science & engineering from IIT Indore, India. He is currently a Postdoctoral Researcher with the Department of Computer Science, The State University of New York Korea, Korea, and a Visiting Scholar with the Department of Computer Science, Stony Brook University, USA. From 2013 to 2014, he was a Visiting Research Student and DAAD Scholar with the B-IT, Rheinische-Friedrich-Wilhelms Universitt, Bonn, Germany. He was also a TCS Research Scholar from 2012 to 2014. His current research interests include smart grid security, vehicle-to-grid security and privacy, cryptography, security and privacy in the cellular networks, and secure mobile applications. He is a member of ACM and CSI.



Bong Jun Choi (S'09-M'11) received his B.Sc. and M.Sc. degrees from Yonsei University, Korea, both in electrical and electronics engineering, and the Ph.D. degree from University of Waterloo, Canada, in electrical and computer engineering. He is currently an assistant professor at the Department of Computer Science, State University of New York Korea, Korea, and jointly a research assistant professor at the Department of Computer Science, Stony Brook University, USA. His current research focuses on energy efficient networks, distributed mobile wireless networks, smart grid communications, and network security. He serves as an editor of KSII Transactions on Internet and Information Systems and a member of the Smart Grid Core Security Technology Development Steering Committee, Korea. He also serves on the technical program committees for many international conferences. He is a member of the IEEE and the ACM.