Pairing-based Authentication Protocol for V2G Networks in Smart Grid

Luis Fernando Arias Roman[1], Paulo R. L. Gondim[1] and Jaime Lloret[2]

[1] Departamento de Engenharia Elétrica – Universidade de Brasília (UnB) - Brasília - Brasil;
[2] Integrated Management Coastal Research Institute - Universitat Politecnica de Valencia (UPV) – Valencia – Spain.

Abstract
Vehicle to Grid (V2G) network is a very important component for Smart Grid (SG), as it offers new services that help the optimization of both supply and demand of energy in the SG network and provide mobile distributed capacity of battery storage for minimizing the dependency of non-renewable energy sources. However, the privacy and anonymity of users' identity, confidentiality of the transmitted data and location of the Electric Vehicle (EV) must be guaranteed. This article proposes a pairing-based authentication protocol that guarantees confidentiality of communications, protects the identities of EV users and prevents attackers from tracking the vehicle. Results from computing and communications performance analyses were better in comparison to other protocols, thus overcoming signalling congestion and reducing bandwidth consumption. The protocol protects EVs from various known attacks and its formal security analysis revealed it achieves the security goals.

Keywords: Authentication, Electric Vehicles (EVs), Local Aggregator (AG), Session Key, Bilinear Pairing, Vehicle-to-Grid, V2G.

1. Introduction

Smart Grid has been developed as the next generation of energetic infrastructure. The mixture of the current electrical network and information technologies enables both clients and enterprises to participate in the monitoring management and energy distribution for a better demand-response balance.

Electric Vehicles (EVs) have been one of the most researched topics over the past years and can be easily integrated as part of a Smart Grid infrastructure. They have gained popularity towards reducing the air pollution (17% of the CO_2 global emissions) caused by fuel-operated vehicles. Studies have indicated 70% of the CO_2 emissions might be reduced if EVs were used to replace vehicles powered by traditional fuels [1].

An important part of EVs is their battery, considered a promising means of energy storing. They are stable storing units (their energy-loss rate is low) of fast charge and discharge, therefore, their integration with traditional energy plants is feasible for balancing changes in electricity demands. For instance, if the electricity demand increased, EVs would rapidly provide electricity from their batteries to the network and if it decreased, they could rapidly store the extra energy of the network. Such an interaction between EVs and

Smart Grid occurs through a bidirectional communication called Vehicle-to-Grid (V2G) [2-5].

V2G communication systems display special characteristics, as vehicle mobility [6], geographic location of the vehicle, charge and discharge operations [7], conduction pattern, among others. Several security and privacy challenges in communications can affect the V2G system, therefore, confidential information, as identity of the vehicle, user´s identity, identification of the charging station, type of vehicle, time of charge and discharge, and localization of the vehicle must be protected.

An EV has two operation modes, namely residential and visiting. The home mode refers to stations in the geographic area of a home area network where the vehicle resides and is registered, whereas the visiting mode includes stations outside of the residence area and is served by a visiting area network of the vehicle. Both modes have different security requirements ([5], [8-11]).

On the other hand, privacy and confidentiality are two very important concepts for informatics security. Private information must be kept confidential, which is one of the great challenges of V2G and Smart Grid networks. Every SG network is vulnerable to attacks to its different components, from EVs to Control Center (CC), therefore, security measures must comprehend the entire SG network infrastructure for ensuring availability, integrity, confidentiality and non-repudiation. However, some attacks may occur, such as replication, spoofing / sniffing of payload, Denial of Service (DoS) and Man-in-the-Middle attacks.

An authentication protocol is fundamental in the V2G networks for guaranteeing that only authorized EVs can access them. Therefore, an effective and efficient authentication system is highly required for guaranteeing privacy and confidentiality of data in V2G networks [12-15].

In addition to having security features, an authentication protocol must have low computational and communication costs. One of the techniques used by different works focused on V2G networks use group authentication [12, 16-17], since it offers optimum or very good performance, avoiding that the same operation has to be done for each member of the group.

The bilinear pairing technique has been widely applied in recent years by researchers for the generation of novel authentication protocols, with most known utilities as message encryption, key creation and digital signatures, among others. In this article, this technique will be applied to generate the session keys and ensure mutual authentication.

This article proposes a group authentication protocol for the administration and distribution of keys in a V2G architecture. The protocol is based on groups for managing secret keys, Elliptic Curve - Diffie Hellman (ECDH)[18] for sharing secrets and bilinear pairing for providing authentication and generation of simultaneous and efficient session keys for EVs grouped under aggregators.

The remainder of the manuscript is organized as follows: Section 2 describes some works related to the authentication of EV in the V2G network; Section 3 introduces the proposed protocol; Section 4 reports on a performance analysis of the protocol and describes the characteristics of the security properties; Section 5 addresses a formal verification of the protocol; finally, Section 6 provides the conclusions and suggests some future work.

2. Related Work

Significant security concerns for the V2G connection include the guarantee of the services provider, i.e., EV privacy and its authentication in the network. EV requires the preservation of its private information from any intermediate device in the connection between EVs and the authentication provider. Several protocols have been proposed for authenticating EVs in a V2G network, as presented below.

Abdallah et al. [3] designed protocols for insuring the confidentiality and integrity of exchanged information during (dis)charging sessions. The possible situations of the EVs are defined, i.e., energy storer, when CC produces more energy than that demanded and sends a message to the EVs of the area for them to purchase this energy and avoid energy loss; energy provider, when CC produces less energy than that demanded and sends a message to the EVs of the area for them to sell part of their energy and avoid overcharge; energy consumer, when the EV must charge energy; and energy seller, when the EV wishes to sell unnecessary energy.

Sun [20] proposed an authentication scheme that works with time intervals, so that a local aggregator can verify the authenticity of vehicles requesting connection. After such verification, the aggregator sends a confirmation message with the Boneh´s group signature scheme back to all EVs. The author also proposed a distributed system conformed to a trusted authority (TA) that generates public and private keys of entities, certificate of EV, and tracing of signature. The Central aggregator (CAG) collects information on local aggregators distributed in recharge areas and the local aggregator (LAG) registers the EVs and generates the group´s public key and the private key of each member. A charging / discharging station (ST) provides and monitors the loading / unloading of the EV and sends the information collected to the LAG.

Wan et al. [21] proposed an authentication protocol called PRAC (Privacy via Randomized Anonymous Credentials), which guarantees privacy of users through anonymous credentials (associated certificates) generated by the EVs. Anonymous credentials enable EVs to authenticate the system several times without contacting third parties. The authors considered a system conformed by a trusted third party (TTP) that generates private keys for LAs (local aggregators) and authentication credentials of EVs, a central aggregator (CA) that monitors and manages user´s account, and local aggregators distributed in an area to supervise and collect data from the EVs located there (the information collected is sent to the CA).

Shuaib et al. [22] developed three authentication schemes, namely local, internal roaming charging (IRC) and external roaming charging (ERC), to meet specific needs in local, visitor and commercial scenarios, respectively, and guarantee confidentiality, integrity and anonymity of systems through symmetrical and asymmetric cryptography. The authentication process depends on the scenarios, however, in general, the proposed architecture is composed of a Certifying Entity (CA) that provides trust for communication between entities of the system, external aggregator (EAG)/ visiting aggregator (VAG) / home aggregator (HAG), which collect information on consumption and send it to ES or HS, and Smart Meters (MS), which measure the energy consumed by the EV.

Jie et al. [12] designed an authentication protocol that preserves the privacy of users´ data in the connection of their electric vehicles for the charging or discharging of batteries in

the V2G network. It also optimizes communications through aggregators and dynamically manages the system. It uses group signatures and a partially blind signature restrictive technique based on identity. The architecture comprises five entities, namely Central Aggregator (CAG), LAG, Charging/discharging station (ST), Plug-in electric vehicle (PEV) and a trusted authority (TA). The protocol consists of the following four phases:

- Initial Configuration: all entities send their identities to the TA, which generates a pair of keys (public and private) for each entity and sends them to their corresponding entity. Each LAG generates a security parameter for each ST connected to it and defines functions and mathematical operations to be used and the group keys (public and private). It then defines the "Commitment" vectors and their public key and a signature.
- Generation and verification of permission: each ST generates a temporary pair of public/private keys and sends them in an encrypted message with the LAG public key to the LAG with the ST information and the temporary public key is generated. LAG checks the authenticity of both message and sender through a bilinear pairing operation. If it succeeds, ST is included in the LAG group.
- Generation of group blind certificate: each PEV calculates a random value and sends it to LAG, which builds a tree where each leaf is a PEV. It also calculates a compacted path value and a signature for each PEV. LAG sends a message to each PEV containing the compacted path and a verification value. PEVs check the message and, if the verification is successful, each PEV calculates a signature with the message received and sends it to LAG, which calculates a certificate for each PEV and sends them a message containing elements, so that they can calculate their certificate.
- Access of PEV to the V2G network through ST: PEV sends a message with its signature to ST, which checks if the signature is valid in the group. If the validation is met, ST enables PEV to connect with V2G. Finally, the information exchange between PEV and ST requires the generation of a session key in a bilinear pairing operation with their public and private keys.

Saxena et al. [13] proposed two authentication protocols for the access of EVs in the Smart Grid system for the recharge and discharge of their batteries in both residential and visiting modes, so that the following security requirements can be met: integrity of messages, confidentiality of data and users´ identity, mutual authentication of the entities and resistance to attacks to the system. However, for the sake of comparisons, only the protocol for the residential mode was described. The architecture designed by Saxena et al. [13] is composed of five entities, namely EVs, Charging Station (CS), LAG, Certification/Registration Authority (CA/RA) and Control Center (CC). The protocol proposed by Saxena et al. [13] consists of the following four parts:

- Initial configuration, where all entities generate a pair of public and private keys;
- Registration of EVs: each EV sends information to CA/RA and returns a temporary identity to the EV.

- LAG - CA/RA communication: all LAG must have the register of the temporary identities of all EVs registered in CA/RA, therefore, the communication between LAG - CA/RA occurs for updating the register of such entities.
- Protocol execution: when an EV must charge or discharge (sell) part of its energy, it approaches a CS, establishes communication with LAG and generates a session key that guarantees a mutual authentication between EV and LAG. The EV calculates an identity verification parameter and sends an encrypted message to the LAG with the session key. The LAG decrypts the researched message, adds information for the verification of the EV identity, and sends all parameters to the CA/RA in a message encrypted with the CA / RA digital signature generated by the LAG. Finally, CA/RA checks the EV identity and returns a message of commands to the EV. The remaining messages exchanged between the EV and CA/RA are encrypted under asymmetrical encryption based on blind digital firms.

In this paper, we propose a protocol that exhibits the following differences in comparison to the above described protocols, as discussed in the next section:

a) a distributed architecture for authentication services that considers a Centralized Authentication Server (CAS) and several Substation Authentication Servers (SAS) and delegates authorities towards speeding up the authentication process and avoiding pitfalls related to centralization;

b) the use of a binary tree for group management that efficiently controls the groups; the structure was successfully used by Parne et al. [25] in the context of an LTO / LTE-A network enabled for IoT. In our proposal, it is used to manage association and disassociation functions to groups by ensuring forward / backward secrecy (FS / BS) and anonymity among group members;

c) the partial use of Identity-based Signcryption (IBSC) [23] through which the protocol securely sends group broadcast messages, containing data for each EV and AG to generate the session key. The technique helps to improve the communication costs of the protocol;

d) in terms of attacks, we note that the [3], [12], [20-22] proposals do not consider protection against MITM, replay and injection, known key and DoS attacks; moreover, the proposals [3], [12], [20] and [21] do not consider prevention against personification and redirection attacks; our protocol is able to assure protection against all these attacks;

e) lower response times and bandwidth consumption, once it outperformed the protocols of Jie et al. [12] and Saxena et al. [13] regarding computational and communication costs;

f) the proposed protocol was formally validated, while some of the other proposals (for example [3], [12], [20], [21]) do not provide validation of security properties.

3. Proposed Protocol

The need for a new authentication protocol can be justified by the following arguments:

a) in a scenario where billions of users and devices, including vehicles, must be authenticated, authentication must be rapid and involve the minimum number of resources (e.g. bandwidth and processing). Wireless networks are commonly overloaded by voice and data traffics, which lead to the development of new alternatives for a better resource allocation (e.g. cognitive radio and traffic offloading/steering);
b) due to severe resource limitations, especially those related to bandwidth (spectrum scarcity), the number of bits sent on communication channels must be minimized. The quantification of communication costs has been discussed for our protocol and compared to two other recent alternatives (Jie et al. [12] and Saxena et al. [13]), with promising results;
c) low computing costs contribute to smaller response times and low consumption of processing resources, which leads to a protocol that imposes low delay and does not overload the components of the architecture;
d) SG is a system with a large number of threats and vulnerabilities, therefore, protocols that efficiently protect identities and control access to resources must be designed.

This section describes the proposed protocol. Initially, a possible architecture of V2G network is presented, and some concepts used (e.g. bilinear pairing) are briefly addressed for a better understanding of the intricacies of the protocol. A set of 3 (three) phases for the operation of the protocol is described and group membership is discussed.

3.1 Architecture of V2G Network

Figure 1 shows a possible V2G network architecture, composed of EVs recharging/discharging their batteries, where:

- Electric Vehicle (EV) refers to cars, motorcycles, boats, planes and other vehicles powered by electric energy stored in batteries.
- Charge/Discharge Stations (CDS) - installed in strategic locations, that charge or discharge the electrical energy of the vehicles´ batteries.
- Aggregators (AGs) are distributed in different regions of a city; a Local Aggregator (LAG) groups information from several EVs for decreasing the network communication costs; a Central Aggregator (CAG) concentrates information received from EV's;
- Authentication Server (AS) that validates the identity and credentials of EVs and stores their corresponding attributes. A distributed architecture with a Central Authentication Server (CAS) located in a control center and connected to several Substation Authentication Servers (SAS) can be used for a large system, as the SG network.
- Control Center (CC) an operations center that controls the whole electric network. The CAS that concentrates the information safely sent by SAS is installed in the CC.

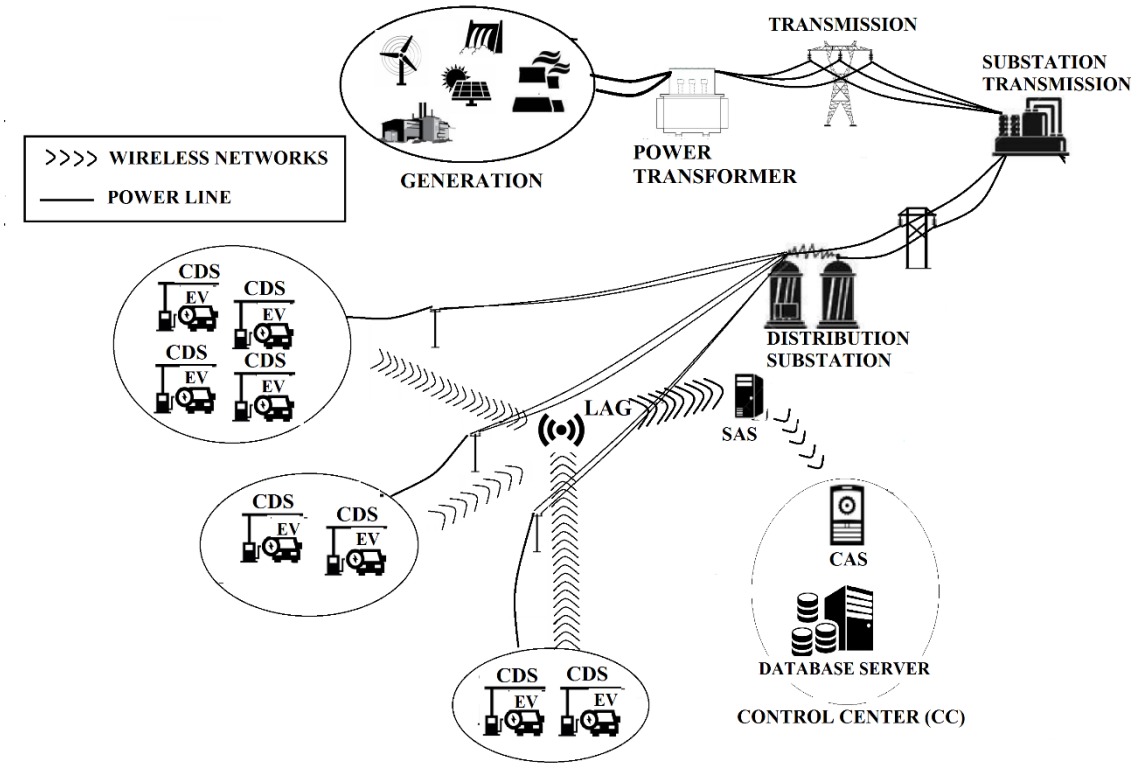


Figure 1 – Architecture of a V2G Network

An EV can charge or discharge its battery in any CDS of the V2G network through the same protocol for both residential and visiting modes. Several CDSs can be connected to a LAG that sends information to SAS. The communications between CDSs and LAGs and between LAGs and SAS commonly occur through wireless networks.

3.2 Preliminaries

3.2.1 Bilinear Pairing

Elliptic curve-based encryption (ECC) must be addressed prior to the understanding of pairing-based cryptography (PBC). ECC has been coined from the use of cyclic groups G conformed by finite points of elliptic curves. Its security effectiveness is based on the discrete logarithm problem, in which given a cyclic additive group $(G, +)$ generated by some element g and given a random element $\in G$, a unique integer must be found, such that $0 \leq b \leq |G| - 1$ and $b = g^a$. In comparison to other asymmetric encryption systems, ECC is more efficient and has a shorter encryption key length [23].

Bilinear pairing was developed as an attack method for finding keys generated by the ECC scheme, taking advantage of the characteristics of some elliptical curves called singular and making the discrete logarithm problem treatable [24].

The academic community has observed a great opportunity for applying bilinear pairing towards creating novel cryptographic schemes. Over the past few years, extensive research on the efficient and safe design and implementation of bilinear pairing has been conducted [23-24].

In general, bilinear pairing is defined as the projection of two points of additive set G formed by points on an elliptic curve E of order $r \in Z_p^+$, towards a same point of a multiplicative set G_T formed by the elements of order $r \in Z_p^+$: $\hat{e} = (G,+) \times (G,+) \rightarrow (G_T,\cdot)$.

The pairing between groups has three properties for all $c, d \in G$

1) Bilinear:

$$\hat{e}(a+c,d) = \hat{e}(c,d)\,\hat{e}(a,d)$$
$$\hat{e}(c,d+a) = \hat{e}(c,d)\,\hat{e}(c,a)$$

2) Non-degenerative:

$$\hat{e}(c,d) \neq 1_{G_T}$$

3) Computationally efficient.

There are different types of bilinear pairings:

- Symmetric: pairing points from the same set to the other set

$$(G_1,+) \times (G_1,+) \rightarrow (G_T,\cdot)$$

- Asymmetric: pairing of points from different sets to the other set $G_1 \neq G_2$

$$(G_1,+) \times (G_2,+) \rightarrow (G_T,\cdot)$$

The following properties of bilinear pairs can be easily verified. For all x, $y \in G$:

1) $\hat{e}(x,\infty) = 1$ e $\hat{e}(\infty,x) = 1$
2) $\hat{e}(c,-d) = \hat{e}(-d,c) = \hat{e}(d,c)^{-1}$
3) $\hat{e}(ac,bd) = \hat{e}(d,c)^{ab}\ for\ all\ a,b \in Z$
4) $\hat{e}(c,d) = \hat{e}(d,c)$

Some applications of bilinear pairing include three-party one-round key agreement, short signatures and identity-based encryption.

3.2.2 Identity-Based Signcryption (IBSC)

IBSC is an identity-based encryption scheme, where each entity has a key pair (private key, public key) generated from the user´s confidential information. A trusted entity that manages the keys of the system entities is required and, once it has them, it can encrypt and sign the messages and decrypt and verify the identity of the messages concomitantly. The proposed protocol uses the IBSC scheme proposed by Li et al. [23] partially to encrypt and validate the sender of some messages.

3.2.3 Short Signatures

A short signature is a digital signature that authenticates messages exchanged in an electronic system and is characterized by its short length in conjunction with other relatively long signature schemes, as RSA and DSA. The implementation of short signatures offers advantages regarding performance in communications in a system [24].

3.2.4 Group management using binary tree.

The use of binary tree by group member management enables the service provider to modify the number of members in a secure way by dynamically changing the group key whenever a member has been added or deleted.

The generation of the binary tree (Figure 2) is based on Parne et al. [25], who proposed the following steps for initializing a group:
1) The system chooses the EVs and the AG that will be members of the group. The selection of the members is based on their location, services and characteristics, among other aspects;
2) The system assigns an identifier of group ID_G to be added to all members of the group;
3) The system creates a binary tree with the group members in the leaves and assigns a group private key to each member; and
4) Group key KGi is computed.

In the proposed group management scheme, two children are assigned to each node in the tree. The EVs in AG are associated with the leaf node and the key value calculated in the root node is the common group key (KGi). The group members use KGi for providing privacy protection and mutual authentication between EV and the service provider. The entire inner node Ni in the binary tree calculates the secret value of the Ki node as:

$$K_i = H(H(K_{left(i)}) \oplus H(K_{righ(i)})) \quad (1)$$

where left(i) and right(i) denote the left and right children of a Ni node, respectively. Function H is a hash function.

The nodes in the path from the leaf nodes (associated with the group members) to the root node are called ancestors and together they form an ancestral set. The leaf nodes also have a set of siblings that are the nodes born from the same parent node. Figure 2 shows the set ancestor and set of siblings of node N11 (EV_3). Each member of the group maintains a group private key (KG_{EV_i} or KG_{AG}) and the associated node has a blinded key ($H(KG_{EV_i})$ or H (KG_{AG})). Each member has a list of blinded keys of the sibling nodes set and the ancestor nodes set along the route from that node to the root for the generation of a group key KG_i. For example, in Figure 2, EV_3 knows blinded key K_{11} and the blinded key of its siblings K_{10}, K_4 and K_3 and, therefore, can calculate all keys in its predecessor set K_5, K_2 and K_1, i.e., the group key (KG_i). This approach maintains the security of the group key.

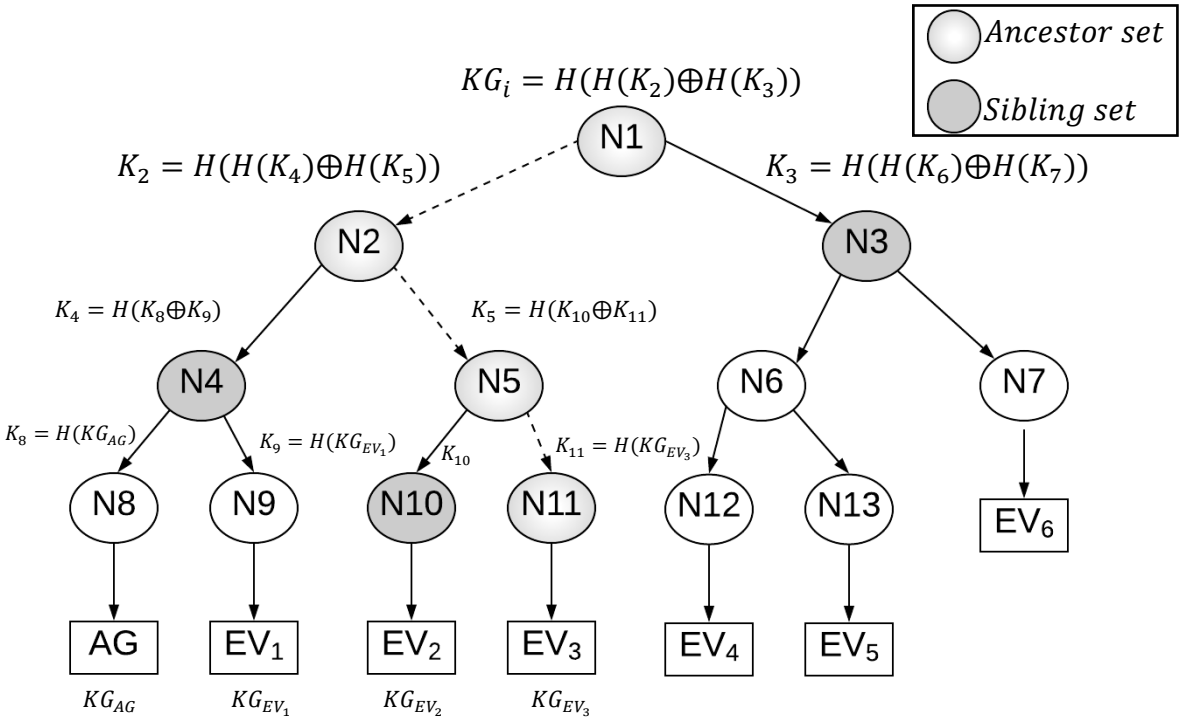


Figure 2. Binary Tree for Group Management

3.3 Description of Phases

Figure 3 shows an overview of the operation of the proposed protocol, as described below:

1. A group of EVs located in a specific area sends a connection request of Charging/Discharging to the aggregator, to be sent to the CDS;
2. The aggregator groups the connection requests of the EVs and sends the connection requests in a group so that the AS validates the identities;
3. In the sequence, there are two possibilities:

a) In case the SAS does not have a registration of the EV, it requests the CAS to authenticate the EV; in case the CAS does not have the user's information, it sends a message to the SAS to disconnect the communication with that user.

b) If the EV is authenticated, the CAS sends necessary information for the connection between EV and SAS; once the EVs have been authenticated, the SAS sends by a secure channel a message to the EVs with the temporary external identity (TEID) as calculated by the AG;

4. The SAS calculates values that will be sent by broadcast, which will allow the EVs and AG to calculate the session key and verify the authenticity of the message.

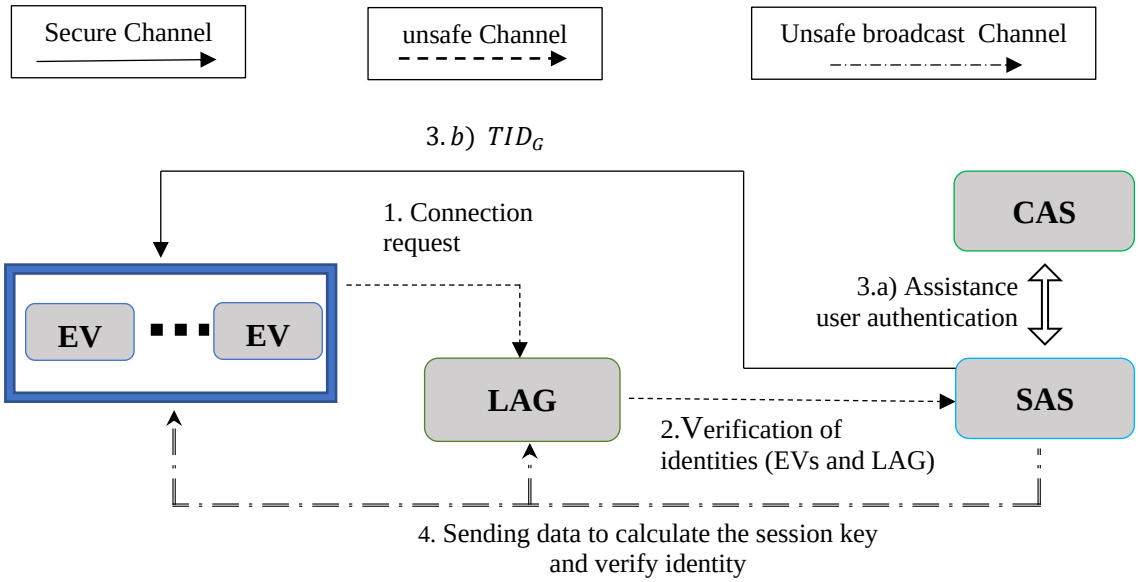


Figure 3. General scheme of the proposed protocol.

The proposed protocol has 3 phases, shown in figure 4, as an extension of previous work by Luis F. A. Roman, Paulo R. L. Gondim and Jaime Lloret [26]:

- Initialization, where the mathematical elements and entities to be used are defined;
- Registration, in which all network entities associate their characteristic data with a public key and are linked to a group;
- Authentication, where some entities not connected to the network try to demonstrate they are a legitimate part of it and, once correctly identified, proceed to use their services through a session. When the use of the service is finished, the session is completed and the entity is disconnected from the network.

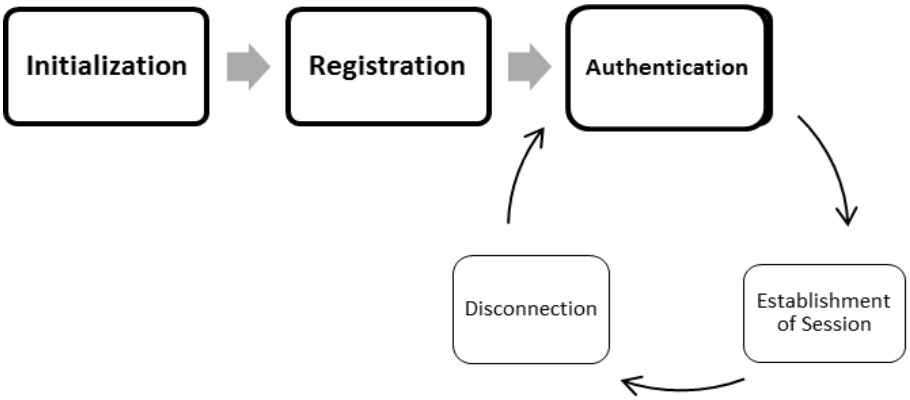


Figure 4. Phases of the Proposed Protocol

The mentioned phases are described in detail below.

1st. phase: **Initialization of the System**

Two cyclic groups G and G_T of order q and P, and a generator element of group G are chosen. G and G_T are supposedly related to a non-degenerative pairing and a bilinear map that can be efficiently computed:

$\hat{e}$: G × G → G_T such that $\hat{e}(P, P) \neq 1G_T$ and $\hat{e}(aP_1, bQ_1) = \hat{e}(b P_1, a Q_1) = \hat{e}(P_1, Q_1)^{ab} \in G_T$ for every $a, b \in Z_q^*$ and every $P_1, Q_1 \in$ G. Moreover, the hash functions of the system are defined: $H_1: \{0,1\}^* \rightarrow G$, $H_2: G \rightarrow \mathbb{Z}_q^*$ and $H_3: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.

Finally, the central authentication server (AS) and all aggregators (AG) define an elliptical curve on a finite field E (Fq) and parameters {G, G_T, $\hat{e}$, P, H_1, H_2, H_3} are published.

AS then chooses a private key $x_{AS}, \in Z_q^*$ and calcultates its public key $Y_{AS} = x_{AS} * P$ to be published.

2nd. phase: **Registration and Group Initialization**

All EVs and AGs must register on-site in the energy supplier´s system. An identity (ID_{AG}) must be chosen for the registration of AG*s*. The aggregator then chooses a random number $x_{AG} \in Z_q^*$ to be its private key and calculates a public key $y_{AG} = x_{AG} * P$. AG sends AS a message containing the public key and the identity of the device $\{y_{AG}, \mathrm{ID_{AG}}\}$. CAS stores the data received y_{AG} and $\mathrm{ID_{AG}}$, and calculates group private key KG_{AG_i} and temporary group identity TID_{AG_i}:

$$KG_{AG_i} = H_1\left(ID_{AG} || y_{AG} || LAI_{AG}\right) * x_{CAS} \quad (2)$$

$$TID_{AG_i} = H_1\left(ID_{AG_i}\right) * H_3(\beta_i), \quad (3)$$

where *LAI (local area identifier)* identifies the area where the aggregator is located and $\beta_i \in Z_q^*$ are random numbers.

The registration of an EV is initialized when it chooses an ID_{EV} identity and an $x_{EV} \in Z_q^*$ private key. It calculates $y_{EV} = x_{EV} * P$ public key. The user sends a message containing the public key and the user´s identity $\{\mathrm{y_{EV}}, \mathrm{ID_{EV}}\}$ to AS through a safe channel. CAS saves the data received, i.e., $\mathrm{y_{EV}}$ and ID_{EV}, associates the EV attributes, as model, make, owner, chassis number and telephone numbers related to the vehicle and chooses random numbers β_{i-j} *and* $V_{i-j} \in Z_q^*$. It then calculates group private key $KG_{EV_{i-j}}$, a temporary identity $TID_{EV_{i-j}}$ and a temporal visitor Identity $TVID_{EV_{i-j}}$.

$$KG_{EV_{i-j}} = H_1(ID_{EV} || \, model || \, make || chassis \; number) * x_{CAS} \quad (4)$$

$$TID_{EV_{i-j}} = H_1\left(ID_{EV_{i-j}}\right) * H_3(\beta_{i-j}), \quad (5)$$

$$TVID_{EV_{i-j}} = H_1(ID_{EV}) * H_3(V_{i-j}). \quad (6)$$

The system creates a user account for a web service in the cloud for the sending of data necessary for the authentication phase. The web service stores the hash of the user's identity $h_{EV} = H_3(ID_{EV})$, and requires the user must change the password in the first access.

CAS initializes the group as follows:

- it defines the EVs and AG that will be part of the group and its identity ID_{G_i};
- generates a binary tree where leaves are the private group keys $(KG_{EV_{i-j}}, KG_{AG_i})$ of each entity in the group (see Figure 2); and
- computes group key KG_i.

Finally, it sends the group private key $(KG_{EV_{i-j}}, KG_{AG_i})$ and a list of the blinded keys of its siblings $(LS = K_a, K_b, \dots, K_z)$ to calculate group key KG_i, random numbers β_{i-j} and V_{i-j} for each EV and random number β_i for AG to calculate the temporary identifications. It then sends all necessary data $(TIDs, KG\_EV, KG_{AG}, KG_i, Y_{EV_{i-j}}, Y_{AG_i})$ to SAS for it to authenticate the group members.

Figure 5 shows a summary of the registration phase, where continuous arrows represent the sending of messages through secure channels. In Figure 6 and the 3rd. phase, dotted arrows represent the sending of messages through unsafe channels.

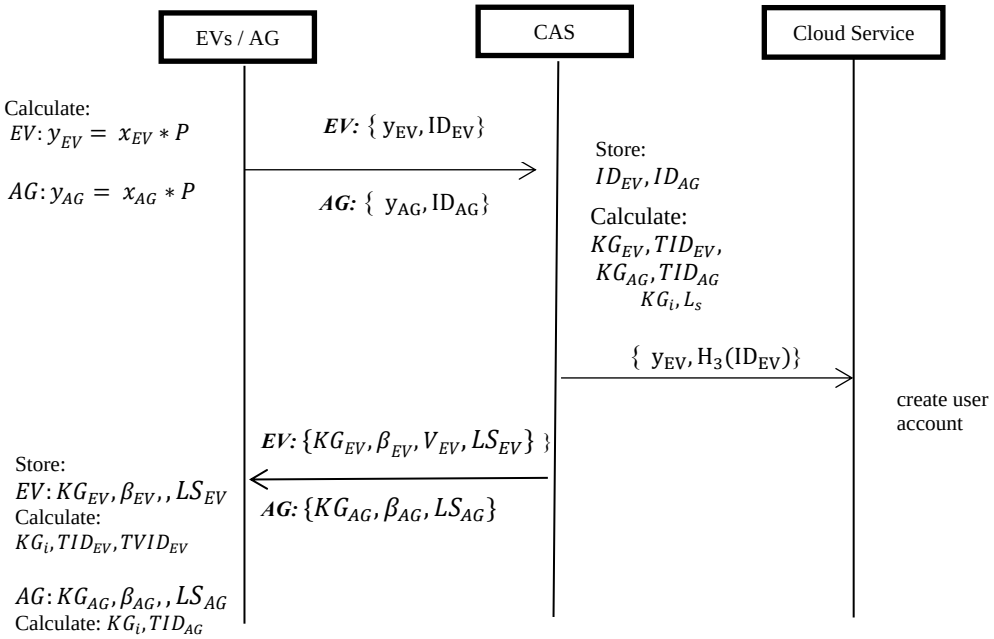


Figure 5 – Registration and Group Initialization Phase

3rd. phase: **Authentication of EV and AG**

In the authentication phase, the proposed protocol exchanges four messages:

1) $\boldsymbol{EV}$ $\{\mathrm{MG}_{\mathrm{EV_{i-j}}}, MAC_{i-j}\}$ $\boldsymbol{AG}$

The EV_j calculates the following values:

$$TID_{EV_{i-j}} = H_1\left(ID_{EV_{i-j}}\right) * H_3(\beta_{i-j}). \quad (7)$$

$$M_{EV_{i-j}} = \left\{KG_i \,||LAI_{i-j}\right\}_{\mathrm{KG_{EV_{i-j}}}} \quad (8)$$

$$MG_{EV_{i-j}} = (M_{EV_{i-j}}||TID_{EV_{i-j}}) \quad (9)$$

$$MAC_{EV_{i-j}} = h_3\left(MG_{EV_{i-j}}\right) \quad (10)$$

where $MAC_{EV_{i-j}}$ is the message authentication code and $MG_{EV_{i-j}}$ contains the group key and location area identifier (LAI), encrypted with the group private key and EV_{i-j} temporary identity. Message $\{\mathrm{MG}_{\mathrm{EV_{i-j}}}, MAC_{i-j}\}$ is sent to AG_i.

2) $\boldsymbol{AG}$ $\{MC_{G_i}, MAC_{G_i}\}$ $\boldsymbol{SAS}$

AG_i verifies the authentication code of the message ($MAC_{EV_{i-j}}$), with the authentication code it calculated ($MAC'_{EV_{i-j}}$):

$$MAC_{EV_{i-j}} = MAC'_{EV_{i-j}} = h_3\left(MC_{EV_{i-j}}\right) \quad (11)$$

If the comparison is satisfactory, AG_i adds message $M_{EV_{i-j}}$ and $TID_{EV_{i-j}}$ to a group message M_{G_i}:

$$M_{G_i} = (M_{EV_{i-1}}||TID_{EV_{i-1}}|| \ldots ||M_{EV_{i-j}} \left|\left|TID_{EV_{i-j}}\right|\right| \ldots \left|\left|M_{EV_{i-n}}\right|\right| TID_{EV_{i-n}}) \quad (12)$$

Otherwise, the connection with EV is terminated.

Finally, the aggregator chooses value $v_{G_i} \in Z_q^*$, and calculates

$$TID_{AG_i} = H_1\left(ID_{AG_i}\right) * H_3(\beta_i) \quad (13)$$

$$M_{AG_i} = \{ID_{AG_i}||LAI_{AG}\}_{KG_{AG_i}} \quad (14)$$

$$TID_{G_i} = H_1\left(ID_{G_i}\right) * v_{G_i} \quad (15)$$

$$MAC_{AG_i} = h_2\left(M_{AG_i}||TID_{AG_i}\right) \quad (16)$$

It adds its message M_{AG_i} and TID_{AG_i} to group M_{G_i} and calculates the authentication message of group MAC_G.

$$M_{G_i} = \left\{\left\{M_{EV_{i-1}} \left|\left|TID_{EV_{i-1}}\right|\right| \ldots \left|\left|M_{EV_{i-n}}\right|\right| TID_{EV_{i-n}} \left|\left|M_{AG_i}\right|\right| v_{G_i}\right\}_{KG_i} ||TID_{AG_i}\right\} \quad (17)$$

$$MAC_{G_i} = \left(MAC_{AG_i} \oplus MAC_{EV_{i-1}} \oplus MAC_{EV_{i-2}} \oplus \ldots \oplus MAC_{EV_{i-n}}\right) \quad (18)$$

Message group M_{G_i} and MAC_{G_i} are immediately sent to AS.

3) **AG** $\{\varphi, X_1, X_2, t_4\}$ **SAS**

<----------

AS decrypt the message with the group key and calculates MAC'_{AG_i} and all $MAC'_{EV_{i-j}}$ and the total MAC of the message

$$MAC'_{G_i} = \left(MAC'_{AG_i} \oplus MAC'_{EV_{i-1}} \oplus MAC'_{EV_{i-2}} \oplus \ldots \oplus MAC'_{EV_{i-j}}\right) \quad (19)$$

for checking the integrity of all messages with the following comparison: $MAC_{G_i} = MAC'_{G_i}$. If the verification fails, AS sends a MAC failure message to the group. Otherwise, decrypts the messages with the group private keys of each of the EVs and AG, and verifies the identities and location. After, it chooses a random number $v_{AS1}, v_{AS2}, r \in Z_p^*$ and calculates a temporary identity and a temporary key for the group.

$$TID_{G_i} = H_1(ID_{G_i}) * v_{G_i} \quad (20)$$

$$TKG_i = h_2(KG_i || v_{AS1}) \quad (21)$$

The TID_{G_i} is sent to the cloud web service with an account associated with the user. The user can join the cloud service through an application on the cell phone, computer, or with a user interface installed in the CDS; this latter feature ensures that the EV owner could join their account to acquire TID_{G_i}, in a situation where, for some reason, he does not have a device with Internet access to join the service in the cloud. Then, once the TID_{G_i} were obtained, some values for a broadcast message are calculated, according to Table 1.

Table 1. Calculation of Values for a Broadcast Message.

$F = v_{AS2} * TKG_i * y_{SAS}$	$X_2 = r * y_{SAS}$
$X_1 = r * TID_{G_i}$	$h = H_1(X_1 \|\| X_2 \|\| TKG_i)$
$w_1 = r * H_2(TID_{G_i}) * y_{SAS}$	$w_2 = r * KG_i * TID_{Gi}$
$z = H_2(h + TKG_i + w_1)$	

$$\varphi = H_2(w_1||w_2) \oplus (z \left|\left| ID_{G_i} \right|\right| v_{AS1}||F)$$

AS sends a broadcast message $\{\varphi, \mathrm{X}_1, \mathrm{X}_2, \mathrm{t}_4\}$, where t_4 is a timestamp, to all group members and calculates the session keys and the hash of each EV_{i-j} and AG_i. The operations are shown in Table 2.

Table 2. Calculation of the SAS Session Keys.

$\boldsymbol{EV_{i-j}}$	$\boldsymbol{AG_i}$
$Ks'_{i-j} = \hat{e}\left(TID_{EV_{i-j}}, F\right) \hat{e}\left(x_{SAS}, v_{sp2} * TKG_i * y_{EV_{i-j}}\right)$	$Ks'_i = \hat{e}\left(TID_{AG_i}, F\right) \hat{e}\left(x_{SAS}, v_{sp2} * TKG_i * y_{AG_i}\right)$
$Hks_{i-j} = H_2(Ks_{i-j})$	$Hks_i = H_2(Ks_i)$
$Mk_{i-j} = (Hks_{i-j}\|\|\mathrm{TID}_{\mathrm{EV_{i-j}}})$	$Mk_i = (Hks_i\|\|\mathrm{TID}_{\mathrm{AG_i}})$

4) **EV/AG** $\{Mk_{i-j}\}\{Mk_{i-j}\}$ **SAS**

When EVs and AG_i receive the message from SAS, they calculate the following values: $w'_1 = H_1(TID_{G_i}) * X_2$; $w'_2 = X_1 * KG_i$. Then the EVs and the AG perform an ***xor*** operation to extract the parameters to calculate the session key and check the message sent by SAS.

$$\varphi \oplus \mathrm{H}_2(\mathrm{w'}_1||\mathrm{w'}_2) = (z \left|\left| ID_{G_i} \right|\right| v_{SAS1}||F) \quad (22)$$

With z, ID_{G_i}, v_{AS1} and F values found in the message, EV and AG do the following actions:

- Verification of the message:

To check the message sent by SAS, the EVs and the AG must calculate $TKG_i' = H_2\left(TID_{G_i}||v_{AS1}\right)$ and $h' = H_1(X_1\left|\right|X_2\left|\right|TKG_i)$, where X_1 and X_2 are the values received in the message and TKG_i is the group key found in the message. EV must then verify $z' = H_2(h' + TKG_i' + w_1)$

If the verification succeeds, EV_s and AG_i calculate the session key; otherwise, they close communication.

- Session key

The EVs and the AG must use the following elements to calculate the session key:

- Private Keys
- Random values generated
- Value obtained from the message sent by SAS (v_{AS1})
- Identification value of the group (TID_{G_i})

Once the session key is generated, the EVs and AG calculate a hash of that key and form a message that contains the temporal identity (EVs or AG) and the session key hash. This message is encrypted by an XOR operation with the group key. The operations described above are shown in Table 3:

Table 3. Calculation of EVs and AG Session Keys.

$\boldsymbol{EV_{i-j}}$	$\boldsymbol{AG_i}$
$Ks_{i-j} = \hat{e}\left(\left(TID_{EV_{i-j}} + x_{EV_{i-j}}\right), F\right)$	$Ks_i = \hat{e}\left(\left(TID_{AG_i} + x_{AG_i}\right), F\right)$
$Hks_{i-j} = H_1(Ks_{i-j})$	$Hks_i = H_1(Ks_i)$
$Mk_{i-j} = (Hks_{i-j}\|\|\mathrm{TID}_{\mathrm{EV_{i-j}}})\oplus TKG_i$	$Mk_i = (Hks_i\|\|\mathrm{TID}_{\mathrm{AG_i}})\oplus TKG_i$

The encrypted messages of EV ($\{Mk_{i-j}\}$) and AG $\{Mk_i\}$ are sent to the AS for verification.

AS immediately receives the messages from each EV_{i-j} and AG_i, groups them and calculates their MAC'_{Mk_i}, groups them and calculates MAC_{Mk_i} of the keys and temporal identity's calculated by SAS:

$$MAC'_{Mk_i} = H_2((Hks'_i||Hks'_{i-1}||Hks'_{i-2}|| \dots || Mks'_{i-j})\oplus TKG'_i) \quad (23)$$

$$MAC_{Mk_i} = H_2((Hks_i||Hks_{i-1}||Hks_{i-2}|| \dots || Mks_{i-j})\oplus TKG_i) \quad (24)$$

If $MAC'_{Mk_i} = MAC_{Mk_i}$ are the same, all group members have the correct session key, therefore, communication is established. On the other hand, if the verification fails, SAS checks, one by one, the ***Hash*** of the keys sent. When it finds the wrong key, it closes communication with this member and creates a new temporary group key, which is sent to each member in an encrypted mode with the session key established.

Below is the mathematical proof of the establishment of the session keys.

$$Ks_{i-j} = \hat{e}\left(\left(TID_{EV_{i-j}} + x_{EV}\right), F\right) \quad (25)$$

$$= \hat{e}\left(TID_{EV_{i-j}}, F\right)\hat{e}\left(x_{EV_{i-j}}, v_{sp2} * TKG_i * Y_{SAS}\right) \quad (26)$$

$$= \hat{e}\left(TID_{EV_{i-j}}, F\right)\hat{e}\left(x_{EV_{i-j}}, v_{sp2} * TKG_i * x_{SAS} * P\right) \quad (27)$$

$$= \hat{e}\left(TID_{EV_{i-j}}, F\right)\hat{e}\left(x_{SAS}, v_{sp2} * TKG_i * x_{EV_{i-j}} * P\right) \quad (28)$$

$$= \hat{e}\left(TID_{EV_{i-j}}, F\right)\hat{e}\left(x_{SAS}, v_{sp2} * TKG_i * Y_{EV_{i-j}}\right) \quad (29)$$

Figure 6 shows the flow of messages exchanged among the entities in the authentication phase.

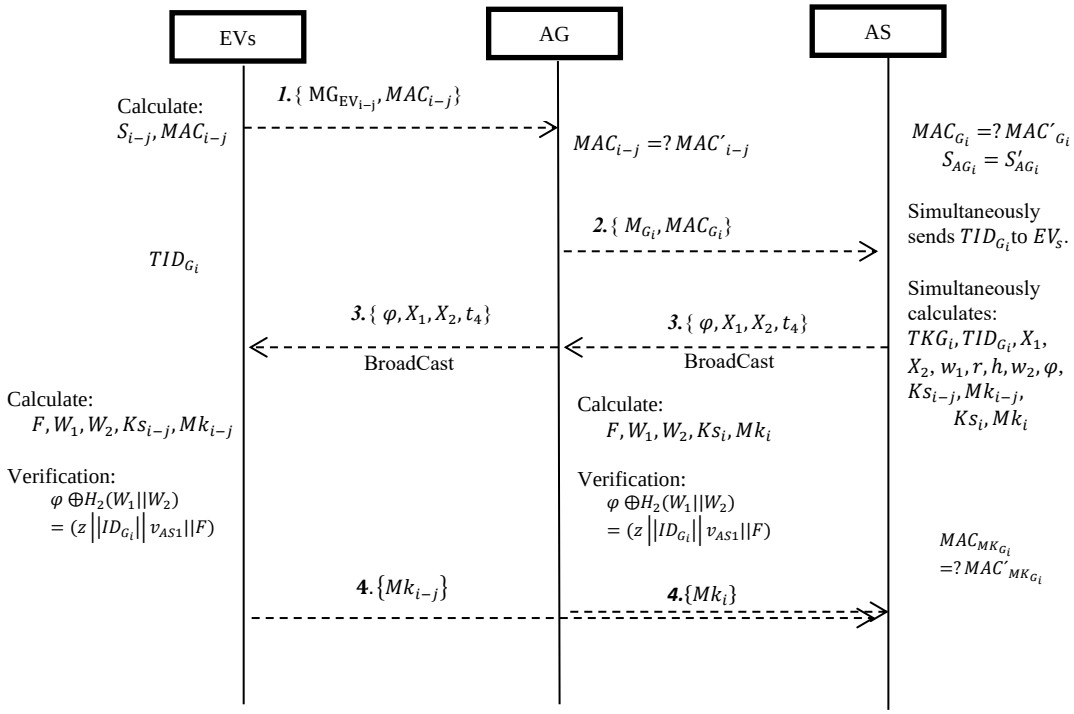


Figure 6. Authentication Phase.

3.4 Members Joining/Leaving a Group:

Group management with binary tree guarantees security by updating the group key whenever a new member enters or is removed from the group. All members update information on the new blinded keys calculated along the route affected by the member's entry or removal for updating the group key individually. Below are the details of the join and leave operations of EVs [25]:

- EV Joins a group:

Whenever a new EV joins a group, it is assigned to the leaf node of a binary tree. When the leaf node becomes the parent of two leaf nodes, the member associated with the parent node is associated with the new left leaf node and the new EV is associated with the new right leaf node. A new group key is then generated.

For example, if EV_5 aims at joining a group (Fig. 7), leaf node N6 becomes a parent and creates two leaf nodes (N12 and N13). EV_4 associated with node N6 is associated with leaf node N12 and the new member of group EV_5 is associated with leaf node N13. The updated key value of Node N6 affects all nodes along their route to the root node.

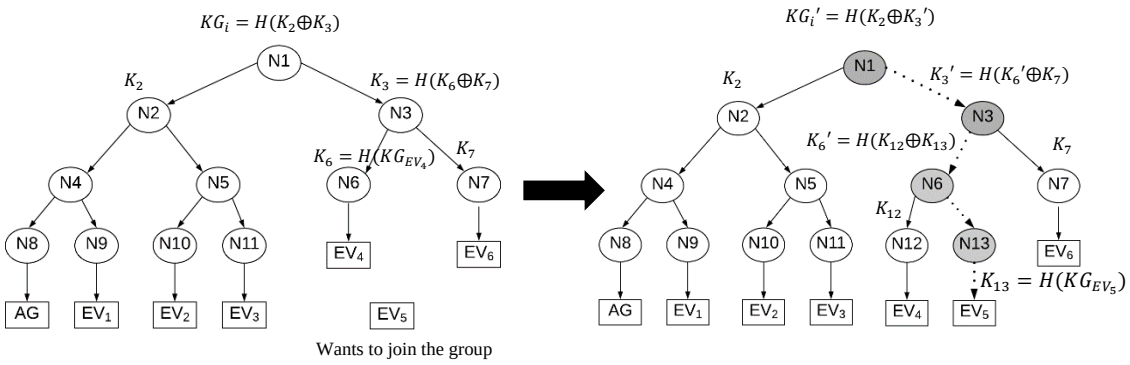


Figure 7. EV joining a group.

- EV Leaves a group:

The group key must be changed whenever an EV leaves a group (see Fig. 7, adapted from [25]). Both leaf node associated with outgoing EV and the sibling leaf node are eliminated. The EV associated with the sibling node of the deleted leaf is assigned to the parent node, which becomes a leaf node. The value of the EV group private key of the node that remained in the tree is modified and, consequently, the blinded keys of the nodes of the trail up to the root are updated. New values are then secretly transmitted to their EVs, which calculate the new group key.

For example, if EV_1 aims at leaving the group (see Figure 8), leaf nodes N8 and N9 are eliminated and the member of group AG is associated with the new leaf node N4 (it was previously a parent node of N8 and N9). SAS sends a new group private key to AG and calculates a new value for N4 through a message encrypted with the group private key. All blinded keys of the nodes along their route to the root node are then updated and sent safely (with group private keys) to each EV for the calculation of the new group key.

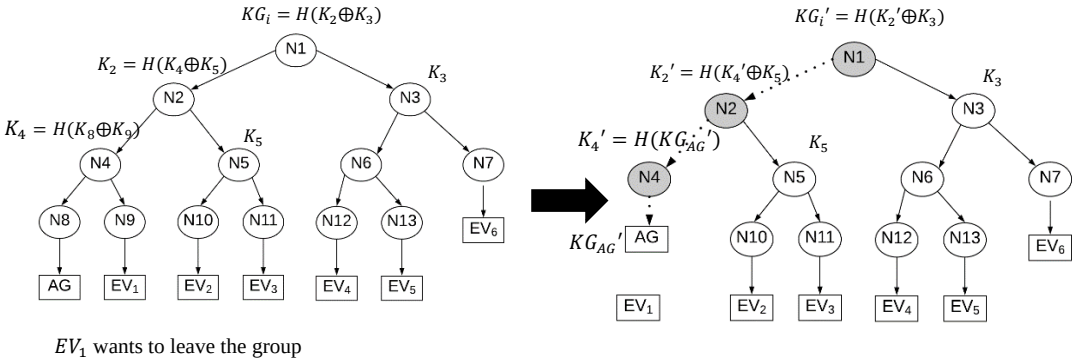


Figure 8. EV leaving a group.

3.5 Modes and Functionalities of EVs:

The proposed protocol can support authentication in both residential and visiting modes, once the hierarchic distribution of AS enables the authentication of EVs anywhere. The protocol authenticates the visiting EVs individually, however, with an equivalent TID_G called Temporary Visitor Identity (TVID) provided by the CAS in the registration phase, for ensuring safety of the resident EV group; additionally, an individual temporary identifier (TID_i) is sent to the cloud web service with an account associated with the user. With the TID_i the visiting EV will compute an individual temporary key (TK_i), which will be required to validate the SAS identity and generate session key.

It also can operate in different situations/cases, such as storer, provider, consumer and seller, where the interaction of an EV in the connection with V2G occurs as described below:

- Energy Storer: when CC detects power plants are producing more energy than that demanded in a certain area, it sends a broadcast message to the EVs group through *AS* and LAG of the area for them to purchase such energy for the avoidance of loss. If an EV wishes to purchase the energy, it must only respond to *AS* with a message containing the EV temporal identity encrypted with the group key. The remaining communication will be established with the session key of each EV;
- Energy Provider: when CC detects power plants are producing less energy than that demanded in a certain area, is sends a broadcast message to the group of EVs through *AS* and LAG of the area for them to sell part of their energy for the avoidance of overcharge in the power plant. If an EV wishes to sell energy, it must only respond to *AS* with a message containing the EV temporal identity encrypted with the group key. The remaining communication will be established with the session key of each EV;
- Energy Consumer or Seller: when EV approaches an CDS to charge or discharge its battery, an encrypted communication is established with CC through a session key employing AS.

Below is a comparative table of the entities that compose the V2G architecture of the above-mentioned studies and the protocol proposed.

Table 4 shows the difference among the entities of the architecture proposed in this paper and those proposed by Jie et al. [12] and Saxena et al. [13]. According to those authors, aggregators perform most tasks of verification of messages and authentication of EVs, consequently, LAG must have high processing power for avoiding overcharge. Conversely, as the authentication server of the proposed protocol has high processing power, the aggregator is used only for grouping messages and reducing communication costs, which results in a more flexible V2G network.

Table 4. Comparisons of Entities of V2G Architectures

Entities	Jie et al. [12]	Saxena et al. [13]	Proposed protocol
EV	✓	✓	✓
ST/CE/CDS	✓	✓	✓
LAG	✓	✓	✓
CAG	✓	--	--
SAS	--	--	✓
CAS	--	--	✓
TA/TTP/CA/RA	✓	✓	--
CC	--	✓	✓
Total Number of Entities	**5 Entities**	**5 Entities**	**6 Entities**

4. Security and Performance Analyses

This section reports on an analysis of the security and performance of the proposed protocol and a comparison with the other protocols used for authentication of a V2G system.

4.1. Security Analysis

4.1.1 Security Properties

Below is a description of the processes related to authentication, preservation of privacy and integrity and analytical evaluation of the resistance of the proposed protocol to attacks [26].

1) Mutual Authentication: Mutual Authentication is established among EVs, AG and AS. AS authenticates AG and EVs through the use in the authentication phase of the pre-shared keys (KG_i, KG_{AG_i}, $KG_{EV_{i-j}}$) in the registration phase. EVs authenticate AG and AS by means of *token* TID_{G_i} in the calculation phase of the group´s temporal key through a pairing operation of the message sent by AS.
2) Preservation of privacy: The identity of the EV is kept confidential by the authentication servers; the other entities of the V2G network know only the temporary identity of EV ($TID_{EV_{i-j}}$). The location privacy is also guaranteed in both residential and guest modes. The use of encrypted messages in the residential mode ensures only SAS can decipher the location of the vehicle. Such a location is important for the tracking and establishment of responsibilities in case of security incidents. Regarding the visitor mode, Section 4.1.2 is devoted to analyses of the preservation of privacy.
3) Protection to integrity: The integrity of the messages exchanged is maintained with the MAC generation. An adversary cannot make changes to an intercepted

message without the MAC value changing, so the system would identify if a message was manipulated.

4) Prevention against attacks: we will describe the different types of attacks that can affect the V2G network and how the proposed protocol can resist them:

- Impersonation: an attacker that aims at impersonating a valid EV must know its the identity and secret key. However, parameter $\mathrm{TID}_{\mathrm{EV_{i-j}}}$ or $\mathrm{TID}_{\mathrm{AG_i}}$ cannot be obtained without the secret keys of the involved entities. A session key is generated whenever an EV_s is authenticated for the avoidance of use of old parameters in other devices;

- MITM: after receiving a message from AG, AS sends to EVs an One Time Password (OTP) through another channel to check the identity of EVs towards protecting the system from such an attack. $\mathrm{EV_s}$ must perform operations with both the values contained in the message received and the OTP (TID_i) sent by the server for obtaining the session key and validating the identity of AG_{AG_i} and AS;

- Replay and Injection: an attacker can intercept a message to carry out a repetition attack and inject data in the message. Therefore, random numbers chosen for each session, as TKG,v,TID_i,ks are implemented and $hash$ functions check the integrity of the message;

- Redirection: whenever a new EV tries to access the system, it is associated with a group attended by an AG_i. If the same user tries a second access to either the same group, or a different one, AS rejects the second connection;

- Known key: the proposed protocol generates temporary identities and sends an OTP (TID_{G_i}) to the EV to calculate a key for each session, so that an attacker cannot use old keys or data to establish a communication.

- DoS: The Server will enable a valid EV to access the V2G network by calculating the $\mathrm{TID}_{\mathrm{EV_{i-j}}}$. If more than one session is requested, the server checks the location of the request and if differences between AG_i of the requests sent by the same user are detected, the system rejects the communication of this user to avoid even DDoS attacks

4.1.2 Preservation of privacy

Several metrics have been adopted for assessments of privacy, e.g., anonymity set and entropy, used to measure uncertainty. We will consider mutual information, an entropy-based metric that quantifies the information shared between two random variables and measures the amount of information leaked from a privacy mechanism [27].
The privacy properties of the proposed protocol will be evaluated regarding privacy of users' identity and location.
The following privacy protection mechanisms were provided in the design of the protocol:
• protection of the EV identity with respect to SAS, since SAS cannot know the real identity of the EV, due to the use of temporary identities;
• protection of identity and location with respect to an attacker, since all messages with confidential information are encrypted and the use of two identities (residential and visitor) prevents the EV from being traced on the network; and

• protection of location with respect to SAS, since SAS does not have enough information to connect the two identities (residential and visitor) of the same EV.

We consider a scenario where an EV is commonly served by an aggregator and, due to its movement, another aggregator can be accessed (visited). Below is an analysis of the ability of SAS to correlate a local temporary identity (TID) of a particular vehicle with a temporary visiting identity (TVID). The analysis is based on the technique used by Eiza et al. [28] for a further exploration of the relationship between TID and TVID.

Let us assume N EVs grouped in a single SAS system that manages two aggregators $AG_{resident}$ and $AG_{visitant}$,. A subset E of EVs ($1 \leq |E| \leq |N|$) may recharge in the $AG_{visitant}$ zone.

Poisson distribution was assumed for modeling the number of vehicles that arrived at $AG_{visitant}$ in a given duration of time at λ rate.

Let (A) and (B) be two discrete random variables with marginal probability functions p(A) and p(B), respectively. (A) represents the probability of EV_1 with TID_1 not recharging in $AG_{resident}$, whereas B denotes the probability of EV_1 reloading in $AG_{visitant}$ with a $TVID_1$ identity.

The probabilities follow a uniform distribution for all EVs of set E. The metric used in the evaluation of the degree of privacy is called mutual information (MI) *I (B; A).* In our case, it is used to measure the uncertainty of SAS when the EV1 with local temporary identity TID_1 loads into $AG_{visitant}$ with $TVID_1$. *I (B; A)* is defined as:

$$I(B;A) = H(B) - H(B|A) \tag{30}$$

where H (B) measures the amount of information SAS has regarding B and H(B | A) is the conditional entropy that measures the amount of information necessary for SAS to describe B, since the value of A is known. Using probability notations p(a) and p(b), the previous equation can be rewritten as:

$$I(B;A) = \sum_b p(b) \log_2 p(b) - \sum_b p(a,b) \log_2 \frac{p(a)}{p(a,b)} \tag{31}$$

where p (a, b) is the joint probability distribution function of A and B. Since a uniform distribution is considered, $p(a) = \frac{1}{|E|}$ is defined as the probability of EV1 not recharging in $AG_{resident}$.

The probability of EV_1 recharging in the zone of $AG_{visitant}$ despite belonging to the zone serviced by $AG_{resident}$ is given by $p(b) = \frac{1}{|E|} . \frac{1}{\lambda t+1}$, where λt is the average number of arrivals per *t* units.

Figure 9 shows the amount of uncertainty reduction over B in relation to the size of E and mean arrival rate λ when *t* is set to 1 second.

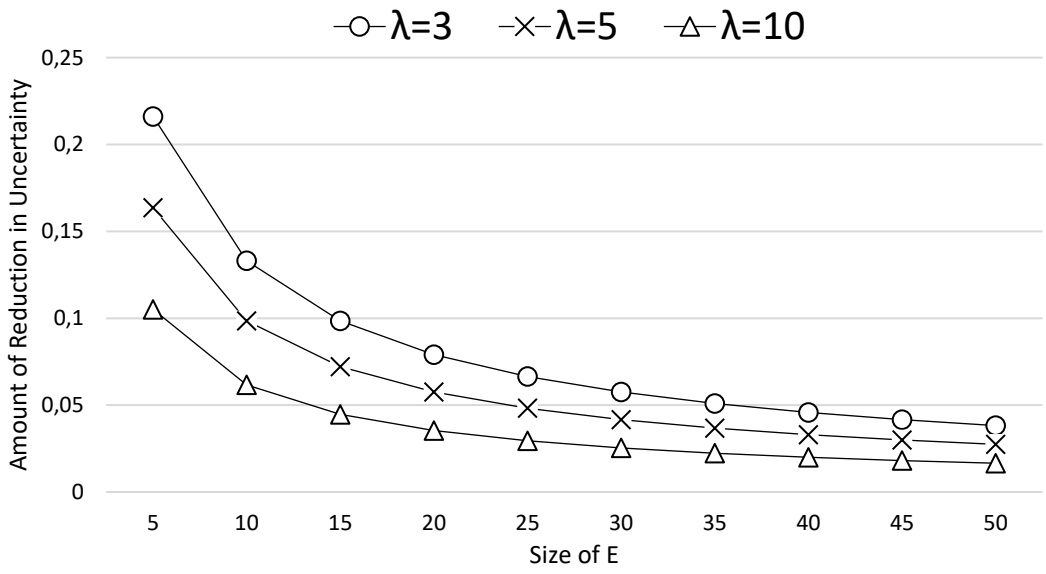


Figure 9. Amount of Reduction in Uncertainty

According to Figure 9, the amount of reduction in uncertainty decreases when both size of E and arrival rate λ increase for SAS. Similarly to the result obtained in [28] in the context of mobility management protocols, SAS remains uncertain regarding whether TID_1 and $TVID_1$ belong to the same EV1 in an architecture that has only two AGs and one SAS. Therefore, the proposed protocol guarantees a high level of anonymity for EVs that aim at recharging in other areas.

4.2. Performance Analysis

This subsection addresses an analytical evaluation of the communication and computational costs of the proposed protocol and a comparison with the other protocols cited.

a) Communication Cost

Communication cost refers to the total number of bits transmitted by a network during the execution of the protocol. The same table of values from Saxena et al. [13], showed in Table 5, was used for providing an adequate comparison with other protocols.

Table 6 shows a comparison of the communication costs by entities for a group of n EVs connected to an AG. Such costs were measured in bits using Table 5.

Table 5. Symbols and Cost in bits [13]

Symbol	Description	Length (bits)
Name	User´s name	128
ID	User´s identification	128
TID/TVID	Temporary identity / Temporary Visiting ID	128
$H()$	Hash function	64
x	Private key	128
y	Public key	128
k	Session key	128
KG	Group Key	128
Role	User´s role	64
β_i / V_i	Random values	16
LAI	Local Area Identifier	40
t	Timestamp	64
*	Multiplication operator	-
$\hat{e}$	Bilinear Pairing	-
SAS	Authentication Server of the substation	-
CAS	Central Authentication Server	-
MAC	Message authentication code	64
P	Point of the elliptical curve	128
$\oplus$	XOR operator	-

Table 6. Communication Cost in bits per message

	M1	M2	M3	M4	M5	M6	M7	M8	TOTAL
Jie et al. [12]	257n	64n	128n	256n	128n	128n	128n	192n	**1281n**
Saxena et al. [13]	384n	704n	320n	128n+320	-	-	-	-	**1536n+320**
Proposed Protocol	376n	312n+376	704	192n+192	-	-	-	-	**880n+1272**

The total communication cost of the proposed protocol is 880(n) + 1272 bits for **n** EVs per aggregator. According to Table 6, the protocol achieves better communication performance than the protocol designed by Jie et al. [12] for a number of EVs higher than 4, and better performance than the protocol of Saxena et al. [13] for a number of EVs higher than or equal to 1,45, i.e., approximately 2.

Figure 10 shows graphs of the communication costs of the proposed protocol and the protocols proposed by Jie et al. [12] and Saxena et al. [13]. The communication costs of all protocols increase linearly according to the number of EVs. The superior performance of our protocol in aggregators with medium or high number of EVs is clearly demonstrated.

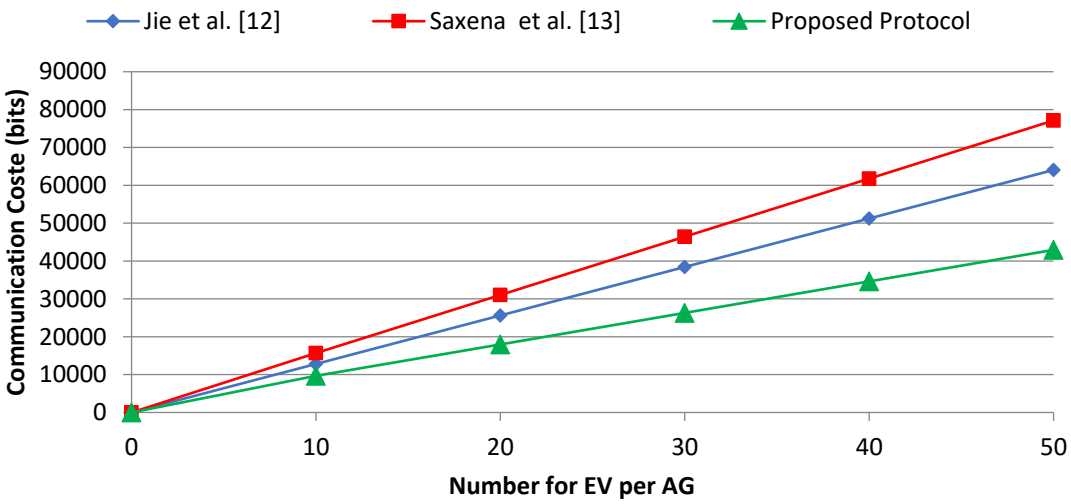


Figure 10 - Communication Costs of the Protocols

b) Computational Cost

Here is made a comparison of the computational costs of our protocol and the protocols proposed by Jie et al. [12] and Saxena et al. [13]. The run-time values of the Multiplication (T_{mul}), Exponentiation (T_{exp}) and Bilinear Pairing (T_{pair}) functions are based on Tao et al. [29], shown in Table 7, and processing parameters of involved entities. The time costs of operations, such as hash functions, symmetric encryption / decryption, XOR, Message Authentication Code (MAC), and addition, will be omitted because their execution times are very short [19].

Table 7. Cost in ms of each operation and entity considered [29]

Entity	Performance parameters of involved entities			costs (ms)		
	CPU(GHz)	RAM	OS	T_{mul}	T_{exp}	T_{pair}
EV	Qualcomm(R) Octa-core 1.5	2	Android 4.2.2	0,54	0,5	16,6
LAG	Intel(R) Dual-core 3.1	4	64-bit Win-7	0,36	0,38	11,5
ST/CS	Intel(R) Hexa-core 1.6	16	16 Win server 2012	0,3	0,31	8,6
AS/CA/RA	Intel(R) Hexa-core 1.6	16	16 Win server 2012	0,3	0,31	8,6

- Costs of the authentication phase and generation of keys

According to Table 8, the largest number of operations of the proposed protocol is concentrated on the entity of best computational properties, i.e., *AS*. Such a characteristic offers better performance and flexibility to the V2G network and avoids the overload of operations in elements of limited resources.

Table 8. Computational Cost of the Authentication Phase

Protocol	Jie et. al[12]	Saxena et. al [13]	**Proposed protocol**
EV/PEV	$3nT_{mul} + nT_{pair}$ nT_{exp}	$nT_{mul} + nT_{pair}$ $+ 3nT_{exp}$	$3nT_{mul} + nT_{pair}$
ST/CS	$nT_{mul} +$ nT_{pair}	--	--
LAG	$(n+1)T_{mul}$ $+ nT_{pair} + nT_{exp}$	$(n+1)T_{mul}$ $+nT_{pair} +$ $+ 5T_{exp}$	$3T_{mul}$ $+ 1T_{pair}$
AS	--	--	$(2n+11)T_{mul}$ $+(n+1)T_{pair}$
CA/RA	--	$(3n)T_{mul} +$ $(3n)T_{exp}$	--

Figure 11 shows a comparison of the total computational cost of the authentication phase of the proposed protocol and the protocols of Jie et al. [12] and Saxena et al. [13]. Our protocol also provides better computational performance than the protocols designed by Jie et al [12] and Saxena et al. [13].

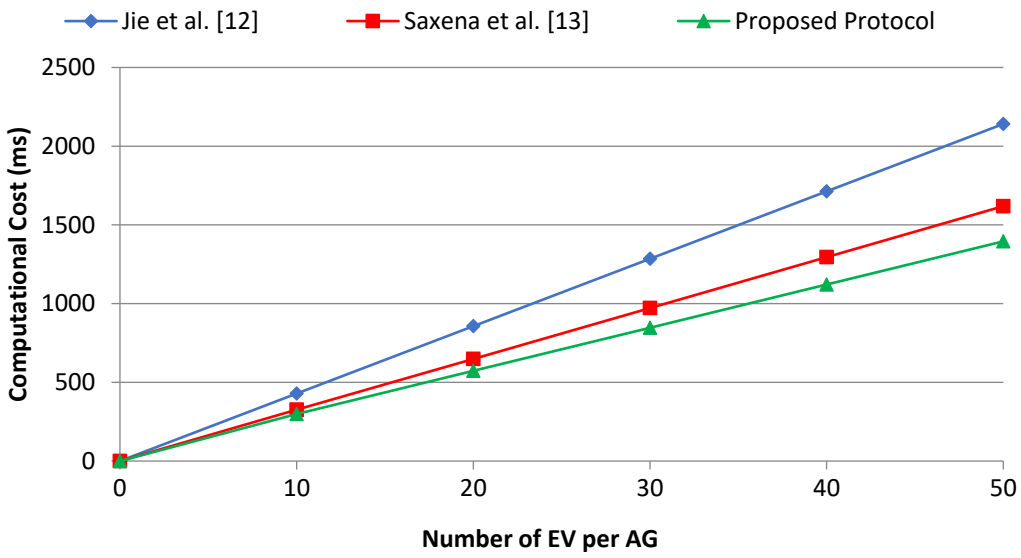
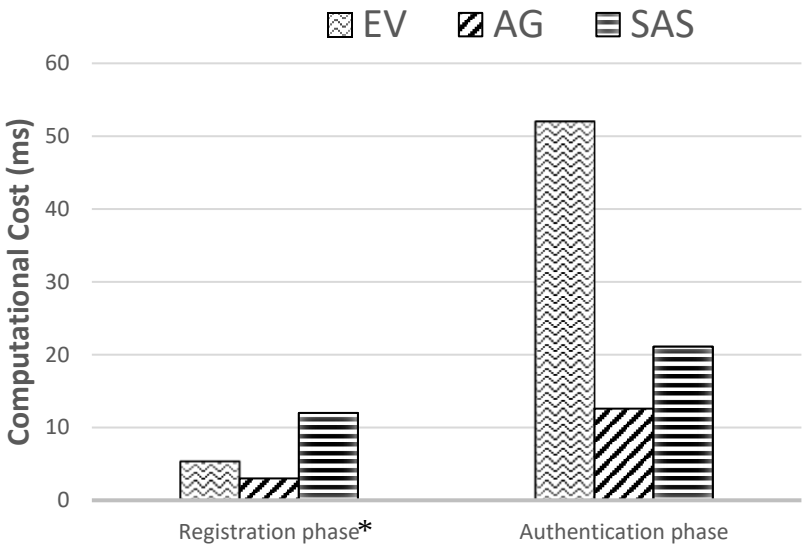


Figure 11. Comparison of Computational Costs among Protocols

- Computational Cost per Entity of the Proposed Protocol

Figure 12 shows the comparison of the computational costs of the entities of the proposed protocol when an EV registers and authenticates in the V2G system. In the registration phase the EV, AG and SAS have the same number of operations to execute, but the computational cost of the EV is higher, since its processing power is lower than AG and SAS. AG has a lower cost because has fewer operations to do, compared to SAS.

In the authentication phase the computational cost raises considerably, due to the number of operations on each entity, as shown in Tables 8 and 9 for n=1. AG has the lowest computational cost, just for grouping the information of the EVs connected to it and sends that information to the SAS, so the number of operations that it executes are smaller in with the EVs and SAS. SAS concentrates several tasks involved in this phase, thus its computational cost is the largest.



* the original values were multiplied by 10 to better visualize the differences.

Figure 12. Computational Cost in Registration and Authentication phases, for entities of the proposed protocol

c) Storage Cost

The next step was to compare the storage cost of the proposed protocol and the proposed ones by Jie et al [12] and Saxena et al [13]. In this comparison will be considered the parameters created in the authentication phase and that need to be stored to carry out the authentication process. Table 9 shows the comparison of storage costs in bits:

Table 9. Storage Cost of the Authentication Phase

Protocol	Jie et. al[12] (bit)	Saxena et. al [13] (bit)	**Proposed protocol** (bit)
EV/PEV	$896n$	$768n$	$952n$
ST/CS	$256n$	--	--
LAG	$394n + 640$	$640n + 128$	1064
AS	--	--	$312n + 1272$
CA/RA	--	$256n$	--
Total	$1546n + 640$	$1664n + 128$	$1264n + 2336$

In Table 9 and Figure 13 it can be seen that the storage cost of the proposed protocol is lower than the protocols of Jie et al. [12] and Saxena et al. [13] for n > 5. The best performance in terms of storage is due to the cryptographic scheme, that involves the creation and storage of less data (bits) to carry out the authentication process.

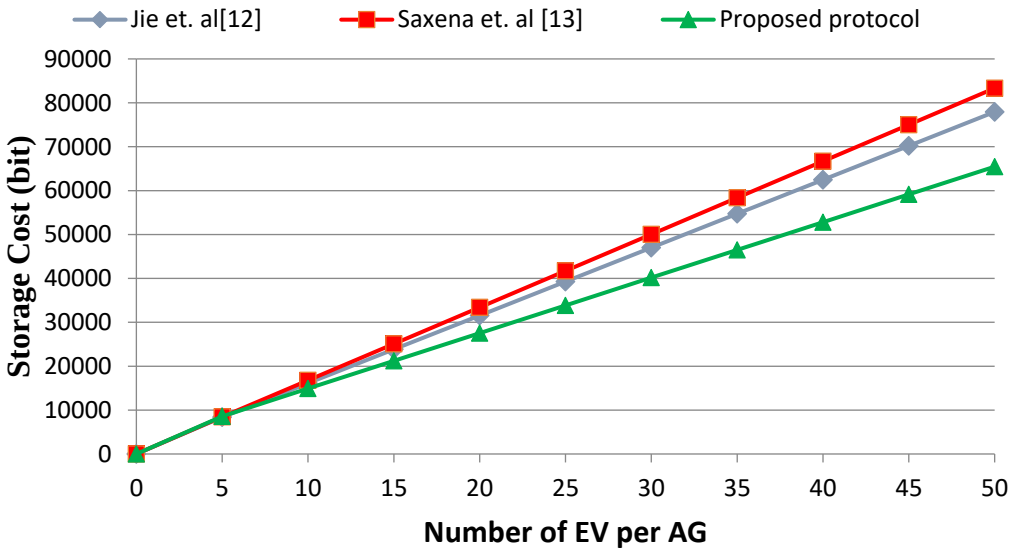


Figure 13 – Comparison of Storage Costs among Protocols

6. Formal Verification of the Proposed Protocol.

This section discusses the formal verification of the proposed protocol, introduces codes that represent the protocol in a high level language and provides the results of a simulation with AVISPA tool [30].

AVISPA is a formal verification tool vastly used for internet security assessment. It uses the HLPSL (High Level Protocol Specification Language) language that enables the description of entities, as well as exchange of messages necessary for the operation of the protocol.

The tool has four back-ends, among which we use On-the-Fly Model Checker (OFMC) and the Constraint-based Attack Finder (CL-AtSe). The verification of results is simple, i.e., “SAFE" is shown if no problem has been detected, and "UNSAFE" is shown otherwise. It is then possible to verify the security properties as well as vulnerability to various types of attacks [28].

6.1 Modeling of the Proposed Protocol in HLPSL

HLPSL allows the construction of protocol models that requires the specification of the sequence of actions of each type of protocol participant in a module. Part of the HLPSL codes is shown in Figures 14, 15 and 16 to illustrate how the proposed protocol was modeled for the simulation of its behavior and validation of security in the "AVISPA" tool.

Figure 14 shows the HLPSL code that models the developed behavior or role of one of the entities considered in the protocol. The structure of the HLPSL code of the EV role is the same of those of the codes of the other entities (AG and AS) and consists of the following parts:

- Statement of the agents, communication channels and constants known by the entity.
- Declaration of variables calculated or received by other entities.
- Statement of the functions to be used.

Once the above-mentioned statements have been made, the states are created. Such states describe the operations and messages to be exchanged with the other entities and are differentiated by a number assignment. At the end of each State, the elements that must be kept secret are declared.

```
role role_EV(EV,AG,AS:agent,P,Xev,Yij,Yag,Yas,IDev:text,Kgij,Kgi:symmetric_key,SND1,RCV1:channel(dy))

played_by EV
def=
        local
         State:nat,
         T1,G,X1,X2,T4,W1,W2,Z,Vev,Vas1,F,TKGi,Mev,TIDij,Mij,V,Mkij,Hkij,TIDg,LAI:text,
         Kij:symmetric_key,
         MAC,H1,H2,M,E,Sum:function

        init
         State := 0
        transition
         1. State = 0 /\ RCV1(staRt) =|> State':=1 /\ SND1(Mev',TIDij,MAC(Mev',TIDij))
         /\ Vev':=new()                    /\ TIDij':= M(Vev',P)
         /\ Mij':= {IDij'.Vev'.LAI}_kgij   /\ LAI':=new()


         4. State=2 /\ RCV1(G'.X1'.X2'.T4') =|> State':=3
         /\ TKGi':=H2(TIDg',Vas1)                  /\ W1' := M(H1(TIDg'),X2)
         /\ W2' := M(TKGi',X1')            /\ V' := xor(G',H2(W1'.W2'))
         /\ F' := H2(TIDg',Vas1')          /\ Kij' := E(Sum(Aij',Xev),F)
         /\ Hkij':= H1(Kij')               /\ Mkij':=xor((Hkij'.TIDij'),TKGi')
         /\ SND1(Mkij')                    /\ secret(TKGi',sec_1,{})
         /\ secret(Kij',sec_2,{AS,EV})

end role
```

Figure 14. Role of EV in HLPSL

Figure 15 shows (in HLPSL language) a role session that describes how a session is established and the role environment that describes the environment where the protocol is executed. The elements (variants, keys, agents, etc.) of the protocol an attacker can somehow acquire are also declared.

```
role session(EV,AG,AS:agent,
         P,Xev,Yev,Yag,Yas,IDag,Xag,Xas,TIDij,TIDag:text,
         KGij,KGi,KGag:symmetric_key,
         SND1,RCV1:channel(dy))

def=

        composition

         role_EV(EV,AG,AS,P,Xev,Yev,Yag,Yas, IDev,TIDij, KGij,KGi,SND1,RCV1)
         /\ role_AG(EV,AG,AS,P,IDag, TIDag,Yas,Xag,Yev,KGag,SND1,RCV1)
         /\ role_AS(EV,AG,AS,P,Yag,Yev,IDag,Yas,Xas,TIDij,TIDi,TIDag,KGij,KGi,KGag ,SND1,RCV1)
end role

role environment()
def=

        const
        p,xev,yev,yag,yas,tidij,tidi,tidag,idag,idij,xag,xas:text,
        ev,ag,as:agent,
        sec_1,sec_2,sec_3,sec_4,sec_5,sec_6,sec_7,sec_8,sec_9:protocol_id,
        snd1,rcv1:channel(dy)
        kgij,kgi:symmetric_key;


        intruder_knowledge = {}


        composition
         session(ev,ag,as,p,xev,yev,yag,yas,tidij,tidj,idag,idijxag,xas,kgij,kgi,kgag,snd1,rcv1)
end role
```

Figure 15. Specification of the role of session in HLPSL

Finally, Figure 16 shows the security objectives the protocol must guarantee, considering the definition of elements declared as secrets in the entity roles.

```
goal
        secrecy_of sec_1
        secrecy_of sec_2
        secrecy_of sec_3
        secrecy_of sec_4
        secrecy_of sec_5
        secrecy_of sec_6
        secrecy_of sec_7
        secrecy_of sec_9

end goal

environment()
```

Figure 16. Security objectives and related secrets of the proposed protocol in HLPSL

6.2 Security Check Results

Simulations performed with the OFMC and CL-AtSe back ends verified the protocol security. If the simulated protocol shows security problems, AVISPA provides a detailed result of the successful attack, whereas if the protocol is safe, AVISPA shows summarized information of the simulation. The simulation results show the protocol is safe for both back-ends, with the results shown in figure 17.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/artigo_3v_2.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.07s
  visitedNodes: 27 nodes
  depth: 3 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/artigo_3v_2.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 1 states
  Reachable  : 1 states
  Translation: 0.02 seconds
  Computation: 0.00 seconds
```

Figure 17. Security Simulation Results for OFMC and CL-AtSe

7. Conclusions

Due to the global need of reductions in air pollution, EVs have been a trend in research in many countries, as they can consume little or no petroleum, a scarce and non-renewable resource.

Part of the research related to EVs has been directed towards the creation of V2G networks for integrating EVs into SG networks. A fundamental part of the V2G network is the batteries of EVs, as they interact with an electricity network controlled by a bidirectional communication. Batteries can permit an EV to realize different functions within the V2G network, such as a provider, consumer or power storer.

Several security challenges in V2G communications can achieve confidentiality and privacy of data, e.g. vehicle identity, user´s identity, vehicle type, vehicle location, and other information to be protected. On the other hand, group-based organization of EVs [31] allows to improve energy distribution in SGs.

Part of the mentioned security challenges regards the authentication needs of EVs for their access to the V2G network. Some group-based authentication protocols have been proposed, however, their communication costs must be improved. Some of them also show computational overload in some elements of their infrastructure and a weak security analysis.

This article has introduced a new group authentication protocol for the V2G network based on ECDH and bilinear pairing. A brief description of some studies on security in V2G networks and solutions proposed for authentication in such networks are also provided.

In comparison with other proposals, our protocol shows better computational and communication costs and provides better results regarding security analysis. Moreover, it avoids centralization-related problems, due to a better distribution of the computational processing of operations in the devices and assures authentication of more entities. The protocol proposed by Jie et al. [12] has a low number of messages exchanged among entities, but a high processing cost in devices of limited computational resources, as EVs and LAGs, due to the calculation of exponential functions. On the other hand, the use of asymmetrical encryption in the communication between EV and CA/RA decreases its efficiency.

The AVISPA simulation tool formally proved the protocol is secure and guarantees successful authentication. It can meet the security and performance objectives and has proven an optimal choice in comparison to other authentication protocols for V2G networks.

Future work involves a simulation of the protocol in a network simulator and its adaption for integration in the V2G network for the cloud. Another line of work involves authentication and authorization protocols for cyber physical systems (CPS) considering communication models such as the model presented in [32]. Ongoing work is devoted to secure EV authentication schemes on charge while driving (CWD) systems, aiming to extend the reach of batteries through wireless power transfer (WPT) technologies with vehicle in motion [33].

Bibliography

[1] K. Shuaib, E. Barka, J. A. Abdella, F. Sallabi, M. Abdel-Hafez, Ala Al-Fuqaha, “Secure Plug-in Electric Vehicle (PEV) Charging in a Smart Grid Network”, Journal Energies, v.10, 2017.

[2] B. Vaidya, D. Makrakis, H. T. Mouftah, “Security Mechanism for Multi-domain Vehicle-to-Grid Infrastructure”, IEEE Global Telecommunication Conference, 2011.

[3] A. Abdallah, X. Shen, “Lightweight Authentication and Privacy-Preserving Scheme for V2G Connections”, IEEE Transactions on Vehicular Technology, v. 66, No. 3, pp. 2615-2629, 2017.

[4] H. Liu, H. Ning, Y. Zhang, Q. Xiong, L. T. Yang, “Role-Dependent Privacy Preservation for Secure V2G Networks in the Smart Grid”, IEEE Transactions on Information Forensics and Security, v. 9, No. 2, pp. 208-220, 2014.

[5] N. Saxena, S. Grijalva, V. Chukwuka, A. V. Vasilakos, “Network Security and Privacy challenges in Smart vehicle-to-grid” IEEE Wireless Communications, v.24, pp. 88-98, 2017.

[6] H Wang, X. Yu, H Song, Z. Lu, J. Lloret, F. You, “A Global Optimal Path Planning and Controller Design Algorithm for Intelligent Vehicles”, Mobile Networks and Applications (2016). https://doi.org/10.1007/s11036-016-0778-5.

[7] Keiko Karaishi, Masato Oguchi, “Evaluation of Smart Grid Simulation System with Power Stabilization by EV”, Network Protocols and Algorithms, Vol 5, No 1 (2013). Pp. 71-89

[8] W. Han, Y. Xiao, “Privacy preservation for V2G networks in smart grid -A survey”, Computer Communications, pp. 17-28, 2016.

[9] W. Han, Y. Xiao, “IP2DM-integrated privacy-preserving data management architecture for smart grid V2G networks”, Wireless Communications and Mobile Computing, v.16, pp. 2956-2974, 2016.

[10] M. Tao; K. Ota; M. Dong, “Foud - Integrating Fog and Cloud for 5G-Enabled V2G Networks”, IEEE Network, v. 31, pp. 8-13, 2017.

[11] Shuaib, K.; Barka, E.; Abdella, J.A.; Sallabi, F.; Abdel-Hafez, M.; Al-Fuqaha, A. “Secure Plug-in Electric Vehicle (PEV) Charging in a Smart Grid Network”, Energies, v.10, 2017.

[12] C. Jie, Z. Yueyu1, S. Wencong, “An anonymous authentication scheme for plug-in electric vehicles joining to charging-discharging station in V2G networks”. China Communication, v.12, pp. 9-19, 2015.

[13] N. Saxena, B. J. Choi; S. Cho, “Lightweight Privacy-Preserving Authentication Scheme for V2G Networks in the Smart Grid” IEEE Trustcom/BigDataSE/ISPA, v.1, pp. 604-611, 2015.

[14] Sania Yaqoob and Taeshik Shon, “A Hybrid EV Authentication Approach in Smart Grid Based Distributed Network”, Ad Hoc and Sensor Wireless Networks, Vol. 31, Number 1-4, Pp. 89-99, 2016.

[15] J. Lloret, P Lorenz, A Jamalipour,” Communication protocols and algorithms for the smart grid”, IEEE Communications Magazine 50 (5). 2012.

[16] F. Wang, C. Chang, Y. Chou2, “Group Authentication and Group Key Distribution for Ad Hoc Networks”, International Journal of Network Security, v.17, pp. 199-207, 2015.

[17] H. Li, “Privacy-preserving authentication and billing for dynamic charging of electric vehicles”, Doctor Dissertation, University of Illinois at Urbana-Champaign, 2016.

[18] Certicom Research, ”Standards for efficient cryptography, SEC 1: Elliptic Curve Cryptography”, Version 2.0, May 21, 2009.

[19] H. Liu; H. Ning, Y. Zhang, M. Guizani, “Battery Status-aware Authentication Scheme for V2G Networks in Smart Grid”, IEEE Transactions on Smart Grid, v.4, pp. 99-110, 2013.

[20] Z. Sun “An Anonymous Authentication Scheme for Vehicle-to-Grid Networks” Int. J. Communications, Network and System Sciences , v.10, p.p 316-323, 2017.

[21] Z. Wan, W. Zhu, G. Wang “PRAC: Efficient privacy protection for vehicle-to-grid communications in the smart grid” Computers & Security, v.62, pp. 246-256, 2016.

[22] K. Shuaib, E. Barka, J. Abdella, F. Sallabi, M Abdel-Hafez, A. Al-Fuqaha, “Secure Plug-in Electric Vehicle (PEV) Charging in a Smart Grid Network”, Journal Energies, 2017.

[23] Li, F., Xin, X. e Hu, Y. (2008) “Efficient Certificate – Based Singcryption Scheme From Bilinear Pairings”, International Jurnal of Computers and Applications, v.30, No 2, 2008.

[24] Menezes, Alfred. (2005) “An Introduction to Pairing-Based Cryptography”, Recent Trends in Cryptography, v. 477, p. 47-65.

[25] B. L. Parne, S. Gupta, N. S. Chaudhari, “SEGB: Security Enhanced Group Based AKA Protocol for M2M Communication in an IoT Enabled LTE/LTE-A Network” , IEEE Access, v.6, pp. 3668 – 3684, 2018.

[26] Luis F. A. Roman, Paulo R. L. Gondim and Jaime Lloret. ”A Lightweight Authentication Protocol for V2G Networks in Smart Grid”, Pervasive and Embedded

Computing and Communication Systems, 8th International Joint Conference on, pp. 1-10, 2018.

[27] I. Wagner, D. Eckhoff, “Technical Privacy Metrics: A Systematic Survey”, arXiv: 1512.00327 [cs, math], Dec. 2015.

[28] M. Eiza, Q. Shi, A. Marnerides, T. Owens, “Secure and Privacy-Aware Proxy Mobile IPv6 Protocol for Vehicle-to-Grid Networks”, International Conference on Communications (ICC), 2016.

[29] M. Tao; K. Ota; M. Dong, Z. Qian, “AccessAuth: Capacity-aware security access authentication in federated-IoT-enabled V2G networks”, Journal of Parallel and Distributed Computing, 2017.

[30] The AVISPA Project: European Union in the Future and Emerging Technologies (FET Open). Retrieved Nov 26, 2016, from http://www.avispa-project.org.

[31] J. Lloret, M Gilg, M. Garcia, P. Lorenz, “A group-based protocol for improving energy distribution in smart grids”, Communications (ICC), 2011 IEEE International Conference on, pp. 1-6, 2011.

[32] Jafar Rasouli, Seyed Ahmad Motamedi,Mohamad Baseri and Mahshad Parsa, “A Reliable Communication Model based on IEEE802.15.4 for WSANs in Smart Grids”, Ad Hoc and Sensor Wireless Networks, vol. 39, pp. 313-343, 2017.

[33] T. V. Theodoropoulos, I. G. Damousis, A. J. Amditis, “Demand-Side Management ICT for Dynamic Wireless EV Charging”, IEEE Transactions on Industrial Electronics, v. 63, p.p 6623-6630, 2016.