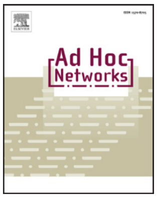
Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Authentication protocol in CTNs for a CWD-WPT charging system in a cloud environment

Luis F.A. Roman*, Paulo R.L. Gondim

Departamento de Engenharia Elétrica, Universidade de Brasília (UnB), Brasília, Brazil

ARTICLE INFO

Article history:
Received 8 April 2019
Revised 3 July 2019
Accepted 2 September 2019
Available online 5 September 2019

Keywords:
Authentication
CTNs
EVs
Fog Computing
VANET
CWD

ABSTRACT

The Internet of Things (IoT) has developed very rapidly in recent years, becoming increasingly complex. The communication things networks (CTNs) is a very important element in IoT for the interconnection of several objects (between objects or objects with the internet). In the context of electric vehicles, the development of CTNs represents a pillar for the implementation of new services such as charge while driving (CWD) based on wireless power transfer (WPT) technology. Cloud-based vehicular ad-hoc networks (VANETs) are one of the networks that can support the high mobility, low latency and connectivity required for a CWD-WPT system. The CWD-WPT charging system provides comfort and time optimization for users if the privacy, integrity and availability of the system are guaranteed. This paper proposes an authentication protocol that uses different cryptographic schemes for key management and distribution in a CWD-WPT cloud charging system that guarantees message privacy and integrity, mutual authentication of system elements and anonymity of EVs.

1. Introduction

Research on the Internet of Things (IoT) has been substantially increased in recent years because of the applications and capabilities it offers. IoT is a complex and heterogeneous ecosystem that interconnects several objects on a large scale to offer innovative services, such as drone-based services, health care services, smart grid features and electric vehicles [1,2].

The widespread adoption of IoT depends on communicating things networks (CTNs), which must adapt to new quality of service (QoS) requirements, carry large volumes of data, and support heterogeneity in different traffic pattern devices for ensuring a reliable delivery of services.

The benefits of IoT and CTN technologies can reach several application areas. Among these areas, we observe that the popularity of electric vehicles (EVs) has grown over the past years, mainly due to the scarcity of fossil fuels and for environmental reasons. According to the Organization for Economic Cooperation and Development, the transportation sector consumes over 50% of the world's oil and is responsible for the emission of approximately 20% of carbon dioxide worldwide. Although the adoption of EVs can improve the environment and reduce the oil dependency, several technological and operational challenges must be overcome [3,4].

One of such challenges is the long duration of charging and battery life, which generate time and mobility restrictions for users [5]. Researchers have been working on the development of a new method of charge while driving (CWD) based on wireless power transfer (WPT) technology. Installed in strategic places, it ensures the EV can travel further and in less time (by popping the charging time) [4,6–8].

The dynamic charging infrastructure or CWD-WPT consists of a series of charging coils called pads embedded in the road pavement. Unlike long and continuous static charging, CWD-WPT comprises a large number of pads that power the EV (micro charging) in only a few milliseconds, depending on the speed of the vehicle [5,6,8].

One of the most outstanding features of the CWD-WPT system is mobility, which promotes changes in the context (location, type of vehicle), access and network connectivity (connection time, wireless or wired network), energy availability (state of charge (SoC)), security and privacy [9].

For the treatment of mobility and connectivity among vehicles, a vehicular ad-hoc network (VANET) is one of the networks that can be considered to support a CWD-WPT system. VANETs have drawn the attention of researchers due to their large variety of applications and services and a safe, efficient, trouble-free and entertaining intelligent transportation system (ITS). VANETs provide vehicles with an onboard communication unit called On-Board Unit (OBU), through which they communicate with both other vehicles and the infrastructure via Roadside Units (RSUs). IEEE 802.11p

* Corresponding author.
E-mail address: lfroman@aluno.unb.br (L.F.A. Roman).

https://doi.org/10.1016/j.adhoc.2019.102004

standard provides the Wireless Access in Vehicle System (WAVE) protocol and the basic radio standard for dedicated short-range communications (DSRC) at a 5.9 GHz frequency [5,10].

Due to the technological evolution and exponential growth in the number of intelligent vehicles, traditional VANETs have faced flexibility and scalability problems, amongst others. The integration of the cloud with VANET networks seeks to solve the problems of flexibility and scalability, as well as to foster the evolution and creation of new services. Cloud-based VANET communications are comprised of a number of elements and environments that integrate seamlessly to provide users with efficient, scalable and secure services. To achieve this harmonic integration in cloud-based VANET networks, several authors have proposed layered systems with different focuses, where security is a layer that interacts throughout the system [11–13].

Cloud computing is a new paradigm that proposes allocating servers geographically but next the devices to collect, process, organize and store data in real time. Its use in vehicular networks tends to facilitate or provide a great variety of services, besides being a solution to reduce the costs of communication [13]. Cloud computing presents several security challenges, which include data storage, computing, virtualization and network security issues, as well as access control, software security and trust management issues [14].

More specifically, cloud-based vehicular networks security is a challenging problem because of its additional characteristics of heterogeneity and the high volume of vehicles. According to Ziquia et al. [11] the most important security requirements for these networks are: authentication, data integrity, confidentiality, access control, non-repudiation and availability.

The next-generation VANETs must also support high mobility, low latency, real-time services and connectivity, which cannot be provided by conventional cloud computing. An effective solution to vehicular network problems is the fusion of fog computing with cloud computing [10,15], allowing to extend to the edge of wireless networks the conventional paradigm of cloud computing and meeting requirements related to low latency, seamless mobility, data storage close to users and adequate localization of mobile devices. Moreover, the use of fog servers allows better mobility management of vehicles and redirectioning of mobile applications to the closest fog server [15].

Such a cloud environment creates a scalable and hierarchical architecture, which is convenient for the sake of distribute processing and storage capabilities. In our architecture, the company charging server (CCS) is installed in the cloud computing and connected to a group of secondary servers (fog servers - FS), where the fog computing is installed. Each FS groups several RSUs, and each RSU groups several pads together.

The CWD-WPT charging technology in a cloud and fog computing environment can provide comfort and time optimization for EV users, if security, privacy, authentication and anonymity are considered. Mechanisms for EVs to enter a carrier charging service in a controlled and anonymous manner require efficient mutual authentication [16,17].

Proposals for authentication protocols have been presented in the literature. For example the protocols presented by Li et al. [18] and Hussain et al. [19] which focused on the mutual authentication between entity and the preservation of privacy; however, the analysis of security problems is poorly detailed. Other proposals such as those presented by Gunukula et al. [20] and Rabieh and Wei [21] guarantee anonymous authentication, privacy, unlinkability and prevent double spending; however, disregard some attacks that may affect the system. Other shortcomings that the protocols proposed so far have in common is the lack of a formal verification and a comparison of performance with other protocols.

This article proposes a protocol for the administration and distribution of keys in a CWD-WPT charging system in a cloud and fog computing environment, which guarantees privacy and integrity of messages, mutual authentication between the EV and the CWD-WPT charging station and EV anonymity. Its contributions include:

- an authentication and authorization protocol, enabling privacy and integrity preservation as well as key agreement and distribution;
- design of a new CWD-WPT dynamic charging architecture based on a fusion of fog computing with cloud computing;
- preservation of the anonymity of EVs, since the protocol is based on download tickets purchased offline and signed blindly by the system;
- use of cryptographic primitives, such as short signatures and blind signatures based on bilinear pairing for authentication with no jeopardy to the true identity of the EV;
- mutual authentication among the EV and all entities of the CWD-WPT charging station;
- a formal security verification of the protocol by AVISPA tool;
- a security analysis considering several attacks that can affect the system, where a larger number of attacks has been considered, when compared to other proposals (as [18–21]);
- a comparison of performance with other protocols, involving communication and computational costs.

The remainder of the paper is organized as follows: Section 2 addresses related works; Section 3 describes the system model and adversary models; Section 4 provides preliminary information for the understanding of the protocol; Section 5 introduces the protocol; Section 6 reports on performance evaluations and a safety performance analysis; finally, Section 7 is devoted to the conclusions.

2. Related work

Li et al. [18] presented an authentication protocol called "Fast Authentication for Dynamic EV Charging (FADEC)", which has a dedicated short-range communication (DSRC) based on the IEEE 802.11p standard and a five-element architecture, i.e., the utility in charge of the management and administration of the CWD system, a Certification Authority (CA) that certifies all system keys, a set of pads installed on the highway for inducing energy to EVs, RSUs, which are wireless communication devices distributed over the sidewalk and interconnected through a backbone network, and EVs equipped with On-board Units (OBU) that use DSRC to communicate with RSUs.

The authentication protocol was based on the hash-based message authentication code (HMAC), which authenticates entities that rely on a symmetric key shared between two parties, the Elliptic Curve Digital Signature Algorithm (ECDSA), which authenticates vehicle safety messages, and Just Fast Keying (JFK), a key exchange protocol based on the Diffie-Hellman protocol. Li et al. [18] do not emphasize the authentication process and establishment of the session key (JFK protocol). The security based on the JFK protocol has some flaws, since it does not protect the privacy of the user and is susceptible to repetition attacks.

Hussain et al. [19] designed a mutual authentication protocol that ensures privacy for a CWD system via charging plates (CPLs) installed under boards. The authors adopted the concept of online electric vehicle (OLEV) used in South Korea to name vehicles that receive an electric charge from the power line installed below the road surface. The network model is based on a typical VANET network consisting of EVs equipped with an OBU to communicate with the charging infrastructure via DSRC and a tamper-resistant module (TRM) that stores the confidential information of the EV; CPLs installed on the surface of the road and responsible for the

EV charging, VANET Authority, responsible for the registration and revocation of the system, and charging service providing authority (CSPA), responsible for delivering power to the CPs. The Department of Motor Vehicles (DMV) is at the top of the hierarchy, where each VANET Authority must be registered.

The protocol of Hussain et al. [19] uses the following cryptographic primitives to ensure protocol security: El Gamal encryption algorithm over elliptic curve cryptography (ECC), hash, hash chain, and XOR functions, for security analysis, which prove the resistance of the protocol against replaying attacks and impersonation, and dispute resolutions between EVs and the charging system. They have focused only on efforts to ensure mutual authentication and have not analyzed other security issues that may affect the system, such as integrity, DoS attack, Man-In-the-Middle attack, amongst others.

Gunukula et al. [20] designed a protocol that preserves the security of the dynamic charging system and payment of the service. The network model considered in Gunukula et al. [20] is composed of a bank responsible for the sales of charging coins and verification of the validity of currencies. A carrier service provider (CSP) manages the RSU group that is part of the charging station, the RSUs responsible for the management of the group of charging pads installed on the highway, and the charging pads responsible for the induction of energy to the EV.

Towards guaranteeing the security of the system, the protocol was based on the following cryptographic primitives: ECC-based partial blind signature, Diffie-Hellman-based key agreement in ECC, Exclusive-OR and modified hash chain. The safety analysis describes the protocol of Gunukula et al. [20], which guarantees the anonymous authentication of the EV prior to the charging and disassociation of the EV with the currencies purchased. It also provides a description of resistance to attacks such as double spending, man-in-the-middle, and others related to payment for the service; however, it does not analyze attacks that can affect the overall system.

Rabieh and Wei [21] proposed an efficient authentication protocol that guarantees the privacy of drivers. It is composed of EVs that use the charging system, and a harging management center (CMC), i.e., the main component of the architecture, controls the charging controllers and the charging pad (CP). The CPs are installed under the road and induce electric charge to the EVs. A charging controller is installed next to the highway and interconnects the CMC and the pads of the charging station. Finally, the charging carrier implements the necessary infrastructure for charging the EVs (CMC, charging controllers and CPs).

The protocol guarantees the security of client information through the following cryptographic primitives: hash chain, hash, Exclusive-OR operations and blind signatures based on bilinear pairing. The security analysis describes the way the protocol performs a mutual authentication between the EVs and the system and guarantees the privacy of the EVs, unlinkability, double spending and anonymity of the EVs. Differently from other protocols, the one designed by Rabieh and Wei [21] considers an specific architecture of VANET and the security analysis does not consider several attacks that can affect the system such as injection, known key and impersonation attacks, among others.

Laporte et al. [22] described an experimental investigation for characterizing the actual performance of a WPT charging system for EV, in order to carry out a feasibility analysis of the wireless charging technologies that extend the distance traveled by the EV. The work also describes multidisciplinary technical challenges that must be solved, for example, controlling the speed of the EV in the road of load, energy efficiency of transmission from the pads to the EV, and the impact that the WPT system has in the power network.

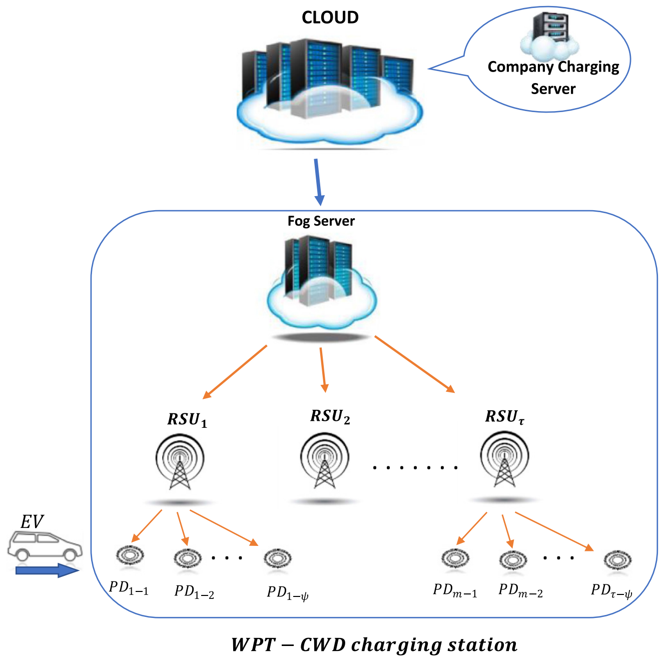


Fig. 1. Network model.

In the paper by Roberts et al. [23], the authors analyzed the high economic costs of the load infrastructure and the potential problems in the power system caused by the high levels of penetration of the EVs. To solve these problems, they proposed a ubiquitous charging system based on a vehicle-to-vehicle (V2V) energy transfer and an certificateless authentication protocol between supplier and customer for a system of Vehicle to Vehicle loads (V2V).

3. Network model and adversary model

This section describes the network and adversary models considered in our study.

3.1. Network model

Fig. 1 shows the network model with company charging service (CCS) (located in a cloud), EVs and a WPT-CWD charging station. Each WPT-CWD charging station is comprised of a fog server, multiple RSUs and charging pads.

The system is assumed to have several WPT-CWD charging stations that communicate with the CCS. EVs can communicate with the CCS via the Internet. RSUs are access points installed on the roadside of the WPT-CWD charging station and can cover several kilometers. We consider there are "τ" RSUs for one WPT-CWD charging station, and each RSU can communicate with a group of "ψ" pads, while the fog server can communicate with all RSUs of the WPT-CWD charging station. Pads are elements that induce an electric charge to the EVs in motion using WPT. Each pad is activated through the validation of a unique key delivered by an EV. EVs can communicate with FS and RSUs through wireless networks, and with the pads through a short-range wireless communication device. Table 1 shows a comparison of the entities of different architectures that support the WPT-CWD service.

3.2. Adversary (attack) model

The Dolev–Yao attack (adversary) model [24] is adopted; in this sense, inspite of messages that can be composed and replayed by

Table 1
Comparison among entities and primitives.

	Entities considered		Cryptographic primitives	Cloud based?	Formal verification security?	Comparison with other protocols?
Li et al. [18]	EV, Pad, RSU, Utility, CA	5 entities	HMAC, symmetric key; ECDSA and Just Fast Keying (JFK)	Not	Not	Not
Hussain et al. [19]	EV, CP, CSPA, VANET Authority	4 entities	ElGamal over ECC, hash, hash chain, and XOR functions.	Not	Not	Not
Gunukula et al. [20]	EV, CSP, RSU, Pad, Bank	5 entities	ECC-based partial blind signature, Diffie-Hellman key agreement based on ECC, XOR and modified hash chain	Not	Not	Not
Rabie et al. [21]	EV, Pad, C-Company, CMC, C-controller.	5 Entities	hash chain, hash, Exclusive-OR operations and blind signatures based on bilinear pairing	Not	Not	Not
Proposed protocol	EV, Pad, RSU, Fog Server, Cloud(CCS)	5 Entities	Diffie-Hellman Key Agreement based on ECC, Short Signatures and Blind signatures, bilinear pairing and Hash Chain.	Yes	Yes	Yes

an adversary, he/she cannot decipher them without knowing the correct cryptographic keys. Moreover, one-way functions are considered unbreakable.

In the proposed scheme, only the CCS entity is trustworthy regarding the real identity of the EV (for collecting tickets). The fog server, the RSU, and the charging pads should do not reveal the real identity of the EV or its owner. Although trustworthy, EVs are curious about private information from the other EVs (SoC, Drivers' identities, etc.), but they do not disturb the operation of the system.

The VANET infrastructure is assumed secure and the RSU has a private key X_{RSU} and a public key Y_{RSU}. The RSUs are connected to the fog server and have a group key K_{G-RSU}. On the other hand, the pads are connected to the RSUs. Finally, a group key for the pads K_{G-pads} is defined.

4. Preliminaries

4.1. Bilinear pairing

Bilinear pairing is defined as the projection of two points of additive set G_1 formed by points on an elliptic curve E of order $l \in Z_p^+$, towards a same point of a multiplicative set G_2 formed by the elements of order $l \in Z_p^+$. The discrete logarithm problem (DLP) is assumed hard in both G_1 and G_2. A mapping $\hat{e} = (G_1, +)^2 \rightarrow (G_2, \cdot)$ satisfies the following properties for all a, b $\in Z_q^*$ and $c, d \in G$ ([25]).

(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:

$$\hat{e}(c, d) \neq 1_{G_2}$$

(3) Computationally efficient.

Bilinear pairings have other easily verifiable properties, such as:

(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) $\hat{e}(ac, bd) = \hat{e}(d, c)^{ab}$
(4) $\hat{e}(c, d) = \hat{e}(d, c)$,

and can be used for data encryption, digital signatures and key agreements. In our protocol they are employed for the generation of digital signatures.

4.2. Digital signatures

A digital signature is one of the most important cryptography-based resources. It indicates the owner or creator of a document or clarifies someone agrees on the content of a document. Some digital signatures are based on a public key that links the identity of the user with its public key, whereas others are based on the identity of the that generates the public key from the user's identity through a deterministic algorithm. The public key verification is based on the use of the user's identity, making this scheme more efficient. The first short bilinear pairing scheme was created by Boneh et al. [26], and from it were created a large number of signature schemes based on the coincidence for different applications [25]. Below is a description of the digital signature schemes used in our protocol.

4.2.1. Short signatures

Short signatures work well in environments of memory and bandwidth restrictions. The most used signature schemes are RSA (Rivest, Shamir and Adleman) and DSA (Digital Signature Algorithm), however, the signatures they generate are long. For example, if the 1024-bit module is used, the signatures of RSA and DSA are 1024 bits long. The bilinear pairing scheme provides short-length signatures of approximately 160 bits with a security level similar to those of 1024-bit RSA and DSA signatures [25].

A signature scheme based on bilinear pairing commonly involves [25]:

- Initialization: Let H: $\{0, 1\}^* \rightarrow G1$ be a map-to-point hash function. The secret key is $X, \in Z_q^*$, and the public key is $Y = X^* P$ for a signer.
- Sign: Given secret key x and a message $m \in \{0, 1\}^*$, compute signature $\sigma = X^* H(m)$
- Verify: Given public key $Y = X^* P$, a message m and a signature σ, verify $e(P, \sigma) = e(Y, H(m))$.

4.2.2. Blind signatures

Blind signatures have been widely used in digital payment schemes for the obtaining of the signature of a document without

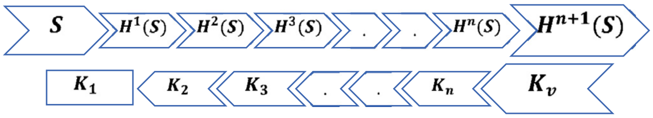


Fig. 2. Hash chain.

the signatory knowing the information of the document. Moreover, the user cannot obtain other valid signatures of the same document after an interaction with the subscriber. The scheme used for our protocol was created by Zhang et al. [27] and is called "ID-Based Blind Signature and Ring Signature from Pairings". It is characterized by the use of an identity-based cryptosystem over bilinear pairings for the verification and authentication of the signed information without knowing the identity of the sender.

4.3. Hash chain

Hash chain is a computational operation for the efficient authentication of one-time passwords, extending the lifetime of digital certificates, building one-time signatures, amongst other functions. It was used in this study for the authentication and creation of session keys [28].

A hash chain is generated by a hash algorithm, as SHA (Secure Hash Algorithm), through which a user randomly selects a seed (S) and calculates the entire key chain. Fig. 2 shows the process of creation of keys with a chain hash.

The keys generated must be used in the opposite order of their generation, i.e., the last generated key K_n must be the first one used and the first key K_1 must be the last key used, such that an attacker listening to the channel cannot calculate a valid key from a used one. In our protocol, a public verification key K_v, is calculated applying $n+1$ hashes to S for the validation of the keys. To verify a hash chain, an entity only applies successive hashes until it reaches the value of key K_v. If the key received after the application of n hash at maximum is not given the same value of the verification key, it is discarded.

5. Proposed protocol

Our protocol is divided into four phases, namely initialization, registration, ticket purchasing and charging request (see Fig. 3). In the initialization phase, sets, functions and master keys necessary for the start of the operation of the scheme are defined. In the registration phase, the data of the EV are stored in the system. In the phase of purchasing tickets, EVs purchase one or several tickets to perform the EV charge in the charging station. Finally, in the charging request phase, the delivery, validation, authentication and generation of keys necessary for the charging of EV through the WPT-CWD system are performed.

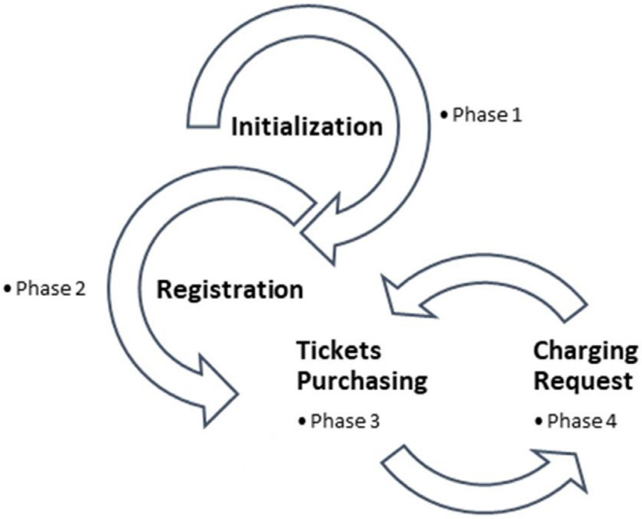


Fig. 3. Phases of the proposed protocol.

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

In this phase, the use of the pseudorandom random number generator (PRNG) is considered for the generation of nonces and seeds. The PRNG will be reinitialized at random times, and the random value generated by the PRNG will be processed by a hash function to be used by the system. In PRNG, the initial state is changed with parameters that are the product of applying hash functions over input values concatenated with timestamps [29].

The system had chosen two cyclic groups G_1 and G_2 of orders q and P and a generator element of group G_1 are chosen. G_1 and G_2 are supposedly related to a non-degenerative pairing and a bilinear map that can be efficiently computed:

ê: $G_1 \times G_1 \rightarrow G_2$ such that ê(P, P) $\neq$ 1G_2 and ê(aP,bQ) = ê(b P,a Q) = ê(P_1, Q_1)$^{\mathrm{ab}}$ $\in G_2$ for every a, b $\in Z_q^*$ and every P, Q $\in G_1$. Moreover, the hash functions of the system are defined: H: $\{0, 1\}^* \rightarrow G_1$ and $H_1 : \{0, 1\}^*.G \rightarrow \mathbb{Z}_q^*$.

CCS then chooses a master private key $Y_{ccs}, \in Z_q^*$ and calculates its global public key $Y_{pub} = X_{ccs}{}^*P$. Additionally, it computes its own public key $Q_{ccs} = H(ID_{ccs})$ and private key $S_{ccs} = X_{ccs}{}^*Q_{ccs}$.

Finally, the company charging center (CCS) defines an elliptical curve on a finite field E (Fq) and parameters {G_1, G_2, ê, P, H, $\mathrm{H}_{1,}$ $P_{pub},$ Q_{ccs}} are published.

2nd phase: **EV registration**

All owners of EVs who want to use the CWD charging system register with the CCS through a secure channel. The user chooses a random number $X_{EV}, \in Z_q^*$ and calculates $Y_{EV} = X_{EV}{}^*P$, where X_{EV} will be his/her private key and Y_{EV} will be the public key. This public key along with identity (ID_{EV}) and vehicle charging parameters (VCP) are sent to the CCS to be stored. Finally, the CCS creates a certificate $Cert_{EV} = X_{ccs}{}^*Q_{EV}$ where $Q_{EV} = H(ID_{EV})$ and sends it to the EV.

3rd phase: **Tickets Purchasing**

Each ticket is assumed to have a specified amount of energy to be induced to the EV through a certain number of pads. The tickets are purchased through a secure channel and the EV has an associated bank account in the CCS, with enough money for their purchase..

The first message, m_1, requesting the purchase of n tickets to the CCS is sent by the EV.

$$m_1 = \{n, ID_{EV}, Cert_{EV}\}$$

The CCS receives it and generates n random values $\{r_1, r_2, \ \ldots, r_n\} \in Z_q^*$. For each r_i for $0 \ \leq \ i \ \leq \ n$, $R_i = r_i{}^*P$ is calculated and a message m_2 containing set $R = \{R_1, R_2, \ldots.,\ R_n\}$ is sent to the EV:

$$m_2 = \{R\}$$

The EV receives it, creates $\boldsymbol{n}$ random pseudonyms {PID_1, PID_2, ..., PID_i...., PID_n}, and applies a blind signature to each $\boldsymbol{n}$ PID. It then chooses two random numbers a, $b \in Z_q^*$ and computes the blind pseudonym (B) for every pseudonym PID:

$$B_i = H\left(PID_i, \hat{e}\left(bQ_{ccs} + R_i + aP, Y_{pub}\right)\right) + \mathrm{b}$$

The EV sends message m_3 with the $B = \{B_1, B_2, \ldots.,\ B_i, \ldots.B_n\}$ to the CCS to receive the system signature.

$$m_3 = \{B\}$$

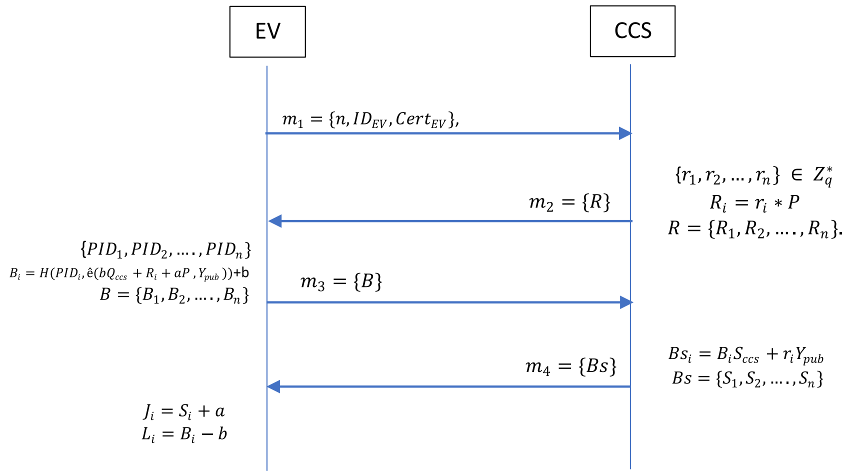

EV
CCS
$m_1 = \{n, ID_{EV}, Cert_{EV}\},$
$\{r_1, r_2, \ldots, r_n\} \in Z_q^*$
$R_i = r_i * P$
$R = \{R_1, R_2, \ldots., R_n\}.$
$m_2 = \{R\}$
$\{PID_1, PID_2, \ldots., PID_n\}$
$B_i = H(PID_i, \hat{e}(bQ_{ccs} + R_i + aP, Y_{pub}))$+b
$B = \{B_1, B_2, \ldots., B_n\}$
$m_3 = \{B\}$
$Bs_i = B_i S_{ccs} + r_i Y_{pub}$
$Bs = \{S_1, S_2, \ldots., S_n\}$
$m_4 = \{Bs\}$
$J_i = S_i + a$
$L_i = B_i - b$

Fig. 4. Ticket purchasing.

The CCS receives the message and signs all blind pseudonyms from set B:

$$Bs_i = (B_i * S_{ccs}) + \left(r_i * Y_{pub}\right)$$

It then sends message m_4 ($Bs = \{Bs_1, Bs_2, \ldots., Bs_n\}$) to the EV.

Finally, the EV receives m_4 containing set Bs and calculates two values (J and L) for signature verification to obtain the signature of each blind pseudonym set $B = \{B_1, B_2, \ldots., B_i, \ldots. B_n\}$:

$J_i = Bs_i + aY_{pub}$, and $L_i = B_i - b$, therefore, the signature of each blind pseudonym B_i will be the pair of values (J_i, L_i). The Fig. 4 shows a summary of the ticket purchase phase and a summary of the ticket purchasing phase, respectively.

4th phase: **Charging Request**

This phase describes the verification, authentication, and creation of session keys between the EV and the WPT-CWD charging station.

Once the EV owner has a valid ticket (PID_1, J_1, L_1) and wants to charge his/her EV in a WPT/CWD charging station, the EV system selects a random number $\varphi_{EV} \in Z_q^*$, calculates $\phi_{EV} = \varphi_{EV} {}^* P$, and sends an m_1 message to the fog server

$$m_1 = \{\phi_{EV}, t_5, H(\phi_{EV} \| t_5)\},\ \text{where } t_5 \text{ is a timestamp.}$$

The fog server checks the hash and message timestamp m_1. If it succeeds, the server chooses a random value $\varphi_{fs} \in Z_q^*$ and calculates session $k_{fs-EV} = \varphi_{fs} {}^* \phi_{EV}$ and values, such that the EV can calculate session key $\phi_{fs} = \varphi_{fs} {}^* P$, verification key $VK = H(k_{fs-EV})$, and signature message $\sigma_{fs} = x_{fs} {}^* H(\phi_{fs}, CK, t_5)$. The fog server immediately sends message m_2 to the EV.

$$m_2 = \left\{\phi_{fs}, VK,\ t_6, \sigma_{fs}\right\}$$

When $m_2{}' = \{\phi_{fs}{}', VK',\ t_6{}', \sigma_{fs}\text{'}\}$ arrives, the EV checks fog server's signature σ_{fs}': ê (σ_{fs}', P) =?ê ($H(\phi_{fs}\text{'}, VK\text{'}, t_6\text{'})$, Y_{fs}). If the equality is successful, the EV authenticates the fog server, uses the message values to calculate session key $k_{fs-EV} = \varphi_{EV} {}^* \phi_{fs}$, and verifies the integrity of the key calculating $VK = H(k_{fs-EV})$ and checking if $VK\text{'} =?VK$. If the equality is successful, the EV uses the session key to crypt and send message m_3 containing the ticket (PID_1,J, L) and a timestamp to the fog server.

$$m_3 = \{PID_1, J_1, L_1, t_7\}_{k_{fs-EV}}$$

When the message arrives at the fog server, it is deciphered with session key k_{fs-EV}, the timestamp is checked and the pseudonym validity is immediately verified: $L_i = H(PID_i, \hat{e}(J_i, P)\hat{e}(Q_{ccs}, Y_{ccs})^{-L_i})$. If the validation is successful, the fog server chooses random seeds α_1, α_2, creates a new pseudonym $PID2_1 = H_1(PID_1 + \alpha_1)$, and sends an encrypted message m_4 containing seed α_1, τ and a timestamp to the EV. A message broadcast m_5 encrypted with key K_{G-RSU} and containing seeds α_1, α_2, τ and a timestamp is also sent to the group of RSUs. Finally, the fog server revokes pseudonym PID_1 to prevent its reuse.

$$m_4 = \{\alpha_1, \tau, PID2_1, t_8\}_{k_{fs-EV}},\ \text{sent to EV}$$

$$m_5 = \{\alpha_1, \alpha_2, \tau, PID2_1, t_9\}_{k_{G-RSU}},\ \text{sent to RSU}$$

When the EV receives m_4, it decrypts it and checks its timestamp. If the verification is successful, it calculates, offline, a verification key for each RSU using a hash chain $H^{RSU}(\alpha_1) = \{H(\alpha_1),\ H^2(\alpha_1), \ldots H^\tau(\alpha_1)\}$. It also calculates, offline, and with each verification key, a message authentication code $HMAC_{RSU}^d = \{PID2_d || 1 || t_8 || H^d(\alpha_1)\}$, and authenticates each RSU.

All RSUs receive the message m_5 from the fog server, decrypt with the group key (k_{RSU-G}) and check the timestamp. If the check succeeds, each RSU calculates the a check key $H^d(\alpha_1)$, a session key $k_{RSU-PID2} = H(H^d(\alpha_1 || d) \oplus H^d(\alpha_2))$, a verification key (***VK***) and a message authentication code $HMAC_{RSU}^d = H(H^d(\alpha_2) || VK_2 || t_{10} || H^d(\alpha_1))$, where ***d*** is the position of the RSU at the charging station d: $1 \le d \le \tau$.

The authentication of the first RSU is explained in what follows for simplifying the description of the protocol. The authentication of the EV with the other RSUs and the group of pads managed by it undergo the same authentication process.

When the EV is authenticated with the first RSU, it sends a message m_6 containing message pseudonym $PID2_{EV}$, the sequence number of RSU, a timestamp, and an $HMAC_{RSU}^1 = H(PID2_{EV} || 1 || t_9 || H^1(\alpha_1))$.

$$m_6 = \left\{PID2_{EV}, 1, t_{10}, HMAC_{RSU}^1\right\}$$

When the message arrives, the RSU checks if its database contains PID_{EV}. If so, it checks $HMAC_{RSU}^1$ with the values associated with $PID2_{EV}$. If the verification is successful, the RSU computes session key $k_{RSU-EV} = H(H^1(\alpha_1 || 1) \oplus H^1(\alpha_2))$, and sends message m_7 containing a value $H^1(\alpha_2)$, a key verification code $VK_2 = H(k_{RSU-EV})$, and its signature $HMAC_{EV}^1 = H(H^1(\alpha_2) || VK_2 || t_{10} || H^1(\alpha_1))$ to the EV. It also adds the check key to a revocation list of RSUs to prevent reuse of the key.

$$m_7 = \left\{H^1(\alpha_2), VK_2, t_{11}, HMAC_{EV}^1\right\}$$

When $m_7{}' = \{H^1(\alpha_2)', VK_2{}', t_{10}{}', HMAC_{EV}^{1'}\}$ arrives, the EV checks the RSU's $C_{EV}^{1'} =?HMAC_{EV}^\tau = H(H^1(\alpha_2)' \| VK_2{}' \| t_{10}{}' \| H^1(\alpha_1)$ If the

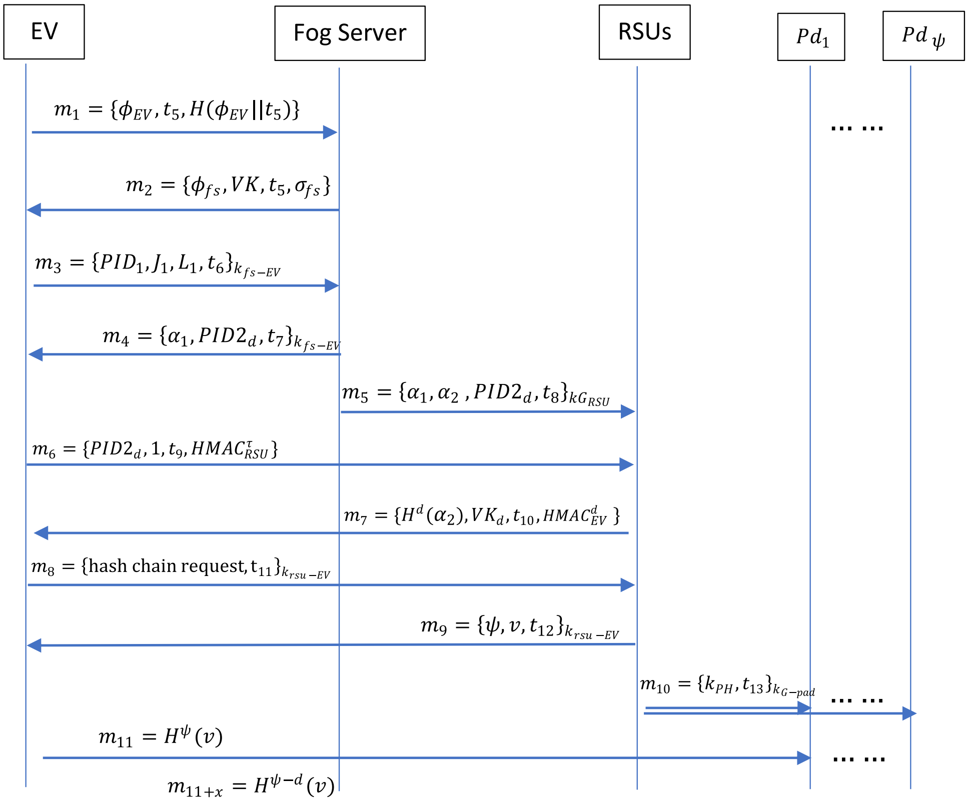

EV
Fog Server
RSUs
Pd_1
Pd_ψ
$m_1 = \{\phi_{EV}, t_5, H(\phi_{EV}||t_5)\}$
$m_2 = \{\phi_{fs}, VK, t_5, \sigma_{fs}\}$
$m_3 = \{PID_1, J_1, L_1, t_6\}_{k_{fs-EV}}$
$m_4 = \{\alpha_1, PID2_d, t_7\}_{k_{fs-EV}}$
$m_5 = \{\alpha_1, \alpha_2, PID2_d, t_8\}_{kG_{RSU}}$
$m_6 = \{PID2_d, 1, t_9, HMAC^{\tau}_{RSU}\}$
$m_7 = \{H^d(\alpha_2), VK_d, t_{10}, HMAC^{d}_{EV}\}$
$m_8 = \{\text{hash chain request}, \text{t}_{11}\}_{k_{rsu-EV}}$
$m_9 = \{\psi, v, t_{12}\}_{k_{rsu-EV}}$
$m_{10} = \{k_{PH}, t_{13}\}_{k_{G-pad}}$
$m_{11} = H^{\psi}(v)$
$m_{11+x} = H^{\psi-d}(v)$
… …
… …
… …

Fig. 5. Charging request phase.

equality is successful, the EV authenticates the RSU and uses the message values to calculate session key $k_{RSU-EV}' = H(H^1(\alpha_1||1) \oplus H^1(\alpha_2)')$. It also verifies the integrity of the key calculating $K_2 = H(k_{rsu-EV}')$, and compares $VK_2' =?VK_2$. If the equality is successful, the EV uses the session key to send an m_8 message containing a hash chain request to the RSU.

$$m_8 = \{\text{hash chain request},\ \text{t}_{12}\}_{k_{rsu-EV}}$$

The RSU receives, decrypts, checks the timestamp (t_{12}), and sends message m_9 to the EV. ψ is the number of keys to be authenticated in each pad and $\nu\, \varepsilon\, Z$ is a random number used as the initial value for the calculation of the hash chain. Additionally, the RSU sends all pads a message broadcast m_{10} encrypted with group key (k_{G-pad}) that contains public hash chain verification key $k_{PH} = H^{\psi+1}(\nu)$ used for the verification of the keys sent by EV.

$$m_9 = \{\psi,\ v,\ t_{13}\}_{k_{rsu-EV}}$$

$$m_{10} = \{k_{PH},\ t_{13}\}_{K_{G-pad}}$$

The EV receives and decrypts m_9 with values ψ and v, and computes hash chain $H^{\psi}(\nu)$. Each block of pads managed by the RSU receives and decrypts broadcast message m_{10} with the group key. The message contains public hash chain verification key $k_{PH} = H^{\psi+1}(\nu)$. Whenever a key from a hash chain is sent by the EV (m_{11}) to one of the pads, the pad checks if the key has been validated by iteratively applying $\xi - \psi$ (for $0 \le \xi \le \psi + 1$) times the hash function and compares it to the public key hash chain (verification key). If the verification is successful, the pad checks the status of the key in the revocation list. If the key has not been revoked, it accepts the key sent by the EV and revokes it to avoid double use. The process ends when the EV has passed over all pads.

Below is the mathematical proof of the signing blind pseudonym and fog server's signature verification:

- Signing blind pseudonym verification:

$$\begin{aligned} L &= ?H\left(PID, \hat{e}(J, P)\hat{e}(Q_{ccs}, Y_{ccs})^{-L}\right) \\ L &= H\left(PID, \hat{e}(J, P)\hat{e}(Q_{ccs}, Y_{ccs})^{-L}\right) \\ &= H\left(PID, \hat{e}\left(B.S_{ccs} + r.Y_{pub} + a.Y_{pub}, P\right)\hat{e}(-L.Q_{ccs}, x_{ccs}.P)\right) \\ &= H\left(PID, \hat{e}\left(B.S_{ccs} + r.Y_{pub} + a.Y_{pub}, P\right)\hat{e}(-(B-b).Q_{ccs}, x_{ccs}.P)\right) \\ &= H\left(PID, \hat{e}\left(B.S_{ccs} + r.Y_{pub} + a.Y_{pub}, P\right)\hat{e}((-B+b)(Q_{ccs}.x_{ccs}), P)\right) \\ &= H\left(PID, \hat{e}\left(B.S_{ccs} + r.Y_{pub} + a.Y_{pub}, -B.S_{ccs} + b.S_{ccs}, P\right)\right) \\ &= H\left(PID, \hat{e}\left(r.Y_{pub} + a.Y_{pub} + b.(Q_{ccs} * x_{css}), P\right)) \\ &= H\left(PID, \hat{e}\left(b.Q_{ccs} + R_i + a.P, Y_{pub}\right)\right) \end{aligned}$$

- Fog server's signature verification: $\hat{e}(\sigma_{fs}', P) =?\hat{e}(H(\phi_{fs}', VK', t_5'), Y_{fs})$

$$\begin{aligned} \hat{e}\ \left(\sigma_{fs}', P\right) &= \hat{e}\left(H\left(\phi_{fs}', VK', t_5'\right), Y_{fs}\right) \\ &= \hat{e}\left(H\left(\phi_{fs}', VK', t_5'\right), x_{fs} * P\right) \\ &= \hat{e}\left(x_{fs} * H\left(\phi_{fs}', VK', t_5'\right), P\right) \\ &= \hat{e}\left(\sigma_{fs}', P\right) \end{aligned}$$

Fig. 5 shows the flow of messages exchanged among the entities in the charging request phase.

6. Security and performance analyses

This section addresses an analysis of the security and performance of the protocol and a comparison with other protocols used for the authentication of a WPT-CWD system.

6.1. Security analysis

6.1.1. Security properties

Below is an analytical description of the security attributes, like mutual authentication, privacy preservation and integrity protection guaranteed by our protocol and a description of the way it resists attacks.

(1) Mutual Authentication: this process is established among FS, RSU and EVs. EVs authenticate FS by verifying message (m_2) signature. FS authenticates the valid ticket of an EV by verifying the blind signature sent in message 3 and using public parameters of the system. The RSU authenticates the EV by calculating the hash of message 6 containing an $\boldsymbol{\alpha}_1$ (delivered by the FS to the EV in message 4, and the RSU in message 6) sent by the EV. EVs authenticate to RSUs by verifying message 7 HMAC.

(2) Preservation of privacy: the EV identity is kept confidential by the CCS during the purchase of the tickets; FS, RSUs and pads are unable to obtain the user's identity from the ticket. The privacy of the location is also guaranteed, since the tickets and PIDs used by the EV in different locations cannot be correlated with a single vehicle.

(3) Protection to integrity: the integrity of the messages exchanged is maintained with the hash function and digital signatures. The system can identify whether an adversary manipulates the message by verifying the hash function value or the digital signature of the message.

(4) Perfect Forward Secrecy (PFS): the proposed protocol guarantees PFS as follows:

- In the process of creating session key $k_{(fs-EV)}$ between EV and FS to encrypt the messages, the random elements φ_{EV}, φ_{fs} and a blind message signature are used. Even if the session key $k_{(fs-EV)}$ is compromised, the previous messages cannot be recovered because of the CDH problem;
- In the process of creating a session key $k_{(RSU-EV)}$ between the EV and the RSU to encrypt the messages, the random elements α_1, α_2 and $PID2_1$ are used. Even if some or all of the random values are committed and the attacker manages to recreate the session key $k_{(RSU-EV)}$, previous messages cannot be recovered due to the CDH problem;
- In the process of creating the key $H^{\psi(v)}$ between the EV and the Pads, in the worst case when the seed v is compromised, the attacker will not be able to decipher the previous messages;
- If the CCS (X_{ccs}) private key is compromised, an attacker will not be able to recreate previous session keys and therefore decrypt old messages due to the random values used for generating session keys.

6.1.2. Prevention against attacks

Below are the different types of attacks that can affect the VANET network and a description of the way our protocol can resist them:

- Impersonation: An attacker that aims to enter the system using a false ticket cannot deceive the system, since it cannot sign the ticket correctly. On the other hand, session keys are generated whenever an EV uses a new ticket. It prevents the use of old parameters in other EVs or by itself.
- MITM: The use of digital signatures for the verification of the authenticity and integrity of messages m_2 and m_7 ensures that an MITM attack cannot be succesfull. On the other hand, when the EV performs an authentication process with the RSU, the EV sends a hash chain generated by the seed α_1 in message m_6, taking into account only an authentic EV can generate the valid hash chain, the MITM attack is mitigated.
- Replay and Injection: The use of a timestamp and random numbers in the messages avoids repetitive attacks and hash functions and digital signatures evidence the injection of data in the messages.
- Known key: Our protocol generates tickets which can be used only once. The ticket is added to the revocation list after its validity has been checked. Both system and EV generates random values for to create session keys, i.e., new session keys are generated for every new ticket for EV communication with the charging station, thus preventing an attacker from charging his/her car using old keys they may know.
- DoS: DoS attacks can affect the fog server and RSUs. In the first case, the fog server resists DoS attacks by validating tickets with public system parameters and revocation lists. In the second case, RSUs resist DoS attacks by efficiently validating connection requests using an HMAC code and verifying the auth variable $\boldsymbol{\alpha}_1$ in the revocation lists. Only users previously authenticated by the fog server have a valid $\boldsymbol{\alpha}$ **(alpha)** to create a valid HMAC. If an attacker attempts to connect to the RSU using an already used HMAC or a false HMAC, the RSU rejects the communication.
- Unlinkability: No entity can link PID_{ev} with a single EV, because the CCS blindly signs this value on the ticket (c_i). Moreover, the fog server checks the PID_{ev} (C', S') signature only with public parameters of the system.
- Double Spending: When an EV uses PID_{ev} and its signatures (C ', S') to authenticate to the fog server, PID_{ev} is revoked and published on a fog server's revocation list. In the authentication process, the fog server checks if PID_{ev} is on the list for terminating the continuing authentication process at the charging station. The same occurs in the EV authentication process in the RSU. PID_{ev} is revoked and published on a revocation list of RSUs.
- Random number leakage attack: to prevent this type of attack, the protocol uses the following operations and controls in relation to the PRNG system [29]:
  - A hash function will be executed on the inputs that are counted with a timestamp;
  - A hash function will be executed on the PRNG outputs;
  - In a period of random time, a new initial PRNG state will be generated;
  - Smart seed will be used at the starting points of the PRNG.
- Privileged insider attack: to prevent this type of attack, the company must establish security policies, internal processes and mechanisms for the prevention and detection of attacks. The following is a set of policies to be implemented in the system to prevent such attacks or mitigate damages [30]:
  - Awareness of security: the company's security policies and procedures must be known to all internal staff and external partners;
  - Classification of duties: it is necessary to classify the duties of employees and employers, to prevent or detect the attacks effectively;
  - Whirling of duties: when you have several important jobs, you should have several employees with the knowledge of the execution of these jobs; in each time period, these officials have to turn to different jobs to avoid malicious actions;
  - Limited privileges: limited access privileges (physical and in systems) must be given to officials to restrict access to confidential information or important company equipment;
  - Encrypt sensitive data: confidential data must be encrypted and stored in secure locations. The company must be backed up in the event that the system data is corrupted;

Table 2
Comparison of security properties.

	Li et al. [18]	Hussain et al. [19]	Gunukula et al. [20]	Rabie et al. [21]	Proposed protocol
Mutual authentication and key agreement	Yes	Yes	Yes	Yes	Yes
Confidentiality	**No**	Yes	Yes	Yes	Yes
Integrity	Yes	Untreated	Untreated	Untreated	Yes
Privacy	**No**	Yes	Yes	Yes	Yes
Injection attacks	Untreated	Untreated	Untreated	Untreated	Yes
Forward secrecy	Untreated	Untreated	Untreated	Untreated	Yes
Replay attack	**No**	Yes	Untreated	Yes	Yes
Known key attack	Untreated	Yes	Untreated	Untreated	Yes
DoS attack	Yes	Untreated	Untreated	Untreated	Yes
Man-in-the-middle attack	Yes	Untreated	Yes	Untreated	Yes
Impersonation attack	Untreated	Yes	Untreated	Untreated	Yes
Unlinkability	Untreated	Untreated	Untreated	Yes	Yes
Double spending	Untreated	Untreated	Yes	Yes	Yes
Random number leakage attack	Untreated	Untreated	Untreated	Untreated	Yes
Privileged insider attack	Untreated	Untreated	Untreated	Untreated	Yes
Masquerade attack	Untreated	Untreated	Untreated	Untreated	Yes

```
role role_EV(EV:agent,FS:agent,RSU:agent,PAD:agent,
            H1:function,H2:function,H3:function,H4:function,
            H5:function,CK:function,Kfsev:symmetric_key,
            Krsuev:symmetric_key,SND,RCV:channel(dy))
played_by EV
def=
      local
       State:nat,T5:text,Sigfs:text,T6:text,Vfifs:text,Vfiev:text,
       C:text,PID:text,S:text,T7:text,Tao:text,T8:text,Y:text,
       PID2:text,T10:text,HMAC:function,Sigrsu:text,T11:text,
       Alf1:text,M:function,Alf2:text,P:text,Req:text,T12:text,
       T13:text,Is:text,Psi:text
      init
       State := 0
      transition
       1. State=0 /\ RCV(start) =|> State':=1 /\ T5':=new()
       /\ P':=new() /\ Vfiev':=new() /\ secret(Vfiev',sec_5,{})
       /\ SND(M(Vfiev'.P').T5'.H1(M(Vfiev'.P').T5'))
       2. State=1 /\
RCV(M(Vfifs'.P).T6'.CK(M(Vfifs'.M(Vfiev.P))).Sigfs')
       =|> State':=2 /\ secret(Vfiev',sec_5,{}) /\
secret(Vfifs',sec_6,{})
       /\ T7':=new() /\ C':=new() /\ S':=new() /\ PID':=new()
       /\ SND({PID'.S'.C'.T7'}_Kfsev)
       4. State=2 /\ RCV({Alf1'.Tao'.PID2'.T8'}_Kfsev) =|>
State':=3
       /\ secret(Alf1',sec_1,{}) /\ T10':=new() /\ Y':=new()
       /\ SND(PID2'.Y'.T10'.HMAC(PID2'.Y'.T10'.Alf1'))
       7.            State=3            /\
RCV(M(H3(Alf2').P).T11'.CK(M(Alf1.M(Alf2'.P))).Sigrsu')
       =|> State':=4 /\ witness(EV,RSU,auth_10,Sigrsu')
       /\ secret(Alf2',sec_2,{}) /\ secret(Alf1',sec_1,{})
       /\    T12':=new()    /\    Req':=new()    /\
SND({Req'.T12'}_Krsuev)
       9. State=4 /\ RCV({Psi'.Is'.T13'}_Krsuev) =|>
State':=5
       /\ secret(Is',sec_4,{}) /\ secret(Psi',sec_3,{})
       /\ SND(H5(Psi'.Is'))
end role
```

Fig. 6. Role of EV in HLPSL.

  - ○ Defense in depth: a layered security policy must be implemented, where each layer has specific tasks for system protection.
- Masquerade attack: the proposed protocol is safe against server masking attacks, because an attacker cannot represent the response messages that are sent by the FS or RSU. The FS and RSU sign the contents of the response messages with their private key, so an attacker cannot recreate the signature of the response messages because they do not have the FS or RSU private key.

Table 2 shows a comparison of the security analysis between our protocol and other schemes for authentication for CWD-WPT load stations.

6.1.3. Formal verification of the proposed protocol

The protocol was formally verified by AVISPA, a commonly used tool for security protocol assessments. The entities and message exchanges were described by the HLPSL (High Level Protocol Specification Language) language [31].

AVISPA has four protocol validation modes called “Backends”, including On-the-Fly Model Checker (OFMC) and CL-AtSe (Constraint-Based Attack Locator). The results of the verification of a protocol are “SAFE”, if no problem has been detected, and “UNSAFE”, if an attack has been successful. AVISPA provides a report only when the result is "UNSAFE". The report addresses the successfully executed attack.

Modeling of the proposed protocol in HLPSL. The HLPSL language enables the construction of a protocol model to be evaluated. Figs. 6–8 show some of the HLPSL codes that modelled our protocol.

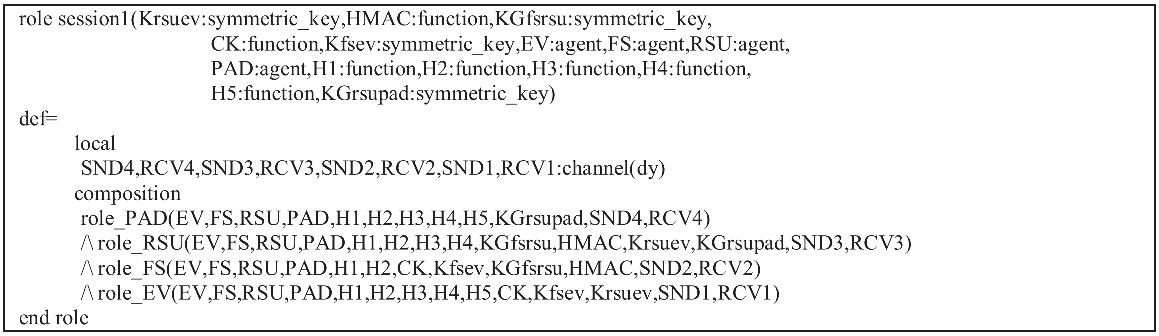
Fig. 6 displays the HLPSL code that models the behavior of the EV in our protocol. The structure of the code is the same as those of the codes of other entities (CCS, FS and PAD) and consists of the following parts:

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

States are created immediately after the creation of the aforementioned declarations and describe the operations and messages to be exchanged between entities. At the end of each state, the elements that must be kept confidential and authenticated variables are declared.

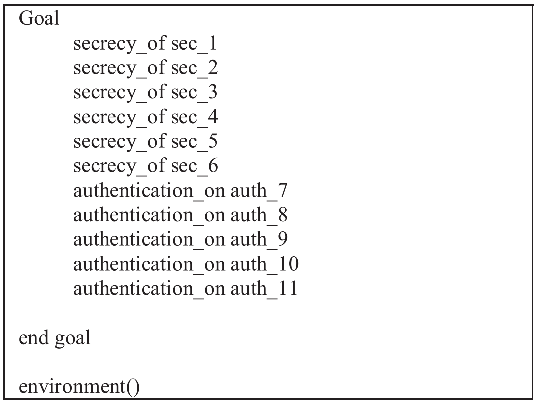
Fig. 7 shows the HLPSL language code that describes the establishment of the sessions and the environment of the execution of the protocol. The elements (variants, keys, agents, etc.) likely to be acquired by an attacker are also declared.

Finally, Fig. 8 shows the security objectives to be guaranteed by the protocol according to the definition of elements declared as se-



```
role session1(Krsuev:symmetric_key,HMAC:function,KGfsrsu:symmetric_key,
                              CK:function,Kfsev:symmetric_key,EV:agent,FS:agent,RSU:agent,
                              PAD:agent,H1:function,H2:function,H3:function,H4:function,
                              H5:function,KGrsupad:symmetric_key)
def=
        local
         SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
        composition
         role_PAD(EV,FS,RSU,PAD,H1,H2,H3,H4,H5,KGrsupad,SND4,RCV4)
         /\ role_RSU(EV,FS,RSU,PAD,H1,H2,H3,H4,KGfsrsu,HMAC,Krsuev,KGrsupad,SND3,RCV3)
         /\ role_FS(EV,FS,RSU,PAD,H1,H2,CK,Kfsev,KGfsrsu,HMAC,SND2,RCV2)
         /\ role_EV(EV,FS,RSU,PAD,H1,H2,H3,H4,H5,CK,Kfsev,Krsuev,SND1,RCV1)
end role
```

Fig. 7. Specification of the session role in HLPSL.



```
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
        authentication_on auth_7
        authentication_on auth_8
        authentication_on auth_9
        authentication_on auth_10
        authentication_on auth_11

end goal

environment()
```

Fig. 8. Security objectives and related secrets of our protocol in HLPSL.

crets in the functions of the entity and the values that authenticate the entities.

- secrecy_of sec_1: keep secret α_1;
- secrecy_of sec_2: keep secret α_2
- secrecy_of sec_3: keep secret ψ
- secrecy_of sec_4: keep secret υ
- secrecy_of sec_5: keep secret ϕ_{EV}
- secrecy_of sec_6: keep secret ϕ_{fs}
- authentication_on auth_7: EV authenticates FS on σ_{fs};
- authentication_on auth_8: FS authenticates EV on PID_1;
- authentication_on auth_9: RSU authenticates EV on α_1;
- authentication_on auth_10: EV authenticates RSU on $H(\alpha_2)$;
- authentication_on auth_11: PAD authenticates EV on υ;

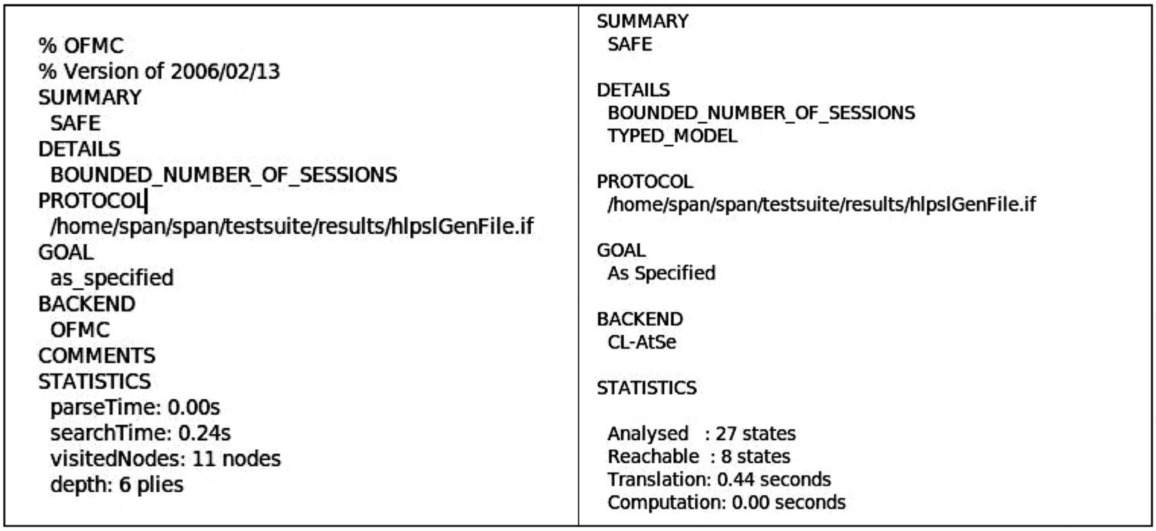
Security check results. Simulations in AVISPA with OFMC and CL-AtSe backends checked the security of the protocol, which was considered safe for both backends, according to the results (see Fig. 9).

6.2. Performance analysis

This subsection reports on a performance analysis of computational and communications costs. The authentication procedures between the fog server and EVs (FS-EVs), EV and RSUs (EVs-RSU), and EVs and pads (EVs-pads) were assumed independent, since those processes can be conducted in different time periods and locations. For example, an EV can authenticate the fog server far from the charging station with considerable time in advance. The following EVs-RSUs authentication process can be performed hundreds of meters from the first pad and several seconds in advance. Lastly, an EV must be authenticated by the pad a few centimeters from it and microseconds in advance.

6.2.1. Communication costs

We consider that this transmission uses high-coverage communication technology such as LTE, so that the EV is able to perform the exchange of information with the FS before entering the CWD-WPT charging station. For communications within the CWD-WPT



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/hlpslGenFile.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.24s
 visitedNodes: 11 nodes
 depth: 6 plies
```

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

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

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 27 states
 Reachable  : 8 states
 Translation: 0.44 seconds
 Computation: 0.00 seconds
```

Fig. 9. Security simulation results for OFMC and CL-AtSe backends.

Table 3
Symbols and costs in bytes [21].

Symbol	Description	Length (Bytes)
ID	Identification	128
PID	Pseudo identity	32
$H()$	Hash function	32
X	Private key	32
Y, Q	Public key	32
k	Session key	32
σ	Digital signature	32
(J, L)	Blind signature	96
ϕ	Pre-key of session	32
τ	Number of RSUs for fog server	8
ψ	Number of pads for RSU	8
α, v	Seed	20
t	Timestamp	8
VK	Verification key	32
hash chain request	Hash chain request	8
*	Multiplication operator	–
ê	Bilinear Pairing	–
CCS	Authentication Server of the substation	–
RSU	Central Authentication Server	–
HMAC	Hash-based message authentication code	32
P	Point of the elliptical curve	32

Table 4
Communication costs in bytes.

Message	Gunukula et al. [20]	Rabie et al. [21]	Propoced
M1	$32n$	$224n$	$72n$
M2	$128n$	$248n$	$104n$
M3	$168n$	$128n$	$136n$
M4	$136n$	$128n$	$64n$
M5	$32(n^*\tau)$	$40(n^*\tau)$	80n
M6	$32(n^*\tau)$	$40(n^*\tau)$	$80(n^*\tau)$
M7	$32(n^*\tau)$	$32(n^*\tau^*\psi)$	$104(n^*\tau)$
M8	$20(n^*\tau)$	–	$16(n^*\tau)$
M9	$32(n^*\tau^*\psi)$	–	$32(n^*\tau)$
M10	–	–	$32n$
M11	–	–	$32(n^*\tau^*\psi)$
Total	$\boldsymbol{n}\,(464+\boldsymbol{\tau}\,(116+32\boldsymbol{\psi}))$	$\boldsymbol{n}\,(728+\boldsymbol{\tau}\,(80+32\boldsymbol{\psi}))$	$\boldsymbol{n}\,(488+\boldsymbol{\tau}\,(232+32\boldsymbol{\psi}))$

charging station (EV-RS and RSU-PAT Communications) DSRC communications technology would be used which, within the effective communication range, has better communication performance than LTE. As in [32], the combination of DSRC and LTE has been considered a good solution for VANET".

Communication cost refers to the total number of bytes transmitted by a network during the execution of a protocol. Table 3 shows the values in bytes of each variable used. (Values taken from Rabieh and Wei [21]).

To calculate the communication costs using Table 3 of an EV that will authenticate to the fog server, the first RSU and the first pad, we have:

- $m_1 = \{\phi_{EV},\ t_5,\ H(\phi_{EV}||t_5)\} = 32+8+32 = 72\ Bytes$
- $m_2 = \{\phi_{fs},\ VK,\ \ t_6,\ \sigma_{fs}\} = 32+32+8+32 = 104\ Bytes$
- $m_3 = \{PID_1, J_1, L_1, t_7\}_{k_{fs-EV}} = 32+96+8 = 136\ Bytes$
- $m_4 = \{\alpha_1, \tau,\ \ PID2_1,\ \ t_8\}_{k_{fs-EV}} = 16+8+32+8 = 64\ Bytes$
- $m_5 = \{\alpha_1, \alpha_2, \tau, PID2_1, t_9\}_{k_{G-RSU}} = 16+16+8+32+8 = 80\ Bytes$
- $m_6 = \{PID2_{EV}, 1, t_{10}, HMAC^{\tau}_{RSU}\} = 32+8+8+32 = 80\ Bytes$
- $m_7 = \{H(\alpha_2)^{\tau}, VK_2, t_{11}, HMAC^{\tau}{}_{EV}\} = 32+32+8+32 = 104\ Bytes$
- $m_8 = \{\text{hash chain request},\ \ \text{t}_{12}\}_{k_{rsu-EV}} = 8+8 = 16\ Bytes$
- $m_9 = \{\psi,\ \ v,\ \ t_{13}\}_{k_{rsu-EV}} = 8+16+8 = 32\ Bytes$
- $m_{10} = \{k_{PH},\ \ t_{13}\}_{K_{G-pad}} = 32+8 = 40\ Bytes$
- $m_{11} = \{H(v)^{\psi}\} = 32\ Bytes$

Table 4 shows the comparison of communication costs between the protocols proposed by Gunukula et al. [20], Rabie et al. [21] and our protocol, counting the bytes (according to Table 3) of the messages exchanged between entity pairs and the total number of messages exchanged by $\boldsymbol{n}$ EVs that try to enter the wireless charging system composed of $\boldsymbol{\tau}$ RSUs and $\boldsymbol{\psi}$ pads charging by RSUs.

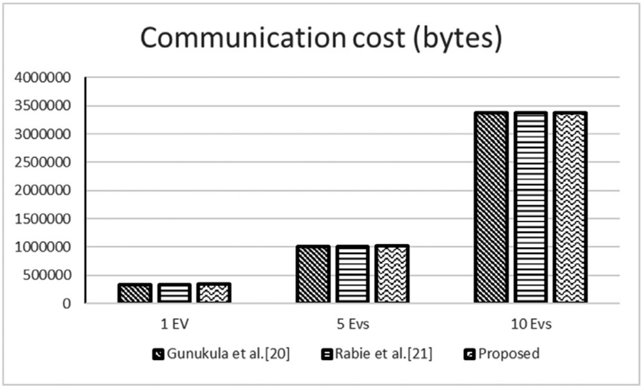


Fig. 10. Communication cost comparisons.

Fig. 10 shows a comparison of the communication costs the protocols proposed in references [20,21] and our protocol. The values adopted for evaluation of computational costs are based on Li et al. [33], who proposed parameters for the modeling of a typical CWD-WPT charging station. According to [33], CWD-WPT is 4,2 km long and the pads are 40 cm long and separated by a 40 cm length. In [20], the RSUs are distributed every 600 m, i.e., 7 RSUs are managed by the fog server and 1500 charging pads are managed by an RSU. It can be verified that the costs of the 3 (three) proposals are very similar, and can be slightly differ, depending on the values of $\boldsymbol{n}$, $\boldsymbol{\tau}$ and $\boldsymbol{\psi}$, reflecting the structure of CWD-WPT based network.

Table 5
Costs in ***ms*** of each operation and entity considered (adapted from [34]).

Entity	Parameters of the entities involved			Costs (ms)					
	CPU(GHz)	RAM	OS	T_{mul}	T_{exp}	T_{pair}	T_{hash} [20]	T_{g-sig} [21]	T_{v-sig} [21]
EV/PAD	Qualcomm(R) Octa-core 1.5	2	Android 4.2.2	0,50	0,54	16,6	$0{,}043x10^3$	0,6	0,78
RSUs	Intel(R) Dual-core 3.1	4	64-bit Win-7	0,36	0,38	11,5	$0{,}03x10^3$	0,42	0,55
CCS/CMC/FS	Intel(R) Hexa-core 1.6	16	16 Win server 2012	0,3	0,31	8,6	$0{,}025x10^3$	0,36	0,47

Table 6
Computational costs.

Protocols	EV	CSP-BNK/CMC/FS	RSU	PAD
Gunukula et al. [20]	$2\boldsymbol{T_{exp}}+((\boldsymbol{\tau}+1)^2+(\boldsymbol{\psi}+1)+4)\boldsymbol{T_{hash}}$ $+1\boldsymbol{T_{v-sig}}$	$2\boldsymbol{nT_{exp}}+4\boldsymbol{nT_{mul}}+((\boldsymbol{\tau}+1)\boldsymbol{n})^2\boldsymbol{T_{hash}}+$ $1\boldsymbol{nT_{g-sig}}+2\boldsymbol{nT_{pair}}$	$(2\boldsymbol{n}+((\boldsymbol{\tau}+1)\boldsymbol{n})^2\boldsymbol{T_{hash}}$	$\boldsymbol{n}(1-\boldsymbol{\psi}))\boldsymbol{T_{hash}}$
Rabie et al. [21]	$2\boldsymbol{T_{exp}}+(3+\boldsymbol{\psi})\boldsymbol{T_{hash}}\ +2\boldsymbol{T_{v-sig}}$	$5\boldsymbol{nT_{exp}}+4\boldsymbol{nT_{mul}}((3+\boldsymbol{\psi})\boldsymbol{\tau})\boldsymbol{nT_{hash}}+2\boldsymbol{nT_{g-sig}}$ $+2\boldsymbol{nT_{pair}}$	–	$\boldsymbol{n}(\boldsymbol{\psi})\boldsymbol{T_{hash}}$
Proposed	$2\boldsymbol{T_{mul}}+((1+\boldsymbol{\psi})+4)\boldsymbol{T_{hash}}+\ 1\boldsymbol{T_{v-sig}}$	$2\boldsymbol{nT_{mul}}+1\boldsymbol{nT_{exp}}+1\boldsymbol{nT_{g-sig}}+4\boldsymbol{nT_{hash}}+2\boldsymbol{nT_{pair}}$	$4\boldsymbol{nT_{hash}}$	$\boldsymbol{n}(\boldsymbol{\psi})\boldsymbol{T_{hash}}$

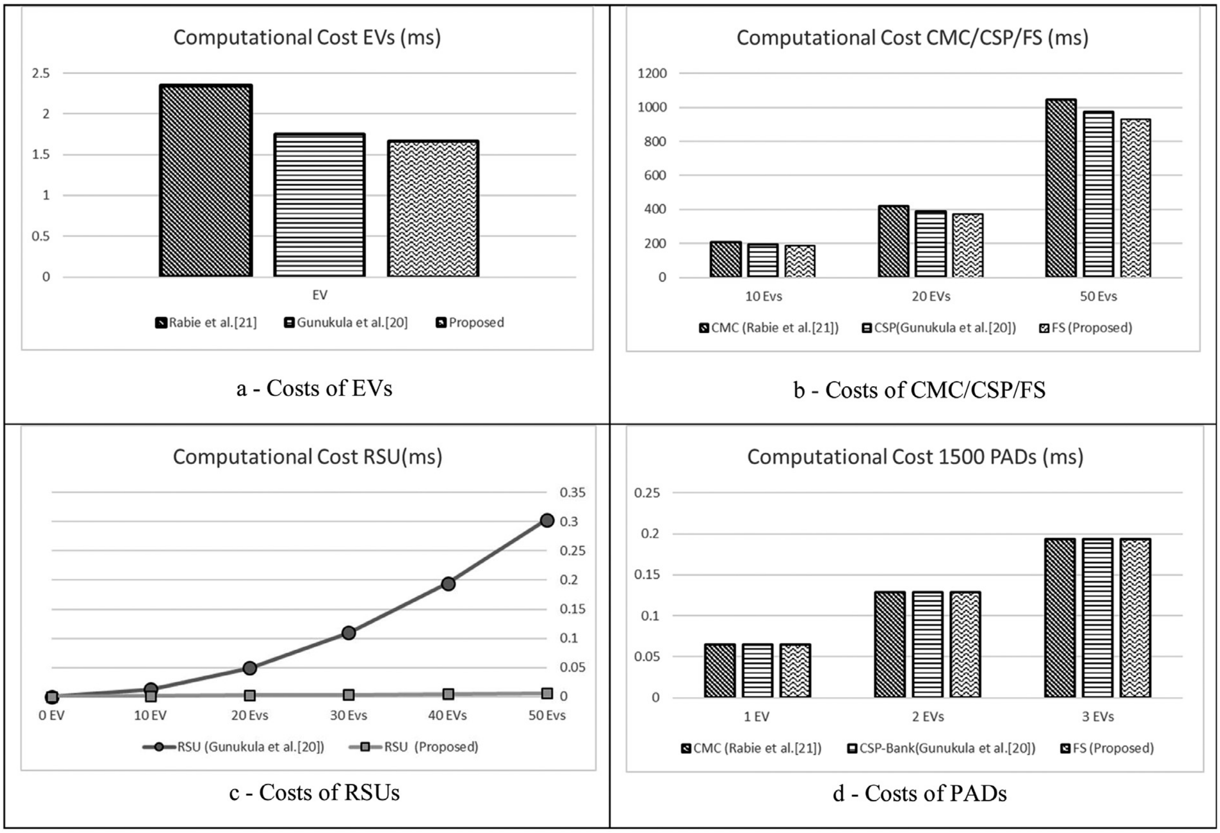
6.2.2. Computational costs

Below is the calculation of the computational costs of the entities of the network model. Table 5 shows the execution times of the Multiplication (T_{mul}), Exponentiation (T_{exp}) and Bilinear Pairing (T_{pair}) functions based on Tao et al. [34], for each entity. The execution costs of the Hash (T_{hash}) function for EV are based on Gunukula et al. [20]. The execution costs for generating a signature message (T_{g-sig}) and its verification (T_{v-sig}) are based on Rabieh et al. [21]. The execution costs of the hash function, signature message and message signature verification for RSU and FS were calculated analytically, taking 70% and 60%, respectively, from the cost of executing these operations to an EV.

The time costs of operations, as symmetric encryption/decryption and addition, have been omitted, because their execution times are very short and rarely used in the protocol, in comparison to the Hash operation.

Table 6 shows a comparison of the number of operations performed by the protocols of Gunukula et al. [20], Rabie et al. [21] and the proposed protocol. Like the other protocols, the proposed protocol performs the operations with higher computational costs in the entity with greater computational capacity (in our case the FS). On the other hand, entities with lower capacity such as EV, RSU, and PADs perform less complex operations to ensure lower latency for the CWD-WPT scheme.

In Fig. 11a comparison of the total computational costs of each entity is shown in the authentication phase of the protocols of Gunukula et al. [20], Rabie et al. [21] and the proposed protocol. The proposed protocol has a better computational cost for EVs, FS



a - Costs of EVs

b - Costs of CMC/CSP/FS

c - Costs of RSUs

d - Costs of PADs

Fig. 11. Computational costs.

and RSU, and maintains the same computational costs of the other protocols for a group of 1500 pads.

7. Conclusions

Communicating things networks (CTN) is the basis for IoT services and therefore must adapt to the particular requirements of each service, such as QoS, data volumes, mobility and interconnection between different devices.

The combination of cloud computing and fog computing in a hierarchical scheme is an effective solution to support next-generation VANETs that are compatible with high mobility, low latency, real-time services and connectivity.

Part of the research related to EVs has been directed at the creation of VANET networks in a cloud environment to support CWD-WTP charging stations. Such stations aim at the optimization and simplification of the charge of EV batteries, since, in this system, cables are not necessary and power is induced while the EV owners drive to their destination.

This work addresses the problems of network security and access control in cloud-based vehicular networks, meeting the most important security requirements such as: authentication, data integrity, confidentiality, access control, non-repudiation and availability. In this sense, this paper aims to contribute for the optimization and security of vehicular networks that support EVs, which has become a trend in several countries due to the global objective to reduce air pollution. The manuscript introduced a new authentication protocol for CWD-WPT charging systems on a VANET network in a cloud and fog computing environment; it is based on digital signatures, HMACs and hashing chains. A short description of some work on authentication in CWD-WPT charging systems has also been provided.

In comparison with other proposals, our scheme has yielded better computational costs and provides better results regarding security analysis and more complete results regarding safety analysis, and avoided problems related to centralization caused by the use of a cloud environment composed of fog computing and cloud computing. Such a combination promotes a better distribution of the computational processing of operations in the devices and guarantees lower latency in communications. Moreover, the protocol has met the security objectives, according to a formal verification conducted by AVISPA tool.

Future work will involve the interaction of EVs in provider, consumer or energy storage modes in CWD-WPT systems, and a simulation of the protocol will be conducted in a network simulator. Another line of work involves authentication and authorization protocols for cyber-physical systems (CPS), and the development of computational trust models for CWD-WPT systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Sathishkumar, D.R. Patel, Enhanced location privacy algorithm for wireless sensor network in Internet of Things, in: Proceedings of the International Conference on Internet Things Applications IOTA, 2016, pp. 208–212.

[2] T. Park, N. Abuzainab, W. Saad, Learning how to communicate in the Internet of Things: finite resources and heterogeneity, IEEE Access 4 (2016) 7063–7073.

[3] F.J. Soares, D. Rua, C. Gouveia, B.D. Tavares, A.M. Coelho, J.A.P. Lopes, Electric vehicles charging: management and control strategies, IEEE Veh. Technol. Mag. 13 (1) (2018) 130–139.

[4] X. Mou, O. Groling, H. Sun, Energy-efficient and adaptive design for wireless power transfer in electric vehicles, IEEE Trans. Ind. Electron. 64 (9) (2017) 7250–7260.

[5] T.V. Theodoropoulos, I.G. Damousis, A.J. Amditis, Demand-side management ICT for dynamic wireless EV charging, IEEE Trans. Ind. Electron. 63 (10) (2016) 6623–6630.

[6] Y.J. Jang, Survey of the operation and system study on wireless charging electric vehicle systems, Transp. Res. Part C Emerg. Technol. 95 (2018) 844–866 November 2017.

[7] C. Li, T. Ding, X. Liu, C. Huang, An electric vehicle routing optimization model with hybrid plug-in and wireless charging systems, IEEE Access 6 (2018) 27569–27578.

[8] D. Bavastro, A. Canova, V. Cirimele, F. Freschi, L. Giaccone, P. Guglielmi, M. Repetto, Design of wireless power transmission for a charge while driving system, IEEE Transactions on Magnetics 50 (2) (2014) 2–5.

[9] K. Nahrstedt, H. Li, P. Nguyen, S. Chang, L. Vu, Internet of mobile things: mobility-driven challenges, designs and implementations, in: Proceedings of the IEEE 1st International Conference on Internet-of-Things Design Implementation, IoTDI, 2016, pp. 25–36.

[10] R. Shrestha, R. Bajracharya, S.Y. Nam, Challenges of future VANET and cloud-based approaches, Wirel. Commun. Mob. Comput. 2018 (2018).

[11] M. Ziqian, Z. Guan, Z. Wu, A. Li, Z. Chen, Security enhanced internet of vehicles with cloud-fog-dew computing, ZTE Commun. 15 (S2) (2018) 47–51.

[12] Q.G.K. Safi, S. Luo, C. Wei, L. Pan, Q. Chen, PIaaS: cloud-oriented secure and privacy-conscious parking information as a service using VANETs, Comput. Netw. 124 (2017) 33–45.

[13] C. Huang, R. Lu, K.K.R. Choo, Vehicular fog computing: architecture, use case, and security and forensic challenges, IEEE Commun. Mag. 55 (11) (2017) 105–111.

[14] A. Singh, K. Chatterjee, Cloud security issues and challenges: a survey, J. Netw. Comput. Appl. 79 (2017) 88–115 August 2016.

[15] R. Shrestha, R. Bajracharya, S.Y. Nam, Challenges of future VANET and cloud-based approaches, Wirel. Commun. Mob. Comput. 2018 (2018).

[16] R. Akalu, Privacy, consent and vehicular ad hoc networks (VANETs), Comput. Law Secur. Rev. 34 (1) (2018) 175–179.

[17] R.G. Engoulou, M. Bellaïche, S. Pierre, A. Quintero, VANET security surveys, Comput. Commun. 44 (2014) 1–13.

[18] H. Li, G. Dán, K. Nahrstedt, Proactive key dissemination-based fast authentication for in-motion inductive EV charging, in: Proceedings of the IEEE International Conference on Communications, 2015.

[19] R. Hussain, D. Kim, M. Nogueira, J. Son, A. Tokuta, H. Oh, A new privacy-aware mutual authentication mechanism for charging-on-the-move in online electric vehicles, in: Proceedings of the Eleventh International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2015, 2016.

[20] S. Gunukula, A.B.T. Sherif, B. Ausby, M. Mahmoud, and X.S. Shen, Efficient scheme for secure and privacy-preserving electric vehicle dynamic charging system, 2017.

[21] K. Rabieh, M. Wei, Efficient and privacy-aware authentication scheme for EVs pre-paid wireless charging services, in: Proceedings of the IEEE International Conference on Communications, 2017.

[22] S. Laporte, G. Coquery, M. Revilloud, V. Deniau, Experimental performance assessment of a dynamic wireless power transfer system for future ev in real driving conditions 1 extended abstract, in: Proceedings of the Ninth International Conference on Future Energy System, e-Energy, 2018, pp. 570–578.

[23] B. Roberts, K. Akkaya, E. Bulut, M. Kisacikoglu, An authentication framework for electric vehicle-to-electric vehicle charging applications, IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2017, pp. 565–569.

[24] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inf. Theory IT-29 (2) (1983) 198–208.

[25] R. Dutta, R. Barua, S. Palash, Pairing-based cryptographic protocols : a survey, in: Proceedings of the IACR Cryptology, 2004, pp. 1–45. ePrint Arch..

[26] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, in: Proceedings of the Seventh International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ASIACRYPT, 2001.

[27] F. Zhang, K. Kim, ID-based blind signature and ring signature from pairings, in: Advances in Cryptology — ASIACRYPT, 2501, 2002, pp. 533–547. December 2002.

[28] Y. Hu, M. Jakobsson, and A. Perrig, Efficient constructions for one-way hash chains, ICH Q6B, Specifications: Test Procedures and Acceptance Criteria for Biotechnological/Biological Product, no. 1, pp. 1–13, 2001.

[29] J. Kelsey, B. Schneier, D. Wagner, C. Hall, Cryptanalytic attacks on pseudorandom number generators, in: S. Vaudenay (Ed.), Fast Software Encryption - FSE, vol. 1372, Springer, 1998, pp. 168–188.

[30] T. Gunasekhar, K.T. Rao, M.T. Basu, Understanding insider attack problem and scope in cloud, in: Proceedings of the IEEE International Conference on Circuit, Power Computing Technologies ICCPCT, 2015 March 2015.

[31] The AVISPA project: Eeuropean union in the future and emerging technologies (FET open)-AVISPA v1. 1 user manual, 2006. [Online]. Available: http://www.avispa-project.org.

[32] Z. Xu, X. Li, X. Zhao, M.H. Zhang, Z. Wang, DSRC versus 4G-LTE for connected vehicle applications: a study on field experiments of vehicular communication performance, J. Adv. Transp. 2017 (2017) 1–10.

[33] H. Li, G. Dan, K. Nahrstedt, Portunes+: privacy-Preserving fast authentication for dynamic electric vehicle charging, IEEE Trans. Smart Grid 8 (5) (2017) 2305–2313.

[34] M. Tao, K. Ota, M. Dong, Z. Qian, AccessAuth: capacity-aware security access authentication in federated-IoT-enabled V2G networks, J. Parallel Distrib. Comput. 118 (2018) 107–117.

Luis Fernando Arias Roman is a Doctorate degree student in Electrical Engineering at the Universidade de Brasília (Brazil) with CAPES scholarship. He holds a Master's degree in Electrical Engineering from Universidade de Brasília (Brazil), a specialization course in Computer Security from Universidad Autonoma de Occidente (Colombia) and also a degree in Electronic and Telecommunications Engineering from Universidad del Cauca(Colombia). He has 8 years of professional experience in areas of communications networks and computer security, and is currently developing research projects in same areas.

Paulo R. L. Gondim obtained a Bachelor's degree in Computer Engineering in 1987, a Master's degree in Systems and Computing from Instituto Militar de Engenharia in 1992 in Rio de Janeiro, and a Doctorate degree in Electrical Engineering from Pontifícia Universidade Católica do Rio de Janeiro in 1998. Since 2003, He has been working at Universidade de Brasília, Brasília, Brazil. He has authored over 70 papers in international periodicals and events and advised 25 master's degree dissertations and 03 doctoral theses. He has been a member of Technical Program Committees of several technical-scientific events and contributed as a reviewer of papers submitted to high-quality periodicals and events. He is a member of the Editorial Advisory Boards of two international journals and has experience in Computer Science, particularly in wireless networking, video streaming, performance evaluation and quality of service/quality of experience assessment. He is a Senior Member of IEEE.