electronics

MDPI

Article

Certificate Based Security Mechanisms in Vehicular Ad-Hoc Networks based on IEC 61850 and IEEE WAVE Standards

Shaik Mullapathi Farooq [1], S. M. Suhail Hussain [2,*], Siddavaram Kiran [1] and Taha Selim Ustun [2]

[1] Department of Computer Science and Engineering, YSR Engineering College, Yogi Vemana University, Andhra Pradesh 516360, India; shaikfaroq@gmail.com (S.M.F.); rkirans125@yogivemanauniversity.ac.in (S.K.)

[2] Fukushima Renewable Energy Institute, AIST (FREA), Koriyama 963-0215, Japan; selim.ustun@aist.go.jp

* Correspondence: s.suhail.md@gmail.com; Tel.: +81-804-373-5146

Received: 13 November 2018; Accepted: 10 January 2019; Published: 15 January 2019

Abstract: When equipped with an on-board wireless kit, electric vehicles (EVs) can communicate with nearby entities, e.g., road side units (RSUs), via a vehicle ad-hoc network (VANET). More observability enables smart charging algorithms where charging stations (CSs) are allocated to EVs based on their current state of charge, destination, and urgency to charge. IEEE 1609 WAVE standard regulates VANETs, while IEC 61850 is emerging as the smart grid communication standard. In order to integrate these two domains of energy management, past research has focused on harmonizing these two standards for a full smart city solution. However, this solution requires very sensitive data to be transmitted, such as ownership of EV, owners' personal details, and driving history. Therefore, data security in these networks is of prime concern and needs to be addressed. In this paper, different security mechanisms defined by the IEEE 1609 WAVE standard are applied for both vehicle-to-infrastructure (V2I) and vehicle-to-grid (V2G) communication. The former relates to EV–RSU, while the latter covers EV–CS communication. The implicit and explicit certificate mechanism processes proposed in IEEE 1609 WAVE for authentication are studied in great detail. Furthermore, a performance evaluation for these mechanisms is presented in terms of total time lapse for authentication, considering both the computational time and communication time delays. These results are very important in understanding the extra latency introduced by security mechanisms. Considering that VANETs may be volatile and may disappear as EVs drive away, overall timing performance becomes vital for operation. Reported results show the magnitude of this impact and compare different security mechanisms. These can be utilized to further develop VANET security approaches based on available time and the required security level.

Keywords: electric vehicle (EV); road side unit (RSU); V2G; V2I; implicit and explicit certificate mechanisms; cybersecurity in smart grids

1. Introduction

Intelligent transport systems (ITS) improve traveler safety, decrease traffic congestion, facilitate the reduction of air pollution, provide vehicle information, and contribute to protecting natural resources. The applications of ITS extend to incident management system processes, electronic road toll collection, and in-car navigation systems [1]. The use of wireless technology fits well in providing the medium to connect any vehicle to the outside world [2]. An electric vehicle (EV) integrated with wireless sensors and communication devices forms vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication networks, which enable EVs to communicate with surroundings such as road side

units (RSUs) and nearby EVs. For this purpose, dedicated short-range communication (DSRC) is used in vehicular networks [3].

EVs offer a cleaner transportation solution, which can help achieve CO_2 emission reduction goals [4]. During idle times, EVs can supply energy to a grid via vehicle-to-grid (V2G) technology to meet peak level demands [5]. Similarly, when plenty of renewable energy is available within the grid, EVs can charge with grid-to-vehicle (G2V) technology. To effectively manage the participation of EVs in V2G and G2V, robust, interoperable, and standardized communication is required [6]. In this regard, the authors in Reference [7] developed a charging management scheme over a communication network that harmonizes IEEE 1609 WAVE [8] services and IEC 61850-90-8 information models [9]. The proposed EV charging management scheme [7] executes V2I and V2G communications, i.e., EV–RSU and EV–CS communication, following IEEE 1609 WAVE over IEC 61850-90-8 standards, respectively. However, the cybersecurity related aspects of V2I and V2G communication were not considered. Open communication between EVs and RSUs or CSs creates vulnerabilities for EV owners which must be addressed.

The authors in Reference [10], discussed different types of possible attacks on V2I communication, such as impersonation attack, Sybil attack, modification attack, ID disclosure, and Denial of Service (DoS) attack. IEEE WAVE 1609.2 defines security algorithms and mechanisms to be employed in vehicular networks [11]. The major security requirements identified in IEEE 1609.2 are authenticity, authorization, integrity, and non-repudiation. In the literature, authentication rises up as the key security requirement in vehicular networks [12,13]. The authors in Reference [14] proposed a secure authentication scheme, based on Rivest-Shamir-Adleman (RSA) digital signatures, for authenticating vehicle eID cards in V21 and V2V communication. It was concluded that faster cryptographic algorithms, such as elliptic curve cryptography (ECC), can be applied to improve the timing performance of the authentication process [14,15]. The authors in Reference [16] presented an authentication and key agreement protocol using ECC for smart vehicles, and presented performance evaluations in terms of communication overhead and cost. The authors in Reference [17] proposed a privacy-preserving fast authentication scheme, portunes+, for V2G communication, and presented the computational time needed to implement the authentication scheme. Similarly, in References [18,19], authors proposed light weight authentication schemes and presented performance evaluations of the proposed authentication schemes in terms of computational time and communication overhead. However, the communication delays in transmitting the required messages during the authentication process were not considered. The communication delays incurred while transmitting authentication messages also contribute to the overall time lapse for the authentication process. The time lapse for implementing security mechanism holds more importance in V2I communication as compared to V2G communication, since the V2I communication is a dynamic VANET which has a small time window for implementing the security mechanism and message exchanges. V2G communication, on the other hand, is based on a static network and has no time constraints for implementing security mechanisms.

Regular authentication mechanisms generally consist of two parts: the first part is related to signing of messages, and the second to the verification of these messages. IEEE 1609.2 has proposed the use of implicit and explicit certificate mechanisms for authentication. This paper analyses these mechanisms in detail and presents performance evaluations in terms of total time lapse for authentication. The performance evaluations were carried out using the Python OpenSSL Library [20] implementations and Riverbed Modeler [21] simulation tools. The python-based program was developed using OpenSSL libraries to generate certificates, implement the authentication mechanisms, and calculate the required computational time. Riverbed Modeler simulations were run to observe the time delays for transporting the certificates between the EV and RSU.

The rest of this paper is organized as follows: Section 3 presents a brief introduction of the protocol stack of IEEE WAVE and its security vulnerabilities. Section 4 describes the public key infrastructure (PKI) and certificate authority (CA) utilized to implement security mechanisms. Section 5

presents performance evaluations of different authentication implementations. Finally, Section 6 draws conclusions.

2. Related Work

Researchers have proposed a number of authentication schemes to achieve fast computation, privacy preservation, and scalability in V2I and V2G VANET environments. The authors in References [22,23] proposed pseudonym-based, privacy preserving authentication schemes based on PKI, which consists of thousands of public and private key pairs along with corresponding certificates installed in a vehicle's onboard unit (OBU). Each beacon message is signed by private key and attached with a public key certificate. A pseudo identity is also attached to the certificate. The receiver verifies the message using the public key of the sender. In case of malicious activity, a third party certificate authority (CA) identifies the real identity of sender using the pseudo identity. The scheme, however, incurs high communication overhead and computational latency, due to exponential increases in the certificate revocation list (CRL) in the case of revocation of certificates. The authors in References [24,25] describe an ID-based signature scheme that does not use a public key certificate. The authors in References [26,27] proposed a group authentication scheme to overcome high communication overhead and verification latency. The scheme allows vehicles to communicate securely within a known group of vehicles. In this scheme, a group of vehicles is formed to hide the identity of the sender. Each group has an administrator. The message is signed by the individual group key and subsequently verified by the group public key certificate. If the group administrator is compromised, then entire group of vehicles is compromised. The scheme reduces complexity in management of the CRL. By combining group signature and ID-based signature schemes, the authors in Reference [28] proposed a secure VANET identity authentication scheme. RSU-aided authentication schemes were proposed in Reference [29]. RSUs share the system load, which improves overall performance, but the scheme requires pervasive deployment of RSUs. To eliminate the role of RSUs, cooperative authentication schemes have been proposed [6,7]. The methods proposed can verify messages only in the case of high vehicle density on the road. If the vehicle density is low, then it may lead to unreliable authentication, causing location-based attacks. The authors in Reference [11] proposed a ring signature scheme which, unlike group signatures, protects vehicle identity. Table 1 gives a brief summary of the different authentication mechanisms for VANETs reported in the literature.

Table 1. Authentication mechanisms in the literature.

S. No	Scheme	Description	Limitations	Reference
1	Pseudonym based authentication	Make use of X.509 certificates and long key pairs.	High communication overheads and computational latency, difficulty managing CRL.	[22,23]
2	ID-based signature	RSU signs and authorize each of the messages.	If RSU is compromised, then entire system will be compromised.	[24,25]
3	Group signature	Privacy preservation by hiding the real identity of sender among the other group members.	If group administrator is compromised, then entire group will be compromised.	[26,27]
4	ID-based group signature	Generation of ID-based signature.	–	[28]
5	RSU-aided authentication	RSU performs authentication.	Does not work if RSU is not available in the vicinity or road side, and location of vehicle can be traced using RSU.	[29]
6	Cooperative authentication	Performs authentication by sending verification results to another vehicle cooperatively without the role of RSU.	Performance degrades in the case of low vehicle density and heavy revocation costs.	[25,28]
7	Ring signature	No central administrator in the group of ring vehicles.	Communication efficiency may be hampered	[30]

All the above schemes do not consider the communication delay of checking the overall performance of the system. This paper analyzes the explicit and implicit certificate authentication mechanisms recommended by the IEEE WAVE 1609.2 standard. Performance evaluations of these mechanisms were carried out in terms of total time lapse for authentication, which includes communication latency and computational latency.

3. Cybersecurity Considerations of IEEE WAVE

EVs communicate through the WAVE protocol, which is an expansion of 802.11 with some additions to physical and data link layers. In order to understand security vulnerabilities and the solutions implemented in this paper, it is important to understand this standard's protocol stack and how it relates to other standards, as shown in Figure 1. IEEE 1609.4 defines multichannel operation that specifies extensions to the IEEE 802.11 MAC layer protocol. IEEE 1609.3 defines networking services such as service advertisements, channel scheduling, and WAVE short message protocol (WSMP). IEEE 1609.2 defines security services for application and management of messages.

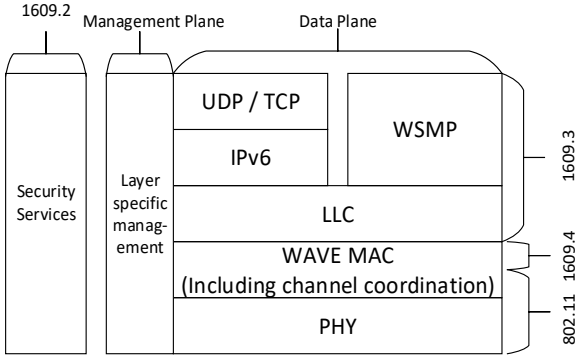

Figure 1. WAVE protocol stack and different standards.

Due to high mobility and low communication latency in vehicular networks, it is highly likely that an intruder might gain access to a RSU or an EV. This may have severe consequences, such as infrastructure damage, road accidents, and loss of lives. Consider a scenario where an EV with low battery charge tries to get the information of a nearby CS. The EV establishes communication with a nearby RSU to get information regarding nearby available CSs. The information may include number of charging slots available, distance to the CSs, etc. If the RSU is compromised or impersonated, it may misguide the EV. The problem of impersonation arises due to authentication failure of the devices.

IEEE 1609.2 defines the security services and mechanisms needed to protect VANETs. In IEEE 1609.2, the issue of authentication is addressed through public key infrastructure (PKI) and CA mechanisms. Furthermore, IEEE 1609.2 WAVE specifies the implicit or explicit certificate mechanisms for authentication of different entities in VANETs. However, IEEE 1609.2 does not specify a particular mechanism for authentication in VANET communication for a particular application.

Furthermore, the IEEE 1609.2 standard specifies some relevance conditions, based on which the validity of the received message can be verified. Relevance conditions are parameters, such as freshness, expiry, and reply, used to check for any security vulnerabilities. The freshness parameter ensures that the signature generation time is not too long for a service—that is, that the signature was generated recently. If freshness is compromised, then it may lead to reply attacks and masquerade attack using old credentials of the service. The expiry parameter ensures that received message is not expired. It can be checked either the certificate revocation list or by comparing the received time of message with the expiry time in the certificate field of message. The reply parameter ensures that the message received is not a duplicate one.

Charging management of EV applications requires minimum delays in communication between EVs and RSUs. Hence, it is required that the authentication mechanism for EV–RSU communication has

a low computational time. In this paper, performance evaluations of both implicit and explicit certificate authentication mechanisms are carried out in terms of overall time required for authentication.

4. Certificate Based Authentication Mechanisms for EV–RSU Communication

The architectural model of EV communication in smart cities incorporates wireless ad-hoc communication among EVs, CSs, and RSUs. Figure 2 is an illustrative example of V2I and V2G communication. Here, as EVs are going through the same geographical area, they can form a wireless ad-hoc network with RSUs and CSs based on a cell layout: that is, EVs, RSUs, and CSs within a one square kilometer radius.

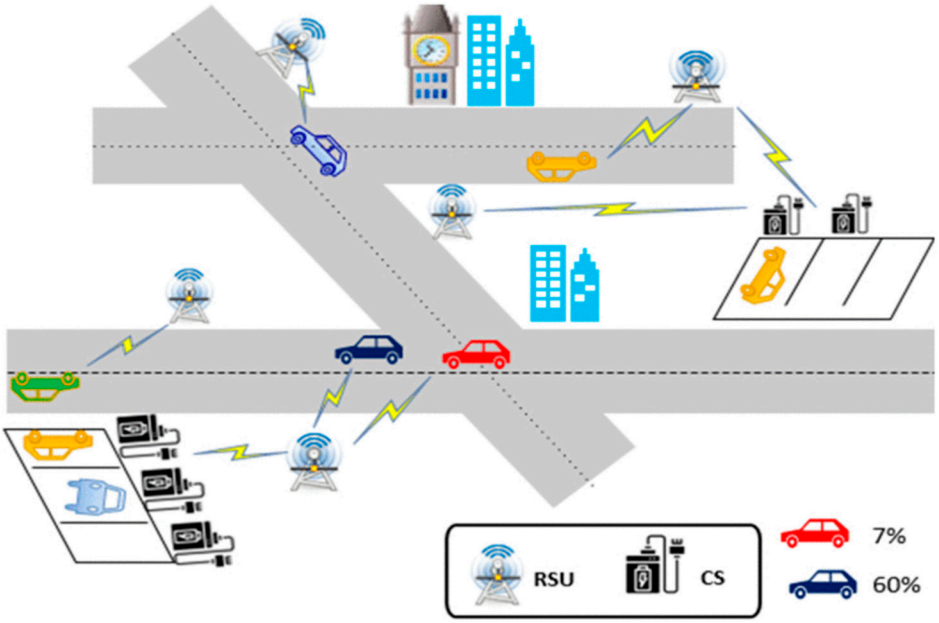

Figure 2. Conceptual architectural model for the vehicle-to-grid (V2G) and vehicle-to-infrastructure (V2I) systems.

CSs share information about the type of charging they offer and the number of available charging slots to the RSUs. On request from an EV, RSUs will run the charging slot allocation algorithm and send information about the allotted slot at a particular CS. Random EV load connections to CS will have a major impact on power distribution networks in the form of load boosting and power imbalances. This problem can be mitigated by employing effective EV charge management schemes. In Reference [7], the authors proposed an EV charging allocation algorithm which allocates a specific slot for charging to each EV. While allocating the charging slot, the algorithm proposed in Reference [7] takes into account the load at different CSs as well as other factors like power outages and shortages at the location of the CSs, number of empty charging slots, etc. Further, while allocating charging slots to EVs, the algorithm gives priority to the EVs with low battery charges, to avoid the problem of being left stranded due to a flat battery.

Implementation of the above discussed EV charging slot algorithm requires message exchanges between EV–RSU and EV–CS. Furthermore, these message exchanges must be standardized in order to achieve interoperability and plug-and-play operation. Hence, in Reference [7], V2I communication is based on the IEEE WAVE 1609 standard, while the V2G communication is based on the IEC 61850 and ISO/IEC 15118 standards.

For the purposes of this work, the security of the EV message exchanges is considered. Should there be an imposter acting as an RSU, it can receive all the sensitive information from EVs. Since no authentication scheme is implemented, EVs do not have the means to detect that the entity to which

they are sending their private information is not legitimate. This work addresses this knowledge gap by implementing authentication mechanisms in VANETs based on IEC 61850 and IEEE WAVE standards.

The first step to achieve this is by implementing a certificate-based mechanism to authenticate entities that want to take part in communication networks. As shown in Figure 3, a certificate authority (CA) is needed to issue individual certificates to each entity, denoted as step 1. There are different ways to implement certificate acquisition. In this paper, two such mechanisms, explicit and implicit certificate mechanisms, are considered and discussed in Sections 4.1 and 4.2, respectively. After implementation, they are tested to compare their performances and assess their feasibility for VANETs. Once received, these certificates are exchanged between entities prior to establishing a communication link. Each entity sends its own certificate to the other party, denoted as step 2, and verifies the certificate it received from the other party. If both certificates are valid, then both entities are authenticated as legitimate, i.e., authentication with certificate.

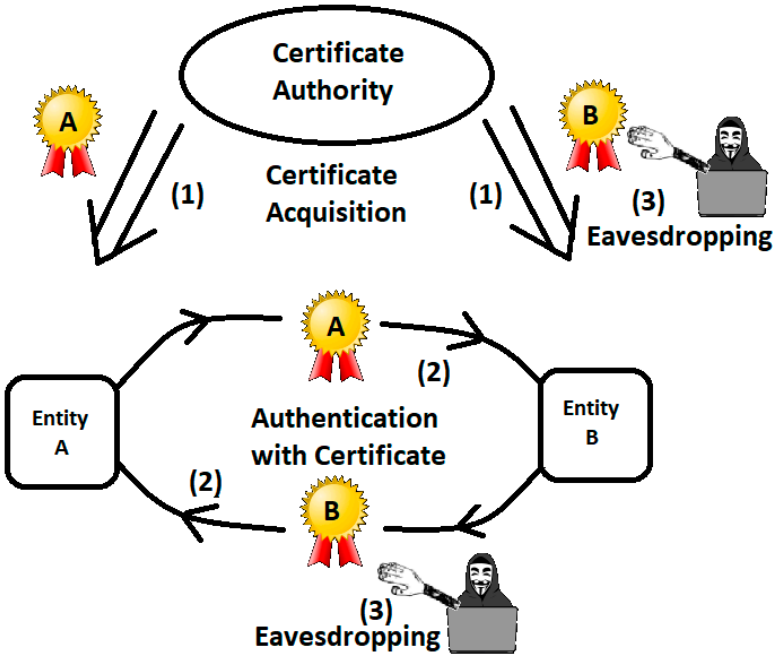

Figure 3. Certificate acquisition (**1**), authentication with certificates (**2**), and eavesdropping (**3**).

It is perfectly possible that a hacker eavesdrops, denoted as step 3, during steps 1 and/or 2. In this case, the hacker can snatch the certificate and use it as their own. In this fashion, they can authenticate themselves with other entities, as in step 2. In order to mitigate this, an additional security layer is required so that third parties cannot use other certificates, even if they could snatch them during communication. This is achieved by digital signatures and PKI. Since the certificates are signed by CA, the hacker will not be able to change the public key in the certificate. If the hacker changes the public key in the certificate, the certificate becomes invalid. Without the information of the private key associated with the public key in the certificate, the hacker cannot continue further communication. There are different signing algorithms that can be implemented for signing certificates. In this research, RSA and Elliptic Curve Digital Signature Algorithm (ECDSA) algorithms were implemented in explicit certificate mechanisms due to their strength. As discussed in Section 4, these different approaches were implemented, and their performances were studied for feasibility assessment.

In short, in the CA and PKI approach, certificates are used to identify legitimate nodes and keys are used to secure certificates, so that they are not stolen and used by unauthorized entities. The rest of the paper focuses on different implementation alternatives for CA and PKI, and investigates their feasibility for securing VANETs.

4.1. Explicit Certificate Mechanism

In this approach, the CA issues a certificate, which is a type of endorsement of a user with a unique public key. It contains a format specified by X.509 [31]. The process of issuing a certificate is binding a public key to a user's identity. The X.509 certificate format includes name, serial number, issuer signature value, and time of expiry. The authors in Reference [31] define terms such as certificate, end entity, CA, and revocation service. A certificate contains information that binds a public key to the identity of an end entity. This type of certificate is called an explicit certificate, where the public key of the subject is explicitly specified, hence the name. The explicit certificate contains additional information as specified in X.509. The CA also maintains a certificate revocation list (CRL), which is helpful in the certificate verification process.

Figure 4 explains the process of certificate signing by a CA. The EV constructs a certificate request with all the credentials required by the X.509 format and sends it to CA. The CA has its pair of public and private keys. The private key is only known to CA, whereas the public key is known to everyone. The CA signs the credentials of the certificate request, as shown in Figure 4, with a message digest algorithm (MDA), and generates a message digest (MD1). The MD1 is a hash value which is further encrypted with CA's private key to generate an encrypted digest (ED), also known as a signature. This process is called signing the certificate by CA. This paper uses the secure hash algorithm (SHA256) as the MDA and RSA for public key encryption. The signed certificates are loaded in tamper-free onboard units of EVs.

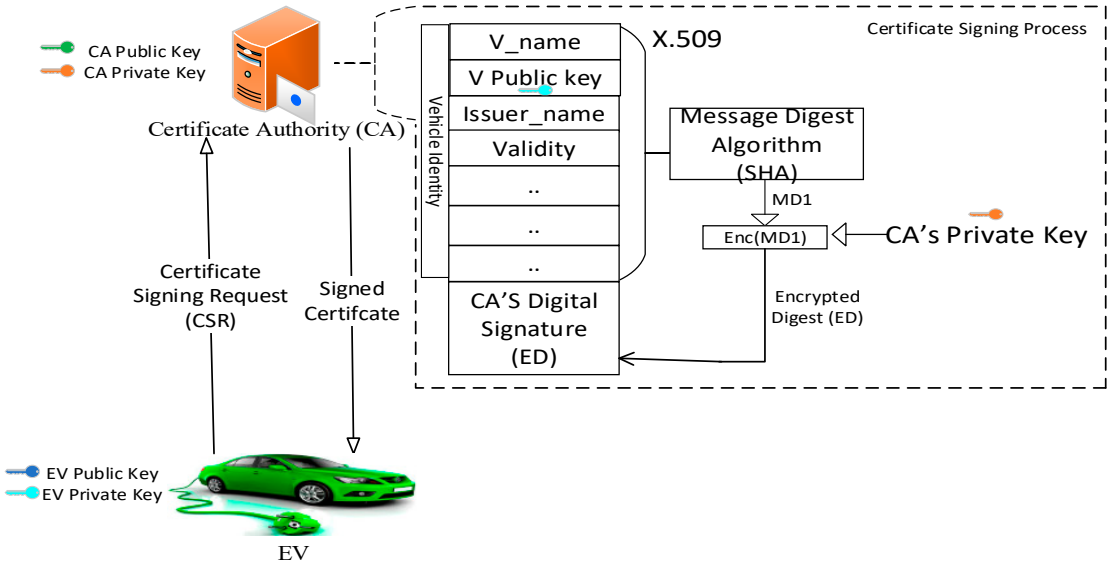

Figure 4. Certificate signing of electric vehicle (EV) by certificate authority (CA).

Algorithm 1 describes the certificate signing mechanism. The certificate request (X509Cert) received from the EV contains credentials such as the vehicle identity (ID), EV public key, etc. The CA first generates the hash value (h1) of the certificate request, using any hash algorithm such as SHA256. The generated hash value h1 is further encrypted with the public key cryptography algorithm (ENCCAPrKey(h1)) using the CA's private key (CAPrKey). The encrypted value (x) is stored in the signature field (X509Cert.Signature) of the certificate. Once the signing process is complete, the resulting format is a signed certificate (SX509Cert).

When an EV wants to communicate with a RSU, it first sends its signed certificate, wherein the certificate credentials are encrypted with the EV's private key, to the RSU for authentication. The signed certificate consists of encrypted credentials of the EV and a signature signed by the CA. The RSU picks up the signature from the certificate and performs decryption with the RSA public key algorithm. The public key of the CA is used in the decryption process. The output of the decryption is the message digest (MD1), which is generated by the CA. Following on, the RSU decrypts the certificate

credentials of the EV with the EVs public key and generates a new message digest (MD2) using the MDA. If the newly generated digest MD2 is the same as MD1, then RSU confirms the authenticity of the EV. The entire process is to verify whether the signature is signed by the CA or not. Figure 5 illustrates this verification process.

Algorithm 1. Signing (X509Cert)

1: $ID \leftarrow X509Cert.credentials$
2: $h1 \leftarrow H(ID)$
3: $x \leftarrow ENC_{CAPrKey}\ (h1)$
4: $X509Cert.Signature \leftarrow x$
5: *return SX509Cert*

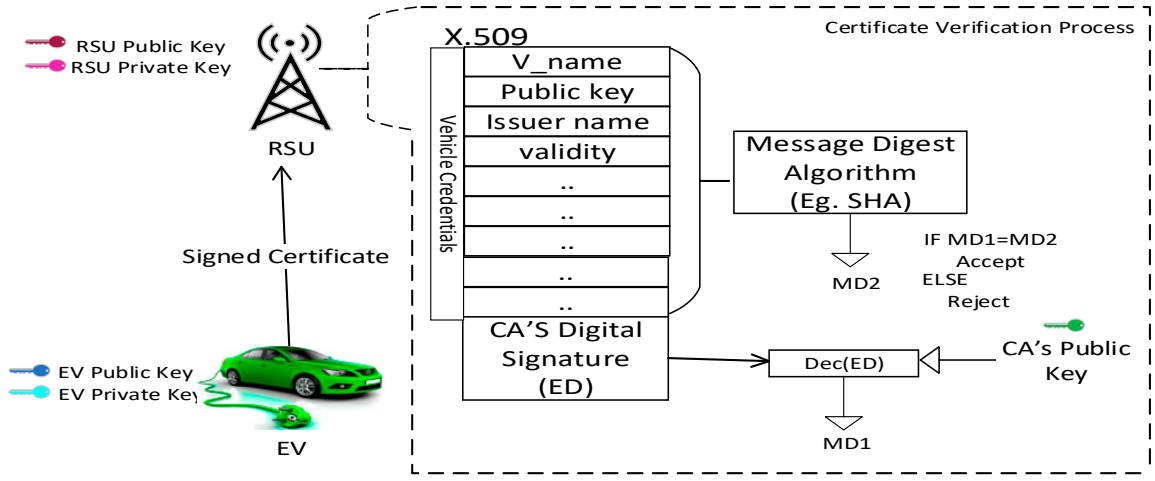

Figure 5. Certificate verification of EV by road side unit (RSU).

Algorithm 2 describes the certificate verification mechanism. Algorithm 2 essentially checks whether a certificate that is sent to a RSU is original or not. The signed certificate (SX509Cert) is received by a peer, such as a RSU. After receiving the signed certificate at the RSU, the certificate credentials part of the signed certificate is decrypted with the public key of the CA. The signature field of the signed certificate (SX509Cert) is first decrypted with a public key cryptographic algorithm (DEC(x)) using the public key of the CA (CAPubKey). Thereafter, the RSU hashes the credentials part of the certificate to generate digest h2. The decrypted signature value, h1, is compared with the newly generated hash value, h2. The original hash value (h1) cannot be changed as it can only be created and encrypted by the CA. If the original hash value is tampered with, it already shows the certificate is not authentic. If both hash values are identical, then authentication is successful, otherwise, authentication fails. Once the process of authentication is complete, data will be exchanged between the EV and the RSU.

Algorithm 2. Verify(SX509Cert)

1: $x \leftarrow SX509Cert.Signature$
2: $h1 \leftarrow DEC_{CAPubKey}\ (x)$
3: $ID \leftarrow SX509Cert.credentials$
4: $h2 \leftarrow H(ID)$
5: if $h1 = h2$ then
6: return True
7: else
8: return False
9: end if

Sometimes, a CA can be a subordinate of a parent CA. The subordinate CA's certificate is signed by the parent CA, while the parent CA's certificate is self-signed. Figure 6 depicts a scenario where there is more than one CA. EV1 has a certificate signed by Subordinate CA (Y1). Subordinate CA (Y1)'s identity is recognized by X1, which is in turn recognized by the Root CA. The Root CA is also called a trust anchor. The trust anchor has a self-signed certificate.

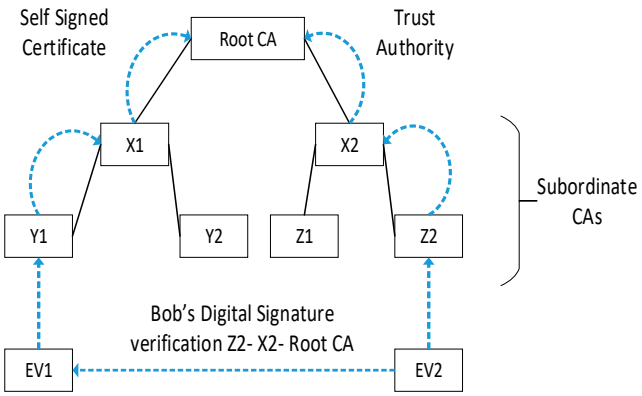

Figure 6. Digital certificate hierarchy for verification.

The authentication process through the explicit certificate mechanism between EV, RSU, and CS is shown in Figure 7. Initially EV, RSU, and CS receives their signed certificates from the CA. Next, the EV communicates with a nearby RSU to get charging station information. The RSU first verifies the received certificate through the CA. If the certificate is valid and not revoked, then communication is initiated. The charging station information is transmitted between the EV and the RSU through WSMP (wireless short message protocol) messages, as specified in Reference [7].

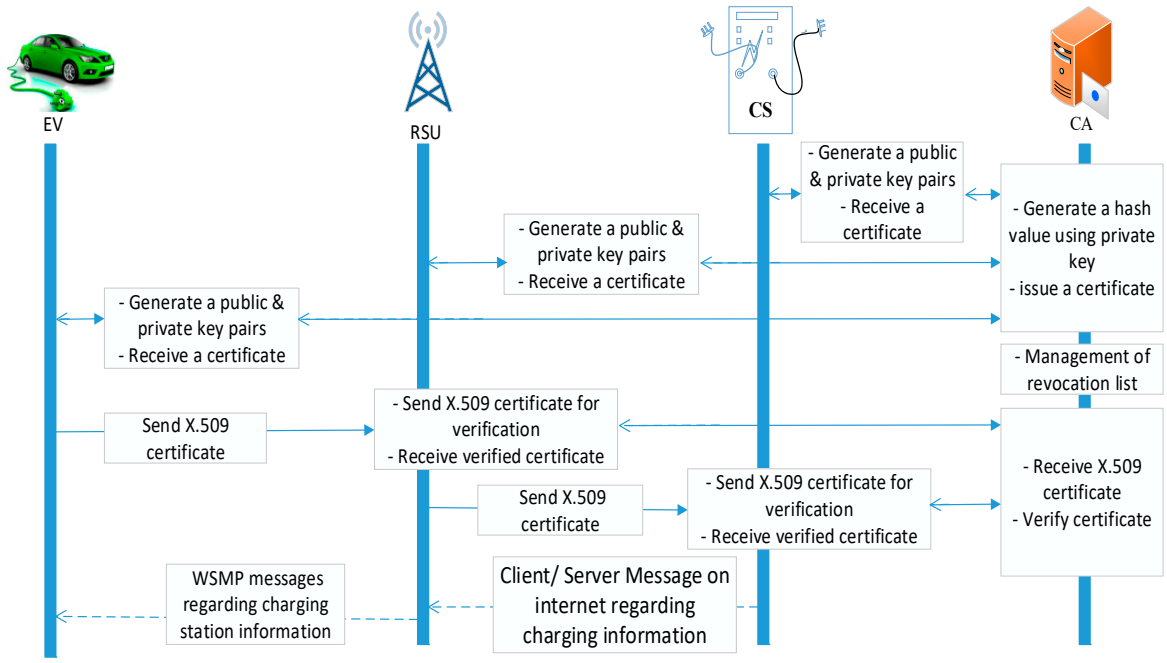

Figure 7. Message exchanges for explicit certificate-based authentication process.

Another significant functionality of the CA is the revocation service. An intruder EV can send a revoked certificate and try to establish communication with other entities on VANET. The revocation service keeps a list of revoked certificates inside CA. Whenever an EV sends its certificate to a RSU, the RSU verifies the certificate credentials with the CA. The CA checks this certificate against the

revocation list, and notifies the RSU of the result. This feature is very important in mitigating security attacks. Any anomalies found in the certificate are reported by the CA to the RSU, hence, the intruder EV can be identified. IEEE WAVE specifies the mechanism of revocation of certificates by maintaining parameters of relevance conditions such as freshness, expiry, and reply, as discussed in Section 3.

4.2. Implicit Certificate Mechanism

Another kind of certificate is the implicit certificate [32], which does not contain a user's public key inside. The implicit certificate mechanism is a concept, and Elliptic Curve Qu-Vanstone (ECQV) is a mathematical incarnation of this concept [33]. An ECQV certificate contains a user's identity and public key reconstruction data, along with the CA's public key. When an RSU receives a certificate from an EV for establishment of communication, the RSU constructs the transmitter's public key from the user identity, the CA's public key, and the public parameter. The user's public key is not explicitly contained in the implicit certificate. It is computationally infeasible for an intruder to compute a private key corresponding to an EV's public key, or to construct a matching EV identity and reconstruct data for which a corresponding private key may also be computed [34].

Figure 8 explains the steps involved in implicit certificate generation process. Elliptic curve domain parameters are defined as per Reference [35]. Let q be the order of finite field Fq, E is elliptic curve defined over Fq, and G is a generator point in E(Fq). The prime number n denotes the order of G. The CA picks its private key $t \in [1, n-1]$ and computes the public key C = c*G. In the same way, the EV picks its private key $v \in [1, n-1]$ and computes the public key V = v*G. ID_{EV} is information related to the EV, which consists of the EVs identifier and the CA identifier, serial number, and validity period included in EV's certificate, and P denotes the EV's reconstruction public data. H is a hash function, such as SHA256.

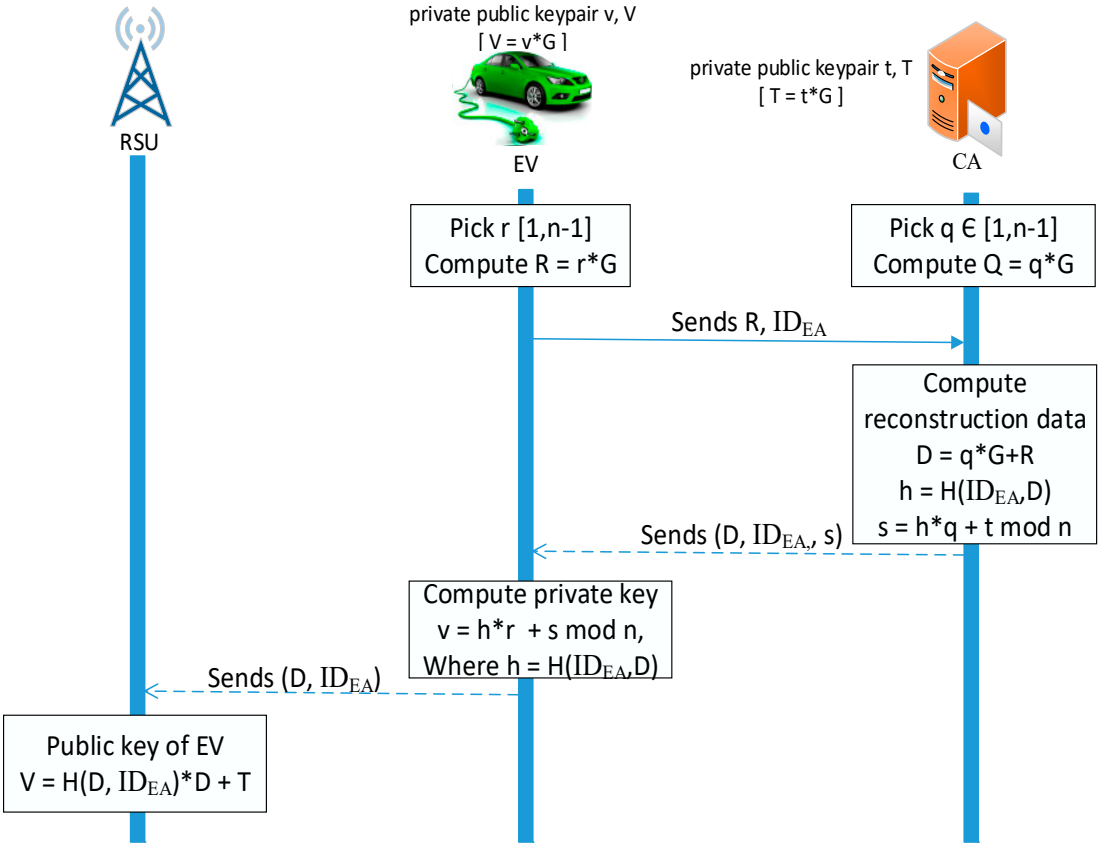

Figure 8. Implicit certificate mechanism using elliptic curve cryptography (ECC).

IEEE1609.2 recommends the use of the Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Integrated Encryption Standard (ECIES) for generating signature and encryption operations, respectively. The standard also specifies the use of two NIST-approved elliptic curves: P-224 and P-256. When a data packet is received, say, from an EV to a nearby RSU, the authentication process using the implicit certificate mechanism takes place as follows.

Initially, the EV and the CA compute their private and public key pairs ((v, V), (t, T)). The EV then picks a random number, r, between 1 and n, computes $R = r*G$ and sends it to the CA along with its identification data, ID_{EA}. The CA also picks a random number, q, between 1 and n, and computes the EV's reconstruction public data D = q*G + D. The CA computes $h = H$ (ID_{EA}, D) and s = h*q + t mod n. Finally, the CA sends (D, ID_{EA}, s) to the EV. The EV sets its private key to $v = h*r$ + s mod n, where h = H (ID_{EA}, D). The EV's public key, V [V = v* G], can be reconstructed by the RSU from the the implicit certificate (ID_{EA}, D) using the equation:

$$V = H\ (ID_{EA},\ D) * D + T \tag{1}$$

where $T = t*G$, which is the public key of the CA. The equation holds, since:

$$\begin{aligned} V &= v * G\ [\text{public key of EV}] \\ &= (h * r + s) * G \\ &= (h * r + h * q + t) * G \\ &= h(r + q) * G + t * G \\ &= h(r * G + k * G) + T \\ &= h * D + C \\ &= Hash\ (ID_{EA},\ D) * D + C \end{aligned}$$

5. Performance Evaluation

In order to assess the suitability of these different mechanisms for VANETs, the total time lapse for authentication was calculated. The total authentication time is the combined time required for transmission of certificate from EV to RSU, and computational time to verify the certificate at the RSU. A combinational approach, depicted in Figure 9, was used. As shown, authentication mechanisms were implemented using Python OpenSSL library. Through these implementations two parameters were acquired. The first is the certificate size for each individual implementation, and the second is the computational time required to verify a certificate. These parameters are given in Tables 2 and 3, third and fourth columns from the right.

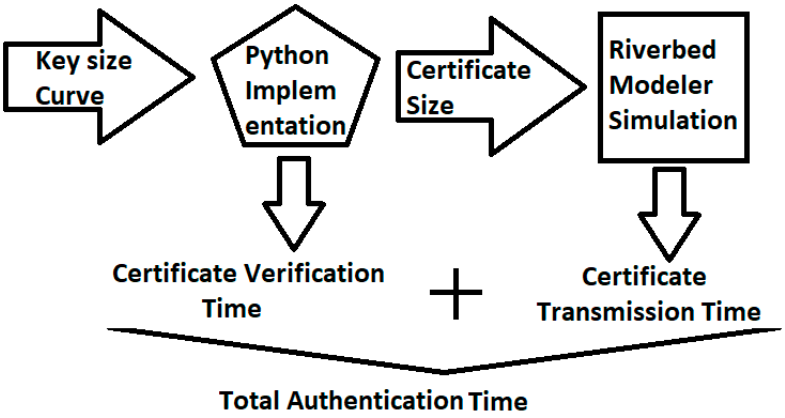

Figure 9. Timing performance with OpenSSL implementation and Riverbed Simulations.

Table 2. Computational times for explicit certificate verification with different key sizes of RSA and ECDSA.

Type of Signature		Private Key (Bytes)	CSR Size (Bytes)	Certificate Size (Bytes)	Certificate Verification Computational Time (ms)	Certificate Transmission Time (ms)	Total Time for Authentication (ms)
Algorithm	**Key Size/Curve**						
RSA	1024	891	660	847	7	6.1	13.1
	2048	1675	1013	1200	9	8.4	17.4
	3072	2455	1358	1545	9	9.0	18
	7680	5977	2918	3105	10	13.4	23.4
	15,360	11,823	5518	5709	12	16.9	28.9
ECDSA	secp224r1	278	530	700	7	5.1	12.1
	secp521r1	436	737	904	9	6.4	15.4
	prime192v1	270	505	668	8	5.0	13
	prime256v1	302	542	696	7	5.0	13
	brainpoolP384r1	367	627	778	8	5.2	13.2
	brainpoolP512r1	436	725	875	9	6.2	15.2
	brainpoolP384t1	367	639	794	8	6.0	14
	brainpoolP512t1	436	729	887	9	6.2	15.2

Table 3. Computational times for ECQV implicit certificate verification for authentication.

Type of Signature		Private Key (Bytes)	Certificate Size (Bytes)	Certificate Verification Computational Time (ms)	Certificate Transmission Time (ms)	Total Time for Authentication (ms)
Algorithm	**Curve**					
ECQV	secp224r1	28	343	3	4	7
	secp521r1	65	574	6	5.3	11.3
	secp256k1	32	386	3.5	4.4	7.9
	prime192v1	24	327	3	4	7
	prime256v1	32	371	3.5	4.4	7.9
	brainpoolP384r1	48	469	4.5	5.0	9.5
	brainpoolP512r1	64	564	6	5.2	11.2
	brainpoolP384t1	48	469	4.5	5.0	9.5
	brainpoolP512t1	64	562	6	5.3	11.3

To study the impact of the communication network on the time lapse, Riverbed Modeler simulations were performed to calculate the time delays for transmission of certificates from EV to RSU. This heavily depends on the size of the packet that is transmitted. Therefore, certificate sizes obtained from OpenSSL implementations were inputted to Riverbed Simulations. All other parameters, shown in Table 4, were kept the same to compare and contrast different implementations. It was assumed that 50 EVs and 5 RSUs were spread over an area of 1 km $\times$ 1 km. The 'manet_station' mobile and fixed nodes, available in Riverbed Modeler's library, were selected to model the EVs and RSUs, respectively. The movement of EVs was configured in the 'Mobility_Config' node by creating different mobility profiles to define trajectories for the EVs. The trajectories were defined by specifying the speed, destination (coordinates), start time, and stop time in the mobility profiles, as specified in Table 4. The simulation was carried out by setting traffic scenarios for exchanging the different certificate sizes, as specified in Tables 2 and 3.

Table 4. Simulation parameters.

Parameters	Values
Wireless Protocol	WiFi 802.11p
Base Frequency	5.9 GHz
Data Rate	24 Mbps
Channel Bandwidth	10 MHz
Modulation Type	OFDM
Maximum Transmission Power	0.5 W
FFT Period	6.4 µs
No. Sub Carriers	64
Cyclic Prefix Duration	1.6 µs
Size	1 km $\times$ 1 km
Vehicle Density	50 EVs and 5 RSUs
Speed of EV	12 m/s

First, using Python-based implementations, a pair of public and private keys for the EV and CA were generated. Next, a certificate with X.509 format was generated with the required credentials and sent to the CA for signing. Using the Secure Hash Algorithm (SHA256) with 256 bit key, a hash value was generated. The hash value was further encrypted using RSA and ECDSA public key cryptographic algorithms. Here, RSA algorithms with different key sizes and various elliptic cures defined by NIST for ECDSA were considered. Once the signing process was completed, the signed certificate was sent to the EV. The EV then sent this certificate to the RSU for authentication. The communication overheads of keys, CSRs, and certificates for different algorithms are given in Table 2.

(1) EV -> CA: CSR (X509Cert) Where CSR size is 660 bytes for RSA digital signature algorithm with 1024 bit public key. Similarly, CSR size for different RSA (different key sizes) and ECDSA curves are listed in Table 2.
(2) CA -> EV: SX509Cert The size of certificate issued by the CA (SX509Cert) was 849 bytes for the RSA digital signature algorithm with 1024 bit public key. Similarly, certificate sizes for different RSA (different key sizes) and ECDSA curves are listed in Table 2.

The computational time for carrying out verification processes for different algorithms, such as RSA and ECDSA, was computed though Python-based authentication programs, and is listed in Table 2. It was assumed that an EV has its certificate signed by CA and is ready to establish communication with a RSU. From the results, it was observed that the ECDSA curves with larger key sizes had slightly more computational time compared to curves with smaller key sizes. Elliptic curves had smaller key and certificate sizes compared to RSA, hence required less computational time for verification process.

The time delays for transporting the different certificates from EV to RSU are given in Table 2. From Table 2, it is evident that the RSA algorithm-based authentication had larger certificate sizes,

and thus, larger verification computational time and transmission time. Therefore, it can be concluded that ECDSA algorithm-based authentication results in lower certificate sizes and reduces overall time required for authentication in an EV communication environment.

Similarly, as discussed above for the explicit certificate mechanism, the performance evaluation of implicit mechanism was carried out with Python and Riverbed Modeler implementations. ECQV is an implementation of implicit certificate mechanism, as discussed in Section 4.2. ECQV has reduced key size as compared to the X.509 certificate mechanism developed by RSA. For different elliptic curves of ECQV, implicit certificates were generated, and the computational time required for verification of certificate was calculated through Python implementations. Next, the certificate transmission time for different ECQV curves was obtained through the Riverbed Modeler simulations, as discussed for the explicit mechanisms. It is clearly evident from Tables 2 and 3 that curves with less key size generate smaller certificates and, consequently, result in smaller certificate transmission and computational times. Furthermore, it was noted that the ECQV curve 'secp256k1' gave the best time performance and provided optimum security. Since 'secp256k1' gave the best time performance and provided the same level of security as other curves, it turned out to be the most optimum solution for implementing certificate-based authentication.

It is observed from the obtained results that implicit certificate mechanisms for authentication are well suited to EV–RSU communications for functions such as charging management. Explicit certificate mechanisms with RSA and ECDSA algorithms need longer times for overall communication. They can still be used for many other network applications where vehicles travel slower, such as electronic toll collection points. For EV–CS communication, time performance is not crucial and either of the mechanisms can be implemented.

6. Conclusions

EVs are increasingly becoming popular due to environmental concerns. Their interaction with the power grid creates opportunities for better grid operation and more renewable energy use. Achieving this is dependent on providing a robust communication network between EVs and the infrastructure surrounding them. Smart city concepts make use of the increased connectivity and observability provided by V2V and V2I communication.

Security is the major concern in these VANETs. IEEE 1609.2 WAVE specifies different security mechanisms for secure data transmission. This paper analyses the use of different certificate-based authentication mechanisms in VANETs, and studies their performances in realistic network traffic conditions. Performance calculations include the certificate transmission time and certificate verification time, leading to computation of total authentication time. In computing total authentication time, communication overheads of keys, CSRs, Certificates for different algorithms were considered. From the results it was observed that in both implicit and explicit certificate mechanisms, the curves with larger key sizes had slightly more computational time compared to curves with smaller key sizes. While explicit certificate mechanisms showed slow performance, implicit certificate-based authentication mechanism were found to be well-suited for EV–RSU and EV–EV communication.

Author Contributions: Conceptualization and methodology, S.M.S.H. and T.S.U.; software and validation, S.M.F.; writing—original draft preparation, S.M.F. and S.M.S.H.; writing—review and editing, S.M.S.H and T.S.U.; supervision, T.S.U. and S.K.; funding acquisition, T.S.U.

Funding: This work was supported by Research and Innovation Fund 2018 and by KEIDANREN (Japan Business Federation) Promotion of Environmental Protection Foundation's Research Grant-2018.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

x	Encrypted text
CA	Certificate Authority
EV	Electric Vehicle
$ENC_k(M)$	Encryption function for the input message M using key k
$DEC_k(M)$	Decryption function for the input message M using key k
$H(M)$	Hash function for the input message M
h	Hash value
X509Cert	X.509 format certificate
SX509Cert	Signed certificate
CAPubKey	CA Public key
CAPrKey	CA Private Key
ID	Vehicle identity

References

1. Newman-Askins, R.; Ferreira, L.; Bunker, J. Intelligent transport systems evaluation: From theory to practice. In Proceedings of the 21st ARRB and 11th REAAA Conference, Cairns, Australia, 18–13 May 2003.
2. IEC. *International Standard. Part 41: Standard Transfer Specification (STS)—Application Layer Protocol for one-way Token Carrier Systems Electricity Metering—PAYMENT Systems—IEC 62055-41*, 2nd ed.; International Electrotechnical Commission: Geneva, Switzerland, 2014.
3. US Department of Transportation, Intelligent Transportation Systems [Internet]. Available online: http://www.its.dot.gov (accessed on 14 January 2019).
4. Begwani, A.K.; Ustun, T.S. Electric bus migration in Bengaluru with dynamic charging technologies. *AIMS Energy* **2017**, *5*, 944–959. [CrossRef]
5. Ustun, T.S.; Ozansoy, C.R.; Zayegh, A. Implementing vehicle-to-grid (V2G) technology with IEC 61850-7-420. *IEEE Trans. Smart Grid* **2013**, *4*, 1180–1187. [CrossRef]
6. Nsonga, P.; Hussain, S.M.S.; Garba, A.; Ustun, T.S.; Ali, I. Performance Evaluation of Electric Vehicle Ad-Hoc Network Technologies for charging management. In Proceedings of the 9th IEEE PES Asia-Pacific Power and Energy Engineering Conference, Bangalore, India, 8–10 November 2017.
7. Hussain, S.M.S.; Ustun, T.S.; Nsonga, P.; Ali, I. IEEE 1609 WAVE and IEC 61850 Standard Communication Based Integrated EV Charging Management in Smart Grids. *IEEE Trans. Veh. Technol.* **2018**, *67*, 7690–7697. [CrossRef]
8. *IEEE Guide for Wireless Access in Vehicular Environments (WAVE)—Architecture*; IEEE Std 1609.0-2013; IEEE: New York, NY, USA, 5 March 2014; pp. 1–78.
9. *Communication Networks and Systems for Power Utility Automation—Part 90-8: Object Model for E-Mobility*; IEC 61850-90-8, Ed.1.0; International Electrotechnical Commission: Geneva, Switzerland, 2016.
10. Mejri, M.N.; Ben-Othman, J.; Hamdi, M. Survey on VANET security challenges and possible cryptographic solutions. *Veh. Commun.* **2014**, *1*, 53–66. [CrossRef]
11. *IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages*; IEEE Standard 1609.2; IEEE: New York, NY, USA, 2006.
12. Bernardini, C.; Asghar, M.R.; Crispo, B. Security and privacy in vehicular communications: Challenges and opportunities. *Veh. Commun.* **2017**, *10*, 13–28. [CrossRef]
13. Manvi, S.S.; Tangade, S. A survey on authentication schemes in VANETs for secured communication. *Veh. Commun.* **2017**, *9*, 19–30. [CrossRef]
14. Sánchez-García, J.; García-Campos, J.M.; Reina, D.G.; Toral, S.L.; Barrero, F. On-siteDriverID: A secure authentication scheme based on Spanish eID cards for vehicular ad hoc networks. *Future Gener. Comput. Syst.* **2016**, *64*, 50–60. [CrossRef]
15. Chen, C.L.; Chen, Y.X.; Lee, C.F.; Deng, Y.Y. A Survey of Authentication Protocols in VANET. In *Advances on Broadband and Wireless Computing, Communication and Applications, Lecture Notes on Data Engineering and Communications Technologies*; Springer: Cham, Switzerland, 2019; Volume 25.
16. Mohit, P.; Amin, R.; Biswas, G.P. Design of authentication protocol for wireless sensor network-based smart vehicular system. *Veh. Commun.* **2017**, *9*, 64–71. [CrossRef]

17. Li, H.; Dán, G.; Nahrstedt, K. Portunes+: Privacy-Preserving Fast Authentication for Dynamic Electric Vehicle Charging. *IEEE Trans. Smart Grid* **2017**, *8*, 2305–2313. [CrossRef]
18. Saxena, N.; Choi, B.J. Authentication Scheme for Flexible Charging and Discharging of Mobile Vehicles in the V2G Networks. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1438–1452. [CrossRef]
19. Abdallah, A.; Shen, X.S. Lightweight Authentication and Privacy-Preserving Scheme for V2G Connections. *IEEE Trans. Veh. Technol.* **2017**, *66*, 2615–2629. [CrossRef]
20. Python Socket Library. Available online: https://docs.python.org/2/howto/sockets.html (accessed on 14 January 2019).
21. Riverbed Modeler—(Formerly OPNET). Available online: http://goo.gl/72SgAM (accessed on 14 January 2019).
22. Tsai, J.L.; Lo, N.W.; Wu, T.C. Novel anonymous authentication scheme using smart cards. *IEEE Trans. Ind. Inform.* **2013**, *9*, 2004–2013. [CrossRef]
23. Choi, J.; Jung, S. A security framework with strong non-repudiation and privacy in VANETs. In Proceedings of the 2009 6th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 10–13 January 2009; pp. 1–5.
24. Zhang, C.; Lu, R.; Lin, X.; Ho, P.-H.; Shen, X. An efficient identity-based batch verification scheme for vehicular sensor networks. In Proceedings of the 27th IEEE Conference Computer Communication (INFOCOM), Phoenix, AZ, USA, 13–18 April 2008; pp. 1–9.
25. Shao, J.; Lin, X.; Lu, R.; Zuo, C. A threshold anonymous authentication protocol for VANETs. *IEEE Trans. Veh. Technol.* **2016**, *65*, 1711–1720. [CrossRef]
26. Chim, T.; Yiu, S.; Hui, L.; Li, V. SPECS: Secure and privacy enhancing communications schemes for VANETs. *Ad Hoc Netw.* **2011**, *9*, 189–203. [CrossRef]
27. Horng, S.-J.; Tzeng, S.-F.; Pan, Y.; Fan, P.; Wang, X.; Li, T. b-SPECS+: Batch Verification for Secure Pseudonymous Authentication in VANET. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1860–1875. [CrossRef]
28. Tiwari, D.; Bhushan, M.; Yadav, A.; Jain, S. A novel secure authentication scheme for VANETs. In Proceedings of the 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 12–13 February 2016; pp. 287–297.
29. Lo, N.-W.; Tsai, J.L. An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without bilinear pairings. *IEEE Trans. Intell. Trans. Syst.* **2015**, *17*, 1319–1328. [CrossRef]
30. Chaurasia, B.K.; Verma, S. Conditional privacy through ring signature in vehicular ad-hoc networks. In *Transactions on Computational Science XIII*; Springer: Berlin, Germany, 2011; pp. 147–156.
31. Cooper, D.; Santesson, S.; Farrell, S.; Boeyen, S.; Housley, R.; Polk, W. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC 5280. May 2008. Available online: https://tools.ietf.org/html/rfc5280 (accessed on 14 January 2019).
32. Brown, D.R.L.; Gallant, R.P.; Vanstone, S.A. Provably secure implicit certificate schemes. In Proceedings of the 5th International Conference on Financial Cryptography, Southampton, Bermuda, 11–14 March 2002; pp. 156–165.
33. SEC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV). Standard for Efficient Cryptography, January 2013. Available online: http://www.secg.org/sec4-1.0.pdf (accessed on 14 January 2019).
34. Menezes, A.; van Oorschot, P.; Vanstone, S. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
35. SEC 1: Elliptic Curve Cryptography. Standard for Efficient Cryptography, May 2009. Available online: http://www.secg.org/sec1-v2.pdf (accessed on 14 January 2019).