sensors

MDPI

Article

An Enhanced Secure Identity-Based Certificateless Public Key Authentication Scheme for Vehicular Sensor Networks

Congcong Li [1,*], Xi Zhang [1], Haiping Wang [2] and Dongfeng Li [3]

1 School of Traffic and Transportation, Beijing Jiaotong University, Haidian District, Beijing 100000, China; xizhang@bjtu.edu.cn
2 Research and Development Department, Beijing Zhonghaiwenda Information Technology Company, Haidian District, Beijing 100000, China; whp@mail.ustc.edu.cn
3 Electronic Transaction Cryptographic Application Group, State Cryptography Administration Office of Security Commercial Code Administration, Fengtai District, Beijing 100000, China; lidongfeng66@163.com
* Correspondence: 14114206@bjtu.edu.cn; Tel.: +86-186-1403-2212

Received: 7 December 2017; Accepted: 8 January 2018; Published: 11 January 2018

Abstract: Vehicular sensor networks have been widely applied in intelligent traffic systems in recent years. Because of the specificity of vehicular sensor networks, they require an enhanced, secure and efficient authentication scheme. Existing authentication protocols are vulnerable to some problems, such as a high computational overhead with certificate distribution and revocation, strong reliance on tamper-proof devices, limited scalability when building many secure channels, and an inability to detect hardware tampering attacks. In this paper, an improved authentication scheme using certificateless public key cryptography is proposed to address these problems. A security analysis of our scheme shows that our protocol provides an enhanced secure anonymous authentication, which is resilient against major security threats. Furthermore, the proposed scheme reduces the incidence of node compromise and replication attacks. The scheme also provides a malicious-node detection and warning mechanism, which can quickly identify compromised static nodes and immediately alert the administrative department. With performance evaluations, the scheme can obtain better trade-offs between security and efficiency than the well-known available schemes.

Keywords: authentication; identity-based; certificateless; vehicular sensor network (VSN)

1. Introduction

According to a report by the World Health Organization (WHO), the total number of worldwide road traffic deaths caused by various traffic accidents is 1.25 million per year [1]. To manage increasingly heavy traffic scenarios and enhance driving safety, wireless sensor networks and smart devices have recently been implemented on a large scale in the transportation systems of many countries. As part of an intelligent transportation system (ITS), vehicle sensor networks (VSNs) provide a better resolution to traffic problems via the collection, processing and dissemination of traffic information within the scope of interconnected sensor nodes, which are mounted on vehicles and roadsides. The static wireless access nodes alongside the roads, which are called Road Side Units (RSUs), are used to provide communication to vehicles and infrastructure in their coverage area. VSNs involve different network modules, such as Wireless Access in Vehicular Environment (WAVE) [2]/Dedicated Short-Range Communication (DSRC), Wireless Fidelity (Wi-Fi) and the 4th Generation Communication System (4G)/Long Term Evolution (LTE) that work together. Among them, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are two main forms of VSNs that use the DSRC protocol [3] and WAVE to perform their operations in collaboration.

VSNs are rapidly changing and self-organizing with multiple-hops topologies over wireless links. Various wireless communication devices on vehicles broadcast traffic information to RSUs or other vehicles every 100–300 milliseconds according to the DSRC. Thus, it must take a short amount of time to deal with a message without delay for VSN entities.

The information among VSN entities include traffic conditions (e.g., road defects, congestion situations and temperature conditions, etc.) and vehicle conditions (e.g., location, speed, traffic status, etc.) [2]. These messages are indispensable for vehicles and infrastructure, such as traffic control centers, which use these messages to make critical decisions in an emergency situation. If an adversary modifies messages or inserts malicious messages to the network, it will result in traffic chaos or even accidents. Furthermore, DSRC/WAVE is inferior to other network modules in terms of security support [4]. DSRC is a wireless protocol that makes data to be easily monitored, altered and forged, including sensitive data concerning drivers' privacy [5]. Therefore, to protect users' privacy and information integrity in VSNs is important. In addition, RSUs are always deployed in an unattended environment. Hardware tampering occurs when the sensors and other on-board hardware RSUs are manipulated by adversaries [6]. Adversaries may capture and take control the RSUs via a physical attack and extract all cryptographic information from the compromised RSUs, and they may relocate a tampered RSU to launch a malicious attack [7] or make many clones from the tampered RSU. Therefore, resisting RSU compromise and replication attacks is a key consideration in the designed authentication. However, many existing secure schemes fail to withstand RSU compromise attacks.

This paper presents an enhanced identity-based (ID-based) certificateless authentication scheme to solve the aforementioned problems. The main contributions provided are as follows:

1. The proposed scheme is based on the certificateless public key cryptograph (CLPKC) [8], which can solve the certificate management problem in the public key infrastructure (PKI) [9] and the key escrow' problem in identity-based encryption (IBE) [10,11]. The scheme use the elliptic curve multiplication instead of the bilinear pairing because that the relative computational costs of a pairing operation are approximately 20 times higher than that of an elliptic curve scalar multiplication [12]. In addition, this scheme supports batch authentication by simultaneously verifying several messages. Moreover, the proposed scheme is provably secure against the adaptive chosen message attack in the random oracle model as long as the computational elliptic curve discrete logarithm problem (ECDLP) is intractable.
2. In the scheme, an anonymous communication and conditional privacy-preserving authentication are supported to protect users' privacy. Every user is issued a smart card with distinct pseudo identities, which are generated by trusted authorities (TAs) according to user's actual identity and secret information. The user's actual identity can be uniquely revealed by the TA when necessary.
3. The proposed scheme uses a position-based authentication scheme to reduce the possibility of RSU capture attacks. The proposed scheme also provides a compromised-RSU detection and alarm mechanism to identify misbehaving RSUs and immediately alert the traffic administrative department.

2. Related Work

In this section, we provide a brief summary of the related literature focused on authentication schemes in VSNs. Many authentication schemes have been proposed in recent years, and most of them are certificate-based or ID-based authentication schemes. Paruchuri et al. [13] proposed a certificate-based scheme, which provides anonymous authentication and location privacy using a smart card that stores the session keys of RSUs. However, this scheme fails to support V-to-V authentication. The RSUs and vehicles require additional computations to verify the certificates issued by the TA. In addition, each on-board unit (OBU) stores many session keys from different RSUs. And during the authentication process, the encrypted message is transmitted to identify the owner of the session key to be decrypted, which is inefficient for VSN authentication. Finally, if one RSU

is compromised, then the stored session keys in the RSU, including the session keys of neighboring RSUs, are leaked.

Almeida et al. [14] proposed a PKI-related key distribution protocol for VSNs that alleviates the burden of traditional PKI authentication schemes. However, many different keys are stored in each vehicle, and when a node is compromised, it will trigger a key revocation in a distributed fashion, which may cause an undesirable communication overhead. In addition, the PKI-based authentication mechanisms require additional computational overhead to verify the certificates of others.

To improve the scalability of certificate-based authentication schemes for VSNs, Calandriello [15] proposed a pseudonym-based authentication scheme to achieve efficiency and robustness. This scheme authorizes each OBU to generate its own pseudonyms without affecting the system security. However, each mobile node (vehicle) preloads many pseudonyms and related certificates in the story, which uses a considerable amount of memory. During a time period of τ, the scheme can also suffer from a tracking attack if the signature $Cert^{H}_{CA}(K^{i}_{v})$ is unchanged. Moreover, the scheme does not address certificate revocation.

Zhang et al. [16] proposed an RSU-aided message authentication scheme in which a vehicle obtains a symmetric key from a RSU and communicates with other vehicles using a keyed hash message authentication code (HMAC). However, the scheme is fully relies on RSUs. If one RSU is controlled or compromised, the scheme will collapse.

Because of the certificate management problem, an ID-based scheme is a more precise replacement for the PKI-based scheme for vehicular-network applications [17]. Authentication schemes that use IBE, which was proposed by Shamir [10] in 1984 have been implemented in VSNs. Chim [18] proposed an ID-based authentication scheme with batch verification based on the above bilinear pairings for secure V-to-I communications. This scheme has lower communication costs than previously proposed ID-based schemes. However, Horng et al. [19] found that Chim's scheme was vulnerable to impersonation attacks, in which a malicious vehicle can impersonate a valid vehicle and send fake messages to the RSUs or other vehicles. Horng et al. provided a secure scheme that overcame the weaknesses of the scheme in [18]. However, because the computational costs of one pairing operations are at least three times higher than that of a one point multiplication operation [20], these two schemes require heavy computational costs in the signature verification phase and are not suitable in rapidly changing networks. Furthermore, these mechanisms are only considered suitable for private networks [21] because of the key escrow problem based on IBE.

In 2003, Al-Riyami and Paterson [8] developed the concept of CLPKC. In this scheme, the full private key consists of two parts: the partial private key generated by the Private Key Generator (PKG) and the secret key selected by the user. Therefore, this scheme can solve the certificate management problem in PKI and the key escrow problem in IBE. Shim [22] proposed a secure conditional privacy-preserving authentication scheme (CPPA) using a pseudo-identity-based signature (IBS) scheme without using the MaptoPoint hash function [23]. This scheme achieves anonymous authentication, message integrity, traceability, and unlinkability, and it also maintains a balance between privacy and traceability. However, Liu [24] noted that Shim's scheme could not be normal existential unforgeable against adaptive chosen-identity and chosen-message attacks. Pankaj [25] proposed an efficient certificateless signature scheme in HWSN. However, the scheme is lack of traceability and preserving identity privacy. Also, it suffered from a high overhead using bilinear pairing operation.

To reduce the authentication time and improve the computational efficiency for VSNs, He et al. [26] propose an ID-based CPPA scheme for VSNs based on the Elliptic Curve Cryptography (ECC), which satisfies security and privacy requirements. The scheme is more efficient than previously proposed schemes for VSNs. However, this scheme heavily relies on a tamper-proof hardware device in which an important master secret key is preloaded for each vehicle. If the master secret key is extracted by adversaries though side-channel attacks, such as power analyses and laser scanning [22], all malicious messages generated by the adversaries can be successfully verified

and the entire system will be compromised. Lo et al. [27] proposed a faster ID-based scheme for VSNs based on ECC without using the special MaptoPoint hash function, which is efficient and consumes more computing time. This scheme also supports the batch signature and conditional privacy-preserving authentication; however, it is significantly dependent on secure communication channels. In the particle scenario, the vehicle-specific information is easily collected from overhearing the wireless network [7]. From the implementation perspective, the scheme has high costs and lacks of scalability. In addition, the schemes [26,27] suffered from privileged insider attacks in the PKG. If an adversary obtains the private key of one user issued by the PKG, he can easily forge a valid signature.

3. Background

In this section, we briefly introduce the network model and adversary model of our scheme.

3.1. Network Model

The proposed scheme applies a two-layer network model. The upper layer consists of the PKG, TA and a traffic information service center. The bottom layer includes vehicles equipped with wireless communication device and RSUs, which can communicate with one another using the DSRC/WAVE protocol.

Here, we should consider two application scenarios according to different locations of RSUs. First, RSUs are built on main roadways, which are the focus of most other schemes. The infrastructure and RSUs communicate through secure channels, such as the transport layer security protocol via wired connections [19]. Second, RSUs are deployed in unattended environments, such as highway roads. Thus, the cost of constructing optic and electric composite cables to provide power and communication between the RSUs and the infrastructure is high. In the second scenario, we deploy RSUs with batteries and short wireless communication ranges. Users can contact RSUs via single-hop or multi-hop communication, which is more robust and suitable for the second scenarios.

The two scenarios are shown in the Figures 1 and 2.

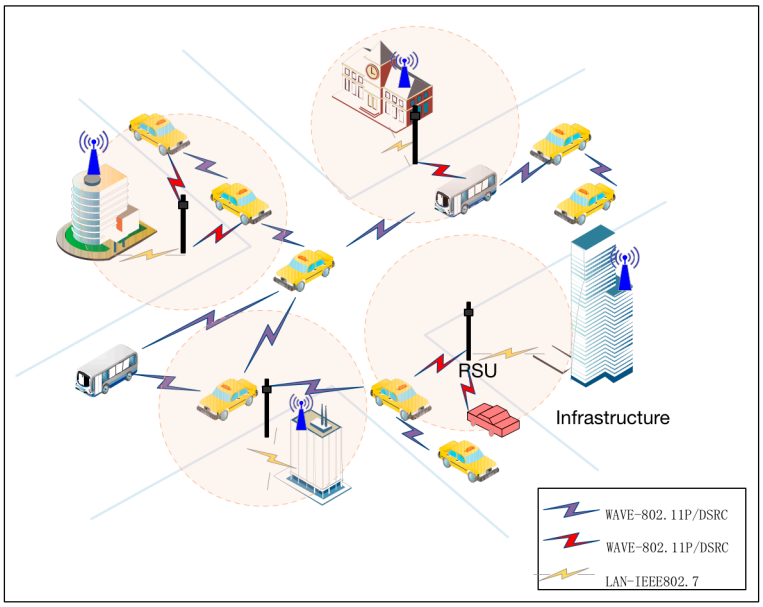

Figure 1. Network architecture on the main roadways.

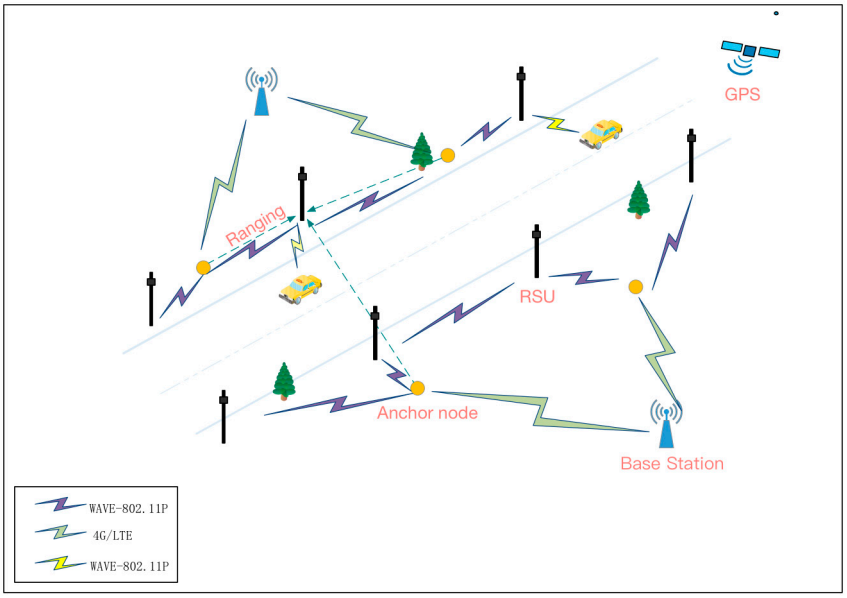

Figure 2. Network architecture in a desolate environment.

TA: The TA registers the drivers and generates pseudo identities for valid users. The TA is the only party that can trace the vehicle and reveal the identities from the signers. The TA cannot be compromised and is fully trusted by all parties in the system.

PKG: The PKG is a trusted third party that generates partial private keys for the signers.

RSUs: RSUs are distributed along road sides equipped with an on-board sensory, processing, and wireless access point, and they are mainly used to verify the messages and transfer data among the vehicles and infrastructure in its coverage area, such as the traffic information service center, TA and PKG.

Vehicle: All vehicles are equipped with card reader, on-board sensory, processing, and wireless communication modules. All users who want to access the services from the VSNs will be issued a smart card with system parameters, which can help the TA to track the behaviors back to the owner of the smart card instead of the car. Smart card technology conforms to international standards (ISO/IEC 7816 and ISO/IEC 14443) [28,29]. With an embedded microcontroller, each smart card can store large amounts of data, and they have the computing ability to perform on-card functions (e.g., signature and authentication). The smart card can interact with card reader, which is mounted on the car. The communication protocol with neighboring vehicles and RSUs is 5.9-GHz DSRC [3] IEEE 802.11p.

Anchor nodes: In Figure 2, to prevent adversaries from inserting malicious nodes into the networks, the key point of our approach is to deploy certain anchor nodes with higher processing capabilities and a global position system (GPS) receiver. These nodes can help the system to reduce the possibility of static nodes (RSUs and anchor nodes) compromise attacks and immediately detect nearby controlled nodes using our method. We elaborate on the function of anchor nodes in Section 4.3.

3.2. Adversary Model

In reality, all communication channels among VSN entities are not explicitly secure. In Lo's scheme, every transmit channel is assumed to be secure without considering this fact. In this paper, we assume that the communication channels are public and adversaries can conduct attacks, such as eavesdropping, insider attacks, stolen smart-card attacks and impersonation attacks, in which adversaries attempt to impersonate a legitimate user or a node. In addition, the adversary can conduct a physical attack on static nodes (RSUs and anchor nodes) and retrieve secret information and stored data from them particularly in an unwatched location. In further attacks, the adversary

attempts to replicate the controlled nodes, deploy them in other places and manipulate the network with the clones or captured nodes.

4. Proposed Scheme

In this section, we proposed an enhanced ID-based certificateless authentication scheme based on the modification of the original CLPKC mechanism [8]. The scheme supports the V2I and V2V communication, and it consists of five phases: System Initialization, Register, Login, Signing and Verification. The symbols of our scheme are described in Table 1.

Table 1. List of notations.

Symbol	Descriptions	Symbol	Descriptions
RSU	A roadside unit	d_1	A secret key of a user
TA	A Trusted Authority	d_2	The partial secret keys of a user issued by the PKG
PKG	A Private Key Generator	P_1	A public key of a user
n	A k-bit prime number	P_2	A public key of users issued by the PKG
F_n	A finite field with n elements	r	A private key of the TA
$E(F_n)$	An Elliptic Curve over a finite field F_n, $y^2 = x^3 + ax + bmodn, a, b, x, y \in F_n$	s	A private key of the PKG
b	A secret number in a smart card	PW	The password of the smart card
G	An additive group with the order q	P_{TA}	A public key of the TA
q	The order of the group G	P_{PKG}	A public key of the PKG
P	The point generator of the group G_q	$time$	A timestamp
PID	The pseudo identity of a user	$\oplus$	Exclusive-OR operation
RID	The real identity of a user	$\|\|$	Message concatenation operation

4.1. System Initialization

The PKG generates system parameters via running following steps. First, the PKG chooses a k-bit *prime* number n and generates the tuple $\{F_n, E(F_n), G_q, P\}$. Then the PKG picks a random number $s \in Z_q^*$ as its private key and computes $P_{PKG} = s{\cdot}P$. Furthermore, the PKG determines four one way hash functions: $h_0 : \{0,1\}^* \rightarrow Z_q^*$, $h_1 : \{0,1\}^* \times G_q \times \{0,1\}^* \rightarrow Z_q^*$, $h_2 : G_q^2 \times \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q^*$, $h_3 : \{0,1\}^* \times G_q^2 \times \{0,1\}^* \times G_q \times \{0,1\}^* \rightarrow Z_q^*$. The TA also selects a random $r \in Z_q^*$ as its private key and computes $P_{TA} = r{\cdot}P$. At last, the PKG publish system parameters $Z = \{F_n, E(F_n), G_q, P, P_{PKG}, P_{TA}, h_0, h_1, h_2, h_3\}$. The PKG and TA keep s and r secret, respectively.

4.2. Vehicle to RSU (the RSU Verifies the Vehicle)

4.2.1. Register

Every user who wants to access the services from VSNs is issued a smart card with system parameters offline from the TA at first. Note that the user must disclose his valid credentials such as ID card or driving license to the TA to get the smart card. The user's credential number (the real identity ID of the user) is input to the smart card by the TA and will be recorded in the list of TA. In the beginning of the smart card activation, the user inserts his smart card into a card reader mounted on a car, and input his real identity ID' and password PW. Note that the real identity is registered in the TA offline and can uniquely identify the user.

Upon receiving the ID' and PW, in which $ID \in Z_q^*$ and $PW \in Z_q^*$, the smart card compares ID' with the stored one. If true, the smart card calculates $h_0(PW \oplus b)$ and $h_0(ID)$, in which the $b \in Z_q^*$ is an arbitrary number and the length of b is enough large. Then the smart card selects a random number $d_1 \in Z_q^*$ as the user's secret value and generates the public key $P_1 = d_1{\cdot}P$. Subsequently, the smart

card sets $s_1 = h_0(PW \oplus b) \oplus ID$ and $s_2 = s_1 + d_1$. The smart card encrypts $\{ID, h_0(PW \oplus b), P_1\}$ using the TA's public key and sends it to the TA.

Upon receiving the register request, the TA decrypts it using the TA's private key r and checks whether the ID is legal, and if so, the TA will make m pseudo identities for the user. The TA computes:

$$PID_{1,i} = r \times h_1(Enc_{P_{TA}}(ID) \oplus h_0(PW \oplus b)||P_1||T) + n_i \bmod q\,, N_i = n_i \cdot P, (i = 1 \ldots m), \quad (1)$$

where $n_i \in Z_q^*$ is a random number, $T \in Z_q^*$ is the expiration date of the PID_1 and m is the number of PIDs. For convenience, we set $\{Enc_{P_{TA}}(ID) \oplus h_0(PW \oplus b)\} = H_1$. The TA encrypts these PIDs $\{PID_1, H_1, N, T\}$ using P_1 and sends it to the smart card. Note that the TA stores the $Enc_{P_{TA}}(ID)$ instead of the ID to prevent stolen ID list attacks. The TA stores the $\{PID, Enc_{P_{TA}}(ID), h_0(PW \oplus b), H_1, N\}$ in its memory.

When receives $Enc_{P_1}\{PID_1, H_1, N, T\}$, the smart card decrypts and checks them via running $PID_{1,i} \cdot P = P_{TA} \cdot h_1(H_1||P_1||T) + N_i, (i = 1 \ldots m)$. If the equations hold, which mean that adversaries do not tamper the pseudo identities, and the smart card calculates $PID_i = PID_{1,i} + d_1, (i = 1 \ldots m)$. Otherwise, reject the PID_1. Here, every PID. is generated as a combination of secret value of the TA and the user-chosen secret. Thus, adversaries cannot forge the valid PID without the user-chosen secret d_1. Subsequently, the smart card sends the tuples $\{PID, H_1, P_1, N, T\}$ to the PKG through a public channel.

Upon receiving the partial-secret-key request $\{PID, H_1, P_1, N, T\}$, the PKG validates the PIDs by checking whether the following equations:

$$PID_i \cdot P = P_{TA} \cdot h_1(H_1||P_1||T) + N_i + P_1, (i = 1 \ldots m) \quad (2)$$

hold within the validity of T. If yes, then the PKG generates partial secret keys for users as below:

$$P_{2,i} = k_i \cdot P$$

$$d_{2,i} = k_i + h_2(P_1, P_{2,i}, PID_i, T) \times s \bmod q, \ (i = 1 \ldots m) \quad (3)$$

where $k_i \in Z_q^*$ is a random number. The PKG sends $\{PID, P_2, d_2\}$ back to the smart card. Else, reject the partial-secret-key request.

Upon receiving the partial secret keys, the smart card checks the authenticity of $\{PID, P_2, d_2\}$ via running:

$$d_{2,i} \cdot P = P_{2,i} + h_2(P_1, P_{2,i}, PID_i, T) \cdot P_{PKG}, \ (i = 1 \ldots m). \quad (4)$$

If the equations hold, which imply that the $\{P_2, d_2\}$ are generated by the PKG. Otherwise, reject them. Then the smart card stores $\{PID, h_0(PW \oplus b), h_0(ID), s_2, P_1, P_2, d_2, b, T, N, H_1\}$ in the memory and deletes d_1, ID, PW, s_1 to prevent smart card compromise attacks. The steps of the phase are depicted in Figure 3.

4.2.2. Login and Message Signing

The user inserts his smart card into a card reader, and inputs ID' and PW'. Then the smart card compares $h_0(PW' \oplus b)$ and $h_0(ID')$ with the stored ones in it. If true, the smart card computes $s_1' = h_0(PW' \oplus b) \oplus ID'$ and $d_1' = s_1' \oplus s_2$, and checks the validity period of PIDs, then performs the following operations. Otherwise, reject the request. The smart card deletes the ID', s_1' and PW'.

1. Generate a traffic-related message M, then pick a random number $l \in Z_q^*$ and calculate $L = l \cdot P$ to give a freshness.
2. Choose a PID_i and its corresponding $d_{2,i}$, and calculate:

$$v = l + d_{2,i} + d_1' \times h_3(PID_i, P_1, P_{2,i}, M, L, time) \bmod q, \quad (5)$$

where *time* is the current timestamp of the users' system.

3. Send $\{PID_i, P_1, P_{2,i}, M, L, T, v, time\}$ to another VSN entities.

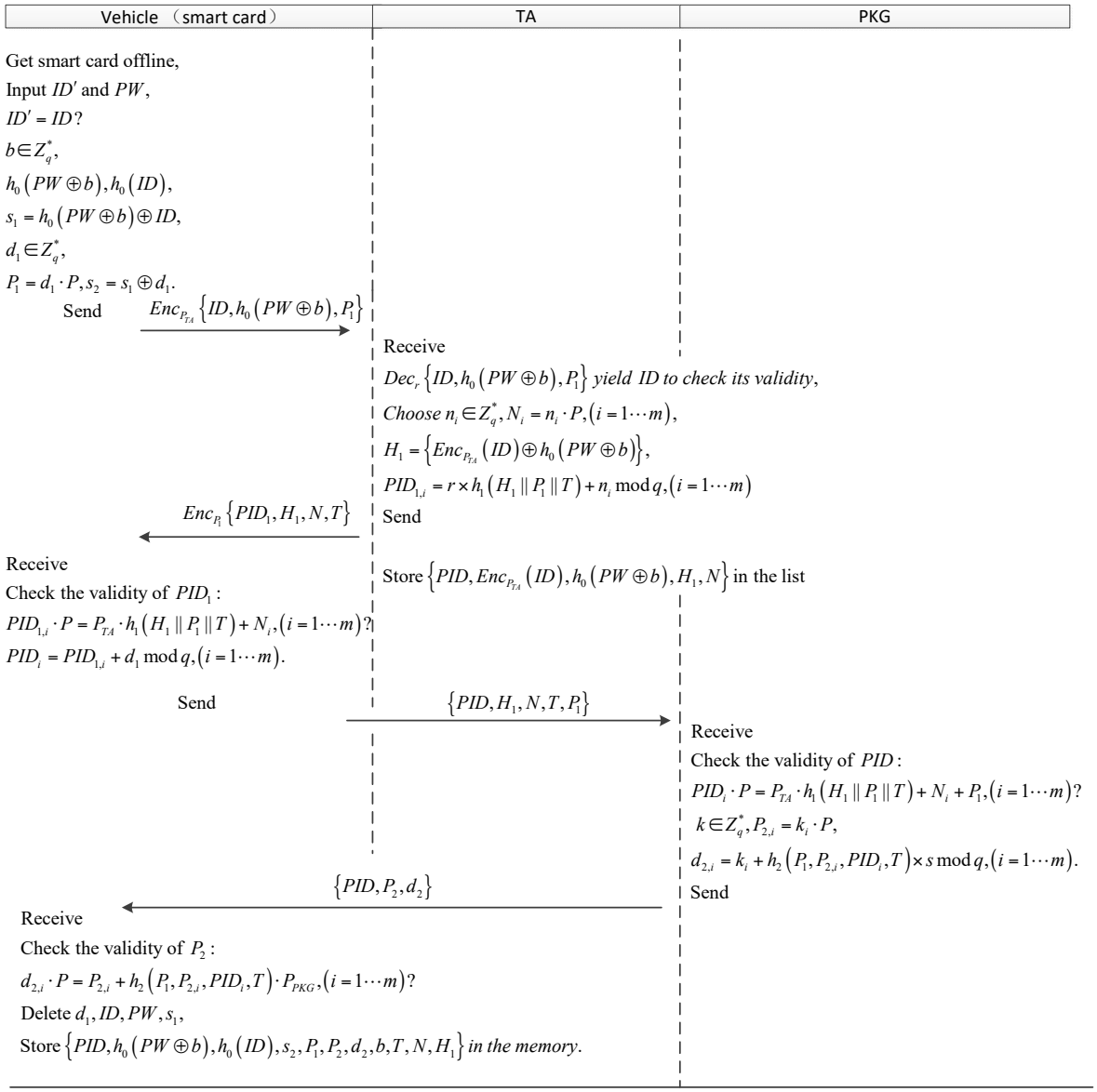

Figure 3. The vehicle to RSU (vehicle) registration process.

4.2.3. Verification

This phase is invoked when the verifier (a vehicle or RSU) receives the information $\{PID_i, P_1, P_{2,i}, M, L, T, v, time\}$ at the time $time^*$, it uses the system parameters $Z = \{F_n, E(F_n), G_q, P, P_{PKG}, P_{TA}, h_0, h_1, h_2, h_3\}$ to perform the following steps:

1. Validate the freshness of $time^*$. If $time^* - time \leq \cdot T$, then the verifier proceeds to the next step, else rejects the request, where $\cdot T$ indicates the valid time interval.
2. Then the verifier checks the expire time T of PID_i.
3. The verifier checks the equation:

$$v \cdot P = L + P_{2,i} + h_2(P_1, P_{2,i}, PID_i, T) \cdot P_{PKG} + P_1 \cdot h_3(PID_i, P_1, P_{2,i}, M, L, time) \quad (6)$$

If it holds, the verifier accepts the M, else outputs "invalid".

After the user log out, the smart card delete the d_1 from its memory to prevent stolen smart card attacks. The steps of the phase are depicted in Figure 4.

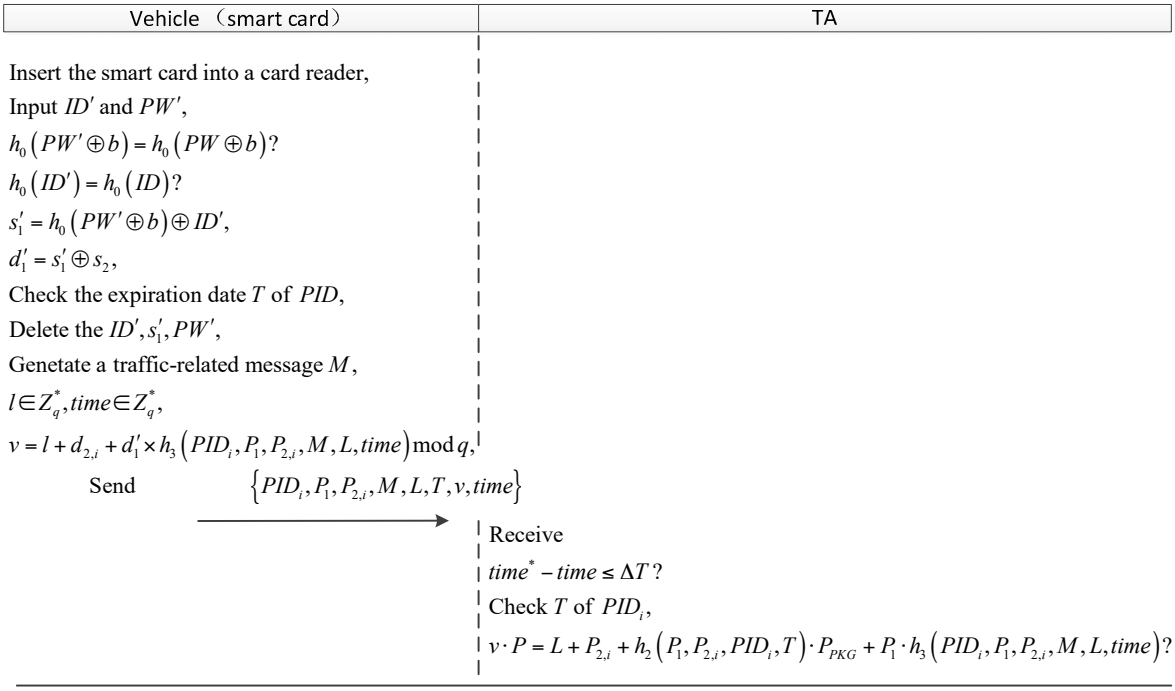

Figure 4. The vehicle to RSU (vehicle) authentication process.

4.2.4. Batch Verification

To enhance the effectiveness of the message verification, we require that vehicles or RSUs can aggregate n signatures into a single one and handle it at the same time. In the batch verification scheme, if one of the signatures is invalid, all signatures will be dropped or rejected. The proposed scheme supports batch verification. When the verifier receives numbers of requests, denoted as $\{PID_{i,x}, P_{1,x}, P_{2i,x}, M_x, L_x, T_x, v_x, time_x\}$, $(x = 1 \cdots n)$, it adds several random numbers to quickly detect which message is invalid in the batch. The concept is regarded as an efficient method in the batch verification [24].

The verifier checks the following equation:

$$\left(\sum_{x=1}^{n} y_x v_x\right) \cdot P = \sum_{x=1}^{n} y_x L_x + \sum_{x=1}^{n} y_x P_{2i,x} + \left(\sum_{x=1}^{n} y_x h_{3,x}(PID_{i,x}, P_{1,x}, P_{2i,x}, M_x, L_x, time_x) \cdot P_{1,x}\right) + \left(\sum_{x=1}^{n} y_x h_{2,x}(P_{1,x}, P_{2i,x}, PID_{i,x}, T_x)\right) \cdot P_{PKG}, \quad (7)$$

where $y_x (x = 1 \cdots n)$ are small random numbers.

If the equation holds, than the verifier accepts these messages, else detects the invalid messages and rejects them.

4.3. RSU to Vehicle (the Vehicle Verifies the RSU)

In this subsection, we use a position-based authentication method to reduce the possibility of node capture attacks.

As indicated in Section 3.1, there are two types of nodes. The anchor nodes and normal RSUs. The difference between them is that the anchor nodes obtain their position with the help of the built-in GPS receivers, whereas they are unknown for the RSUs. The anchor nodes have more computation

and energy power than that of the RSUs. The anchor node has two main functions. First, it broadcasts its position in real time to help nearby RSUs calculate their coordinates. Second, it can immediately detect abnormal RSUs inside its range.

We implement an efficient approach based on the Received Signal Strength Indication (RSSI) combined with the centroid algorithm [30], which is high accurate to obtain the position. RSSI-based location schemes are the most prevalent ones due to their easier implementation and less complexity [31], especially for the energy-constrained nodes. Therefore, with this method, if a RSU is captured and moved to another location, it will fail to be verified because that the new position incorporated in the signature is changed. Furthermore, the anchor node can immediately detect abnormal RSUs via comparing the two locations, and the first one is obtained by the GPS and the other one is calculated by nearby RSUs. If the value does not change a lot within the measurement uncertainties, then the nearby RSUs are valid, else abnormal RSUs must be surrounding the anchor node, say get captured, replicated, or moved by adversaries, and the anchor nodes will immediately alert to the PKG.

4.3.1. Initialization

Every RSU is preloaded a legitimate ID_{R1} assigned by the PKG, which is stored in its tamper-proof device. Every anchor node is assigned a ID_c and deployed in its pre-setup position by the PKG. After deployment, the RSU receives the position information from nearby anchor nodes at the first time. The details of the information are as follows:

$$\begin{aligned} L_{c1} &= \{ID_{c1},\ P_{c1},\ (x_{c1},\ y_{c1})\} \\ L_{c2} &= \{ID_{c2},\ P_{c2},\ (x_{c2},\ y_{c2})\} \\ L_{c3} &= \{ID_{c3},\ P_{c3},\ (x_{c3},\ y_{c3})\} \\ &\vdots \\ L_{ci} &= \{ID_{ci},\ P_{ci},\ (x_{ci},\ y_{ci})\}, \end{aligned} \tag{8}$$

where L_{ci} denotes the position information broadcasted by the anchor node, and $P_{ci} = d_{ci} \cdot P$ is its public key, in which $d_{ci} \in Z_q^*$ is a random number as its secret key, and $(x_{ci},\ y_{ci})$ is the current coordinates measured by the GPS.

The RSU computes its current coordinates $(x_R,\ y_R)$ according to the any of three coordinates of anchor nodes through centroid algorithm based on the RSSI [30] mentioned above and sets $ID_{R2} = h_0((x_R, y_R))$. Subsequently, the RSU chooses a random number $d_{R1} \in Z_q^*$ as its secret key, and sets $P_{R1} = d_{R1} \cdot P$. Then the RSU set $S_{d_{R1}} = Sign_{d_{R1}}\{ID_{R1} \parallel ID_{R2} \parallel L_{c1} \parallel L_{c2} \parallel L_{c3} \parallel \cdots \parallel L_{cn} \parallel P_{R1}\}$ signing with the secret key d_{R1} and encrypts the tuple $\{S_{d_{R1}} \parallel ID_{R1} \parallel ID_{R2} \parallel L_{c1} \parallel L_{c2} \parallel L_{c3} \parallel \cdots \parallel L_{cn} \parallel P_{R1}\}$ using the public key of the PKG, and the RSU sends it to the PKG.

Upon receiving the tuple, the PKG decrypts it and verifies the signature. Then the PKG compares the L_{ci} and ID_{R1} with the stored list to make sure that they are legitimate ones without being modified at the initialization step.

The PKG generates the partial secret key for RSUs as follows:

$$P_{R2} = k_R \cdot P$$

$$d_{R2} = k_R + h_2(P_{R1},\ P_{R2},\ ID_{R2},\ t) \times s \bmod q, \tag{9}$$

where $k_R \in Z_q^*$ is a random number and t is the expiration date of d_{R2}, then the PKG sends $\{ID_{R2},\ P_{R2},\ d_{R2},\ t\}$ back to the RSU.

The PKG calculates $ID_R = ID_{R1} \oplus ID_{R2}$ and $h_0(ID_{R1})$ in the next step, and deletes ID_{R1} and ID_{R2} from the list to avoid the stolen ID list attacks.

Upon receiving the $\{ID_{R2},\ P_{R2},\ d_{R2},\ t\}$, the RSU verifies the validity of d_{R2} via checking the equation $d_{R2}{\cdot}P\ =\ P_{R2}\ +\ h_2(P_{R1},\ P_{R2},\ ID_{R2},\ t){\cdot}P_{PKG}$. If the equation holds, then it accepts the d_{R2}, else it applies the PKG for the partial secret key again. Then the RSU calculates the short-term pairwise encryption keys:

$$\begin{aligned} k_1 &= d_{R1} \cdot P_{c1} \\ k_2 &= d_{R1} \cdot P_{c2} \\ k_3 &= d_{R1} \cdot P_{c3} \\ &\vdots \\ k_n &= d_{R1} \cdot P_{cn} \end{aligned} \tag{10}$$

between the anchor nodes and RSUs.

4.3.2. Message signing

The RSU picks a random number $l_R \in Z_q^*$ and sets $L_R\ =\ l_R{\cdot}P$, and it receives the location information from the anchor nodes and calculates the current coordinates $(x'_R,\ y'_R)$ by the location algorithm. Let B be a position tolerance value, and the RSU should compare the new coordinates $(x'_R,\ y'_R)$ with the previous one. If the distance $d\ =\ \sqrt{(x'_R\ -\ x_R)^2\ +\ (y'_R\ -\ y_R)^2} \le B$, then the RSU sets $ID'_{R2}\ =\ ID_{R2}$, else renews the value $ID_{R2}\ =\ ID'_{R2}$.

Then the RSU calculates:

$$v_R\ =\ l_R\ +\ d_{R2}\ +\ d_{R1}{\times}h_3(ID'_{R2},\ P_{R1},\ P_{R2},\ M,\ L_R,\ time)\ \mathrm{mod}\ q, \tag{11}$$

in which *time* is the current timestamp of the RSU's system and M is a traffic-related message.

Send $\{(x'_R,\ y'_R),\ ID'_{R2},\ P_{R1},\ P_{R2},\ M,\ L_R,\ t,\ time,\ v_R\}$ to another VSN entities.

4.3.3. Verification

When verifier such as a vehicle, anchor node or a RSU receives $\{(x'_R,\ y'_R),\ ID'_{R2},\ P_{R1},\ P_{R2},\ M,\ L_R,\ t,\ time,\ v_R\}$ at time $time^*$, it firstly checks the fressness of $time^*$ and the expiration time t of the partial private key d_{R2}.

The verifier checks the equation:

$$v_R{\cdot}P\ =\ L_R\ +\ P_{R2}\ +\ h_2(P_{R1},\ P_{R2},\ ID'_{R2},\ t){\cdot}P_{PKG}\ +\ P_{R1}{\cdot}h_3(ID'_{R2},\ P_{R1},P_{R2},\ M,\ L_R,\ time) \tag{12}$$

If the equation holds, the verifier accepts the message M.

Upon receiving the signed message, the nearby anchor nodes perform the different steps inside their range, which firstly check the list and if there is no short-term pairwise encryption key k_i with the RSU, the nodes calculate the k_i via $k_i\ =\ d_{cj}{\cdot}P_{R1,i}$. Furthermore, the anchor nodes recount their coordinates according to ID'_{R2} and compare with previous ones. If the value significantly changes, then the RSU is abnormal, which is forged by the adversaries, and the anchor node generates an alert that is sent to the PKG. To prevent location information tampering attacks by adversaries, the anchor node encrypts its location using k_i and broadcasts $L_{cj}\ =\ \{ID_{cj},\ P_{cj},\ (x_{cj},\ y_{cj}),\ h_{k_i}((x_{cj},\ y_{cj}))\}$ to RSUs next time.

Here, $h_{k_i}((x_{cj},\ y_{cj}))$ is an encrypted digest called HMAC, which is viewed as a hash function and encrypted by the session key k_i shared between the two entities. The steps of the phase are depicted in Figure 5.

The proposed scheme also supports the batch verification, and the process is as same as the one in Section 4.2.4.

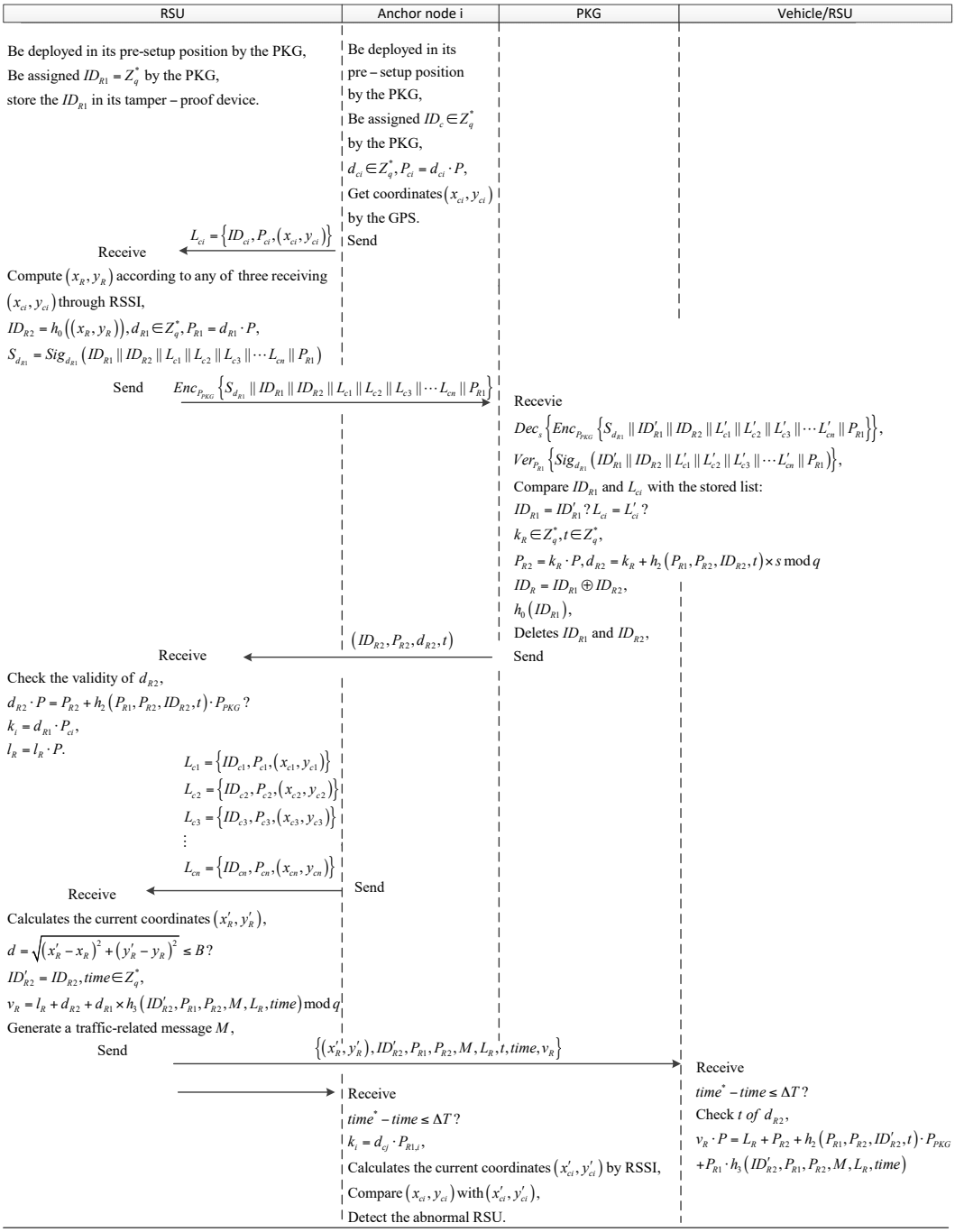

Figure 5. The RSU to vehicle (RSU) authentication process.

4.4. Key Update

To prevent key compromise attacks for a long time, key update periodically is required. We divide this section into two parts, the user-key update and the RSU-key update:

(1) Updating a user's P_W^i. This function is invoked whenever the user wants to update his password of the smart card. First, the user inserts his card into a card reader and inputs the original ID_i' and PW_i'. Then, the smart card calculates $h_0(PW_i' \oplus b)$ and $h_0(ID_i')$, and it checks whether $h_0(PW_i' \oplus b) = h_0(PW_i \oplus b)$ and $h_0(ID_i') = h_0(ID_i)$. If yes, the user will be allowed to input his new password PW_i^* and proceed to the next step, else abort. Subsequently, the smart card recounts $h_0(PW_i^* \oplus b_\prime)$ and $h_0(ID_i^*)$, in which $b_\prime$ is a new arbitrary number picked by the smart

card, then it updates $s_1^* = h_0(PW_i^* \oplus b_\prime) \oplus ID_i^*$ and $s_2^* = s_1^* \oplus d_1^*$, in which d_1^*, as the user's new secret value, is a random number reselected by the smart card. The subsequent steps are as same as the ones in Section 4.2.1.

(2) Updating a user's pseudo identities and partial secret keys. User's pseudo identities PIDs and partial secret keys share a same refresh cycle T. Every PID is appended an expiring time T by the TA for all users. Note that the period of T, which is relative to the key length and the complexity of circumstances, can be fixed by the administrator of the TA. When a user logs in the smart card, it firstly checks the T of PIDs, if the T is out of the valid date, the smart card terminates the following authentication process and informs the user to update the PIDs and related the partial secret keys. Note that any user cannot change the valid date T without the secret key of the PKG.

(3) Updating a RSU's partial secret key. In general, the process is as same as the one of user's. In addition, the updating phase is invoked when a valid RSU is authorized by the PKG to change its position. After deploying in a new location, the RSU will lunch a new handshake with the PKG to get a new partial secret key as same as the one in Section 4.3.1. Any node that attempts to change the position and tries to get a new key without the PKG's authority is considered as a malicious node.

5. Security Proof

In this section, we design four experiments to prove the security of the proposed scheme.

5.1. Experiment 1

We divide the kinds of adversaries into three according to their attack abilities in the scheme. The Type I adversary A1 is not able to access the master key of the PKG or the secret keys of users. The Type II adversary A2 represents a curious PKG who can access the master key of the PKG and obtain the partial secret keys of users but cannot forge secret keys of users. The type III adversary A3 represents a malicious PKG who not only obtains the master key of the PKG but also has the right to generate secret keys of users at will, but the keys are different from that of users.

Theorem 1. *We will demonstrate that our scheme is unforgeable against adaptive chosen message attacks of the adversary A1 under the random oracle due to the intractability of ECDLP.*

Proof. There are two roles in the game, the challenger C and the adversary A. C can solve the ECDLP problem with a non-negligible probability by running A as a subroutine. For instance, when C receives a problem $Q = s \cdot P$, $s \in Z_q^*$ is a random number, to calculates s is his target. C picks PID^* as a challenged identity and sets system public key $P_{PKG} = x \cdot P$, then C sends the system params $(p, q, P, P_{PKG}, h_1, h_2)$ to the adversary A1. We show the process, in which C can break ECDLP by using the adversary A as follows. C maintains 4 lists h_1^{list}, h_2^{list}, d_1^{list}, d_2^{list}, which are initially empty, and simulates oracles queried by A.

1. h_1 query. C maintains a list with the form of $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$. When A makes a query on $(PID_i, P_{1i}, P_{2i}, T_i)$, if the list contains the tuple $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$ matched PID_i, C returns B_i to A as a response. Otherwise, C chooses a random number coin $\leftarrow_R \{0, 1\}$ and sets $Pr[coin = 0] = \delta$, in which $coin = 0$ means that this PID_i is the challenged identity. Then C picks $B_i \leftarrow_R Z_q^*$ and sends $B_i = h_1(PID_i, P_{1i}, P_{2i}, T_i)$ to A as a response. C adds $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$ to h_1^{list}.
2. h_2 query. When A makes a query on $(PID_i, P_{1i}, P_{2i}, M_i, L_i, time_i)$, if the tuple $(PID_i, P_{1i}, P_{2i}, M_i, L_i, time_i, D_i)$ exists in the list, then C sends it to A as a response. Otherwise, C picks a random $D_i \in Z_q^*$ and sets $D_i = h_2(PID_i, P_{1i}, P_{2i}, M_i, L_i, time_i)$, and C sends it to A as a response. C adds $(PID_i, P_{1i}, P_{2i}, M_i, L_i, time_i, D_i)$ to h_2^{list}.

3. Private-key-extract query.
   If $coin = 0$, then C stops the session. Otherwise, C chooses a random number $d_{1i} \in Z_q^*$ as a private key of PID_i, and generates another two random numbers $d_{2i}, a_i \in Z_q^*$, and C sets $P_{1i} = d_{1i} \cdot P$, $h_{1i} \leftarrow a_i$ and $P_{2i} \leftarrow d_{2i} \cdot P - h_{1i} \cdot P_{PKG}$. C adds (PID_i, d_{1i}, P_{1i}) and (PID_i, d_{2i}, P_{2i}) to d_1^{list} and d_2^{list} respectively, then C returns d_{1i} to A as a response.
4. Partial-private-key-extract query.
   If $coin = 0$, then C stops the session. Otherwise, C looks up d_2^{list} and checks whether the tuple (PID_i, d_{2i}, P_{2i}) exist in the list first. If yes, C returns d_{2i} to A as a response. Else, C makes a private-key-extract query on PID_i itself and returns d_{2i} to A as a response.
5. Sign query.
   A makes a query on PID_i and M_i. C looks up $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$ firstly. If $coin = 0$, then C finds (PID_i, d_{1i}, P_{1i}) and (PID_i, d_{2i}, P_{2i}) in d_1^{list} and d_2^{list} respectively, and generates two random numbers $b_i, v_i \in Z_q^*$, and sets $h_{2i} \leftarrow b_i$, $L_i = v_i \cdot P - P_{2i} - h_{1i} \cdot P_{PKG} - P_{1i} \cdot b_i$. C returns $(PID_i, M_i, v_i, L_i, P_{1i}, P_{2i})$ to A as a response. Note that it is easy to verify the equation $v_i \cdot P = L_i + P_{2i} + c \cdot P_{PKG} + P_{1i} \cdot h_{2i}$ holds.
   If $coin = 1$, the signature is ordinary because that C knows the private key and partial private key.
6. Finally, A outputs (PID^*, M^*, v^*). Note that (PID^*, M^*) is not submitted to the query of private key, partial private key and signature. If $coin = 1$, then C stops the simulation. Otherwise, according to [32], A can generate another valid signature with the same random tape but the different value of h_{1i} as follows:

$$v' \cdot P = L_i + P_{2i} + h'_{1i} \cdot P_{PKG} + P_{1i} \cdot h_{2i} \tag{13}$$

$$v'' \cdot P = L_i + P_{2i} + h''_{1i} \cdot P_{PKG} + P_{1i} \cdot h_{2i} \tag{14}$$

According to the Equations (13) and (14), we can get:

$$v' - v'' \cdot P = (h'_{1i} - h''_{1i}) x \cdot P \tag{15}$$

$$x = (v' - v'')/(h'_{1i} - h''_{1i}) \bmod q \tag{16}$$

Thus, C outputs x as the solution of ECDLP problem $P_{PKG} = x \cdot P$. It is contradict to solve the ECDLP hard problem. □

Theorem 2. *Our scheme is secure against adaptive chosen message attacks of the super adversary A2 under the random oracle.*

Proof. There are two roles in the game, the challenger C and the adversary A. C use A as a subroutine to break our scheme via solving the ECDLP problem with a non-negligible probability. C picks a random number $s \in Z_q^*$ as the master key of the PKG and sets $P_{PKG} = s \cdot P$, then C generates the system params $(p, q, P, P_{PKG}, h_1, h_2)$. C sends s and the params $(p, q, P, P_{PKG}, h_1, h_2)$ to the adversary A2. C maintains 4 lists h_1^{list}, h_2^{list}, d_1^{list}, d_2^{list}, which are initially empty. C answers h_1 query and h_2 query like it does in the first oracle query phase. C simulates another oracles queried by A as follows.

1. Partial-private-key-extract query. If $coin = 0$, then C looks up h_1^{list} and identifies the tuple $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$, then C picks a random number $k_i \in Z_q^*$, and calculates $d_{2i} = k_i + s \times h_{1i} \bmod q$. C adds $(PID_i, \perp, P_{1i})$ and (PID_i, d_{2i}, P_{2i}) to d_1^{list} and d_2^{list} respectively. C returns d_{2i} to A as a response.
   If $coin = 1$, then C looks up h_1^{list} and identifies the tuple $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$, then C picks two random numbers $a_i, k_i \in Z_q^*$. C sets $d_{1i} \leftarrow a_i$, and calculates $d_{2i} = k_i + s \times h_{1i} \bmod q$ and $P_{1i} = d_{1i} \cdot P$. C adds (PID_i, d_{1i}, P_{1i}) and (PID_i, d_{2i}, P_{2i}) to d_1^{list} and d_2^{list} respectively. C returns d_{2i} to A as a response.

2. Private-key-extract query. When A makes the query, C does as follows:
   If $coin = 0$, then C stops the session. Otherwise, C looks up d_1^{list} and identifies the tuple (PID_i, d_{1i}, P_{1i}), and sends d_{1i} to A as a response. If there is no tuple in the list, C makes a partial-private-key-extract query on PID_i itself, then C returns d_{1i} as a response.
3. Sign query. A makes a query on PID_i and M_i. C looks up $(PID_i, P_{1i}, P_{2i}, T_i, B_i, coin)$ firstly. If $coin = 0$, then C finds $(PID_i, \perp, P_{1i})$ and (PID_i, d_{2i}, P_{2i}) in d_1^{list} and d_2^{list} respectively. C picks three random numbers $x, b_i, v_i \in Z_q^*$ and sets $P_{1i} = x \cdot P$, $h_{2i} \leftarrow b_i$ and $L_i = v_i \cdot P - P_{2i} - h_{1i} \cdot P_{PKG} - P_{1i} \cdot b_i$. C returns $(PID_i, M_i, v_i, L_i, P_{1i}, P_{2i})$ to A as a response. If $coin = 1$, the signature is ordinary.
4. Finally, A outputs (PID^*, M^*, v^*). Note that (PID^*, M^*) is not submitted to the query of private key and signature. If $coin = 1$, then C stops the simulation. Otherwise, according to [32], A can generate another valid signature with the same random tape but the different value of b_i as follows:

$$v' \cdot P = L_i + P_{2i} + h_{1i} \cdot P_{PKG} + P_{1i} \cdot b_i' \tag{17}$$

$$v'' \cdot P = L_i + P_{2i} + h_{1i} \cdot P_{PKG} + P_{1i} \cdot b_i'' \tag{18}$$

According to the Equations (17) and (18), we can obtain:

$$v' - v'' \cdot P = (b_{1i}' - b_{1i}'') x \cdot P \tag{19}$$

$$x = (v' - v'')/(b_{1i}' - b_{1i}'') \bmod q \tag{20}$$

Thus, C outputs x as the solution ECDLP problem $P_{1i} = x \cdot P$. □

Theorem 3. *Our scheme is secure against the super adversary A3 attacks.*

Proof. In this scenario, A3 presents a malicious PKG who can obtain the master key s of the PKG and forge the secret key d_i' at will. His target is to obtain the successful verification by another valid VSN entities. Nevertheless, a valid signature cannot be produced without the unique secret key d_1. In our scheme, PID is generated via calculating $PID_i = r \times h_1(H_1||P_1||T) + n_i + d_1 \bmod q$. Thus, the adversary has to obtain d_1 from valid users. It is difficult to steal d_1 from the smart card without the user's PW because that there is no d_1 stored in the smart card after logging out. Moreover, because of the intractability of ECDLP problem, the adversary cannot obtain d_1 from $P_1 = d_1 \cdot P$ and the TA's master key r from $P_{TA} = r \cdot P$. The probability of this malicious PKG managing to collude with the TA and stealing the master key from the TA is negligible. Therefore, the scheme is secure against this kind of adversary attacks, which leaves the opportunity to adversaries in [26,27], though. □

5.2. Experiment 2

In the register phrase, the proposed scheme can resist against the inner attacker from the TA. Every pseudo identity PID_i contains the TA's master secret key r and the user's private key d_1. Without knowing the user's private key d_1, any insider adversaries fail to impersonate the valid user to proceed with the next step. In this experiment, if the adversary cannot forge a valid pseudo identity PID_i verified by PKG successfully, the proposed scheme is secure against impersonation attacks by insider adversaries. The secure module with proof in the random oracle is as follows:

Proof. Suppose there is an adversary A that represents an inner attacker from TA and he is able to access TA's master secret key r but cannot get user's private key d_1 or forge it. This assumption is reasonable, because that the adversary has no right to modify the ID table in the TA. We construct a challenger C, which can solve ECDLP with a non-negligible probability by running A as a subroutine. C picks ID^* as a challenged identity and sets system public key $P_{TA} = r \cdot P$, in which $r \in Z_q^*$ is the

master secret key, then C sends the system params $(p,\ q,\ P,\ P_{TA},\ h)$ to the adversary A. C maintains 3 lists h^{list}, d_1^{list} and TA^{list} which are initially empty.

1. h query. C maintains a list with the form of $(ID_i,\ P_{1i},\ T_i,\ H_1, \delta_i,\ coin)$. When A makes a query on $(ID_i,\ P_{1i},\ T_i,\ H_1)$, C checks whether the tuple exist in the list h^{list}. If so, C responds $\delta_i = h(ID_i,\ P_{1i},\ T_i,\ H_1)$; otherwise, C generates a random number $coin \leftarrow_R \{0,\ 1\}$ and sets $\Pr[coin = 0] = \eta$, in which $coin = 0$ means that this ID_i is the challenged identity. Then C picks $\delta_i \leftarrow_R Z_q^*$ and sends $\delta_i = h(ID_i,\ P_{1i},\ T_i,\ H_1)$ to A as a response. C adds $(ID_i,\ P_{1i},\ T_i,\ H_1,\ \delta_i,\ coin)$ to h^{list}.
2. Master-secret-key query. When A makes the query, C does as follows:
   C looks up $(ID_i,\ P_{1i},\ T_i,\ H_1,\ \delta_i,\ coin)$ firstly. If $coin = 1$, C picks a random number $a_i \in Z_q^*$. C sets $d_{1i} \leftarrow a_i$ and calculates $P_{1i} = d_{1i} \cdot P$, then C adds $(ID_i,\ d_{1i},\ P_{1i})$ and $(ID_i,\ r)$ to d_1^{list} and TA^{list} respectively. C returns r to A as a response.
   If $coin = 0$, C adds $(ID_i,\ \perp,\ P_{1i})$ and $(ID_i,\ r)$ to d_1^{list} and TA^{list} respectively. C returns r to A as a response.
3. Private-key-extract query. C looks up $(ID_i,\ P_{1i},\ T_i,\ H_1,\ \delta_i,\ coin)$ firstly. If $coin = 0$, then C stops the session. Otherwise, C looks up d_1^{list} and identifies the tuple $(PID_i,\ d_{1i},\ P_{1i})$. Then C sends d_{1i} to A as a response. If there is no tuple in the list, C makes a master-secret-key query on ID_i itself, then C returns d_{1i} as a response.
4. PID query. A makes a PID_i query on ID_i. C looks up $(ID_i,\ P_{1i},\ T_i,\ H_1,\ \delta_i,\ coin)$ firstly. If $coin = 0$, then C finds $(ID_i,\ \perp,\ P_{1i})$ and $(ID_i,\ r)$ in d_1^{list} and TA^{list} respectively. C picks three random numbers $x, b_i, PID_i \in Z_q^*$, then C sets $P_{1i} = x \cdot P$, $h_i \leftarrow b_i$ and $N_i = PID_i \cdot P - b_i \cdot P_{TA} - P_{1i}$. C returns $(ID_i,\ v_i,\ N_i,\ P_{1i})$ to A as a response. If $coin = 1$, the PID_i is ordinary.
5. Finally, A outputs $(ID^*,\ PID^*)$. Note that $(ID^*,\ PID^*)$ is not submitted to the query of private key and PID. If $coin = 1$, then C stops the simulation. Otherwise, according to [32], A can generate another valid pseudo identity with the same random tape but the different coefficient m of P_{1i} as follows:

$$PID' \cdot P = N_i + P_{1i} + P_{TA} \cdot b_i \tag{21}$$

$$PID'' \cdot P = N_i + m \cdot P_{1i} + P_{TA} \cdot b_i \tag{22}$$

According to the Equations (21) and (22), we can obtain:

$$(PID' - PID'') \cdot P = (1 - m) x \cdot P \tag{23}$$

$$x = (PID' - PID'')/(1 - m) \bmod q \tag{24}$$

Thus, C outputs x as the solution ECDLP problem $P_{1i} = x \cdot P$. The ability of solving the ECDLP problem contradicts the hardness of the ECDLP problem. Therefore, the proposed scheme is secure against impersonation attacks by insider attackers from TA. □

5.3. Experiment 3

In the authentication process, we make use of two elements to provide the freshness of the signed message. The comparison of different schemes in the Figure 6 shows the importance of k_i and l in the signed message $\{PID_i,\ P_1, P_{2,i},\ M,\ L,\ T,\ v,\ time\}$.

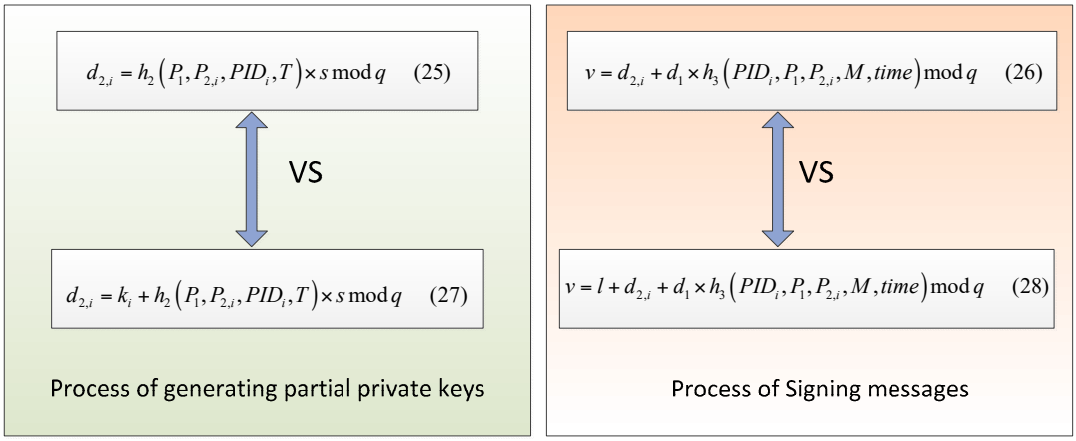

Figure 6. Comparison of two different schemes.

Proof. Note that without k_i and l it is easy for adversaries to get master secret key s and of PKG and private key d_1 in the Equations (25) and (26).

The adversary can acquire $\{PID,\ P_2,\ d_2\}$ from the public channel. It is easy to compute s by following steps:

(1) Get P_1 and T from the public message $\{PID,\ H_1,\ P_1,\ N,\ T\}$.
(2) Get $\{PID,\ P_2,\ d_2\}$ from the public channel.
(3) Compute s:

$$d_{2,i} = h_2(P_1,\ P_{2,i},\ PID_i,\ T) \times s \bmod q \tag{29}$$

$$s = d_{2,i}/h_2(P_1,\ P_{2,i},\ PID_i,\ T) \bmod q \tag{30}$$

It is easy to compute d_1 for adversaries in the same way.

(1) Get d_2 from the public message $\{PID,\ P_2,\ d_2\}$.
(2) Compute $h_3(PID_i, P_1,\ P_{2,i},\ M,\ time)$ by $\{PID_i,\ P_1,\ P_{2,i},\ M,\ T,\ v,\ time\}$ from the public channel.
(3) Compute d_1:

$$v = d_{2,i} + d_1 \times h_3(PID_i, P_1, P_{2,i}, M, time) \bmod q \tag{31}$$

$$d_1 = (v - d_{2,i})/h_3(PID_i, P_1, P_{2,i}, M, time) \bmod q \tag{32}$$

□

In order to protect the master key of PKG and user's private key, we add two elements to the Equations (25) and (26). The secure module with proof using random oracle is as follows:

In this experiment, assume that to forge the valid k that make $d_{2,i} = k_i + h_2(P_1, P_{2,i}, PID_i, T) \times s \bmod q,\ (i = 1 \ldots m)$ be verified successfully is the adversary's target. That means the adversary can compute right k and then achieve the value of s.

Proof. Suppose there is an adversary A that is not able to access the master key of the PKG or the secret value k but can access the partial private key d_2 of users. Note that in this experiment the adversary just play this game by himself to forge the k, so d_2 can be seemed as a public number without being verified by others. We construct a challenger C, which can solve ECDLP with a non-negligible probability by running A as a subroutine. C picks PID^* as a challenged identity and sets system public key $P_{PKG} = s \cdot P$, in which $s \in Z_q^*$ is the master secret key, then C sends the system params(p, q, P, P_{PKG}, h) to the adversary A. C maintains 2 lists h^{list} and PKG^{list} which are initially empty.

1. h query. C maintains a list with the form of $(PID_i, P_{1i}, P_{2i}, \theta_i, coin)$. When A makes a query on (PID_i, P_{1i}, P_{2i}), C checks whether the tuple exist in the list h^{list}. If so, C responds

$\theta_i = h(PID_i, P_{1i}, P_{2i})$; otherwise, C generates a random number $coin \leftarrow_R \{0,1\}$ and sets $\Pr[coin = 0] = \eta$, in which $coin = 0$ means that this PID_i is the challenged identity. Then C picks $\theta_{ii} \leftarrow_R Z_q^*$ and sends $\theta_i = h(PID_i, P_{1i}, P_{2i})$ to A as a response. C adds $(PID_i, P_{1i}, P_{2i}, \theta_i, coin)$ to h^{list}.

2. Master-secret-key query. When A makes the query, C does as follows:
   C looks up $(PID_i, P_{1i}, P_{2i}, \theta_i, coin)$ firstly. If $coin = 1$, C adds (PID_i, s) to PKG^{list}. C returns s to A as a response.
   If $coin = 0.$, then C stops the session.
3. k query. When A makes a k query on PID_i. C looks up $(PID_i, P_{1i}, P_{2i}, \theta_i, coin)$ firstly. If $coin = 0$, then C finds (PID_i, s) in the PKG^{list}. C picks a random number $b_i \in Z_q^*$, then C sets $h_i \leftarrow b_i$ and $D_i = k_i \cdot P + b_i \cdot P_{PKG}$, in which $D_i = d_{2,i} \cdot P$. C returns (PID_i, k_i, D_i) to A as a response. If $coin = 1$, the k_i is ordinary.
4. Finally, A outputs (PID^*, k^*). Note that (PID^*, k^*) is not submitted to the query of k. If $coin = 1$, then C stops the simulation. Otherwise, according to [32], A can generate another valid pseudo identity with the same random tape but the different values of b_i as follows:

$$k' \cdot P = D_i - P_{PKG} \cdot b_i' \tag{33}$$

$$k'' \cdot P = D_i - P_{PKG} \cdot b_i'' \tag{34}$$

According to the Equations (33) and (34), we can obtain

$$(k' - k'') \cdot P = (b_i'' - b_i') s \cdot P \tag{35}$$

$$s = (k' - k'')/(b_i'' - b_i') \bmod q \tag{36}$$

Thus, C outputs s as the solution ECDLP problem $P_{PKG} = s \cdot P$. The ability of solving the ECDLP problem contradicts the hardness of the ECDLP problem. Thus, the adversary cannot forge a valid k to compute the master key of the PKG.

The freshness of L in the Equation (27) that has the same function with k is to protect the private key of users. We will omit the same proof. □

5.4. Experiment 4

The proposed scheme implements a location-based method, with which every RSU can acquire their current coordinates and apply them in every signature. The freshness of current location protects RSUs from being captured and compromised.

Furthermore, every signature including a timestamp *time* is to record the current sending time of the signer. Verifiers can check out the replay attack easily by validating the freshness of receiving $time^*$. If $time^* - time > \Delta T$, in which ΔT indicates the valid time interval, the verifier will reject the signature. Figure 7 shows the function of the coordinates (x_R', y_R') and the timestamp $time^*$ included in the signature.

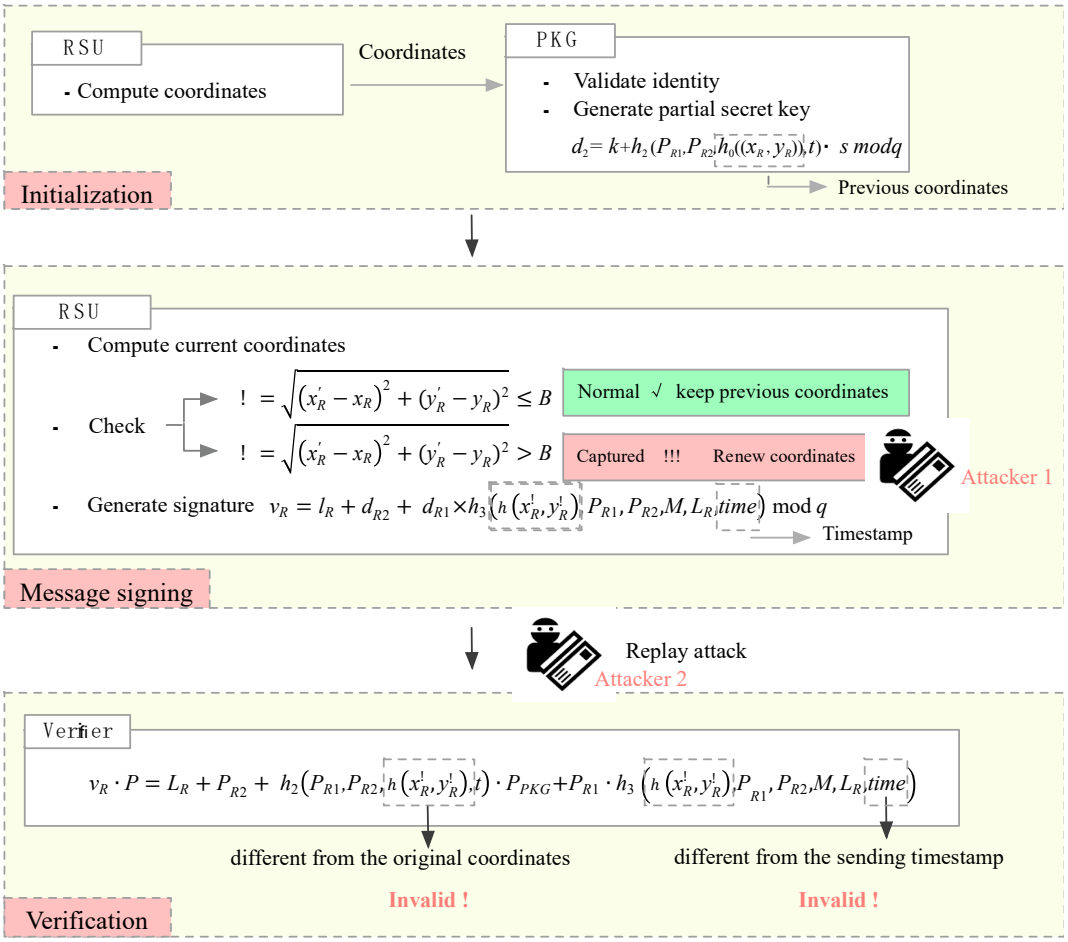

Figure 7. Freshness of timestamp and coordinates.

Analysis: In Figure 7, there are two attackers. The first one implements node captured attacks and the second one captures valid signatures to carry out replay attacks. Because of the different location, the attacker 1 can access any of information in the compromised RSU expect d_2. The ability of this kind of attackers is weaker than the adversary A3 as mentioned in the experiment 1. The ability of the attacker 2 is as same as the adversary A1 that is not able to access the master key of the PKG or the secret keys of users. However, they all fail to generate valid signatures and the proof is mentioned above.

6. Security Analysis

Considering the implementation costs, it's difficult to make all communication channels secure in VSNs. In our scheme, all communication channels are public, which is different from that in [27]. The TA is credible without being stolen its secret key by adversaries and its master key must be strongly protected by hardware technology.

The proposed scheme is on the basis of the CLPKC. Thus, our scheme can provide message authentication and integrity. The unforgeability against adaptive chosen messages attacks is defined in Section 5, which also provides the details of the scheme and its security proof. Thus, our scheme supports message authentication, integrity and unforgeability. The other security analyses are given in details as follows.

6.1. Traceability

The proposed scheme provides traceability. If one message is disputable, TA, the only authorized entity, can perform the tracing procedure and extract the real identity from the signature

$\{PID, P_1, P_2, M, L, T, v, time\}$ via calculating $PID \cdot P = P_{TA} \cdot h_1(H_1||P_1||T) + N + P_1$, in which H_1 and N are stored in its repository. If one $H_{1,j}$ satisfied the equation as above, the TA can obtain the $(ID_j)_{P_{TA}}$ from $(ID_j)_{P_{TA}} \oplus h_0(PW_j \oplus b) = H_{1,j}$ and extract the real identity ID_j by decrypting $(ID_j)_{P_{TA}}$ using the secret key r of the TA. Note that no one can obtain ID_j since r is only known by the TA itself.

6.2. Unlinkability

Unlinkability is that an adversary cannot link the signature messages generated by the same vehicle. Every signature message $\{PID, P_1, P_2, M, L, T, v, time\}$ is different, because it is signed by different PIDs and related partial private keys. $PID = r \times h_1(H_1||P_1||T) + n + d_1 \bmod q$ is generated by the random number n which any adversary who want to obtain will encounter the ECDLP problem. Therefore, the proposed scheme supports unlinkability.

6.3. Resistance against Impersonation Attacks

An adversary can impersonate a legitimate user to access RSUs by generating a valid PID and a signature message $\{PID, P_1, P_2, M, L, T, v, time\}$. With our scheme, every pseudo identity PID_i contains the TA's master secret key r and the user's private key d_1. Furthermore, every signature includes the PKG's master secret key s and d_1. Without knowing the user's private key d_1, any insider adversaries of the PKG fail to calculate the valid PIDs and signatures. The proof is given in Section 5.2. Note that d_1 is not transferred through any channels or stored in the smart card, and when the user does not input his valid PW, the smart card cannot obtain the valid d_1. Therefore, it is difficult for any adversaries to obtain d_1 by various methods of attack and because of the ECDLP problems, they cannot extract d_1 from $P_1 = d_1 \cdot P$. Assume that there is an adversary who eavesdrops the information $\{PID_1, H_1, N, T\}$ of one user or eavesdrops $\{P_2, d_2\}$ from the PKG through the public channels instead of the valid user, they all fail to generate valid PIDs and signatures because of lacking d_1.

6.4. Resistance against Node Compromise Attacks and Node Replication Attacks

The proposed scheme can prevent against node compromise and replication attacks to a large extent, and it incorporates three subsections according to the attacker's abilities:

(1) We assume that an adversary captures a node RSU_i and does not move this node to another location. The adversary extracts all stored information from the node, however, the information is independent of other nodes. And the adversary modifies the safety messages according to his specific needs and causes data anomalies. The position-based authentication method can help the PKG identify the malicious node based on its coordinates. Note that the adversary cannot change the node's coordinates or it will fail to be verified. In addition, there is no need to compromise the anchor node because this type of node does not contain important traffic information or privacy of users.

(2) Assuming that an adversary captures a node RSU_i and replicates it in another place, this new replicated node executes the same program as before. However, the node cannot generate valid signatures because it computes a current position $ID'_{R2} = h_0(x'_R, y'_R)$ according to new nearby anchor nodes. Note that ID'_{R2} is different from the original ID_{R2} in $d_{R2} = k_R + h_2(P_{R1}, P_{R2}, ID_{R2}, t) \times s \bmod q$. Therefore, these malicious nodes will be identified quickly by the verifiers because of their invalid signatures.

(3) We assume that there is a powerful adversary who can modify the original program in the node after capturing and replicating it in another location. Note that the adversary cannot change ID_{R2} in $d_{R2} = k_R + h_2(P_{R1}, P_{R2}, ID_{R2}, t) \times s \bmod q$. without knowing the master private key s. Therefore, to generate a valid signature the adversary only uses the original value of ID_{R2} instead of updating it vie the new anchor nodes. Unfortunately, these malicious nodes will be identified rapidly by the detection mechanism of the proposed method because of their wrong coordinates. When the adjacent anchor nodes receive the signature

$\{(x'_R,\ y'_R),\ ID'_{R2},\ P_{R1},\ P_{R2},\ M,\ L_R,\ t,\ time,\ v_R\}$, they compare their current location calculated by $(x'_R,\ y'_R)$ with the previous one, which is obtained from the GPS. If the value significantly changes, then abnormal RSUs must be surrounding the anchor node, and the anchor node will generate an alert to the PKG. Therefore, our scheme can withstand node compromise and replication attacks.

6.5. Resistance against Stolen Smart Card Attacks

We assume that the smart card of user U_i has been lost or stolen by an adversary. The adversary can then extract the parameters $\{h_0(PW \oplus b), h_0(ID), s_2, P_1, P_2, d_2,\ b,\ T,\ N, H_1\}$ stored in the smart card, although the user's independent information $\{d_1,\ PW,\ ID, s_1\}$ is not contained in the card. Moreover, calculating or guessing the user's correct value of PW_i, ID_i and $d_{1,i}$ is difficult. Therefore, the adversary cannot acquire the secret credentials of the target user. In addition, our proposal does not maintain any real-identity table, such as the RSU's ID_{R1}, ID_{R2} in the PKG and the user's ID_i in the TA to safeguard against stolen identity attacks by privileged insiders.

6.6. Resistance against Replay Attacks

All valid signatures maintain the timestamp *time*. The verifiers can find the replay message via checking whether $time^* - time \leq \Delta T$. Therefore, the proposed scheme can withstand the replay attacks. Table 2 shows the security compared with recently proposed authentication schemes in [15,22,27].

Table 2. Security Comparisons of Related Schemes and Our Scheme.

The Types of Attacks	Calandriello 's Scheme	Shim's Scheme	Lo's Scheme	Our Scheme
Traceability	No	YES	YES	YES
Unlinkability	YES	YES	YES	YES
Resistance to impersonation attack	YES	YES	YES	YES
Resistance to node replication attack	No	No	No	YES
Resistance to node compromise attack	No	No	No	YES
Resistance against replay attack	No	YES	YES	YES

7. Performance Evaluation

In this section, we analyze the computational costs and transmission overhead of our scheme. We implement our scheme using a Lenovo computer (Beijing, China) equipped with an Intel I7 dual-core processor, a 2.60 GHZ clock frequency and 1 gigabytes of memory running the VMWare Ubuntu12.03 operating system. For our ID-based scheme with ECC, we use an additive group G generated by a point p with the order q on the secp256r1 elliptic curve to achieve the security level of 128 bits, in which p and q are two 256-bit prime numbers. For the bilinear pairings based scheme, we use the bilinear pairings $y = x^3 + b \bmod q$ with embedding degree 12 and the q is a 256-bit prime number.

7.1. Computational Overhead

For convenience, we define some notations about the execution time as follows. First, Let T_{bp} denote the execution time of a bilinear pairing operation, T_{hmtp} be the time to execute one MapToPoint hash operation that is different from the general hash function operation T_h. Then T_{epm} and T_{epa} denote the time of executing one point multiplication and one point addition over an elliptic curve respectively. T_{RSSI} represents the time of computing coordinates of a RSU. At last, $T_{ecc-sign}$ and $T_{ecc-verify}$ represent the time of signing one message and verifying one message based on the secp256r1 elliptic curve respectively. The execution time of aforementioned operations is listed in Table 3.

Table 3. Execution Time of Different Operations.

Operation	Execution Time (Microsecond)
T_{bp}	2000
T_{hmtp}	4.398
T_{epm}	4.46×10^{-6}
T_{epa}	6.552
T_h	2.294
T_{RSSI}	11.072 [a]
$T_{ecc-sign}$	3460
$T_{ecc-verify}$	7634

[a] $T_{RSSI} = 2.649 \times 4 + 0.1584 \times 2 + 0.0272 \times 4 + 0.0486 = 11.072$ µs.

We compare the execution time of our scheme with other related works in [15,19,22,27]. Table 4 shows the execution time of signing a single message and a batch verification of five different schemes.

Table 4. Comparisons of the execution time of five schemes.

Method	Signing a Single Message (µs)	Verify a Single Message (µs)	Verify *n* Messages (µs)
Giorgio's scheme	$T = T_{ecc-sign} = 3460$	$T = T_{ecc-verify} = 7634$	$T = nT_{ecc-verify} = 7634n$ [a]
Shim's scheme	$T = 2T_{epm} + T_{epa} + T_h = 8.6$	$T = 3T_{bp} + T_{epa} + 2T_{epm} + 2T_h = 6011$	$T = 3T_{bp} + (3n-2)T_{epa} + (n+1)T_{epm} + 2nT_h = 24.2n + 5986.6$
Lo's scheme	$T = T_h + T_{epm} = 2.3$	$T = 2T_h + 3T_{epm} + 2T_{epa} = 17.7$	$T = 2nT_h + 2nT_{epa} + (n+2)T_{epm} = 17.7n$
Horng's scheme	$T = T_h + 4T_{epm} + T_{epa} + 2T_{hmtp} = 17.64$	$T = 2T_{bp} + T_h + T_{epa} + 2T_{epm} + T_{hmtp} = 4013.2$	$T = 2T_{bp} + nT_h + (3n-1)T_{epa} + 3nT_{epm} + nT_{hmtp} = 26.3n + 3993.5$
Our scheme	Vehicle: $T = T_h + T_{epm} = 2.3$ RSU: $T = T_{RSSI} + T_h + T_{epm} = 13.4$	$T = 2T_h + 3T_{epa} + 3T_{epm} = 24.2$	$T = 2nT_h + 3nT_{epa} + (n+2)T_{epm} = 24.2n$

[a] n is the number of messages.

In our scheme, a vehicle signing a message takes 2.3 µs and the RSU processing 13.4 µs, which is slightly slower than that of Lo's scheme. However, the proposed scheme provides better scalability without providing a specific secure channel, which is different from Lo's scheme, and our scheme can resist node compromise attacks, which other schemes do not consider. Therefore, the proposed scheme is efficient in terms of computational overhead and more secure than other schemes. More precisely, the proposed scheme can obtain better trade-offs than the four other schemes.

Next, we compare the performance of batch verification in the proposed scheme with that of the other three proposed ID-based batch verification schemes.

Figure 8 shows the relationship between the density of signing messages at a VSN entity inside its wireless range and the verification delay. The verification delay of the proposed scheme, which is 6.5 µs for one message, is slightly longer than the one in Lo's scheme. However, the difference is small, and the safety of our scheme is enhanced largely.

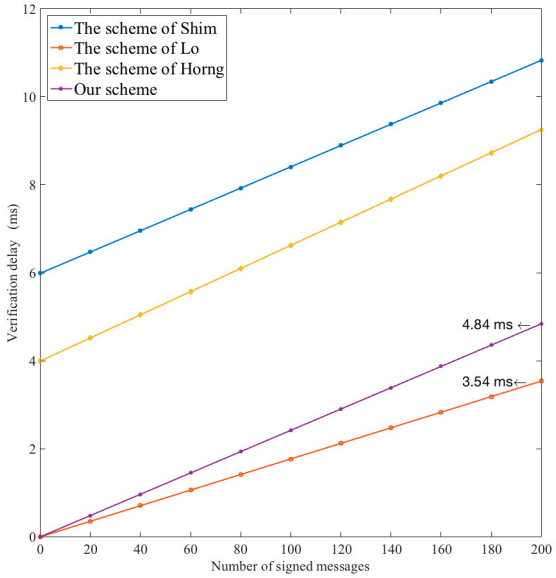

Figure 8. Comparison of execution time for the batch verification.

7.2. Communication Overhead

In this subsection, we analyze the communication overhead in our scheme and compare it with other proposed schemes. In our scheme, the signed message contains $\{PID,\ P_1, P_2,\ M,\ L,\ T,\ v,\ time\}$ and $\{(x'_R, y'_R),\ ID'_{R2},\ P_{R1},\ P_{R2},\ M,\ L_R,\ t,\ time,\ v_R\}$ for a vehicle and a RSU respectively. Since the length of p and q is 256 bits, so the length of element of G is 512 bits. The length of M is about 256 bits, which is the same as the value of the general hash function. Let timestamp, expiration time and the coordinates of one node be 32 bits. Table 5 shows the communication costs of our scheme and Table 6 shows the comparison of communication overhead among four schemes.

Table 5. Communication costs of the proposed scheme.

Communication Costs for a Vehicle (bit)	PID	P_1	P_2	M	L	v	Timestamp	T	-
	256	512	512	256	512	256	32	32	-
Communication Costs for a RSU (bit)	(x'_R, y'_R)	ID'_{R2}	P_{R1}	P_{R2}	M	L_R	Timestamp	t	v_R
	32	256	512	512	256	512	32	32	256

Table 6. Comparison of communication costs.

Method	Communication Overhead	After Reduction (byte)
Shim's Scheme	512 + 512 + 32 + 256 + 32 + 512 + 512 + 512 = 2880 bits = 360 bytes	232
Lo's Scheme	512 + 512 + 32 + 256 + 32 + 512 + 512 + 256 = 2624 bits = 328 bytes	232
Horng's Scheme	512 + 512 + 512 + 256 + 512 = 2304 bits = 288 bytes	224
Our Scheme	For a vehicle: 296 bytes For a RSU: 300 bytes	200 204

The communication overhead of proposed scheme is about 296 bytes and 300 bytes for a vehicle and a RSU respectively. To reduce the communication overhead, the key point in the proposed scheme is how to reduce the costs of the elements in G. Shim [22] developed a method, which can reduce the size of a point (x, y) in G. In this method, the entity (RSU or vehicle) only sends the x-coordinate of the point, and the receiver can acquire the y-coordinate by calculating the square root. Therefore, the size of the (x, y) is reduced by applying this method, and in our scheme, the total communication overhead for a vehicle is about 256 + 256 + 256 + 256 + 256 + 256 + 32 + 32 = 1600 bits = 200 bytes, and for a RSU is

about 32 + 256 + 256 + 256 + 256 + 256 + 256 + 32 + 32 = 1632 bits = 204 bytes. Therefore, the proposed method obtains the smallest communication overhead compared with the other three schemes.

Figure 9 shows the relationship between the communication overhead and the number of received messages. Obviously, the communication costs for RSUs are the smallest for the proposed scheme compared with the other three schemes.

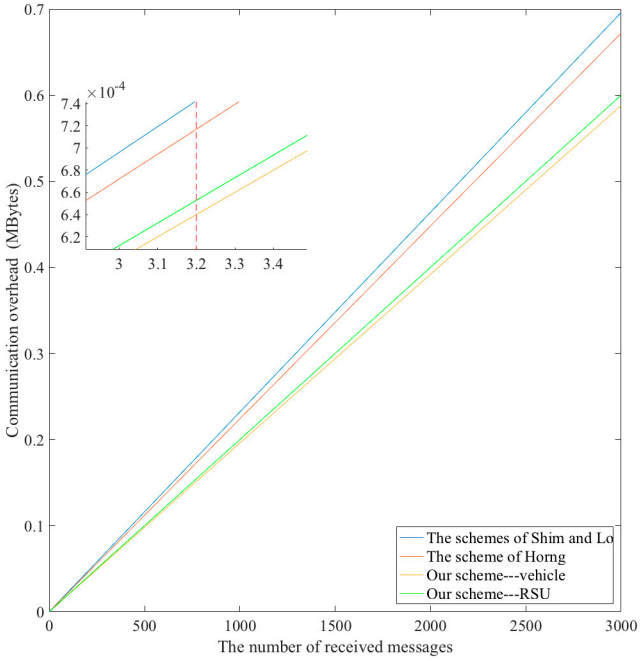

Figure 9. Comparison of the communication overhead.

In summary, the proposed scheme requires a smaller communication bandwidth than the other schemes when it transmits signed messages to other VSN entities.

8. Conclusions

In this work, we have proposed an enhanced secure ID-based, certificateless authentication scheme for VSNs that supports batch verification and conditional privacy-preserving authentication. In addition, the proposed scheme provides compromised-RSU detection and an alarm mechanism, which many related works have not considered. The security analysis shows that the proposed scheme is secure against adaptive chosen message attacks by three types of adversaries under a random oracle. Furthermore, the proposed scheme can resist against major threats like impersonation attacks, node replication attacks, hardware (RSU) tampering attacks, stolen smart card attacks and replay attacks. At last, the scheme can obtain better trade-offs between security and efficiency than other proposed schemes.

In future studies, researchers will focus on different network architectures of VSNs. We will focus on different scenarios in VSNs and consider compatible secure models that can co-exist in heterogeneous networks of VSNs. A designed scheme with better compatibility and scalability will be more suitable for the VSNs.

Acknowledgments: All authors, especially the corresponding author Congcong Li, would like to thank the anonymous reviewers for their time and invaluable comments and suggestions on this paper.

Author Contributions: Congcong Li designed the experiments and wrote the paper with the assistance of Xi Zhang. Haiping Wang performed the experiments, and Dongfeng Li analyzed the data, with assistance of Haiping Wang.

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

References

1. Khatib, N.; Vémola, A. Global status report on road safety. *World Health Organ.* **2015**, *15*, 286.
2. Armstrong, L. Dedicated Short Range Communications (Dsrc) Home. 2002. Available online: http://www.leearmstrong.com/dsrc/dsrchomeset.htm (accessed on 8 January 2018).
3. Std, I. 1609.2-2006-IEEE Trial-Use Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages. 2006. Available online: http://ieeexplore.ieee.org/document/1653011/ (accessed on 8 January 2018).
4. Kaiwartya, O.; Abdullah, A.H.; Cao, Y.; Altameem, A.; Prasad, M.; Lin, C.T.; Liu, X. Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. *IEEE Access* **2016**, *4*, 5356–5373. [CrossRef]
5. Cheng, X.; Wang, C.X.; Laurenson, D.I.; Salous, S.; Vasilakos, A.V. An adaptive geometry-based stochastic model for non-isotropic MIMO mobile-to-mobile channels. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 4824–4835. [CrossRef]
6. Qian, Y.; Moayeri, N. Design of secure and application-oriented VANETs. In Proceedings of the Vehicular Technology Conference, Singapore, 11–14 May 2008; pp. 2794–2799.
7. Qu, F.; Wu, Z.; Wang, F.Y.; Cho, W. A security and privacy review of VANETs. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2985–2996. [CrossRef]
8. Al-Riyami, S.S.; Paterson, K.G. *Certificateless Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 452–473.
9. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
10. Shamir, A. Identity-based cryptosystems and signature schemes. *Lect. Notes Comput. Sci.* **1984**, *21*, 47–53.
11. Gong, P.; Li, P. Further improvement of a certificateless signature scheme without pairing. *Int. J. Commun. Syst.* **2014**, *27*, 2083–2091. [CrossRef]
12. Cao, X.; Kou, W.; Du, X. A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Inf. Sci.* **2010**, *180*, 2895–2903. [CrossRef]
13. Paruchuri, V.; Durresi, A. PAAVE: Protocol for anonymous authentication in vehicular networks using smart cards. In Proceedings of the Global Telecommunications Conference, Miami, FL, USA, 6–10 December 2010; pp. 1–5.
14. Almeida, J.; Shintre, S.; Boban, M.; Barros, J. Probabilistic key distribution in vehicular networks with infrastructure support. In Proceedings of the Global Communications Conference, Anaheim, CA, USA, 3–7 December 2012; pp. 973–978.
15. Calandriello, G.; Papadimitratos, P.; Hubaux, J.P.; Lioy, A. Efficient and robust pseudonymous authentication in VANET. In Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks, Montreal, QC, Canada, 10 September 2007; pp. 19–28.
16. Zhang, C.; Lin, X.; Lu, R.; Ho, P.H.RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks. In Proceedings of the International Conference on Communications, Beijing, China, 19–23 May 2008; pp. 1451–1457.
17. Biswas, S.; Misic, J.; Misic, V. ID-based safety message authentication for security and trust in vehicular networks. In Proceedings of the International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, USA, 20–24 June 2011; pp. 323–331.
18. Chim, T.W.; Yiu, S.M.; Hui, L.C.K.; Li, V.O.K. SPECS: Secure and privacy enhancing communications schemes for VANETs. *Ad. Hoc. Netw.* **2011**, *9*, 189–203. [CrossRef]
19. Horng, S.J.; Tzeng, S.F.; Pan, Y.; Fan, P.; Wang, X.; Li, T.; Khan, M.K. b-SPECS+: Batch verification for secure pseudonymous authentication in VANET. *IEEE Trans. Inf. Forens. Secur.* **2013**, *8*, 1860–1875. [CrossRef]
20. Tsai, J.L.; Lo, N.W. A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Syst. J.* **2015**, *9*, 805–815. [CrossRef]
21. Shim, K.A. An ID-based aggregate signature scheme with constant pairing computations. *J. Syst. Softw.* **2010**, *83*, 1873–1880. [CrossRef]
22. Shim, K.A. An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. *IEEE Trans. Veh. Technol.* **2012**, *61*, 1874–1883. [CrossRef]

23. Dan, B.; Lynn, B.; Shacham, H. Short signatures from the Weil pairing. In Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, 9–13 December 2001; pp. 514–532.
24. Liu, J.K.; Yuen, T.H.; Man, H.A.; Susilo, W. Improvements on an authentication scheme for vehicular sensor networks. *Exp. Syst. Appl. Int. J.* **2014**, *41*, 2559–2564. [CrossRef]
25. Kumar, P.; Kumari, S.; Sharma, V.; Sangaiah, A.K.; Wei, J.; Li, X. A Certificateless aggregate signature scheme for healthcare wireless sensor network. *Sustain. Comput. Inf. Syst.* **2017**. [CrossRef]
26. He, D.; Zeadally, S.; Xu, B.; Huang, X. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Trans. Inf. Forens. Secur.* **2015**, *10*, 2681–2691. [CrossRef]
27. Lo, N.-W.; Tsai, J.-L. An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1319–1328. [CrossRef]
28. Rankl, W.; Effing, W. *Smart Card Handbook*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2004.
29. Mayes, K.E.; Markantonakis, K. *Smart Cards, Tokens, Security and Applications*; Springer: New York, NY, USA, 2008; pp. 519–527.
30. Ding, E.J.; Qiao, X.; Chang, F.; Qiao, L. Improvement of weighted centroid localization algorithm for WSNs based on RSSI. *Trans. Microsyst. Technol.* **2013**, *32*, 53–56.
31. Patwari, N.; Ash, J.N.; Kyperountas, S.; Hero, A.O.; Moses, R.L.; Correal, N.S. Locating the nodes: Cooperative localization in wireless sensor networks. *IEEE Signal Process. Mag.* **2005**, *22*, 54–69. [CrossRef]
32. Pointcheval, D.; Stern, J. Security arguments for digital signatures and blind signatures. *J. Cryptol.* **2000**, *13*, 361–396. [CrossRef]