Hindawi
Mobile Information Systems
Volume 2018, Article ID 4070283, 11 pages
https://doi.org/10.1155/2018/4070283

Research Article

An Efficient V2I Authentication Scheme for VANETs

Yousheng Zhou[1,2,3] **Siling Liu**[1] **Min Xiao**[1,3] **Shaojiang Deng**[2] **and Xiaojun Wang**[4]

[1]*College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*College of Computer Science, Chongqing University, Chongqing 400044, China*
[3]*School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[4]*School of Electronic Engineering, Dublin City University, Dublin, Ireland*

Correspondence should be addressed to Shaojiang Deng; sj_deng@cqu.edu.cn

Received 27 July 2017; Revised 13 November 2017; Accepted 4 December 2017; Published 1 March 2018

Academic Editor: María Calderon

The advent of intelligent transportation system has a crucial impact on the traffic safety and efficiency. To cope with security issues such as spoofing attack and forgery attack, many authentication schemes for vehicular ad hoc networks (VANETs) have been developed, which are based on the hypothesis that secret keys are kept perfectly secure. However, key exposure is inevitable on account of the openness of VANET environment. To address this problem, key insulation is introduced in our proposed scheme. With a helper device, vehicles could periodically update their own secret keys. In this way, the forward and backward secrecy has been achieved. In addition, the elliptic curve operations have been integrated to improve the performance. The random oracle model is adopted to prove the security of the proposed scheme, and the experiment has been conducted to demonstrate the comparison between our scheme and the existing similar schemes.

1. Introduction

Due to the growing demands for a safer and more efficient intelligent transportation system, the development of vehicular ad hoc networks (VANETs) has captured a large amount of attentions from research institutions and industries in recent years. VANETs are deemed to be a variant of the mobile ad hoc network, which is a type of continuously self-configuring, wirelessly connected, and infrastructure-less network of mobile devices [1].

There are two indispensable infrastructure elements in VANETs: on-board units (OBUs), which are mounted in each vehicle, and roadside units (RSUs), which are used to communicate and to assist authentication [2]. In addition, a third trusted party (TA) should also be deployed in VANETs, which mainly provides services of registration and authentication.

A common model of VANETs is exhibited in Figure 1. Communication modes in VANETs could be sorted into two categories: vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication. By employing the dedicated short range communication (DSRC) protocol [3], these dynamic nodes (vehicles) could broadcast and exchange traffic information via RSU and other nearby moving vehicles. Upon receiving those messages including location, speed, and traffic conditions, vehicles would take reasonable actions immediately such as rerouting and braking to avoid possible traffic emergency.

As we all know, the communication channels in VANETs are open, so an attacker could capture, modify, replay, and delete messages transmitted in VANETs easily, leading to a large number of security problems, which will have a strong impact on the whole system. Assume that an original message is actually a warning that there is a serious traffic jam ahead, if it is tampered to a different message which tells vehicles that the road is unblocked, a completely opposite result would be caused. Therefore, authentication between vehicle and infrastructure should be employed to guarantee the authenticity of the transmitted messages in this situation.

Moreover, many devices of infrastructure such as RSU are unmanned, and the units installed on the vehicles which are used to run cryptographic algorithms are resource limited; thus, the risk of key exposure is unavoidable and

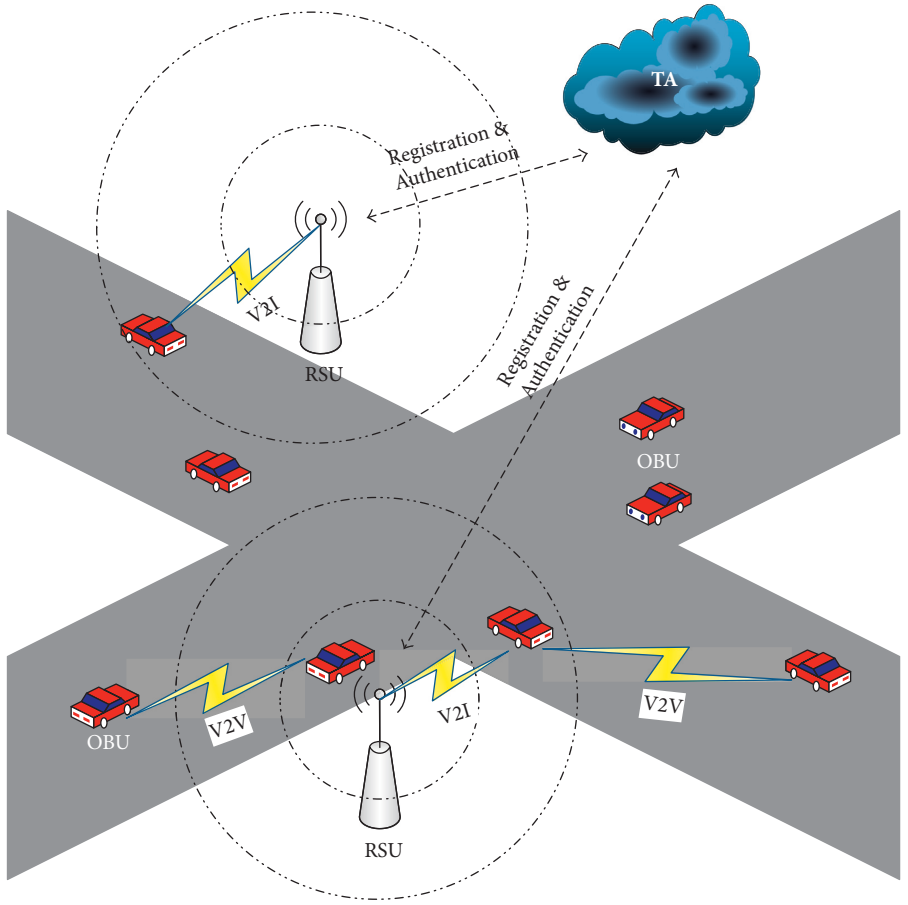

Figure 1: A common model of VANETs.

should not be overlooked. In addition, in the majority of cases, it is much easier for an attacker to obtain a secret key from an insecure device than to get it by breaking cryptogrammic hypothesis which system's security relies on. Once key exposure occurs, it means that security of the whole system loses. Taken into account efficiency, difficulties of construction, and security, key insulation is a desirable method to deal with the key exposure issue.

However, the vast majority of security protocols for VANETs are established on bilinear pairing, which would inevitably cause heavy computation costs. In order to enhance the security and performance of VANET-oriented authentication schemes, a novel practical V2I authentication scheme for VANETs is proposed, which attempts to lower computation complexity and the risk of losing secret keys.

To be specific, the main contributions of this paper represent as follows:

(1) Firstly, the key-insulated method is applied into V2I authentication for VANETs. In our proposed scheme, the user's private key is divided into two portions: one is managed by a secure device called helper or assistant and the other is held by the user, and both of them are updated periodically.

(2) Secondly, ECC, instead of bilinear pairing, is utilized to construct the proposed scheme. As most of devices in VANETs are resource limited, the computation consumption of the adopted schemes should be minimized as much as possible. Operations based on ECC in our construction can save far less time and computation burden than bilinear pairing operations, which is expected to gain higher efficiency.

(3) Finally, the forward and backward secrecy is achieved. The secret key of OBU consists of two fractions in which private information is involved. The secret key must be generated with the helper's participation, and it updates periodically so that malicious attackers cannot obtain the user's private key in the previous periods or in the subsequent periods.

The remainder of this paper proceeds as follows. Section 2 reviews the related work about V2I authentication scheme for VANETs. Section 3 presents the related essential knowledge. Section 4 describes the construction of the proposed scheme in detail. The security analysis and performance evaluation are given in Sections 5 and 6, respectively. Finally, this paper is concluded.

2. Related Works

In recent years, efforts on authentication have been made to address the problems of verification and efficiency. The

privacy-preserving scheme [4] introduced by Wang et al. employs membership validity to replace the certificate revocation list and batch verification to improve efficiency, which achieved nonreputation, anonymity, traceability, and forward and backward secrecy. Wang [5] developed a privacy-preserving and accountable authentication protocol for IoT end-devices by adopting short group signature and secret sharing scheme. Shen et al. [6] proposed a multilayer authentication protocol with session key generation for wireless body area networks which is used for one-to-many group authentication scenario. The scheme with group testing towards a secure batch verification was introduced by Lee and Lai [7]. Unfortunately, this scheme is vulnerable to the impersonation attack since a malicious user could generate a fake signature on behalf of other vehicle. Based on this defect, another secure authentication scheme was introduced by Bayat et al. [8] to improve it. Wang and Yao [9] proposed the LIAP scheme, in which the vehicle and RSU are assigned with a long-term certification from the certificate authority (CA). If the vehicle is compromised, CA could easily revoke the vehicle's long-term certificate to terminate its behavior in the network. Jiang et al. [10] proposed an efficient anonymous batch authentication scheme to replace the CRL checking process by calculating the hash message authentication code, which divides the whole area into several domains. However, every vehicle has stored enough pseudonyms; if any of them is revoked, the rest pseudonyms are wasted. Azees et al. [11] proposed another anonymous scheme to avoid malicious nodes attending the activities in VANETs, which provides conditional tracking mechanism, low-cost certificate, and signature verification. Many secure schemes have achieved authentication by various means; however, most of them adopt bilinear pairing to realize their security characteristics. Actually, the bilinear pairing is not efficient for limited VANET devices on account of its vast computation costs. In view of this, pairing-free schemes have been put forward over the past years. For instance, Cui et al. [12] proposed a privacy-preserving scheme, using cuckoo filter and the binary search methods instead of map-to-point hash function and bilinear pairing operations to achieve high efficiency. Xie et al. [13] proposed an ECC-based authentication scheme to realize reliability and integrity of message. Lo and Tsai [14] proposed an efficient authentication scheme for V2I in vehicular sensor networks without bilinear pairing to improve performance, which achieves message integrity, traceability, and unlinkability. He et al. [15] proposed a new ID-based and elliptic curve-based authentication scheme, which withstands diverse types of attacks and yields better performance.

To address the problem of key exposure, Dodis et al. presented the idea of key insulation and came up with the first key-insulated public key cryptosystem [16] and the first strong key-insulated signature scheme [17]. Following the pioneering works, great efforts have been devoted to the key-insulated signature (KIS) schemes [18–20]. The scheme proposed by Gonzlez-Deleito et al. [18] uses numerous power operations, which adopts multiple private keys and master keys to achieve security. Le et al. [19] utilized multiple certification authorities to shorten verification path and mitigated damage. Hanaoka et al. [20] used two helpers to update the secret key and to enhance the system security. Later, quantities of identity-based or attribution-based key insulation schemes have been proposed [21–27], which are all based on bilinear pairing with random key updating. Additionally, key insulation has been applied into various other research fields. Zhou et al. [28] proposed a certificateless key-insulated generalized signcryption scheme without bilinear pairing in the context of cloud, which is proved to be secure under the computational Diffie–Hellman (CDH) assumption and the elliptic curve discrete logarithm (EC-DL) assumption. Hong et al. [29] proposed a key-insulated attribute-based signature without pairings for wireless communications, which attempts to minimize the potential threat and to relieve the computational burden. Kun et al. [30] and Shi et al. [31] put key insulation into peer-to-peer (P2P) networks and electronic commerce environment, respectively. Moreover, key insulation was introduced into mobile ad hoc networks (MANETs) by V. Kumar and R. Kumar [32]. Park et al. [33] proposed the EA2P scheme and first used key insulation in VANET environment. Although EA2P provides anonymity, identity extraction, and traceability, it only isolates the public key certification, not the private key, which actually fails to achieve a practical sense of key insulation.

3. Preliminaries

In this part, some necessary knowledge including system model, KIS framework, the random oracle model, and discrete logarithm (DL) problem is introduced.

3.1. System Model. As shown in Figure 2, the whole system model in this paper consists of three kinds of entities as follows:

PKG: Private key generator, which is deemed to be fully trusted, is responsible for producing keys including secrete keys as well as public keys.

RSU: Roadside unit, the infrastructure of VANET. It is a kind of computing device located on the roadside, which uses DSRC protocol and provides connectivity support for passing vehicles [34].

Vehicle: Each vehicle is equipped with an on-board unit (OBU) and a tamper-proof device (TPD). OBU is used to help vehicle communicate wirelessly with RSU. TPD acts as a helper which is physically secure but computationally limited, and its stored information can never be disclosed.

There are mainly four procedures in our proposed schemes as shown in Figure 2. Firstly, PKG preloads the related keys into TPD, produces, and publishes system parameters in initialization phase (Phase 1). Secondly, TPD helps the OBU to generate the vehicle's temporary secret key in Phase 2. Then OBU generates the signature and sends it to RSU in Phase 3. Finally, RSU validates the signature in Phase 4.

As illustrated in [15], the formalized definition of key-insulated signature (KIS) displays as follows:

Definition 1 (key-insulated signature (KIS)). A 5-tuple of polynomial time algorithm $(Kgen, UpdD, UpdU, Sig, Ver)$ makes up a key-insulated signature scheme as listed below:

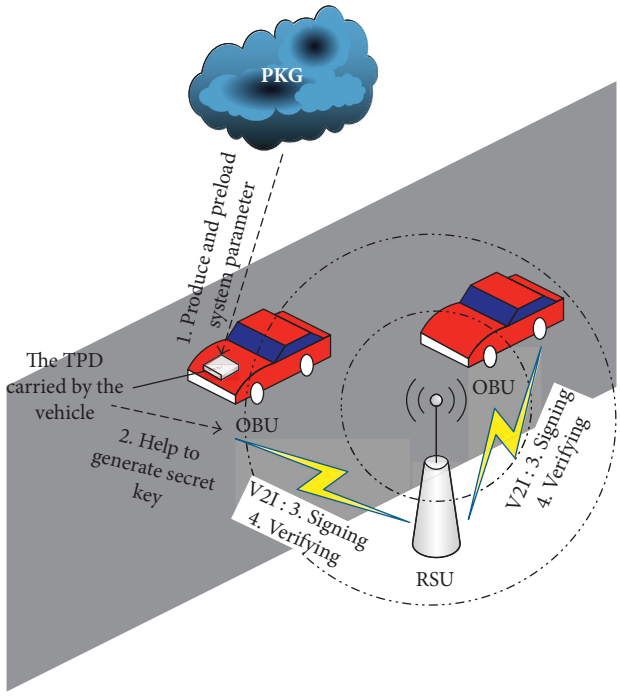

Figure 2: Model of VANET in this paper.

Kgen: the key generation algorithm, which falls into the initialized stage, takes a security parameter 1^k and the total number of time periods N as input to return a public key PK, a master key SK^*, and an initial key SK_0.

UpdD: the key update algorithm for the device, which takes indices i, j for time periods (throughout, $1 \le i, j \le N$) and the master key SK^* as input to return a partial secret $SK'_{i,j}\varepsilon$.

UpdU: the key update algorithm for the user, which takes indices i, j, a secret key SK_i, and a partial secret key $SK'_{i,j}$ as input to return the secret key SK^* for the time period j.

Sign: the signing algorithm, which takes an index i of a time period, a message M, and a secret key SK_i as input. Then $Sign_{SK_i}(i, M)$ returns a signature $< i, s >$ constituting the time period i and a signature S.

Verf: the verification algorithm, which takes the public key PK, a message M, and a pair $< i, s >$ as input. Then $Verf_{PK}(M, < i, s >)$ returns a bit b, where $b = 1$ means that the signature is accepted.

If $Verf_{PK}(M, < i, s >) = 1$, we say that $< i, s >$ is a valid signature of M for the time period i.

3.2. Security Model. The random oracle model was first proposed by Bellare and Rogaway [35] to prove the security of cryptographic protocols, and it is quoted in our proof. Oracle is an external device (it is usually being treated as a theoretical black box) that could provide true outputs for any inputs. In the case of inputting x, running a random oracle could be thought to pick a hash function $h(\cdot)$ at random and outputs $h(x)$. Besides, the relationship between the oracle's output and input satisfies properties of function; namely, the same input corresponds to the same output. As a matter of fact, each output is selected from its output domain, and acquired inputs/outputs are completely independent of current inputs/outputs on account of randomness.

Definition 2 (unforgeability). To prove the unforgeability, a game played between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$ is defined. Our scheme is unforgeable against the malicious OBU if the following condition is satisfied: for any probabilistic polynomial time (PPT) adversary $\mathscr{A}$, the probability that $\mathscr{A}$ wins the following game is negligible. The adversary can adaptively issue a series of undermentioned queries in the game.

Configuration-oracle: this can be considered as an initialization stage. The challenger $\mathscr{C}$ generates the system secret key and public parameters. $\mathscr{C}$ conveys these public parameters to the adversary $\mathscr{A}$.

H_1-*oracle*: upon receiving the query issued by the adversary $\mathscr{A}$ with the message m, the challenger $\mathscr{C}$ picks a stochastic number $e \in Z_q$, puts the tuple (m, e) into the list L_{h_1}, and returns e to the adversary $\mathscr{A}$.

H_2-*oracle*: upon receiving the query issued by the adversary $\mathscr{A}$ with the message m, the challenger $\mathscr{C}$ picks a stochastic number $e \in Z_q$, puts the tuple (m, e) into the list L_{h_2}, and returns e to the adversary $\mathscr{A}$.

TSK-oracle: upon receiving the query issued by the adversary $\mathscr{A}$ on the secret key SK_i, the challenger $\mathscr{C}$ computes it, puts the tuple (m, SK_i) into the list L_{key}, and returns SK_i to $\mathscr{A}$.

H_3-*oracle*: upon receiving the query issued by the adversary $\mathscr{A}$ with the message m, the challenger $\mathscr{C}$ picks a stochastic number $e \in Z_q$, puts the tuple (m, e) into the list L_{h_3}, and returns e to the adversary $\mathscr{A}$.

H_4-*oracle*: upon receiving the query issued by the adversary $\mathscr{A}$ with the message m, the challenger $\mathscr{C}$ picks a stochastic number $e \in Z_q$, puts the tuple (m, e) into the list L_{h_4}, and returns e to the adversary $\mathscr{A}$.

Signing-oracle: the challenger $\mathscr{C}$ generates the message required by the adversary $\mathscr{A}$ and sends it to $\mathscr{A}$.

Verification: the adversary $\mathscr{A}$ inputs the signature $\{U_i, \sigma_i, \lambda_i, M_i, T_i\}$ given by the challenger $\mathscr{C}$, and the verification algorithm returns a bit b, where $b = 1$ means that the signature is valid.

After a polynomial number of queries, if the adversary $\mathscr{A}$ can violate the unforgeability of the proposed scheme by generating a tuple $\{U_i^*, \sigma_i^*, \lambda_i^*, M_i^*, T_i^*\}$ on the condition that the verification phase outputs 1, then we say the adversary $\mathscr{A}$ wins the game.

3.3. Discrete Logarithm (DL) Problem. Provided with two stochastic points P, Q over an elliptic curve E, the DL problem is to compute a number x to meet the equation $Q = x \cdot P$.

4. The Proposed V2I Authentication Scheme

The proposed scheme consists of four phases: system initialization, key generation, signing stage, and verification stage. In the first place, notations used are defined in Table 1.

4.1. System Initialization. In this phase, every appliance in VANETs performs initialization.

(1) PKG generates fundamental system parameters including a group over the chosen elliptic curve $E_p(a,b)$, a random number $SK_{msk} \in Z_q^*$ as the system master key, and the system public key computed as follows:

$$PK_{pub} = SK_{msk} \cdot P. \tag{1}$$

(2) PKG selects a random number $SK_{TPD} \in Z_q^*$ as the private key of the helper (TPD) and calculates its corresponding public key

$$PK_{TPD} = SK_{TPD} \cdot P. \tag{2}$$

All these four parameters should be preloaded into TPD.

(3) RSU selects a random number $SK_{rsu} \in Z_q^*$ as its private key and computes the corresponding public key

$$PK_{rsu} = SK_{rsu} \cdot P. \tag{3}$$

(4) PKG publishes the public parameter set:

$$\text{param} = \{a, b, p, q, P, PK_{pub}, PK_{TPD}, PK_{rsu}, H_1, H_2, H_3, H_4\}. \tag{4}$$

In this paper, we assume that the OBU's public key and its TPD public key have been preloaded.

4.2. Key Generation

4.2.1. Initial Key Generation. Set the parameter α_i corresponding to the time period i as $\alpha_i = H_2(H_1(ID_{obu})SK_{TPD} \cdot PK_{rsu}T_i)$. Note that it is default for TPD to keep its OBU's identity. TPD computes $\alpha_0 = H_2(H_1(ID_{obu})||SK_{TPD} \cdot PK_{rsu}||T_0)$ and the initial private key of the OBU as

$$SK_{obu}^0 = SK_{msk} \cdot H_1(ID_{obu}) + SK_{TPD} \cdot \alpha_0, \tag{5}$$

which is preloaded into the OBU.

4.2.2. Partial Key Generation. TPD calculates

$$K_{part}^i = SK_{TPD} \cdot (\alpha_i - \alpha_{i-1}), \tag{6}$$

as the partial key corresponding to the time period i, and sends it to the OBU to assist in generating the temporary secret key.

4.2.3. Temporary Secret Key Generation. OBU calculates its own temporary secret key in the time period i

$$SK_{obu}^i = SK_{obu}^{i-1} + K_{part}^i, \tag{7}$$

as soon as it receives K_{part}^i from the TPD.

The temporary public key in the time period i of OBU is set as

$$PK_{obu}^i = SK_{obu}^i \cdot P, \tag{8}$$

and it is published by the OBU, while the partial key K_{part}^i and the initial key SK_{obu}^{i-1} are removed after key updating.

4.3. Signing Stage. An OBU can generate the signature on message M_i in the time period i as follows.

Step 1. Selects the random number $u_i \in Z_q^*$ to compute.

$$U_i = u_i \cdot P. \tag{9}$$

Step 2. Uses the identity ID_{obu}, the temporary secret key in the time period i SK_{obu}^i, the public key of RSU PK_{rsu}, the corresponding time stamp T_i, and hash functions to compute

$$\omega_i = H_1(ID_{obu}) \oplus H_3\left(SK_{obu}^i \cdot PK_{rsu}\right), \tag{10}$$

$$\eta_i = H_2.\left(H_1(ID_{obu})SK_{obu}^i \cdot PK_{rsu}T_i\right). \tag{11}$$

Step 3. Selects another random number $\lambda_i \in Z_q^*$ and uses the identity and η_i to compute

$$\theta_i = \eta_i + \lambda_i \cdot H_1(ID_{obu}). \tag{12}$$

Step 4. Concatenates the hash value of identity $H_1(ID_{obu})$, λ_i, U_i, the message about traffic status M_i and current time stamp T_i to compute

$$\beta_i = H_4\left(H_1(ID_{obu})||\lambda_i||U_i||M_i||T_i\right). \tag{13}$$

Step 5. Uses the two random numbers u_i and λ_i, β_i, and the temporary secret key SK_{obu}^i to compute

$$\sigma_i = \lambda_i \cdot SK_{obu}^i + \beta_i \cdot u_i \bmod p. \tag{14}$$

Step 6. Sends the message $\{U_i, \sigma_i, \omega_i, \theta_i, M_i, T_i\}$ to the regional RSU.

4.4. Verification Stage. Upon receiving the signature, RSU proceeds the following steps to verify it.

Step 1. Examines the freshness of T_i. If it is fresh, goes to step 2; otherwise, the signature is rejected.

Step 2. Uses own secret key SK_{rsu}, the private key in the time period i of the vehicle PK_{obu}^i and ω_i to count the hash value of identity of the vehicle:

$$H_1(ID_{obu}) = \omega_i \oplus H_3\left(SK_{rsu} \cdot PK_{obu}^i\right). \tag{15}$$

Table 1: Definition of notations.

Notations	Definition
p, q	Two large prime numbers
$E_p(a,b)$	An elliptic curve defined by the equation $y^2 = x^3 + ax + b \bmod p (a, b \in F_p)$
G	An additive group with the order q, where G is constitutive of all points on E and the point at infinity O
P	A generator of the group G
H_1, H_2, H_3, H_4	Four security functions, where $H_1 : \{0,1\}^* \to Z_q$, $H_2 : \{0,1\}^* \to Z_q$, $H_3 : \{0,1\}^* \to Z_q$ and $H_4 : \{0,1\}^* \to Z_q$
SK_{msk}	The master secret key of system
PK_{pub}	The public key of system
SK_{TPD}	The secret key of assistant device (some papers also called it a base or a helper)
PK_{TPD}	The public key of assistant device
SK_{rsu}	The secret key of RSU
PK_{rsu}	The public key of RSU
T_i	The time of the time period i
ID_{obu}	The identity of OBU
K^i_{part}	The partial key in time period i which is generated by TPD offering assistance to key update of secret key of OBU
SK^i_{obu}	The temporary secret key in time period i of OBU
PK^i_{obu}	The temporary public key in time period i of OBU
σ_i	The signature of OBU
$\oplus$	The exclusive disjunction operation
$\|$	The message concatenation operation

Step 3. Uses the hash value of ID_{obu}, its own secret key SK_{rsu}, the private key of the vehicle, the private key of TPD, and current time stamp T_i to evaluate

$$\alpha_i = H_2\left(H_1\left(ID_{\text{obu}}\right)\left\|SK_{\text{rsu}} \cdot PK_{\text{TPD}}\right\|T_i\right), \tag{16}$$

$$\eta_i = H_2\left(H_1\left(ID_{\text{obu}}\right)\left\|SK_{\text{rsu}} \cdot PK^i_{\text{obu}}\right\|T_i\right). \tag{17}$$

Step 4. Uses the hash value of ID_{obu}, θ_i, and η_i to evaluate

$$\lambda_i = \frac{(\theta_i - \eta_i)}{H_1\left(ID_{\text{obu}}\right)}. \tag{18}$$

Step 5. Concatenates the hash value of ID_{obu}, λ_i, U_i, the message about traffic status M_i, and current time stamp T_i to evaluate

$$\beta_i = H_4\left(H_1\left(ID_{\text{obu}}\right)\left\|\lambda_i\right\|U_i M_i T_i\right). \tag{19}$$

Step 6. Checks whether the equation

$$\sigma_i \cdot P = \left(H_1\left(ID_{\text{obu}}\right) \cdot PK_{\text{pub}} + PK_{\text{TPD}} \cdot \alpha_i\right) \cdot \lambda_i + U_i \cdot \beta_i \tag{20}$$

holds. If it holds, the signature is valid.

5. Security Analysis

In this part, the correctness and the security analysis under the random oracle model of our proposed scheme are illustrated.

5.1. Correctness Proof

Theorem 1. *A signature from the OBU could pass the verification of the RSU.*

Proof. Actually, given a signature $\{U_i, \sigma_i, \omega_i, \theta_i, M_i, T_i\}$ from an OBU, the RSU could compute

$$\begin{aligned}
\sigma_i &= \lambda_i \cdot SK^i_{\text{obu}} + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left(SK^{i-1}_{\text{obu}} + K^i_{\text{part}}\right) + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left(SK^{i-2}_{\text{obu}} + K^{i-1}_{\text{part}} + K^i_{\text{part}}\right) + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left(SK^{i-3}_{\text{obu}} + K^{i-2}_{\text{part}} + K^{i-1}_{\text{part}} + K^i_{\text{part}}\right) + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left(SK^0_{\text{obu}} + K^1_{\text{part}} + K^2_{\text{part}} + \ldots K^i_{\text{part}}\right) + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left[SK^0_{\text{obu}} + SK_{\text{TPD}} \cdot (\alpha_1 - \alpha_0 + \alpha_2 - \alpha_1 + \ldots \alpha_{i-1}\right. \\
&\quad \left. - \alpha_{i-2} + \alpha_i - \alpha_{i-1})\right] + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left[SK^0_{\text{obu}} + SK_{\text{TPD}} \cdot (\alpha_i - \alpha_0)\right] + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left[SK_{\text{msk}} \cdot H_1\left(ID_{\text{obu}}\right) + SK_{\text{TPD}} \cdot \alpha_0\right. \\
&\quad \left. + SK_{\text{TPD}} \cdot (\alpha_i - \alpha_0)\right] + \beta_i \cdot u_i \\
&= \lambda_i \cdot \left[SK_{\text{msk}} \cdot H_1\left(ID_{\text{obu}}\right) + SK_{\text{TPD}} \cdot \alpha_i\right] + \beta_i \cdot u_i,
\end{aligned} \tag{21}$$

$$\begin{aligned}
\sigma_i \cdot P &= \lambda_i \cdot \left[SK_{\text{msk}} \cdot H_1\left(ID_{\text{obu}}\right) \cdot P + SK_{\text{TPD}} \cdot \alpha_i \cdot P\right] + \beta_i \cdot u_i \cdot P \\
&= \left(H_1\left(ID_{\text{obu}}\right) \cdot PK_{\text{pub}} + PK_{\text{TPD}} \cdot \alpha_i\right) \cdot \lambda_i + U_i \cdot \beta_i.
\end{aligned} \tag{22}$$

Therefore, the signature is verified to be valid.

5.2. Security Proof

Theorem 2. *Our proposed V2I authentication scheme is secure under the random oracle model.*

Proof. Assume that there is a PPT adversary $\mathscr{A}$ who could forge a signature to pass the verification successfully. The challenger $\mathscr{C}$ is constructed to tackle the DL problem with a nonnegligible probability by interacting with $\mathscr{A}$. Given a DL instance $(P, PK_{\text{TPD}} = x \cdot P = X)$, the game between $\mathscr{A}$ and $\mathscr{C}$ is played as follows.

5.2.1. Query Phase.

Configuration-oracle: The challenger $\mathscr{C}$ allocates $PK_{\text{pub}} = s \cdot P$ and $PK_{\text{TPD}} = x \cdot P$, generates the public parameter $param = \{a, b, p, q, P, PK_{\text{pub}}, PK_{\text{TPD}}, PK_{\text{rsu}}\}$, and conveys these parameters to the adversary $\mathscr{A}$.

$H_1 - oracle$: A list H_1^{list} is set up and retained by the challenger $\mathscr{C}$, which is initialized to empty. Upon receiving the query about (ID_{obu}, T_i) from the adversary $\mathscr{A}$, $\mathscr{C}$ first examines whether the tuple $(ID_{\text{obu}}, T_i, R_1)$ is in H_1^{list}. If so, $\mathscr{C}$ returns $R_1 = H_1(ID_{\text{obu}})$ to $\mathscr{A}$; otherwise, $\mathscr{C}$ picks a random number $R_1 \in Z_q^*$, puts the tuple $(ID_{\text{obu}}, T_i, R_1)$ into H_1^{list}, and returns $R_1 = H_1(ID_{\text{obu}})$ to $\mathscr{A}$.

$H_2 - oracle$: A list H_2^{list} is set up and retained by the challenger $\mathscr{C}$, which is initialized to empty. Upon receiving the query about $(ID_{\text{obu}}, T_i, D_2)$ from the adversary $\mathscr{A}$, $\mathscr{C}$ first extracts the tuple $(ID_{\text{obu}}, T_i, R_1)$ from the list H_1^{list}. Then, $\mathscr{C}$ examines whether the tuple $(ID_{\text{obu}}, T_i, D_2, R_2)$ is in H_2^{list}. If so, $\mathscr{C}$ returns $R_2 = H_2(R_1||D_2||T_i)$ to $\mathscr{A}$; otherwise, $\mathscr{C}$ picks two random numbers $\omega, R_2 \in Z_q^*$ and allocates $D_2 = \omega \cdot PK_{\text{rsu}}$. Finally, $\mathscr{C}$ puts the tuple $(ID_{\text{obu}}, T_i, D_2, R_2)$ into H_2^{list} and returns $R_2 = H_2(R_1||D_2||T_i)$ to $\mathscr{A}$.

$Temporarysecretkey - oracle$: The adversary $\mathscr{A}$ asks for the temporary private key SK_{obu}^i in the current time period i. The challenger $\mathscr{C}$ first extracts the tuples $(ID_{\text{obu}}, T_i, R_1)$ and $(ID_{\text{obu}}, T_i, D_2, R_2)$ from lists H_1^{list} and H_2^{list}, respectively. Then, $\mathscr{C}$ computes

$$\begin{aligned} SK_{\text{obu}}^i &= SK_{\text{msk}} \cdot H_1(ID_{\text{obu}}) + SK_{\text{TPD}} \cdot \alpha_i \\ &= s \cdot R_1 + SK_{\text{TPD}} \cdot R_2. \end{aligned} \tag{23}$$

Finally, $\mathscr{C}$ adds the tuple $(ID_{\text{obu}}, T_i, SK_{\text{obu}}^i)$ into the list Key^{list} and returns SK_{obu}^i to $\mathscr{A}$.

$H_3 - oracle$: A list H_3^{list} is set up and retained by the challenger $\mathscr{C}$, which is initialized to empty. Upon receiving the query about (ID_{obu}, T_i) from the adversary $\mathscr{A}$, $\mathscr{C}$ first examines whether the tuple $(ID_{\text{obu}}, T_i, R_3)$ is in H_3^{list}. If so, $\mathscr{C}$ returns R_3 to $\mathscr{A}$; otherwise, $\mathscr{C}$ picks a random number $R_3 \in Z_q^*$, puts the tuple $(ID_{\text{obu}}, T_i, R_3)$ into H_3^{list}, and returns R_3 to $\mathscr{A}$.

$H_4 - oracle$: A list H_4^{list} is set up and retained by the challenger $\mathscr{C}$, which is initialized to empty. Upon receiving the query about $(ID_{\text{obu}}, T_i, U_i, \lambda_i, M_i)$ from the adversary $\mathscr{A}$, $\mathscr{C}$ first extracts the tuple $(ID_{\text{obu}}, T_i, R_1)$ from the list H_1^{list}. Then, $\mathscr{C}$ examines whether the tuple $(ID_{\text{obu}}, T_i, U_i, \lambda_i, M_i, R_4)$ is in the list H_4^{list}. If so, $\mathscr{C}$ returns $R_4 = H_4(R_1||U_i||\lambda_i||M_i||T_i)$ to $\mathscr{A}$; otherwise, $\mathscr{C}$ picks a random number $R_4 \in Z_q^*$, puts the tuple $(ID_{\text{obu}}, T_i, U_i, \lambda_i, M_i, R_4)$ into H_4^{list}, and returns $R_4 = H_4(R_1 U_i \lambda_i||M_i||T_i)$ to $\mathscr{A}$.

Signing – oracle: Upon receiving the query about the signature on message M_i from the adversary $\mathscr{A}$, the challenger $\mathscr{C}$ picks random numbers $u_i, \sigma_i \in Z_q^*$ and $r_1, r_2, r_4, d_2 \in Z_q^*$. $\mathscr{C}$ sets $U_i = u_i \cdot P$ and $\lambda_i = (r_1 \cdot P_{\text{pub}} + r_2 \cdot X)^{-1} \cdot (\sigma_i \cdot P - r_4 \cdot U_i)$ and adds the tuples $(ID_{\text{obu}}, T_i, r_1)$, $(ID_{\text{obu}}, T_i, d_2, r_2)$, and $(ID_{\text{obu}}, T_i, U_i, \lambda_i, M_i, r_4)$ into the lists H_1^{list}, H_2^{list}, and H_4^{list} separately. At last, $\mathscr{C}$ returns $\{U_i, \sigma_i, \lambda_i, M_i, T_i\}$ to $\mathscr{A}$.

Verification: The verifier checks if the equation $\sigma_i \cdot P = \lambda_i \cdot (R_1 \cdot PK_{\text{pub}} + PK_{\text{TPD}} \cdot R_2) + U_i \cdot R_4$ holds. If it does not hold, the verification algorithm outputs 0 and the process is aborted. Otherwise, the verification algorithm outputs 1 and the signature is accepted.

5.2.2. Forgery Phase. If an adversary $\mathscr{A}$ could successfully output a signature which can pass the verification with nonneglectable probability according to the forking lemma [36], then $\mathscr{A}$ could output a second signature in an attack by using a different random oracle with nonneglectable probability. Consequently, $\mathscr{A}$ could output another signature $\{U_i^*, \sigma_i^*, \lambda_i^*, M_i^*, T_i^*\}$ by repeating the process with a different choice of R_2, which leads to a distinguishing σ_i, while the value of λ_i remains unchanged. Such that, the following equation is obtained:

$$\begin{aligned} \sigma_i \cdot P &= \lambda_i \cdot \left(R_1 \cdot PK_{\text{pub}} + X \cdot R_2\right) + U_i \cdot R_4 \\ \sigma_i' \cdot P &= \lambda_i \cdot \left(R_1 \cdot PK_{\text{pub}} + X \cdot R_2'\right) + U_i \cdot R_4, \end{aligned} \tag{24}$$

$$\begin{aligned} (\sigma_i - \sigma_i') \cdot P &= \lambda_i \cdot X \cdot (R_2 - R_2{}') \\ x &= (R_2 - R_2')^{-1} \cdot \lambda_i^{-1} \cdot (\sigma_i - \sigma_i'). \end{aligned} \tag{25}$$

At last, the challenger $\mathscr{C}$ outputs $(R_2 - R_2')^{-1} \cdot \lambda_i^{-1} \cdot (\sigma_i - \sigma_i')$ as the answer of the DL problem instance $(P, PK_{\text{TPD}} = x \cdot P = X)$. This contradicts with the difficulty of the DL problem. Hence, our proposed V2I authentication scheme is secure against forgery under the random oracle model on the condition of adaptive chosen message attack.

6. Performance Evaluation

To examine the performance of our proposed scheme in reality, an experiment has been conducted on a Windows 10-installed laptop with Intel(R) Core(TM) i7, and the cryptographic operations have been implemented by using the TEPLA library [37], which requires GMP library and OpenSSL [38].

TEPLA Elliptic Curve and Pairing Library is a free C library which provides functions such as finite field arithmetic with 254-bit prime number, elliptic curve arithmetic over Barreto–Neahrig curve, and pairing arithmetic using optimal ate pairing over BN curve. To set up the system

Table 2: Definition of symbols.

Symbols	Definition
T_{exp}	The running time of an exponentiation operation
$T_{\text{bili-pairing}}$	The running time of a bilinear pairing operation
$T_{\text{bilipair-mul}}$	The running time of multiplication operation with two bilinear pairings
$T_{\text{poi-add}}$	The running time of a point addition operation correlated with the ECC
$T_{\text{poi-mul}}$	The running time of a scale multiplication operation correlated with the ECC
T_h	The running time of a common hash function operation

Table 3: Computational costs comparison of the signature and verification phases in theory (ms).

	Signing	Verification
Wang and Yao [9]	$T_{\text{poi-add}} + 3T_h + 4T_{\text{poi-mul}}$	$2T_h + 3T_{\text{bili-pairing}} + T_{\text{bilipair-mul}}$
Weng et al. [22]	$T_{\text{poi-mul}} + T_h$	$3T_h + 4T_{\text{bili-pairing}} + 2T_{\text{bilipair-mul}}$
Zhou et al. [23]	$2T_{\text{exp}} + T_h$	$3T_h + 4T_{\text{bili-pairing}} + 2T_{\text{bilipair-mul}}$
Wan et al. [24]	$6T_{\text{exp}}$	$5T_{\text{bili-pairing}} + 3T_{\text{bilipair-mul}}$
Zhao et al. [25]	$2T_{\text{poi-mul}} + T_h$	$T_h + 3T_{\text{bili-pairing}} + T_{\text{bilipair-mul}} + T_{\text{poi-mul}} + T_{\text{poi-add}}$
Weng et al. [26]	$6T_{\text{exp}}$	$6T_{\text{bili-pairing}} + 4T_{\text{bilipair-mul}}$
Chen et al. [27]	$7T_{\text{exp}}$	$T_{\text{exp}} + 4T_{\text{bili-pairing}} + 2T_{\text{bilipair-mul}}$
Our scheme	$2T_{\text{poi-mul}} + 4T_h$	$7T_{\text{poi-mul}} + 2T_{\text{poi-add}} + 4T_h$

Table 4: Computational costs comparison of the signature and verification phases in simulation (ms).

	Signing	Verification
Wang and Yao [9]	2.104	4.724
Weng et al. [22]	5.24×10^{-1}	6.305
Zhou et al. [23]	6.3×10^{-2}	6.305
Wan et al. [24]	1.86×10^{-1}	7.882
Zhao et al. [25]	1.047	5.255
Weng et al. [26]	1.86×10^{-1}	9.462
Chen et al. [27]	2.17×10^{-1}	6.333
Our scheme	1.050	3.683

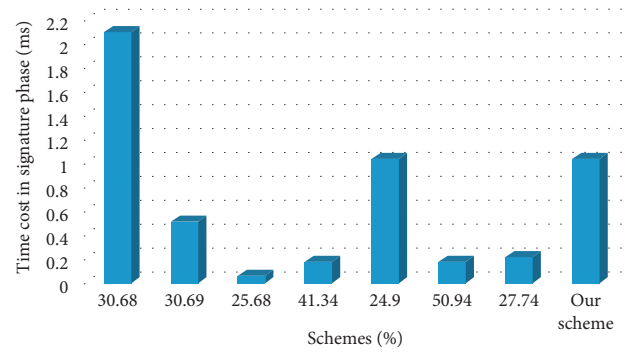

Figure 3: Computational costs of signature phase in different schemes.

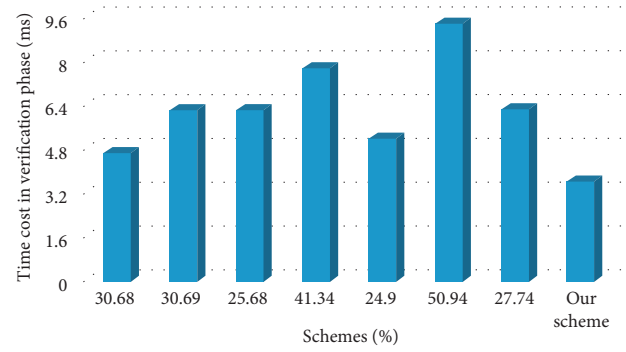

Figure 4: Computational costs of verification phase in different schemes.

environment as the library required, MinGW64 and MSYS are needed to simulate Linux environment to compile cryptography libraries. To install TEPLA, GNU MP library and OpenSSL are required. GNU MP is a free library for big number related operations, and OpenSSL is used to realize cryptographic operations. After finishing compiling, environment variables for Visual Studio are configured and header files of these cryptography libraries are included to conduct the experiment for our scheme.

The experimental results show that the proposed scheme costs 1.050ms and 3.683ms in terms of signing and verification, respectively. We compare our proposed scheme with seven existing similar authentication schemes [9, 22–27], including a pairing-free authentication scheme [9] for VANETs and six key-insulated authentication schemes [22–27]. Note that in our comparison, only consumption of signing phase and verification phase is put into consideration. For convenience, the description of the symbols used in the comparison is listed in Table 2, the results of the comparison on computational costs of various KIS schemes in theory are listed in Table 3, the real running time is displayed in Table 4, and the intuitive comparison about signing and verification is shown in Figure 3 and Figure 4, respectively.

Except the operations listed in Table 2, other operations have not been considered since their running time is ignorable. In terms of the verification phase, according to Table 3, in the scheme of Wang and Yao [9], RSU needs to run two hash operations, three bilinear pairing operations, and one multiplication operation. In the scheme of Weng [22] and Zhou [23], RSU needs to run three hash operations, four bilinear pairing operations, and two multiplication operations. In the scheme of Wan [24], RSU needs to run five bilinear pairing operations, and three multiplication operations. In the scheme of Zhao [25], RSU needs to run one hash operations, three bilinear pairing operations, one multiplication operation, one point addition operation, and one multiplication operation related to ECC. In the scheme of Weng [26], RSU needs to run six bilinear pairing operations and four multiplication operations. In the scheme of Chen [27], RSU needs to run one exponentiation operation, four bilinear pairing operations, and two multiplication operations. As we all know, the bilinear pairing operation is the most time-consuming, while hash function is the least time-consuming. Furthermore, time consumption for the operations listed could be ranked as follows: $T_{\text{bili-pairing}} > T_{\text{poi-mul}} > T_{\text{exp}} > T_{\text{bilipair-mul}} \approx T_{\text{poi-add}} > T_h$. It could be seen from Table 4 that our scheme possesses comparatively high efficiency in verification since our scheme is constructed by using comparatively lightweighted operations. To show the advantage clearly, the improved ratio of our scheme against other seven schemes is defined as $(T_{[\text{num}]} - T_{[\text{ours}]})/T_{[\text{num}]}$, where $T_{[\text{num}]}$ refers to time costs of the scheme with the reference number "num" and $T_{[\text{ours}]}$ refers to time costs of our scheme. In terms of the verification stage, the improved ratios of our scheme against the schemes [9, 22–27] are $(4.724 - 3.683)/4.724 = 22.04\%$, $(6.305 - 3.683)/6.305 = 41.59\%$, $(6.305 - 3.683)/6.305 = 41.59\%$, $(7.882 - 3.683)/7.882 = 53.27\%$, $(5.255 - 3.683)/5.255 = 29.91\%$, $(9.462 - 3.683)/9.462 = 61.08\%$, and $(6.333 - 3.683)/6.333 = 41.84\%$, respectively. When it comes to the total costs of signing and verifying stages, the improved ratios of our scheme against other seven schemes [9, 22–27] are 30.68%, 30.69%, 25.68%, 41.34%, 24.9%, 50.94%, and 27.74% individually. From Figures 3 and 4, the computation costs of singing phase of our proposed scheme are lightly higher than some schemes, because indispensable operations in this phase are needed to achieve key insulation and to provide better security. Even that our promotion comes at a little price of efficiency of signing, the gain of key-insulated secrecy deserves it, and on the whole, the proposed scheme achieves a better trade-off between security and efficiency than the compared schemes.

7. Conclusion

Vehicular ad hoc networks (VANETs) are one of the most promising technologies nowadays. For the sake of providing efficient and secure authentication for VANETs, a key-insulated V2I authentication scheme has been constructed in this paper. The core idea of the proposed scheme is dividing private key of the vehicle into two parts which are, respectively, held by a temper-proofing device (TPD) and the vehicle itself, and these two parts of the private key are used to generate a signature. The proposed scheme supports dynamically updating private key in different time periods. For the vehicle, it obtains its updated secret key by the help of TPD before signing. For the RSU, it first checks whether the time stamp is valid before verification, and then it validates the signature from the vehicle. The security analysis manifests that the proposed scheme is secure under the adaptive chosen message attack. The comparison is also conducted among our scheme and other similar schemes. The performance evaluation shows that our bilinear pairing-free scheme harvests a better trade-off between security and efficiency, and it is feasible for VANET environment.

In our proposed scheme, the helper is assumed as a fully-trusted device, and the private key of the vehicle is generated by its helper. However, the helper is actually semitrusted in some situations, which means that the assistant device can generate signature without the user's approval. In this situation, 2-out-of-2 threshold manner should be a considerable method to prevent the misuse of the user's secret key by the helper. The core idea of 2-out-of-2 threshold manner is that the user and the helper device could share the threshold value n using standard threshold techniques, where the user keeps n_1 and the helper keeps n_2 such that $n_1 + n_2 = n$. In addition, because the RSU would receive and verify numerous signatures from the vehicles within its region, this would inevitably cause burden of computational consumption if the RSU proceeds verification on by one. Taken into account the requirements for efficiency and security in the context discussed above, design of an efficient threshold key-insulated authentication scheme is our future work, which aims to achieve feasible secure V2I communication for VANETs.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was jointly supported by the National Social Science Foundation of China (no. 14CTQ026), the National Natural Science Foundation of China (no. 61702067, no. 61672119, and no. 61472464), the Chongqing Research Program of Application Foundation and Advanced Technology (no. cstc2017jcyjAX0201), the Natural Science Foundation of Shandong Province, China (no. ZR2015FL024), and the Science and Technology Research Project of Chongqing Municipal Education Commission (no. KJ1600445).

References

[1] C. K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Pearson Education, London, UK, 2001.

[2] S. Biswas, J. Mišić, and V. Mišić, "DDoS attack on WAVE-enabled VANET through synchronization," in *Proceedings of the Global Communications Conference (GLOBECOM)*, pp. 1079–1084, Anaheim, CA, USA, December 2012.

[3] H. Oh, C. Yae, D. Ahn, and H. Cho, "5.8 GHz DSRC packet communication system for ITS services," in *Proceedings of the Vehicular Technology Conference*, pp. 2223–2227, Amsterdam, The Netherlands, September 1999.

[4] Y. Wang, H. Zhong, Y. Xu, and J. Cui, "ECPB: efficient conditional privacy-preserving authentication scheme supporting batch verification for VANETs," *International Journal of Network Security*, vol. 18, no. 2, pp. 374–382, 2016.

[5] Z. Wang, "A privacy-preserving and accountable authentication protocol for IoT end-devices with weaker identity," *Future Generation Computer Systems*, 2017, in press.

[6] J. Shen, S. Chang, J. Shen, Q. Liu, and X. Sun, "A lightweight multi-layer authentication protocol for wireless body area networks," *Future Generation Computer Systems*, vol. 78, no. 3, pp. 956–963, 2016.

[7] C. C. Lee and Y. M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Networks*, vol. 19, no. 6, pp. 1441–1449, 2013.

[8] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Networks*, vol. 21, no. 5, pp. 1733–1743, 2015.

[9] S. Wang and N. Yao, "LIAP: a local identity-based anonymous message authentication protocol in VANETs," *Computer Communications*, vol. 112, pp. 154–164, 2017.

[10] S. Jiang, X. Zhu, and L. Wang, "An efficient anonymous batch authentication scheme based on HMAC for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2193–2204, 2016.

[11] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[12] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: a secure privacy-preserving authentication scheme for VANET with cuckoo filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10283–10295, 2017.

[13] Y. Xie, L. B. Wu, J. Shen, and A. Alelaiwi, "EIAS-CP: new efficient identity-based authentication scheme with conditional privacy-preserving for VANETs," *Telecommunication Systems*, vol. 65, no. 2, pp. 229–240, 2017.

[14] N. W. Lo and J. L. Tsai, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1319–1328, 2016.

[15] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[16] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 65–82, Amsterdam, The Netherlands, April-May 2002.

[17] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Strong key-insulated signature schemes," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 130–144, Miami, FL, USA, January 2003.

[18] N. Gonzlez-Deleito, O. Markowitch, and E. Dall'Olio, "A new key-insulated signature scheme," in *Proceedings of the International Conference on Information and Communications Security*, pp. 465–479, Malaga, Spain, October 2004.

[19] Z. Le, Y. Ouyang, J. Ford, and F. Makedon, "A hierarchical key-insulated signature scheme in the CA trust model," in *Proceedings of the International Conference on Information Security*, pp. 280–291, Palo Alto, CA, USA, September 2004.

[20] G. Hanaoka, Y. Hanaoka, and H. Imai, "Parallel key-insulated public key encryption," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 105–122, New York, NY, USA, April 2006.

[21] D. H. Yum and P. J. Lee, "Efficient key updating signature schemes based on IBS," in *Proceedings of the IMA International Conference on Cryptography and Coding*, pp. 167–182, Cirencester, UK, December 2003.

[22] J. Weng, S. Liu, K. Chen, and X. Li, "Identity-based key-insulated signature with secure key-updates," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 13–26, Beijing, China, November-December 2006.

[23] Y. Zhou, Z. Cao, and Z. Chai, "Identity based key insulated signature," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 226–234, Hangzhou, China, April 2006.

[24] Z. Wan, J. Li, and X. Hong, "Parallel key-insulated signature scheme without random oracles," *Journal of Communications and Networks*, vol. 15, no. 3, pp. 252–257, 2013.

[25] H. Zhao, J. Yu, S. Duan, X. Cheng, and R. Hao, "Key-insulated aggregate signature," *Frontiers of Computer Science*, vol. 8, no. 5, pp. 837–846, 2014.

[26] J. Weng, X. Li, K. Chen, and S. L. Liu, "Identity-based parallel key-insulated signature without random oracles," *Journal of Information Science and Engineering*, vol. 24, no. 4, pp. 1143–1157, 2008.

[27] J. Chen, W. Wang, and K. Yu, "Attribute-based threshold key-insulated signature," *Revista Tecnica de la Facultad de Ingenieria Universidad del Zulia*, vol. 39, no. 5, pp. 378–386, 2016.

[28] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Security and Communication Networks*, vol. 2017, Article ID 8405879, 17 pages, 2017.

[29] H. Hong, Y. Xia, and Z. Sun, "Provably secure key insulated attribute based signature without bilinear pairings for wireless communications," *DEStech Transactions on Computer Science and Engineering*, 2016.

[30] H. Kun, D. Xuefeng, and L. Jing, "ID-based key-insulated encryption with message linkages for peer-to-peer network," *Computer Engineering*, vol. 40, no. 4, pp. 124–129, 2014.

[31] Y. Shi, J. Lin, G. Xiong, X. Wang, and H. Fan, "Key-insulated undetachable digital signature scheme and solution for secure mobile agents in electronic commerce," *Mobile Information Systems*, vol. 2016, Article ID 437507, 18 pages, 2016.

[32] V. Kumar and R. Kumar, "An optimal authentication protocol using certificateless ID-based signature in MANET," in *Proceedings of the International Symposium on Security in Computing and Communication*, pp. 110–121, Kochi, India, August 2015.

[33] Y. Park, C. Sur, C. D. Jung, and K. H. Rhee, "Efficient anonymous authentication protocol using key-insulated signature scheme for secure VANET," in *Proceedings of the International Conference on Mobile Lightweight Wireless Systems*, pp. 35–44, Athens, Greece, May 2009.

[34] P. Manzoni, C. T. Calafate, and J. C. Cano, *Encyclopedia of Information Science and Technology*, Information Resources Management Association, New York, NY, USA, 2nd edition, 2009.

[35] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pp. 92–111, Perugia, Italy, May 1994.

[36] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 387–398, Saragossa, Spain, May 1996.

[37] E. Okamoto, N. Kanayama, A. Kanaoka, F. Rodrguez-Henrquez, and T. Teruya, "TEPLA (University of Tsukuba Elliptic Curve and Pairing Library) manual," January 2013. http://www.cipher.risk.tsukuba.ac.jp/tepla/install_e.html.

[38] OpenSSL Management Committee, "Cryptography and SSL/TLS toolkit," January 2017, https://www.openssl.org/.