An Efficient Pairing-free Batch Verification Scheme for VANET

Yan-Ming Lai1, * Pu-Jen Cheng1

1Department of Computer Science and Information Engineering
National Taiwan University
No. 1, Sec. 4, Roosevelt Rd.,
Taipei 10617, Taiwan, R.O.C.
Email: *d05922014, pjcheng@csie.ntu.edu.tw

Abstract—An intelligent traffic system, which can flexibly allocate traffic resources, serves as a good assistant to help us improve traffic safety and efficiency of controlling traffic volume, providing instant traffic information and giving priority to ambulances. Although such system is powerful, it could be misused without proper protection. For example, malicious drivers might forge the message of the ambulance so that they can quickly pass through intersections. In addition, because traffic information is huge and needs to be processed immediately, traditional schemes that process the information one by one are not competent. For this issue, a lot of batch schemes have been proposed. Most of them adopt the algorithm of Bilinear pairing while the others try to avoid it since pairing operations are complex. However, such pairing-free schemes are not applicable because their calculation time will explode when there are more data waiting to be processed. In this article, we briefly describe those schemes and propose a more effective one to solve the problems mentioned above.

Index Term—Ad hoc networks, Elliptic curves cryptography, Intelligent transportation systems, Message authentication, Vehicle.

I. INTRODUCTION

Nowadays, people crowd to metropolitan cities to pursue more job opportunities, living facilities, and better living quality. However, it is a big challenge to maintain the quality of traffic as more and more people move in.

Since the existing urban road nets are hard to be improved, such as widening, increasing, and extending, a smarter traffic scheme is needed to fully utilize the traffic network. A combination of intelligent traffic system (ITS) and Vehicular Ad-Hoc networks (VANET, which also called wireless access in vehicular environments (WAVE)), will be a good assistant. VANET allows vehicles to communicate with other neighboring objects, such as other vehicles or road-side units (RSU), by Dedicated Short Range Communications (DSRC) [1] Among that, RSU helps ITS collect information about local area. With that information, ITS can form an overview of local traffic and appropriately adjust the traffic resources. For example, ITS can flexibly reduce the time of stoplight when there are too many waiting vehicles. Compared with the traditional method, i.e., controlling by manpower, ITS can adjust not only a single intersection but all of them in the area. Single intersection adjustments will sometimes make a huge impact on other intersections, because they will receive additional traffic flows which are not expected. From another viewpoint, man-control can only involve after congestion has occurred. In contrast, ITS can collect traffic data through RSU to predict which road will be crowded and adjust holistic traffic signs appropriately in the early stage. Besides, the collected data can be shared to drivers to help them plan their own paths. During the rush hours, more people tend to avoid the traffic jam in heavy-loaded road sections. In this case, ITS can help arrange traffic overload and keep the region from being impeded.

Although the ITS system looks ideal, it may be misused if the system collects the data unconditionally. For example, a malicious user may generate a lot of signals or repeat expired legal signals to forge a lot of cars waiting for a traffic light. Under this scenario, normal signals will be covered and ITS will misjudge and make wrong resource allocation. To avoid this case, verifying the received data is necessary. However, traditional verification methods face a bottleneck caused by two major features of traffic environment, i.e., immediate demand and huge volumes. Because traffic conditions change rapidly, information loses worth if the execution time of message verification is too long. On the other hand, the number of vehicles under the same area will be very large during the peak traffic hours [2, 3], and the amount of
information will increase accordingly. To overcome the bottleneck, batch verification will be a good solution. In 1989, Fiat proposed a novel scheme based on RSA batch verification [4]. After that, Lin et al. introduce their batch verification for VANET environment based on bilinear pairing in 2007 [5]. Thereafter, other researchers also propose relevant schemes [2, 3, 6]. These schemes are designed to significantly reduce the complex calculation caused by bilinear pairing, i.e., map-to-point operation, point multiplication, and pairing operations. In 2014, Lee and Lai proposed their scheme which further reduces the amount of point multiplication operation to a constant [7]. It means that even though the number of messages waiting for verification becomes larger, the time and cost of the verification protocol still can be kept in a manageable scale. We can say that the scheme has good scalability. However, Lee and Lai’s scheme suffers from some security risks. Tzeng et al. show the weaknesses of Lee and Lai’s scheme and proposed their scheme in 2017 [8].

Although the bilinear pairing algorithm is powerful and easy to use, some researchers think that pairing operating is a complex calculation and thus trying to propose novel schemes without pairing [9-13]. Unfortunately, most of the pairing-free schemes we found have a basic problem - complex calculations will grow linearly as the number of verified information increases, so as required time. Thus, it cannot be called an improvement when the total costs are more, even if it avoids the pairing operation. As a result, to achieve the pairing-free requirement and good scalability, we propose a novel scheme in this article.

Our contributions are shown as follows. First, we analyze the feature of two proposed schemes to tell whether it is good at scalable or not. After that, we propose a novel pairing-free scheme for VANET environment, which satisfies the following security requirements: message authentication, integrity, freshness, user privacy, and traceability. The necessities of those security requirements shall be presented in Section 2-1. Finally, we prove that the the amount of complex point multiplication calculations of our scheme is constant even when verified messages increase. It is also more suitable for batch verification when compared with previous schemes.

Before explaining the proposed scheme, we briefly introduce the used preliminaries including communication model, elliptic curve cryptography (ECC), and bilinear pairing in Section 2. In Section 3 we will talk about schemes which are pairing-free but not scalable. Then, we present the novel scheme we propose in Section 4. After that, we provide a security analysis and a performance analysis of the proposed scheme in Section 5 and Section 6, respectively. Finally, the conclusion will be in Section 7.

II. METHODS

In this section, we introduce the used methods in this paper, including communication model, ECC, and bilinear pairing.

A. Communication model and security requirements

Many previous VANET verification schemes’ communication models are based on a two-layer model [7-9, 13]. The model includes three roles, the trusted authority (TA), road side unit (RSU), and onboard unit (OBU), which is deployed on vehicles and gives vehicles the ability to communicate. In some schemes, there is a tamper-proof device (TPD) in OBU, and those schemes suppose the TPD is ideally secure, i.e. the secure value is safe and the calculation cannot be interfered in TPD [3, 7, 8]. In the top layer, TA is responsible for the system parameter generation, key management, and new entity registration. Most of the time, TA doesn’t communicate with other roles, except key issuance and malicious user response. The bottom layer consists of RSU and OBU, and they exchange messages frequently by DSRC. RSU collects nearby information or provides other services to vehicle users. Vehicles can also commutate with each other.

We classify communication into V2V (Vehicle to Vehicle) and V2R/V2I (Vehicle to RSU or Vehicle to Infrastructure) by its participants. When classified by security requirement, they can be divided into safe and unsafe communication. Notice that unsafe communication is not directly explained by its literal meaning. In general, safe communication is used for transferring private information, such as messages between two drivers. So, it has to ensure the confidentiality of the message. Unsafe communication is used for submitting local information, such as vehicle status or reporting traffic accidents. Contrast with safe communication, those messages’ confidentiality is not important because people can easily observe those states if they are present. Without building a secret channel, which includes protocol deciding, key exchanging, and encrypting/decrypting the message, the time for processing received information can be
shortened. For this reason, unsafe communication is suitable for local information collecting of ITS, and we focus on it in this paper.

However, unsafety communication still needs to keep some security requirements [3, 7-9, 13]:

The first one is message authentication. No authentication mechanism will open the door of ITS for malicious attackers; a well-established authentication mechanism cannot only eliminate malicious attackers but also terminate the legitimate users’ right when they become the evil.

The next is integrity. It means the receiver has the ability to judge if the transferred information has been distorted or not.

The freshness of the received message has the same importance. An undistorted message still misleads ITS if the local state has changed. The sources of out-of-date messages include delayed reception, repeated messages, and messages not processed in time due to inefficient protocol. In addition, during peak traffic periods, it processes very large amounts of information per unit of time. Therefore, WAVE faces more severe problems of system performance compared to other environments.

Next one is to ensure user privacy. Although local information can be revealed to others, the sender’s privacy still needs to be protected.

The following requirement is how to trace the malicious users behind user privacy, and it is also an important requirement. If the system only provides user privacy protection, malicious users will be able to hide and spread a lot of information for their benefits. It can be as small as defrauding the ITS so that it can quickly cross the intersection, or as large as threatening the government and getting a ransom. For the reason, the system needs the ability to catch malicious users and revoke their right. This concept is called conditional privacy. In ITS, the jurisdiction is owned by TA, and reporting malicious messages to TA is the major content between the top and bottom layers.

B. Elliptic curve cryptography

In the field of VANET batch verification, elliptic curve cryptography (ECC) is a recommended cryptosystem [9-11]. Compared with RSA, it only needs the obviously shorter key to achieve the same security strength. An ECC secret key with 160-bits order is considered as strong as a RSA secret key with 1024-bits modulus [14, 15]. The advantage allows ECC to require less transferring cost and calculating time. For this reason, ECC become a popular tool for the novel or new version methods, such as Transport Layer Security (TLS) 1.3 [16].

We slightly present ECC as follows. In 1986 and 1987, Miller [17] and Koblitz [18] proposed their study for using ECC as a crypto tool respectively. In this cryptosystem, it defines an elliptic curve \( E: y^2 = (x^3 + ax + b) \mod q \), where \( a, b \in \{1, q\} \) over a finite field. We let all solutions \((x, y)\) belong to a point set \( G \), and choose a generator \( P \in G \). When operating a point addition, we can obtain a result \( 2P = P + P \), and \( 2P \) is another entity of \( G \). If we continue to do the point addition, the result will come to a point, which is at infinity. We define the infinitely far away point as \( O \), \( O + P = P \), and \( O - P = nP \). When the generator \( P \) adds itself \( n \) times, it comes back to origin \( O \). It means that there are \( n \) entities in the cycle generated by \( P \). It says that the order of \( P \) is \( n \), and the length of \( n \) decides the secret strength of ECC cryptosystem.

Because ECC point multiplication, e.g., \( k \cdot P \), is actually the generator \( P \) adds itself \((k - 1)\) times by point addition for each \( k \in \mathbb{Z}_q \) and point addition is very time-consuming compared to real number additions. Therefore, finding unknown \( k \) form \( k \cdot P \) is very hard if \( n \) is large enough, which is called Elliptic Curve Discrete Logarithm Problem (ECDLP); on the other hand, if we know \( k \), we can use \( 2P = P + P, 4P = 2P + 2P \ldots \) or other faster method to fast obtain \( kP \).

C. Bilinear pairing

Bilinear pairing [19] is good at checking coefficient conversion without a known coefficient in some cryptosystems, such as ECC [19] and ElGamal [20]. Hence, it is popular for designing a scheme which needs to hide its private key from the verifier and the verifier has insufficient permissions for holding the system master key [21]. In this cryptosystem, Weil pairing and Tate pairing are more famous, and we use Weil pairing and ECC to introduce this tool in the following subsection.

When defining a bilinear map \( \hat{e}: G \times G \rightarrow G_T \), we need to prepare two groups \( G \) and \( G_T \). In the two groups, \( G \) is an additive group and \( G_T \) is a multiplicative group, and they have the same order. In this \( \hat{e} \), we can use an ECC point set as \( G \). Here we do not introduce the complete operations of Weil pairing, but explain its properties as follows [3, 7, 8, 19, 22].

1. Bilinear

\[
\forall P, Q, R \in G \quad \hat{e}(Q, P + R) = \hat{e}(Q, P) \cdot \hat{e}(Q, R), \quad \forall P, Q \in G \quad \text{and} \quad \forall a, b \in \mathbb{Z}_q, \hat{e}(aQ, bP) = \hat{e}(bQ, aP) = \hat{e}(Q, P)^{ab}.
\]
2. Non-degenerate
There exist \( P, Q \in G \) such that \( \hat{e}(P, Q) \neq 1_{G_{T}} \), where \( 1_{G_{T}} \) is the identity element of \( G_{T} \).

3. Computable
There is an efficient algorithm to compute \( \hat{e}(P, Q) \) for each \( P, Q \in G \).

The same as ECC, security of the pairing-based scheme is also ensured by hard mathematical problems. There are other well-known hard problems, such as Bilinear Diffie-Hellman problem (BDHP) [21, 22]. However, most of proposed pairing-based schemes in VANET batch verification only transfer the entities of ECC and the pairing calculation is used for verification, because ECDLP is enough of guaranteeing the system security.

III. RELATED WORK
In this session, we briefly introduce two proposed schemes to explain and show the scalability of the scheme. We will present the pairing-based scheme proposed by Tzeng et al.[8] as a positive example. After that, we reveal another scheme that we regard as not suitable for an environment that requires scalability. Please notice that we only briefly describe their schemes, and will not explain how those schemes satisfy their security requirements.

Before presenting those schemes, we define some general conditions.
1. TA selects an elliptic curve \( E \), and a bilinear map \( \hat{e}: G \times G \to G_{T} \) as aforementioned if the scheme is pairing based.
2. We choose two generators \( P, Q \in G \), where \( P \) and \( Q \) have the same order and the descriptions may only use one of them.
3. All used hash functions must satisfy semantic security.
4. Those schemes will be divided into two phases: the key generation and pre-distribution phase and the message signing/verification phase.
5. In following schemes, \( v_i \) is a random integer that is small enough to ignore its calculation cost, such as smaller than 1,000. Its function is to fix the order of the verified message to prevent attackers from exchanging \( S_i \) and \( M_i \) from different received messages in batch verification.

A. Tzeng et al.’s scheme

Tzeng et al.’s scheme was proposed in 2017 [8]. The scheme is pairing-based and scalable.

1. In the key generation and pre-distribution phase

Step 1: TA selects \( x \in Z_{q}^{*} \) as the system’s master key, and computes \( P_{pub} = xP \).

Step 2: TA publishes \( \{ h_{1}(\cdot), h_{2}(\cdot), \{ P, Q, G, P_{pub}, \hat{e} \} \} \), where \( h_{1}(\cdot) \) and \( h_{2}(\cdot) \) are hash functions.

For each vehicle \( V_i \), TA generates its RID and preload \( \{ RID, x \} \) in TPD.

2. In the message signing/verification phase

When \( V_i \) wants to send state of its public message \( M_i \) to RSU in a normal situation, \( V_i \)’s TPD generates \( \{ T_i, AID_{i}, M_i, S_i \} \) with the following steps and sends them to RSU.

Step 1: TPD randomly selects \( r_{1i} \in Z_{q}^{*} \).

Step 2: TPD calculates \( AID_{i} = \{ AID_{i,1} = r_{1i}P, AID_{i,2} = RID \oplus h_{1}(T_i, x) \} \).

Step 3: TPD calculates \( S_i = \left( r_{1i} + vxh_{2}(M_{i} || AID_{i,1} || AID_{i,2} || T_i) \right) Q \).

Upon message reception, RSU can operate single verification or batch verification depending on the amount of message to be verified. The batch verification of Tzeng et al.’s scheme is shown as follows.

\[
\hat{e}(\Sigma v_i S_i, P) = \hat{e}\left( \Sigma v_i AID_{i,1} \right)
+ \left( \Sigma v_i h_{2}(M_{i} || AID_{i,1} || AID_{i,2} || T_i) \right) P_{pub}Q \]

In Tzeng et al.’s scheme, the complex operations of batch verification are two pairing calculations and an ECC point multiplication, \( \left( \Sigma v_i h_{2}(M_{i} || AID_{i,1} || AID_{i,2} || T_i) \right) P_{pub} \). Because the RSU can calculate \( \Sigma v_i h_{2}(M_{i} || AID_{i,1} || AID_{i,2} || T_i) \) before doing point multiplication, the point multiplication only needs to be operated once in total, regardless of the number of verified messages. Finally, the number of complex operations of Tzeng et al.’s scheme will not increase as the number of verified messages grows. Therefore, we can know that the scheme is scalable.

B. Lo and Tsai’s scheme
Lo and Tsai think that pairing operation and MapToPoint operation are not applied in an authentication scheme of VANET because of their computing time, so they proposed their solution in 2016 [9]. In terms of the results, their
method is not more effective if the amount of waiting information is huge.

1. In the key generation and pre-distribution phase

Step 1: TA selects \( x \in \mathbb{Z}_q^* \), and computes \( P_{pub} = xP \).

Step 2: TA publishes \( \{ h_1(\cdot), h_2(\cdot), P, E \} \), where \( h_1, h_2 \) are hash functions.

Step 3: For each vehicle \( (V_i) \), TA generates its \( ID_i \) and secret key \( k_i \in \mathbb{Z}_q^* \), and preload \( \{ ID_i, K_i = k_iP, S_{ID_i} = k_i + xh_1(ID_i, K_i) \mod q \} \) in \( V_i \)'s TPD.

Step 4: In message signing/verification phase

When \( V_i \) wants to send state of its public message \( M_i \) to RSU in a normal situation, \( V_i \)'s TPD generates \( \{ AID_{i,1}, M_i, K_i, ID_i, S_i \} \) with the following steps and sends them to RSU.

Step 1: TPD randomly selects \( r_{1i} \in \mathbb{Z}_q^* \).

Step 2: TPD calculates \( AID_{i,2} = \{ AID_{i,1}, r_{1i}P, AID_{i,2} = RID \oplus h_1(T_i, x) \} \).

Step 3: TPD calculates \( S_i = h_2(K_i||AID_{i,1}||ID_i||M_i) r_{1i} + S_{ID_i} \mod q \).

Notice that there are two schemes in Lo and Tsai’s study. Broadcasted message in one of the schemes doesn’t involve \( ID_i \) while the other one involves a pseudo ID \( PID_i \) in Lo and Tsai’s description. Their scheme cannot operate if \( ID_i \) is unknown. For the convenience of explanation, we integrated their scheme and assume that the broadcasted message involves \( ID_i \).

Upon message reception, RSU can operate single verification or batch verification depending on the amount of message to be verified. The batch verification of Lo and Tsai’s scheme is shown as follows.

\[
(\Sigma v_iS_i)P = ? \Sigma v_i( h_2(K_i||AID_{i,1}||ID_i||M_i) AID_{i,1} + \Sigma v_iK_i + \Sigma (v_ih_1(ID_i, K_i)) P_{pub} 
\]

Lo and Tsai’s scheme doesn’t use any pairing operation indeed. However, the necessary calculation of the verification phase, i.e., \( h_2(K_i||AID_{i,1}||ID_i||M_i) AID_{i,1} \), causes a problem that cannot be ignored. Because \( AID_{i,1} \) and \( h_2(K_i||AID_{i,1}||ID_i||M_i) \) depend on different messages, the verifier cannot summarize them before doing point multiplication operation. According to Miyaji et al.’s study [23] and He et al.’s study [13], the ratio of a point multiplication operation and a pairing operation is about 1:7.5-10.5, which is not a gap that cannot be recovered. It means Lo and Tsai’s will easily lag behind other methods that use a fixed number of pairing operations if there are dozens of messages to be verified. Thus, we know why Lo and Tsai’s scheme is not scalable.

IV. THE PROPOSED SCHEME

Although Lo and Tsai’s scheme is not good at scalability, their viewpoint is commendable. Based on the pairing free idea, we propose our scheme in this section.

The proposed scheme also follows the general conditions of Section 3. We also introduce the proposed scheme from two phases - the key generation and pre-distribution phase and the message signing/verification phase.

1. Key generation and pre-distribution phase

In the key generation and pre-distribution phase, TA selects a random \( x \in \mathbb{Z}_q^* \) as the system’s master key and computes \( P_{pub} = xP \). Then, TA publishes \( \{ h_1(\cdot), h_2(\cdot), P, G, P_{pub} \} \), where \( h_1(\cdot) \) and \( h_2(\cdot) \) are two one-way hash functions, \( h_1(\cdot): G \rightarrow \mathbb{Z}_q^* \), \( h_2(\cdot): \{0, 1\}^* \rightarrow \mathbb{Z}_q^* \).

For each vehicle \( (V_i) \), TA generates its \( RID \) and preloads \( \{ RID, PID, x \} \) in TPD. For each RSU, TA generates its security key \( y_i \) and \( x^{-1}P \). After that, TA transfers \( y_i \) and \( x^{-1}P \) to the RSU via a secure channel. Finally, TA stores each \( RID \) of \( V_i \) and \( y_i \) of RSU in the database.

2. Message signing/verification phase

Different from the previously mentioned schemes, RSU needs to periodically do the following operations.

a. RSU selects a random \( u_j \in \mathbb{Z}_q^* \), computing \( u_j = \mu_j \cdot y_i \mod q \) and \( u_jP, u_jx^{-1}P \).

b. After that, RSU broadcasts the \( u_jP \). RSU can keep the current \( \{ \mu_j, u_jP \} \), denoted as \( \{ \mu_{j,c}, u_{j,c}P \} \), and some previous records, i.e., \( \{ \mu_{j,c-1}, u_{j,c-1}P \}, \{ \mu_{j,c-2}, u_{j,c-2}P \} \) etc., and drop the outdated records.

When \( V_i \) wants to send the state of its public message \( M_i \) to RSU in a normal situation, the following steps are necessary.

Step 1: TPD receives \( \{ u_jP, u_jx^{-1}P \} \) and verifies \( u_jP = ? u_jx^{-1}P \cdot x \).

Step 2: TPD randomly selects \( r_{1i} \in \mathbb{Z}_q^* \).

Step 3: TPD calculates \( AID_{i,1} = \)

Step 4: TPD calculates \( \)

...
\[
\begin{align*}
\text{AID}_{i,2} &= x \cdot r_{i_1} \cdot P, \\
\text{AID}_{i,1} &= \text{RID} \oplus \text{h}_1(x^2 \cdot r_{i_1} \cdot P, u_i P)
\end{align*}
\]

Note that the \( \text{AID}_{i,1} = x \cdot r_{i_1} \cdot P \) calculation can be operated before receiving the broadcasted \( u_i P \). After obtaining the current \( u_i P \), the final step is executed.

Step 5: TPD calculates \( S_i = x \cdot (h_2(\text{AID}_i, M_i, u_i P) + r_{i_1}) \cdot u_i P \). After that, \( V_i \) sends \( \{\text{AID}_i, M_i, u_i P, S_i\} \) to RSU.

Upon message reception, RSU can operate single verification or batch verification depending on the amount of message to be verified. The verification phase of the proposed scheme is shown as follows.

Step 1: RSU checks the \( u_i P \). If the \( u_i P \) is not kept by itself, RSU drops the message.

Step 2: If the amount of verified message is not enough, RSU operates single verification for different \( u_i P \) as follows.

\[
\text{u}_j \cdot (h_2(\text{AID}_i, M_i, u_i P) \cdot P_{pub} + \text{AID}_{i,1}) = S_i
\]

Otherwise, RSU operates batch verification for different \( u_i P \) as follows.

\[
\Sigma V_i S_i = ? u_j \cdot \left( \Sigma V_i h_2(\text{AID}_i, M_i, u_i P) \cdot P_{pub} + \Sigma V_i \text{AID}_{i,1} \right)
\]

The proof of the proposed batch verification is shown as follows.

**Proof:***

\[
\Sigma V_i S_i = \Sigma V_i \cdot x \cdot (h_2(\text{AID}_i, M_i, u_i P) + r_{i_1}) \cdot u_i P
\]

\[
= u_j \cdot x \cdot \Sigma V_i h_2(\text{AID}_i, M_i, u_i P) \cdot x \cdot P + x \cdot u_i P
\]

\[
\cdot \Sigma V_i r_{i_1}
\]

\[
= u_j \cdot \Sigma V_i h_2(\text{AID}_i, M_i, u_i P) \cdot P_{pub} + u_j \\
\cdot \Sigma V_i (x \cdot r_{i_1} \cdot P)
\]

\[
= u_j \cdot \Sigma V_i h_2(\text{AID}_i, M_i, u_i P) \cdot P_{pub} + u_j \\
\cdot \Sigma V_i (\text{AID}_{i,1})
\]

\[
= u_j \cdot \left( \Sigma V_i h_2(\text{AID}_i, M_i, u_i P) \cdot P_{pub} + \Sigma V_i (\text{AID}_{i,1}) \right)
\]

V. **RESULTS: SECURITY ANALYSIS**

In this section, we analyze the proposed scheme mentioned before in respect of security requirements - message authentication, integrity, freshness, user privacy, and traceability. We analyze the security of the proposed scheme as follows and prove that it is secure for VANET.

A. **Message authentication**

We prove that the legal message is sent only from the legal user firstly. Based on the proposed scheme and the adversaries’ ability, we design a game played by an adversary A and the challenger C as \([8, 13]\) to prove the security of the scheme. In this game, we assume that ECDLP can be solved, and C tries to reveal the master key \( x \). There are two oracles in the game as follows.

1. **\( h(.) \)-Oracle:** In this oracle, C maintains a list \( h_k \), which consists of the tuples \((m, y)\) and it is initialized to be emptied. Upon receiving or generating a query \( m_q \), C checks if \((m_q, y_{\text{query}})\) exists in \( h_k \). If \((m_q, y_{\text{query}})\) exists in \( h_k \), C returns \( y_{\text{query}} \) to inquirer; otherwise, C randomly generates a \( y_{\text{query}} \in G \), inserts \((m_q, y_{\text{query}})\) into \( h_k \), and returns \( y_{\text{query}} \) to inquirer.

2. **Sign-Oracle:** Upon receiving a sign query from A, C shall generate a new signature as following steps even if C doesn’t keep master key \( x \).

C randomly selects \( \{s_i, \text{AID}_{i,2}, \text{u}_i, \text{M}_i, \text{h}_i\} \in Z_q \), and if \( h_i \) has existed in \( h_k \), C randomly selects again. After that, C calculates \( \text{AID}_{i,3} = s_i \cdot u_j^{-1} \cdot P - h_i \cdot P_{pub} \), and inserts tuple \((\text{AID}_i, \text{M}_i, u_i P), y = h_i^i \) into \( h_k \). It makes sure that \( \{s_i, \text{AID}_i, \text{M}_i\} \) are a valid signature. The verification of this signature is shown as follows.

\[
h_2(\text{AID}_i, \text{M}_i, u_i P) \cdot u_j \cdot P_{pub} + u_j \cdot \text{AID}_{i,1} = S_i
\]

After that, R returns \( \{s_i, \text{AID}_i, \text{M}_i, u_j\} \) to A.

According to Forking Lemma [24], A can build another valid message by selecting another \( s_i' \in Z_q \) and \((\text{AID}_{i,3}', u_i, \text{M}_i'), y = h_i' \). The following formula is generated by A:

\[
\text{A}' = \text{AID}_{i,3}' \cdot P = u_j \cdot (\text{AID}_{i,3} + h_i' \cdot P_{pub})
\]

And it can also be verified as follows:

\[
h_2(\text{AID}_{i,3}', \text{M}_i, u_i P) \cdot u_j \cdot P_{pub} + u_j \cdot \text{AID}_{i,1} = S_i
\]

According to the above formulas, we can know that

\[
\text{S}_i' - S_i = \text{S}_i' \cdot P - \text{S}_i \cdot P
\]

\[
= (h_2(\text{AID}_{i,3}', \text{M}_i, u_i P) \cdot u_j \cdot P_{pub} + u_j \cdot \text{AID}_{i,1})
\]

\[
- (h_2(\text{AID}_i, \text{M}_i, u_i P) \cdot u_j \cdot P_{pub} + u_j \cdot \text{AID}_{i,1})
\]

\[
= u_j (h_i' \cdot xP + \text{AID}_{i,1} - h_i \cdot xP - \text{AID}_{i,1})
\]

\[
= u_j (h_i' - h_i) xP
\]

Then C outputs \( \frac{(s_i' - s_i)}{u_j (h_i' - h_i)} \) as the answer of \( x \).

However, it is hard to solve the coefficients of ECC, which is according to ECDLP. Another hard problem is generating the match AID_{i,1} and...
In the simulation, we let \( (\text{AID}_i, M_i, u_j P) \), \( y = h_1 \) be an established result, but an ideal cryptography hash is unpredictable. For the reason, the proposed scheme is secure.

According to the above proof, we know that the master key \( x \) cannot be revealed in polynomial time, and the proposed scheme’s security strength is based on this NP problem. Because only the legal user’s TPD keeps the master key \( x \), we can say that a valid signature is surely sent from a legal user. Note that an adversary may try to broadcast his/her \( u_i P \) to feign a RSU, but the attack is worthless. Each message is in plaintext, and the adversary can easily receive them if he/she is really interested in that information.

### B. Integrity

In the proposed scheme, \( u_j \) prevents the adversary from making a fake signature at will, and \( r_{1j} \) provides further protection. If it is without \( r_{1j} \), i.e., \( S_t = x \cdot \left( h_2(\text{AID}_i, M_i, u_j P) \right) \cdot u_j P \), the adversary can use the following formula to generate a legal signature without the master key:

\[
S_t^* = S_t \cdot \left( h_2(\text{AID}_i, M_i, u_j P) \right)^{-1} \cdot \left( h_2(\text{AID}^*_i, M^*_i, u_j P) \right)
\]

And the proposed scheme resists this attack now.

### C. Freshness

In the proposed scheme, there is no timestamp. Instead, time-limited parameter \( u_j P \) ensures the freshness of the transferred message, and the immediate advantage of the design is avoiding the synchronize problem. Although, it cannot resist the replay attack in the same time interval, the same problem also exists in the timestamp-based scheme. On the other hand, if the replay attack can only affect the system in the same time interval, RSU can resist it by holding the received messages and drop the stale received messages to reduce the cost. Overall, the time-limited parameter is better than timestamp-based scheme.

### D. User privacy

Even though the transferred messages are plaintext, the system still needs to hold user privacy, because it is personal information and it may cause damages if an adversary gets it.

To protect the privacy of the user, the scheme should possess two properties - user anonymity and message unlink-ability. The user anonymity means the third party cannot obtain the real identity from transferred messages. However, it is not enough because the adversary can find the relationship from different messages sent from the same user and take them as target. The adversary may not know the relation between messages and the user, but he/she can always find the target user after comparing the collected information with the tracking information in real world. In the proposed scheme, there are two parameters which provide protection by frequent value changing. The first is the random parameter \( r_{1j} \), which is different between each message even from the same \( V_i \). The second is time-limited \( u_j P \), which is changed over time. In addition, there isn’t any fixed information in the transferred message. For the above reasons, the third party can’t link different messages sent from the same user, hence achieving real user privacy in the proposed scheme.

### E. Traceability

In the proposed scheme, TA can trace an anonymous user \( V_{m} \) by the following formula if a RSU reports the misleading information \( \{\text{AID}_m, M_m, y_j, u_j P, S_m\} \) and the corresponding \( \mu_j \).

Upon message reception, TA verifies \( y_j \) and \( u_j P \neq \mu_j \cdot y_j \cdot P \) firstly for ensuring that the report is sent from a legal RSU. After that, TA obtains the real ID \( \text{RID}_m \) as the following formula:

\[
\text{RID}_m = \text{AID}_m \Theta h_1(x \cdot \text{AID}_m, u_j P)
\]

Afterward, TA shall inform the management for proper legal processing. In the proposed scheme, we add a design which lets TA verify the reporter for resisting a compromised RSU attempting to frame innocent user. If RSU always reports innocent users, TA can find the RSU by \( y_j \) and revoke its permission.

### VI. RESULTS: PERFORMANCE ANALYSIS

In this section, we analyze the performance of the proposed scheme. According to He et al.’s study [13], we list the following operations and show the execution time for each of them in Table 1.

1. \( T_{bp} \): the execution time of a bilinear operation \( \hat{e}(Q, P) \), where \( P, Q \in G \).
2. \( T_{sm-bp} \): the execution time of an ECC point multiplication \( xP \) related to the bilinear pairing, where \( x \in \mathbb{Z}_n \) and \( P \in G \).
3. \( T_{sm-bp-s} \): the execution time of a simple ECC point multiplication \( v_i P \) related to the bilinear pairing, where \( v_i \in Z_{p^2} \), \( P \in G \), and \( v_i \) is a small integer.

4. \( T_{pa-bp} \): the execution time of an ECC point addition \( P + Q \) related to the bilinear pairing, where \( P, Q \in G \).

5. \( T_{h} \): the execution time of a hash-to-point \( h(*) \rightarrow P \).

6. \( T_{sm-ecc} \): the execution time of an ECC point multiplication \( xP \), where \( x \in Z_n \) and \( P \in G \).

7. \( T_{sm-ecc-s} \): the execution time of a simple ECC point multiplication \( v_i P \), where \( v_i \in Z_{p^2} \), \( P \in G \), and \( v_i \) is a small integer.

8. \( T_{pa-ecc} \): the execution time of an ECC point addition \( P + Q \), where \( P, Q \in G \).

Note that the point operations are classified into \{\( T_{sm-bp} \), \( T_{sm-bp-s} \), \( T_{pa-bp} \)\} and \{\( T_{sm-ecc} \), \( T_{sm-ecc-s} \), \( T_{pa-ecc} \)\}, because we need to choose the special elliptic curve to satisfy the conditions of bilinear operations. The execution time of the other used operations, e.g., hash and multiplication operations, can be ignored because they are too small.

After obtaining the execution times of different operations, we can compare the performance of the proposed scheme with other related, recently proposed schemes. Note that we focus on the verification step because the step needs the most execution time and is the most influential when the count of the verified message becomes more. When calculating the operations, we shall try to reduce the time-consuming steps as much as possible. Table 2 shows the optimized steps of the batch verification of the proposed scheme, which let the execution time become the shortest with the same result. Note that the “ignored” in the cumulative calculation time column means that the previous steps use only simple operations, such as one-way function, integer addition, and integer multiplication, etc. The time of those executions are small enough to be ignored.

Finally, all execution times of the verification of the proposed scheme is

\[
2T_{sm-ecc} + (2m)T_{sm-ecc-s} + (2m - 1)T_{pa-ecc} \approx 0.8422 + 0.0588m
\]

The unit is milliseconds (ms).

Except the schemes presented in related works, we also compare four other schemes: Asaar et al.’s scheme [10], Cui et al.’s scheme [11], Gayathri et al.’s scheme [12], Sutrala et al.’s scheme [25] and Wu et al.’s scheme [26]. Note that their schemes also use other ideas, such as proxy vehicles, Cuckoo filter, and certificate-less, to solve other relevant situations. We only compare the batch verification phase of those schemes.

Table 3 shows the necessary operators and execution time of different schemes when verifying \( m \) messages. Figure 1 shows the line chart, which presents the trend of \( m \) from 1 to 110. According to the table, the proposed scheme has the least expansion coefficient 0.0588m. Compared with the second-ranked one of the pairing-free scheme, Cui et al.’s scheme’s 0.4438m [11], the proposed scheme has an expansion rate of only 13%. It means that the proposed scheme can verify more messages in the same time and will not be outdated. On the other hand, the proposed scheme also has the shortest execution time of single verification. For the above reasons, we claim the proposed scheme is the most efficient. [25]

VII. CONCLUSIONS AND DISCUSSION

We propose an efficient batch verification scheme for VANET in this paper. Different from the previous schemes, the proposed scheme not only achieves pairing-free, but also keeps the amount of the complex operations a constant even if the number of verified messages is large. In addition to that, we also prove that the proposed scheme holds the following security requirements: message authentication, integrity, freshness, user privacy, and traceability via the security analysis. By evaluating other schemes, the results show that the proposed scheme has the shortest execution time of verification phase in both of signal verification and batch verification. Therefore, we prove that the proposed scheme has the advantage in this field of batch verification.

In the future, we can extend the work. Some studies suggest that TPD still has potential risks even if we all hope that it is secure. If the TPD were broken, the master key would be retrieved by the adversary. Maybe the proposed scheme can be modified to a scheme which doesn’t rely on TPD and still keeps effectiveness, such as using pairing or letting TA generate pseudo-ID lists for each vehicle. On the other hand, batch verification always needs to face a problem: illegal messages discrimination. If one or more illegal message is hidden in the batch verified messages, it will make the verification fail. Therefore, distinguishing illegal messages from
all received messages is a crucial task for the verifier. It is a daunting problem related to this work, and we will challenge it in the near future.

Abbreviations

BDHP: Bilinear Diffie-Hellman problem
DSRC: Dedicated Short Range Communications
ECC: Elliptic curve cryptography
ECDLP: Elliptic Curve Discrete Logarithm Problem
ITS: Intelligent traffic system
OBU: Onboard unit
RSU: Road-side units
TA: Trusted authority
TLS: Transport Layer Security
TPD: Tamper-proof device
V2I: Vehicle to Infrastructure
V2R: Vehicle to RSU
V2V: Vehicle to Vehicle
VANET: Vehicular Ad-Hoc networks
WAVE: Wireless access in vehicular environments

Authors Contribution

Y.-M. Lai: providing the major idea, designing the protocol, designing analysis methods, and writing the manuscript
P.-J. Cheng: Advising the detail idea’s completeness, article structure, text smooth, and grammatical modification.

Availability of data and materials

Open Access This article is published under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Competing interests

The authors declare that they have no competing interests.

References

[1] R. Bera, J. Bera, S. Sil, S. Dogra, N. B. Sinha and D. Mondal, (2006). Dedicated short range communications (DSRC) for intelligent transport system. 2006 IFIP International Conference on Wireless and Optical Communications Networks, https://doi.org/10.1109/WOCN.2006.1666607.
[2] J.-L. Huang, L.-Y. Yeh and H.-Y. Chien. (2011). ABAKA: An Anonymous Batch Authenticated and Key Agreement Scheme for Value-added Services in Vehicular Ad Hoc Networks. IEEE Transactions on Vehicular Technology, 60(1), 248-262.
[3] C. Zhang, P.-H. Ho and J. Tapolcai. (2011). On Batch Verification with Group Testing for Vehicular Communications. Wireless Networks, 17(8), 1851-1865.
[4] A. Fiat. (1990). Batch RSA. Cryptology -- - CRYPTO' 89 Proceedings, https://doi.org/10.1007/s001459900021.
[5] X. Lin, X. Sun, P.-H. Ho and X. Shen. (2007). GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications. IEEE Transactions on Vehicular Technology, 56(6), 3442 - 3456.
[6] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li and M. K. Khan. (2013). B-SPECS+: Batch Verification for Secure Pseudonymous Authentication in VANET. IEEE Transactions on Information Forensics and Security, 8(11), 1860-1875.
[7] C.-C. Lee and Y.-M. Lai. (2013). Toward a Secure Batch Verification with Group Testing for VANET. Wireless Networks, 19(6),1441-1449.
[8] S.-F. Tzeng, S.-J. Horng, T. Li, X. Wang and P.-H. Huang. (2017) Enhancing Security and Privacy for Identity-Based Batch Verification Scheme in VANETs. IEEE Transactions on Vehicular Technology, 66(4), 3235-3248.
[9] N.-W. Lo and J.-L. Tsai. (2016). An Efficient Conditional Privacy-Preserving Authentication Scheme for Vehicular Sensor Networks Without Pairings. IEEE Transactions on Intelligent Transportation Systems, 17(5), 1319-1328.
[10] M. R. Asaa, M. Salmaszadeh, W. Susilo and A. Majidi. (2018). A Secure and Efficient Authentication Technique. IEEE Transactions on Vehicular Technology, 67(6), 5409-5423.
[11] J. Cui, L. Wei, J. Zhang, Y. Xu and H. Zhong. (2019). An Efficient Message-Authentication
Scheme Based on Edge Computing for Vehicular Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(5), 1621-1632.

[12] N. B. Gayathri, G. Thumber, P. V. Reddy and M. Z. U. Rahman. (2018). Efficient Pairing-Free Certificateless. *IEEE Access*, 6, 31808-31819.

[13] D. He, S. Zeadally, B. Xu and X. Huang. (2015). An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks. *IEEE Transactions on Information Forensics and Security*, 10(12), 2681-2691.

[14] E. Barker, D. Johnson and M. Smid. (2013). Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. https://doi.org/10.6028/NIST.SP.800-56Ar3.

[15] R. Schroeppel, H. Orman, S. O’Malley and O. Spatscheck. (1995). Fast Key Exchange with Elliptic Curve Systems. *Cryptology --- CRYPTO ’95*. https://doi.org/10.1007/3-540-44750-4_4

[16] E. Rescorla. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. Internet Engineering Task Force. https://tools.ietf.org/html/rfc8446.

[17] V. S. Miller. (1986). Use of Elliptic Curves in Cryptography. *Conference on the Theory and Application of Cryptographic Techniques*, https://doi.org/10.1007/3-540-39799-X_31.

[18] N. Koblitz. (1987). Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177), 203-209.

[19] A. Menezes. (1991). An Introduction to Pairing-Based Cryptography. *Mathematics Subject Classification*. Primary 94A60. https://www.math.uwaterloo.ca/~jmenez/papers/pairings.pdf.

[20] N. Lan. (2005). Accumulators from Bilinear Pairings and Applications. *Topics in Cryptology --- CT-RSA 200*. https://doi.org/10.1007/978-3-540-30574-3_19.

[21] S. Zhao, A. Aggarwal, R. Frost and X. Bai. (2012). A Survey of Applications of Identity-Based Cryptography in Mobile Ad-Hoc Networks. *IEEE Communications Surveys & Tutorials*, 14(2), 380-400.

[22] R. Dutta, R. Barua and P. Sarkar. (2004). Pairing-Based Cryptographic Protocols : A Survey. https://eprint.iacr.org/2004/064.pdf.
### Table 1. The execution time of operations

| Step | Calculation | Unit: ms |
|------|-------------|---------|
| 1    | $T_{bp}$    | 4.211   |
| 2    | $T_{sm-bp}$ | 1.709   |
| 3    | $T_{sm-bp-s}$ (t = 10) | 0.1068 |
| 4    | $T_{pa-bp}$ | 0.0041  |
| 5    | $T_{mtp}$   | 4.406   |
| 6    | $T_{sm-ecc}$ | 0.442  |
| 7    | $T_{sm-ecc-s}$ (t = 10) | 0.0276 |
| 8    | $T_{pa-ecc}$ | 0.0018  |

### Table 2. The cumulative calculations time of the proposed scheme

| Step | Calculation | Cumulative calculation time |
|------|-------------|----------------------------|
| 1    | $h_2(AID, M, u_1P)$ for m message | Ignored |
| 2    | $v_1h_2(AID, M, u_1P)$ for m message | Ignored |
| 3    | $\Sigma v_1h_2(AID, M, u_1P)$ | Ignored |
| 4    | $\Sigma v_1h_2(AID, M, u_1P) \cdot P_{pub}$ | $1T_{sm-ecc}$ |
| 5    | $v_1(A_{i,1})$ for m message | $1T_{sm-ecc} + (m)T_{sm-ecc-s}$ |
| 6    | $\Sigma v_1(A_{i,1})$ | $1T_{sm-ecc} + (m)T_{sm-ecc-s} + (m - 1)T_{pa-ecc}$ |
| 7    | $\Sigma v_1h_2(AID, M, u_1P) \cdot P_{pub} + \Sigma v_1(A_{i,1})$ | $1T_{sm-ecc} + (m)T_{sm-ecc-s} + (m)T_{pa-ecc}$ |
| 8    | $u_1 \cdot (\Sigma v_1h_2(AID, M, u_1P) \cdot P_{pub} + \Sigma v_1(A_{i,1}))$ | $2T_{sm-ecc} + (m)T_{sm-ecc-s} + (m)T_{pa-ecc}$ |
| 9    | $v_1S_i$ | $2T_{sm-ecc} + (2m)T_{sm-ecc-s} + (m)T_{pa-ecc}$ |
| 10   | $\Sigma v_1S_i$ | $2T_{sm-ecc} + (2m)T_{sm-ecc-s} + (2m - 1)T_{pa-ecc}$ |
Table 3. The execution time of comparison schemes

| Scheme | Batch verification for \( m \) message | Execution time (ms) | Single (\( m=1 \)) |
|--------|---------------------------------------|---------------------|------------------|
| Tzeng et al.’s scheme [8] | \((2)T_{bp}
+ (1)T_{sm-bp}
+ (2m)T_{sm-bp-s}
+ (2m - 1)T_{pa-bp}\) | \(\approx 10.1269 + 0.1109m\) | \(\approx 10.2378\ ms\) |
| Lo and Tsai’s scheme [9] | \((m + 2)T_{sm-ecc}
+ (m)T_{sm-ecc-s}
+ (3m - 1)T_{pa-ecc}\) | \(\approx 0.8422 + 0.475m\) | \(\approx 1.3172\ ms\) |
| Asaar et al.’s scheme [10] | \((m + 1)T_{sm-ecc}
+ (m)T_{sm-ecc-s}
+ (m - 1)T_{pa-ecc}\) | \(\approx 0.4402 + 0.4714m\) | \(\approx 0.9916\ ms\) |
| Cui et al.’s scheme [11] | \((m + 2)T_{sm-ecc}
+ (m)T_{pa-ecc}\) | \(\approx 0.884 + 0.4438m\) | \(\approx 1.3278\ ms\) |
| Gayathri et al.’s scheme [12] | \((3m + 1)T_{sm-ecc}
+ (3m)T_{pa-ecc}\) | \(\approx 0.442 + 1.3314m\) | \(\approx 1.7734\ ms\) |
| Sutrala et al.’s scheme [25] | \((3m)T_{sm-ecc}
+ (4m)T_{pa-ecc}
+ (m)T_{sm-ecc-s}\) | \(\approx 1.3608m\) | \(\approx 1.3608\ ms\) |
| Wu et al.’s scheme [26] | \((m + 2)T_{sm-ecc}
+ (m)T_{pa-ecc}\) | \(\approx 0.884 + 0.4438m\) | \(\approx 1.3278\ ms\) |
| The proposed scheme | \((2)T_{sm-ecc}
+ (2m)T_{sm-ecc-s}
+ (2m - 1)T_{pa-ecc}\) | \(\approx 0.8422 + 0.0588m\) | \(\approx 0.901\ ms\) |
Figure 1

*x-coordinate: the number of verified messages (1 to 100)
*y-coordinate: execution time of verified messages (unit: ms)

Figure 1. Execution time for m verified messages