A Secure and Efficient Authentication Technique for Vehicular Ad-Hoc Networks

Maryam Rajabzadeh Asaar, Mahmoud Salmasizadeh, Willy Susilo*, *Senior Member, IEEE,* Akbar Majidi

***Abstract*—Vehicular ad-hoc networks (VANETs) have been emerging due to the recent technologies in wireless and network communications. The most fundamental part in VANETs is to enable message authentications between vehicles and roadside units. Message authentication using proxy vehicles has been proposed to reduce the computational overhead of roadside units significantly. In this type of message authentication schemes, proxy vehicles with verifying multiple messages at the same time improve computational efficiency of roadside units when there are a large number of vehicles in their coverage areas. In this paper, first we show that the only proxy-based authentication scheme (PBAS) presented for this goal by Liu et al. cannot achieve authenticity of messages, and also it is not resistant against impersonation and modification attacks and false acceptance of batching invalid signatures. Next, we propose a new identity-based message authentication using proxy vehicles (ID-MAP). Then, to guarantee that it can satisfy message authentication requirement, existential unforgeability of underlying signature against adaptively chosen-message and identity attack is proved under Elliptic Curve Discrete Logarithm Problem in the random oracle model. It should be highlighted that ID-MAP not only is more efficient than PBAS since it is pairing-free and does not use map-to-point hash functions, but also it satisfies security and privacy requirements of vehicular ad hoc networks. Furthermore, analysis shows that the required time to verify 3000 messages in ID-MAP is reduced by 76% compared to that of PBAS.**

***Index Terms*—proxy vehicles, authentication, privacy preserving, vehicular ad-hoc network.**

I. Introduction

In the last few years, the vehicular ad hoc network (VANET) has been emerged due to the advances in wireless communications and networking technologies [1]–[3]. The VANETs improve traffic safety and efficiency. For communications in VANETs, each vehicle has a wireless communication device named as an on board unit (OBU), and a wireless communication protocol named as dedicated short range communication (DSRC), which applies the IEEE 802.11p standard for wireless communication, and is used for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications.

Because of the wireless communication mode, it is easy for an adversary to take control of communication links and can change, delete and replay messages. Hence, the impersonation, modification, replay and man in the middle attacks are serious threats for VANETs. These threats may lead to traffic chaos or accident [4], [5]. Therefore, security of transmitted messages is one of the main requirements in VANETs. In addition, privacy of the vehicle's identity must be achieved since leakage of their identities may result in serious threats for drivers since malicious entities can trace their messages and traveling roads for crimes [6]. However, unconditional privacy preserving is not desirable for VANETs, since malicious vehicles should be traced and punished in case of any misbehavior [7], [8].

To satisfy security and privacy issues in VANETs, some Public Key Infrastructure-based (PKI-based) authentication schemes [4], [6] have been proposed. These schemes are not efficient since vehicles need to store a large number of key pairs and their corresponding certificates, and these certificates are required to be transmitted with messages. To address certificate management in PKI-based authentication schemes, various privacy preserving identity-based authentication schemes [8]–[15] have been proposed. These authentication schemes are designed based on bilinear pairings and due to their heavy computational cost, recently two efficient authentication schemes by Lo and Tsai [16] and He et al. [17] have been proposed. In fact, they proposed identity-based signatures without employing bilinear pairings to improve performance of these schemes. However, these schemes are not enough fast when there are a large number of messages in the coverage area of a roadside unit (RSU). For example, consider this scenario: since each vehicle broadcasts its traffic safety message every 100-300 milliseconds according to the specification of DSRC protocol, when there are 500 vehicles in the coverage area of an RSU, the RSU has to verify around 2500-5000 signatures in a second. This issue is a big challenge for the current authentication schemes [16]–[18] as stated by Liu et al. [19] in 2015. To tackle the aforementioned problem, Liu et al. [19] proposed an interesting authentication protocol using proxy vehicles for vehicular networks, and called it as PBAS. In PBAS, proxy vehicles help RSUs to verify a large number of signatures simultaneously using distributed computing. In fact, Liu et al. [19] claimed that in their proposal the time required to verify 3000 signatures is decreased by 88% compared to previous efficient authentication schemes based on batch verification method at RSUs.

M. R. Asaar is with Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
(asaar@srbiau.ac.ir)
M. Salmasizadeh is with Electronics Research Institute of Sharif University of Technology, Tehran, Iran.
(salmasi@sharif.edu)
W.Susilo is with Centre for Computer and Information Security Research, School of Computing and Information Technology University of Wollongong, Wollongong, Australia
(wsusilo@uow.edu.au)
Corresponding Author: Willy Susilo. Phone: +61-2-4221-5533. Fax: +61-2-4221-5550.
A. Majidi is with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.
(majidi@sjtu.edu.cn)

A. Our contributions

Contributions of this paper are as follows.

- First, we show that the proxy-based authentication scheme (PBAS) for vehicular networks presented by Liu et al. [19] is not resistant against false acceptance of invalid signatures sent by vehicles, and also it does not have message authentication which is the main requirement of authentication schemes for VANETs. Consequently, we show that it is not secure against impersonation and modification attacks.
- Second, to tackle the aforementioned problems and have a more efficient scheme, a new identity-based authentication scheme using proxy vehicles (ID-MAP) without bilinear pairings is proposed.
- Third, security analysis of ID-MAP is presented to show that it can satisfy security and privacy requirements of VANETs. In this direction, unforgebility of the underlying signature scheme against adaptively chosen-message and identity attack is proved under Elliptic Curve Discrete Logarithm Problem in the random oracle model to guarantee resistancy against modification and impersonation attacks.
- Finally, its performance analysis including comparison of computation and communication overheads and simulation is presented to show that ID-MAP is more efficient than previous schemes for VANETs.

B. Organization of the paper

The rest of this paper is organized as follows. Sections II and III present related works and background information used in the paper, respectively. Review of PBAS [19] and its security weaknesses are given in Section IV. The proposed ID-MAP and its security analysis are presented in Section V. Sections VI and VII present the performance analysis and conclusion, respectively.

II. Related works

In 2006, Raya et al. [4] introduced potential security and privacy threats for VANETs, and presented a robust security architecture which is resistant against these threats. They modified PKI to achieve authentication, integrity and privacy in a way that many key pairs and their corresponding certificates has been preloaded into vehicles which every pair is used for each communication. In Raya et al. 's scheme [4], each vehicle needs to have a large storage space to keep key pairs and their certificates, and also the trusted authority requires to have a large storage space to save vehicles'certificates to check their validity and trace them in case of any disputes. In 2008, Lu et al. [8] proposed a new authentication scheme using temporary anonymous certificates issued by RSUs to improve efficiency and large storage space issues of Raya et al.'s scheme [4]. Due to the high interactions of vehicles with RSUs to get anonymous certificates, Freudiger et al. [20] used the idea of mix-zones to have an efficient scheme, while in their scheme RSUs and vehicles have to have large storage space. In 2008, Zhang et al. [21] exploited message authentication codes along with a key agreement protocol to create a shared key between a vehicle and an RSU to propose an efficient authentication scheme. However, their scheme needs a large storage space to keep a large number of key pairs and their certificates to satisfy privacy requirement.

In 2008, Zhang et al. [9] exploited identity-based cryptography [22] in designing authentication schemes for VANETs to address certificate management problem of the previous schemes [4], [8], [20], [21]. They proposed an identity-based signature scheme with batch verification, and employed it in their scheme to reduce verification costs at RSUs [9]. In addition, their scheme satisfies conditional privacy preserving. However, in 2011, Chim et al. [10] showed that Zhang et al.'s scheme [9] is not resistant against impersonation, antitractability and privacy violating attacks, and proposed a new identity-based authentication scheme using two shared secrets to satisfy the privacy requirement and also the bloom filter and the binary search techniques. Their scheme is efficient in terms of communication overhead and for message verification by a factor of 45% compared to Zhang et al.'s scheme [9].

In addition, Lee and Lai [11] in 2013 showed that Zhang et al.'s scheme is vulnerable to the replay attack and also it does not have non-repudiation property, then they modified their scheme to have a secure identity-based authentication scheme, while keeping Zhang et al.'s scheme efficiency. In 2013, Horng et al. [12] indicated that Chim et al.'scheme is not resistant against impersonation attack, and then they modified the message signing phase of their scheme in a way that it can meet security and privacy requirements of Chim et al.'scheme [10]. In 2012, Shim [13] presented an efficient identity-based signature with batch verification, and used it in proposing an efficient conditional privacy preserving authentication scheme. In 2014, Liu et al. [18] explained that Shim's scheme [13] has some security weaknesses, i.e., false acceptance of batching invalid signatures and security flaws in the proof of Shim's signature scheme. Furthermore, they showed Shim's authentication scheme is vulnerable to modification attack, and presented some improvements for that [18]. In 2014, Zhang et al. [14] indicated that Lee and Lie's authentication scheme [11] is vulnerable to impersonation attack and does not have non-repudiation, and presented an improved scheme by modifying the signing algorithm. Furthermore, in 2015 Bayat et al. [15] presented an impersonation attack for Lee and Lie's authentication scheme [11], and tried to solve their security weakness which lead to a new and efficient authentication scheme. Unfortunately, Bayat et al.'s scheme [15] and Zhang et al.'s scheme [14] are vulnerable to the modification attack. In 2015, Liu et al. [19] proposed a new scheme for VANETs to improve computational overheads at RSUs, and named it proxy-based authentication scheme, and they showed that it has a great advantage in verification of vehicles' signatures when many vehicles are in the coverage areas of an RSU. Recently, Lo and Tsai [16] and He et al. [17] proposed efficient authentication schemes without employing bilinear pairings. However, their schemes are not suitable for situations in which it is necessary to verify a large number of messages by an RSU in a second.

III. Background

In this section, first the used notations in the paper are introduced, then, we review several fundamental backgrounds employed in this research, including outline of algorithms for a typical identity-based signature scheme and its security model.

A. Notations

In this subsection, the notations used in the paper are defined.

- TA: a trusted authority.
- RSU: a roadside unit.
- v_i: the ith vehicle.
- ID_i: the real identity of a vehicle.
- PID_i: the pseudo identity of a vehicle, where $PID_i = (PID_{i,1}, PID_{i,2})$.
- m_i: a message sent by the vehicle v_i.
- c_i: the total computational cost of a vehicle v_i.
- c_s: the computational cost of one signature generation of a vehicle v_i.
- c_v: the computational cost of one signature verification of a vehicle v_i.
- $c_{i,r}$: the extra computational resources of a vehicle v_i.
- y: the number of vehicles communicating directly with each other.
- u: the number of signed messages by a vehicle v_i.
- p: the number of proxy vehicles.
- T_i: the timestamp of a message.
- p, q: two large primes.
- P: a generator of the group $\mathbb{G}$
- $\mathbb{G}$: an additive group with order q.
- $\mathbb{G}_T$: a multiplicative group.
- e : a bilinear pairing, where $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$.
- P_{pub}, $P_{pub,1}$ and $P_{pub,2}$: system public keys.
- β, β_1 and β_2: system secret keys.
- $P_{r,1}$ and $P_{r,2}$: RSU's public keys.
- $ID_r = (ID_{r,1}, ID_{r,2})$: RSU's identity.
- β_3, β_r: RSU's secret keys.
- x_i, $x_{i,1}$ and $x_{i,2}$: vehicle's secret keys.
- x_r, $x_{r,1}$ and $x_{r,2}$: RSU's secret keys.
- $g(.)$, $k(.)$, $h(.)$, $f(.)$ and $H(.)$: secure hash functions, where $g(.)$, $k(.)$, $h(.)$ and $f(.) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H(.) : \{0,1\}^* \rightarrow \mathbb{G}$.
- $|w|$: the number of bits of the string w.
- $\perp$: an empty string.
- $\theta \leftarrow B(w_1, ...)$: the operation of assigning the output of algorithm B on inputs $w_1, ...$ to θ.
- $w \xleftarrow{\$} W$: the operation of assigning a uniformly random element of W to w.
- $\oplus$: X-OR operation.

B. Outline of identity-based signature schemes

A signer with identity ID and a verifier are participants of an identity-based signature, and the scheme consists of Setup, KeyExtract, Sign and Ver algorithms as follows [23].

- Setup: Given the system security parameter λ, it outputs system's parameters $Para$ and the master key pair (msk, mpk), i.e. $(Para, (msk, mpk)) \leftarrow Setup(\lambda)$.
- KeyExtrat: Given the system's parameter $Para$, master secret key msk and an identity ID, it outputs its corresponding secret key x, i.e. $x \leftarrow KeyExtract(Para, msk, ID)$
- Sign: Given the system's parameter $Para$, signer's secret key x and the message m to be signed, it outputs the signature θ, i.e. $\theta \leftarrow Sign(Para, x, m)$.
- Ver: Given the system's parameter $Para$, the master public key mpk, the signer's identity ID, the signature θ and the message m, it outputs 1 if θ is a valid signature of the message m and outputs 0 otherwise, i.e. $\{0,1\} \leftarrow Ver(Para, mpk, ID, \theta, m)$.

C. Security model of identity-based signature schemes

An identity-based signature scheme should be secure against existential forgery under an adaptive-chosen-message and identity attack [23].

To have a formal definition for existential unforgeability, the adversary A and a challenger C should interact through the following game [23].

1) Setup: Algorithm C runs the Setup algorithm with a security parameter λ to obtain system's parameter $Para$ and master key pair (mpk, msk), then it sends $(mpk, Para)$ to A.
2) The adversary A in addition to making queries to random oracles adaptively issues a polynomially bounded number of questions to the KeyExtract and Sign oracles as follows.
   - KeyExtract: Adversary A can request for the secret key of an identity ID of its choice. Then, C returns $x \leftarrow KeyExtract(Para, msk, ID)$ to A.
   - Sign: Adversary A can request for a signature on the message m with respect to identity ID of its choice. Algorithm C makes a KeyExtract query on identity ID to obtain $x \leftarrow KeyExtract(Para, msk, ID)$, and then returns $\theta \leftarrow Sign(Para, x, m)$ to A.
3) Eventually, A returns a valid signature θ^* on the message m^* under identity ID^*, and wins the game if m^* has not been requested to the Sign oracle, and ID^* has not been requested to KeyExtract oracle.

The formal definition of existential unforgeability is expressed in Definition 1.

Definition 1. *A signature is $(\tau, q_{ro}, q_e, q_s, \epsilon)$-existentially unforgeable against adaptive chosen message and identity attack if there is no adversary which runs in time at most τ, makes at most q_{ro} random oracle queries, q_e KeyExtract queries and q_s Sign queries, and can win the aforementioned game with probability at least ϵ.*

IV. Review and security analysis of PBAS

In this section, first we briefly review PBAS [19], and then we show its security drawbacks.

A. Review of PBAS

The PBAS [19] consists of the following phases:

1) Setup: In this phase, the trusted authority (TA) chooses two cyclic additive and multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order q, and P as a generator of the group $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following conditions hold true.
- The map $e(.,.)$ is bilinear, i.e. $e(aP, bP) = e(P, P)^{ab}$ for all a and $b \in \mathbb{Z}_q^*$.
- The value $e(P, P)$ is non-degenerate, i.e. $e(P, P) \neq 1_{\mathbb{G}_T}$.
- The map $e(.,.)$ is efficiently computable.

We refer readers to [24] for more details on the construction of bilinear pairings.
Then, TA chooses random numbers β_1, β_2, β_3 and $\beta_r \xleftarrow{\$} \mathbb{Z}_q^*$, where β_1 and β_2 are secret keys of the system and β_3 is RSU's secret key, and computes the system public keys as $P_{pub,1} = \beta_1 P$ and $P_{pub,2} = \beta_2 P$ and RSU's public keys as $P_{r,1} = \beta_3 P$ and $P_{r,2} = \beta_r P$. The tamper proof device of each vehicle is preloaded with $(\beta_1, \beta_2, \beta_3)$. In addition, each RSU computes $x_{r,2} = \beta_r P_{r,1}$, where $P_{r,1} = \beta_3 P$ and $P_{r,2} = \beta_r P$. Therefore, the secret key of the RSU is $(x_{r,1}, x_{r,2})$ and the public key is $(P_{r,1}, P_{r,2})$, where $x_{r,1} = \beta_3$. Also each vehicle chooses $k_i \xleftarrow{\$} \mathbb{Z}_q^*$, computes $PID_{i,1} = k_i P$ and $PID_{i,2} = ID_i \oplus g(k_i P_{pub,1})$. Hence, vehicle's secret key is the tuple $(x_{i,1} = \beta_1 PID_{i,1}, x_{i,2} = \beta_2 H(PID_{i,1}, PID_{i,2}))$. The TA also selects three secure hash functions $g(.)$, $H(.)$ and $h(.)$, where $g(.) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $H(.) : \{0,1\}^* \rightarrow \mathbb{G}$ and $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. Therefore, the public parameters are $Para = \{\mathbb{G}, q, P, P_{pub,1}, P_{pub,2}, P_{r,1}, P_{r,2}, H(.), g(.), h(.)\}$.
2) Message signing: Given a message (m_i, T_i), where T_i is the message timestamp and m_i is a traffic message, a vehicle computes $s_{i,1} = x_{i,1} + h(m_i, T_i)x_{i,2}$, and its tamper proof device computes $s_{i,2} = (k_i + \beta_3(h(m_i, T_i) + s_{i,1}))P_{r,2}$. Then, the vehicle sends $(PID_i, T_i, m_i, s_{i,1}, s_{i,2})$ to the proxy vehicle.
3) Batch verification by proxy vehicles: given $(PID_i, T_i, m_i, s_{i,1}, s_{i,2})$ for $1 \leq i \leq d$, a proxy vehicle verifies received signatures in batch by checking if Equation (1) holds or not.

$$\begin{aligned} &e(\textstyle\sum_{i=1}^{d} s_{i,1}, P) = e(\sum_{i=1}^{d} PID_{i,1}, P_{pub,1}) \times \\ &e(\textstyle\sum_{i=1}^{d} h(m_i, T_i) H(PID_{i,1}, PID_{i,2}), P_{pub,2}) \end{aligned} \quad (1)$$

If it holds, the proxy vehicle computes $\sigma_1 = \sum_{i=1}^{d} s_{i,1}$, $\sigma_2 = \prod_{i=1}^{d} s_{i,2}$ and $s_{p,1}$ as given in Equation (2).

$$s_{p,1} = x_{p,1} + h(m_p, T_p)x_{p,2}, \quad (2)$$

where $m_p = (b, \sigma_1, \sigma_2, PID_i, T_i, m_i, 1 \leq i \leq d, T_p)$, and T_p is a timestamp, and b indicates that the batch result is valid ($b = 1$) or invalid ($b = 0$).
Then, it sends $(b, \sigma_1, \sigma_2, PID_i, m_i, 1 \leq i \leq d, T_p, s_{p,1})$ to the RSU.
4) Verification by an RSU at proxy vehicle's output: The RSU verifies the proxy vehicle's signature $s_{p,1}$ by checking Equation (3) to be sure about integrity of the received message.

$$\begin{aligned} &e(s_{p,1}, P) = e(PID_{p,1}, P_{pub,1}) \times \\ &e(h(m_p, T_p) H(PID_{p,1}, PID_{p,2}), P_{pub,2}), \end{aligned} \quad (3)$$

where $m_p = (b, \sigma_1, \sigma_2, PID_i, T_i, m_i, 1 \leq i \leq d, T_p)$. If $s_{p,1}$ is valid and the received message is fresh by checking T_i, $1 \leq i \leq d$ and T_p, then, it checks if Equation (4) holds or not.

$$\begin{aligned} &e(\textstyle\prod_{i=1}^{d} s_{i,2}, P_{r,2}) = \\ &e(\textstyle\prod_{i=1}^{d} PID_{i,1}[\sum_{i=1}^{d}(h(m_i, T_i)) + \sigma_1] x_{r,2}, x_{r,1}) \end{aligned} \quad (4)$$

If Equation (4) holds, the batch result, σ_1, is correctly generated by the proxy vehicle, and is accepted by the RSU. If Equation (4) holds and $b = 0$, or if Equation (4) does not hold, the RSU asks TA to revoke the privacy of the malicious proxy vehicle.

B. Security analysis of PBAS

In this subsection, we show that PBAS [19] cannot preserve message authentication requirement, and consequently it is vulnerable against common attacks such as modification and impersonation attack. In addition, it is vulnerable against false acceptance of invalid signatures. In what follows, the mentioned weaknesses are described.

1) **Weakness 1.** The PBAS [19] does not satisfy authenticity of messages, the main requirement of VANETs. Hence, it is not resistant against modification and impersonation attacks. To show this weakness, we indicate that the signature $s_{i,1}$ is forgeable. For this goal, it is assumed that a vehicle sends a message in form of $(PID_i, T_i, m_i, s_{i,1}, s_{i,2})$ to a proxy vehicle, and also it plays the role of a proxy vehicle and sends $(b, \sigma_1, \sigma_2, PID_i, m_i, 1 \leq i \leq d, s_{p,1}, T_p)$ to an RSU, every malicious entity can extract the vehicle's secret keys $x_{i,1}$ and $x_{i,2}$ from two relations $s_{i,1} = x_{i,1} + h(m_i, T_i)x_{i,2}$ and $s_{p,1} = x_{i,1} + h(m_p, T_p)x_{i,2}$ as given in Equation (5).

$$\begin{aligned} &x_{i,2} = (h(m_i, T_i) - h(m_p, T_p))^{-1}(s_{i,1} - s_{p,1}), \\ &x_{i,1} = s_{i,1} - h(m_i, T_i)x_{i,2}. \end{aligned} \quad (5)$$

Hence, the adversary with having vehicle's secret keys $x_{i,1}$ and $x_{i,2}$ can forge new signatures on each new message m_i' and also on each message sent to the RSU in the validity period of pseudo identities. As a consequence, the message authentication cannot be preserved. In fact, the main reason for this vulnerability is that the underlying signature scheme used to generate $s_{i,1}$ is not secure against adaptively-chosen-message attack, and can be forged. In what follows, we show its vulnerabilities to modification and impersonation attacks.

- **Vulnerability to the modification attack**: To show this weakness, consider the following scenario. An

adversary can modify some messages in transmission from proxy vehicles to an RSU such as b in a way that $b = 0$ is replaced with $b = 1$, and then signed the new message with proxy vehicle's secret keys $x_{i,1}$ and $x_{i,2}$ obtained from Equation (5). When the RSU checks the correctness of proxy vehicle's output, it seems to the RSU that the proxy vehicle is malicious since Equation (4) holds but $b = 0$. Hence, RSU asks TA to revoke proxy vehicle's privacy, while the proxy vehicle is honest and the adversary has modified the message.

- **Vulnerability to the impersonation attack**: To show this weakness, consider the following scenario. A malicious entity with having vehicle's secret keys $x_{i,1}$ and $x_{i,2}$ obtained from Equation (5) can forge new signatures on new messages m'_i it wants, and plays the role of a vehicle v_i. For this purpose, it generates the new signature $s'_{i,1} = x_{i,1}+h(m'_i, T'_i)x_{i,2}$, and sends it to the proxy vehicle. Since the signature $s'_{i,1}$ is generated based on the protocol, it will be accepted by the proxy vehicle since Equation (1) holds. Then, the proxy vehicle sends the message $(b, \sigma'_1, \sigma_2, PID_i, m'_i, 1 \leq i \leq d, s_{p,1}, T_p)$ to an RSU, where σ_2 is RSU's signature on messages m_i, $1 \leq i \leq d$, and cannot be changed by the adversary, while σ'_1 is adversary's signature on messages m'_i, $1 \leq i \leq d$. When the RSU verifies the correctness of proxy vehicle's operations, it seems to the RSU that the proxy vehicle is malicious since Equation (4) does not hold (since signatures σ_1 and σ'_1 are generated on different messages m_i and m'_i) and $b = 1$. Hence, the RSU asks TA to revoke proxy vehicle's privacy, while the adversary has forged the signature $s'_{i,1}$ on the message m'_i and the proxy vehicle is honest.

2) **Weakness 2.** The PBAS [19] is not resistant against false acceptance of invalid signatures. To show this weakness consider the following scenario.
If an adversary changes two valid signatures in transmission $s_{1,1}$ and $s_{2,1}$ to two invalid signatures $s'_{1,1} = s_{1,1} + wP$ and $s'_{2,1} = s_{2,1} - wP$, where $w \xleftarrow{\$} \mathbb{Z}_q^*$. The batch verification of the proxy vehicle outputs validity of these signatures since the Equation (1) holds, while vehicles' signatures are invalid. This is due to the fact that in batch verification, the term wP is omitted by the term $-wP$ in the relation $\sum_{i=1}^{2} s_{i,1}$. As a consequence, proxy vehicles cannot detect this issue, and send the batch result and its corresponding signature σ_2 to an RSU. Since in verification of the result by RSUs in Equation (4) there exists the term σ_1, hence, RSUs cannot detect this issue, while two invalid signatures have been batched. Therefore, this scheme is not resistant against batching two or more invalid signatures, and verification by proxy vehicles and also by the RSU output validity of received invalid signatures. Actually, the reason of this vulnerability is that signatures are added simply to verify signatures in batch.

V. ID-MAP Scheme

In this section, first the system model, problem definition and security model of ID-MAP are reviewed. Then, details of ID-MAP and its security analysis are given.

A. System model

In ID-MAP, there are three participants which are explained below :

- Trusted authority (TA): The TA is a trusted third party which generates system parameters and master public key and secret key, generates members' secret key, preloads them into vehicles, and can trace vehicles from their pseudo identities in case of any misbehavior. In addition, TA considers some benefits for vehicles with extra computational capabilities to promote them to behave as proxy vehicles. Note that computation and communication capabilities of TA are high.
- Roadside units (RSUs): The RSUs are at roadsides, communicate with vehicles (proxy vehicles), can check the validity of received messages from vehicles (proxy vehicles), and sends them to the traffic control center. In addition, RSUs record proxy vehicles' pseudo identities and their history to send to TA. Note that since the number of proxy vehicles are small, it does not have any impact on efficiency of the protocol.
- Vehicles: These are equipped with tamper-proof devices on board units (OBU), and communicate with each others and RSUs. In addition, if they have extra computational resources, they can be proxy vehicles and serve for RSUs in authenticating received messages.

Communications between vehicles and between vehicles and RSUs are through DSRC protocol as identified in IEEE 802.11p [16], [17], [19].

B. Problem definition

An important and challenging issue for VANETs is fast validation of vehicles' messages by RSUs when there are many vehicles in the coverage area of an RSU. For example, an RSU must verify 5000 messages in one second once the number of vehicles is 500 and each vehicle broadcasts a traffic message every 100 ms according to the specification of DSRC protocol, while as shown in the recently proposed efficient authentication schemes [17], [19], the maximum number of messages can be verified in a second are about 2500.

Proxy-based authentication technique which was proposed by Liu et al. [19] is a solution to this issue in which vehicles with extra computational capacities can behave as proxy vehicles to help RSUs to validate multiple messages at the same time. For this purpose, proxy vehicles send the verification result along with a proof to show the correctness of the result to RSUs. In fact, the goal of employing proxy vehicles is to reduce centralized computing overheads at RSUs with using distributed computing to have efficient authentication for VANETs when there are a large number of vehicles in the coverage area of an RSU. The only proposed scheme based on this technique is not secure as aforementioned in

Subsection IV-B. Therefore, it is necessary to propose a secure and efficient scheme for this aim.

C. Security model

The ID-MAP needs to satisfy the following security and privacy requirements:

- Message authentication: (Proxy) vehicles and RSUs should be able to check authenticity, integrity and validity of the received messages.
- Identity privacy preserving: Vehicles and RSUs except for TA should not be able extract real identity of a vehicle or a proxy vehicle from its messages.
- Traceability: The TA can find out the real identity of a (proxy) vehicle from its message in case of any misbehavior.
- Unlinkability: Vehicles and RSUs should not be able to find a link between two messages sent by the same vehicle.
- Resistance to attacks: Common attacks in VANETs such as the impersonation attack, modification attack, the replay attack and man in the middle attack should be prevented.

D. Details of ID-MAP

There are five phases in this scheme, Setup, Anonymous identity generation, Message generation, Verification of messages by proxy vehicles and Verification of proxy vehicles' output by RSUs, which are described in what follows.

1) Setup: In this phase, system parameters are generated by TA, and have been loaded into vehicles' tamper proof devices and into RSUs. For this goal, the following steps are done by TA.
   - The TA chooses two large prime numbers p and q, and an elliptic curve E over a prime finite field F_p defined by equation $y^2 = x^3 + ax + b$ for a and $b \in F_p$ such that $\Delta = 4a^3 + 27b^2 \neq 0$.
   - The TA chooses a cyclic additive group, $\mathbb{G}$, with order q, and P as the generator of $\mathbb{G}$. The group $\mathbb{G}$ consists of all points on the elliptic curve E and the point at infinity O.
   - The TA chooses $\beta \xleftarrow{\$} \mathbb{Z}_q^*$ as the system secret key. Then, it computes the system public key as $P_{pub} = \beta P$.
   - The TA selects four secure hash functions $h(.)$, $k(.)$, $g(.)$ and $f(.)$, where $h(.)$, $k(.)$, $g(.)$ and $f(.) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. Therefore, public parameters are $Para = \{\mathbb{G}, p, q, P, P_{pub}, h(.), k(.), g(.), f(.)\}$.
   - The TA for RSU's identity ID chooses $\beta_r \xleftarrow{\$} \mathbb{Z}_q^*$, and sets $ID_{r,1} = \beta_r P$, $ID_{r,2} = ID$ and $ID_r = (ID_{r,1}, ID_{r,2})$. Then, TA computes RSU's secret key x_r as $x_r = \beta_r + \beta f(ID_r)$. Hence, RSU's secret key corresponding to its identity ID_r is x_r.
   - The TA puts $\{Para, ID_i, \beta, x_r, ID_r\}$ into tamper-proof devices of each vehicle.
2) Anonymous identity generation: In this phase, each vehicle v_i hides its real identity, ID_i, through getting a registered pseudo identity PID_i, and then generates its corresponding secret key. To do this, the tamper proof device of a vehicle v_i, which is preloaded with $Para$ and β, chooses α_i at random from $\mathbb{Z}_q^*$, computes $PID_{i,1} = \alpha_i P$, $PID_{i,2} = ID_i \oplus g(\alpha_i P_{pub})$ to attain the pseudo identity $PID_i = (PID_{i,1}, PID_{i,2})$. Then, it computes $x_i = \alpha_i + \beta g(PID_i) \bmod q$ to generate vehicle's secret key x_i, and gives (x_i, PID_i) to the vehicle.
3) Message generation: In this phase, a vehicle chooses a random number r_i from $\mathbb{Z}_q^*$, computes $R_i = r_i P$, $h_i = h(m_i, PID_i, T_i, R_i)$ and $s_{i,1} = r_i h_i + x_i \bmod q$, and also its tamper proof device selects a random number w_i from $\mathbb{Z}_q^*$, computes $W_i = w_i P$ and $s_{i,2} = x_r(k(m_i, T_i, PID_i, W_i) + s_{i,1}) + w_i \bmod q$, and sends $(PID_i, T_i, m_i, R_i, W_i, s_{i,1}, s_{i,2})$ to proxy vehicles, where T_i is a timestamp.

   Note that methodology of proxy vehicle selection presented by Liu et al. [19] is given in Algorithm 1. In fact, this selection method algorithm is based on calculating vehicles' extra computational resources, and this algorithm can be run by each vehicle without help of TA. If there is no vehicle with this criteria, the validation of vehicle's signatures is done by RSUs as done in previous works [16], [17]. In Algorithm 1, let c_i be the total computational cost of a vehicle v_i, c_s be the computational cost of one signature generation, c_v be the computational cost of one signature verification, u be the number of signed messages by v_i, y be the number of vehicles communicating directly with each other, $c_{i,r}$ be the extra computational resources of a vehicle v_i and p be the number of proxy vehicles.

   Remark 1. *To promote vehicles with extra computational resources to behave as a proxy vehicle, RSUs record their pseudo identities and their history, and send them to TA to consider some benefits for them such as a reduction at vehicle's payments or at vehicles' taxes. Note that this does not have any impact on the efficiency of the protocol since the number of proxy vehicles are small as mentioned by Liu et al. [19].*
4) Verification of messages by proxy vehicles: In

Algorithm 1 . Proxy vehicle selection algorithm

Input: (c_i, c_s, c_v, u, y)
Output: $(p, \{v_{p,1}, ..., v_{p,p}\})$
1: $z = 0$.
2: $p = 0$.
3: **for** $1 \leq i \leq y$ **do**
4: the computation of $c_{i,r} = c_i - uc_s$.
5: **if** $c_{i,r} > 0$ **then**
6: v_i is a potential proxy vehicle **and** $d_{i,r} = c_{i,r}$.
7: **end if**
8: $z = z + 1$
9: **end for**
10: compute the mean value d_{mr} of $d_{i,r}$ for $1 \leq i \leq z$.
11: **for** $1 \leq i \leq z$ **do**
12: **if** $d_{i,r} > d_{mr}$ **then**
13: v_i is selected as a proxy vehicle $v_{p,i}$ **and** the number of signatures that can be verified are $\frac{d_{mr} - uc_s}{c_v}$.
14: **end if**
15: $p = p + 1$
16: **end for**

this phase, a proxy vehicle verifies the integrity and senders' identities of received messages, $(PID_i, T_i, m_i, R_i, W_i, s_{i,1}, s_{i,2})$ for $1 \leq i \leq d$. For this goal, the proxy vehicle first checks the freshness of the received message by the timestamp T_i and the validity period of pseudo identities. If messages are fresh and pseudo identities are valid, the proxy vehicle computes $h_i = h(m_i, PID_i, T_i, R_i)$, $g_i = g(PID_i)$, for $1 \leq i \leq d$, and it chooses a vector $\overrightarrow{A} = (a_1, ..., a_d)$, where a_i is an integer from $[1, 2^\gamma]$ for the security parameter γ. Note that the size of γ is selected according to the trade-off between security and efficiency, and this value usually is set to 80 in VANET scenarios [18]. Then, it checks if Equation (6) holds or not.

$$\begin{aligned}&(\textstyle\sum_{i=1}^{d} a_i s_{i,1})P = \\ &\textstyle\sum_{i=1}^{d}(a_i h_i)R_i + \sum_{i=1}^{d} a_i PID_{i,1} \\ &+(\textstyle\sum_{i=1}^{d}(a_i g_i))P_{pub}.\end{aligned} \tag{6}$$

If Equation (6) holds, d distinct signatures $s_{i,1}$ are valid. Then, the proxy vehicle computes $\sigma_1 = \sum_{i=1}^{d} s_{i,1}$. After that, the proxy vehicle computes $k_i = k(m_i, T_i, PID_i, W_i)$ for $1 \leq i \leq d$. Then, it checks if Equation (7) holds or not.

$$\begin{aligned}&(\textstyle\sum_{i=1}^{d} a_i s_{i,2})P = \\ &(\textstyle\sum_{i=1}^{d} a_i(k_i + s_{i,1}))(ID_{r,1} + f(ID_{r,1}, ID_{r,2})P_{pub}) \\ &+\textstyle\sum_{i=1}^{d} a_i W_i.\end{aligned} \tag{7}$$

If Equation (7) holds, d distinct signatures $s_{i,2}$ are valid. Then, the proxy vehicle computes $\sigma_2 = \sum_{i=1}^{d} s_{i,2}$, and sends $\{b, PID_p, PID_i, W_i, T_i, 1 \leq i \leq d, \sigma_1, \sigma_2, R_p, s_p\}$ to an RSU. Here, the value of b indicates that the result of the batch is valid or not, $b = 1$ means that the batch result is valid and $b = 0$ indicates that the result is invalid. The signature (R_p, s_p) is proxy vehicle's signature on the message $m_p = (b, PID_p, PID_i, W_i, T_i, 1 \leq i \leq d, \sigma_1, \sigma_2)$ to guarantee message integrity, where $R_p = r_p P$, $h_p = h(m_p, PID_p, T_p, R_p)$ and $s_p = r_p h_p + x_p \bmod q$. The correctness of Equation (6) is verified as follows.

$$\begin{aligned}&(\textstyle\sum_{i=1}^{d} a_i s_{i,1})P \\ &= \textstyle\sum_{i=1}^{d} a_i(r_i h_i + x_i))P \\ &= \textstyle\sum_{i=1}^{d} a_i(h_i R_i + x_i P) \\ &= \textstyle\sum_{i=1}^{d}((a_i h_i)R_i + a_i x_i P) \\ &= \textstyle\sum_{i=1}^{d}((a_i h_i)R_i + a_i(PID_{i,1} + g_i P_{pub})) \\ &= \textstyle\sum_{i=1}^{d}(a_i h_i)R_i + \sum_{i=1}^{d} a_i PID_{i,1} \\ &+(\textstyle\sum_{i=1}^{d}(a_i g_i))P_{pub},\end{aligned} \tag{8}$$

where $h_i = h(m_i, PID_i, T_i, R_i)$, $g_i = g(PID_i)$, for $1 \leq i \leq d$. Similarly, the correctness of Equation (7) can be shown.

5) Verification of proxy vehicles' output by RSUs: In this phase, an RSU verifies the results received from proxy vehicles to detect false results and revoke malicious proxy vehicles. For this purpose, the following tasks are done as described below.

- An RSU first verifies proxy vehicle's signature, (R_p, s_p), to check integrity and sender's identity of the received message as given in Equation (9). If it is valid, the RSU goes to the next step; otherwise, it rejects the received message.

$$s_p P = h_p R_p + PID_{p,1} + g_p P_{pub}, \tag{9}$$

  where $h_p = h(m_p, PID_p, T_p, R_p)$, $g_p = g(PID_p)$ and $m_p = (b, PID_p, PID_i, W_i, T_i, 1 \leq i \leq d, \sigma_1, \sigma_2)$.
- It checks the freshness of the received message by timestamp T_i and validity of pseudo identities PID_i. If messages are fresh and PID_i are valid, the RSU goes to the next step; otherwise, it rejects the received message.
- The RSU checks correctness of the received result generated by the proxy vehicle. If Equation (10) holds, the correctness of the batch result is checked.

$$\begin{aligned}&\sigma_2 P = ((\textstyle\sum_{i=1}^{d} k_i) + \sigma_1) \\ &(ID_{r,1} + f(ID_{r,1}, ID_{r,2})P_{pub}) + \textstyle\sum_{i=1}^{d} W_i,\end{aligned} \tag{10}$$

  where $\sigma_2 = \sum_{i=1}^{d} s_{i,2}$ and $\sigma_1 = \sum_{i=1}^{d} s_{i,1}$, $k_i = k(m_i, T_i, PID_i, W_i)$ for $1 \leq i \leq d$.
  The correctness of Equation (10) is verified as follows.

$$\begin{aligned}&\sigma_2 P = (\textstyle\sum_{i=1}^{d} s_{i,2})P \\ &= \textstyle\sum_{i=1}^{d}[x_r(k(m_i, T_i, PID_i, W_i) + s_{i,1}) + w_i]P \\ &= \textstyle\sum_{i=1}^{d}[(k_i + s_{i,1}) \\ &(ID_{r,1} + h(ID_{r,1}, ID_{r,2})P_{pub}) + W_i] \\ &= (\textstyle\sum_{i=1}^{d}(k_i + s_{i,1}))(ID_{r,1} + h(ID_{r,1}, ID_{r,2})P_{pub}) \\ &+\textstyle\sum_{i=1}^{d} W_i \\ &= (\textstyle\sum_{i=1}^{d} k_i + \sum_{i=1}^{d} s_{i,1}) \\ &(ID_{r,1} + h(ID_{r,1}, ID_{r,2})P_{pub}) + \textstyle\sum_{i=1}^{d} W_i \\ &= (\textstyle\sum_{i=1}^{d} k_i + \sigma_1)(ID_{r,1} + h(ID_{r,1}, ID_{r,2})P_{pub}) \\ &+\textstyle\sum_{i=1}^{d} W_i\end{aligned} \tag{11}$$

- If Equation (10) does not hold or Equation (10) holds and $b = 0$, the RSU indicates that the proxy vehicle is malicious, and asks TA to revoke the privacy of that proxy vehicle. This action avoids disturbing authentication process later. Methodology of malicious proxy vehicle revocation is given in Algorithm 2.
  Note that if some proxy vehicles are malicious and revoked, a backup server must immediately take over failed vehicles, which are authenticated by malicious proxy vehicles, for time-critical information. In addition, this revocation mechanism makes the number of malicious proxy vehicles be below 10% of proxy vehicles as mentioned by Liu et al. [19] and Huang et al. [25].

Algorithm 2 . Malicious proxy vehicle revocation algorithm

Input: $(Para, b, \beta, ID_r, PID_p, PID_i, W_i, T_i, m_i, 1 \leq i \leq d, \sigma_1, \sigma_2)$
Output: ID_p
1: **if** Equation (10) does not hold **then**
2: TA computes $ID_p = PID_{p,2} \oplus g(\beta PID_{p,1})$ to revoke PID_p.
3: **end if**
4: **if** Equation (10) holds **and** $b = 0$ **then**
5: TA computes $ID_p = PID_{p,2} \oplus g(\beta PID_{p,1})$ to revoke PID_p.
6: **end if**

E. Security analysis

In this subsection, before we show that ID-MAP has the security and privacy requirements, existential unforgeability of the signature on the batch result, σ_2, is proved in the random oracle model (see [26] for the background). In order to prove unforgeability of the proposed scheme, we need to show that it is unforgeable against adversary A (as defined in Section III-C). Our main result on the security of σ_2 or equivalently $s_{i,2}$ is summarized in Theorem 1.

To start let us present the mathematical problem which is used in the proof of our scheme.

Definition 2. Elliptic Curve Discrete Logarithm Problem (ECDLP). Given $\mathbb{G}$, P as the generator of $\mathbb{G}$ and $Q = \gamma P$, output $\gamma \in \mathbb{Z}_q^*$.

Theorem 1. *If ECDLP problem is $(\tau_{ECDLP}, \epsilon_{ECDLP})$-hard, then the signature scheme is $(\tau, q_k, q_f, q_e, q_s, \epsilon)$-existentially unforgeable against adaptively chosen-message and identity attack in the random oracle model such that*

$$\begin{aligned} &\epsilon_{ECDLP} \geq \epsilon_1\left(\frac{\epsilon_1^3}{q_k+q_f} - \frac{1}{|\mathbb{G}|}\right), \\ &\tau_{ECDLP} \leq 4(\tau + (3q_s + 2q_e)t_m), \end{aligned} \tag{12}$$

where $\epsilon_1 = \epsilon - \frac{q_s(2q_s+q_k)}{|\mathbb{G}|} - \frac{q_e(q_s+2q_e+q_f)}{|\mathbb{G}|}$ and t_m is the required time for scalar multiplication. In addition, q_k, q_f, q_e and q_s are the number of queries to oracles $k(.)$, $f(.)$, KeyExtract and Sign, respectively.

Proof. It is supposed that there is an adversary A against unforgeability of the scheme with success probability ϵ. We construct another algorithm C to solve ECDLP problem with success probability ϵ_{ECDLP}. Given a random instance of ECDLP $(\mathbb{G}, P, Q = \gamma P)$, algorithm C outputs γ.

The algorithm C runs Setup on a security parameter λ, and gets a random instance of the ECDLP $(\mathbb{G}, P, Q = \gamma P)$, to set the system public key, P_{pub}, to Q and generate the public parameters $Para = \{\mathbb{G}, q, P, P_{pub}\}$, and then invokes the adversary A on $Para$. The adversary A runs in time at most τ, makes q_k and q_f queries to the random oracles $k(.)$, $f(.)$, respectively, and also makes q_e and q_s queries to the KeyExtract and Sign oracles, respectively. Finally, it can win the unforgeability game with probability at least ϵ. Algorithm C maintains initially empty associative tables $T_k[.]$ and $T_f[.]$ to simulate random oracles $k(.)$ and $f(.)$, and answers A's oracle queries as described below.

- $k(.)$ queries: If $T_k[.]$ is defined for the query (m_i, T_i, PID_i, W_i), then, C returns its value; otherwise, C chooses $T_k[m_i, T_i, PID_i, W_i] \xleftarrow{\$} \mathbb{Z}_q^*$, and returns $k(m_i, T_i, PID_i, W_i)$ to A.
- $f(.)$ queries: If $T_f[.]$ is defined for query $ID_r = (ID_{r,1}, ID_{r,2})$, then, C returns its value; otherwise, C chooses $T_f[ID_r] \xleftarrow{\$} \mathbb{Z}_q^*$, and returns $f(ID_r)$ to A.
- KeyExtract queries: For a query ID, C sets $ID_{r,2} = ID$, chooses two random numbers f and x_r from $\mathbb{Z}_q^*$, computes $ID_{r,1} = -x_r P + fP_{pub}$. If $T_f[ID_{r,1}, ID_{r,2}]$ has already been defined, then, C halts, returns $\perp$ and sets $bad \leftarrow true$; otherwise, it sets $T_f[ID_{r,1}, ID_{r,2}] \leftarrow f$, and returns RSU's secret key $(x_r, ID_{r,1})$ corresponding to the identity ID to A.
- Sign queries: For a query $(m_i, s_{i,1}, T_i, PID_i)$ under identity ID_r, C first makes an f query on ID_r to attain the value of f. Then, C chooses two random numbers k_i and $s_{i,2}$ from $\mathbb{Z}_q^*$, computes $W_i = -s_{i,2}P + (k_i + s_{i,1})(ID_{r,1} + f(ID_{r,1}, ID_{r,2})P_{pub})$. If $T_k[m_i, T_i, PID_i, W_i]$ has already been defined, then, C halts, returns $\perp$ and sets $bad \leftarrow true$; otherwise, it sets $T_k[m_i, T_i, PID_i, W_i] \leftarrow k_i$, and returns the signature $(s_{i,2}, k_i, W_i)$ on the message $(m_i, s_{i,1}, T_i, PID_i)$ under identity ID_r to A.
- Finally, it is assumed that A outputs a forged signature $(s_{i,2}, k_i, W_i)$ on the message $(m_i, s_{i,1}, T_i, PID_i)$ under identity ID_r. The forgery is non-trivial if A has not made a Sign query on the input of $(m_i, s_{i,1}, T_i, PID_i)$ under ID_r, and a KeyExtract query on the input of ID.

The probability of A in returning a forged signature $(s_{i,2}, k_i, W_i)$ on the message $(m_i, s_{i,1}, T_i, PID_i)$ under identity ID_r is $\epsilon_1 = \Pr[E_1]\Pr[E_2|E_1]$ which is computed as follows. First of all, we define events E_1 and E_2.

- Event E_1 : Algorithm C does not abort as a result of signature simulation.
- Event E_2: Adversary A returns a non-trivial forgery.

To lower-bound the probability of $\Pr[E_1]$ and $\Pr[E_2|E_1]$, we need to compute the probability $\Pr[\neg bad]$, where the event bad indicates that C aborts in signature simulation as a result of A's KeyExtract and Sign queries. This probability is computed as follows.

Claim 1. $\Pr[E_1] = \Pr[\neg bad] \geq 1 - \frac{q_s(2q_s+q_k)}{|\mathbb{G}|} - \frac{q_e(q_s+2q_e+q_f)}{|\mathbb{G}|}$.

Proof. The value of $\Pr[bad]$, is multiplication of the following probabilities.

- Case 1. We have $bad \leftarrow true$ if the pair $(ID_{r,1}, ID_{r,2})$ generated in a KeyExtract simulation has been occurred by chance in a previous query to the oracle $f(.)$. Since there are at most $q_e + q_f + q_s$ entries in the table $T_f[.]$ and the number of $ID_{r,1}$, uniformly distributed in $\mathbb{G}$, is $|\mathbb{G}|$, the probability of this event for one KeyExtract query is at most $\frac{(q_e+q_f+q_s)}{|\mathbb{G}|}$. Hence, the probability of this event for q_e queries is at most $\frac{q_e(q_e+q_f+q_s)}{|\mathbb{G}|}$.
- Case 2. We have $bad \leftarrow true$ if C previously used the same randomness $ID_{r,1}$ in one KeyExtract simulation. Since there are at most q_e KeyExtract queries, this probability is at most

$\frac{q_e}{|\mathbb{G}|}$. Therefore, for q_e KeyExtract queries, the probability of this event is at most $\frac{q_e^2}{|\mathbb{G}|}$.

- Case 3. We have $bad \leftarrow true$ if the pair (m_i, T_i, PID_i, W_i) generated in a Sign simulation has been occurred by chance in a previous query to the oracle $k(.)$. Similar to the probability analysis given in Case 1, the probability of this event for q_s queries is at most $\frac{q_s(q_s+q_k)}{|\mathbb{G}|}$.
- Case 4. We have $bad \leftarrow true$ if C previously used the same randomness W_i in one Sign simulation, and similar to the probability analysis given in Case 2 this probability for q_s Sign queries is at most $\frac{q_s^2}{|\mathbb{G}|}$.

Claim 2. $\Pr[E_2|E_1] \geq \epsilon$.

Proof. The value of $\Pr[E_2|E_1]$ is the probability that A returns a valid forgery provided that C does not abort as a result of A's KeyExtract and Sign queries. If C did not abort as a result of A's queries, all its responses to those queries are valid. Therefore, by hypothesis A will produce a non-trivial forgery with probability at least ϵ.

Therefore, the probability that A returns a valid forgery $(s_{i,2}, k_i, W_i)$ on the message $(m_i, s_{i,1}, T_i, PID_i)$ under identity ID_r is at least

$$\epsilon_1 = \epsilon - \frac{q_s(2q_s + q_k)}{|\mathbb{G}|} - \frac{q_e(2q_e + q_f + q_s)}{|\mathbb{G}|}.$$

Then, C runs Multiple Forking algorithm of Boldyreva et al. [27] to obtain four valid forgeries on the same tuple $(m_i, s_{i,1}, T_i, PID_i, W_i)$ with different values for $k(m_i, T_i, PID_i, W_i)$ and $f(ID_{r,1}, ID_{r,2})$ under ID_r as presented in Equation (13).

$$\begin{aligned} s_{i,2}^1 &= (\gamma f + \beta_r)(k_i^1 + s_{i,1}) + w_i \bmod q \\ s_{i,2}^2 &= (\gamma f + \beta_r)(k_i^2 + s_{i,1}) + w_i \bmod q \\ s_{i,2}^3 &= (\gamma f' + \beta_r)(k_i^3 + s_{i,1}) + w_i \bmod q \\ s_{i,2}^4 &= (\gamma f' + \beta_r)(k_i^4 + s_{i,1}) + w_i \bmod q, \end{aligned} \tag{13}$$

where $f \neq f'$, $k_i^2 \neq k_i^1$ and $k_i^3 \neq k_i^4$.

As a result, the solution to ECDLP, γ, is computed in Equation (14).

$$\gamma = \frac{s_{i,2}^3 + s_{i,2}^2 - s_{i,2}^1 - s_{i,2}^4}{f'(k_i^3 - k_i^4) - f(k_i^1 - k_i^2)} \tag{14}$$

The success probability of C according to the Multiple-Forking Lemma of Boldyreva et al. [27] is bounded by $\epsilon_1(\frac{\epsilon_1^3}{q_k+q_f} - \frac{1}{|\mathbb{G}|})$, where $\epsilon_1 = \epsilon - \frac{q_s(2q_s+q_k)}{|\mathbb{G}|} - \frac{q_e(q_s+2q_e+q_f)}{|\mathbb{G}|}$.

In order to compute the value τ_{ECDLP}, it is assumed that a scalar multiplication takes time t_m, while all other operations take zero time. The running time of C is A's run-time, τ, plus the time required to respond to hash queries, q_e KeyExtract and q_s Sign queries. Therefore, C's run-time is $\tau_{ECDLP} \leq 4(\tau + (2q_e + 3q_s)t_m)$. This completes the proof.

□

- Message authentication: The proposed ID-MAP provides message integrity and validity of the sender's identity due to the following reasons:
  - Signatures $s_{i,1}$ and $s_{i,2}$, $1 \leq i \leq d$ are used to check authenticity of the received messages from vehicles to proxy vehicles and from proxy vehicles to RSUs, respectively. In addition, signatures $s_{i,1}$ and $s_{i,2}$, $1 \leq i \leq d$ are existentially unforgeable against adaptively chosen-message and identity attack under difficulty of ECDLP problem in the random oracle model as proved by Lo and Tsai [16] and in Theorem 1, respectively.
    To make it clear, proxy vehicles verify vehicles' signatures, $s_{i,1}$ and $s_{i,2}$ for $1 \leq i \leq d$ in batch to check message integrity and vehicles' identities. As a consequence, authenticity and validity of vehicle's identities are checked by proxy vehicles. In addition, with employing the small exponent test methodology [12], [14] in the batch verification of multiple messages as used in Equation (6) and Equation (7), ID-MAP is resistant against false acceptance of invalid signatures.
  - The validity of proxy vehicle's identity, integrity and authenticity of received messages from proxy vehicles which include the batch result σ_1 and its corresponding signature σ_2 are also checked by proxy vehicle's signature (R_p, s_p) which is existentially unforgeable against adaptively chosen-message and identity attack in the random oracle model as proved by Lo and Tsai [16].
  - An RSU can check the correctness of the batch result which is generated by proxy vehicles by verifying σ_2 as presented in Equation (10). In addition, security of our proposed signature scheme is proved under the difficulty of ECDLP problem in the random oracle model in Theorem 1. In fact, the tamper proof device of each vehicle generates $s_{i,2}$ for RSUs to check integrity of the signature $s_{i,1}$. As a consequence, informally without knowing RSUs' secret keys, x_r, it is impossible to generate $s_{i,2}$. Therefore, malicious proxy vehicles cannot generate σ_2 for its false results.

- Identity privacy preserving: The ID-MAP has conditional privacy preserving since the real identity of each vehicle, ID_i, is converted to a pseudo identity PID_i by vehicle's tamper proof device, where $PID_{i,1} = \alpha_i P$, $PID_{i,2} = ID_i \oplus g(\alpha_i P_{pub})$. In addition, the pseudo identity and its corresponding secret key of each vehicle dynamically changes. To extract the real identity of a vehicle ID_i from PID_i, an adversary needs to compute $\alpha_i P_{pub}$ from P_{pub} and $PID_{i,1} = \alpha_i P$ which means that it has to solve Computational Diffie-Hellman Problem (CDHP). As a consequence, due to the difficulty of CDHP or informally without knowing TA's secret key, β, no one can find out the real identity of a vehicle from its pseudo identity PID_i. Hence, vehicles and RSUs except for TA cannot extract the real identity of a vehicle from its messages.
- Traceability: The TA can find out the real identity, ID_i, of a vehicle from its pseudo identity $(PID_{i,1}, PID_{i,2})$, where $PID_{i,1} = \alpha_i P$, $PID_{i,2} = ID_i \oplus g(\alpha_i P_{pub})$. To

TABLE I
COMPARISON OF COMPUTATION OVERHEADS

Schemes	Computational cost of a proxy vehicle	Computational cost of an RSU
ID-MAP	$306T_{mul}$	$5\lceil\frac{n}{300}\rceil T_{mul}$
PBAS [19]	$300(4T_{mul}+5T_p+T_{mtp})$	$2\lceil\frac{n}{300}\rceil T_{mul}+(2\lceil\frac{n}{300}\rceil+3)T_p+T_{mtp}$
CPPA-1 [16]	NA	$(n+2)T_{mul}$
CPPA-2 [17]	NA	$(n+2)T_{mul}$

do this, TA uses its secret key β, and computes $ID_i = PID_{i,2} \oplus g(\beta PID_{i,1})$. Therefore, TA can trace vehicles from its messages in case of any misbehavior.

- Unlinkability: In this scheme, since two different messages generated by the same vehicle are signed by different pseudo identities and their corresponding secret keys, and also these pseudo identities are not related since different α_i are used in their generation. As a consequence, vehicles and RSUs cannot link two messages sent by the same vehicle.
- Resistance to attacks: Since $s_{i,1}$ and $s_{i,2}$ are existentially unforgeable against adaptively chosen-message and identity attack, and our authentication scheme is designed based on those signature schemes to provide authenticity, integrity and validity of the sender's identities. This leads to resistancy against common attacks such as the impersonation attack, modification attack and man in the middle attack. In addition, timestamp T_i is used in signatures $s_{i,1}$, $s_{i,2}$ and in transmitted messages to proxy vehicles and RSUs to provide freshness of messages and avoid replay attack.

VI. PERFORMANCE ANALYSIS

A. Computational overhead

Comparison of ID-MAP, PBAS [19], the only proxy-based authentication scheme, and two recently proposed authentication schemes [16], [17] in terms of computational overhead at an RSU and at a proxy vehicle for proxy-based schemes is summarized in Table I. In Table I, T_{mtp}, T_{mul} and T_p denote the time required for computing a Map-to-Point hash function, scalar multiplication and one pairing, respectively. Also, NA means not applicable.

In the comparison, it is assumed that each proxy vehicle can verify at most 300 messages ($d = 300$) which is reasonable as assumed by Liu et al. in PBAS [19], and traffic density is the number of messages, n, in a verification period. Hence, $\lceil\frac{n}{300}\rceil$ is the number of proxy vehicles.

Since verifying a single message at an RSU in ID-MAP costs about $5T_{mul}$, and that in PBAS [19] costs about $2T_{mul}+5T_p+T_{mtp}$, verifying n messages, which are sent by $\lceil\frac{n}{300}\rceil$ proxy vehicles, at an RSU costs about $5\lceil\frac{n}{300}\rceil T_{mul}$, and that costs about $(2\lceil\frac{n}{300}\rceil T_{mul}+(2\lceil\frac{n}{300}\rceil+3)T_p+T_{mtp})$ in PBAS [19]. Computational cost of a proxy vehicle in ID-MAP includes the cost of verifying 300 messages in batch and the cost of one signature generation, $306T_{mul}$, while the computational cost of a proxy vehicle in PBAS [19] is $300(4T_{mul}+5T_p+T_{mtp})$. In addition, as shown in Table I, verifying n messages in an RSU in both CPPA-1 [16] and CPPA-2 [17] costs about $(n+2)T_{mul}$.

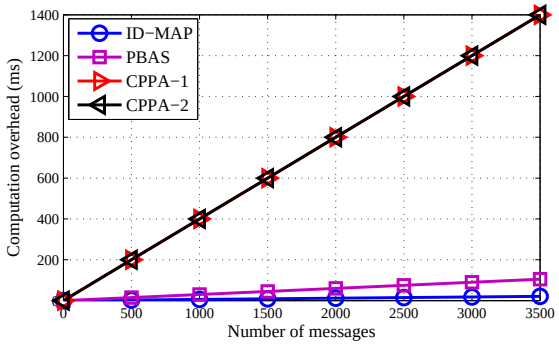

Fig. 1. Comparison of computation overheads in terms of number of messages in an RSU

According to the experimental results, which calculate the required time for employing MIRACL cryptographic library [28] by choosing the Tate pairing on a 160-bit subgroup of an MNT curve [29] with an embedding degree 6 for the security level of 2^{80}, running on an Intel i7 3.07GHz machine, T_{mtp}, T_{mul} and T_p take 0.09 ms, 0.39 ms and 3.21 ms, respectively. Computational costs of other operations are negligible, and they are not considered in the comparison [12].

To verify 3000 signatures, the required time at an RSU in ID-MAP is 19.5 ms $(= 5\lceil\frac{n}{300}\rceil T_{mul} = 5\lceil\frac{3000}{300}\rceil \times 0.39)$, while this value in PBAS [19], CPPA-1 [16] and CPPA-2 [17] is approximately 81.6 ms $(= 2\lceil\frac{n}{300}\rceil T_{mul} + (2\lceil\frac{n}{300}\rceil+3)T_p + T_{mtp} = 2\lceil\frac{3000}{300}\rceil \times 0.39 + (2\lceil\frac{3000}{300}\rceil+3) \times 3.21 + 0.09)$, 1170 ms $(= (3000+2)0.39)$ and 1170 ms $(= (3000+2)0.39)$, respectively. Therefore, ID-MAP is more efficient than PBAS [19] and other efficient schemes [16], [17] in terms of computational overhead at an RSU since it is pairing-free, and also it employs distributed computing method.

In addition, the required time at a proxy vehicle in ID-MAP is 119.5 ms $(= 306T_{mul} = 306 \times 0.39)$, while this value in PBAS [19] is approximately 5010 ms $(= 300(4T_{mul}+5T_p+T_{mtp}) = 300(4 \times 0.39 + 5 \times 3.21 + 0.09))$. Hence, ID-MAP has a better performance compared to PBAS [19].

It should be mentioned that the computational load of proxy vehicles in proxy-based schemes ID-MAP and PBAS which are based on distributed computing method is increased compared to efficient schemes [16], [17] as shown in Table I. But this issue does not have any impact on the efficiency of these schemes since proxy vehicles are selected vehicles with extra computational resources as given in Algorithm 1.

In other words, as shown in Fig.1, the maximum number of the messages can be verified at an RSU per second in PBAS [19], CPPA-1 [16] and CPPA-2 [17] is approximately 25650, 2562 and 2562, respectively, while in ID-MAP this number reaches to 153846. As a consequence, ID-MAP is a good candidate to improve computational cost at RSUs with employing proxy vehicles when there are a large number of vehicles in their coverage areas compared to previous efficient

TABLE II
COMPARISON OF COMMUNICATION OVERHEADS (IN BYTES)

Schemes	Sending 300 messages to a proxy vehicle	Sending n messages to an RSU
PBAS [19]	$164(300)$	$204\lceil\frac{n}{300}\rceil + 84n$
CPPA-1 [16]	NA	$144n$
CPPA-2 [17]	NA	$144n$
ID-MAP	$204(300)$	$184\lceil\frac{n}{300}\rceil + 124n$

schemes.

B. Communication overhead

In this subsection, comparison of communication costs of ID-MAP, PBAS [19] and two recently proposed schemes; CPPA-1 [16] and CPPA-2 [17] is given in Table II. Note that NA means not applicable. For the security level of 2^{80}, it is assumed that q be 160 bits or 20 bytes, and each element in $\mathbb{G}$ is 40 bytes. In addition, the size of the timestamp is 4 bytes. This comparison is in terms of transmitting n messages to an RSU. In addition, it is assumed that each proxy vehicle receives 300 ($d = 300$) messages, so the number of proxy vehicles is $\lceil\frac{n}{300}\rceil$. In the comparison, the size of the message m_i is not considered since they are the same in all authentication schemes. In PBAS [19], the message sent by a vehicle to a proxy vehicle (V2PV) is $(PID_{i,1}, PID_{i,2}, T_i, s_{i,1}, s_{i,2})$, where $PID_{i,1}$, $PID_{i,2}$, $s_{i,1}$ and $s_{i,2} \in \mathbb{G}$, and so its size is $40\times4+4 = 164$ bytes, and so for sending 300 messages to a proxy vehicle we have $300(164)$ bytes, while in ID-MAP the message sent by a vehicle to a proxy vehicle (V2PV) is $(PID_{i,1}, PID_{i,2}, T_i, W_i, s_{i,1}, s_{i,2})$, where $PID_{i,1}$, $PID_{i,2}$ and $W_i \in \mathbb{G}$, $s_{i,1}$ and $s_{i,2} \in \mathbb{Z}_q^*$, and its size is $40\times4+2\times20+4 = 204$. Hence, the communication overhead for transmitting 300 messages is $300(204)$ bytes.

Note that since it is assumed that each proxy vehicle verifies 300 messages, so the number of proxy vehicles to verify n messages is $\lceil\frac{n}{300}\rceil$. In PBAS [19], the message sent by a proxy vehicle to an RSU (PV2R) is $(PID_{p,1}, PID_{p,2}, T_p, s_{p,1}, \sigma_1, \sigma_2, PID_{i,1}, PID_{i,2}, T_i,$ $1 \le i \le 300)$, where $PID_{p,1}$, $PID_{p,2}$, $PID_{i,1}$, $PID_{i,2}$, $s_{p,1}$, σ_1 and $\sigma_2 \in \mathbb{G}$. Hence, the size of the transmitted messages by $\lceil\frac{n}{300}\rceil$ proxy vehicles is $(40 \times 5 + 4)\lceil\frac{n}{300}\rceil + (2 \times 40 + 4)n = 204\lceil\frac{n}{300}\rceil + 84n$ bytes, while in ID-MAP the transmitted message from a proxy vehicle to an RSU is $(PID_{p,1}, PID_{p,2}, T_p, R_p, s_{p,1}, \sigma_1, \sigma_2, PID_{i,1}, PID_{i,2}, W_i,$ $T_i, 1 \le i \le 300)$, where $PID_{p,1}$, $PID_{p,2}$, $PID_{i,1}$, $PID_{i,2}$, W_i and $R_p \in \mathbb{G}$, $s_{p,1}$, σ_1 and $\sigma_2 \in \mathbb{Z}_q^*$, and its size is $(40\times3+3\times20+4)\lceil\frac{n}{300}\rceil+(3\times40+4)n = 184\lceil\frac{n}{300}\rceil+124n$ bytes. By the same analysis, the message size sent by vehicles to an RSU (V2R) for n messages in both CPPA-1 [16] and CPPA-2 [17] is $144n$ bytes.

To send 3000 signatures to an RSU, the message size sent to an RSU in ID-MAP is 373840 bytes ($= 184\lceil\frac{3000}{300}\rceil+124\times3000$), while this value in PBAS [19] is 254040 bytes ($= 204\lceil\frac{3000}{300}\rceil+$ 84×3000). In addition, message size transmitted to an RSU in both CPPA-1 [16] and CPPA-2 [17] is 432000 bytes ($= 144 \times 3000$). It should be noted that $300(164) = 49200$ and $300(204) = 61200$ bytes are sent to a proxy vehicle by vehicles in PBAS [19] and ID-MAP schemes, respectively. But this issue does not have any impact on VANET's efficiency since vehicles are close to proxy vehicles, and communicated directly with them, and also this communication overhead is distributed between vehicles, while all communication load is put on RSUs in both CPPA-1 [16] and CPPA-2 [17]. This result is consistent with the goal of proxy-based authentication schemes which reduces the communication load of RSUs. However, the communication overhead of ID-MAP compared to that of PBAS is increased, it should be highlighted that PBAS is not secure as aforementioned in Subsection IV-B. As a consequence, ID-MAP has a better communication overhead at the side of RSUs compared to the previous efficient and secure authentication schemes as shown in Fig. 2.

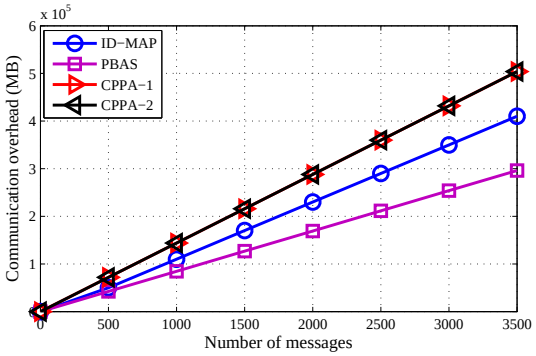

Fig. 2. Comparison of communication overheads in terms of number of messages

C. Simulations

We simulate the proposed protocol in NS2.35, which is flexible and provides us a comparison environment with existing protocols. VanetMobiSim [30] is used to simulate mobility model of vehicles. The simulation results show that the average message delay and the average loss ratios in RSUs with running each simulation result 100 times to analyze the performance of ID-MAP and compare it with the recently proposed efficient schemes: PBAS [19], CPPA-1 [16] and CPPA-2 [17]. Note that in the first two simulation results, the speeds of vehicles are about $10 \sim 30$ m/s. Parameters and road scenario of mobility model used by Liu et al. [19] are given in Table III.

Fig. 3 demonstrates the comparison of average message delays (AMD), the required time for transmission of messages from vehicles to an RSU, of ID-MAP, PBAS [19], CPPA-1 [16] and CPPA-2 [17] in terms of the number of vehicles. It is obvious that the value of AMD becomes large with an increase in the number of vehicles. As shown in Fig. 3, ID-MAP has the lowest AMD when the number of vehicles are increased since ID-MAP has faster message validation by RSUs. As a consequence, the simulation results indicate that performance of ID-MAP is slightly affected by an increase in the number of vehicles or traffic density.

In Fig. 4, comparison of average message loss ratio (AMLR), the ratio between the number of dropped messages and the total number of messages received by an RSU in every

TABLE III
SIMULATION PARAMETERS [19]

Parameters	Values
Coverage area	$8000 \times 16m^2$
No. of traffic lanes	4
No. of RSUs	5
Maximum No. of proxy vehicles	20
Simulation duration	100 s
MAC layer protocol	802.11p
Channel bandwidth	6 Mbps
Transmission range of a vehicle	300m
Transmission range of an RSU	1000 m
Minimum inter-vehicle distance	40 m
Routing protocol	AODV
Slot time	$13\mu s$
SIFS	$32\mu s$
AIFS (high priority)	$58\ \mu s$
Contention window size (CW)	$15 \sim 1023\ \mu s$

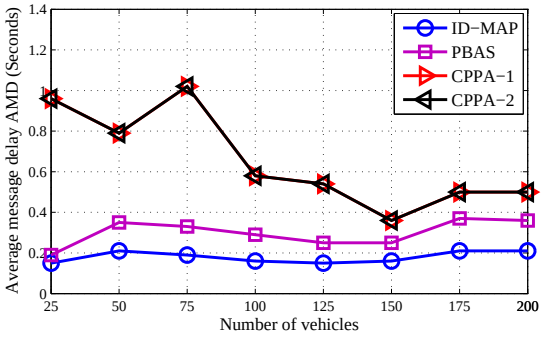

Fig. 3. Comparison of average message delays in terms of vehicles' number in RSUs

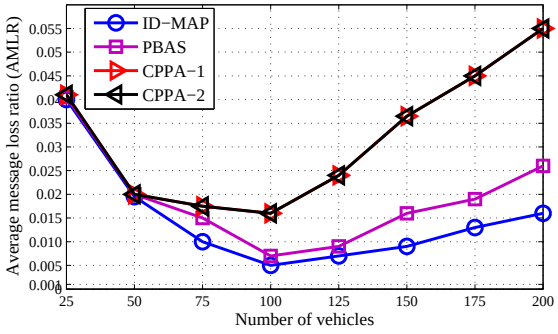

Fig. 4. Comparison of average message loss ratios in terms of vehicles' number in RSUs

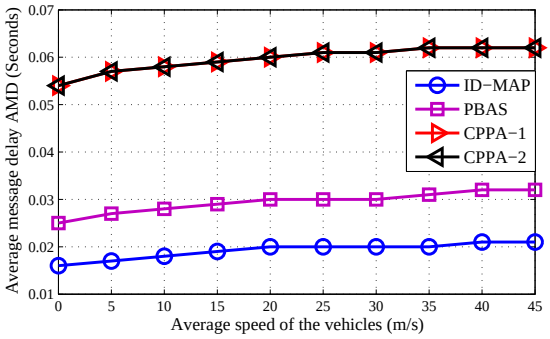

Fig. 5. Comparison of average message delays in terms of vehicles' speed in RSUs

100s, of ID-MAP, PBAS [19], CPPA-1 [16] and CPPA-2 [17] in terms of the number of vehicles is given. Since AODV protocol which employs relays for routing is employed in simulation, and assists vehicles with 300 m transmission range in forwarding messages, the AMLR of schemes reduces at the beginning of the frame with an increase in the number of vehicles in the light of this fact that the quantity of relays increments as the quantity of vehicles increments. However, the AMLR of schemes increases due to the collisions caused by hidden terminal problem and frequent transmissions between RSUs and vehicles in the same communication area, ID-MAP has the lowest AMLR compared to previous schemes as shown in the simulation result.

Fig. 5 represents the comparison of average message delays (AMD) of our proposed scheme ID-MAP, PBAS [19], CPPA-1 [16] and CPPA-2 [17] in terms of average speed of vehicles. As shown in Fig. 5, AMD of schemes is approximately constant for different values of speeds. Hence, the simulation results indicate that AMD of schemes is slightly affected by an increase at vehicles' speeds, and also ID-MAP has the lowest AMD compared to previous schemes.

In Fig. 6, comparison of average message loss ratio (AMLR) of our proposed scheme ID-MAP, PBAS [19], CPPA-1 [16] and CPPA-2 [17] in terms of average speed of vehicles is given. However, an increase in the average speed of vehicles incredibly affects AMLR of schemes, ID-MAP has the lowest AMLR compared to other schemes as shown in the simulation result.

VII. CONCLUSION

In this paper, we showed that PBAS proposed by Liu et al. [19] for VANETs has some security drawbacks: it does not satisfy message authentication, and also it is not resistant against impersonation and modification attacks and false acceptance of invalid signatures. Then, to tackle the security weaknesses of PBAS, we proposed ID-MAP for vehicular networks. To show that it is secure against aforementioned attacks and it has message authenticity, we proved that the underlying signature scheme is secure against adaptively chosen-message and identity attack under ECDLP problem in the random oracle model. As shown in the comparison, ID-MAP is more efficient than PBAS, and also the simulation results show that ID-MAP is an efficient candidate for VANET's authentication in realistic environments. We should emphasize that ID-MAP is useful where there are a large number of vehicles in the coverage area of an RSU, and it was shown from the analysis that the required time to verify 3000 signatures in one second for ID-MAP was improved by 76% and 98% compared to PBAS [19] and the two recently efficient authentication schemes [16], [17], respectively.

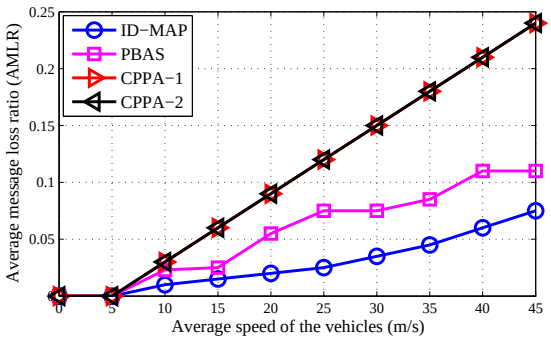

Fig. 6. Comparison of average message loss ratios in terms of vehicles' speed in RSUs

ACKNOWLEDGEMENTS

The authors would like to appreciate anonymous reviewer for their valuable comments on this work. In addition, we are grateful to Dr. Liu for useful discussions and providing us the exact code for comparison with their study.

REFERENCES

[1] S. Zeadally, R. Hun, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETs): Status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[2] M. Ghosh, A. Varghese, A. Gupta, A. A. Kherani, and S. N. Muthaiah, "Detecting misbehaviors in VANET with integrated root-cause analysis," *Ad Hoc Networks*, vol. 8, no. 7, pp. 778–790, 2010.

[3] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: Applications and related technical issues," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 74–88, 2008.

[4] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 8–15, 2006.

[5] J. T. Isaac, S. Zeadally, and J. S. Camara, "Security attacks and solutions for vehicular ad hoc networks," *IET Communications*, vol. 4, no. 7, pp. 894–903, 2010.

[6] J. P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[7] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[8] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. of the 27th Int. Conf. on the Computer Communications-IEEE INFOCOM 2008*. Phoenix, AZ, USA: IEEE, 13-18 April 2008, pp. 1903–1911.

[9] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. of the 27th Int. Conf. on Computer Communications-IEEE INFOCOM 2008*. Phoenix, AZ, USA: IEEE, 13-18 April 2008, pp. 816–824.

[10] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K. Li, "SPECS: Secure and privacy enhancing communications schemes for VANETs," *Ad Hoc Networks*, vol. 9, no. 2, pp. 189–203, 2011.

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

[12] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li, and M. K. Khan, "b-SPECS+: Batch verification for secure pseudonymous authentication in VANET," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1860–1875, 2013.

[13] K.-A. Shim, "CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1874–1883, 2012.

[14] J. Zhang, M. Xu, and L. Liu, "On the security of a secure batch verification with group testing for VANET," *International Journal of Network Security*, vol. 16, no. 5, pp. 355–362, 2014.

[15] 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.

[16] N.-W. Lo and J. L. Tsai, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without bilinear pairings," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1319–1328, 2015.

[17] 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.

[18] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2559–2564, 2014.

[19] Y. Liu, L. Wang, and H.-H. Chen, "Message authentication using proxy vehicles in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3697–3710, 2015.

[20] J. Freudiger, M. Raya, M. Felegyhazi, P. Papadimitratos, and J.-P. Hubaux, "Mix-zones for location privacy in vehicular networks," in *Proc. of the 1st Int. ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*. Vancouver, BC, Canada: ACM, 14 August 2007, pp. 1–7.

[21] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks," in *Proc. of IEEE Int. Conf. on Communications (ICC 2008)*. Beijing, China: IEEE, 30 May 2008, pp. 1451–1457.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of 4th Annual Int. Cryptology Conf. on Advances in Cryptology-CRYPTO 1984*. Santa Barbara, CA, USA: Springer-Verlag, Berlin, 19-22 August 1985, pp. 47–53.

[23] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," *Journal of Cryptology*, vol. 22, no. 1, pp. 1–61, 2009.

[24] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. of the 21st Ann. Int. Crypto. Conf., Advances in Cryptology-Crypto 2001*. Santa Barbara, California, USA: Springer-Verlag, Berlin, 19-23 August 2001, pp. 213–229.

[25] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 248–262, 2011.

[26] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. of the 1st ACM Conf. on Computer and Communications Security (CCS 1993)*. Fairfax, VA, USA: ACM, New York, NY, 3-5 November 1993, pp. 62–73.

[27] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *Journal of Cryptology*, vol. 25, no. 1, pp. 57–115, 2012.

[28] MIRACL Cryptographic Library: Multiprecision Integer and Rational Arithmetic C/C++ Library. Available: http:// indigo.ie /

[29] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals*, vol. E84-A, no. 5, pp. 1234–1243, 2001.

[30] VanetMobiSim Project Home Page. Available: http://vanet.eurecom.fr.

Maryam Rajabzadeh Asaar received her B.S. degree in Electrical Engineering from Shahid Bahonar University of Kerman, Kerman, Iran, in 2004, and received her M.S. and Ph.D. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran in 2008 and 2014, respectively. She is currently an asistant professor at Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. Her research interests include provable security, digital signatures, design and analysis of cryptographic protocols and network security and security in industrial control systems.

Mahmoud Salmasizadeh received the B.S. and M.S. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1972 and 1989, respectively. He also received the Ph.D. degree in information technology from Queensland University of Technology, Australia, in 1997. Currently, he is an associate professor in the Electronics Research Institute and adjunct associate professor in the Electrical Engineering Department, Sharif University of Technology. His research interests include Design and Cryptanalysis of cryptographic algorithms and protocols, E-commerce Security, and Information Theoretic Secrecy. He is a founding member of Iranian Society of Cryptology.

Willy Susilo (SM'01) received a Ph.D. in Computer Science from University of Wollongong, Australia. He is a Professor and Head of School of Computing and Information Technology at the University of Wollongong. He is also the director of Centre for Computer and Information Security Research (CCISR) at the University of Wollongong. Previously, he held a prestigious ARC Future Fellow awarded by the Australian Research Council (ARC). His main research interests include cryptography and information security. His main contribution is in the area of digital signature schemes. He has served as a program committee member in dozens of international conferences. He has published numerous publications in the area of digital signature schemes and encryption schemes. He is a senior member of IEEE since 2001.

Akbar Majidi received the B.S. and M.S. degrees in computer engineering in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, He is conducting research in networking and computer engineering. His research interests are wireless body area networks, 5th generation wireless network, software defined networking and network function virtualization.