Proxy-based Authentication Scheme for Vehicular Ad Hoc Networks: Security Analysis and an Efficient Scheme

Maryam Rajabzadeh Asaar, Mahmoud Salmasizadeh, Willy Susilo

***Abstract*—In vehicular ad hoc networks, message authentication using proxy vehicles was proposed to reduce the computational overhead of roadside unites. In this type of message authentication schemes, proxy vehicles with verifying multiple messages at the same time improve computational efficiency of roadside unites 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 presented for this goal by Liu et al. is not resistant against false acceptance of batching invalid signatures and modification attack. Next, we propose an new identity-based message authentication scheme with employing proxy vehicles. Then, unforgeability of underlying signature is proved under Elliptic Curve Discrete Logarithm Problem in the random oracle model to show that it is secure against modification attack. It should be highlighted that our proposed scheme not only is more efficient than Liu et al.'s scheme 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.**

Keywords: 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 Unite (OBU), and a wireless communication protocol named as Dedicated Short Range Communication (DSRC) which 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 [6, 8, 15] 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 [10–16] 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 [17] and He et al. [18] 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 an RodeSide Unite (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 [17–19] as stated by Liu et al. [20] in 2015. To tackle the aforementioned problem, Liu et al. [20] proposed an interesting authentication protocol using proxy vehicles for vehicular networks. In their scheme, proxy vehicles help RSUs to verify a large number of signatures simultaneously

M. R. Asaar and M. Salmasizadeh are with the Electronics Research Institute of Sharif University of Technology, Tehran, Iran. W.Susilo is with Centre for Computer and Information Security Research, University of Wollongong, Wollongong, Australia
E-mail: {asaar}@ee.sharif.edu, {salmasi} @sharif.edu, {wsusilo} @uow.edu.au

M. Salmasizadeh is partially supported by Iran Energy Efficiency Organization (IEEO-SABA), Contract Number: 94-279.

using distributed computing. Actually, in their proposal the time required to verify 3000 signatures is decreased by 88% compared to previous efficient authentication schemes [17–19] based on batch verification method at RSUs.

A. Our contributions

Contributions of this paper are three-fold.

- First, we show that the proxy-based authentication scheme for vehicular networks presented by Liu et al. [20] is not resistant against false acceptance of invalid signatures sent by vehicles and modification attack.
- Second, to tackle the aforementioned problems and have a more efficient scheme, a new authentication scheme using proxy vehicles without bilinear pairings is proposed.
- Third, security analysis of the proposed scheme 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 attack is proved under Elliptic Curve Discrete Logarithm Problem in the random oracle model.
- Finally, its performance analysis including computation and communication overheads is presented to show that this proposed scheme 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 Liu et al.'s scheme [20] and its security weaknesses are given in Section IV. Our proposed authentication scheme and its formal security analysis are presented in Section V. Sections VI and VII present the comparison and conclusion, respectively.

II. Related works

In 2008, Zhang et al. [10] exploited identity-based cryptography [21] in designing authentication schemes for VANETs to address certificate management problem. They proposed an identity-based signature scheme with batch verification, and employed it in their scheme to reduce verification costs at RSUs [10]. However, in 2011, Chim et al. [11] showed that Zhang et al.'s scheme [10] is not resistant against impersonation and anti-tractability attacks, and proposed a new identity-based authentication scheme which is efficient in terms of communication overhead. In addition, Lee and Lai [12] 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 proposed a new identity-based authentication scheme. In 2013, Horng et al. [13] indicated that Chim et al.'scheme is not resistant against impersonation attack.

In 2012, Shim [14] 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. [19] explained that Shim's scheme [14] has some security weaknesses, i.e., false acceptance of batching invalid signatures and security flaws in the proof of Shim's signature. Furthermore, they showed Shim's authentication scheme is vulnerable to modification attack, and presented some improvements for that [19]. In 2014, Zhang et al. [15] indicated that Lee and Lie's authentication scheme [12] 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. [16] presented an impersonation attack for Lee and Lie's authentication scheme [12], and tried to solve their security weakness which lead to a new and efficient authentication scheme. Unfortunately, Bayat et al.'s scheme [16] and Zhang et al.'s scheme [15] are vulnerable to the modification attack. In 2015, Liu et al. [20] 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 [17] and He et al. [18] proposed efficient authentication schemes without employing bilinear pairings.

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 signature scheme and its security model, the network model and security and privacy requirements of VANETs.

A. Notations

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

- $\oplus$: X-OR operation.
- $|y|$: the number of bits of the string y.
- $\perp$: an empty string.
- $\theta \leftarrow B(y_1, ...)$: the operation of assigning the output of algorithm B on inputs $y_1, ...$ to θ.
- $y \xleftarrow{\$} Y$: the operation of assigning a uniformly random element of Y to y.

B. Outline of signature schemes

A signer with public key pk and a verifier are participants of a signature, and the scheme consists of Setup, Sign and Ver algorithms as follows [22].

- Setup: Given the system security parameter λ, it outputs system's parameters $Para$ and the users' key pair (sk, pk), i.e. $(Para, (sk, pk)) \leftarrow Setup(\lambda)$.
- Sign: Given the system's parameter $Para$, signer's secret key sk and the message m to be signed, it outputs the signature θ, i.e. $\theta \leftarrow Sign(Para, sk, m)$.
- Ver: Given the system's parameter $Para$, the signer's public key pk, 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. $\{0, 1\} \leftarrow Ver(Para, pk, \theta, m)$.

C. Security model of signature schemes

A signature scheme should be secure against existential forgery under an adaptive-chosen-message attack [22].

To have a formal definition for existential unforgeability, the adversary A and a challenger C should interact through the following game [22].

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

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

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

D. Network model

There are four participants in vehicular ad-hoc networks as explained below [17, 18, 20]:

- Trusted Authority (TA): The TA is a trusted third party which generates system parameters, preloads them into vehicles, and can trace vehicles from their pseud identities. Computation and communication capabilities of TA are high.
- RoadSide Unites (RSUs): The RSUs are at roadsides, communicate with vehicles, can check the validity of received messages, and sends them to the traffic control center.
- Application servers (AS): The AS supports safety-related applications at traffic control center, and communicates with RSUs to provide application support.
- Vehicles: These are equipped with tamper-proof devices On Board Unites (OBU), and communicate with each others and RSUs.

Researchers consider a two-layer network model for VANETs [17, 18, 20]. Lower layer of the network consists of vehicles and RSUs, and the upper layer consists of TA and AS. In the former, they communicate with each other through DSRC, while in the latter the communications are done through secure socket layer (SSL).

E. Security and privacy requirements

A message authentication scheme should meet the following requirements [17, 18, 20]:

- Message authentication: Vehicles and RSUs should be able to check integrity and validity of the received messages.
- Identity privacy preserving: Vehicles and RSUs except for TA cannot extract real identity of a vehicle from its messages.
- Traceability: The TA can find out the real identity of a vehicle from its message in case of any misbehaviours.
- Unlinkability: Vehicles and RSUs cannot link 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.

IV. Review and security analysis of Liu et al.'s proxy-based authentication scheme

In this section, first we briefly review Liu et al.'s authentication scheme [20], then we show that it is not resistant against false acceptance of invalid signatures and the modification attack.

A. Review of Liu et al.'s proxy-based authentication scheme

Liu et al.'s proxy-based authentication scheme [20] consists of the following phases:

1) Setup: In this phase, system parameters are generated by the trusted authority (TA), and have been loaded into vehicles' temper proof devices and into RSUs. For this goal, the following steps are done by TA.

   - The TA chooses two cyclic additive and multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p. It selects 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 [23] for more details on the construction of bilinear pairings.
   - The 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 $(\beta_1, \beta_2, \beta_3)$.
   - An 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 $x_{r,1} = \beta_3$.
   - 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 selects three secure hash functions $g(.)$, $H(.)$ and $h(.)$, where $g(.) : \{0,1\}^* \rightarrow \{0,1\}^*$, $H(.) : \{0,1\}^* \rightarrow \mathbb{G}$ 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. If it holds, the proxy vehicle computes $\sigma_1 = \sum_{i=1}^{d} s_{i,1}$ and $\sigma_2 = \prod_{i=1}^{d} s_{i,2}$, and sends $(b, \sigma_1, \sigma_2, PID_i, m_i, 1 \leq i \leq d, s_{p,1})$ to the RSU, where b indicates that the batch result is valid ($b = 1$) or invalid ($b = 0$).

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

4) Verification by an RSU at proxy vehicle's output: The RSU verifies the proxy vehicle's signature $s_{p,1}$ to be sure about integrity of the received message. Then, it checks if $e(\prod_{i=1}^{d} s_{i,2}, P_{r,2}) = e(\prod_{i=1}^{d} PID_{i,1}[\sum_{i=1}^{d}(h(m_i, T_i) + s_{i,1})]x_{r,2}, x_{r,1})$ holds. If it holds, the batch of the received messages is authenticated; otherwise, the RSU asks the vehicle to revoke the malicious proxy vehicle.

B. Security analysis of Liu et al.'s proxy-based authentication scheme

In this subsection, we show that Liu et al.'s scheme [20] has two security drawbacks: false acceptance of the batch result and vulnerability to modification attack as described below.

- False acceptance of the batch result: 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 $-wP$. 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 there exists the term σ_1, hence RSUs cannot detect this issue, while two invalid signatures has been batched. Therefore, this scheme is not resistant against batching two or more invalid signatures, verification by proxy vehicles and 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
- Vulnerability to the modification attack: To show this weakness, consider the following scenario. If a vehicle sends a message as $(PID_i, T_i, m_i, s_{i,1}, s_{i,2})$ to a proxy vehicle and also it plays the role of a proxy vehicle and sends $(b, \sigma_1, \sigma_2, PID_i, m_i, 1 \leq i \leq d, s_{p,1})$ to an RSU, a 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 $s_{p,1} =$

$x_{i,1}+h(m_p, T_p)x_{i,2}$. Hence, the adversary can forge new signatures on each batch results σ'_1 it wants and also on new messages m'_i in the validity period of pseudo identities. As a consequence, message authentication cannot be preserved. In addition, when the adversary modifies some messages during the transmission and an RSU checks the correctness of the operation of a proxy vehicle, it seems to the RSU that the proxy vehicle is malicious, and its privacy is revoked, while the adversary has been modified the result. 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.

V. Our identity-based authentication scheme with proxy vehicles

In this section, details of our efficient identity-based authentication scheme using proxy vehicles are explained, and then its security analysis is given.

A. Details of the proposed signature scheme

There are five phases in this scheme, Setup, Anonymous identity generation, Message signature generation, Batch verification by proxy vehicles and Verification by an RSU at the outputs of proxy vehicles, 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 β and $\beta_r \in_R \mathbb{Z}_q^*$, where the former is the system secret key and the latter is RSU's secret key. Then, it computes the system public key as $P_{pub} = \beta P$, and RSU's public key as $P_r = \beta_r P$.
   - The TA selects three secure hash functions $h(.)$, $k(.)$ and $g(.)$, where $h(.) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $k(.) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $g(.) : \{0,1\}^* \rightarrow \{0,1\}^*$. Therefore, public parameters are $Para = \{\mathbb{G}, p, q, P, P_{pub}, P_r, h(.), k(.), g(.)\}$.
   - The TA puts $\{Para, ID_i, \beta\}$ into tamper-proof devices of each vehicle.

Algorithm 1 . Proxy vehicle selection algorithm

Let c_i be the total cost of a vehicle v_i.
Let c_s be the cost of one signature generation.
Let c_v be the cost of one signature verification.
Let u be the number of signed messages by v_i.
Let $c_{i,r}$ be the extra resource of a vehicle v_i.
Let p be the number of proxy vehicles.
for $1 \leq i \leq u$ **do**
    the computation of $c_{i,r} = c_i - uc_s$.
    if $c_{i,r} > 0$ **then**
        v_i is a potential proxy vehicle **and** $d_{i,r} = c_{i,r}$.
    end if
end for
compute the mean value d_{mr} of $d_{i,r}$ for $1 \leq i \leq p$.
if $d_{i,r} > d_{mr}$ **then**
    v_i is selected as a proxy vehicle **and** the number of signatures that can be verified are $\frac{d_{mr} - uc_s}{c_v}$.
end if

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 signing: In this phase, a vehicle generates a random number r_i, 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 computes $s_{i,2} = x_r(k(m_i, T_i, PID_i) + s_{i,1}) + \alpha_i \bmod q$ and sends $(PID_i, T_i, m_i, R_i, s_{i,1}, s_{i,2})$ to proxy vehicles, where T_i is the timestamp.
   Methodology of proxy vehicle selection presented by Liu et al. [20] is given in Algorithm 1.
4) Batch verification by proxy vehicles: In this phase, a proxy vehicle verifies the integrity and senders' identities of received messages, $(PID_i, T_i, m_i, R_i, s_{i,1}, s_{i,2})$ 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 $1 \leq i \leq d$, then it chooses a vector $\overrightarrow{A} = (a_1, ..., a_d)$, where a_i is a small integer from $[1, 2^\gamma]$ for small γ. Then, it checks if Equation 2 holds or not.

$$\begin{aligned}&(\textstyle\sum_{i=1}^{d} a_i s_{i,1})P = \\ &\textstyle\sum_{i=1}^{d}(a_i h_i)R_i + \sum_{i=1}^{d} a_i PID_{i,1} \\ &+(\textstyle\sum_{i=1}^{d}(a_i g_i))P_{pub}.\end{aligned} \tag{2}$$

If Equation 2 holds, d distinct signatures are

valid; the proxy vehicle computes $\sigma_1 = \sum_{i=1}^{d} s_{i,1}$ and $\sigma_2 = \sum_{i=1}^{d} s_{i,2}$. Then, it sends $\{b, PID_p, PID_i, T_i, 1 \leq i \leq d, \sigma_1, \sigma_2, R_p, s_p\}$ 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 $(b, PID_p, PID_i, T_i, 1 \leq i \leq d, \sigma_1, \sigma_2)$ to guarantee message integrity.
The correctness of Equation 3 is verified as follows.

$$\begin{aligned}
&(\textstyle\sum_{i=1}^{d} a_i s_{i,1})P \\
&= \textstyle\sum_{i=1}^{d} a_i(r_i h_i + x_i))P \\
&= \textstyle\sum_{i=1}^{d} a_i(h_i R_i + x_i P) \\
&= \textstyle\sum_{i=1}^{d}((a_i h_i)R_i + a_i x_i P) \\
&= \textstyle\sum_{i=1}^{d}((a_i h_i)R_i + a_i(PID_{i,1} + g_i P_{pub})) \\
&= \textstyle\sum_{i=1}^{d}(a_i h_i)R_i + \sum_{i=1}^{d} a_i PID_{i,1} \\
&+(\textstyle\sum_{i=1}^{d}(a_i g_i))P_{pub}.
\end{aligned} \tag{3}$$

5) Verification by an RSU at the outputs of proxy vehicles: In this phase, an RSU verifies the results received from proxy vehicles to detect false results and revokes 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. If it is valid, the RSU goes to the next step; otherwise, rejects the received message.
- It checks the freshness of the received message by the timestamp T_i and validity of pseud identities PID_i. If messages are fresh and PID_i are valid, go to the next step; otherwise, rejects the received message.
- The RSU checks authentication of the received result generated by the proxy vehicle. If the Equation 4 holds, authenticity and integrity of the batch result are checked.

$$\sigma_2 P = ((\textstyle\sum_{i=1}^{d} k_i) + \sigma_1)P_r + \sum_{i=1}^{d} PID_{i,1}, \tag{4}$$

where $\sigma_2 = \sum_{i=1}^{d} s_{i,2}$ and $\sigma_1 = \sum_{i=1}^{d} s_{i,1}$, $k_i = k(m_i, T_i, PID_i)$ for $1 \leq i \leq d$.
The correctness of Equation 4 is verified as follows.

$$\begin{aligned}
&\sigma_2 P = (\textstyle\sum_{i=1}^{d} s_{i,2})P \\
&= \textstyle\sum_{i=1}^{d}[x_r(k(m_i, T_i, PID_i) + s_{i,1}) + \alpha_i]P \\
&= \textstyle\sum_{i=1}^{d}[(k_i + s_{i,1})P_r + PID_{i,1}] \\
&= (\textstyle\sum_{i=1}^{d}(k_i + s_{i,1}))P_r + \sum_{i=1}^{d} PID_{i,1} \\
&= (\textstyle\sum_{i=1}^{d} k_i)P_r + (\sum_{i=1}^{d} s_{i,1})P_r + \sum_{i=1}^{d} PID_{i,1} \\
&= (\textstyle\sum_{i=1}^{d} k_i)P_r + \sigma_1 P_r + \sum_{i=1}^{d} PID_{i,1}
\end{aligned} \tag{5}$$

- If Equation 4 does not hold or Equation 4 holds and $b = 0$, the RSU indicates that the proxy vehicle is malicious, and asks TA to revoke that proxy vehicle. This action avoids disturbing authentication process later.

B. Analysis of the proposed scheme

In this subsection, existential unforgeability of the signature on the batch result, σ_2, is proved in the random oracle model (see [24] 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 proposed scheme is $(\tau, q_k, q_s, \epsilon)$-existentially unforgeable against adaptively chosen message attack in the random oracle model such that*

$$\begin{aligned}
\epsilon_{ECDLP} &\geq \tfrac{\epsilon_1^2}{q_k + q_s} - \tfrac{1}{q}, \\
\tau_{ECDLP} &\leq 2\tau + 2q_s t_m,
\end{aligned} \tag{6}$$

where $\epsilon_1 = \epsilon - \frac{q_s(2q_s + q_k)}{q}$ and t_m is the required for scalar multiplication. In addition, q_k and q_s are the number of queries to oracles $k(.)$ 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 RSU's public key, P_r, to Q and generate the public parameters $Para = \{\mathbb{G}, q, P, P_r\}$ and invokes the adversary A on $Para$. The adversary A runs in time at most τ, makes q_k queries to the random oracle $k(.)$ and q_s queries to the Sign oracle, and can win the unforgeability game with probability at least ϵ. Algorithm

C maintains initially empty associative table $T_k[.]$ to simulate random oracle $k(.)$, 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), then, C returns its value; otherwise, C chooses $T_k[m_i, T_i, PID_i] \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$, and returns $k(m_i, T_i, PID_i)$ to A.
- Sign queries: For a query $(m_i, s_{i,1}, T_i)$ under public key P_r, C chooses two random numbers k_i and $s_{i,2}$ from $\mathbb{Z}_q^*$ and $PID_{i,2}$, computes $PID_{i,1} = -s_{i,2}P + (k_i + s_{i,1})P_r$. If $T_k[m_i, T_i, PID_i]$ has already been defined, then, C halts, returns $\perp$ and sets $bad \leftarrow true$; otherwise, it sets $T_k[m_i, T_i, PID_i] \leftarrow k_i$, and returns the signature $(s_{i,1}, s_{i,2}, k_i, PID_{i,1}, PID_{i,2})$ on the message $(m_i, s_{i,1}, T_i)$ under public key P_r to A.
- Finally, it is assumed that A outputs a forged signature $(s^*_{i,1}, s^*_{i,2}, k^*_i, PID^*_{i,1}, PID^*_{i,2})$ on the message $(m^*_i, s^*_{i,1}, T^*_i)$ under public key P_r. The forgery is non-trivial if A has not made a Sign query on the input of $(m^*_i, s^*_{i,1}, T^*_i)$ under P_r.

The probability of A in returning a forged signature $(s^*_{i,1}, s^*_{i,2}, k^*_i, PID^*_{i,1}, PID^*_{i,2})$ 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 $\Pr[E_1]$ 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 Sign queries. This probability is computed as follows.

Claim 1.$\Pr[E_1] = \Pr[\neg bad] \geq 1 - q_s(\frac{(q_s+q_k)}{q}) - \frac{{q_s}^2}{q}$.

Proof. The probability of the event E_1, $\Pr[\neg bad]$, is multiplication of the following probabilities.

- Case 1. We have $bad \leftarrow true$ if the pair (m_i, T_i, PID_i) generated in a Sign simulation has been occurred by chance in a previous query to the oracle $k(.)$. Since there are at most $q_s + q_k$ entries in the table $T_k[.]$ and the number of $PID_{i,1}$, uniformly distributed in $\mathbb{Z}_q^*$, is $\frac{1}{q}$, the probability of this event for one Sign query is at most $\frac{(q_s+q_k)}{q}$. Hence, the probability of this event for q_s queries is at most $\frac{q_s(q_s+q_k)}{q}$.
- Case 2. We have $bad \leftarrow true$ if C previously used the same randomness $PID_{i,1}$, uniformly distributed in $\mathbb{Z}_q^*$, in one Sign simulation. Since there are at most q_s Sign simulations, this probability is at most $\frac{q_s}{q}$. Therefore, for q_s Sign queries the probability of this event is at most $\frac{q_s^2}{q}$.

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 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 $(s^*_{i,1}, s^*_{i,2}, k^*_i, PID^*_{i,1}, PID^*_{i,2})$ on the message $(m^*_i, s^*_{i,1}, T^*_i)$ under public key pk_r is at least

$$\epsilon_1 = \epsilon - \frac{q_s(2q_s + q_k)}{q}.$$

Then, C runs Forking algorithm [25] to obtain two valid forgeries on the same tuple $(m^*_i, s^*_{i,1}, T^*_i, PID^*_{i,1}, PID^*_{i,2})$ with different values for $k(m_i, T_i, PID_i)$ under P_r as presented in Equation 7.

$$\begin{aligned} s^*_{i,2} &= \gamma(k^*_i + s^*_{i,1}) + \alpha^*_i \bmod q \\ {s'}^*_{i,2} &= \gamma({k'}^*_i + s^*_{i,1}) + \alpha^*_i \bmod q, \end{aligned} \tag{7}$$

where $k^*_i \neq {k'}^*_i$.

As a result, the solution to ECDLP, γ, is computed in Equation 8.

$$\gamma = \frac{s^*_{i,2} - {s'}^*_{i,2}}{k^*_i - {k'}^*_i} \tag{8}$$

The success probability of C according to the generic Forking Lemma of Bellare and Neven [25] is bounded by $\frac{\epsilon_1^2}{q_k+q_s} - \frac{1}{q}$, where $\epsilon_1 = \epsilon - \frac{q_s(2q_s+q_k)}{q}$.

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 and q_s Sign queries. Therefore, C's run-time is $\tau_{ECDLP} \leq 2\tau + 2q_s t_m$. This completes the proof.

□

C. Security analysis

In this subsection, we show that the proposed scheme has the following security requirements.

- Message authentication: The proposed proxy-based authentication scheme provides message integrity and validity of the sender's identity due to the following reasons:
  - Lo and Tsai's identity-based signature scheme with batch verification [17] is used to check authenticity of the received messages from vehicles by proxy vehicles. Also, they showed that their scheme is existentially unforgeable against adaptive chosen message and identity

attack under difficulty of ECDLP problem in the random oracle model. To make it clear, proxy vehicles verify vehicles' signatures, $s_{i,1}$ for $1 \leq i \leq d$ in batch to check message integrity and vehicles' identities. As a consequence, authenticity and validity of vehicle's identity is checked by proxy vehicles. In addition, with employing the small exponent test methodology [13**?**] in the batch verification of multiple messages, our scheme is resistant against false acceptance of invalid signatures.
  - The integrity, validity of proxy vehicle's identity and authenticity of received messages from proxy vehicles which include batch result σ_1 and its corresponding signature σ_2 is also checked by (R_p, s_p), which is existentially unforgeable against adaptively chosen message attacks as proved by Lo and Tsai [17].
  - An RSU can check the correctness of the batch result which is generated by proxy vehicles by verifying σ_2. In addition, security of our proposed signature scheme is proved under under 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 each 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 proposed proxy-based authentication scheme has conditional privacy preserving since the real identity of each vehicle is converted to a pseudo identity by TA, and also the pseudo identity and its corresponding secret key of each vehicle dynamically changes. Without knowing TA's secret key, β, no one can find out the real identity of each vehicle due to the Computational Diffie-Hellman Problem (CDHP). Hence, vehicles and RSUs except for TA cannot extract real identity of a vehicle from its messages.
- Traceability: The TA can find out the real identity, ID_i, of a vehicle with pseudo identity $(PID_{i,1}, PID_{i,2})$ by computing $ID_i = PID_{i,2} \oplus g(\beta PID_{i,1})$. Therefore, TA can trace vehicles from its messages in case of any misbehaviour.
- 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. As a consequence, vehicles and RSUs cannot link two messages sent by the same vehicle.
- Resistance to attacks: Since the used signature schemes are existentially unforgeable against chosen message attack, and our authentication scheme is designed based on those, common attacks such as the impersonation attack, modification attack and the man in the middle attack are prevented. In addition, timestamp T_i for freshness of messages is used to avoid replay attack.

TABLE I
COMPARISON OF COMPUTATION OVERHEADS IN AN RSU

Schemes	Verifying a single message	Verifying n messages
Our Scheme	$5T_{mul}$	$5\lfloor \frac{n}{300} \rfloor T_{mul}$
Liu et al.'s Scheme [20]	$4T_{mul}$ $+5T_p$ $+T_{mtp}$	$2\lfloor \frac{n}{300} \rfloor T_{mul}$ $+(2\lfloor \frac{n}{300} \rfloor + 3)T_p$ $+T_{mtp}$
Lo and Tsai 's Scheme [17]	$3T_{mul}$	$(n+2)T_{mul}$
He et al.'s Scheme [18]	$3T_{mul}$	$(n+2)T_{mul}$

VI. COMPARISONS

A. Computational overhead

Computational overhead of an RSU for our proposed scheme, Liu et al.'s scheme [20], the only proxy-based authentication scheme, and two recently proposed authentication schemes [17, 18] in terms of verifying a single message and n messages 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. In the comparison, it is assumed that each proxy vehicle can verify at most 300 signatures which is reasonable as used in Liu et al.'s scheme [20], and traffic density is the number of signatures, n, in a verification period. Hence, $\lfloor \frac{n}{300} \rfloor$ are the number of proxy vehicles.

According to the experimental results presented by Horng et al. [13] which run on Intel i7 3.07 GHZ machine, T_{mtp}, T_{mul} and T_p take 0.09 ms, 0.39 ms and 3.21 ms, respectively. Since the computational costs of other operations are negligible, they are not considered in the comparison. As a consequence, we have $T_p = 8.2T_{mul}$.

Hence, verifying a single message at an RSU in our scheme costs about $5T_{mul}$, while that in Liu et al.'s scheme [20] costs about $40T_{mul}$. Similarly, verifying n messages, which are sent by $\lfloor \frac{n}{300} \rfloor$ proxy vehicles, at an RSU costs about $5\lfloor \frac{n}{300} \rfloor T_{mul}$, while that costs about $(16\lfloor \frac{n}{300} \rfloor + 24)T_{mul}$ in Liu et al.'s scheme [20]. In addition, as shown in Table I, verifying a single message and n messages in an RSU in Lo and Tsai's scheme [17] and He et al.'s scheme [18] are $3T_{mul}$ and $(n+2)T_{mul}$, respectively. To verify 3000 signatures, the required time in our scheme is 19.5 ms, while this value in Liu et al.'s scheme [20], Lo and Tsai's scheme [17] and He et al.'s

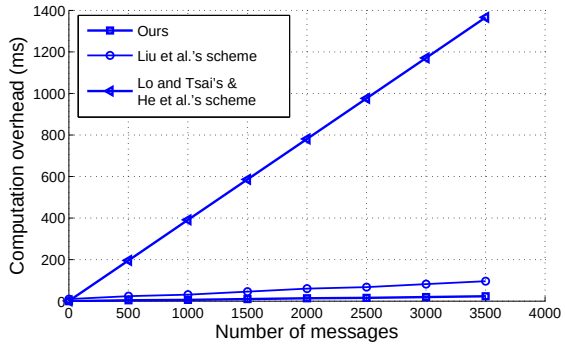

Fig. 1. Comparison of computation overheads in terms of number of messages in an RSU

scheme [18] is approximately 81.6 ms, 1170 ms and 1170 ms, respectively. Therefore, our scheme is more efficient than Liu et al.'s scheme [20] and other efficient schemes [17, 18].

In other words, as shown in Fig.1, the maximum number of the messages can be verified per second in Liu et al.'s scheme [20], Lo and Tsai's scheme [17] and He et al.'s scheme [18] are approximately 25650, 2562 and 2562, respectively, while in our scheme this number reaches to 153846. As a consequence, our proposal is a good candidate to improve computational cost at RSUs when there are a large number of messages in its coverage area compared to previous efficient schemes.

B. Communication overhead

In this subsection, comparison of communication costs of our scheme, Liu et al.'s scheme [20] and two recently proposed schemes; Lo and Tsai's scheme [17] and He et al.'s scheme [18] is given in Table II. 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 sending one message and n messages to an RSU. In the comparison, the size of the message m_i is not considered since they are the same in all authentication schemes. In Liu et al.'s scheme [20], the signature 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 $40 \times 4 + 4 = 164$ bytes, while in our scheme the signature sent by V2PV is $(PID_{i,1}, PID_{i,2}, T_i, R_i, s_{i,1}, s_{i,2})$, where $PID_{i,1}$, $PID_{i,2}$ and $R_i \in \mathbb{G}$ and $s_{i,1}$ and $s_{i,2} \in \mathbb{Z}_q^*$, and its size is $40 \times 3 + 2 \times 20 + 4 = 164$.

Note that it is assumed that each proxy vehicle verifies 300 signatures, so the number of proxy vehicles to verify n signatures are $\lfloor \frac{n}{300} \rfloor$. In Liu et al.'s scheme [20], the message sent by a proxy vehicle to an RSU (PV2R) is $(PID_{p,1}, PID_{p,2}, T_p, s_{p,1}, \sigma_1, \sigma_2, PID_{i,1}, PID_{i,2}, T_i, 1 \leq i \leq n)$, 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 signatures by $\lfloor \frac{n}{300} \rfloor$ proxy vehicles is $(40 \times 5 + 4)\lfloor \frac{n}{300} \rfloor + (2 \times 40 + 4)n = 204\lfloor \frac{n}{300} \rfloor + 84n$ bytes, while in our scheme the transmitted message is $(PID_{p,1}, PID_{p,2}, T_p, R_p, s_{p,1}, \sigma_1, \sigma_2, PID_{i,1}, PID_{i,2}, T_i, 1 \leq i \leq n)$, where $PID_{p,1}$, $PID_{p,2}$, $PID_{i,1}$, $PID_{i,2}$ and $R_p \in \mathbb{G}$, $s_{p,1}$, σ_1 and $\sigma_2 \in \mathbb{Z}_q^*$, and its size is $(40{\times}3{+}3{\times}20{+}4)\lfloor \frac{n}{300} \rfloor{+}(2{\times}40{+}4)n = 184\lfloor \frac{n}{300} \rfloor{+}84n$ bytes. By the same analysis, the signature size sent by vehicles to an RSU for one single and n messages in both Lo and Tsai's [17] and He et al.'s schemes [18] are 144 and $144n$ bytes, respectively.

TABLE II
COMPARISON OF COMMUNICATION OVERHEAD (IN BYTES)

Schemes	Sending a single message	Sending n messages
Liu et al.'s Scheme [20]	164	$204\lfloor \frac{n}{300} \rfloor + 84n$
Lo and Tsai's Scheme [17]	144	$144n$
He et al.'s [18] Scheme	144	$144n$
Our Scheme	164	$184\lfloor \frac{n}{300} \rfloor + 84n$

To send 3000 signatures to an RSU, the signature size in our scheme is 253840 bytes, while this value in Liu et al.'s scheme [20], Lo and Tsai's [17] and He et al.'s schemes [18] is 272400, 432000 and 432000 bytes, respectively. As a consequence, our scheme has a better communication overhead compared to the previous authentication schemes.

VII. CONCLUSION

In this paper, we showed that Liu et al.'s proxy-based authentication scheme [20] for VANETs has some security drawbacks: it is not resistant against modification and false acceptance of the batch result attacks. Then, to tackle security weaknesses of Liu et al.'s scheme, we proposed a new proxy-based authentication scheme for vehicular networks. To show that it is secure against modification attack, we proved that the underlying signatures in the scheme is secure against adaptively chosen message attack under ECDLP problem in the random oracle model. As shown in the comparison, our proposed scheme not only is more efficient than Liu et al.'s scheme, but also it has a better communication overhead compared to Liu et al.'s scheme. We should emphasize that this proposal is useful where there are a large number of vehicles in the coverage area of an RSU, and it was shown from the analysis the required time to verify 3000 signatures in one second of our scheme was improved by 75% and 98% compared to Liu et al.'s proxy-based

authentication scheme [20] and the two recently efficient authentication schemes [17, 18], respectively.

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, 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.

[10] 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.

[11] 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.

[12] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Network*, vol. 19, no. 6, p. 14411449, 2013.

[13] 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, p. 18601875, 2013.

[14] K.-A. Shim, "CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, p. 18741883, 2012.

[15] 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, p. 355362, 2014.

[16] 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.

[17] 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.

[18] 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.

[19] 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, p. 25592564, 2014.

[20] 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, p. 36973710, 2015.

[21] 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.

[22] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[23] 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.

[24] 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.

[25] M. Bellare and G. Neven, “Multi-signatures in the plain public-key model and a general forking lemma,” in *Proc. of the 13th ACM Conf. on Computer and Communications Security-CCS 2006*. New York, NY, USA: ACM, 30 October-3 November 2006, pp. 390–399.