Wireless Netw (2011) 17:1851–1865
DOI 10.1007/s11276-011-0383-2

On batch verification with group testing for vehicular communications

Chenxi Zhang · Pin-Han Ho · Janos Tapolcai

Published online: 14 September 2011

Abstract In this paper, an efficient identity-based batch signature verification scheme is proposed for vehicular communications. With the proposed scheme, vehicles can verify a batch of signatures once instead of in a one-by-one manner. Hence the message verification speed can be tremendously increased. To identify invalid signatures in a batch of signatures, this paper adopts group testing technique, which can find the invalid signatures with few number of batch verifications. In addition, a trust authority in our scheme is capable of tracing a vehicle's real identity from its pseudo identity, and therefore conditional privacy preserving can also be achieved. Moreover, since identity-based cryptography is employed in the scheme to generate private keys for pseudo identities, certificates are not required and thus transmission overhead can be significantly reduced.

Keywords Vehicular ad hoc networks · Security · Scalability · Batch verification · Group testing

Part of this paper was presented in Infocom 2008 [1].

C. Zhang (✉) · P.-H. Ho
Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada
e-mail: c14zhang@uwaterloo.ca; c14zhang@engmail.uwaterloo.ca

P.-H. Ho
e-mail: p4ho@uwaterloo.ca

J. Tapolcai
Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudosok krt. 2, H-1117 Budapest, Hungary
e-mail: tapolcai@tmit.bme.hu

1 Introduction

Wireless communication technologies have been used to facilitate our transportation system by making vehicles more intelligent. Car manufactures and telecommunication industries are gearing up to equip each vehicle with devices, called on-board units (OBUs [17]) that allow vehicles to communicate with each other as well as with the roadside unit (RSU) or infrastructure [1]. These "talk-enable" vehicles and RSUs self-organize a novel network, a Vehicular Ad-hoc Network (VANET), which has two types of communications: vehicle to vehicle (V2V) communication and vehicle to RSU (V2R) communication.

VANETs provide us many promising applications, which are in general grouped into two categories: safety related applications and non-safety related applications. To achieve safety related applications, Dedicated Short Range Communications (DSRC) [5] protocol requires each vehicle in VANETs broadcast a traffic related message every 100–300 ms. The message includes a vehicle's instant driving status information, such as location, speed, turning intention, and driving status (e.g., regular driving, waiting for a traffic light, traffic jam, etc.). Facilitated by these messages, vehicles can be aware of their neighboring vehicles' driving behavior in real time. Therefore potential collisions or accidents can be alerted and might be avoided under the assistance of warning messages sent from other vehicles.

VANETs also provide us many promising non-safety related applications. The first is Location Based Service (LBS). Vehicles on the road may send RSUs a request asking the closest location information, such as the closest gas station, shopping center, coffee shop, etc. RSUs that connect with a location server respond vehicles with the related location information. The second is traffic

management. RSUs that are pervasively located in a city can real-time collect and monitor traffic flow information, which can be used to assist us in predicting traffic congestion and controlling traffic signals. The third is Internet access providing. Vehicles can download/upload data information such as mp3/email through RSUs. In addition, a VANET can also be used as a vehicle-based Delay Tolerant Networks (DTN), which takes advantage of RSUs and vehicles to buffer and distribute data information.

Even though VANETs provide us many promising applications, some challenging security and privacy issues in VANETs have been identified [2–18], which have to be well addressed before VANETs can be put into practical use. To ensure both identity authentication and message integrity in VANETs, one appealing solution is to sign each message with a digital signature before the message is sent. However, conventional signature schemes that verify the received messages one after another may fail to satisfy the stringent time requirement for safety related applications. Note that a vehicle could communicate with hundreds of vehicles at the same time, each sending a safety related message every 100–300 ms. Although only one signature is received at any time through DSRC transmission, there are still a large number of signatures buffered at receivers. It's because the time consuming on the DSRC transmission of a message is much shorter than the time consuming on verifying a signature [46]. In this case, verifying a large number of signatures sequentially could take a long time and will become the processing bottleneck at each vehicle. Many useful safety messages will have to be discarded because they cannot be verified within the accepted time range. The verification bottleneck also happens at RSUs in non-safety related applications when an RSU receives hundreds of forwarding messages per seconds. Therefore, the fast signature verification in VANETs is a tough requirement for any current digital signature scheme. Furthermore, the maintenance of public key certificates under the traditional Public Key Infrastructure (PKI) also incurs huge communication overhead. In addition, conditional privacy also needs to be taken into consideration as well.

To deal with the aforementioned challenges of security and privacy issues, this paper introduces an efficient batch signature verification scheme for VANETs. The proposed scheme has the following features: (1) A batch of signatures can be verified once instead of one after another. As such, the message verification speed can be tremendously increased. (2) Identity-based cryptography is employed, and thus efforts on certificate management are alleviated and the transmission overhead is reduced significantly. (3) Conditional privacy is achieved, in which a distinct pseudo identity and the corresponding private key are generated for each message. A trusted authority is able to trace the real identity of a vehicle from any of its pseudo identities. (4) To find invalid signatures in a batch of signatures, this paper investigates and adopts some group testing approaches which can find invalid signatures efficiently.

The remainder of the paper is organized as follows. In Sect. 2, background and preliminary knowledge related to the proposed research are given, including the network model, pairing technique, batch verification, and security requirements. In Sect. 3, the proposed batch verification scheme is described in details, and the security of the proposed scheme is analyzed. In Sect. 4, the group testing technique is introduced and adopted. In Sect. 5, the performance evaluation is presented. Section 6 surveys the related work. Finally, Sect. 7 concludes the paper.

2 Background and preliminaries

2.1 Network model

We introduce a two-layer vehicular network model. The lower layer is composed of vehicles and RSUs as shown in Fig. 1. The communication among them is based on the DSRC protocol. Each vehicle has its own public keys and private keys, with which all messages are signed and then sent to its neighboring RSU. Each vehicle receiving the traffic related or non-safety related messages is responsible for verifying their digital signatures.

In general, the top layer is comprised of a Trust Authority (TA) and application servers (such as traffic control analysis center) for non-safety applications. Vehicles communicate with TA off-line and occasionally. TA is responsible for tracing the real identity of vehicles in case that an abuse happens. We assume that the TA is always trusted and can never be compromised, which is also responsible for assigning master private keys for vehicles.

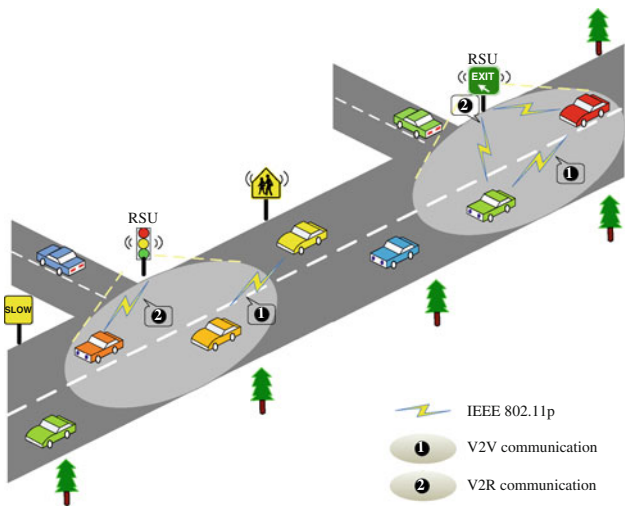


Fig. 1 The network model

The application server is responsible for non-safety related applications, in which RSUs are responsible for storing and forwarding messages, for example in vehicle-based DTN applications. In addition, application server can also aid to gather and analyze the traffic density of a whole city, and predict the traffic distribution in order to optimize the traffic light control.

2.2 Security requirements

The V2V and V2R communication scenarios are subject to the following three security requirements: *message authentication*, *identity privacy preserving*, and *traceability*, which are further discussed as below.

2.2.1 Message authentication

Messages in VANETs have to be authenticated to confirm that they are indeed sent unaltered by legitimate entities, vehicles or RSUs. In addition, when the number of vehicle increases, the speed of entities for signature verification should be faster in order to avoid any possible performance bottleneck.

2.2.2 Identity privacy preserving

In vehicular communication, due to its broadcasting nature, overhearing an identity-specific information could happen frequently. If the employed signature scheme is an ordinary digital signature, the signature would easily leak one's identity information [19]. Even though a pseudo identity is employed as a mask, an outside observer can also link multiple signatures to one vehicle through traffic analysis. This issue is called linkability, which may incur a location privacy violation problem [20]. Therefore, identity privacy preserving is required.

2.2.3 Traceability

The TA should have the ability to retrieve a vehicle's real identity from its pseudo identity when the signature is in dispute or when the content of a message is bogus.

In this paper, we aim to address all the aforementioned issues.

2.3 Bilinear maps

Since bilinear maps work as the basis of our proposed scheme in this paper, we briefly introduce the bilinear maps in this section.

Let $\mathbb{G}$ be a cyclic additive group generated by P, and $\mathbb{G}_T$ be a cyclic multiplicative group. $\mathbb{G}$ and $\mathbb{G}_T$ have the same prime order q, i.e., $|\mathbb{G}| = |\mathbb{G}_T| = q$. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be an bilinear map, which satisfies the following properties:

- *Bilinear*: For all $P, Q, R \in \mathbb{G}$, and $a, b \in \mathbb{Z}_q^*$, $\hat{e}(Q, P+R) = \hat{e}(P+R, Q) = \hat{e}(P, Q) \cdot \hat{e}(R, Q)$. In particular, $\hat{e}(aP, bP) = \hat{e}(P, bP)^a = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab}$.
- *Non-degenerate*: There exist $P, Q \in \mathbb{G}$ such that $\hat{e}(P, Q) \neq 1_{\mathbb{G}_T}$.
- *Computable*: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}$.

Such an bilinear map $\hat{e}$ is called an admissible pairing, and can be constructed by the modified Weil [21] or Tate pairings [22] on elliptic curves. The group that possesses such a map $\hat{e}$ is called a bilinear group, on which the Decisional Diffie-Hellman (DDH) problem is easy to solve while the Computational Diffie-Hellman (CDH) problem is believed hard [23]. For example, given $P, aP, bP, cP \in \mathbb{G}$ and any $a, b, c \in \mathbb{Z}_q^*$, there exists an efficient algorithm to determine whether $ab = c \bmod q$ by checking $\hat{e}(aP, bP) \stackrel{?}{=} \hat{e}(P, cP)$, while there exists no algorithm that can compute $abP \in \mathbb{G}$ with non-negligible probability within polynomial time.

2.4 Batch verification

With the pervasiveness of telecommunication applications, the demand and requirement on authentication for communication security become more stringent. The delay caused by verification of a bulk of signatures may dramatically impede transmission throughput and impair the system applicability. In order to speed up the process of verification, a batch verification scheme should be a good alternative solution since it can verify all the signatures received in a time window with rather short time compared to verify each signature one after the other. The batch cryptography based on RSA was introduced by Fiat [24] in 1989. Some other batch signature schemes were proposed later [25–29]. The latest batch verification scheme proposed in [30] is based on the CL signature scheme [31], and is the first solution on batch verification without using random oracles, in which the computation efficiency can be significantly improved. For instance, 3 pairing operations are required to verify a single signature. With the batch verification scheme of [30], verifying n signatures also takes 3 pairing operations instead of $3n$ pairing operations. In other words, the verification time of the dominant operation (i.e., paring) is independent of the number of signatures to verify. Therefore, the batch verification can dramatically decrease the time spent on verifying a large number of signatures, which can achieve much better scalability. In this paper, we propose an efficient

identity-based batch verification scheme based on the improved CL signature scheme in [30].

3 Batch verification for vehicular communications

In this section, we propose a novel Identity-based Batch Verification (IBV) scheme for traffic related message transmission. The proposed scheme includes the following four phases: the key generation and pre-distribution phase, the pseudo identity and private key generation phase, the message signing phase, and the batch verification phase. The notations throughout this paper are listed in Table 1.

3.1 Key generation and pre-distribution

Firstly, let each vehicle be equipped with a tamper-proof device, which is secure against any compromise attempt in any circumstance. With the tamper-proof device on vehicles, an adversary cannot extract any data stored in the device including key material, data, and code [6, 8]. We assume that there is a TA which is in charge of checking the vehicle's identity, and generating and pre-distributing the private master keys of the vehicles. Prior to the network deployment, the TA sets up the system parameters for each vehicle and RSU as follows:

- Let $\mathbb{G}$ be a cyclic additive group generated by P, $\mathbb{G}_T$ be a cyclic multiplicative group, and $\mathbb{G}$ and $\mathbb{G}_T$ have the same order q. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map.
- TA first randomly chooses $s_1, s_2 \in \mathbb{Z}_q^*$ as its two master keys, and computes $P_{pub1} = s_1 P$, $P_{pub2} = s_2 P$ as its public keys. These two master keys of the TA are then loaded in the vehicles' tamper-proof device.
- Each RSU and vehicle are preloaded with the public parameters $\{\mathbb{G}, \mathbb{G}_T, q, P, P_{pub1}, P_{pub2}\}$. In addition, the tamper-proof device of each vehicle is preloaded with the parameters $\{s_1, s_2\}$.
- To activate the tamper-proof device, each vehicle is assigned with a real identity, denoted as $RID \in \mathbb{G}$, and a password, denoted as *PWD*, where the *RID* uniquely identifies the vehicle, while the *PWD* is required in the authentication process by the tamper-proof device. Therefore, an adversary cannot take advantages of the tamper-proof device even if the vehicle is stolen.

Table 1 Notations

Notation	Descriptions
V_i	The ith vehicle
RSU	A roadside unit
TA	A trust authority
$\mathbb{G}$	A cyclic additive group
$\mathbb{G}_T$	A cyclic multiplicative group
P	The generator of the cyclic additive group $\mathbb{G}$
$\hat{e}$	A bilinear map: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$
q	The order of the group $\mathbb{G}$
r	A random nonce
s_i	The ith private master key of the tamper-proofdevice, where i is equal to 1 or 2
P_{pubi}	The ith public key of the TA, where i is equal to 1 or 2
RID	The real identity of the vehicle
PWD	A password or authentication credential used to activate a tamper-proof device
ID^i	A pseudo identity of the vehicle V_i
ID^i_j:	A part of the ID^i, such that $ID^i = (ID^i_1, ID^i_2)$
SK_i	A private key of the vehicle V_i
SK^i_j:	A part of the SK^i, such that $SK^i = (SK^i_1, SK^i_2)$
$\mathcal{M}_i$	A message sent by the vehicle V_i
$h(.)$	A one-way hash function such that SHA-1 [32]
$H(.)$	A MapToPoint hash [23] function such as $H : \{0,1\}^* \rightarrow \mathbb{G}$
$\|\|$	Message concatenation operation, which appends several messages together in a special format

3.2 Pseudo identity generation

To achieve privacy preservation, we exploit to use the tamper-proof device, which is responsible for generating random pseudo identities and corresponding private keys based on identity-based cryptography [21]. The tamper-proof device is composed of three secure modules: an authentication module, a pseudo identity generation module, and a private key generation module as shown in Fig. 2, which are further described in details as follows.

3.2.1 Authentication module

The authentication module works as an access control mechanism. A vehicle inputs its unique real identity *RID* and the password *PWD* to initiate the device, where the *PWD* can be the signature of the *RID* signed by the TA. If the *RID* and *PWD* successfully pass the verification of the authentication module, the *RID* is delivered to the next module, the pseudo identity generation module. Otherwise, the device denies providing services for the vehicle.

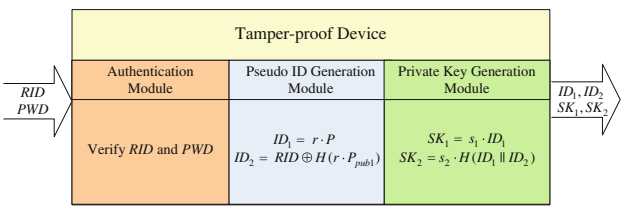


Fig. 2 The tamper-proof device

Obviously, the authentication module enhances the security of the tamper-proof device since a malicious adversary cannot take advantages of it even though the tamper-proof device is physically held by the adversary.

3.2.2 Pseudo identity generation module

This module is responsible for generating a list of random pseudo identities from the authenticated *RID*. Each pseudo identity *ID* is composed of ID_1 and ID_2. In this module, the ElGamal encryption algorithm [33] over the ECC [34] is employed to encrypt the *RID* as shown in Fig. 2. The two items of the cipher texts are taken as ID_1 and ID_2, respectively. In other words, we have $ID_1 = rP$, and $ID_2 = RID \oplus H(rP_{pub1})$, where r is a random nonce. r is changed each time and guarantees the distinction of ID_1 and ID_2 for each pseudo *ID*. $\oplus$ is an Exclusive-OR (XOR) operation. Here, P and P_{pub1} are the public parameters preloaded by the TA. After the encryption, ID_1 and ID_2 are delivered to the private key generation module.

3.2.3 Private key generation module

In this module, identity-based cryptography [21] is employed. Since a pseudo identity has two parts (i.e., ID_1 and ID_2), the private key generation module is responsible for computing a private key based on ID_1 and ID_2. Thus, the resultant private key also contains two parts, which are denoted as SK_1 and SK_2, respectively. As shown in Fig. 2, SK_1 and SK_2 are equal to $s_1\ ID_1$ and $s_2 H(ID_1 \| ID_2)$, respectively.

Finally, a vehicle can obtain a list of pseudo identities $ID = (ID_1, ID_2)$ along with the corresponding private keys $SK = (SK_1, SK_2)$. Note that the pseudo identities and the private keys can be generated offline by the tamper-proof device; thus, no delay will be caused in the signing messages at a vehicle side due to this process.

3.3 Message signing

When vehicles are traveling on the road, they periodically broadcast traffic related information that could be extremely vital and life-critical information for neighboring drivers. To ensure the integrity of the messages, each message sent by a vehicle should be signed and verified when being received. With the proposed IBV scheme, the message signing phase is presented as follows.

- A vehicle, denoted by V_i, first generates the traffic related message denoted by $\mathcal{M}_i$.
- V_i picks a pseudo identity $ID^i = (ID_1^i, ID_j^i)$ and the corresponding private key $SK^i = (SK_1^i, SK_2^i)$ by way of the tamper-proof device.
- With the private key $SK^i = (SK_1^i, SK_2^i)$, V_i can compute the signature σ_i of the message $\mathcal{M}_i$, where

$$\sigma_i = SK_1^i + h(\mathcal{M}_i) SK_2^i.$$

- Subsequently, V_i sends the final message $\langle ID^i, \mathcal{M}_i, \sigma_i \rangle$ to its neighboring RSU.
- These steps are repeated every 100–300 ms according to the DSRC [5].

The signature of the proposed IBV scheme has the following merits. Firstly, the signature overhead is very low. Compared with the ECDSA signature scheme of IEEE1609.2 [35], which is the current standard for VANETs, the length of a signature in the IBV scheme is a half of that of the ECDSA, i.e., $|\sigma_i| = 161\text{bits} \approx 21$ bytes.[1] However, the IBV scheme does not need any signature certificate to be sent along with the message due to the adoption of identity-based cryptography; instead, only a short-length pseudo identity is sent, which is of a length 42 bytes, i.e., $|ID^i| = |ID_j^i| + |ID_2^i| = 42$ bytes. In contrast, the ECDSA scheme has to incorporate a certificate in the message, which is 125 bytes long in the case of using the certificate presented in IEEE 1609.2 Standard [35]. We will further compare our proposed IBV scheme with the ECDSA scheme in terms of the communication overhead in Sect. 5.

Secondly, from the perspective of signing speed, the proposed IBV scheme does not add any extra signature generation delay compared with that in ECDSA, where both of them need two multiplication operations on an elliptic curve. At last, the signature of the IBV scheme does not leak any real identity information of the vehicle because a pseudo identity is used in the scheme. Furthermore, since all the messages are signed with different pseudo identities, thus none of the two messages can be connected to a single vehicle with the IBV signature scheme, which is expected to successfully address the issue of privacy preservation in VANETs.

3.4 Batch verification

Based on the network architecture as described in Sect. 2, once a vehicle receives traffic related messages from other vehicles, the vehicle has to verify the signatures of the messages to ensure that the corresponding vehicles are not attempting to impersonate any other legitimate vehicles or disseminating bogus messages, which may result in tremendous impairment. For ease of presentation, we first introduce the single signature verification process, followed by the presentation on the batch verification of

[1] Note that with the IBV scheme, in order to get a short signature, we use an MNT curve [41] with 160-bit q, where the bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is asymmetric, $\mathbb{G}_1 \neq \mathbb{G}_2$, and elements in $\mathbb{G}_1$ are 161 bits long.

multiple signatures signed by distinct vehicles on different messages.

3.4.1 Single signature verification

Given the system public parameters $\{\mathbb{G}, \mathbb{G}_T, q, P, P_{pub1}, P_{pub2}\}$ assigned by the TA and the message $\langle ID^i, \mathcal{M}_i, \sigma_i \rangle$ sent by the vehicle V_i, the signature σ_i is valid if $\hat{e}(\sigma_i, P) = \hat{e}(ID_1^i, P_{pub1}) \cdot \hat{e}(h(\mathcal{M}_i)H(ID_1^i||ID_2^i), P_{pub2})$, as verified below.

$$\begin{aligned}
&\hat{e}(\sigma_i, P) \\
&= \hat{e}(SK_1^i + h(\mathcal{M}_i)SK_2^i, P) \\
&= \hat{e}(SK_1^i, P)\hat{e}(h(\mathcal{M}_i)SK_2^i, P) \\
&= \hat{e}(s_1 ID_1, P)\hat{e}(h(\mathcal{M}_i)s_2 H(ID_1^i||ID_2^i), P) \\
&= \hat{e}(ID_1^i, s_1 P)\hat{e}(h(\mathcal{M}_i)H(ID_1^i||ID_2^i), s_2 P) \\
&= \hat{e}(ID_1^i, P_{pub1})\hat{e}(h(\mathcal{M}_i)H(ID_1^i||ID_2^i), P_{pub2}) \qquad (1)
\end{aligned}$$

Therefore, the computation cost by a vehicle for verifying a single signature is dominantly comprised of three pairing operations, one multiplication, one MapToPoint hash [23]. Note that the computation cost of a pairing operation is much higher than the cost of a multiplication and a MapToPoint hash operation.

3.4.2 Batch verification

Given n distinct messages denoted as $\langle ID^1, \mathcal{M}_1, \sigma_1 \rangle, \langle ID^2, \mathcal{M}_2, \sigma_2 \rangle, \ldots, \langle ID^n, \mathcal{M}_n, \sigma_n \rangle$, respectively, which are sent by n distinct vehicles denoted as $V_1, V_2, \ldots, V_n$, all signatures, denoted as $\sigma_1, \sigma_2, \ldots, \sigma_n$, are valid if $\hat{e}(\sum_{i=1}^n \sigma_i, P) = \hat{e}(\sum_{i=1}^n ID_1^i, P_{pub1}) \cdot \hat{e}(\sum_{i=1}^n h(\mathcal{M}_i)H(ID_1^i||ID_2^i), P_{pub2})$. Let HID^i denote $H(ID_1^i \| ID_2^i)$. This batch verification equation follows since

$$\begin{aligned}
&\hat{e}\left(\sum_{i=1}^n \sigma_i, P\right) \\
&= \hat{e}\left(\sum_{i=1}^n (SK_1^i + h(\mathcal{M}_i)SK_2^i), P\right) \\
&= \hat{e}\left(\sum_{i=1}^n SK_1^i, P\right)\hat{e}\left(\sum_{i=1}^n h(\mathcal{M}_i)SK_2^i, P\right) \\
&= \hat{e}\left(\sum_{i=1}^n s_1 ID_1^i, P\right)\hat{e}\left(\sum_{i=1}^n s_2 h(\mathcal{M}_i)HID^i, P\right) \\
&= \hat{e}\left(\sum_{i=1}^n ID_1^i, s_1 P\right)\hat{e}\left(\sum_{i=1}^n h(\mathcal{M}_i)HID^i, s_2 P\right) \\
&= \hat{e}\left(\sum_{i=1}^n ID_1^i, P_{pub1}\right)\hat{e}\left(\sum_{i=1}^n h(\mathcal{M}_i)HID^i, P_{pub2}\right). \qquad (2)
\end{aligned}$$

Thus, this batch verification can dramatically reduce the verification delay, particularly when verifying a large number of signatures. From the above batch verification equation, the computation cost that a vehicle spends on verifying n signatures is dominantly comprised of 3 pairings, n multiplication, n MapToPoint hash, $3n$ addition, and n one-way hash operations. This appealing property demonstrates that the verification time for multiple signatures is constant regardless of the size of the batch. Thus, the time for a vehicle to verify a large number of signatures sent by the surrounding vehicles can be dramatically reduced, which can apparently reduce the message loss ratio due to the potential bottleneck of signature verification for vehicles.

Another advantage of IBV is that it can aggregate multiple signatures as one signature. This promising feature is not directly used in our safety related application, but it is used to non-safety related delay and forwarding applications to reduce communication overhead in VANETs. In our scheme, given n distinct signatures, $\sigma_1, \sigma_2, \ldots, \sigma_n$, the aggregate signature is equal to $\sum_{i=1}^n \sigma_i$. Further more, compared with BLS [23], our scheme does not require that n distinct messages have to be sent from the same sender.

3.5 Security analysis

In this section, we analyze the security of the proposed batch verification scheme in terms of the following three aspects: the message authentication, the user identity privacy preservation, the traceability, key management by the TA.

- *Message authentication*. The message authentication is one of the basic security requirements in vehicular communications. In the proposed IBV scheme, the signature $\sigma_i = SK_1 + h(\mathcal{M})SK_2$ is actually a one-time identity-based signature. Without knowing the private key SK_1 and SK_2, it is infeasible to forge a valid signature. Because of the NP-hard computation complexity of Diffie-Hellman problem in $\mathbb{G}$, it is difficult to derive the private keys SK_1 and SK_2 by way of ID_1, P_{pub1}, P, and $H(ID_1\|ID_2)$. At the same time, because $\sigma_i = SK_1 + h(\mathcal{M})SK_2$ is a Diophantine equation, by only knowing σ and $h(\mathcal{M})$, it is still difficult to get the private keys SK_1 and SK_2. Therefore, the one-time identity-based signature is unforgeable, and the property of message authentication is achieved.
- *Identity privacy preserving*. In the proposed scheme, the real identity RID of a vehicle is converted into two random pseudo identities ID_1 and ID_2, where $ID_1 = rP$ and $ID_2 = RID \oplus H(rP_{pub})$ for unknown r. Note that the pseudo identity pair (ID_1, ID_2) is actually an

ElGamal-type ciphertext, which is semanticly secure under the chosen plaintext attacks. Therefore, without knowing the *master-key* (s_1, s_2), it is infeasible for anyone to tell the real identity from the pseudo identity pair. Also, the linkability does not exist because the pseudo identities (ID_1, ID_2) in each signature instance is distinct. Therefore, the identity privacy preservation can be guaranteed.

- *Traceability*. Given the pseudo identity pair ID_1 and ID_2, only the TA, given the *master-key* (s_1, s_2), can trace the real identity of the vehicle by computing $ID_2 \oplus H(s_1 ID_1) = RID \oplus H(rP_{pub}) \oplus H(s_1 rP) = RID$. Therefore, once a signature is in dispute, the TA has the ability to trace the vehicle from the disputed message, in which the traceability can be well satisfied.
- *Key management*. The security system relies on the master keys s_1, s_2. If TA plans to update the master keys (e.g., increase the length of keys) to enhance the system security, TA can replace drivers' old tamper-proof devices with new ones when they renew their license plates or renew driver licenses.

4 False signature detection with group testing technique

It is clear that the proposed batch verification scheme can significantly accelerate the overall signature verification when no false signature is found. When an error is identified in the verification, the false signatures should be identified in the batch, which can be simply done by sequentially verifying each signature using Eq. 1. However, sequential verification obviously causes long delay, particularly when an attacker who attempts to ruin the batch verification periodically sends a small number of invalid signatures. Note that an attacker can use invalid identities when sending each message in order to prevent the TA from tracing its real identity.

To improve the efficiency of false signature identification, the paper investigates group testing techniques for resolving this problem.

4.1 Employment of group testing algorithm

Group testing were motivated by the task to testing blood samplers of draftees to detect syphilis in the World War II. In the application, each draftee was taken a blood sample. There were millions of draftees, and only a few thousand of draftees had the syphilis disease. A single test on the combination of multiple blood samples returned positive if at lease one sample was positive. A positive combination was divided and then further tested with other positive samples. On the other hand, a single test on the combination of multiple samples returned negative if any of the samples was negative. A negative combination can save many individual tests.

The group testing technique was to find an efficient strategy to combine blood samples, aiming to identify positive blood samples with as few number of tests as possible. In our application, the objective of group testing is to find invalid signatures with the minimal number of batch verification.

The task of false signature identification from a batch containing at least one false signature (or termed a "bad batch" in the following context) is formulated as a group testing problem. With Eq. 1, it takes 3 pairings and n operations of $h(\mathcal{M}) \cdot HID$ to verify n signatures. Each of $h(\mathcal{M}) \cdot HID$ takes a MapToPoint hash and a multiplication. It is worth noting that the MapToPoint hash and multiplication can be pre-computed and stored in memory for reuse. Once a batch verification on n signatures is launched, n operations of $h(\mathcal{M}) \cdot HID$ are performed. If later a batch verification on any subset of the n signatures is needed, the MapToPoint hash and a multiplication on $h(\mathcal{M}) \cdot HID$ do not need to be computed again. In this case, only 3 pairing computations are needed for a batch verification (test), and the time of testing the validity of multiple signatures is equal to that of testing a single signature.

Many constructions for adaptive combinatorial group testing were reported in the literature. However, it is difficult to find an optimal algorithm of group testing for a general purpose because the computational complexity of group testing has not been determined [36]. Adaptive group testing algorithms for finding invalid signatures were summarized in [40], and can be generally divided into the following four types: individual testing, binary search, generalized binary splitting (GBS) [36], Li's s-stage [36]. For each algorithm, the number of tests in the worse case is summarized in Table 2. Throughout this section, let n denote the number of signatures to be verified in a batch, and d be the upper bound on the estimated number of invalid signatures.

It is a challenging task to find an appropriate function due to the complexity of the functions $d - 1 + \lceil \log(nd) \rceil$ and $\frac{e}{\log(e)} \cdot d \cdot \log(\frac{n}{d})$. Moreover, the optimal algorithm depends on the values of n and d. Nevertheless, it is convenient to analyze the values of the above functions when a parameter (i.e., d) changes and the other parameter (i.e., n) is fixed. Figure 3 shows the number of required tests (batch verifications) as d changes and n is fixed. In Fig. 3(a)–(d), n is equal to 100, 200, 300, and 400, respectively. It is clear that the function of Li's s-stage and the function of GBS always meet at a certain point, as represented as Point 1 in Fig. 3. When d is less than the x coordinateTo ease our presentation, Point 1 and Point 2, respectively represent the

Table 2 Different adaptive group testing algorithm comparison [40]

Algorithm	Tests (worst case)
Individual testing	$n-1$
Binary search	$d\lceil\log(n)\rceil$
Generalized binary splitting	$d-1+\lceil\log(nd)\rceil$
Li's s-stage	$\frac{e}{\log(e)}\cdot d\cdot\log(\frac{n}{d})$

x coordinate of Point 1 and x coordinate of Point 2 throughout this paper. of Point 1, GBS always has the optimal (minimal) function value. In addition, the function of Li's s-stage and the function of Individual testing always meet at another point that is represented as Point 2. When d is less than Point 1 and larger than Point 2, Li's s-stage always has the optimal function value. When d is larger than Point 2, the Individual testing always has the optimal function value.

In Fig. 3, n is only set to four values, i.e., 100, 200, 300, 400. For better analyzing the relationship between Point 1 (Point 2) and n, more value of n are selected. Given each n, the values of Point 1 and Point 2 are computed, as they are computed in Fig. 3(a)–(d). As such, a set of values of Point 1 and Point 2 can be obtained. Figure 4(a) shows the set of the values of Point 1 and Point 2 given different values of n. As we can see, Point 1 and Point 2 increase linearly as n increases. Thus, it is reasonable to use two linear functions to represent the Point 1 set and Point 2 set. Figure 4(b) shows the two fitting functions: the function $y_1 = 0.17n + 1.31$ fits the Point 1 set; the function $y_2 = 0.34n + 0.44$ fits the Point 2 set. The two lines divide the plane of Fig. 4(b) into three areas. Each area represents a desired group testing algorithm to be used. Therefore, given n and d, an optimal group testing algorithm can be selected. For example, given $n = 200$ and $d = 10$, the point (200, 10) is in the Area 3 in Fig. 4(b). In this case, GBS is the optimal group testing algorithm. Given $n = 300$ and $d = 60$, the point (300, 60) is in the Area 2 in Fig. 4(b). In this case, Li's s-stage is the optimal group testing algorithm. Given $n = 100$ and $d = 60$, the point (100, 60) is in the Area 1 in Fig. 4(b). In this case, the Individual testing is the optimal group testing algorithm. The more generalized solution to choose the optimal algorithm is given below in accordance with the values of d and n.

$$\begin{cases} \text{GBS}, & d \le 0.17n + 1.31 \\ \text{Li's s-stage}, & 0.34n + 0.44 \ge d > 0.17n + 1.31 \\ \text{Individual testing}, & d > 0.34n + 0.44 \end{cases}$$

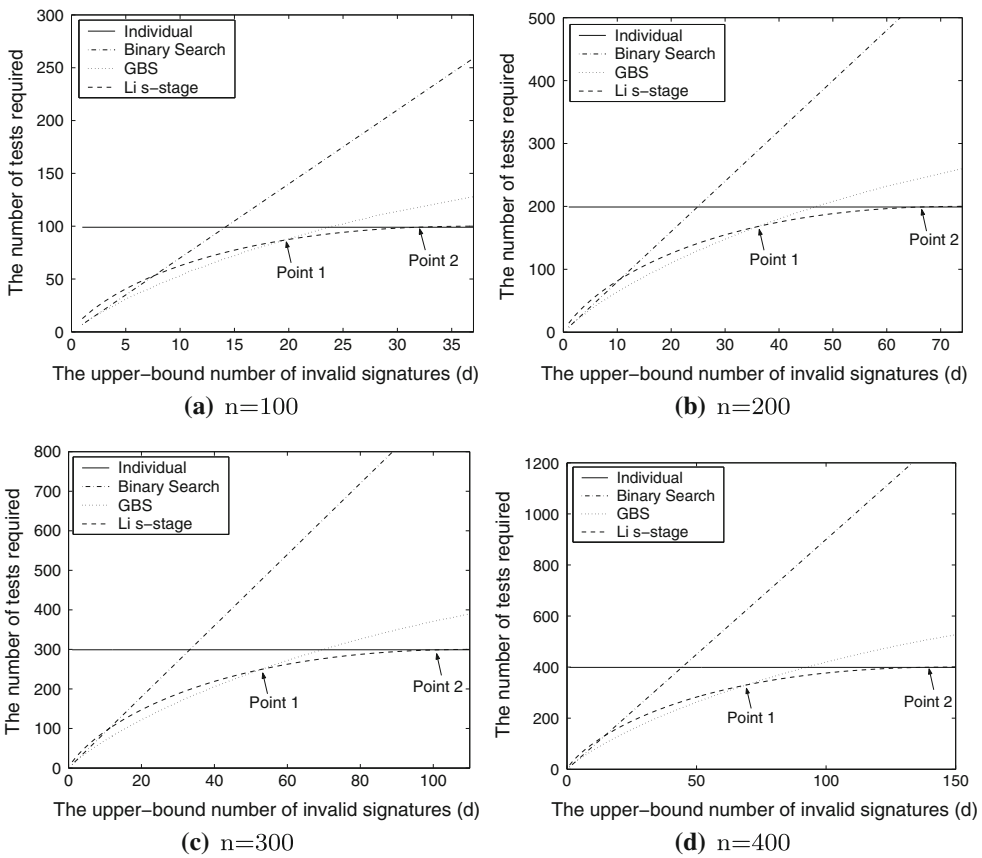
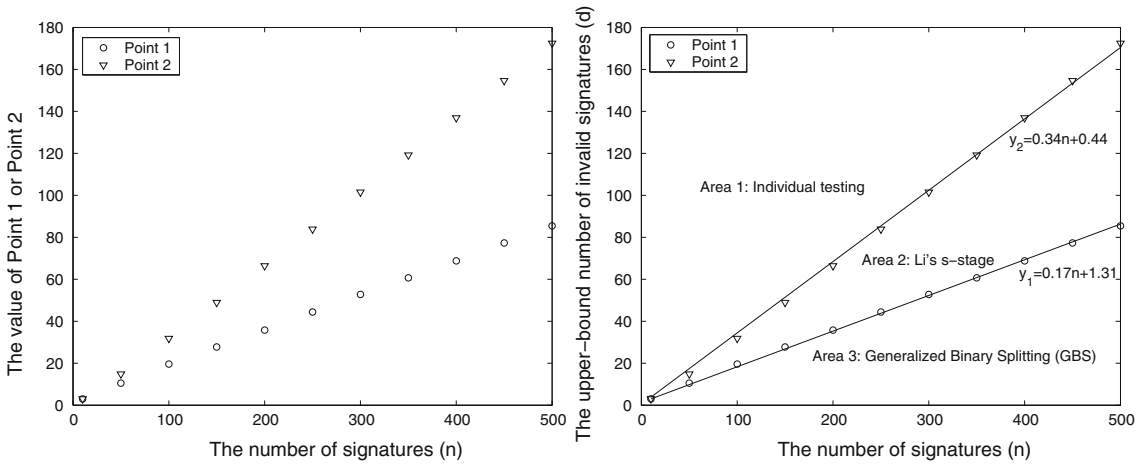


Fig. 3 Show the number of required tests (batch verifications) as d changes and n is fixed. In sub-figure **a**–**d**, n, respectively, is equal to 100, 200, 300, and 400



(a) Show the values of Point 1 (Point 2) given different values of n

(b) Show the functions to fit the samples of Point 1 and Point 2

Fig. 4 Show two sets of values of Point 1, Point 2 and the corresponding fitting functions

4.2 Generalized binary splitting

For batch verification, an attacker needs to send only a few number (at least one) of invalid signatures to launch a DoS attack that makes normal vehicles do group testing to find invalid signatures. In this case, the value of d that has been analyzed in the previous section is small. From the analysis in the previous section, we know that when d is small, GBS is the optimal group testing algorithm.

The GBS algorithm [36] is adopted in our scheme and presented in Algorithm 1. To use Algorithm 1, we need to estimate d, the upper bound on the number of invalid signatures. In VANETs, vehicles send the traffic related messages every 300 ms. In a normal case, the number of signatures that vehicles receive every 300 ms is equal to the number of their neighbors, which can assist in estimating d. Such an estimating work belongs to a category of intrusion detection, and many related work [37, 38] has been conducted. For example, based on a number of ds in previous time periods, a Markov chain [39] can be used to compute the distribution of d and estimate d in the upcoming time period. We adopt the existing solution, and estimating d is beyond the scope of this paper.

Algorithm 1 Generalized binary splitting algorithm

Input: n signatures, where the estimated number of invalid signatures is not more than d

Result: Find out all invalid signatures

If $n \leq 2d - 2$ **then**

  a vehicle tests the n items individually;

  the group testing is done and return.

else

  compute $l = n - d + 1$, and $\alpha = \lfloor \log(l/d) \rfloor$.

end

Test a group of size 2^{α} signatures.

If the *outcome is negative* **then**

  the group of 2^{α} signatures are identified as good.

  set $n = n - 2^{\alpha}$, and go to Step 1.

else

  use binary search to identify 1 invalid signature, and an unspecified number, say

  m, of valid signatures.

  set $n = n - 1 - m$, $d = d - 1$, and go to Step 1.

end

5 Performance evaluation

In this section, we evaluate the performance of the IBV scheme in terms of verification delay and transmission overhead. The IBV scheme can be used in both V2V and V2R communications.

5.1 Verification delay

We define and compute the time cost of the cryptographic operations required in each verification by the proposed IBV scheme. Let T_{mul} denote the time to perform one point multiplication over an elliptic curve, T_{mtp} the time of a MapToPoint hash operation, and T_{par} the time of a pairing operation. Since these operations dominate the speed of a signature verification, we only consider these operations

Table 3 Comparisons of the speed of three signature schemes (ms)

	Verify a single signature	Verify n signatures
IBV	$3T_{\text{par}} + T_{\text{mtp}} + T_{\text{mul}}$	$3T_{\text{par}} + nT_{\text{mtp}} + nT_{\text{mul}}$
BLS	$4T_{\text{par}} + 2T_{\text{mtp}}$	$(2n+2)T_{\text{par}} + 2nT_{\text{mtp}}$
ECDSA	$4T_{\text{mul}}$	$4nT_{\text{mul}}$

and neglect all the other operations such as additive and one-way hash function. We adopt the experiment in [43], which observes the processing time for an MNT curve [41] of embedding degree $k = 6$ and 160-bit q, running on an Intel Pentium IV 3.0 GHZ machine. The following results are obtained: T_{mul} is 0.6 ms, T_{par} is 4.5 ms, and T_{mtp} is 0.6 ms .

Next, we compare the proposed IBV scheme with ECDSA and BLS [23, 42] in terms of the verification delay. Here, the ECDSA scheme is the signature algorithm adopted by IEEE1609.2 standard [35], while BLS is a short signature scheme, which can also be used to perform signature aggregation. Table 3 shows the combination of the dominant operations of the three signature schemes in terms of verifying a single signature and n signatures, respectively. From the batch verification equation in Sects. 3–5, we observe that the time to verify n distinct signatures is $3T_{\text{par}} + nT_{\text{mtp}} + nT_{\text{mul}}$. According to [42], with BLS, the time spent on verifying n signatures is equal to $(n+1)T_{\text{par}} + nT_{\text{mtp}}$; while with the ECDSA, verifying distinct n signatures requires $2nT_{\text{mul}}$. Since ECDSA and BLS are not identity-based signature schemes, additional operations are needed to verify the public key's certificate. Thus, the overall message verification time for ECDSA and BLS should be doubled[2] as shown in Table 3.

In our analysis, the communication coverage of a vehicle is 300 m, and each vehicle periodically broadcasts a traffic related message every 300 ms. The traffic density is taken as the number (#) of vehicles within a vehicle's communication range. The traffic density is also take as the number of signatures to be verified in 300 ms. We compare the performance by using IBV, ECDSA, and BLS to verify the signatures.

Figure 5 shows the relationship between the verification delay and the number of vehicles within a vehicle's communication range. The embedded small figure is a local zoom-in with the traffic load ranging from 8 to 14. From Fig. 5, we can observe that the verification delay by using BLS is always the largest no matter how many messages

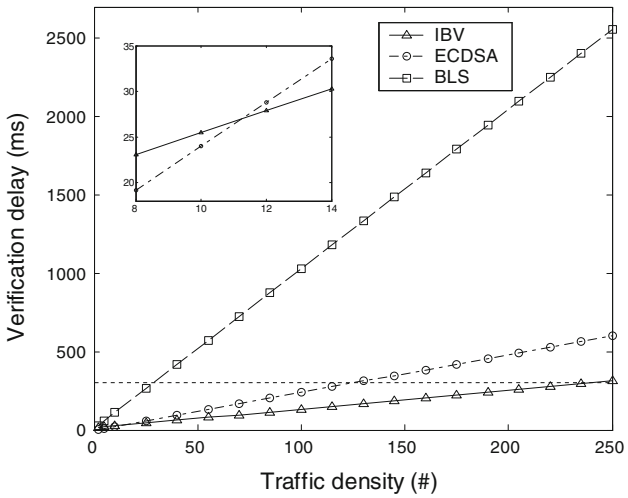


Fig. 5 Verification delay versus Traffic density

are received by a vehicle. Another interesting result is that when the number of messages received within 300 ms is smaller than 11, the ECDSA scheme achieves the smallest message verification latency; however, when the number of messages is greater than 11, the IBV scheme yields much less verification latency. Figure 5 also shows that within a 300 ms interval, the maximum number of signatures that can be verified by a vehicle is equal to 29, 125, and 239 when the BLS, ECDSA, and IBV schemes are adopted, respectively. In other words, when the number of incoming messages is greater than these maximal thresholds, some messages will be lost accordingly. Obviously, the IBV scheme can verify the largest number of signatures, which is observed to achieve the lowest message loss ratio when the traffic load increases.

We compare the message verification delay of these three schemes in terms of the ratio of the verification delays as shown in Fig. 6. We can see that the delay ratio between IBV and ECDSA approaches to a constant, which is approximately 0.641 when the number of messages in one interval is greater than 40. The delay ratio between IBV and BLS is approximately 0.157 when the number of messages is larger than 30. In other words, the speed of IBV is 35.6% faster than that of ECDSA, and is 84.3% faster than that of BLS.

5.2 Expected verification delay with false signatures

In the previous section, the verification delay is evaluated in the situation where no false signature exists in each batch. In this section we will further analyze the verification delay where false signatures exist in a batch.

If the batch verification of IBV fails, the GBS group testing approach is used to find invalid signatures. In this case, more verification delay would occur. To properly

[2] With the IBV scheme, each message sent by a vehicle corresponds to a distinct identity. Thus, to achieve the same privacy level as the IBV's, the vehicle using the public key based schemes also needs to change an identity for each sending message. That is the reason why verification time for ECDSA and BLS should be doubled in this paper.

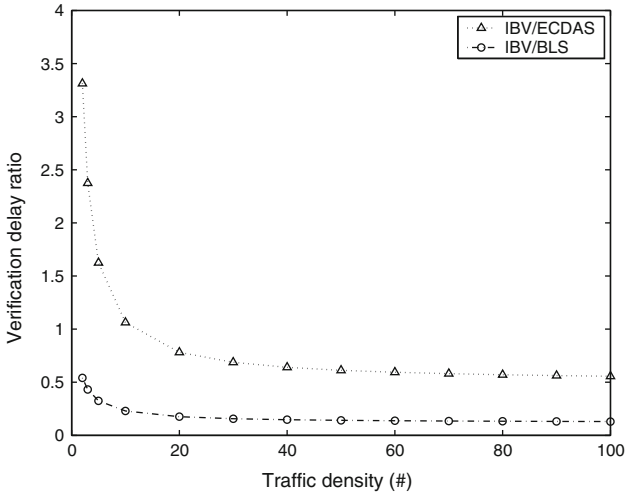


Fig. 6 Verification delay ratio versus Traffic density

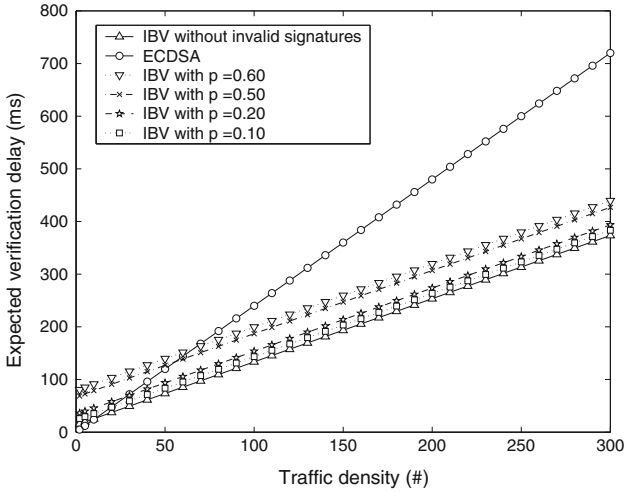


Fig. 7 Expected verification delay versus Traffic density

quantify this delay, we define two probabilities. Let q denote the probability that a signature is invalid, and p denote the probability that a batch of n signatures has at least one invalid signature. Clearly, p also denotes the probability that a DoS attack happens, while $1 - p$ denotes the probability that no DoS attack happens. The relationship between p and q is presented below.

$$p = 1 - (1 - q)^n, \text{ and } q = 1 - (1 - p)^{1/n}. \tag{3}$$

Let T_{IBV} denote the verification delay that IBV is used to perform batch verification on n signatures. As presented in Table 3, $T_{IBV} = 3T_{\text{par}} + nT_{\text{mtp}} + nT_{\text{mul}}$. Let T_{GBS} denote the delay that the GBS group testing approach is used to find invalid signatures. As shown in Sect. 4.1, $T_{GBS} = (d - 1 + \lceil \log(nd) \rceil) \cdot 3T_{par}$, where $d = n \cdot q$. Using the above parameters, we can derive the expected verification delay of IBV. We use E_{IBV} to denote the total expected verification delay, which is derived as below.

$$\begin{aligned} E_{IBV} &= T_{IBV} \cdot (1 - p) + (T_{IBV} + T_{GBS}) \cdot p \\ &= T_{IBV} + T_{GBS} \cdot p \\ &= T_{IBV} + (d - 1 + \lceil \log(nd) \rceil) \cdot 3T_{par} \cdot p \\ &= 3T_{\text{par}} + nT_{\text{mtp}} + nT_{\text{mul}} + \{(1 - (1 - p)^{1/n}) \cdot n \\ &\quad - 1 + \lceil \log(n\lceil (1 - (1 - p)^{1/n}) \cdot n \rceil) \rceil\} \cdot 3T_{par} \cdot p \end{aligned} \tag{4}$$

Figure 7 shows the relationship between E_{IBV} with different values of p and the number of vehicles (signatures) in a vehicle's communication range. From Fig. 7, we can observe that given a fixed number of vehicles (signatures) E_{IBV} increases as p increases, but the increasing amount is not significant. If p is a small value, for example $p = 10\%$, E_{IBV} is close to the verification delay when there is no DoS attack. Compared with ECDSA, IBV with a large p still yields a lower expected verification delay especially in the scenario where the traffic density is high. As shown in Fig. 7, if the probability (p) that DoS happens equals to 50%, the IBV has lower verification delay than ECDSA has when the number of vehicles (signatures) is greater than 55. It is worth noticing that in reality the probability that DoS happens is far less than 50%. Therefore, we conclude that IBV is able to achieve low verification delay even though the DoS attack is taken into consideration.

5.3 Transmission overhead

In this section, we compare the transmission overhead of IBV, ECDSA and BLS. The comparison is in terms of the following two aspects: the transmission overhead in V2V communication and the overhead in non-safety application. Here, the transmission overhead includes a signature and a certificate appended to the original message, while the message itself is not counted.

For IBV and BLS, the length of a signature is 21 bytes, while the length for ECDSA is 42 bytes. When we use BLS or ECDSA, a certificate must be transmitted along with a signature. If we use the certificate presented in IEEE 1,609.2 Standard [35], which has 125 bytes in length, the total transmission overhead of the BLS and ECDSA scheme is 21 + 125 bytes and 42 + 125 bytes, respectively, as shown in Table 4. Since the proposed IBV scheme is based on identity-based cryptography, only a short pseudo identity with 42 bytes is transmitted along with the original message. Thus, the total transmission overhead of IBV is 21 + 42 bytes as shown in Table 4.

Figure 8 shows the relationship between the transmission overhead and the number of messages received by a vehicle in 1 min. Obviously, as the number of messages increases, the transmission overhead increases linearly.

Table 4 Comparisons of transmission overhead of three schemes (ms)

	Send a single message	Send n messages
IBV	21 + 42 bytes	21 + 42n bytes
BLS	21 + 125 bytes	21 + 125n bytes
ECDSA	42 + 125 bytes	42n + 125n bytes

The transmission overheads of ECDSA is the largest among the three schemes, and the transmission overhead of the IBV is much smaller than the other two. We can further observe that the transmission overhead of the IBV scheme is 43.2 percent of that of BLS and 37.7 percent of that of ECDSA. On the other hand, as shown in Fig. 8 , within the observation window of 1 min, when the number of messages increases up to 30,000, IBV saves 2.37 Mbytes and 2.98 Mbytes of bandwidth compared with BLS and ECDSA, respectively. Here, 30,000 corresponds to the number of messages sent by 150 vehicles in 1 min.

On the other hand, IBV can also be used for signature aggregation in non-safety related applications to reduce the communication overhead. For example, in the delay and forwarding application, a vehicle assisting in forwarding messages could aggregate multiple signatures. With IBV, given n distinct signatures, $\sigma_1, \sigma_2, \ldots, \sigma_n$, the aggregate signature is equal to $\sum_{i=1}^{n} \sigma_i$. Further more, it is worth noticing that unlike BLS, IBV does not require that n distinct messages have to be sent from the same sender. We compare the transmission overhead due to signature aggregation with the overhead without signature aggregation.

As shown in Table 4, let an RSU send n distinct signatures to a vehicle for forwarding. With the ECDSA scheme, the transmission overhead is in proportion to the number of signatures, namely $(42 + 125)n$ bytes. In contrast, since BLS and IBV can aggregate signatures, only one aggregate signature is sent. In addition to the signatures, the BLS scheme needs to transmit a certificate with the length of 125 bytes for each message, while the IBV only needs to transmit a pseudo identity with the length of 42 bytes for each message. Thus, the total transmission overhead is $21 + 125n$ and $21 + 42n$ for the BLS and IBV, respectively. Fig. 9 shows the comparisons. The transmission overhead of all the schemes is proportional to the number of aggregate signatures. Compared with ECDSA, BLS is subject to lower transmission overhead; nonetheless, the advantage gained in BLS is not obvious because the certificate dominates the length of the overhead. On the other hand, since no certificate for each message is required in IBV, the advantage gained in the proposed scheme is obvious. From Fig. 9, we can see the transmission overhead of the IBV scheme is 33.6 percent of that by BLS and only 25.1 percent of that by ECDSA.

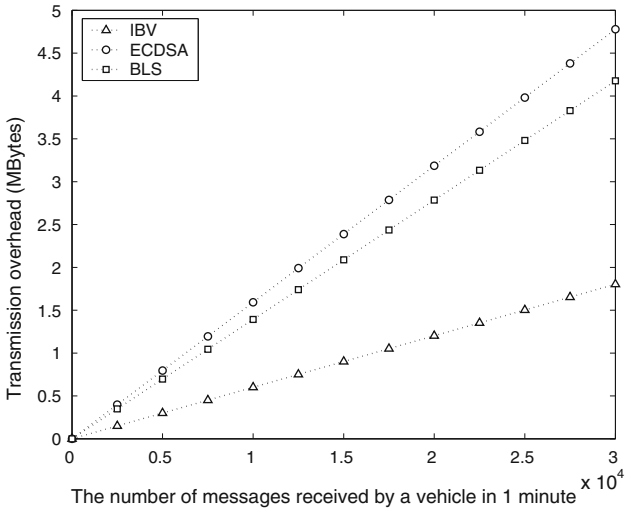


Fig. 8 Transmission overhead versus the number of messages received by an RSU in 1 min (between vehicles and an RSU)

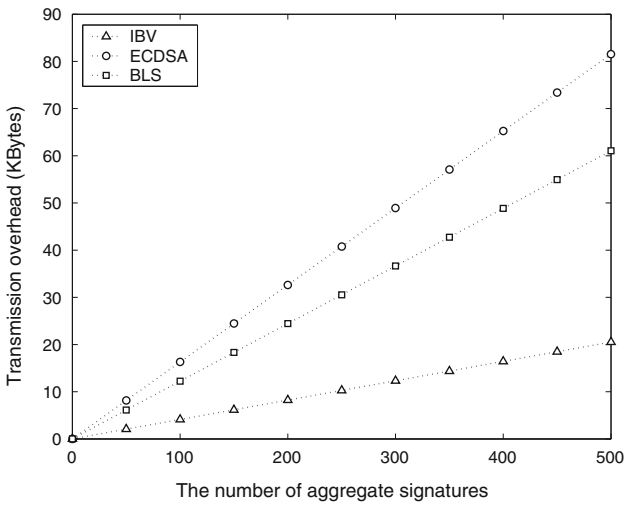


Fig. 9 Transmission overhead versus the number of aggregate signatures

6 Related work

VANETs have been widely used in safety applications [1] and non-safety related applications [45, 47]. The security and privacy issues on VANETs have attracted extensive attentions from both academia and industry. Hubaux et al. [6, 7] first identified the issues of security and privacy preservation in VANETs by claiming that an appropriate Public Key Infrastructure (PKI) must be well devised to protect the transited information and to mutually authenticate among network entities. To address the privacy issue, they suggested to relying on temporary pseudonyms to achieve anonymity.

Raya et al. [8] proposed an anonymous-key-based (HAB) security protocol, which can achieve anonymous message authentication and conditional privacy preservation. With the HAB solution, a huge set of anonymous keys are preloaded in each vehicle, and each vehicle randomly takes one of the keys in the set to sign a safety message. To further prevent movement tracking, each anonymous key has a short lifetime. The HAB scheme presented an efficient and straightforward way in solving the privacy issues, while the central authority simply keeps all the anonymous certificates of all the vehicles in a certain area in order to maintain the traceability. Once a malicious message is detected, the authority has to exhaustedly search in a very huge database (probably 43,800 times millions of cars) to find the real identity related with the compromised anonymous public key which incurs tremendous complexity for the identity and certificate management. Lin et al. [11] proposed an efficient security protocol called GSIS, which is based on the group signature scheme [44]. With this protocol, only a private key and the group public key are stored in the vehicle, and the messages are signed according to the group signature scheme without revealing any identity information to the public. This assures that the trusted authority is equipped with the capability of exposing the sender identity of a message. However, the verification of each group signature requires at least two pairing operations which might not be scalable when the density of the traffic is increasing.

Raya et al. [10] proposed a secure traffic aggregation scheme to minimize the communication overhead and initiate a tradeoff between the security and efficiency. Under their design, firstly, cells are defined and predetermined according to the physical location. When vehicles are located in a cell, the vehicle that is physically closest to the center of the cell is automatically taken as the group leader of the vehicles in the cell, which is delegated to aggregate messages for the whole group when the message is going to be relayed to the leader of the neighbor groups. The aggregation of messages can achieve a significant reduction in the overhead for vehicle to vehicle communications. However, the vehicle closest to the center of a cell could change frequently, leading to a frequent update of the group leader of a cell (e.g., once in a few seconds), which indicates that the approach can be further improved in terms of its efficiency and practical applicability.

Zhang et al. [14, 46] proposed an RSU-aided authentication scheme, called RAISE. With RAISE, RSUs are responsible for verifying the authenticity of messages sent by vehicles and notifying the authentication results back to all the associated vehicles. RAISE not only achieves message integrity and source authentication, but also has lower computation and communication overhead. In addition, RAISE achieves the conditional privacy preservation. However, RAISE highly depends on RSUs, and thus its use is limited when RSU is absent in some situations, for example, at the beginning of a VANETs' deployment period, or due to the physical damage of RSUs.

Unlike all the previous works, the proposed IBV scheme can meet all the security and efficiency requirements for V2V and V2R communications, such as the verification speed, transmission overhead, management efficiency, anonymity, and traceability, which have been verified and analyzed in details through the paper.

7 Conclusions and future work

We have proposed a novel Identity-based Batch Verification (IBV) scheme for V2V and V2I communications in VANETs, which has been identified to be capable of meeting the most important and emerging design requirements on security and privacy preservation ever reported in the literatures. The proposed IBV scheme can significantly improve the system performance by fully taking advantages of verifying multiple message signatures once instead of in a one-by-one manner. Our scheme has also addressed the identity privacy and traceability issues in vehicular networks, where the signature of a message is signed according to a pseudo identity pair and private keys that are generated by the tamper-proof device. The IBV scheme enables the Trusted Authority (TA) to retrieve the real identity of a vehicle from any message signature, such that conditional privacy preservation can be achieved. We adopt group testing technique to efficiently find invalid signatures in a batch of signatures. In addition, extensive analysis and evaluation have been conducted to demonstrate that the IBV scheme can achieve excellent operational efficiency for vehicular communications in terms of signature verification delay and communication overhead, in comparison with existing counterparts.

In our future work, we will put our efforts on addressing more Denial of Service (DoS) attacks in VANETs, such as a dummy message jamming (DMJ) attack, i.e., attackers send a large number of invalid messages. The DMJ attack could delay the verification on legitimate messages. The DMJ attack is not only fatal to safety related applications, but also is hard to defend. Therefore, thwarting the DMJ attack is a challenging and urgent work in our future research.

Acknowledgments The authors thank Prof. Xuemin (Sherman) Shen, Prof. Xiaodong Lin, and Rongxing Lu for providing valuable suggestions and discussions on this paper. The research is financially supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Zhang, C., Lu, R., Lin, X., Ho, P.-H., & Shen, X. (2008). An efficient identity-based batch verification scheme for vehicular sensor networks. *The 27th IEEE international conference on computer communications (INFOCOM 2008)*, pp. 816–824.
2. Misener, J. A. (2005). Vehicle-infrastructure integration (VII) and satety: rubber and radio meets the road in california. *Intellimotion, 11*(2), 1–3.
3. Lee, U., Magistretti, E., Zhou, B., Gerla, M., Bellavista, P., & Corradi, A. (2006). Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wireless Communications, 13*(5), 52–57.
4. Wang, F., Zeng, D., & Yang, L. (2006). Smart cars on smart roads: an IEEE intelligent transportation systems society update. *IEEE Pervasive Computing, 5*(4), 68–69.
5. Dedicated Short Range Communications (DSRC), [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html.
6. Hubaux, J. P., Capkun, S., & Luo, J. (2004). The security and privacy of smart vehicles. *IEEE Security and Privacy Magazine, 2*(3), 49–55.
7. Raya, M., & Hubaux, J. P. (2005). "Security aspects of inter-vehicle communications," in *Proceedings of Swiss Transport Research Conference*.
8. Raya, M., & Hubaux, J. P. (2007). Securing vehicular ad hoc networks. *Journal of Computer Security, 15*(1), 39–68.
9. Raya, M., Papadimitratos, P., & Hubaux, J. P. (2006). Securing vehicular communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, 13*(5), 8–15.
10. Raya, M., Aziz, A., & Hubaux, J. P. (2006). Efficient secure aggregation in VANETs. In *Proceedings of International workshop on Vehicular ad hoc networks*, pp. 67–75.
11. Lin, X., Sun, X., Ho, P. H., & Shen, X. (2007). GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transaction on Vehicular Technology, 56*(6), 3442–3456.
12. Xu, Q., Mak, T., Ko, J., & Sengupta, R. (2004, October). Vehicle-to-vehicle safety messaging in DSRC. In *Proceedings of international workshop on vehicular ad hoc networks*, pp. 19–28.
13. Yang, X., Liu, J., Zhao, F., & Vaidya, N. (Aug. 2004). A vehicle-to-vehicle communication protocol for cooperative collision warning. In *Proceedings of IEEE MobiQuitous*, pp. 114–123.
14. Zhang, C., Lin, X., Lu, R., & Ho, P.-H. (2008). RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks. In *Proceedings of IEEE international conference on communications*, Beijing, China.
15. Sha, K., Xi, Y., Shi, W., Schwiebert, L., & Zhang, T. (2006). Adaptive privacy-preserving authentication in vehicular networks. In *Proceedings of IEEE international workshop on vehicle communication and applications*, pp. 1–8.
16. Lin, X., Lu, R., Zhang, C., Zhu, H., Ho, P.-H., & Shen, X. (2008). Security in vehicular ad hoc networks. *IEEE Communications Magazine, 46*(4), 88–95.
17. U.S. Department of Transportation, National Highway Traffic Safety Adimistrtation, Vehicle Safety Communications Project, Final Repot, (Apr. 2006).
18. Road Weather Management. [Online]. Available: http://www.itsove-rview.its.dot.gov/RWM.asp.
19. Ren, K., Lou, W., Deng, R. H., & Kim, K. (2006). A novel privacy preserving authentication and access control scheme in pervasive computing environments. *IEEE Transactions on Vehicular Technology, 55*(4), 1373–1384.
20. Sampigethava, K., Huang, L., Li, M., Poovendran, R., Matsuura, K., & Sezaki, K. (2006). CARAVAN: Providing location privacy for VANET. In *Proceedings of international workshop on Vehicular ad hoc networks*.
21. Boneh, D., & Franklin, M. (2001). Identity-based encryption from the Weil pairing. In *Proceedings of Crypto*, LNCS, Vol. 2139, pp. 213–229.
22. Miyaji, A., Nakabayashi, M., & Takano, S. (2001). New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals, E84-A*(5), 1234–1243.
23. Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the weil pairing. In *Proceedings of Asiacrypt, 2248*, 514–532.
24. Fiat, A. (1989). Batch RSA. In *Proceedings of Crypto*, pp. 175–185.
25. Naccache, D., M'Raïhi, D., Vaudenay, S., & Raphaeli, D. (1994). Can D.S.A be improved? Complexity trade-offs with the digital signature standard. In *Proceedings of EUROCRYPT*, LNCS, 950, pp. 77–85.
26. Cha, J. C., & Cheon, J. H. (2003). An identity-based signature from gap Diffie-Hellman groups. In *Proceedings of public key cryptography*, pp 18–30.
27. Zhang, F., Safavi-Naini, R., & Susilo, W. (2003). Efficient verifiably encrypted encrypted signature and partially blind signature from bilinear pairings. In *Proceedings of indocrypt*, LNCS, 2904, pp. 191–204.
28. Zhang, F., & Kim, K. (2003). Efficient ID-based blind signature and proxy signature from bilinear pairings. In *Proceedings of ACISP*, LNCS, 2727, pp 312–323.
29. Yoon, H., Cheon, J. H., & Kim, Y. (2004). Batch verification with ID-based signatures. In *Proceedings of information security and cryptology*, pp 233–248.
30. Camenisch, J., Hohenberger, S., & Pedersen, M. Ø. (2007). Batch verification of short signatures. In *Proceedings of EUROCRYPT*, LNCS, 4514, pp 246–263.
31. Camenisch, J., & Lysyanskaya, A. (2004). Signature schemes and anonymous credentials from bilinear maps. In *Proceedings of Crypto*, LNCS, 3152, pp 56–72.
32. Eastlake, D. & Jones, P. (2001). US secure hash algorithm 1 (SHA1). IETF RFC 3174.
33. ElGamal, T. (1985). A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory, 31*(4), 469–472.
34. Miller, V. S. (1985, August). Use of elliptic curves in crytography. In *Proceedings of advance in cryptology*, pp. 417–426.
35. IEEE Standard 1609.2. (2006). IEEE trial-use standard for wireless access in vehicular environments—security services for applications and management messages.
36. Du, D., & Hwang, F. K. (2000). Combinatorial group testing and its applications (2nd Edn.). World Scientific, Singapore.
37. Ye, N. (2004). Robustness of the Markov-chain model for cyber-attack detection. *IEEE Transactions on Reliability, 53*(1), 116–123.
38. Ye, N. (2000). A Markov chain model of temporal behavior for anomaly detection. In *Proceedings of the IEEE systems, man, and cybernetics information assurance workshops*.
39. Markov, A. A. (1971). Extension of the limit theorems of probability theory to a sum of variables connected in a chain. Reprinted in Appendix B of: R. Howard. Dynamic probabilistic systems, Vol. 1, Markov Chains. New York: Wiley.
40. Zaverucha, G. M., & Stinson, D. R. (2009). Group testing and batch verification. The 4th international conference on information theoretic security, ICITS 2009.
41. Miyaji, A., Nakabayashi, M., & Takano, S. (2001). New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals, E84-A*(5), 1234–1243.

42. Boneh, D., Gentry, C., Lynn, B., & Shacham, H. (2003). Aggregate and verifiably encrypted signatures from bilinear maps," In *Procedings of Eurocrypt*, LNCS, 2656, pp 416–432.
43. Scott, M. Efficient implementation of crytographic pairings, [Online]. Available: http://ecrypt-ss07.rhul.ac.uk/Slides/Thursday/mscott-samos07.pdf.
44. Boneh, D., Boyen, X., & Shacham, H. (2004). Short group signatures. In *Advances in cryptology*, LNCS, 3152, pp 41–55.
45. Lee, S.-B., Pan, G., Park, J.-S., Gerla, M., Lu, S. (2007). Secure incentives for commercial ad dissemination in vehicular networks. In *Proceedings of MobiHoc*.
46. Zhang, C., Lin, X., Lu, R., Ho, P.-H., & Shen, X. (2008). An efficient message authentication scheme for vehicular communications. *IEEE Transactions on Vehicular Technology, 57*(6), 3357–3368.
47. He, S., Chen, J., Sun, Y., Yau, D. K. Y., & Yip, N. K. (2010). On optimal information capture by energy-constrained mobile sensors. *IEEE Transactions on Vehicular Technology, 59*, 2472–2484.

Author Biographies

Chenxi Zhang received his B.E. and M.E. degree from the School of Computer Science and Technology in Harbin Institute of Technology, China, in 2003 and 2005, respectively. He received his Ph.D. degree from the Electrical and Computer Engineering department in the University of Waterloo, Canada, in 2010. His research interests include wireless network security and vehicular network security.

Pin-Han Ho received his B.Sc. and M.Sc. degree from the Electrical Engineering department in National Taiwan University in 1993 and 1995, respectively, and Ph.D. degree from Queens University at Kingston at 2002. He is now an associate professor in the department of Electrical and Computer Engineering, University of Waterloo, Canada. Professor Pin-Han Ho is the author/co-author of more than 150 refereed technical papers, several book chapters, and the co-author of a book on optical networking and survivability. His current research interests cover a wide range of topics in broadband wired and wireless communication networks, including survivable network design, wireless Metropolitan Area Networks such as IEEE 802.16 networks, Fiber-Wireless (FIWI) network integration, and network security. He is the recipient of Distinguished Research Excellent Award in the ECE department of U of Waterloo, Early Researcher Award (Premier Research Excellence Award) in 2005, the Best Paper Award in SPECTS'02, ICC'05 Optical Networking Symposium, and ICC'07 Security and Wireless Communications symposium, and the Outstanding Paper Award in HPSR'02.

Janos Tapolcai received his M.Sc. ('00 in Technical Informatics), and Ph.D. ('05 in Computer Science) degrees in Technical Informatics from Budapest University of Technology and Economics (BME), Budapest, Hungary. Currently he is an Associate Professor at the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at BME. His research interests include applied mathematics, combinatorial optimization, linear programming, linear algebra, routing in circuit switched survivable networks, availability analysis, grid networks, and distributed computing. He has been involved in several related European and Canadian projects. He is an author of over 40 scientific publications, and is the recipient of the Best Paper Award in ICC'06.