Efficient HMAC-based Secure Communication for VANETs

Changhui Hu, Tat Wing Chim, S.M. Yiu, Lucas C.K.Hui, and Victor O.K. Li

Abstract—**Vehicular ad hoc network (VANET) is an emerging type of network which facilitates vehicles on roads to communicate for driving safety. It requires a mechanism to help authenticate messages, identify valid vehicles, and remove malevolent vehicles which do not obey the rules. Most existing solutions either do not have an effective message verification scheme , or use the Public Key Infrastructure (PKI). In this network, vehicles are able to broadcast messages to other vehicles and a group of known vehicles can also communicate securely among themselves. So group communication is necessary for the network. However, most existing solutions either do not consider this or use pairing operation to realize this. They are either not secure or not effective. In this paper, we provide a more comprehensive set of secure schemes with Hash-based Message Authentication Code(HMAC) in VANETs to overcome their shortcomings. Of course, we still need to use Pairing operation in some place. Our scheme is composed of three schemes: (1)Communications between Vehicles and Road-Side Units(RSU) (2)One to One Communications within a Group (3)One to One Communications without a Group. Based on our simulation study, we show that our schemes are effective and the delay caused is much lower. The average delay caused by our first scheme is nearly thousands of times lower than prior schemes. The average delay caused by our second scheme is 0.312ms, while the delay caused by prior scheme is 12.3ms. Meanwhile the average delay caused by our third scheme is 0.312ms, and the delay caused by prior scheme is about 9s.**

Index Terms—**Secure vehicular sensor network, authentication, group communication, HMAC, symmetric cryptography**

I. Introduction

Vehicular Ad Hoc Network (VANET) has been attracting more and more attentions from both industry and academia. It is a critical component of the Intelligent Transportation Systems (ITSs) [6] which aims at enhancing driving safety through inter-vehicle communications or communications with roadside infrastructure. In a typical VANET, each vehicle is assumed to have an on-board unit (OBU) and there are road-side units (RSU) installed along the roads. A trusted authority (TA) and maybe some other application servers are installed in the backbone. The OBUs and RSUs communicate using the Dedicated Short Range Communications (DSRC) protocol [7] over the wireless channel while the RSUs, TA, and the application servers communicate using a secure fixed network. Based on this, each vehicle can periodically broadcast safety information [8] containing its current speed, location, road condition and traffic accident information, etc. every 100-300 ms. With the received information, other drivers can make an early response in case of exceptional situations. With multi-hop forwarding, the messages will be either terminated by an RSU or dropped when exceeding their lifetimes. The RSU may also inform the traffic control center to adjust traffic lights for avoiding possible traffic congestion. VANET also provides a platform for a group of known vehicles (e.g. police chasing a bank robber) to establish a secure communication channel (group communication).

Like other communication networks, security issues have to be well-addressed. For example, message integrity must be guaranteed, and the message senders should be authenticated. Otherwise, an attacker can replace the safety message from a vehicle or even impersonate a vehicle to transmit a fake safety message which in turn can cause accidents or even loss of life. In addition, user privacy concerns must also be well mitigated, where the identity, the position, and the trajectory of a specific vehicle should not be obtained by a third party. Otherwise, one may not be willing to use this new type of network. Thus an anonymous and secure communication protocol is vital to VANET. Being motivated by these, we propose an efficient HMAC-based secure communication protocol in this paper. Although all communications are broadcast in nature, we make unicast communications possible in our one-to-one communications schemes by adopting cryptographic techniques such as asymmetric encryption.

To ensure both identity authentication and message integrity in VANETs, encryption and digital signature in conventional public key infrastructure (PKI) [9] is a well accepted choice. One appealing solution is to sign and encrypt each message before the message is sent. However, conventional signature and encryption schemes that decrypt and verify the received messages one after the other may fail to satisfy the stringent time requirement of the vehicular communication applications. And the computation power of an OBU is also not strong enough to handle all verifications in a short time, especially in places where the traffic density is high. Even they are done by RSUs, note that an RSU could communicate with hundreds of OBUs. In this case, dealing with a large number of messages sequentially could take a long time and will certainly become the processing bottleneck at the RSUs. An efficient method for dealing with plenty of messages within a short period of time is desirable.

Hash-based Message Authentication Code (HMAC) [24],

Changhui Hu is with the Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Email: huchanghui@mail.sdu.edu.cn. This work was done while he was visiting Department of Computer Science, The University of Hong Kong.

Tat Wing Chim is with the Department of Computer Science, The University of Hong Kong, Email: twchim@cs.hku.hk

S.M. Yiu is with the Department of Computer Science, The University of Hong Kong, Email: smyiu@cs.hku.hk

Lucas C.K.Hui is with the Department of Computer Science, The University of Hong Kong, Email: hui@cs.hku.hk

Victor O.K. Li is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Email: vli@eee.hku.hk

is a mechanism for message authentication using cryptographic hash functions. It can be used with any iterative cryptographic hash function such as MD5 and SHA-1, in combination with a secret shared key. As one kind of Message Authentication Code (MAC), it provides a way for checking the integrity of information transmitted over or stored in an unreliable medium based on a secret key. Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. The use of elliptic curves in cryptography was suggested independently by Neal Koblitz [25] and Victor S. Miller [26] in 1985. In public key cryptography each user has a pair of keys: a public key and a private key. Only the user knows the private key whereas the public key is distributed to all other users. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithms which is much more difficult to be cracked at equivalent key lengths. Prior schemes mainly use the pairing operation called bilinear map based on ECC for verification. Compared to pairing operation, HMAC is much more effective. We have tested the pairing operation based on an implementation using PBC library [23], its running time is thousands of times longer than HMAC based on sha-1. We will talk about the details in Section VI.A.

Related work: Many related studies have been reported to address security and privacy preservation problems in VANETs [10 - 17]. To achieve both message authentication and anonymity, Raya et al. in [10] proposed that each vehicle should be pre-loaded with a large number of anonymous public and private key pairs and the corresponding public key certificates. Freudiger et al. in [11] addressed the problem of achieving location privacy in VANETs with randomly changing identifiers. In particular, they proposed a protocol to create cryptographic mix-zones at road intersections. They analyzed theoretically and by simulation the location privacy achieved by combining mix-zones into mix-networks. In [12], Wen et al. propose to use the physical property of a transmitting signal to discriminate one transmitter from others because physical measurement is more efficient than software computation. Wasef and Shen [13], on the other hand, aims at enhancing the efficiency of any certificate-based authentication scheme. They propose a HMAC-based solution to replace the time-consuming and traditional certificate revocation list checking process. Sun et al. in [14] introduced a group signature scheme to sign each message. Some recent works [15 - 17] also propose to achieve the goal by using group signature schemes. That is, each vehicle in the system is assigned a group private key. When a vehicle wants to broadcast a message, it signs the message using its group private key. Verifiers such as RSUs can then verify its signature using a common group public key. In this way, a signature can be properly verified but at the same time, the real identity of the signer can be hidden. Only if necessary, a trusted party can use a private key to reveal the real identity of the signer.

In [3], the IBV protocol was proposed for vehicle-to-RSU communications. The RSU can verify a large number of signatures as a batch using just three pairing operations. However, their work has some limitations. First, their protocol relies heavily on a tamper-proof hardware device, installed in each vehicle, which preloads the system-wide secret key. Once one of these devices is cracked, the whole system will be compromised. Second, a vehicle's real identity can be traced by anyone, thus the protocol does not satisfy the privacy requirement. Third, their protocol has a flaw such that a vehicle can use a fake identity to avoid being traced (anti-traceability attack) or even impersonate another vehicle (impersonation attack1). Forth, in their batch verification scheme, if any of the signatures is erroneous, the whole batch will be dropped. This is inefficient because most signatures in the batch may actually be valid, thus may imply a not satisfactory successful rate. Finally, the IBV protocol is not designed for vehicle-to-vehicle communications.

In a more recent work [4], the RAISE protocol was proposed for vehicle-to-vehicle communications. The protocol is software-based. It allows a vehicle to verify the signature of another with the aid of a nearby RSU. However, no batch verification can be done and the RSU has to verify signatures one after another. On the other hand, to notify other vehicles whether a message from a certain vehicle is valid, a hash value of 128 bytes needs to be broadcasted. There can be tens up to thousands of signatures within a short period of time, thus the notification messages induce a heavy message overhead.

Most recently in our prior work [1], we propose two Secure and Privacy Enhancing Communications Schemes for vehicular sensor networks. The schemes can handle ad hoc messages (those sent out by arbitrary vehicles) as well as allow vehicles that know one another in advance to form a group and send group messages securely among themselves. The schemes are software-based and do not rely on any special hardware. The schemes are also based on bilinear pairing. By establishing shared secrets with RSU and TA on the handshaking phase, a vehicle is allowed to use a different pseudo identity for each session (or message) to protect its privacy while the real identity is traceable only by TA. We make use of the techniques of binary search in RSU message verification phase and bloom filter to replace hash values in notification messages to reduce the message overhead substantially and enhance the effectiveness of the verification phase. Any vehicle can form a group with other vehicles after an initial handshaking with a nearby RSU and then can authenticate and communicate with one another securely without the intervention of RSU even after moving into the region of another RSU. However, they use pairing operation for the communication between RSUs and vehicles. As pairing operation costs too much time as we just mentioned, a simpler and more effective method is required.

Several schemes about the group communications also exist. In [18], the PPGCV protocol was proposed. In addition to a scheme for group formation, they provide a protocol to update the group key. However, the setting of their scheme is different from a typical VANET and the key update process relies heavily on a key server which holds the set of all keys distributed to the vehicles. In [19], another group communications protocol, SeGCom, was proposed. However, the privacy is not considered. In our prior work [2], we provide a more comprehensive set of group communication schemes for VANETs, they support dynamic membership in a group. When a new member wants to join an existing group or an existing member wants to leave a group, there is no need to form a new group from scratch. The group secure key can be updated periodically without any help from an RSU to increase the security level of the communication. As an add-on service, members in a group can choose to send a secure message to a dedicated member in the group. To solve the noisy channel problem, they include simple acknowledgement messages in their schemes. Similar to [1], we also use pairing operation for the one-to-one communication between group members. This is not effective too. Thus in this paper, we propose a simpler and more effective solution.

In terms of secure VANET applications, Lu et al. [20] proposes a secure navigation scheme for locating parking lots in a car park while Popa et al. [21] proposes a secure and privacy preserving road toll calculation scheme under the principle of multi-party computation. In [5], the authors propose a VANET-based Secure and Privacy-preserving Navigation scheme which makes use of the collected data to provide navigation service to drivers.

Our contributions In this paper, we propose a more comprehensive set of schemes for VANETs. Our schemes satisfy all security requirement (see Section II.B) for VANETs and include the following three novel schemes: (1)We use only simple HMAC checking and symmetric encryption to replace the complicated Elliptic Curve Cryptographic (ECC) approach to achieve secure communications between Vehicles and RSUs. This is very efficient and can satisfy all the security requirements of VANETs. (2)Without using the complicated ECC approach, we realize the one-to-one secure communications scheme among group members as in the DRS paper [2] by using simple symmetric encryption and HMAC calculation based on a shared key. (3)We define a one-to-one secure communications scheme among vehicles who do not know each other in advance. There is no such scheme in all existing solutions, and our scheme supplements this shortcoming.

Organization: The rest of the paper is organized as follows: In Section II we present our assumption and the security requirement. Some preliminaries are given in Section III. We present our schemes in Section IV, the simulation of our schemes are given in Section V, we conclude our paper and discuss the future work in Section VI.

II. Assumption and Security Requirements

A. Assumptions

Recall that a vehicular network consists of on-board units (OBUs) installed on vehicles, road-side units (RSUs) along the roads, and a trusted authority (TA). We focus on the inter-vehicle communications over the wireless channel. We assume that:

1. The TA is always online and trusted by everyone. RSUs and TA communicate through a secure fixed network.
2. The RSUs have higher computation power than OBUs.
3. The RSU-to-vehicle transmission (RVC) range is at least twice of the inter-vehicle communication (IVC) range to ensure that if an RSU receives a message, all vehicles receiving the same message are in the feasible range to receive the notification from the RSU.
4. There exists a conventional public key infrastructure (PKI) for initial handshaking. The public key of the TA PK_{TA} is known by everyone. The public key of vehicle V_i PK_{V_i} is known by the TA. Also any RSU R broadcasts its public key PK_R with hello messages periodically to vehicles that are traveling in the RVC range of it. Thus PK_R is known by all vehicles nearby. There is no need for vehicles to know the public keys of other vehicles to avoid message overhead for exchanging certificates. The private keys of TA, V_i and R are SK_{TA}, SK_{V_i} and SK_R respectively and are kept secret by the corresponding party. To increase the security level, the master key s that is picked by TA for every vehicle is not preloaded into any hardware on the vehicle like [3].
5. The real identity of any vehicle is only known by the TA and itself but not by others.

B. Security Requirement

1. Message integrity and authentication: A vehicle should be able to verify that a message is indeed sent and signed by another vehicle (or a valid group member) without being modified by anyone.
2. Identity privacy preservation: The real identity of a vehicle should not be linked to any message so that other vehicles or even RSUs cannot reveal a vehicle's real identity by analyzing multiple messages sent by it.
3. Traceability: Although a vehicle's real identity should be hidden, if necessary, TA should have the ability to obtain a vehicle's real identity and relate the message to the sender (for example, in case the real identity of the sender of a fake message causing an accident needs to be revealed).
4. Confidentiality: Group messages cannot be decrypted by vehicles not in the group and a group message sent to a dedicated member can only be readable by the dedicated receiver, other vehicles (including other members) cannot decrypt the message.

III. Preliminaries

Although in our scheme, we use the HMAC to verify instead of using pairing operations, our security schemes are still using two cyclic groups with mapping called bilinear map. We first briefly introduce what a bilinear map is, and then we introduce how our method uses HMAC for signature verification.

A. Bilinear group

Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_T$ be a cyclic multiplicative group. Both groups G and $\mathbb{G}_T$ have the same prime order q. The mapping $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is called a bilinear map if it satisfies the following properties: (1) Bilinear: $\forall P, Q, R \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $\hat{e}(Q, P + R) = \hat{e}(P + R, Q) = \hat{e}(P, Q) \cdot \hat{e}(R, Q)$. Also $\hat{e}(aP, bP) = \hat{e}(P, bP)^a = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab}$. (2) Non-degenerate: There exists $P, Q \in \mathbb{G}$ such that $\hat{e}(P, Q) \neq 1_{\mathbb{G}_T}$. (3) Computable: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}$.

The bilinear map $\hat{e}$ can be constructed on elliptic curves. Each operation for computing $\hat{e}(P, Q)$ is a pairing operation. Pairing operation is the most expensive operation in this kind of cryptographic schemes. The fewer the number of pairing operations, the more efficient the scheme is. So we replace it by HMAC technique. The groups $\mathbb{G}$ and $\mathbb{G}_T$ are called bilinear groups. The security of it relies on the fact that the discrete logarithm problem (DLP) on bilinear groups is computationally hard, i.e., given the point $Q = aP$, there exists no efficient algorithm to obtain a. The implication is that we can transfer Q in an open wireless channel without worrying that a (usually some secret) can be known by the attackers.

B. Verification using HMAC

In our schemes, we use HMAC instead of pairing operation to verify a message. Because the operation time of HMAC is much shorter than that of pairing, it is very efficient. Our idea is as follows:

We assume that two parties $\mathbb{A}$ and $\mathbb{B}$ have a shared secret key t, and $ENC_t(M)$ and $HMAC_t(M)$ are the symmetric encryption and HMAC values of the message M using the key t respectively. If $\mathbb{A}$ wants to send a message M to $\mathbb{B}$, it first computes $ENC_t(M)$ and $HMAC_t(M)$, and sends them to $\mathbb{B}$, $\mathbb{B}$ decrypts $ENC_t(M)$ to get the message M, and then computes the HMAC value of the message M. If the computed HMAC value is the same as the received one, $\mathbb{B}$ accepts the message.

In the above method we mainly use a symmetric encryption and a HMAC computation under a shared secret key to protect the message. The operation of the symmetric encryption can protect the privacy of the message, which makes the message not leaked, and only the person that knows the shared secret key can get the message. Because of the irreversible of the HMAC, only the person that knows the message and the shared secret key can compute the HMAC value, this protects the integrity of the message and meanwhile it can satisfy the requirement of verification. The only shortcoming is that we need to build a shared secret key beforehand.

IV. Specifications

This section presents our new schemes. Note that our schemes are based on the framework of [1] and [2]. There are some initial parameters to be generated by TA using the following steps. This needs to be done once for the whole system unless the master key, or the real identity of a vehicle is believed to be compromised, or TA wants to update the parameters and the master key periodically to enhance the security level of the system.

1) TA chooses $\mathbb{G}$ and $\mathbb{G}_T$ that satisfy the bilinear map properties.

2) TA randomly picks $s \in \mathbb{Z}_q$ as its master key and computes $P_{pub} = sP$ as its public key. The public parameters $\{\mathbb{G}, \mathbb{G}_T, q, P, P_{pub}\}$ are publicly accessible by all RSUs and vehicles.

3) TA assigns each vehicle a real identity $RID \in \mathbb{G}$ and a password PWD. The drivers are informed about them during network deployment or during vehicle first registration.

The schemes can be divided into several modules, with details given in the following subsections.

A. Initial handshaking

This module is executed when a vehicle meets a new RSU. The vehicle authenticates itself with the TA via RSU. TA will then pass information to RSU to allow RSU to verify the vehicle's signature even if it uses pseudo identity to sign the message. Also, RSU will generate a shared secret with the vehicle. If this is the first time the vehicle authenticates itself with the TA, TA will pass the master key s and a shared secret to the vehicle. This only needs to be done once in the whole journey. To increase the security level, s is not preloaded into any hardware on the vehicle like [3]. For the shared secret with RSU, a new secret is generated every time the vehicle moves into the region of another RSU.

We use the notations $CENC_X(M)$, $CDEC_X(M)$ and $CSIG_X(M)$ to denote encrypting, decrypting and signing respectively message M using key X under conventional public key cryptographic system. The detailed processes in this module are as follows:

1. When a vehicle V_i starts up or when it finds the first RSU R after it starts up, it encrypts its RID and PWD using the TA's conventional public key CPK_{TA} and sends $CENC_{CPK_{TA}}(RID, PWD)$ to the closest RSU which in turn forwards it to the TA.
2. The TA verifies RID and PWD, and generates two shared secrets: t_i to be used between itself and V_i; m_i to be used between the connecting RSU and V_i. It then computes V_i's ID Verification Public Key as $VPK_i = t_i \oplus RID$. This VPK_i will be passed to the RSU to enable it to verify signatures from V_i even if V_i uses pseudo identity to sign the message. The TA stores the (RID, t_i, m_i) tuple and forwards VPK_i, m_i and $X = CENC_{CPK_{V_i}}(s, VPK_i, m_i, CSIG_{CSK_{TA}}(s, VPK_i, m_i))$ to the RSU. Note that to let V_i know that s, VPK_i and m_i are really sent by the TA, the TA includes its signature $CSIG_{CSK_{TA}}(s, VPK_i, m_i)$ into the encrypted text.
3. The RSU stores the (VPK_i, m_i) pair into its verification table for later usage. It then forwards X to vehicle V_i.

4. Vehicle V_i decrypts X to obtain s, VPK_i and m_i and verifies the TA's signature on them. It then computes its shared secret with the TA using $t = VPK_i \oplus RID$ and stores its shared secret m_i with the RSU.

This basically completes the initial handshaking phase. When vehicle V_i leaves the range of an RSU and enters the range of another, it includes a simpler authentication process with the TA. TA then generates a new shared secret with V_i and passes necessary information to the new RSU for verifying V_i 's signature. For details, please refer to [2].

B. Communications between Vehicles and RSUs

If V_i wants to send and sign a message M_i to an RSU nearby, the following verification procedures will be carried out:

1. V_i first generates a pseudo identity. Different pseudo identities are used for different messages to avoid being traced. To generate a pseudo identity, V_i first generates a random nonce r, then its pseudo identity becomes: $PID_i = (PID_{i1}, PID_{i2}) = (r \times P_{pub}, VPK_i \oplus H(m_i PID_{i1}))$.
2. V_i sends $< PID_i, ENC_{m_i}(M_i), HMAC_{m_i}(M_i) >$ to the RSU nearby.
3. Upon receiving the message, the RSU finds out V_i's verification public key VPK_i and shared secret m_i by checking which of stored tuples $(VPK_i,\ m_i)$ satisfies the expression $PID_{i2} = VPK_i \oplus H(m_i PID_{i1})$.
4. The RSU then decrypts $ENC_{m_i}(M_i)$ and verifies $HMAC_{m_i}(M_i)$ using m_i. If they are valid, the verification is considered to be successful.

In [1] and [2], an RSU verifies any vehicle's signatures using a complicated ECC approach. However, in our scheme, only HMAC checking is involved, and the previous asymmetric encryption of message is simplified to symmetric encryption in our scheme.

C. One to One Communications within a Group

We assume that vehicles V_1, V_2, ..., V_n have already formed a group using the scheme in [1] and [2], and their pseudo identities and group public keys are GID_1, GID_2, ...,GID_n and $GPK_1, GPK_2, ..., GPK_n$ respectively. Here $GPK_i = m_i P$. Also their shared secret key developed is $\beta = rr \times s$, here rr is generated by the RSU at the stage of the group formation and distributed to all the group members. The RSU does not know the value of s, So it does not know the shared secret key β. Now let us see how they can communicate with each other.

If V_i wants to send a message M_i to another group member V_j, the following procedures will be carried out:

1. V_i sends $< GID_i,\ GID_j,\ ENC_{\beta \times m_i \times GPK_j}(M_i),\ HMAC_{\beta \times m_i \times GPK_j}(M_i) >$ to V_j.
2. V_j verifies the message as follows. It first decrypts $ENC_{\beta \times m_i \times GPK_j}(M_i)$ using the key $\beta \times m_j \times GPK_i$ and obtains M_i. Then it verifies the HMAC signature using the key $\beta \times m_j \times GPK_i$. If the computed HMAC value is the same as the received one, V_j accepts the message as a valid one. The above checking is valid because $\beta \times m_i \times GPK_j = \beta \times m_i \times m_j P = \beta \times m_j \times m_i P = \beta \times m_j \times GPK_i$.

If later V_j wants to send a message M_j back to V_i, it sends $< GID_j, GID_i,\ ENC_{\beta \times m_j \times GPK_i}(M_j),\ HMAC_{\beta \times m_j \times GPK_i}(M_j) >$ to V_i. V_j verifies that message as what V_i does previously using the key $\beta \times m_i \times GPK_j$.

Note that β (not known by RSU) is necessary to be part of the shared key since otherwise, RSU will know how to decrypt their conversations.

The DRS paper [2] also defines a one-to-one secure communications scheme among group members. However, the mechanism is based on a complicated ECC approach. However, in this scheme, only simple symmetric encryption based on an easily-derived shared key is involved.

D. One to One Communications without a Group

This subsection discusses a secure ad-hoc communications scheme among unknown vehicles. Consider the case that a vehicle V_i wants to communicate with a nearby unknown vehicle V_j. Assume that the pseudo identities of V_i and V_j are $PID_i = (PID_{i1},\ PID_{i2}) = (r_1 \times P,\ PID_{i2})$ and $PID_j = (PID_{j1}, PID_{j2}) = (r_2 \times P, PID_{j2})$ respectively (i.e. same definition as in [1]). Then they can use the shared secret key $r_1 \times r_2 \times P$ (instead of $\beta \times m_i \times m_j \times P$ for the group scenario above) for secure communications. We will discuss the details as follows:

If V_i wants to send a message M_i to V_j, the following procedures will be carried out:

1. V_i sends to V_j $< PID_i, PID_j,\ ENC_{r_1 \times PID_{j1}}(M_i),\ HMAC_{r_1 \times PID_{j1}}(M_i) >$, where $PID_i = (r_1 \times P, PID_{i2})$ and $PID_j = (r_2 \times P, PID_{j2})$.
2. V_j verifies the message as follows. It first decrypts $ENC_{r_1 \times PID_{j1}}(M_i)$ using the key $r_2 \times PID_{i1}$ and obtains M_i. Then it verifies the HMAC signature using the key $r_2 \times PID_{i1}$. If the computed HMAC value is the same as the received one, V_j accepts the message as a valid one. The above checking is valid because $r_2 \times PID_{i1} = r_2 \times r_1 P = r_1 \times r_2 P = r_1 \times PID_{i1}$.

All vehicles involved store their current pseudo identities so that they can use the same random nonce for ongoing communications without the need of key change in the middle.

The SPECS [1] and the DRS [2] does not define any one-to-one secure communications scheme among vehicles who do not know each other in advance. Thus this scheme supplements their short-coming.

V. Security Analysis

We analyze our schemes with respect to the security requirements listed in Section II.B.

Message integrity and authentication: In our scheme, we use the packet of $ENC_{m_i}(M_i)$ and $HMAC_{m_i}(M_i)$ to realize the Message integrity and authentication of message M_i, where m_i is the shared secret between two parties. Now we show that an attacker cannot generate a valid signature.

We argue that if the shared secret key m_i is kept secret, then a vehicle's message (either ad hoc message or group message) cannot be forged by the attacker and our scheme is secure against existential forgery, adaptive chosen message attack under random oracle model. We first consider the Game between a challenger and an attacker:

Setup: The challenger starts by giving the attacker a set of system parameters.

Challenge: The challenger asks the attacker to pick one random message M_i and sign it to produce $ENC_{m_i}(M_i)$ and $HMAC_{m_i}(M_i)$.

Guess: Finally, the attacker sends one pair $ENC_{m_i}(M_i)$, $HMAC_{m_i}(M_i)$ to the challenger.

The attacker's advantage in this game is defined to be $\Pr[ENC_{m_i}(M_i)$ and $HMAC_{m_i}(M_i)$ are valid signatures]. We say that our signature scheme is secure against existential forgery, adaptive chosen message attack if the attacker's advantage is negligible.

For the first scheme in Section IV.B, m_i is the shared secret between V_i and RSU. Due to the difficulty of solving the discrete logarithm problem, only V_i and RSU knows m_i. For the second scheme in Section IV.C, we can see that the secret key between V_i and V_j is $\beta \times m_i \times GPK_j = \beta \times m_i \times m_j P = \beta \times m_j \times m_i P = \beta \times m_j \times GPK_i$. Due to the diffculty of solving the discrete logarithm problem and $\beta = rr \times s$ is only known by vehicles, only V_i and V_j know the shared secret key. Similarly to this, the third scheme in Section IV.D can also protect the shared secret key. But this scheme is for unknown vehicles to communicate with each other, so we do not define any authentication additional in our scheme.

From above we can see that the attacker can not obtain the shared secret key, and we can regard the HMAC as the random oracle, so we cannot forgery or differ any valid message, and the attacker's advantage is negligible, our scheme is also security.

Identity privacy preservation: We show that an attacker cannot obtain a vehicle's real identity even it is keeping its pseudo identity. We argue that if DDH is hard, then the pseudo identity of a vehicle will not leak any information of its real identity. we prove it as follows.

We first consider Game 1 between a challenger and an attacker:

Setup: The challenger starts by giving the attacker a set of system parameters including P and P_{pub}.

Choose: The attacker then freely chooses two verification public keys VPK_0 and VPK_1 and sends them to the challenger.

Challenge: The challenger sets a bit x = 0 with probability 1/2 and sets x = 1 with probability 1/2. The challenger then sends the attacker the pseudo identity corresponding to VPK_b together with the group public key.

Guess: The attacker tries to guess the value of x chosen by the challenger, and outputs its guess, x_0.

The attacker's advantage in this game is defined to be $Pr[x = x_0] - 1/2$. We say that our pseudo identity generation algorithm is semantically secure against a chosen plain text attack (CPA) if the attacker's advantage is negligible. Next we assume that we have an algorithm A which runs in polynomial time and has a non-negligible advantage e as the attacker in Game 1. We will construct Game 2 in which a Decisional DiffieCHellman (DDH) attacker B can make use of A to achieve a non-negligible advantage in breaking DDH. B is given a DDH instance (P, aP, bP, T) as input and he is asked to determine whether $T = abP$. We further let t denote a bit that B is trying to guess (i.e. $t = 0$ for positive answer $T = abP$ while $t = 1$ for negative answer $T - abP$). Game 2 runs as follows:

Setup: Based on the DDH instance, B makes up the parameters $(P, P_{pub} = aP)$ and gives them to A. Note that a now plays the role of s in our scheme.

Choose: A then chooses two verification public keys VPK_0 and VPK_1 which it has queried for the corresponding group public keys, m_0P and m_1P respectively, before and sends them to B.

Challenge: B is playing the role of challenger here, so it sets a bit x randomly and generates the pseudo identity $PID = (PID_1, PID_2)$, where $PID_1 = raP, PID_2 = VPK_x \oplus H(rabP)$ and r is a random nonce and sends to A. B also sends A the group public key bP. (Note that b now plays the role of the RSU-vehicle shared secret m_i in our scheme.)

Guess: Finally A sends B a bit x_0 as its guess for x. B answers the DDH problem positively that $T = abP$ if B's guess is correct (i.e. $x = x_0$). Now let us look at why B can answer the DDH problem in this way. If $t = 0$ (i.e. $T = abP$), then $PID_2 = VPK_b \oplus H(-rabP) = VPK_b \oplus H(bPID_1)$ is a valid pseudo identity in proper format. In this case, since A has non-negligible advantage in the game described above, it is likely that A can break our system and can guess x correctly with probability $1/2 + \varepsilon$. Thus, $Pr[Bsucceeds|t = 0] = 1/2 + \varepsilon$. If $t = 1$, we claim that $Pr[Bsucceeds|t = 1] = 1/2$ only. To see why, we observe that when T is randomly chosen, the term $H(rT)$ in PID_2 cannot be cancelled by the term $H(bPID_1)$ and so there is no way to obtain VPK_x. Thus the computation reveals no information about x. In this sense, the value of x is hidden to A, so even A can break our system, the probability that he will guess x correctly is simply 1/2 (by tossing a fair coin). Hence, $Pr[Bsucceeds] = 1/2 \times (1/2 + \varepsilon) + 1/2 \times 1/2 = 1/2 + \varepsilon/2$. Since ε is non-negligible, B can solve the DDH problem but this violates the assumption that DDH is hard. Therefore, our scheme is secure in the sense that the pseudo identity of a vehicle can preserve its real identity.

Traceability: To reveal the real identity of the sender of a message, TA is the only authorized party that can perform the tracing. Given the pseudo identity PID_i of the vehicle V_i and its shared secret with the connecting RSU m_i, TA can search through all the stored (RID_j, t_j) pairs from its repository. Vehicle V_i' real identity is the RID_j value from the entry that satisfies the expression $PID_{i2} \oplus t_j \oplus H(m_i PID_{i1}) = RID_j$. No other party can obtain vehicle V_i's real identity since t_i is only known by the TA and V_i itself. Upon getting V_i's real identity RID_i, TA can revoke it if necessary. This can be done by simply storing RID_i into a revocation list. V_i can no longer obtain VPK_i from it in the future.

Confidentiality: For the first scheme in Section IV.B, the message M_i is protected by the shared secret key m_i, the ciphertext under symmetric encryption cannot leak any information about the message. Similarly for the schemes in Section IV.C and IV.D, confidentiality is protected by the shared secret key developed between the vehicles.

VI. Simulation Results

In this section, we show the simulation results for comparing our schemes with SPECS [1], DRS [2] and IBV [3]. Note that IBV also uses a batch verification scheme, so we choose it as our comparison. We first compare our scheme in Section IV.B with SPECS [1] and IBV [3] in terms of delay. In [5], it finds that the pairing operation does not take constant time, and so we consider both cases. In SPECS1, we consider the pairing time as a constant and in SPECS2 we consider that the pairing time increases linearly as the input parameters increases. Through simulation, we find that since we do not rely on RSUs for the verification of one-to-one messages, the delay caused by our scheme is much shorter. We also compare our schemes in Section IV.C and IV.D with the DRS [2] in terms of delay. We find our schemes are more effective.

A. The simulation model

We implement our schemes, the SPECS schemes, IBV and the DRS schemes on a simulator written in C++. Some of the settings and parameters of our simulation are adopted from [1 - 4]. We assume that there is a highway of length 10 km. A number of RSUs are installed along it. The number of RSUs is a variable and these RSUs are evenly distributed along the highway. Groups of vehicles are traveling on it at speeds varying from 50 km/h to 70 km/h (common cases in Hong Kong). For each group, the number of vehicles is a variable and the vehicles are traveling on the road one after another. The RSU-to-Vehicle Communication (RVC) and the Inter-Vehicle Communication (IVC) ranges are set to 600 m and 300 m respectively, i.e. when a vehicle enters the 600 m RVC range of the RSU, the messages sent by the RSU can be received by it. Two vehicles that are within the 300 m IVC range of each other can communicate. Inter-vehicle messages are sent every 500 ms at each vehicle. IEEE 802.11a is used to simulate the medium access control layer. That is, when a vehicle wants to transmit, it first detects whether the channel is available. If another vehicle is transmitting, it waits until that transmission is completed and then waits for a random delay period before it begins to transmit. The bandwidth of the channel is 6 Mb/s and the average length of inter-vehicle message (not including the cryptographic bits) is 200 bytes. The RSU performs batch verification every 300 ms and each pairing operation is assumed to take 6 ms while each HMAC operation is assumed to take 0.006 ms. We obtained these benchmark values based on an implementation using the PBC library [23]. We set the message size to 512 bits and we run the test for 500 times to obtain an average value.

In the simulation of scheme in Section IV.B, our simulation runs for 1000s. We first vary the total number of vehicles that have ever entered RSU's RVC range during the simulation period from 200 to 1000 in steps of 200 to simulate the impact of different traffic densities. We then vary the inter-vehicle message signature error rate from 1% to 10% to interpret its impact on the performance of our schemes. In the simulation of scheme in Section IV.C and IV.D, our simulation runs for 3600s, and we consider the time of group arrival interval as 60s. We vary the total number of vehicles from 60 to 600 in steps of 60 to simulate the impact of different traffic densities.

B. The simulation results

Based on the concept of RAISE [4] and SPECS [1], we define the average delay of a message as:

$$Delay = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M_i}\sum_{m=1}^{M}(T_{verf}^{m} - T_{recv}^{m})$$

where M_i is the number of messages received by vehicle V_i, T_{recv}^{m} is the time that vehicle V_i receives the verification notification message of message m from the RSU and T_{verf}^{m} is the time that vehicle V_i receives message m from its neighboring vehicle.

Note that In RAISE [4], it defines the average message loss ratio (LR) to express the success rate of the message, and in SPECS [1], it processes the messages with batch verification and deal with the invalid batch with the bloom filter, so it extends LR to IBSR to express the success rate for messages that are included in batches with invalid signatures. It defines as follows:

$$SuccessRate = \frac{1}{N}\sum_{i=1}^{N}(\frac{M_{app}^{i}}{M_{mac}^{i}})$$

where M_{app}^{i} represents the total number of messages that are successfully verified by the RSU and are consumed by vehicle i in the application layer before vehicle i leaves RSU's IVC range. Also the signatures of these messages are batch-processed with at least one invalid signature by the RSU. M_{mac}^{i}, on the other hand, represents the total number of messages received by both vehicle i and RSU in the medium access control layer from other vehicles and again the signatures of these messages are being batch-processed with at least one invalid signature by the RSU. However, in our schemes, instead of the patch-processing we deal with the messages one by one, and the receiver must be able to verify every message, so we do not have the concept of IBSR, and to compare with them is meaningless.

We divided our simulation into the next 4 experiments:

B.1 Experiment 1

In this experiment, we simulate the delay under the schemes we discussed in Section IV.B(Communications between Vehicles and RSUs), Our simulation runs for 1000 s. We vary the number of vehicles from 200 to 1000 in steps of 200, and we fix the error rate to 5%. To express it clearly we divided the results into Figure 1 and Figure

2. In this experiment, we use "SPECSi-BSx" to denote the schemes of SPECS that have different levels of binary search as mentioned in SPECS [1]. In SPECS1 and IBV1, we consider the pairing time as a const, and in SPECS2 and IBV2 we consider that the pairing time increases linearly as the input parameters increase. And we use "Scheme1" in this experiment to denote our scheme.

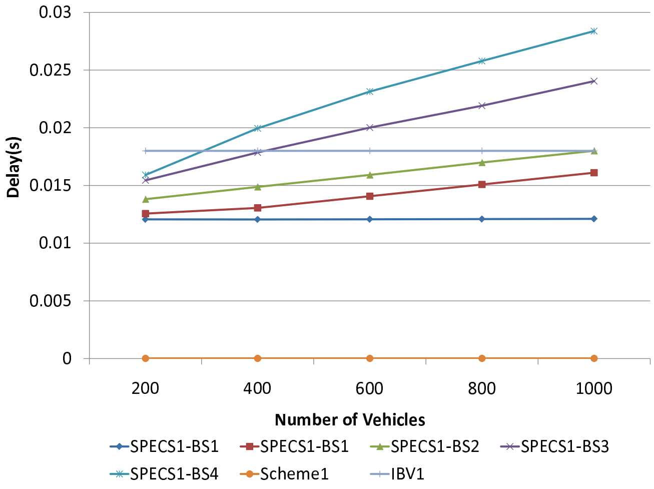

Fig. 1. Delay vs. Number of Vehicles

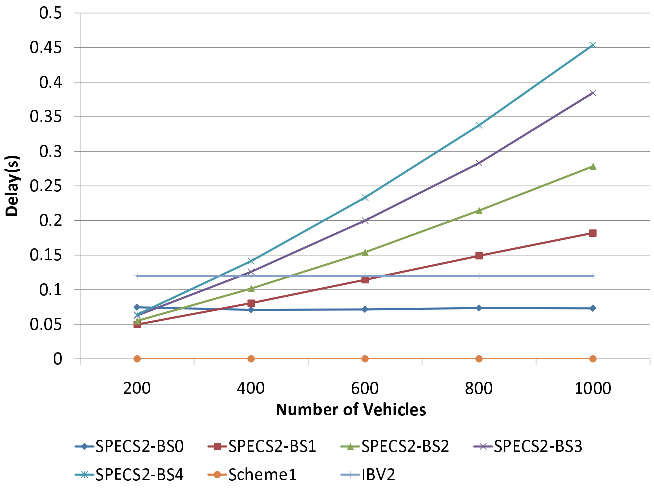

Fig. 2. Delay vs. Number of Vehicles

From this experiment we can see that the delay under the IBV protocol and the SPECS schemes are very close to each other, and in our scheme we deal with the message one by one, so the delay caused by our scheme is almost not changed, and the average delay caused by our scheme is only 1.6×10^{-5} s, and no matter whether we consider the pairing time as const or it increases linearly as the input parameters increase, the delay caused by our schemes is much lower than SPECS and IBV. In figure 1 we consider the pairing time as const and the delay is at least 1.2×10^{-2} s, and in figure 2 we consider the pairing time increases linearly as the input parameters increase and the delay is at least 5×10^{-2} s. So the average delay caused by our scheme is nearly thousands of times lower than prior schemes.

B.2 Experiment 2

The only difference between this experiment and Experiment 1 is that we vary the error rate from 1% to 10% in steps of 1%, and we fix the number of vehicles to 1000. The simulation result is shown in Figure 3 and Figure 4. Similarly to experiment 1 the average delay caused by our scheme is only 1.6×10^{-5} s, and no matter whether we consider the pairing time as const or it increases linearly as the input parameters increase, the delay caused by our schemes is much lower than SPECS and IBV. In figure 3 we consider the pairing time as const and the delay is at least 1.2×10^{-2} s, and in figure 2 we consider the pairing time increases linearly as the input parameters increase and the delay is at least 1.7×10^{-1} s. So the average delay caused by our scheme is much lower than prior schemes.

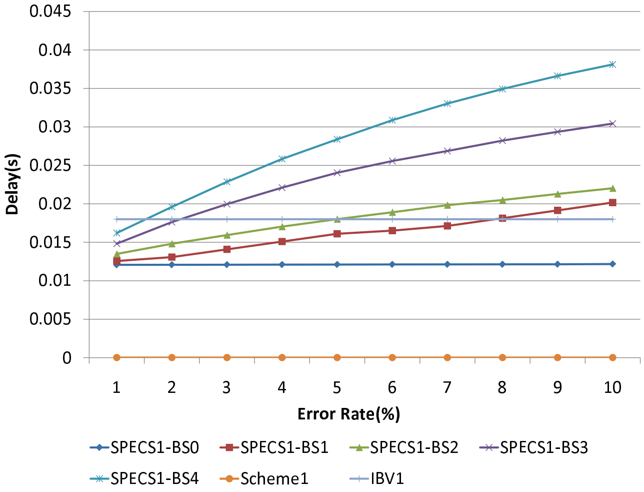

Fig. 3. Delay vs. Error Rate

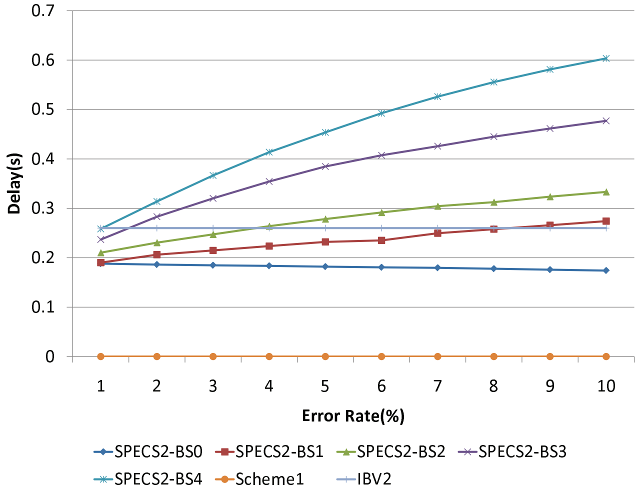

Fig. 4. Delay vs. Error Rate

B.3 Experiment 3

In this experiment, we simulate the delay of our scheme discussed in Section IV.C(One-to-One Communications within a Group). Our simulation runs for 3600s and we set the group arrival interval as 60s. We vary the number of vehicles from 60 to 600 in steps of 60. Then we get the results as in Table I.

TABLE I
DELAY VS. NUMBER OF VEHICLES

NO. of Vehicles	AD of DRS	AD of Scheme2
60	0.0123 s	0.000312 s
120	0.0123 s	0.000312 s
180	0.0123 s	0.000312 s
240	0.0123 s	0.000312 s
300	0.0123 s	0.000312 s
360	0.0123 s	0.000312 s
420	0.0123 s	0.000312 s
480	0.0123 s	0.000312 s
540	0.0123 s	0.000312 s
600	0.0123 s	0.000312 s

In Table I, we use "AD of DRS" to denote the average delay of the schemes of DRS [2] and "AD of Scheme2" to denote the average delay of our scheme discussed in Section IV.C. We only consider the one-to-one communication. From our experiment we can see that the average delay caused by our scheme is only 0.000312 s, and the average delay caused by DRS [2] is 0.0123 s. Thus our scheme is much better.

B.4 Experiment 4

In this experiment, we simulate the delay of our scheme discussed in Section IV.D(One-to-One Communications without a Group). Our simulation runs for 3600s and we set the group arrival interval as 60s, We vary the number of vehicles from 60 to 600 in steps of 60. Then we get the results in Table II.

TABLE II
DELAY VS. NUMBER OF VEHICLES

NO. of Vehicles	AD of DRS	AD of Scheme3
60	9.00081 s	0.000312 s
120	9.00081 s	0.000312 s
180	9.00081 s	0.000312 s
240	9.00081 s	0.000312 s
300	9.00081 s	0.000312 s
360	9.00081 s	0.000312 s
420	9.00081 s	0.000312 s
480	9.00081 s	0.000312 s
540	9.00081 s	0.000312 s
600	9.00081 s	0.000312 s

In Table II, we use "AD of DRS" to denote the average delay of schemes of DRS [2] and use "AD of Scheme3" to denote the average delay of our scheme in Section IV.D. We only consider the one-to-one communication. Our scheme does not include the formation of a group, and the delay of DRS is composed of the delay for group formation and that for one-to-one communication. From our experiment we can see that the average delay caused by our scheme is only 0.000312 s, and the average delay caused by DRS [2] is 9.00081 s. Thus our scheme is much better.

VII. Conclusions and Future Work

In this paper, we propose a more comprehensive set of schemes for VANETs. Our schemes satisfy all security requirement for VANETs and is more efficient than existing solutions. We use only HMAC checking and symmetric encryption to replace the complicated Elliptic Curve Cryptographic (ECC) approach to achieve secure communications between vehicles and RSUs. Compared to using the complicated ECC approach to realize one-to-one secure communications among group members as in DRS [2], we achieve this only using simple symmetric encryption and HMAC calculation based on a shared key. We also define a one-to-one secure communications scheme among vehicles who do not know each other in advance, which fill in the gaps of previous schemes.

In SPECS, a number of message exchanges between vehicles and an RSU are involved when they want to form a group. Now any two vehicles have a shared secret. In the future, we will investigate the possibility of simplifying the group formation procedures involved. Also we will take into consideration the impact of hidden terminal problem and different values for group arrival interval.

Acknowledgement This work was partially supported by the National Natural Science Foundation of China (NSFC Grant No.60931160442 and No.61133013) and The Tsinghua University Initiative Scientific Research Program (No.2009THZ01002). The first author thanks the support of his supervisor Prof. Xiaoyun Wang. On the other hand, Victor O.K. Li is supported in part by a Visiting Professorship at the Department of Computer Engineering, King Saud University, Saudi Arabia. Also we thank the reviewers for their constructive comments.

References

[1] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K Li, *SPECS: Secure and Privacy Enhancing Communications for VANET*, 2009, manuscript

[2] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K Li, *DRS: Dynamic, Reliable and Secure Group Communications Schemes for Vehicular Sensor Networks*, 2009, HKU Technical Report No. TR-2009-06.

[3] C. Zhang, R. Lu, X. Lin, P.H. Ho, X. Shen, *An efficient identity-based batch verification scheme for vehicular sensor networks*, in: Proceedings of the IEEE INFOCOM'08, April 2008, pp. 816 - 824.

[4] C. Zhang, X. Lin, R. Lu, P.H. Ho, *RAISE: an efficient RSU-aided message authentication scheme in vehicular communication networks*, in: Proceedings of the IEEE ICC'08, May 2008, pp. 1451 - 1457.

[5] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K Li, *VSPN: VANET-based Secure and Privacy-preserving Navigation*, manuscript,

[6] F. Wang, D. Zeng, L. Yang, *Smart cars on smart roads: an IEEE*

intelligent transportation systems society update, IEEE Pervasive Computing 5 (4) (2006) 68-69.

[7] H. Oh, C. Yae, D. Ahn, H. Cho, *5.8 GHz DSRC packet communication system for ITS services*, in: Proceedings of the IEEE VTC'99, September 1999, pp. 2223-2227.

[8] U.S. Department of Transportation, National Highway Traffic Safety Adimistrtation, *Vehicle Safety Communications Project, Final Repot.* April 2006.

[9] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF RFC5280, 2008.

[10] M. Raya and J. P. Hubaux, *Securing vehicular ad hoc networks*, Journal of Computer Security, Vol. 15, No. 1, pp. 39-68, 2007.

[11] J. Freudiger, M. Raya, and M. Feleghhazi, *Mix Zones for Location Privacy in Vehicular Networks*, in Proncessings of the First International Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS'07), Vancouver, Canada, August 14, 2007.

[12] H. Wen, P.H. Ho, G. Gong, *A novel framework for message authentication in vehicular communication network*, in: Proceedings of the IEEE GLOBECOM'09, December 2009, pp. 1-6.

[13] A. Wasef, X. Shen, *MAAC: message authentication acceleration protocol for vehicular ad hoc networks*, in: Proceedings of the IEEE GLOBECOM'09, December 2009, pp. 1-6.

[14] X. Sun, X. Lin, and P-H. Ho, *Secure Vehicular Communications Based on Group Signature and ID-based Signature Scheme*, in Proceedings of International Conference on Communications (ICC'07), Scotland, June, 2007.

[15] B.K. Chaurasia, S. Verma, S.M. Bhasker, *Message broadcast in VANETs using group signature*, in: Proceedings of the IEEE WCSN'09, December 2008, pp. 131-136.

[16] Y. Hao, Y. Cheng, K. Ren, *Distributed key management with protection against RSU compromise in group signature based VANETs*, in: Proceedings of the IEEE GLOBECOM'08, December 2008, pp. 1-5.

[17] A. Studer, E. Shi, F. Bai, A. Perrig, *TACKing together efficient authentication, revocation, and privacy in VANETs*, in: Proceedings of the IEEE SECON'09, June 2009, pp. 1-9.

[18] A. Wasef and X. Shen, *PPGCV: Privacy Preserving Group Communications Protocol for Vehicular Ad Hoc Networks*, in IEEE Proceedings of the ICC'08, May 2008, pp. 1458-1463.

[19] M. Verma and D. Huang, *SeGCom: Secure Group Communication in VANETs*, in IEEE Proceedings of the CCNC'09, Jan. 2009, pp. 1-5.

[20] R. Lu, X. Lin, H. Zhu, X. Shen, *SPARK: a new VANET-based smart parking scheme for large parking lots*, in: Proceedings of the IEEE INFOCOM'09, April 2009, pp. 1413-1421.

[21] R.A. Popa, H. Balakrishnan, A.J. Blumberg, *VPriv: protecting privacy in location-based vehicular services*, in: Proceedings of the 18th USENIX Security Symposium, September 2009.

[22] J. Baek, B. Lee, and K. Kim, *Secure Length-Saving ElGamal Encryption under the Computational Diffie-Hellman Assumption*, Lecture Notes in Computer Science - Information Security and Privacy, Vol. 1841, pp. 49 C 58, 2000.

[23] *The Pairing-Based Cryptography Library*, http://crypto.stanford.edu/pbc/.

[24] RFC 2104 - *HMAC: Keyed-Hashing for Message Authentication*, http://www.ietf.org/rfc/rfc2104.txt.

[25] N. Koblitz, *elliptic curve cryptosystems*, in Mathematics of Computation 48, 1987, pp. 203-209

[26] V. Miller, *Use of elliptic curves in cryptography*, CRYPTO 85, 1985.