A Distributed Key Management Framework with Cooperative Message Authentication in VANETs

Yong Hao, *Student Member, IEEE*, Yu Cheng, *Senior Member, IEEE*,
Chi Zhou, *Senior Member, IEEE*, and Wei Song

***Abstract*—In this paper, we propose a distributed key management framework based on group signature to provision privacy in vehicular ad hoc networks (VANETs). Distributed key management is expected to facilitate the revocation of malicious vehicles, maintenance of the system, and heterogeneous security policies, compared with the centralized key management assumed by the existing group signature schemes. In our framework, each road side unit (RSU) acts as the key distributor for the group, where a new issue incurred is that the semi-trust RSUs may be compromised. Thus, we develop security protocols for the scheme which are able to detect compromised RSUs and their colluding malicious vehicles. Moreover, we address the issue of large computation overhead due to the group signature implementation. A practical cooperative message authentication protocol is thus proposed to alleviate the verification burden, where each vehicle just needs to verify a small amount of messages. Details of possible attacks and the corresponding solutions are discussed. We further develop a medium access control (MAC) layer analytical model and carry out NS2 simulations to examine the key distribution delay and missed detection ratio of malicious messages, with the proposed key management framework being implemented over 802.11 based VANETs.**

***Index Terms*—Vehicular ad hoc networks, privacy, distributed key management, RSU compromise, cooperative authentication.**

I. Introduction

THE VEHICULAR ad hoc networks (VANETs) have attracted a lot of attentions due to their interesting and promising functionalities including vehicular safety, traffic congestion avoidance, and location based services [1]. In this paper, we focus on safety driving application, where each vehicle periodically broadcasts messages including its current position, direction and velocity, as well as road information.

Privacy is an important issue in VANETs [2]. As the wireless communication channel is a shared medium, exchanging messages without any security protection over the air can easily leak the information that users may want to keep private. Pseudonym based schemes [3]–[5] have been proposed to preserve the location privacy of vehicles. However, those schemes require the vehicles to store a large number of pseudonyms and certifications, and do not support some important secure functionalities such as authentication and integrity. The *group signature* [6] is a promising security scheme to provide privacy in VANETs. To the best of our knowledge, all of the existing group signature schemes in VANETs [7]–[9] are based on *centralized key management* which preloads keys to vehicles off-line. The centralized key management has some disadvantages. For instance, the system maintenance is not flexible. Another issue regarding the centralized key management is that many existing schemes assume a tamper-proof device [1] being installed in each vehicle. The tamper-proof device normally costs several thousand dollars, such as IBM 4764 card [10]. The framework to be developed in this paper does not require the expensive tamper-proof device.

In this paper, we propose and develop a secure *distributed key management* framework. In our framework, the road side units (RSUs) [11] are responsible for secure group private keys distribution in a localized manner. When a vehicle approaches an RSU, it gets the group private key from the RSU dynamically. All vehicles which get the group private key from the same RSU form a *group*. A new issue induced by the distributed key management framework is that compromised RSUs may misbehave in the key distribution procedure. For example, a compromised RSU may deliver other vehicles' group private keys to its accomplice. Then, the accomplice can send messages under the name of other vehicles. Therefore, we develop security protocols for the distributed key management framework, which are capable of detecting the compromised RSUs and their collusion with the malicious vehicles if any.

Computation overhead is another critical issue in VANETs. In the safety driving application, vehicles broadcast safety messages every 300ms [1]. Since the group signature is expensive, the computation overhead of each vehicle will become intolerable when the density of vehicles is high [12]. In [13], the authors propose a promising protocol which let vehicles verify messages cooperatively by employing probabilistic verification. However, in order to guarantee efficient cooperation, vehicles have to verify at least twenty-five messages within 300ms which is still a heavy computation burden for the on-board unit (OBU) installed on a vehicle. In addition, the impact of packet loss at the medium access control (MAC) layer on security performance is not investigated in [13].

In this paper, we propose a more efficient and practical cooperative message authentication protocol (CMAP) with an assumption that each safety message carries the location information of the sender vehicle (which can be generated by a global positioning system (GPS) device). Verifiers of each message are defined according to their locations in relation to

Manuscript received 5 January 2010; revised 7 May and 12 July 2010. This work was supported in part by NSF grant CNS-0832093.

Y. Hao, Y. Cheng, and C. Zhou are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616, USA (email: {yhao4, cheng, zhou}@iit.edu).

W. Song is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, 3EB 5A3, Canada (email: wsong@unb.ca).

Digital Object Identifier 10.1109/JSAC.2011.110311.

0733-8716/11/$25.00 © 2011 IEEE

the sender. Only the selected verifiers check the validity of the message while other vehicles rely on verification results from these verifiers. Compared with [13], our protocol has smaller packet loss ratio, less computation and communication overhead, as well as better security performance. Hence, it is more efficient and practical in the real application. In summary, this paper has five-fold main contributions:

1) We propose a *distributed key management framework* which has advantages in the revocation of malicious vehicles, system maintenance, and the implementation of heterogeneous security policies.
2) We develop a secure key distribution protocol with the capability of preventing RSUs from misbehaving. The protocol guarantees the traceability of compromised RSUs and malicious vehicles.
3) An efficient cooperative message authentication protocol is developed, by which cooperative verifiers are intelligently selected to significantly reduce the computation and communication overhead in the group signature based implementation.
4) A MAC layer analytical model is developed to quantitatively evaluate the impact of number of verifiers and the size of authentication messages on network utilization.
5) We carry out NS2 simulations of 802.11 based VANETs to examine the key distribution delay and missed detection ratio of malicious messages, with the proposed key management framework being applied.

The remainder of this paper is organized as follows. Section II reviews more related work. Section III describes the system model. Section IV presents the distributed key management framework and associated security protocols for implementation. The cooperative message authentication protocol is developed in section V. Section VI and Section VII analyze the security performance and MAC layer performance, respectively. Section VIII presents the NS2 simulation results. Section IX gives the conclusion remarks.

II. Related Work

A. Privacy

There have been several proposals for privacy preservation of VANETs. Using pseudonyms is a natural idea. It is preferable to preserve the location privacy of a vehicle by breaking the linkability between two locations, for which the vehicle can update its pseudonym after each transmission. Considering that a powerful adversary may still link the new and old pseudonyms by monitoring the temporal and spatial relations between new and old locations, the techniques of mix zone [3] and silent period [4] have been proposed to enhance the pseudonym scheme. Each vehicle in a mix zone will keep silent in transmission, and randomly update its pseudonyms when it travels out of the mix zone and becomes reactivated. Given a reasonable large mix zone, the location privacy can be well protected due to the untraceability of location and pseudonym updating in the silent period. In the AMOEBA [5], vehicles form groups. The messages of all group members are forwarded by the group leader, which implies that the privacy of group members is protected by sacrificing the privacy of group leader. Moreover, if a malicious vehicle is selected as a group leader, all group members' privacy may be leaked by the malicious leader.

While the pure pseudonym schemes do not support the secure functionality of authentication, integrity, and nonrepudiation, an anonymous signing protocol [1] is proposed to provide such functions as well as privacy. In the protocol, each vehicle preloads a large number of certificated anonymous public/private key pairs. A key pair will be used for a short period of time and then be discarded. Each key pair is assigned to only one user, and authorities maintain the key distribution records which can be used to trace possible malicious vehicles. The shortcoming of this protocol is that it requires vehicles to store a large number of pseudonyms and certifications, where a revocation scheme for abrogating malicious vehicles is difficult to implement.

The *group signature* [6] is a promising security scheme to provide privacy in VANETs. In the group signature, one group public key is associated with multiple group private keys. Under the group signature scheme, although an eavesdropper can know that a message is sent by the group, it can not identify the sender of the message. A general vehicular communication framework based on group signature is given in [7]. Lin *et. al.* systematically discuss how to implement group signature protocol in VANETs [8]. The work in [9] combines pseudonym schemes with the group signature to avoid storing pseudonyms and certifications in vehicles. While all these studies assume a centralized key management scheme, we develop a distributed key management framework in this paper to achieve privacy based on group signature.

B. Computation Overhead

In the safety driving application with frequent message communication, it is important to design protocols with small computation overhead for timely and reliable message processing. In [15], the authors propose to employ TESLA, which is a hash based protocol, to reduce the computation overhead. However, the malicious vehicles could not be identified in this protocol. An aggregate signature and certificates verification scheme is proposed in [16], which could verify all received signatures and certificates at one time. This protocol is more efficient when the density of vehicles is high. An RSU aided message authentication protocol is proposed in [14]. The protocols requires RSUs to cover all the area, because RSUs have to be involved in the authentication. A promising protocol based on probabilistic verification is proposed in [13]. Through cooperative verification, the number of messages to be authenticated by each vehicle will be reduced considerably. In this paper, we adopt the concept of cooperative authentication, but design a new method to select verifiers. With our method, a similar security level could be achieved with a much smaller number of verifiers, and the performance is more robust when the MAC-layer collision is nonignorable.

C. Communication Protocols for VANETs

A vehicular network can be established over different communication/networking protocols [11], [17], say, cellular networks, IEEE 802.16 (WiMAX), or IEEE 802.11. There are already some cellular-based vehicular communication services on the market, for example, the GM OnStar service and the BMW Assist service. However, cellular or WiMAX based networking is limited to single-hop base station to vehicle communications, and can hardly be applied to ad hoc vehicle to vehicle communications. Moreover, cellular and WiMAX networking heavily depend on the availability of infrastructure, which is normally expensive and might not be available in those underdeveloped areas. The cellular network is further limited with bandwidth and not suitable for large scale multi-hop vehicle to vehicle networking. The 802.11 based protocol has the flexibility in seamlessly supporting both single-hop RSU to vehicle communications and multi-hop vehicle to vehicle communications, and is the mainstream protocol for VANETs [12]–[14], [18]–[20]. In this paper, we also focus on the 802.11 based VANETs.

III. System Model

A. Network Model

We consider infrastructure based VANETs in this paper, where entities can be classified into three categories: authorities, road side infrastructure, and nodes.

Authorities are responsible for key generation and malicious vehicle judgement. Authorities have powerful firewalls and other security protections. Therefore, they have the highest security level. We assume that they can not be compromised.

Road side infrastructure consists of RSUs deployed at the road sides which are in charge of key management in our framework. Traffic lights or road signs can be used as RSUs after renovation. RSUs communicate with authorities through wired network. We assume a trusted platform module is equipped in each RSU. It can resist software attacks but not sophisticated hardware tampering. The cost of a trusted platform module is only a few tens of dollars which is affordable [1]. RSUs are semi-trust with the medium security level [5].

Nodes are ordinary vehicles on the road that can communicate with each other and RSUs through radio. We assume that each vehicle is equipped with a GPS receiver using DGPS [21] with an accuracy on the order of centimeters and an on board unit (OBU) which is in charge of all communication and computation tasks. Nodes have the lowest security level.

B. Group Signature Based Privacy System

In our framework, the communications can be divided into the *key distribution phase* and the *regular broadcast phase*. Vehicles get keys dynamically in the key distribution phase and then start to broadcast their geographic and road condition messages periodically in the regular broadcast phase. We resort to the group signature scheme for privacy provision. With group signature, members of a group sign messages under the name of the group. In a group, there are one group public key and many corresponding group private keys. A message that is signed by any group private keys can be verified with the unique group public key, and the signer's identifier will not be revealed. However, authorities hold a tracing key which can be used to retrieve the group private key from the signature. If one group private key is assigned to only one user, the signer can be identified after authorities get its group private key.

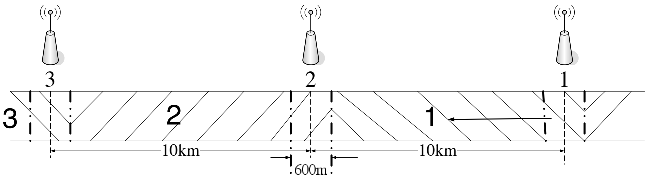


Fig. 1. Group definition.

1) Group Definition: Those vehicles getting keys from the same RSU form a group, as illustrated in Fig. 1, where the communication range of RSUs is 300 meters marked by dashed lines. We consider that RSUs are only deployed at entrances/exits of the road segments. In a highway scenario, RSUs are normally far away from each other. In the region out of the RSU coverage, vehicles in the same group can communicate with each other in an ad hoc manner. In a city area, RSUs might overlap with each other. We define that a vehicle is only associated with one RSU at a moment to get the service.

2) Channel Assignment: In the VANETs, vehicles share the wireless spectrum according to the 802.11p [18] which has seven communication channels. One is used as the control channel for management data and short messages exchange. There is also one accident avoidance channel for safety messages broadcasting. In our system, the key distribution process employs the control channel and regular broadcast messages are transmitted in the accident avoidance channel.

C. Security Model

In this paper, we assume that attackers are inside, rational, active, global [22] and parsimonious [23]. Inside attackers are legitimate members of VANETs. In this paper, attackers can be network nodes or road side infrastructure. Rational attackers only attack for their own benefits. They know the security mechanism and they want to attack without being detected. If there is a mechanism that can detect them and the punishment is severe enough, they tend not to attack. Active attackers have the ability to send packets into wireless channels. Global attackers have an unlimited scope which means they can hear any information in the network. Attackers may have strong transmission power to communicate over long distances. Adversarial parsimony means an attack involving a few malicious nodes is more likely to happen than an attack that requires collusion among a large number of nodes.

We assume that the overwhelming majority of vehicles and RSUs are honest which is reasonable in the civilian use system. We also assume vehicles will report to authorities when they find that other vehicles send a false message. Wired network which connects RSUs and authorities transmits data securely without packet loss. In the key distribution phase, our protocol is used to judge whether a vehicle is a legitimate user. If accusers and the accused are all legitimate users, we assume authorities have an evaluation system [24] to judge whether the contents of messages are false or not. The evaluation system design is out of the scope of this paper.

IV. Distributed Key management

A. Short Group Signature

We adopt short group signature [25] in this paper because it has smaller communication overhead than other group signature schemes. Meanwhile, in the short group signature protocol, there is a group private key generator which can be assigned to key distributors without revealing other secrets. The existence of the generator makes the third party possible to be key distributors. Another attractive feature of the short group signature is that it has a tracing key which can retrieve group private keys from signatures. The short group signature works as following [26]:

1) Key Setup: Authorities generate cryptographic system in this procedure. Let G_1 and G_2 be two bilinear multiplicative groups with generators g_1 and g_2 of the same prime order *p*, respectively. Let ψ be a computable isomorphism from G_2 to G_1 with $\psi(g_2) = g_1$. For the group *t*, authorities select $h_t \leftarrow G_1 \backslash \{1_{G_1}\}$ and ξ_{t1}, ξ_{t2}, $\gamma_t \leftarrow Z_p^*$ randomly and set μ_t, ν_t $\in G_1$, such that $\mu_t^{\xi_{t1}} = \nu_t^{\xi_{t2}} = h_t$, where Z_p^* is a multiplicative group of order *p*-1. Set $\omega_t = g_2^{\gamma_t}$. Authorities publish the group public key (g_1, g_2, μ_t, ν_t, h_t, ω_t) and transmit the group private key generator γ_t to the key distributor of group t, in other words, RSU_t, securely. The group tracing key K_t = (ξ_{t1}, ξ_{t2}) will be held by authorities.

2) Membership Registration: When a user *k* applies to join the group t, the key distributor will generate group private key by selecting $x_{tk} \leftarrow Z_p^*$ randomly and sets $A_{tk} = g_1^{1/(\gamma_t + x_{tk})}$. The group private key for the user k is G_{pri_k} = (A_{tk}, x_{tk}). It will be transmitted to the user securely after the RSU_t receives the valid information of the user, such as its real identifier. Each group private key should only be assigned to one user.

3) Signing and Verification: Vehicles start to sign regular broadcast messages by using the group private key after they pass the corresponding RSU. Receivers only accept messages that are approved by group public key in the verification.

4) Key Retrieve: The group private key of the signer can be retrieved from the signature by authorities if there is a dispute. Authorities first check the validity of the signature after they identify the group through the group ID which is included in each message, such as group t, and then compute A_{ti} as: $A_{ti} \leftarrow T_{t3}/(T_{t1}^{\xi_{t1}}\ T_{t2}^{\xi_{t2}})$, where T_{t1}, T_{t2}, T_{t3} are information included in the signature. Then the corresponding vehicle can be identified by the group ID and A_{ti}.

Compared with existing schemes which preload keys into the vehicle off-line, our key distribution framework has the following advantages [27]. (1) The revocation is more efficient. In our scheme, the revocation list is stored in RSUs. However, in preload schemes, revocation list has to be transmitted to every vehicles through wireless channels. Due to the large number of vehicles, the revocation list must be changed quickly. Meanwhile, both adding or deleting an item in the revocation list that distributes in so many vehicles is resource and time consuming. (2) The system maintenance is easier and more flexible. In our scheme, the number of vehicles that are affected by group-key updating is much smaller than that in the preload scheme. (3) Heterogeneous security policies can be implemented in our scheme. While, in preload schemes, the policy is difficult to be changed after it is deployed.

B. Secure Key Distribution Protocol Design

In this section, we propose a protocol to detect compromised RSUs and their accomplices which is a brand new security issue induced by the distributed key management framework. A misbehaved RSU will let authorities fail to identify malicious vehicles. Our protocol allows vehicles to be authenticated with their real identifiers under protection and guarantees authorities to find compromised RSUs and identities of malicious vehicles if there is a dispute. Our protocol defines message types in registration, messages broadcasting and accusation. Authorities make decisions according to the registration information that vehicles provide. Hereby, the registration procedure is the most important part.

We assume that each vehicle and RSU is preloaded with a global, long term public/private key pair with key size of 224 bits and a corresponding certificate of the public key signed by the certification authority (CA). We define the pair as *identity keys* (I-keys). The group public key and group private keys are local, short term keys in our scheme. We define them as *group keys* (G-keys). Both I-keys and G-keys are unique. Thus they are considered as identifiers of vehicles and RSUs. CA's public key size is 256 bits. Furthermore, a hash function h(x), such as SHA1, is known by authorities, RSUs and all vehicles. In this paper, elliptical curve digital signature algorithm (ECDSA) is employed as the signing protocol and we use elliptical curve integrated encryption scheme (ECIES) as the encryption protocol. Since a reliable key distribution is the foundation for the whole system, all the messages in the key distribution procedure are transmitted over the transmission control protocol (TCP).

1) Registration: The procedure of registration is shown in Fig. 2. In Table I, we list physical meanings of symbols.

Message 1: RSUs broadcast I-public keys, G-public keys of themselves and their neighbor RSUs with certificates and identities of revoked RSUs in their neighborhoods regularly. Authorities employ benign RSUs around compromised RSUs to implement revocation by regular broadcasting those compromised RSUs' identities.

Message 2: When a vehicle detects the hello message, it starts registration by sending its I-public key and the certificate to the RSU if the RSU is not revoked. Normally, a public key should not be encrypted. However, in our system model, each vehicle's I-public key is unique, so it is also an identifier of the vehicle. We encrypt it to protect vehicle's privacy.

Message 3: The RSU sends the hash value of the G-private key which plans to be assigned to the vehicle and the signature of the hash value, vehicle's I-public key and RSU's I-public key to the vehicle. RSU's I-public key is also unique. The vehicle can identify the RSU's legitimacy after it verifies this message because the RSU uses its I-private key in the message.

Message 4: The vehicle encrypts its N_{pri} and the timestamp by using authorities' public key. Then, it sends the encryption data with the timestamp and the signature of corresponding information, shown in Fig. 2 message 4, to the RSU. The encryption of its N_{pri} and the timestamp is a commitment. We will use it to detect illegitimate users later. Meanwhile, the signature signed by the vehicle binds vehicle's information and the assigned G-private key. Then, the RSU can not re-map them because the RSU does not have vehicle's I-private key.

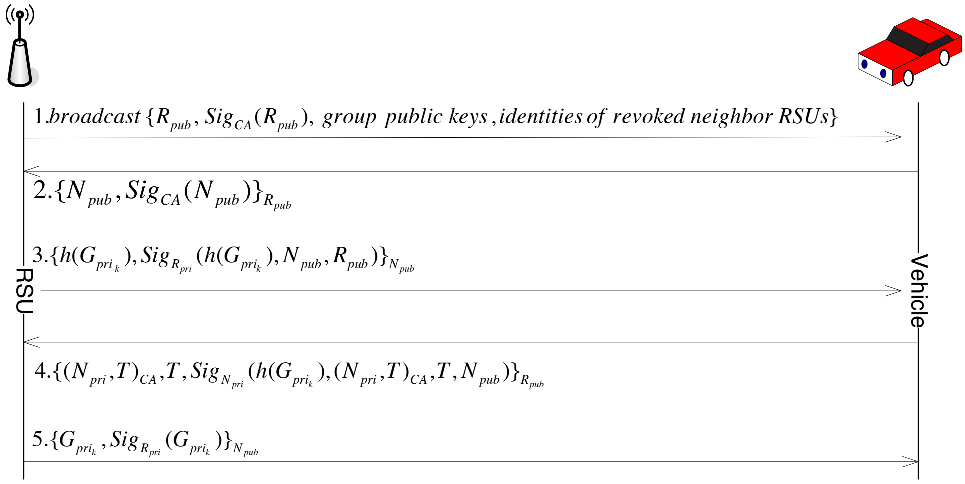

RSU
Vehicle
1.$broadcast\ \{R_{pub}, Sig_{CA}(R_{pub}),\ group\ public\ keys, identities\ of\ revoked\ neighbor\ RSUs\}$
2.$\{N_{pub}, Sig_{CA}(N_{pub})\}_{R_{pub}}$
3.$\{h(G_{pri_k}), Sig_{R_{pri}}(h(G_{pri_k}), N_{pub}, R_{pub})\}_{N_{pub}}$
4.$\{(N_{pri}, T)_{CA}, T, Sig_{N_{pri}}(h(G_{pri_k}), (N_{pri}, T)_{CA}, T, N_{pub})\}_{R_{pub}}$
5.$\{G_{pri_k}, Sig_{R_{pri}}(G_{pri_k})\}_{N_{pub}}$

Fig. 2. Registration message flow.

TABLE I
NOTATIONS AND DESCRIPTIONS

Notations	Descriptions
R_{pub}/R_{pri}	RSU's public/private key pair (I-key)
N_{pub}/N_{pri}	Node(Vehicle)'s public/private key pair (I-key)
$Sig_A(M)$	Signature of message M signed by A's private key
$(M)_k$	Message M is encrypted by k or k's public key
G_{pub_k}/G_{pri_k}	Group public/private key pair (G-key) for user k
T	Timestamp
$h(.)$	A one-way hash function such as SHA-1

Message 5: The RSU sends the G-private key to the vehicle.

The vehicle finishes registration procedure after it gets a valid G-private key. Then, the RSU stores the information, as shown in Table II, in the local database. The signature in the fifth item is the signature that the RSU receives in message 4. If authorities need the information of a vehicle when there is a dispute, the RSU has to send the vehicle's corresponding information to authorities.

Table II presents the message format and we also indicate the size of each field. When the I-keys are involved, the indicated sizes are determined by the ECDSA and ECIES algorithms and the given key size. When the G-keys are involved, the indicated sizes are determined by the short group signature scheme. Numbers in Table II are sizes of each field with unit of bytes. We allocate 4 bytes for the timestamp and 2 bytes for the group ID.

2) Messages Broadcasting: Vehicles can broadcast messages under the name of the group after they get G-private keys from the RSU. In the broadcast message format, the "Grp ID" is the group ID which is used to identify a group. We add a hash value of vehicle's I-private key and the timestamp in the message. The vehicle signs the first five items in this message using the vehicle's G-private key, resulting in the signature item. We allocate 100 bytes to the "Payload" [8].

3) Accusation: When a vehicle finds that other vehicles send false messages, it will report to authorities. For example, a vehicle may maliciously detour traffic by claiming a traffic jam at a certain place but there is not in fact. Other vehicles at that place will report such claim as a false message. The accusation message format is shown in Table II. "Grp ID" is the accuser's group identifier. The "Msg." field copies the whole message that the accusor considers false. An 8-bytes field is used to indicate "Reasons" for the accusation. "h(N_{pri},T)" is the hash value of accuser's I-private key and the timestamp. The accuser signs the first six items in this message by using its G-private key. The entire message should be encrypted by CA's public key so that the accusation messages can not be read by others.

After receiving an accusation, authorities verify the signature in the accusation message by using G_{pub}. Then, authorities perform key retrieve operations to get the accuser's and the accused's G-private keys. Whereafter, authorities contact RSUs which assign G-private keys to the accuser and the accused according to group IDs. RSUs will send corresponding information back to authorities after they receive the requests from authorities. After that, authorities will calculate accuser's and accused's h(N_{pri},T) by using vehicles' I-private keys and timestamps which are obtained from the accusation message and the broadcast message respectively. If the value that authorities calculate is the same with the value they get from the report, the user will be considered as legitimate. If both of them are authorized users, authorities will start the evaluation mechanism to decide which user tells the truth. The evaluation system design is out of the scope of this paper. A reference to this part of work is [24].

V. COOPERATIVE MESSAGE AUTHENTICATION

In this section, we propose a cooperative message authentication protocol, which augments the basic short group signature protocol by mitigating the computation overhead in the regular broadcast phase. According to [12], the verification time for short group signature is 11ms with a 3 GHz Pentium IV system. In a typical public safety application, each vehicle broadcasts safety messages every 300 ms, which implies that each vehicle can at most process messages from 27 ($\lfloor 300/11 \rfloor$) other vehicles in a stable system. However, according to the measurement that is given by [28], there may exist as many as 87 vehicles broadcasting messages within the 300m

TABLE II
MESSAGE FORMATS

Registration Record Format

G_{pri_k}	N_{pub}	$(N_{pri},T)_{CA}$	T	Signature
22	29	85	4	56

Broadcast Message Format

Grp ID	Msg. Type	Payload	$h(N_{pri},T)$	T	Signature
2	2	100	20	4	192

Accusation Message Format

Grp ID	Msg. Type	Msg.	Reasons	$h(N_{pri},T)$	T	Signature
2	2	320	8	20	4	192

Cooperative Message Format

Grp ID	Msg. type	h(N_{pri},T) of the invalid msg	T
2	2	20	4

communication range of a receiving vehicle, far exceeding its processing capability. Therefore, we propose a cooperative message authentication protocol to fill the gap between the workload and the processing capability.

A. Workflow Overview

The work flow of cooperative message authentication protocol is shown in Fig. 3. Each vehicle maintains two processes which are verifiers selection process and cooperative authentication process, a neighborhood list, a process queue and a buffer. The verifiers selection process is in charge of selecting verifiers, neighborhood list and process queue maintenance. The cooperative authentication process controls message authentication and warning message sending. In other words, verifiers selection process fills the process queue while cooperative authentication process clears it up after verifications. The neighborhood list contains neighbor vehicles' geographic information. Messages which will not be processed are stored in the buffer. When a vehicle receives a *regular broadcast message* (RBM), it extracts information of the location, speed, direction and acceleration of the sending vehicle and decides whether to verify the message or not according to geographic information. If a verifier finds an invalid RBM, it will broadcast one-hop warning information, which is termed as *cooperative authentication messages* (CAM), to inform others. A non-verifier resorts to the CAM broadcasted by other vehicles to authenticate RBM. In our protocol, each vehicle only needs to verify a very small amount of RBM.

Before discussing the details of the protocol, we would like to demonstrate two concepts. In the key distribution phase, it is designed that vehicles will report false messages to authorities when there is a dispute. The *false message* means that the content of the message is considered as wrong, but the sender's signature can be verified. For example, a vehicle may claim a traffic jam somewhere; however in fact no traffic jam happens there. The other phrase we are to use in the cooperative message authentication is invalid message. An *invalid message* is a message that can not pass the group signature verification. In such a case, even authorities can not find the signer of an invalid message. For convenience, we denote the vehicle under consideration as *tagged vehicle*.

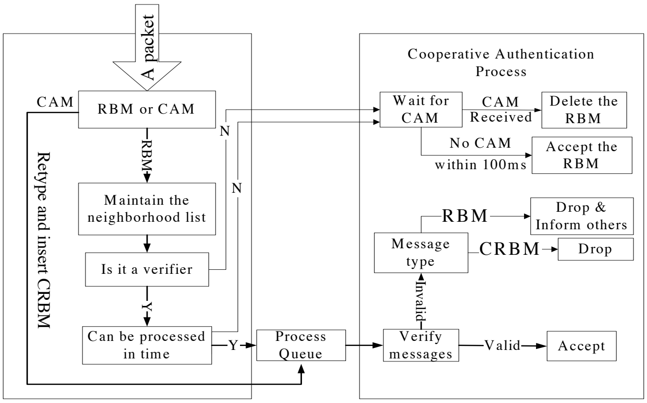


Fig. 3. Work flow of the cooperative message authentication protocol.

B. Verifiers Selection Process

The verifiers selection process starts when the tagged vehicle receives a message. If an RBM is received, the tagged vehicle updates the neighborhood list and calculates the receiver-sender distance (RSD) between itself and the sender at the sending time. After that, it tries to decide whether it is the verifier of the message by comparing its RSD with RSD of its neighbors. If the tagged vehicle is the verifier, it will insert the RBM to the process queue on the condition that it can be processed within the *verification period*, such as 100ms[1]. If the tagged vehicle is not the verifier or the verifier can not process the message in time, the received message will be put into the buffer. When a CAM is received and the corresponding RBM is found in the buffer, the tagged vehicle will change the message type of it from RBM to CRBM (CAM related RBM) as well as delete it from the buffer. Then the tagged vehicle will insert the CRBM to the process queue. A CAM without the corresponding RBM in the buffer will be dropped.

Verifiers are decided in a distributed manner by vehicles themselves according to their locations regarding to the sender. A cartesian coordinate is set up for each sender at the sending time and the location of the sender is its origin. Our verifier selection algorithm is expected to generate verifiers symmetrically and uniformly around a sender. In a *2M*-verifier scenario, the closest vehicle to the sender at each side is a verifier. Then, we draw *M*-1 arcs to find other *M*-1 verifiers at each side. The first arc has radius of 280 meters from the sender (20m for margins) and radii of the rest *M*-2 arcs are evenly distributed between 280 meters and 0 at each side. Verifiers are vehicles closest to each arc with RSD less than the radius of each arc. For example, in a six-verifier scenario, as shown in Fig. 4, vehicles nearest to the sender and the furthest ones from the sender with distances less than 280m and 140m respectively are verifiers.

Our protocol ensures that each RBM will be verified by 2*M* vehicles on average. In practice, the number of verifiers may fluctuate around 2*M* due to randomness. Our scheme is equipped with an *authentication mode switch* mechanism to ensure that the CMAP is activated only when enough vehicles and thus verifiers exist; otherwise the message-by-message protocol is activated. Details about the authentication

[1] The waiting time of a message can be estimated based on the number of messages in the process queue.

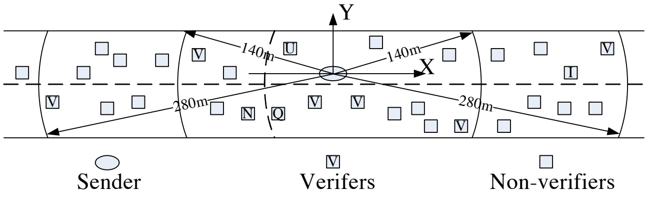


Fig. 4. Illustration of verifiers selection.

mode switch mechanism is to be discussed later. Moreover, the random density variation in a small area could lead to unbalanced verifier distribution, where a vehicle might be a verifier for many senders according to the selection process and thus be overloaded. If this overloaded vehicle is the only verifier in an area for a sender, the overloading might lead to missed detection when the sender is malicious. To avoid such a zero-verifier situation, we set a policy that a verifier will process the RBMs from the closest sender with higher priority over other RBMs; such a policy is termed as the *nearest-priority policy*. This policy can guarantee that there is at least one verifier at each side of the sender (if vehicles exist) that will definitely do the verification.

Verifiers should be as far as possible from each other. In Fig. 4, the border line of vehicle I's interference range is between vehicle Q and N, shown by the dashed arc. If we choose vehicles Q, U and the left nearest V as verifiers instead of those three V vehicles at the left side of the sender, all vehicles at the left side will receive the message except these three verifiers when the sender and vehicle I send simultaneously. Then, no one will do the verification.

The number of verifiers should be neither too small nor too large. A smaller M indicates lower computation overhead; however, some non-verifiers may not be able to receive the CAM if an RBM is invalid. While, a larger M means a higher computation overhead. The number of verifiers will be further discussed in the section VIII. For the illustration purpose, we ignore some trivial procedures in Fig. 3, such as dropping the CAM if there is no corresponding RBM in the buffer.

C. Cooperative Authentication Process

The cooperative authentication process verifies messages in the processing queue one by one. As shown in Fig. 3, if the message is valid, it will be accepted. If a CRBM is invalid, it will be dropped. An invalid RBM will be informed to others by the tagged vehicle. The CAM formate is shown in Table II. In the CAM, there is no signature to guarantee the validity of the whole message. There are several reasons. 1) The vehicle will always check the validity of the RBM by itself after they receive a CAM. Hence, the signature of CAM only wastes computing ability of the OBU. 2) A smart attacker would not attach the valid signature to the CAM if it tries to cheat. Note that messages whose lifetime exceed the verification period will be accepted if there is no CAM about it.

Missed detection means invalid RBM are considered as valid by receivers which is caused by packet loss due to limited computation capacity of verifiers or the collisions in wireless channel. Our protocol improves the performance by reducing the computation overhead of OBUs and the number of CAM that a vehicle needs to send. We will further discuss the performance in the section VIII.

D. Authentication Mode Switching

The CMAP is supposed to operate when the density of vehicle is high. In a low density scenario, message-by-message verification is always preferred for a higher level of security. Thus, in each message, one more bit, the *authentication mode* (A-Mode) flag bit should be added. When the vehicles are under the coverage of an RSU, the RSU could be a controller to initiate the authentication mode switching. However, the vehicle-initiated approach is more flexible. Based on the location information carried by each regular broadcast message, a vehicle can easily estimate the density in the area covered by its communication range. When the estimated density is above a threshold, the vehicle can set the A-Mode flag to turn on the cooperative authentication mode in the group. It is worth noting that even after the cooperative authentication mode is turned on, a vehicle with enough processing capability can still choose to operate the message-by-message verification for its own purpose, which will not impact the whole system.

VI. Security Performance Analysis

Vehicles may be attacked in both the key distribution phase and the regular broadcast phase. We discuss detailed attacks and give corresponding solutions to them in this section.

A. Key Distribution Phase

1) Appropriating the ID of other vehicles: In the accusation, the compromised RSU can launch this attack by replying other vehicle's information to authorities when it requests the registration record for a certain G-private key. Then, the user of the G-private key can not be identified.

In the registration record, each vehicle has to sign its unique I-public key, hash value of G-private key and other information by using its own I-private key. Then, the vehicle's I-public key and its assigned G-private keys are bound together. RSUs can not re-map vehicles' unique I-public keys and G-private keys arbitrarily because RSUs do not have vehicles' I-private keys.

2) Receiving key without acknowledgement: Both RSUs and vehicles can be malicious in this attack. In the key distribution procedure, RSUs have to get registration records, while vehicles need to obtain G-private keys. The one which is defined to send the information later could refuse to transmit after it gets secrets from the counterpart.

In our design, the RSU only sends the hash value of G-private key and the signature of the hash value, RSU's I-public key and vehicle's I-public key to the vehicle, as shown in Fig. 2, message 3. Then the vehicle has to submit a signature including its I-public key and the hash value of G-private key to the RSU as a part of registration record. The RSU will send the G-private key to the vehicle only after it receives this signature. We let RSUs transmit the critical information later because they are semi-trust which are more reliable. Moreover, an RSU has to get the registration record before it assigns the G-private key, so each group private key must have a corresponding registration record. It would be easy to detect

RSUs' compromise if they cannot provide a legal record for a G-private key. Those vehicles which do not get the G-private key, in case the RSU is a malicious, can join the next group.

3) Collusion Attacks: The compromised RSU and its accomplice vehicles will collude to attack. An RSU sends other vehicle's G-private key to its accomplice. Then, the malicious vehicle can broadcast messages on behalf of others.

In the registration procedure, a vehicle sends a commitment to the RSU which is the encrypted vehicle's I-private key and timestamp. Then, in every message that the vehicle broadcasts, the hash value of its I-private key should be included in it. If there is a dispute, authorities get vehicle's information from RSUs. Then, they will calculate accuser's and accused's hash values by using vehicles' I-private keys and timestamps. If values that authorities calculate are different from hash values in the accusation message, the attack can be detected. Both RSUs and malicious vehicles have no access to other vehicles' I-private keys. So, we prevent RSUs and their accomplice from attacking. On the other hand, a malicious vehicle may fill a wrong hash value into a broadcast message to frame up a normal RSU. When authorities find the mismatch, they will consider the RSU as a malicious.

Authorities can not decide which is the malicious, the RSU or the vehicle or both, when they find a mismatch. But they can be sure that, at least, there is one malicious. If authorities check the RSU physically and find that the RSU is working well, they can decide that the vehicle is a malicious one. As we discussed in the security model, RSUs are equipped with trusted platform modules. Only hardware attacks can compromise an RSU. Thus, it must be easy to check whether an RSU is compromised or not. Moreover, we assumed that attackers are rational. Malicious vehicles know that this attack will be detected by authorities, so they tend not to attack in this way.

B. Regular Broadcast Phase

1) Collusion and Sybil Attacks: If vehicles collude with each other, for example, verifiers are all accomplices of a sender, then all invalid messages that are sent by the sender will not be notified although the proportion of malicious vehicles may be not high. Or a malicious vehicle may launch a sybil attack by creating fictitious vehicles to act as its verifiers.

In our protocol, A-Mode is only implemented when the density of vehicles reaches a bottom line. Vehicles travel on the road with high velocities, so it is not easy for accomplice vehicles to get all verifiers' positions at the same time. As we discussed in the security model, attackers are minority. Hereby, it is more difficult to launch the attack when the number of verifiers increases. Another way to defend collusion attack is choosing verifiers from the other side of the road. It would be difficult for an adversary to have colluding vehicles on both directions [29]. Due to limitation of the space, we leave details of collusion attack defence as the future work. For sybil attack, some techniques can be employed to defend it. For instance, signal strength detection [30] in the physical layer can identify the real location of the sender. Rangefinders [31] which cost about 100 EURO is another way to locate vehicles.

2) Selfish Behaviors: Selfish behavior is inherent in the cooperative networks. In the regular broadcast procedure, some nodes may not verify any messages. They only wait for reports from others. Or some nodes verify messages, but they never report invalid messages to others. As we discussed in the security model, the VANETs are civilian networks that overwhelming majority of users are honest. Therefore, the proportion of selfish vehicles should be very small. The performance that is influenced by selfish vehicles can be illustrated by varying the number of verifiers.

VII. MAC-Layer Performance Analysis

In this section, we develop an analytical model for MAC-layer performance analysis of the CMAP. We consider 802.11 based VANETs, where the broadcast from each vehicle is controlled with a distributed coordination function (DCF). It is assumed that the vehicles are uniformly distributed along the road, and thus the number of vehicles in an area has a Poisson distribution [28]. Given the fixed road width, the density of vehicles along the road, denoted as β, is represented as "vehicles per kilometer" along the length direction[2]. We assume that all vehicles have the same communication range R, and the carrier sensing range equals the communication range. For mathematical traceability, the hidden-terminal effect is ignored. Our simulation results presented in Section VIII will show that the analysis inaccuracy due to the hidden-terminal effect is small, because the cooperative authentication scheme can effectively reduce the traffic load generated by each vehicle.

A. Backoff Process in Broadcast

In the DCF based broadcast, each vehicle sense the channel first before transmission. Upon sensing an idle channel, the channel access is controlled by a backoff procedure. In each backoff period, the backoff counter is initialized with a value randomly selected within a contention window W. The backoff counter reduces by 1 each slot when channel is idle and freezes when channel is busy. Transmission in an idle slot is allowed when the backoff counter reaches zero. There is no acknowledgement and retransmission in the broadcasting mode, and the backoff window size W maintains constant in each transmission period.

The backoff process can be described by a discrete-time Markov chain, with the state of the chain defined as the backoff counter value [19]. Use k to denote a possible backoff counter value, the one-step transition probabilities of the Markov chain can be expressed as

$$\begin{cases} P_{k+1,k} = 1, & k \in [0, W-2]; \\ P_{0,k} = 1/W, & k \in [0, W-1]. \end{cases} \tag{1}$$

Let π_k $(k \in [0, W-1])$ denote the steady-state probabilities of the Markov chain, it can be computed that $\pi_0 = \frac{2}{W+1}$ [19]. Let τ denote the channel access probability in an idle slot. We have $\tau = \pi_0$.

[2] The area covered by the transmission of a vehicle can be well approximated by a rectangle if the road width is much smaller than the transmission range.

B. MAC-layer Channel Behavior

We are interested in the MAC-layer channel behavior observed by a tagged vehicle. Let p_i, p_s, and p_c denote the probabilities that the tagged vehicle observes an idle channel, a successful transmission (from other vehicles), and a collision, respectively. Each vehicle can be modeled as a $G/G/1$ queue. Let p_0 denote the probability that the queue is empty; the probability that a vehicle access channel in an idle slot can then be expressed as $(1-p_0)\tau$.

Let n $(= 2\beta R)$ denote the average number of vehicles within the transmission range (equivalently the sensing range according to our assumption) of the tagged vehicle. We can have the channel idling probability regarding the tagged vehicle

$$\begin{aligned} p_i &= \frac{\sum_{i=1}^{\infty} \left[1-(1-p_0)\tau\right]^{i-1} \frac{n^i e^{-n}}{i!}}{1-e^{-n}} \\ &= \frac{e^{-n(1-p_0)\tau} - e^{-n}}{\left[1-(1-p_0)\tau\right](1-e^{-n})}. \end{aligned} \tag{2}$$

where all the other vehicles within the sensing range do not transmit. Note that the normalization factor $(1-e^{-n})$ indicates the condition that at least one vehicle (the tagged vehicle) exists in an area. The probability p_s can be obtained when there is only one vehicle other than the tagged one transmits, thus,

$$\begin{aligned} p_s &= \frac{\sum_{i=2}^{\infty}(i-1)(1-p_0)\tau\left[1-(1-p_0)\tau\right]^{i-2} \frac{n^i e^{-n}}{i!}}{1-e^{-n}} \\ &= \frac{n(1-p_0)\tau\left[e^{-n(1-p_0)\tau} - e^{-n}\right]}{\left[1-(1-p_0)\tau\right](1-e^{-n})} \\ &- \frac{(1-p_0)\tau\left[e^{-n(1-p_0)\tau} - e^{-n} - (1-(1-p_0)\tau)ne^{-n}\right]}{\left[1-(1-p_0)\tau\right]^2(1-e^{-n})}. \end{aligned} \tag{3}$$

Then, the probability of observing a collision

$$p_c = 1 - p_i - p_s. \tag{4}$$

Note that when the tagged vehicle has a packet to send in an idle slot, it is not difficult to see that the packet delivery ratio (PDR) equals the value p_i, i.e.,

$$PDR = p_i = \frac{e^{-n(1-p_0)\tau} - e^{-n}}{\left[1-(1-p_0)\tau\right](1-e^{-n})}. \tag{5}$$

C. Average Packet Service Time

The *average packet service time* is defined as the average time period from the instant that a packet becomes the head of the queue and starts to contend for transmission to the instant when that the packet is transmitted. We resort to the probability generating function (PGF) technique to derive the average packet service time.

With the CMAP, there are two types of packets, one carrying an RBM message and the other carrying a CAM message. Let λ denote the average rate of generating RBM messages in a vehicle. Use p_{mal} to denote the probability that an RBM is generated by a malicious vehicle, and V denotes the average number of verifiers for each RBM. The total average rate of generating CAM messages for verifying the RBM messages in a carrier sensing area is $Vnp_{mal}\lambda PDR$. Thus, the proportion of RBM packets over the aggregate traffic is

$$p_{_{RBM}} = \frac{1}{1+Vp_{mal}PDR}. \tag{6}$$

which is also the probability that a given packet in transmission is an RBM packet.

Let L_H denote the packet header size including both the physical layer and MAC layer header; L_{RBM} and L_{CAM} denote the average length of an RBM and CAM messages, respectively; δ denote the propagation delay; $DIFS$ denote the DCF interframe space; and C denote the wireless channel capacity. Use T_{RBM} and T_{CAM} to denote the average transmission time of an RBM and CAM packet, respectively, we have

$$T_{RBM} = \frac{L_H + L_{RBM}}{C} + DIFS + \delta. \tag{7}$$

$$T_{CAM} = \frac{L_H + L_{CAM}}{C} + DIFS + \delta. \tag{8}$$

We use T_c to denote the average duration of a collision, and approximately set $T_c = T_{RBM}$ considering that the probability that collision happens just among CAM messages is small. If we represent the transmission/collision time in terms of number of slots, the PGF of packet transmission time can be expressed as

$$S(z) = p_{_{RBM}} z^{\lfloor \frac{T_{RBM}}{\sigma} \rfloor} + (1-p_{_{RBM}}) z^{\lfloor \frac{T_{CAM}}{\sigma} \rfloor} \tag{9}$$

where σ denotes the length of a physical slot.

For a vehicle operating under the CMAP, it is not difficult to see that the PGF of the backoff counter transition time (by which the backoff counter decreases one slot) can be expressed as

$$\begin{aligned} H_d(z) &= p_i z + p_s p_{_{RBM}} z^{\lfloor \frac{T_{RBM}+\sigma}{\sigma} \rfloor} \\ &\quad + p_s(1-p_{_{RBM}}) z^{\lfloor \frac{T_{CAM}+\sigma}{\sigma} \rfloor} + p_c z^{\lfloor \frac{T_c+\sigma}{\sigma} \rfloor}. \end{aligned} \tag{10}$$

Furthermore, according to the state transition diagram of the backoff counter [19], the PGF of the average packet service time can be expressed as

$$Q(z) = \frac{S(z)}{W} \sum_{i=0}^{W-1} H_d^i(z). \tag{11}$$

Let μ denote the average service rate in terms of "packets per slot", based on the PGF $Q(z)$, the average service time can be computed as

$$\frac{1}{\mu} = Q^{'}(z) \mid_{z=1} . \tag{12}$$

In order to derive the average service time, the p_0 should be determined. We define p_0 as

$$p_0 = 1 - \frac{\lambda}{\mu}. \tag{13}$$

We can now solve the MAC-layer performance based on the equations we have obtained. Given the traffic load λ and configurations of the VANETs, the results in (2) to (10) can be incorporated with (12) to obtain one equation around parameters p_0 and μ. Such an equation can then be jointly solved with equation (13) to obtain the values of p_0 and μ.

TABLE III
PHYSICAL AND MAC LAYER PARAMETERS [19]

Parameter	Value
Preamble length	40 us
PLCP header length	8 us
Slot time σ	16 us
SIFS	32 us
DIFS	64 us
MAC header size	28 bytes
Wireless channel rate	6 Mbps
Contention window W	16

VIII. Simulation results

In this section, we use NS2 [32] simulations to examine the performance of the proposed key distribution framework and cooperative authentication protocol. We mainly consider a highway scenario with three lanes in each direction as shown in Fig. 11. Vehicles are placed uniformly on the road and travel at speed of 30 ± 5m/s (roughly equivalent to the range of 56 $\sim$ 80 miles/hour). The highway setting gives us the convenience to evaluate the lower bound of the performance, by deploying vehicles with higher speeds and higher densities to push RSUs into a high-load situation. We also simulate a typical city road scenario according to the settings in [20], where the key distribution performance is indeed much better than that under a high-load highway situation. The physical and MAC layer parameters of the 802.11 broadcast protocol used in our simulations are listed in Table III.

A. Key Distribution Performance

In the key distribution phase, it is preferred that vehicles could get their G-private keys promptly for a short service start time. Each RSU broadcasts its own public key, the associated certificate, the G-Public keys of itself and its neighbors periodically in the control channel. When an approaching vehicle receives the broadcast message, it then starts a TCP connection with the corresponding RSU to get its G-private key. RSU broadcasting and key distribution TCP connections share the same control channel. To evaluate the delay performance, we configure the computation overhead for signing, verification, encryption and decryption as that used in [33], assuming a 3GHz Pentium IV CPU.

1) Highway Scenario. Our simulation results show that most of the vehicles get their G-private keys very soon after they start the TCP connection, while some vehicles experience a delay of three or more seconds. Some other vehicles are not able to get the G-keys. The extra delay is due to the collision and the associated TCP timeout. The number of vehicles that will simultaneously start key-request TCP connections, after they hear the RSU broadcasting, is the product of vehicles density, average speed and RSUs' broadcast interval. Hence, we try to avoid collision by adjusting RSUs' broadcast interval.

For the TCP protocol, the initial round-trip time (RTT) (used as the initial timeout value) is defined as three seconds according to the RFC 2988 [34]. Thus, all the vehicles involved in the collision will experience a delay at least three seconds. A higher delay is due to further collisions in the retransmissions. We define those vehicles which get G-keys more than three seconds as *singularity vehicles*. The proportion of singularity vehicles against broadcast intervals at the density of 225 vehicles per kilometer is shown in Fig. 5(a).

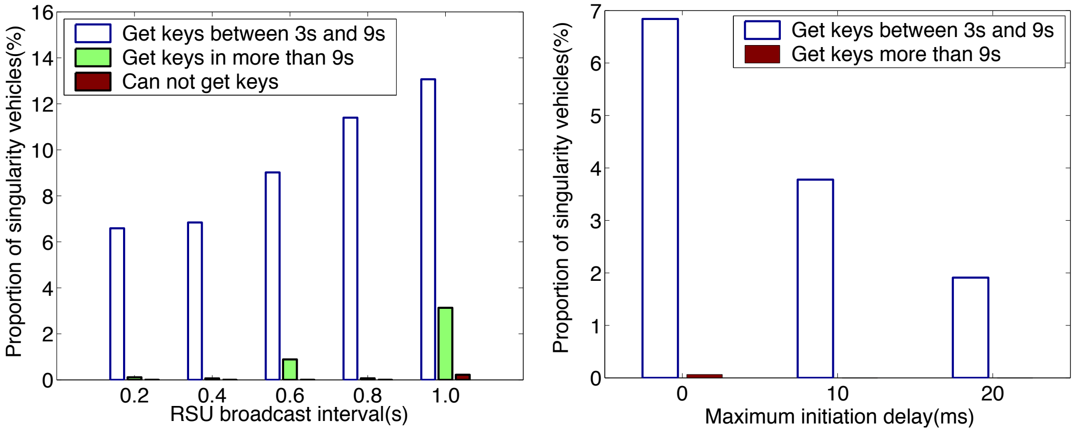
The proportion of singularity vehicles having more than 9 seconds is much less for the intervals of 0.4 second and 0.8 second than other cases. The explanation is that the TCP retransmissions in these two cases deviate from the RSU broadcast epochs for further collisions, whereas the retransmission (based on the timeout value of 3 seconds) will collide with future broadcast epochs, if the broadcast interval is 0.2, 0.6, or 1.0 second. Hence, we set the RSU broadcast interval as 0.4 second in our implementation.

In order to reduce the collisions due to the simultaneous key request, we introduce a *random initiation scheme*. After a vehicle receives the RSU broadcast message, instead of starting key request immediately, it will send the request after a random initiation delay. We use W_I to denote the maximum initiation delay, and each vehicle randomly pick its initiation delay from $(0, W_I)$. The proportion of singularity vehicles against the maximum initiation delay in the highway scenario is shown in Fig. 5(b). From the figure, we can see that when $W_I = 20$ ms, only two percent of vehicles fail in the first key request and incur retransmissions. In this scenario, our simulations further show that all vehicles have a key distribution delay less than four seconds, giving a satisfying service start time.

2) City Road Scenario. To show that the high-load highway scenario indeed gives a lower bound of the performance, we also simulate a typical city road scenario as shown in Fig. 6(a). We follow the configurations used in [20] with a density of 150 vehicles per square kilometers and travel speeds in the range of 15 ± 5m/s (roughly equivalent to the range of 22.5 $\sim$ 45 miles/hour). When a vehicle reaches an intersection, it will randomly choose to turn left, turn right or go forward. A vehicle hitting the boundary will be bounced back into the map to maintain a steady density of vehicles in the map. It is not difficult to check that, in the city road scenario, the average number of vehicles in the area covered by an RSU is much less than that in the highway scenario considered in Fig. 5. Comparing the results in Fig. 6(b) to those in Fig. 5(a), we can see the proportion of singularity vehicles is much smaller in the city road scenario.

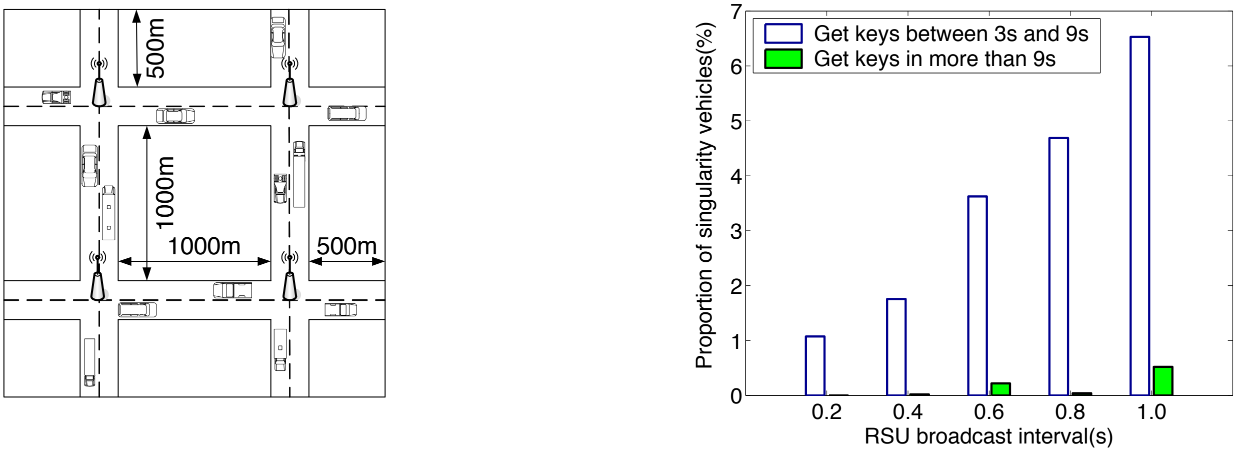
B. Cooperative Authentication Performance

In this part, we evaluate the performance in the regular broadcast phase by simulating packet delivery ratio, computation and communication overheads and missed detection ratio. We also compare both the theoretical and simulation results under our protocol with those under the protocol in [13]. Since the cooperative authentication protocol is of particular importance in the high-load scenario, we thus only focus on the highway scenario in this part. We assume six percent of the vehicles are malicious in our simulations. Malicious vehicles always send invalid RBM, and they never send CAM to help others. The *missed detection ratio* is defined as the percentage



(a) (b)

Fig. 5. Key distribution performance in the highway scenario. (a) Performance versus the RSU broadcast intervals. (b) Performance versus the initiation delay.



(a) (b)

Fig. 6. Key distribution performance in the city road scenario. (a) Road map. (b) Performance versus the RSU broadcast intervals.

of invalid RBM that are considered as valid by a receiver. The missed detection ratio is computed based on well behaved vehicles in our simulation. Considering that the performance of the highway scenario is more severe than that of the local scenario, we focus on highway scenario in this part and leave the local scenario case to the future work.

1) Number of Verifiers: As discussed in the section V, the number of verifiers is a tradeoff between missed detection ratio and computation overhead of OBUs. The missed detection ratios versus different number of verifiers are shown in Fig. 7. It can be seen the performance under 8 verifiers is obviously better than that under 6 ones. Nevertheless, the number of verifiers could not be too large. If the number is large enough to ensure a good CAM for an RBM, the extra number of verifiers will lead to negative impact by incurring unnecessary communication and computation overhead. Our simulation results suggest that 8 verifiers can achieve a good tradeoff.

We would like to emphasize that our nearest-priority policy in cooperative authentication guarantees that every sender has at least one verifier at each side to do the verification. Thus, the missed detection is mainly due to packet losses caused by MAC layer collisions. To demonstrate the impact of MAC collisions, we also evaluate the scenario that vehicles may take different average speeds, and the missed detection ratio in such a scenario is presented in Fig. 7 too. While the heterogeneous average speeds tend to results in an uneven distribution of vehicles and a higher probability of overloaded verifiers, the missed detection ratio in this situation is in fact smaller. The reason is that the speed difference will stretch the area of vehicle distributions, and equivalently reduce the density of vehicles and the frequency of broadcast messages in an area. The reduced traffic load will then result in less MAC collisions and thus smaller missed detection ratio.

2) Packet Delivery Ratio: The packet delivery ratio is defined as the proportion of transmissions that can be successfully received. The PDR is a critical performance measure affecting both the network utilization and security performance. A low PDR (or a high packet loss ratio due to collision) means a low bandwidth utilization, and the loss of CAM tends to result in missed detection. In [13], the authors present a probabilistic verification protocol, in which a vehicle receiving an RBM decides to be a verifier with a probability. However, in order to guarantee that there are verifiers selected at both sides of the sender, on average 25 verifiers should be randomly

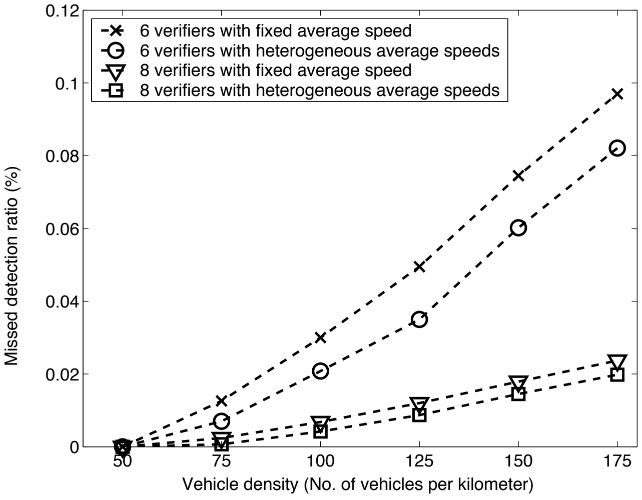


Fig. 7. Missed detection ratio versus the number of verifiers.

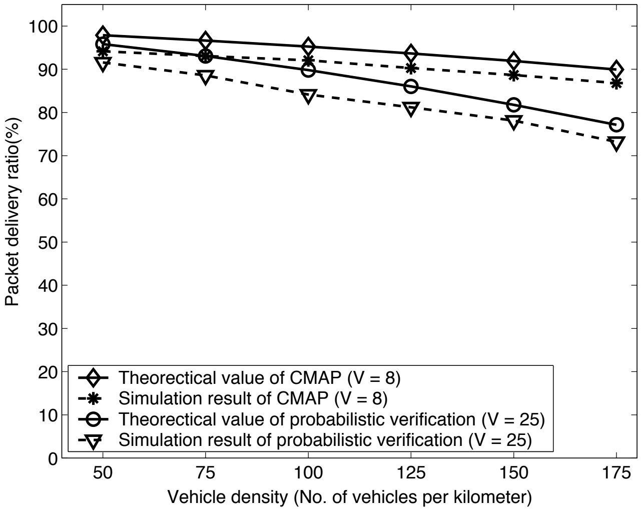


Fig. 8. Packet delivery ratio versus the density of vehicles.

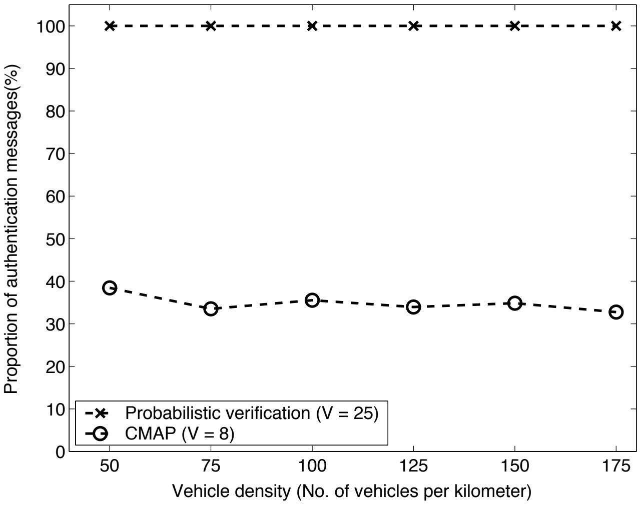


Fig. 9. Communication overhead due to cooperative authentication messages.

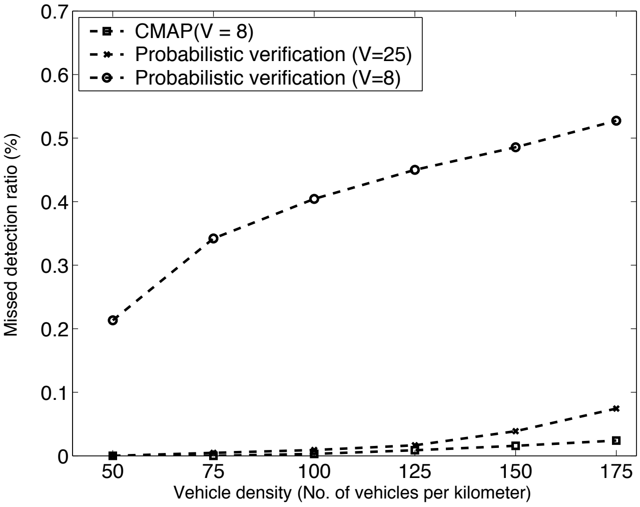


Fig. 10. Missed detection ratio versus the density of vehicles.

incurred for each RBM according to the protocol. Another difference between our CMAP and the protocol in [13] is that CMAP allows a much shorter CAM.

We show the theoretical values and simulation results of PDR for CMAP and probabilistic verification protocol in Fig. 8. The theoretical PDR is computed by (5). We can see that the theoretical values are close to simulation results in both scenarios. Note that the MAC-layer analytical model developed in Section VII can also be applied to the probabilistic verification protocol with a good accuracy. The theoretical values are anyhow lightly higher than the simulation results; it is because that the analysis is optimistic by ignoring the hidden-terminal effect and result in a higher PDR. The PDR under CMAP is higher than that under the protocol in [13]; it is because the smaller number of verifiers and shorter CAM in CMAP gives a smaller traffic load, which thus results in a smaller collision probability and a higher PDR. The higher PDR under CMAP will lead to a better network utilization and security performance.

3) Communication Overhead: The communication overhead of CMAP is explicitly compared with the probabilistic verification protocol in Fig. 9, which shows the proportion of cooperative authentication messages over the total traffic, considered as communication overhead. For the comparison purpose, we normalize the communication overhead under the CMAP against that under the probabilistic protocol. It is clearly shown that CMAP has a communication overhead less than 40% of that under the probabilistic protocol.

4) Missed Detection Ratio and Computation Overhead: Fig. 10 compares the CMAP with the probabilistic verification protocol in terms of missed detection ratio. We can see that with the same number verifiers $V = 8$, the performance of probabilistic verification protocol deteriorates significantly, because $V = 8$ can not ensure with high probability that verifiers exist on both sides of a sender. The good performance of CMAP is because the pattern of selecting verifiers is fixed according to position information.

Another interesting observation is that in the cases of high density, the performance of CMAP is still better than the probabilistic protocol even when it uses 25 verifiers. The reason is due to the hidden-terminal effect as shown in Fig. 11. In the scenario, the hidden terminals at both sides of a sender

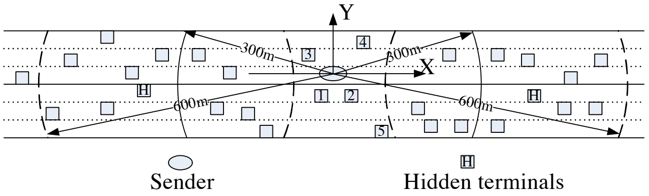


Fig. 11. Impact of hidden-terminals on cooperative authentication.

will result in that most of the vehicles around the sender can not receive the broadcast message. Only a small number of vehicles close to the sender may receive the message, but the small number of survivors may not generate any verifiers according to the pre-configured verifier selection probability. Nevertheless, CMAP always requires the two vehicles on both sides of and closest to the sender to be verifiers; thus in the scenario shown in Fig. 11, CMAP still performs well while the probabilistic protocol leads to missed detection.

We also evaluate the computation overhead through simulations, based on the configuration suggested in [12]. We define the CPU usage as the average proportion of the time that vehicles spend on verification. Our simulation results show that the CPU usage under CMAP never reaches 50% while that under probabilistic verification is always more than 90%. Since the number of verifiers directly determines the computation overhead, Fig. 10 also implies that if CMAP uses the same CPU resource (i.e., the same number of verifiers) as that used by the probabilistic verification protocol, CMAP achieves much better performance in missed detection ratio.

IX. Conclusions

In this paper, we propose a novel distributed key management scheme based on the short group signature to provision privacy in the VANETs. The distributed key management is further enhanced with a cooperative message authentication protocol to alleviate the heavy computation overhead. We investigate the challenging issue that semi-trust RSUs may be compromised, and compromised RSUs may even collude with malicious vehicles. We design a security protocol to prevent compromised RSUs and malicious vehicles from attacking. Our design guarantees that RSUs distribute keys fairly and provide some mechanisms to detect compromised RSUs and malicious vehicles. Moreover, by a cooperative message authentication protocol, a vehicle only needs to verify a small amount of messages, and the computation burden of vehicles is reduced greatly. We give detailed analysis of possible security attacks and the corresponding defence, as well as develop a MAC layer analytical model. Extensive NS2 simulations are also presented to evaluate the performance of the proposed techniques.

References

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

[2] R. Lu, X. Lin and X. Shen, "SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks", in *Proc. IEEE INFOCOM*, San Diego, California, 2010.

[3] J. Freudiger, M. Raya, M. Feleghhazi,P. Papadimitratos and J.-P.Hubaux., "Mix zones for location privacy in vehicular networks," in *Proc. International Workshop on Wireless Networking for Intelligent Transportation Systems*, Vancouver, British Columbia, Aug., 2007.

[4] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki, "Enhancing wireless location privacy using silent period," in *Proc. IEEE WCNC*, pp. 1187-1192, 2005.

[5] K.Sampigethava, L.Huang, M.Li, R.Poovendran, K.Matsuura and K.Sezaki, "AMOEBA: Robust location privacy scheme for VANET," in *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp.1569-1589, 2007.

[6] D. Chaum and E. van Heyst, "Group signatures," in *Proc. Advances in Cryptology - Eurocrypt*, vol. 547, pp. 257-265, 1991.

[7] J. Guo, J.-P. Baugh and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[8] X. Lin, X. Sun, P.-H. Ho and X. Shen, "GSIS: a secure and privacy preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442-3456, 2007.

[9] G. Calandriello, P. Papadimitratos, A. Lloy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in VANET," in *Proc. ACM Mobicom*, pp. 19-28, QC, Canada, Sept. 2007.

[10] IBM 4764 PCI-X Cryptographic Coprocessor. http://www-03.ibm.com/security/cryptocards/pcixcc/order4764.shtml.

[11] N. Banerjee, M.D. Corner, D. Towsley and B.N. Levine, "Relays, base station and meshes: enhancing mobile networks with infrastructure," in *Proc. ACM Mobicom*, San francisco, California, Sep. 2008.

[12] X. Sun, "Anonymous, secure and efficient vehicular communications," Master Thesis, Univeristy of Waterloo, 2007.

[13] C. Zhang, X. Lin, R. Lu, P.-H. Ho and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3357-3368, 2008.

[14] C. Zhang, X. Lin, R. Lu and P.-H. Ho., "RAISE: an efficient RSU-aided message authentication scheme in vehicular communication networks," in *Proc. IEEE ICC*, Beijing, China, May 19-23, 2008.

[15] X. Lin, C. Zhang, X. Sun, P.-H. Ho and X. Shen, "Performance enhancement for secure vehicular communications," in *Proc. IEEE Global Telecommunications Conference*, pp.480-485, Washington DC, Nov. 2007.

[16] A. Wasef and X. Shen, "ASIC: aggregate signatures and certificates verification scheme for vehicular networks", in *Proc. IEEE Globecom*, Honolulu, Hawaii, USA, Nov. 30 - Dec. 4, 2009.

[17] A. Studer, E. Shi, F. Bai and A. Perrig, "TACKing together efficient authentication revocation, and privacy in VANETs," in *Proc. IEEE SECON, 2009.*

[18] D. Jiang and L. Delgrossi, "IEEE 802.11p: towards an international standard for wireless access in vehicular environments," in *Proc. IEEE VTC*, May 2008.

[19] X. Ma, X. Chen and H. Refai, "Unsaturated performance of IEEE 802.11 broadcast service in vehicle-to-vehicle networks," in *Proc. IEEE VTC*, Oct., 2007.

[20] G. Marfia, G. Pau, E. De Sena, E. Giordano and M. Gerla, "Evaluating vehicle network strategies for downtown Portland: opportunistic infrastructure and the importance of realistic mobility models," in *International MobiSys Workshop on Mobile Opportunistic Networking*, San Juan, 2007.

[21] P. Enge, "Retooling the global positioning system," *Scientific American*, May 2004.

[22] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," *Workshop on Security in Ad hoc and Sensor Networks*, 2005.

[23] P. Golle, D. Greene and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. ACM VANET*, Philadelphia, 2004.

[24] M. Raya, P. Papadimitratos, V.-D. Gligor and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. IEEE INFOCOM*, pp. 1238-1246, Apr. 2008.

[25] D. Boneh, X. Boyen and H. Shamcham, "Short group signatures," in *Proc. Advances in Cryptography - Crypto' 04, ser. LNCS*, vol.3152, Springer-Verlag, pp. 41-55, 2004.

[26] X. Sun, X. Lin and P.-H. Ho, "Secure vehicular communications based on group signature and ID-based signature scheme," in *Proc. IEEE ICC*, Scotland, Jun., 2007.

[27] Y. Hao, Y. Cheng and K. Ren, "Distributed key management with protection against RSU compromise in group signature based VANETs," in *Proc. IEEE Globecom*, New Orleans, Nov., 2008.

[28] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp.1538-1556, 2007.

[29] S. Park and C.C.Zou, "Reliable traffic information propagation in vehicular ad-hoc networks," *IEEE Sarnoff Symposium*, Apr. 2008.

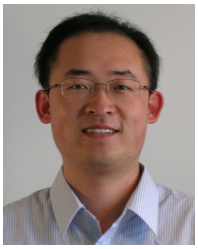
[30] B. Xiao, B. Yu and C. Gao, "Detection and localization of sybil nodes in VANETs," in *Proc. ACM/SIGMOBILE Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, 2006.

[31] K. Ibrahim, M. C. Weigle and G. Yan, "Light-weight laser-aided position verification for CASCADE," in *Proc. International Conference on WAVE*, Dearborn, MI, Dec. 2008.

[32] The network simulator-NS2, http://www.isi.edu/nsnam/ns/tutorial/index.html/.

[33] Shamus Software. MIRACL library, http://www.shamus.ie/index.php?page=Elliptic-Curve-point-multiplication.

[34] V. Paxson and M. Allman, "Computing TCP's Retransmission Timer", IETF RFC 2988.



Yong Hao (S'10) received the B.E. and M.E. degrees in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2003 and 2007 respectively. He is currently pursuing the Ph.D degree in the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, U.S.A. His current research interests include wireless network security and vehicular ad hoc networks.

Yu Cheng (S'01-M'04-SM'09) received the B.E. and M.E. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada, in 2003. From September 2004 to July 2006, he was a postdoctoral research fellow in the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA, as an Assistant Professor. His research interests include next-generation Internet architecture and management, wireless network performance analysis, network security, and wireless/wireline interworking. He received a Postdoctoral Fellowship Award from the Natural Sciences and Engineering Research Council of Canada (NSERC) in 2004, and a Best Paper Award from the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine'07), Vancouver, British Columbia, August, 2007. He served as a Technical Program Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009. He is an Associated Editor for IEEE Transactions on Vehicular Technology and an Area Editor for Elsevier Journal of Computer Networks.

Chi Zhou (SM'09) received two B.S. degrees in both Automation and Business Administration from Tsinghua University, China, in 1997.She received the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Northwestern University in 2000 and 2002, respectively. Between 2002 and 2006, she worked in Florida International University as assistant professor. Since 2006, she has served as an Assistant Professor in the Department of Electrical and Computer Engineering, Illinois Institute of Technology. Her primary research interests include wireless sensor networks for smart grid application, scheduling for OFDMA/MIMO systems, network coding for wireless mesh networks, integration of optical and wireless networks, and security for VANETs.

Wei Song received her Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2007. Since 2008, she has been supported by the Natural Science and Engineering Research Council (NSERC) of Canada and worked as a postdoctoral research fellow at the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. In July 2009, she joined the Faculty of Computer Science, University of New Brunswick, as an assistant professor. She received a Top 10% Award from IEEE Workshop on Multimedia Signal Processing (MMSP) 2009, an NSERC postdoctoral fellowship in 2008, and a Best Paper Award from IEEE WCNC 2007. Her current research interests include the interworking of cellular networks and wireless local area networks (WLANs), resource allocation for heterogeneous wireless networks, vehicular ad hoc networks, and cross-layer optimization for multimedia quality-of-service (QoS) provisioning.