International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 8, Issue 10 (October 2013), PP. 32-44

Decentralized Trust Based Routing Scheme for Vehicular Ad Hoc Networks

Deepika.L[1], Dr.VellingiriJ[2]
(M.E-CSE), Kongunadu College of Engineering and Technology
(Professor), Kongunadu College of Engineering and Technology

Abstract:- The security of vehicular ad hoc networks (VANETs) has been receiving a significant amount of attention in the field of wireless mobile networking because VANETs are vulnerable to malicious attacks.Asymmetric cryptography schemes are not suitable for highly dynamic environments. Cryptanalysis is not verified and protocol is not secure.It adopts transitive trust relationships to improve the performance of the authentication procedure and only needs a few storage spaces.To propose a decentralized trust authentication mechanism for cryptanalysis scheme, inside attacks, secure routing protocol. Intend to develop an intrusion detection mechanism to enhance the network security.

Index Terms:- Authentication, decentralized, trust-extended, vehicular ad hoc networks (VANETs).

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) have been attracted increasing attention from both industry and academia [1]. The major components of a VANET are the wireless on-board unit (OBU), the roadside unit (RSU), and the authentication server (AS). OBUs are installed in vehicles to provide wireless communication capability, while RSUs are deployed on intersections or hotspots as an infras-tructure to provide information or access to the Internet for vehicles within their radio coverage. The AS is responsible for installing the secure parameters in the OBU to authenticate the user. Based on IEEE 802.11p, the dedicated short range communication system [2] supports two kinds of communication environments: vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications.

A number of studies [3]–[5] have focused on the problem of data dissemination in VANETs. However, these schemes do not consider the security problem. Recently, the security issue in VANETs has become a hot topic, and then many researchers provide the V2I and V2V authentication mechanisms to protect valid users. However, the design for an efficient V2V authentication mechanism is more challenge than that for V2I authentication mechanism in VANETs because the vehicle cannot be authenticated via the infrastructure directly in V2V communications. Therefore, we focus on V2V network environments and propose an efficient authentication scheme in this paper.

To address the above need, we propose a decentralized authentication scheme, called TEAM, for V2V communication networks. There exists no centralized authority to perform the authentication procedures of vehicles. TEAM is a lightweight authentication scheme because it only uses an XOR operation and a hash function. Although TEAM needs low computation cost, it still satisfies the following security requirements: anonymity, location privacy, mutual authentication, resistance to stolen-verified attacks, forgery attacks, modification at-tacks and replay attacks, as well as no clock synchronization problem, fast error detection, perfect forward secrecy, man-in-the-middle attack resistance, and session key agreement. Moreover, our scheme only requires a few storage spaces than other schemes because the vehicle does not need to store the authentication information (e.g., public key) of the entire vehicle.

The preliminary version of this paper was published in IEEE CECNET 2011 [10]. In this paper, we describe the proposed scheme in detail. We add the adversary model discussion, secure communication, password change, key update, and key revocation procedures in this enhanced version. Moreover, we propose the analysis of computational and storage costs of TEAM, and then we use the NS-2 network simulator to evaluate the performance of TEAM.

The remainder of this paper is organized as follows. Section II contains a review of related work. In Section III, we introduce some preliminaries, and in Section IV, we describe the proposed scheme in detail. Analyses of the security and performance are presented in Section V. Then, in Section VI, we summarize our conclusions and consider future research avenues.

II. RELATED WORK

Raya and Hubaux [6] preloaded each vehicle with a large number of anonymous public and private key pairs, as well as the corresponding public key certificates. Each of the public key certificates contains a pseudoidentity. Then, traffic messages are signed with a public key-based scheme, and each pair of public and private key has a short lifetime to preserve its privacy. However, this approach works with high computation cost, high storage cost, and high communication overhead. Freudiger *et al*. [7] used the cryptographic MIX-zone to enhance the location privacy, and Sampigethava *et al*.

[8] provided location privacy by utilizing the group navigation of vehicles. However, these approaches [6]–[8] do not work well in highly dynamic environments like VANETs because they use asymmetric cryptography or a digital signature veri-fication scheme, which results in high computation costs, long authentication latency, and a large storage space. Zhang *et al*.

[9] proposed an RSU-aided messages authentication scheme (RAISE), which uses the symmetric key hash message au-thentication code, instead of a public key infrastructure-based message signature, to reduce the signature cost. However, in RAISE, the key agreement process still executes the exponent operations, which leads to a high computation cost. Moreover, the RSU needs to maintain the extra ID-Key table, resulting in more storage cost. Hence, there is still a need for an efficient authentication scheme for VANETs with low computation and low storage costs.

III. PRELIMINARIES

In this section, we introduce the concept of the transitive trust relationships, describe some threat models, and consider the security requirements of VANETs.

A. Transitive Trust Relationships

In VANETs, vehicles can be classified into to the following roles: a law executor (LE), a mistrustful vehicle (MV), and a trustful vehicle (TV) as illustrated in Fig. 1. An LE, such as police car or authorized public transportation (e.g., buses), acts like a mobile AS. Moreover, the LE is trustful permanently. A normal vehicle is regarded as trustful if it can be authenticated successfully; otherwise, it is deemed to be mistrustful. In addition, the TV becomes the MV when the key lifetime is over. To provide a secure communication environment, the OBU should be authenticated successfully before it can access the service. However, in V2V communication networks, as the number of LEs is finite, an LE is not always in the vicinity of the OBU. Even if the user is well meaning, the vehicle must still wait for the nearest LE and then perform the authentication procedure. Hence, there is an urgent need for an efficient authentication scheme. In this paper, we propose a TEAM to improve the performance of the authentication procedure in V2V communication networks. The TEAM is based on the concept of transitive trust relationships, as illustrated in Fig. 2. Initially, there are three vehicles in a VANET: a trustful LE and two other MVs carrying OBUs (i.e., OBU_i and OBU_j in Fig. 2). The state of the first mistrustful OBU (i.e., OBU_i) becomes trustful and obtains the sufficient authorized parameter to authorize other mistrustful OBUs when it is authenticated successfully. Then, it plays the LE role temporarily to assist with the authentication procedure of OBU_j .

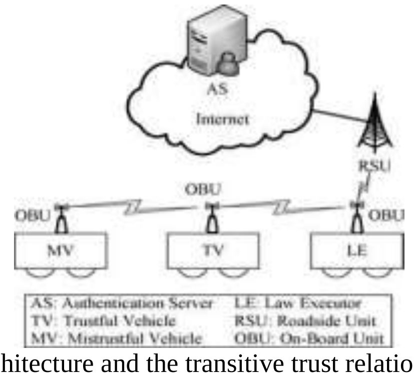


Fig. 1. Network architecture and the transitive trust relationships of VANETs.

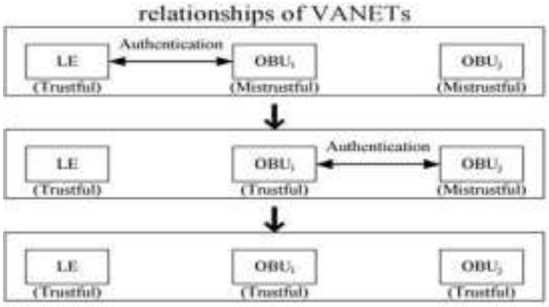


Fig. 2. Transitive trust relationships in a TEAM.

Thus, the other mistrustful OBUs can be authenticated by any trustful OBU without necessarily finding an LE, and all vehicles in a VANET can complete the authentication proce-dure quickly. Therefore, the key design issues of the authenti-cation procedure based on the transitive trust relationships are: 1) how to let the TV own the authentication ability; 2) how to reduce the computational cost; 3) how to prolong the trustful state of the TV; and 4) how to use as little storage cost as possible.

B. Adversary Model

The following possible attack models can be used during the V2V authentication procedure.

1) *Modification attack:* The adversary modifies the packet resulting in the message against the integrity of the information.

2) *Message replay attack:* The adversary resends valid messages sent previously in order to disturb the traffic flow.

3) *Movement tracking:* Since wireless communication is based on a shared medium, an adversary can easily eavesdrop on any traffic. After intercepting a significant number of messages in a certain region, the adversary could trace the physical position and movement patterns of a vehicle by simply analyzing the information.

4) *Impersonation attack:* The adversary pretends to be a valid LE/TV to cheat the unauthenticated OBUs.

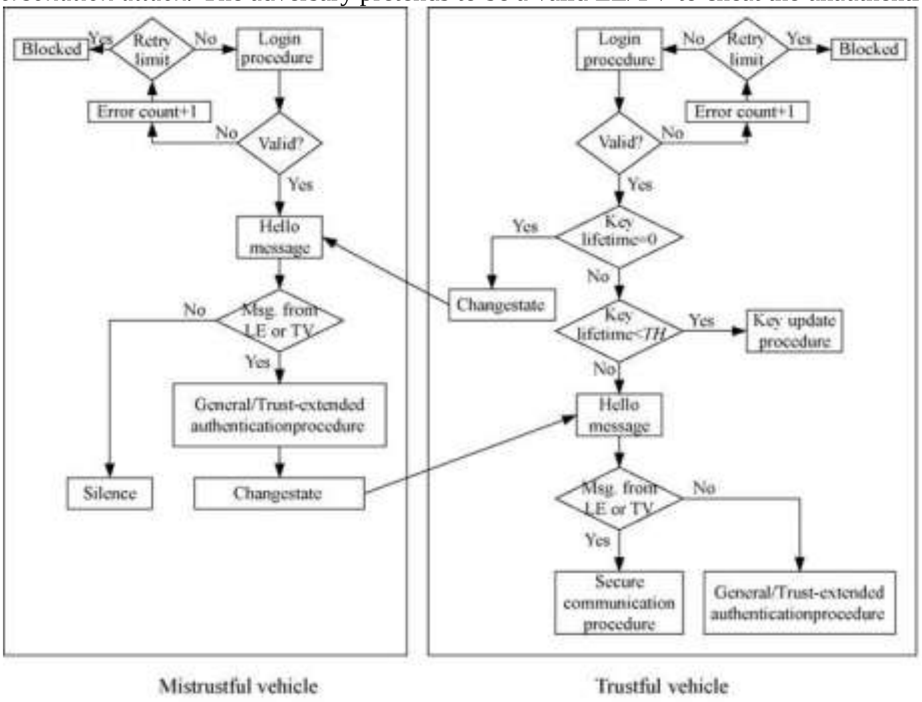


Fig. 3. Operations of the mistrustful/trustful vehicle in a TEAM.

C. Security Requirements

Since the authentication scheme is susceptible to malicious attacks, our objective is to design a scheme that is robust to such attacks. Based on related studies [6]–[14], we define the following key security requirements for VANETs.

1) *Efficiency:* In VANETs, the computational cost of ve-hicles must be as low as possible in order to have a real-time response.

2) *Anonymity:* The anonymous authentication procedure verifies that an OBU does not use its real identity to execute the authentication procedure.

3) *Location privacy:* An adversary collects the serial au-thentication messages of the OBU but it still failed to track the location of the vehicle.

4) *Mutual authentication:* A mutual authentication proce-dure is implemented whereby the LE must verify that the OBU is a legal user and the OBU must ensure that the LE is genuine.

5) *Integrity:* The message integrity means that data cannot be modified undetectably.

IV. TEAM

In this section, we describe the proposed scheme in de-tail. A TEAM is a decentralized authentication scheme, and the LEs need not to keep the authentication information of the entire vehicles. The proposed scheme involves eight procedures: initial registration, login, general authentication, password change, trust-extended authentication, key update, key revocation, and secure communication. Before a vehicle can join a VANET, its OBU must register with the AS. When a vehicle wants to access the service, it has to perform the login procedure. Next, the OBU checks the authentication state itself (i.e., the lifetime of the key). If the lifetime of the key is reduced to zero, the vehicle is mistrustful, and vice versa. The MV performs the general or trust-extended authentication procedure to be authenticated. The trustful vehicles assist other MVs in performing the authentication procedure or commu-nicate with other trustful vehicles (i.e., secure communication procedure) to access the Internet. The trustful vehicle performs the key update procedure with the LE when the key lifetime is below the predefined threshold. Moreover, we also consider the password change procedure for user friendly. Fig. 3 shows the operations of the mistrustful/trustful vehicle in TEAM. The state of the LE does not change because the LE is always trustful.

A. Assumptions

Many related works point out that the system of vehicle is better protected than the general mobile device (e.g., PDA, smartphone, etc.). Therefore, we assume that each vehicle's OBU is equipped with security hardware (e.g., trusted plat-form module), including an event data recorder (EDR), and a tamper-proof device (TPD) [15]–[17] so that an attacker cannot obtain information about the vehicle from the OBU. The EDR is responsible for recording important data about the vehicle, such as the location, time, preload secret key, and access log. The TPD provides the cryptographic processing capabilities. Finally, we assume that the time of every vehicle is synchronous via GPS device.

B. Notations

Before describing the proposed scheme, the notations used throughout this paper are listed in Table I.

TABLE I
Notations

Symbol	Description
x	A secret key protected by the AS
ID_i	The public identification of entity i
AID_i	The alias of entity i
PW_i	The password of user i
$^{SK}i{-}j$	A session key between entity i and entity j, where SK_{i-j} =SK_{j-i}
MSG_{KU}	A key update message
X→Y	User X sends a message to user Y through a secure channel
X→Y	User X sends a message to user Y through a common channel
$h(\,)$	A collision-free one-way hash function
^{N}i	A nonce or random number i
PSK	A secure key set that is preshared among LEs and the AS
$\oplus$	The XOR operator
||	The combination of strings

C. Periodic Hello Message

In VANETs, the vehicles broadcast the hello message peri-odically with the authentication state (i.e., trust or mistrust). In order to ensure the network security, only the trustful vehicle can execute the secure communication procedure (i.e., Section IV-I). On the contrary, the MV must finish the authentication procedure (i.e., Sections IV-E and IV-F) in advance to com-municate with other vehicles.

D. Initial Registration Procedure

1) *LE Registration:* First, the LE performs the LE reg-istration procedure with the AS through the manufacturer or a secure channel. The AS computes the secure key set $\{PSK_i, i = 1, \ldots, n\}$ based on the hash-chain method (e.g., $h^2(x) = h(h(x))$) and sends this key set to the LE. Note that the LE only needs to hold a secure key set that is stored in the security hardware and it does not need to store any authentication information of the user. Moreover, each PSK_i has a short lifetime for robust security. Therefore, each trustful vehicle performs the key update procedure with the LE (i.e., Section IV-K) when the key lifetime is going to end. Fig. 4 shows the key set generation scheme. We can see that the new PSK (e.g., PSK_2) cannot be inferred from the old PSK (e.g., PSK_1) since the key generation scheme has a one-way feature of the hash function.

2) *Normal Vehicle Registration:* Other vehicles need to perform the normal vehicle registration procedure with

the AS through the manufacturer or a secure behavior when the vehicle left the car factory. This initial registration procedure is only performed once. Fig. 5 describes the steps of the normal vehicle registration procedure.

Step 1) User$_i$ → AS: A user sends the public identification ID$_i$ and his chosen password PW$_i$ to the AS via the manufacturer or a secure channel.

Step 2) After receiving the user's ID and password, the AS computes the following secret authentication param-eters for the user: $A_i = h(\mathrm{ID}_i||x)$, $B_i = h^2(\mathrm{ID}_i||x) = h(A_i)$, $C_i = h(\mathrm{PW}_i) \oplus B_i$, and $D_i = \mathrm{PSK} \oplus A_i$. The objective of A_i is to build the relation between the user's ID and AS. Moreover, the objective of C_i is to build the relation among user's password, user's

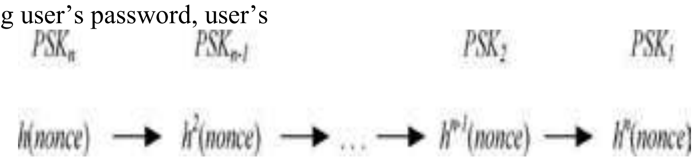


Fig. 4. Key set generation scheme based on the hash-chain method.

method

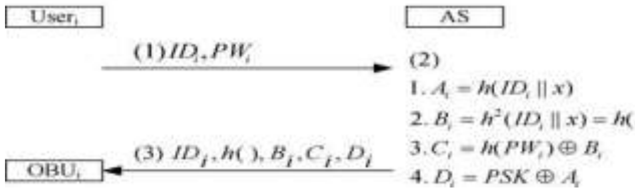

User$_i$
AS
(1) ID_i, PW_i
(2)
1. $A_i = h(ID_i \| x)$
2. $B_i = h^2(ID_i \| x) = h(A_i)$
3. $C_i = h(PW_i) \oplus B_i$
4. $D_i = PSK \oplus A_i$
(3) $ID_i, h(\,), B_i, C_i, D_i$
OBU$_i$

Fig. 5. Normal vehicle registration procedure.

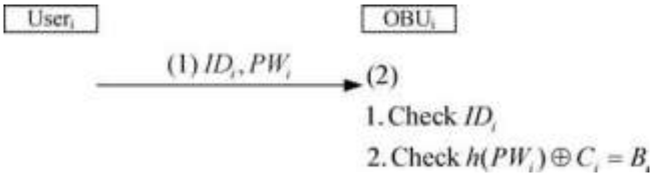


Fig. 6. Login procedure.

ID, and AS. Therefore, the user only keys in the correct personal information (i.e., ID$_i$ and PW$_i$) in the login procedure. Otherwise, the OBU$_i$ rejects this login request.

Step 3) AS→ User$_i$: The AS stores the parameters (i.e., ID$_i$, B_i, C_i, D_i,$h(\ )$) in the OBU's security hardware via the manufacturer or a secure channel.

Note that the AS does not need to store the user's verifi-cation information (e.g., the user's password). Therefore, an adversary cannot obtain the information to launch a stolen-verified attack.

In addition, the registered user cannot impersonate to an-other valid user successfully when the user obtains the above parameters. This is because the user does not know the AS's secret (i.e., x).

E. Login Procedure

The login procedure is the first checkpoint. The OBU will detect an error event immediately if the user has malicious intentions. Fig. 6 shows the steps of the login procedure. Step 1) User$_i$→OBU$_i$: When a user wants to access the service, he/she inputs ID$_i$ and PW_i to the OBU$_i$.

Step 2) The OBU$_i$ checks the ID$_i$ and verifies whether $h(\mathrm{PW}_i) \oplus C_i$ is equal to B_i, where B_i and C_i are obtained from the initial registration procedure. If the information is correct, the OBU$_i$ performs the general authentication procedure. Note that $h(\mathrm{PW}_i) \oplus C_i$ has to be equal to B_i. If the values are not equal, it means that the user inputs the wrong ID$_i$ or PW$_i$, resulting in the login request will be rejected.

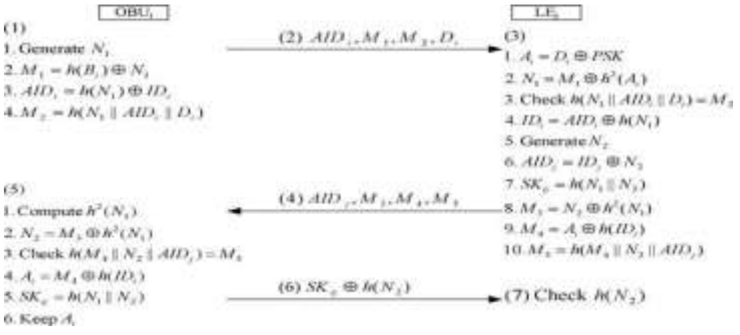

OBU$_i$
LE$_j$
(1)
1. Generate N_1
2. $M_1 = h(B_i) \oplus N_1$
3. $AID_i = h(N_1) \oplus ID_i$
4. $M_2 = h(N_1 \| AID_i \| D_i)$
(2) AID_i, M_1, M_2, D_i
(3)
1. $A_i = D_i \oplus PSK$
2. $N_1 = M_1 \oplus h^2(A_i)$
3. Check $h(N_1 \| AID_i \| D_i) = M_2$
4. $ID_i = AID_i \oplus h(N_1)$
5. Generate N_2
6. $AID_j = ID_j \oplus N_2$
7. $SK_{ij} = h(N_1 \| N_2)$
8. $M_3 = N_2 \oplus h^2(N_1)$
9. $M_4 = A_i \oplus h(ID_j)$
10. $M_5 = h(M_4 \| N_2 \| AID_j)$
(4) AID_j, M_3, M_4, M_5
(5)
1. Compute $h^2(N_1)$
2. $N_2 = M_3 \oplus h^2(N_1)$
3. Check $h(M_4 \| N_2 \| AID_j) = M_5$
4. $A_i = M_4 \oplus h(ID_j)$
5. $SK_{ij} = h(N_1 \| N_2)$
6. Keep A_i
(6) $SK_{ij} \oplus h(N_2)$
(7) Check $h(N_2)$

Fig. 7. General authentication procedure.

F. General Authentication Procedure

The OBU performs the general authentication procedure after the user completes the login procedure. Note that the OBU never uses the real identity of the user to perform the authentication procedure so nobody can obtain the user's real identity (i.e., ID_i) via the intercepted message. Fig. 7 shows the steps of the procedure.

Step 1) The OBU_i generates a random number N_1 and calculates the message M_1 as $h(B_i)\oplus N_1$. Then, it computes the alias AID_i as $h(N_1)\oplus \text{ID}_i$, and generates the message M_2 as $h(N_1||\text{AID}_i||D_i)$.

Step 2) OBU_i →LE_j : The OBU_i sends an authentication request (i.e., AID_i, M_1, M_2, D_i) to the LE_j .

Step 3) The LE_j verifies that the OBU_i is trustful: On receipt of the authentication request, the LE_j uses a secure preshared key (i.e., PSK) to obtain A_i (i.e., $A_i = D_i\oplus$ PSK). The LE retrieves the value of N_1 (i.e., $N_1=M_1\oplus h^2(A_i)$) and then checks whether $h(N_1||\text{AID}_i||D_i)$ is equal to M_2. It rejects the authentication request if $h(N_1 ||\text{AID}_i||D_i)$ and M_2 do not match, which means the authentication message has been modified. Next, the LE_j computes ID_i as $\text{AID}_i\oplus h(N_1)$, generates a random number N_2, computes AID_j as $\text{ID}_j \oplus N_2$, and calculates a session key SK_{ij} as $h(N_1||N_2)$. Finally, the LE_j computes the authentication reply message (i.e., AID_j , M_3, M_4, M_5), where M_3 is $N_2\oplus h^2(N_1)$, M_4 is $A_i\oplus h(ID_i)$, and M_5 is $h(M_4||N_2||\text{AID}_j)$.

Step 4) LE_j →OBU_i: The LE_j returns the authentication reply message (i.e., AID_j , M_3, M_4, M_5) to the OBU_i.

Step 5) The OBU verifies that the LE is trustful: The OBU_i computes the value of $h^2(N_1)$, retrieves the random number N_2 (i.e., $N_2=M_3\oplus h^2(N_1)$), and checks whether $h(M_4||N_2||\text{AID}_j)$ is equal to M_5. If the information is correct, the OBU_i calculates the value of A_i (i.e., $A_i=M_4\oplus h(\text{ID}_i)$), computes the session key (i.e., $\text{SK}_{ij} =h(N_1||N_2)$), and stores A_i in the security hardware.

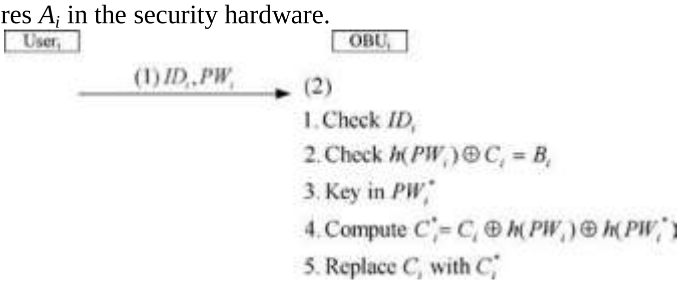


Fig. 8. Password change procedure.

Step 6) OBU_i→LE_j : The OBU_i sends the message (i.e., $\text{SK}_{ij} \oplus h(N_2)$) to the LE_j .

Step 7) The LE uses the session key SK_{ij} to retrieve the value (i.e., $h(N_2)$). Then, it checks this value to prevent an invalid OBU from executing a replay attack.

In this time, this OBU becomes trustful and obtains an authorized parameter (i.e., PSK = $A_i \oplus D_i$) when it is authenticated successfully. Thus, the other mistrustful OBUs can be authenticated by it without necessarily finding an LE.

G. Trust-Extended Authentication Procedure

We adopt the trust-extended mechanism based on the con-cept of transitive trust relationships to improve the perfor-mance of the authentication procedure. The state of a mis-trustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated success-fully. Then, the trustful OBU plays the role of LE temporarily to assist with the authentication procedure of a mistrustful OBU. In this procedure, the trustful vehicle performs the authentication procedure and works as an LE. Note that it still does not need to store the authentication information of the user. Hence, our scheme only has a few storage spaces. Then, the steps of the general authentication and the trust-extended authentication procedures are the same. As a result, all vehicles in a VANET can complete the authentication procedure quickly.

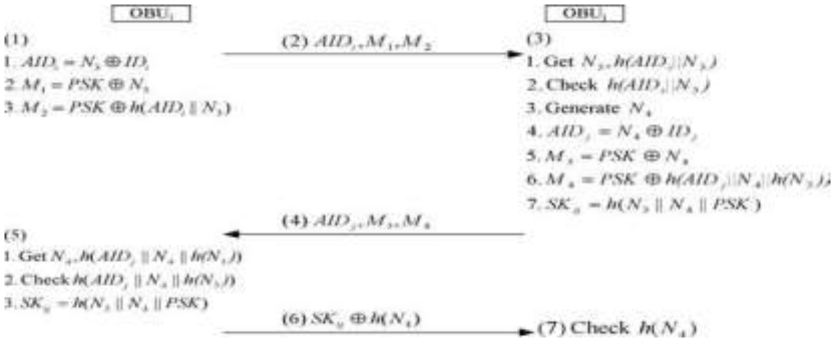

OBU$_i$
OBU$_j$
(1)
1. $AID_i = N_3 \oplus ID_i$
2. $M_1 = PSK \oplus N_3$
3. $M_2 = PSK \oplus h(AID_i \| N_3)$
(2) AID_i, M_1, M_2
(3)
1. Get $N_3, h(AID_i \| N_3)$
2. Check $h(AID_i \| N_3)$
3. Generate N_4
4. $AID_j = N_4 \oplus ID_j$
5. $M_3 = PSK \oplus N_4$
6. $M_4 = PSK \oplus h(AID_j \| N_4 \| h(N_3))$
7. $SK_{ij} = h(N_3 \| N_4 \| PSK)$
(4) AID_j, M_3, M_4
(5)
1. Get $N_4, h(AID_j \| N_4 \| h(N_3))$
2. Check $h(AID_j \| N_4 \| h(N_3))$
3. $SK_{ij} = h(N_3 \| N_4 \| PSK)$
(6) $SK_{ij} \oplus h(N_4)$
(7) Check $h(N_4)$

Fig. 9. Secure communication procedure.

H. Password Change Procedure

Although the password change procedure is optional, we still discuss it for completeness. This procedure is invoked when a user wants to change his password. It can be completed without any assistance from the AS since the security hardware of the OBU stores the parameters B_i and C_i. Fig. 8 shows the password change procedure and the steps are described as follows.

Step 1) The user keys in his ID_i and PW_i.
Step 2) The OBU checks the ID_i and verifies that $h(\mathrm{PW}_i)\oplus C_i$ is equal to B_i. If the information is correct, the user can key in the new password $\mathrm{PW}^*{}_i$. The OBU then
computes $C_i^* = C_i \oplus h(\mathrm{PW}_i) \oplus h(\mathrm{PW}^*{}_i) = B_i \oplus h(\mathrm{PW}^*{}_i)$ as the password and replaces C_i with C_i^* .

I. Secure Communication Procedure

Two trustful vehicles perform the secure communication procedure when they want to communicate with each other, as shown in Fig. 9. The steps are described as follows.

Step 1) After the login procedure, the OBU_i generates an alias AID_i and the messages for the authentication request (i.e., M_1, M_2), where N_3 is another random number, AID_i is $N_3 \oplus \mathrm{ID}_i$, M_1 is $\mathrm{PSK}\oplus N_3$, and M_2 is $\mathrm{PSK}\oplus h(\mathrm{AID}_i \,||\, N_3)$. Note that PSK is obtained from the general/trust-extended authentication procedure.
Step 2) OBU_i→OBU_j : The OBU_i sends a secure communi-cation request (i.e., AID_i, M_1, M_2) to the OBU_j .
Step 3) The OBU_j verifies that the OBU_i is trustful: on re-ceipt of the request, the OBU_j uses PSK to obtain N_3 from M_1 and then checks the value of $h(\mathrm{AID}_i||N_3)$. If the value is not correct, it means the message has been modified, and the OBU_j rejects the request. Next, the OBU_j generates a random number N_4, com-putes its alias AID_j , and calculates a session key SK_{ij} as $h(N_3||N_4||PSK)$. Then, the OBU_j computes the reply message (i.e., M_3, M_4), where M_3 is PSK $\oplus N_4$ and M_4 is PSK $\oplus h(\mathrm{AID}_i||N_4||h(N_3))$.
Step 4) OBU_j →OBU_i: The OBU_j returns the reply message (i.e., AID_j , M_3, M_4) to the OBU_i.
Step 5) The OBU_i verifies that the OBU_j is trustful: the OBU_i computes the value of $h(N_3)$, uses PSK to retrieves the random number N_4, and checks the value of $h(\mathrm{AID}_j\,||N_4||h(N_3))$. If the information is correct, the OBU_i calculates the session key (i.e., $\mathrm{SK}_{ij} = h(N_3||N_4||\mathrm{PSK})$) for this communication.
Step 6) OBU_i→OBU_j : the OBU_i sends the message (i.e., $\mathrm{SK}_{ij} \oplus h(N_4)$) to the OBU_j.
Step 7) The OBU_j uses the session key SK_{ij} to retrieve the value (i.e., $h(N_4)$). It then checks this value to prevent an invalid OBU from executing a replay attack. Then, two trustful vehicles can use this session key to communicate securely.

J. Key Revocation Procedure

In our scheme, the mechanism of key revocation is based on timer which treats as the lifetime of the key. The authentication state of a mistrust vehicle becomes trustfully and obtains an authorized parameter (i.e., *PSK*) when the vehicle performs the authentication procedure successfully. Then, the authentication state in the hello message is changed to trust and the secure hardware sets up a timer to count down. When the lifetime of the key is over, the state of the vehicle is changed to mistrust. Certainly, our scheme is easy to integrate with other key revocation schemes (e.g., token-based mechanism [20]). In fact, the system can ask the trustful vehicle to perform the key update procedure (i.e., Section IV-K) on the hour (or several hours) for reducing the compromised probability.

K. Key Update Procedure

The key update procedure is performed when the key lifetime of the TV will terminate. The TV extends its state of trustfulness after it finishes the key update procedure. Fig. 10 shows the key update procedure and the steps are depicted as follows.

Step 1) The key update procedure is triggered when the key lifetime is below the predefined threshold (i.e., *TH*). The OBU$_i$ prepares to send a key update message to the LE. The OBU$_i$ generates a random number N_5, and then it computes the messages M_1 as PSK$_{\text{old}}\oplus N_5$, M_2 as PSK$_{\text{old}}\oplus$MSG$_{\text{KU}}$, and M_3 as $h(M_1||M_2)$.

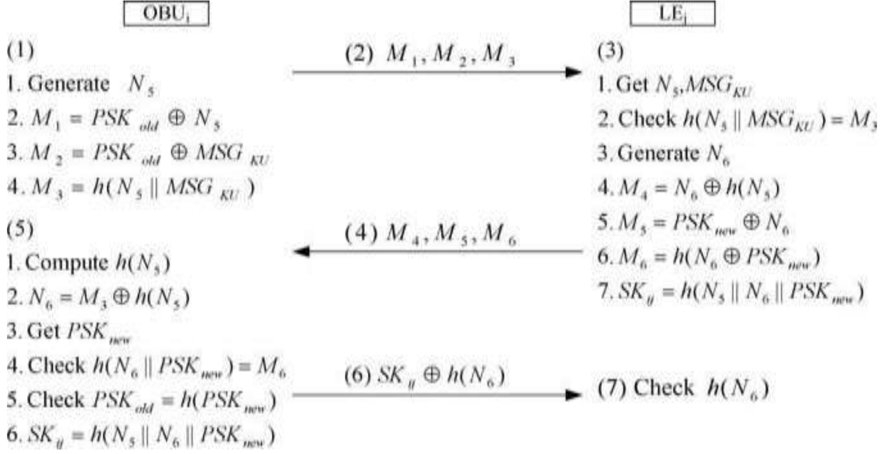

OBU$_i$
LE$_j$
(1)
1. Generate N_5
2. $M_1 = PSK_{old} \oplus N_5$
3. $M_2 = PSK_{old} \oplus MSG_{KU}$
4. $M_3 = h(N_5 \| MSG_{KU})$
(2) M_1, M_2, M_3
(3)
1. Get N_5, MSG_{KU}
2. Check $h(N_5 \| MSG_{KU}) = M_3$
3. Generate N_6
4. $M_4 = N_6 \oplus h(N_5)$
5. $M_5 = PSK_{new} \oplus N_6$
6. $M_6 = h(N_6 \oplus PSK_{new})$
7. $SK_{ij} = h(N_5 \| N_6 \| PSK_{new})$
(4) M_4, M_5, M_6
(5)
1. Compute $h(N_5)$
2. $N_6 = M_3 \oplus h(N_5)$
3. Get PSK_{new}
4. Check $h(N_6 \| PSK_{new}) = M_6$
5. Check $PSK_{old} = h(PSK_{new})$
6. $SK_{ij} = h(N_5 \| N_6 \| PSK_{new})$
(6) $SK_{ij} \oplus h(N_6)$
(7) Check $h(N_6)$

Fig. 10. Key update procedure.

Step 2) OBU$_i$→LE$_j$: The OBU$_i$ sends a key update request (i.e., M_1, M_2, M_3) to the LE$_j$.

Step 3) The LE$_j$ uses the current PSK (i.e., PSK$_{\text{old}}$) to retrieve N_5 and MSG$_{\text{KU}}$. It rejects the key update request if the value of $h(M_1||M_2)$ and M_3 do not match, which means the message has been modified. Next, the LE$_j$ generates a random number N_6 and computes the key update reply messages (i.e., M_4, M_5, M_6), where M_4 is $N_6\oplus h(N_5)$, M_5 is PSK$_{\text{new}}\oplus N_6$, and M_6 is $h(M_4||M_5)$. Note that the key set of PSK is generated by the hash-chain method. Therefore, the OBU cannot use the current PSK to infer the new PSK. Finally, the LE$_j$ calculates the session key (i.e., SK$_{ij}$) as $h(N_5||N_6||$PSK$_{\text{new}})$.

Step 4) LE$_j$→OBU$_i$: The LE$_j$ returns the reply message (i.e., M_4, M_5, M_6) to the OBU$_i$.

Step 5) On receipt of the key update reply message, the OBU$_i$ computes the value of $h(N_5)$, retrieves the random number N_6 (i.e., N_6=$M_4\oplus h(N_5)$), and obtains the new PSK. Next, the OBU$_i$ checks the value of h($M_4||M_5$). Then, the OBU$_i$ checks whether h(PSK$_{\text{new}}$) is equal to PSK$_{\text{old}}$. If the value is equal, the OBU$_i$ updates the PSK and calculates the session key SK$_{ij}$ as $h(N_5||N_6||$PSK$_{\text{new}})$.

Step 6) OBU$_i$→LE$_\text{j}$: The OBU$_i$ sends the message (i.e., SK$_{ij}$ $\oplus h(N_6)$) to the LE$_j$.

Step 7) The LE$_j$ uses the session key SK$_{ij}$ to retrieve the value (i.e., $h(N_6)$). It then checks this value to prevent an invalid OBU from executing a replay attack. Then, two trustful vehicles can use this session key to communicate securely.

V. ANALYSIS

This section discusses the security analysis, computational cost, and storage cost of TEAM. The security properties of TEAM are based on a collision-free one-way hash function (e.g., SHA-512 [18]). For a one-way hash function $h(\ )$, when the value of x is given, it is straightforward to compute $h(x)$; however, given the value of $h(x)$, computing the value of x is very difficult or incurs a high computational cost. Besides, in the login procedure, the security hardware has a retry limit to prevent the attacker using a force technique to guess the user's password. A TEAM satisfies the following security requirements.

A. Security Analysis

Due to the page limit, we only discuss the security features of TEAM. Therefore, we use the same scheme [24]–[29] to present the security analysis. The detailed cryptanalysis of TEAM is listed in our future work.

1) *Anonymity:* Under the proposed scheme, the original identity of every user is always converted into an alias that is based on a random number (e.g., $\text{AID}_i = h(N_1) \oplus \text{ID}_i$). Therefore, an adversary cannot determine the user's original identity without knowing the random number N_1 chosen by the OBU. Moreover, our anonymity mechanism is a dynamic identification process.

2) *No verification table*: The AS, LEs, and TVs do not need to store the user's verification table. Therefore, even if an adversary can access their database, he cannot obtain the user's authentication information.

3) *Location privacy*: Even if an adversary intercepts a number of messages during a certain period, he cannot trace the user's physical position because the system's anonymity mechanism uses a dynamic identification process, and generation of the session key is based on a nonce. Moreover, TEAM can utilize the random silent period scheme [7] or group characteristic [8] to enhance the location privacy when the OBUs do not have to access the service. Therefore, TEAM can improve the location privacy.

4) *Mutual authentication*: A mutual authentication process is necessary. The LE needs to verify that the OBU is a legal user, and the OBU needs to ensure that the LE is genuine. In the general authentication procedure, the LE authenticates the OBU in Step 3, and the OBU authenticates the LE in Step 5, respectively. If the Attacker intercepts the messages and wants to forge a valid OBU/LE, it must generate a valid message to LE/OBU. However, the attacker cannot compute the valid message because he does not know the secure key (i.e., PSK_i) and the random number (i.e., N_1 and N_2). In addition, the secure communication procedure also achieves the mutual authentication (i.e., in Step 3, the OBU_j authenticates the OBU_i, and the OBU_i authenticates the OBU_j in Step 5).

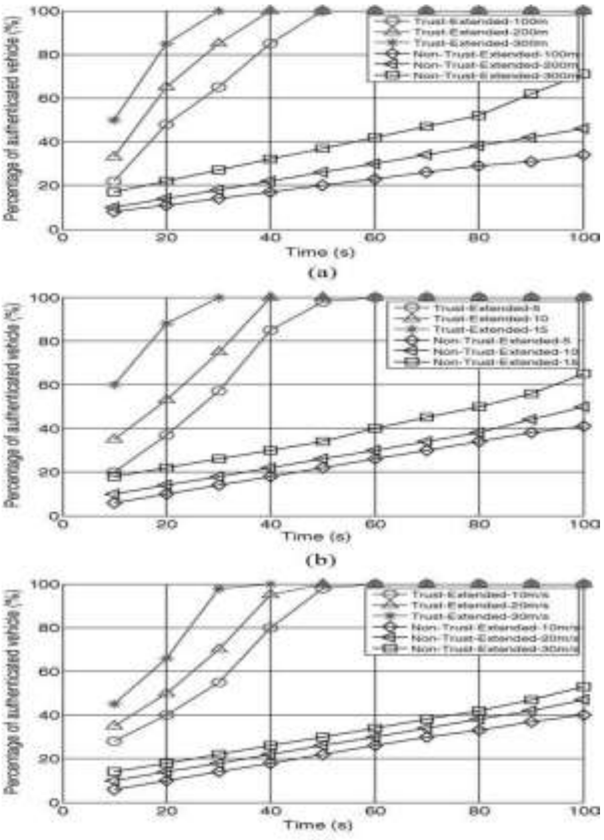
5) *Clock synchronization is not required*: In timestamp-based authentication schemes, the clocks of all vehicles must be synchronized. In TEAM, we provide a nonce-based authentication mechanism instead of timestamps, which cause serious time synchronization problems.

6) *Resistance to replay attacks*: To protect the proposed scheme from replay attacks, we add a random number to the authentication message. If an adversary intercepted the message and tried to impersonate a valid OBU by replaying the message immediately, the LE would reject the request because the nonce in the replayed messages would be invalid. Moreover, the OBU also checks the random number sent by the LE to prevent replay attacks.

7) *Session key agreement*: The proposed approach only makes one round trip between the OBU and the LE to generate the session key. Then, the key is used to encrypt subsequent packets to ensure that the communications are confidential. Since the session key is generated by a random number and a hash function, the adversary is hard to guess or to derive the session key from the intercepted messages. Moreover, the random numbers are different in each session so the session key is capable of resisting the replay attacks.

8) *Resistance to modification attacks*: An adversary can attempt to modify the authentication and reply messages. However, we use a one-way hash function to ensure that information cannot be modified. Therefore, this attack will be detected because an attacker has no way to obtain the value of the random number to generate the legitimate message. If an attacker transmits a modified packet to the LE/vehicle, the packet can be easily identified by checking the hash values. Thus, our scheme ensures the message integrity.

9) *Resistance to forgery attacks*: If an invalid OBU at-tempts to forge another valid OBU's ID (i.e., $\text{AID}^*{}_i$), the authentication will be unsuccessful (i.e., Step 3 in



(a)

(b)

(c)

Fig. 6). Although the attacker forges an alias ID (i.e., $\text{AID}^*{}_i = h(N_1) \oplus ID_i^*$), it cannot determine the valid

authentication parameter (i.e., D_i^*) required to obtain authentication. This is because the OBU does not know the AS's secret key (i.e., x), so it cannot compute the value of A_i correctly. Moreover, the secret key is protected by the one-way hash function $h(\ )$, and it is computationally infeasible to derive x from the value $h(x)$.

10) *Fast error detection*: In the login or password change procedures, the OBU will detect an error immediately if an attacker keys in the wrong user ID or password. (i.e., Step 2 in the login procedure and Step 2 in the password change procedure)

11) *Choose and change password easily*: Users can choose or change their passwords without the AS's assistance and constrains, so that it is easy for them to memorize their passwords.

12) *Perfect forward secrecy*: The perfect forward secrecy means that the secrecy of previous session keys es-tablished by trustful entities is not affected if the new session keys of one or more entities are compromised. Our scheme achieves the perfect forward secrecy. This is because the session key of our scheme is generated by a hash function and the random number.

13) *Resistance to man-in-the-middle attack*: The password and the secret key of the system are used to prevent the man-in-middle attack. The attacker cannot pretend to be

trustful vehicle or LE to authenticate other MVs since he does not own the password (i.e., PW_i) or the secret key (i.e., x).

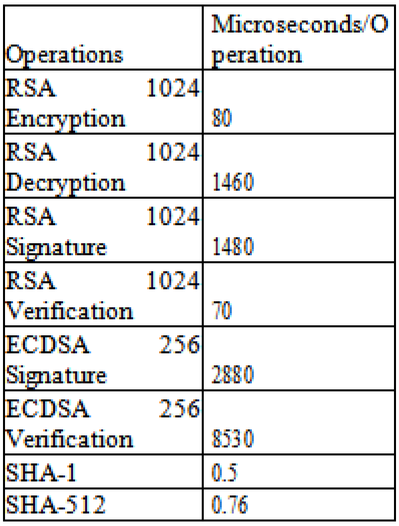
14) *Resistance to key lifetime self-extension attack*: In our scheme, a trustful vehicle cannot extend its authentica-tion key lifetime (i.e., PSK_i) when the key lifetime is over. This is because the generation of the authentication key is based on one-way hash chain function (i.e., Fig. 4). Therefore, the vehicle cannot compute a valid authentication key.

B. Analysis of Computational Cost

In the analysis of the computational cost, we use the following notations: "–" means there is no computational cost in that phase; n: the number of OBUs in the VANET; C_h denotes the cost of executing the one-way hash function; C_{XOR} denotes the cost of executing the XOR operation; and

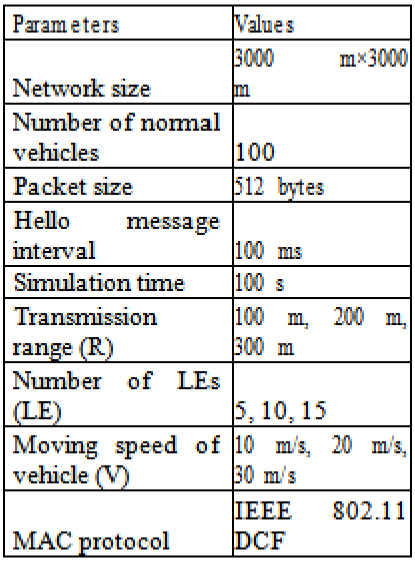
denotes the cost of generating a random number. The computational cost of TEAM is shown in Table II. TEAM is efficient in terms of the computational cost because it is only based on an XOR operation and a hash function without using

TABLE II
Computing Process Time



Operations	Microseconds/Operation
RSA 1024 Encryption	80
RSA 1024 Decryption	1460
RSA 1024 Signature	1480
RSA 1024 Verification	70
ECDSA 256 Signature	2880
ECDSA 256 Verification	8530
SHA-1	0.5
SHA-512	0.76

TABLE Ⅲ
Simulation Parameters



Parameters	Values
Network size	3000 m×3000 m
Number of normal vehicles	100
Packet size	512 bytes
Hello message interval	100 ms
Simulation time	100 s
Transmission range (R)	100 m, 200 m, 300 m
Number of LEs (LE)	5, 10, 15
Moving speed of vehicle (V)	10 m/s, 20 m/s, 30 m/s
MAC protocol	IEEE 802.11 DCF

Asymmetric cryptography. We use Crypto++ Library [19] to evaluate the computing process time of operation. Table III shows the computing process time of each operation. We can see that the computing process time of hash operation (i.e., SHA-1 and SHA-512) is faster than the asymmetric encryption (i.e., RSA-based operations).

C. Analysis of Storage Cost

In the asymmetric cryptography schemes, each vehicle needs to store the entire public key of users. However, this behavior in VANETs is costly and impractical. The complexity of storage cost of asymmetric cryptography is $O(n)$, where n is the total number of vehicles in VANETs. Thus, these asymmetric cryptography schemes are not scalable since the storage cost raises when the number of vehicles increases. On the contrary, the number of vehicles does not affect the storage cost of TEAM, and the complexity of storage cost of TEAM is $O(1)$. The normal vehicle only stores a few security parameters (i.e., ID_i, B_i, C_i, D_i, $h(\ )$) in security hardware of OBU for performing the authentication procedure, and the LE stores a key set. As a result, TEAM saves lots of storage cost and with high scalability compared with the asymmetric cryptography schemes.

Fig. 11. Performance results of the trust-extended and nontrust-extended schemes with different parameters. (a) Varied transmission range: LE = 10 and V = 20 m/s. (b) Varied number of LEs: R = 200 m and V = 20 m/s.
(c) Varied vehicle speed: R = 200 m and LE = 10.

D. Trust-Extended Versus Nontrust Extended

Here, we discuss the performance of authentication pro-cedure of the trust-extended and nontrust-extended schemes via NS-2 simulator [21]. The simulation environment is a grid topology over a 3000 m × 3000 m area. We use a tool (mobility model generator for vehicular networks; MOVE) [22], [23] to rapidly generate realistic mobility models for VANET simulations. The LEs and normal vehicles are dis-tributed randomly in the network. Each simulation result is
the average of ten runs. The parameters and values used in the simulations are listed in Table IV.

Fig. 11 depicts the performance results of the trust-extended and nontrust-extended schemes with different parameters. The percentage of authenticated vehicle (i.e., the value of y-axis) is computed as the authenticated vehicles divide by the entire vehicles. As a result, the larger transmission range, the greater amount of LEs, and the faster vehicle speed are going to quickly increase the percentage of authenticated vehicle due to the MV has higher probability to meet the trustful vehicle. Moreover, we can see that the trust-extended scheme is better than the nontrust-extended scheme. This is because the trustful vehicle plays the LE role temporarily to assist with the authentication procedure of the MV.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a decentralized lightweight authentication scheme called TEAM to protect valid users in VANETs from malicious attacks. The amount of cryptographic calculation under TEAM was substantially less than in exist-ing schemes because it only used an XOR operation and a hash function. Moreover, TEAM is based on the concept of transitive trust relationships to improve the performance of the authentication procedure. In addition, TEAM has a few storage spaces to store the authentication parameters.

REFERENCES

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

[2]. Dedicated Short Range Communications (DSRC) [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html

[3]. M. Nekovee and B. B. Bogason, "Reliable and efficient information dissemination in intermittently connected vehicular ad hoc networks," in *Proc. IEEE Vehicular Technol. Conf.*, Apr. 2007, pp. 2486–2490.

[4]. J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks," *IEEE Trans. Vehicular Technol.*, vol. 56, no. 6, pp. 3266–3277, Nov. 2007.

[5]. J.-F. Lee, C.-S. Wang, and M.-C. Chuang, "Fast and reliable emergency message dissemination mechanism in vehicular ad hoc networks," in

[6]. *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2010, pp. 1–6.

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

[8]. J. Freudiger, M. Raya, and M. Feleghhazi, "Mix zones for location privacy in vehicular networks," in *Proc. First Int. Workshop Wireless Netw. Intell. Transp. Syst.*, Aug. 2007, pp. 1–7.

[9]. K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: Robust location privacy scheme for VANET," *IEEE J. Selected Areas Commun.*, vol. 25, no. 8, pp. 1569–1589, Oct. 2007.

[10]. 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 Int. Conf. Commun.*, May 2008, pp. 1451–1457.

[11]. M.-C. Chuang and J.-F. Lee, "TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks," in *Proc. IEEE Int. Conf. Consumer Electron., Commun. Netw.*, Apr. 2011, pp. 1758–1761.

[12]. K. Sampigethava, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing location privacy for vANET," in *Proc. ACM VANET*, Sep. 2006, pp. 1–15.

[13]. C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 246–250.

[14]. R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular commu-nications," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1229–1237.

[15]. H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 16–22, Aug. 2009.

[16]. P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicu-lar communication systems: Design and architecture," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[17]. G. Guette and C. Bryce, "Using TPMs to secure vehicular ad-hoc networks (VANETs)," in *Proc. Int. Federation Informat. Process.*, May 2008, pp. 106–116.

[18]. A. A. Wagan, B. M. Mughal, and H. Hasbullah, "VANET security framework for trusted grouping using TPM hardware," in *Proc. IEEE Int. Conf. Commun. Software Netw.*, Feb. 2010, pp. 309–3012.

[19]. NIST, U.S. Department of Commerce, "Secure Hash Standard," *U.S. Federal Information Processing Standard (FIPS)*, Aug. 2002.

[20]. *Crypto++ Library 5.6.1* [Online]. Available: http://www.cryptopp.com/

[21]. S. Machiraju, H. Chen, and J. Bolot, "Distributed authentication for low-cost wireless networks," in *Proc. ACM HotMobile*, Feb. 2008, pp. 55–59.

[22]. *The Network Simulator 2 (NS2)* [Online]. Available: http://www.isi.edu/ nsnam/ns/

[23]. *Mobility Model Generator for Vehicular Networks (MOVE)* [Online]. Available: http://mac.softpedia.com/get/Utilities/MOVE.shtml

[24]. F. K. Karnadi, Z. H. Mo, and K.-C. Lan, "Rapid generation of realistic mobility models for VANET," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 2506–2511.

[25]. F. We and X. Li, "An improved dynamic ID-based remote user authen-tication with key agreement scheme," *Comput. Electr. Eng.*, vol. 38, pp. 381–387, Mar. 2012.

[26]. S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *J. Netw. Comput. Appl.*, vol. 34, pp. 609–618, Mar. 2011.

[27]. M. K. Khan, S.-K. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a 'more

efficient & secure dynamic ID-based remote user authentication scheme’,” *Comput. Commun.*, vol. 34, pp. 305–309, Mar. 2011.

[28]. C.-C. Lee, T.-H. Lin, and R.-X. Chang, “A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards,” *Expert Syst. Appl.*, vol. 38, pp. 13863–13870, Oct. 2011.

[29]. T. H. Chen, H. C. Hsiang, and W. K. Shih, “Security enhancement on an improvement on two remote user authentication schemes using smart cards,” *Future Generation Comput. Syst.*, vol. 27, pp. 377–380, Apr. 2011.

[30]. X. Li, Y. Xiong, J. Ma, and W. Wang, “An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards,” *J. Netw. Comput. Appl.*, vol. 35, pp. 763–769, Mar. 2012.