Vehicular Ad Hoc Networks Authentication Mechanism

Priyasi G, Ranjith Kumar M
(M.E-CSE), SRG Engineering College
(Professor), SRG Engineering College

Abstract: In the field of wireless mobile networking the security of vehicular ad hoc networks has been receiving a significant amount of attention because vehicular ad hoc networks 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.

I. INTRODUCTION

Vehicular ad hoc networks have been attracted increasing attention from both industry and academia [1]. The major components of vehicular ad hoc networks 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 infrastructure 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 vehicular ad hoc networks. However, these schemes do not consider the security problem. Recently, the security issue in vehicular ad hoc networks 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 vehicular ad hoc networks 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 attacks 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 pseudo identity. 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 vehicular ad hoc networks because they use asymmetric cryptography or a digital signature verification scheme, which results in high computation costs, long authentication latency, and a large storage space. Zhang et al. [8] proposed an RSU-aided messages authentication scheme (RAISE), which uses the symmetric key hash message authentication 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 vehicular ad hoc networks 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 vehicular ad hoc networks.

#### A. Transitive Trust Relationships

In vehicular ad hoc networks, vehicles can be classified into 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 vehicular ad hoc networks: 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\(_j\).

#### 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.
The OBU verifies that the LE i→OBUj sends the message (i.e., AIDj, M3, M4, Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK = Ai ⊕ Di) when it is authenticated successfully. Thus, the other mistrustful OBUs can be authenticated by it without necessarily finding an LE.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) when it is authenticated successfully. Thus, the other mistrustful OBUs can be authenticated by it without necessarily finding an LE.

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 vehicular ad hoc networks.

1) Efficiency: In vehicular ad hoc networks, the computational cost of vehicles 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 authentication messages of the OBU but it still failed to track the location of the vehicle.

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., IDi) via the intercepted message. Fig. 7 shows the steps of the procedure.

Step 1) The OBUi generates a random number N1 and calculates the message M1 as h(Bi) ⊕ N1.
Step 2) OBUi→LEj: The BUi sends an authentication request (i.e., AIDi, M1, M2, Di) to the LEj.
Step 3) The LEj verifies that the OBUi is trustful: On re-ceipt of the authentication request, the LEj uses a secure preshared key (i.e., PSK) to obtain Ai (i.e., A random number N2, computes AIDj as IDj ⊕ N2, and calculates a session key SKij as h(N1||N2)). Finally, the LEj computes the authentication reply message (i.e., AIDj, M3, M4, M5), where M3 is N 2 ⊕ h2(N1), M4 is A i ⊕ h(IDi), and M5 is h(M4||N2||AIDj).
Step 4) LEj→OBU: The LEj returns the authentication reply message (i.e., AIDj, M3, M4, M5) to the OBUi.
Step 5) The OBU verifies that the LE is trustful: The OBUi computes the value of 2(N1), retrieves the random number N2 (i.e., N2 = M3 ⊕ h2(N1)), and checks whether h(M4||N2||AIDj) is equal to M5. If the information is correct, the OBUi calculates the value of Ai (i.e., Ai = M4 ⊕ h(IDi)), computes the session key (i.e., SKij = h(N1||N2)), and stores Ai in the security hardware.

Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated successfully. 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 vehicular ad hoc networks can complete the authentication procedure quickly.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated successfully. 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 vehicular ad hoc networks can complete the authentication procedure quickly.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated successfully. 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 vehicular ad hoc networks can complete the authentication procedure quickly.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated successfully. 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 vehicular ad hoc networks can complete the authentication procedure quickly.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated successfully. 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 vehicular ad hoc networks can complete the authentication procedure quickly.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) 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 concept of transitive trust relationships to improve the performance of the authentication procedure. The state of a mistrustful OBU becomes trustful and then obtains an authorized parameter (i.e., PSK) when the OBU is authenticated successfully. 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 vehicular ad hoc networks can complete the authentication procedure quickly.

Step 6) OBUi→LEj: The OBUi sends the message (i.e., SKij ⊕ h(N2)) to the LEj.
Step 7) The LE uses the session key SKij to retrieve the value (i.e., h(N2)). 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 = Ai ⊕ Di) when it is authenticated successfully. Thus, the other mistrustful OBUs can be authenticated by it without necessarily finding an LE.
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 IDi and PWi. The OBU 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 $N_3 \oplus ID_i$, $M_1$ is $PSK \oplus N_3$, and $M_2$ is $PSK \oplus h(AID_i || N_3)$. Note that PSK is obtained from the general trust-extended authentication procedure.

Step 2) $OBU_i \rightarrow OBU_j$: The OBUi sends a secure communication request (i.e., $AID_i$, $M_1$, $M_2$) to the OBUj.

Step 3) The OBUj verifies that the OBUi is trustful: on receipt of the request, the OBUj uses PSK to obtain $N_3$ from $M_1$ and then checks the value of $h(AID_i || N_3)$. If the value is not correct, it means the message has been modified, and the OBUj rejects the request. Next, the OBUj generates a random number $N_4$, computes its alias $AID_j$, and calculates a session key $SK_{ij}$ as $h(N_3 || N_4 || PSK)$. Then, the OBUj computes the reply message (i.e., $M_3$, $M_4$), where $M_3$ is $PSK \oplus N_4$ and $M_4$ is $PSK \oplus h(\text{AID}_j || N_3 || h(N_3))$.

Step 4) $OBU_j \rightarrow OBU_i$: The OBUj returns the reply message (i.e., $AID_j$, $M_3$, $M_4$) to the OBUi.

Step 5) The OBUi verifies that the OBUj is trustful: the OBUi computes the value of $h(N_3)$, uses PSK to retrieve the random number $N_4$, and checks the value of $h(\text{AID}_j || N_4 || h(N_3))$. If the information is correct, the OBUi calculates the session key (i.e., $SK_{ij} = h(N_3 || N_4 || PSK)$) for this communication.

Step 6) $OBU_i \rightarrow OBU_j$: The OBUi sends the message (i.e., $SK_{ij} \oplus h(N_4)$) to the OBUj.

Step 7) The OBUj 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 OBUj prepares to send a key update message to the LE. The OBUj generates a random number $N_5$.

Step 2) $OBU_j \rightarrow LE_j$: The OBUj sends a key update request (i.e., $M_1$, $M_2$, $M_3$) to the LEj.

Step 3) The LEj uses the current PSK (i.e., $PSK_{\text{old}}$) to retrieve $N_5$ and $MSG_{KU}$. 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 LEj generates a random number $N_6$ and computes the key update reply messages (i.e., $M_4$).
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., \( \text{AID}_i = h(N1) \oplus \text{Id}_i \)). Therefore, an adversary cannot determine the user’s original identity without knowing the random number \( N1 \) 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., \( \text{PSK}_i \)) and the random number (i.e., \( N1 \) and \( N2 \)). In addition, the secure communication procedure also achieves the mutual authentication (i.e., in Step 3, the OBU authenticates the OBU, and the OBU authenticates the OBU 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 attempts to forge another valid OBU’s ID (i.e., AID+ i), the authentication will be unsuccessful (i.e., Step 3 in Fig. 6). Although the attacker forges an alias ID (i.e., AID* i = h(N1) ⊕ 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 Ai 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 established 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 to prevent the man-in-middle attack. The attacker cannot pretend to be a trustful vehicle or LE to authenticate other MVs since he does not own the password (i.e., PWi) or the secret key (i.e., x).

14) Resistance to key lifetime self-extension attack: In our scheme, a trustful vehicle cannot extend its authentication key lifetime (i.e., PSKi) 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 vehicular ad hoc networks; C denotes the cost of executing the one-way hash function; C XOR denotes the cost of executing the XOR operation; and C hash 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 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 vehicular ad hoc networks is costly and impractical. The complexity of storage cost of asymmetric cryptography is O(n), where n is the total number of vehicles in vehicular ad hoc networks. 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., IDi, Bi, CI, Di, 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 \( V_\text{t} = 20 \) m/s. (b) Varied number of LEs: \( R = 200 \) m and \( V_\text{t} = 20 \) 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 procedure of the trust-extended and nontrust-extended schemes via NS-2 simulator [21]. The simulation environment is a grid topology over a 3000 m \( \times \) 3000 m area. We use a tool (mobility model generator for vehicular networks; MOVE) [22], [23] to rapidly generate realistic mobility models for vehicular ad hoc networks simulations. The LEs and normal vehicles are distributed 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 vehicular ad hoc networks 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


