This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

EMAP: Expedite Message Authentication Protocol for Vehicular Ad Hoc Networks

Albert Wasef and Xuemin (Sherman) Shen, IEEE Fellow Department of Electrical and Computer Engineering

University of Waterloo, Waterloo, Ontario, Canada

Email: awasef@bbcr.uwaterloo.ca, xshen@bbcr.uwaterloo.ca

Abstract

Vehicular Ad Hoc Networks (VANETs) adopt the Public Key Infrastructure (PKI) and Certificate Revocation Lists (CRLs) for their security. In any PKI system, the authentication of a received message is performed by checking if the certificate of the sender is included in the current CRL, and verifying the authenticity of the certificate and signature of the sender. In this paper, we propose an Expedite Message Authentication Protocol (EMAP) for VANETs, which replaces the time-consuming CRL checking process by an efficient revocation checking process. The revocation check process in EMAP uses a keyed Hash Message Authentication Code $(HMAC)$, where the key used in calculating the $HMAC$ is shared only between non-revoked On-Board Units (OBUs). In addition, EMAP uses a novel probabilistic key distribution, which enables non-revoked OBUs to securely share and update a secret key. EMAP can significantly decrease the message loss ratio due to the message verification delay compared with the conventional authentication methods employing CRL. By conducting security analysis and performance evaluation, EMAP is demonstrated to be secure and efficient.

Index Terms

Vehicular networks, Communication security, Message authentication, Certificate revocation.

I. Introduction

Vehicular ad-hoc networks (VANETs) have attracted extensive attentions recently as a promising technology for revolutionizing the transportation systems and providing broadband communication services to vehicles. VANETs consist of entities including On-Board Units (OBUs) and

Digital Object Indentifier 10.1109/TMC.2011.246

infrastructure Road-Side Units (RSUs). Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are the two basic communication modes, which respectively allow OBUs to communicate with each other and with the infrastructure RSUs.

Since vehicles communicate through wireless channels, a variety of attacks such as injecting false information, modifying and replaying the disseminated messages can be easily launched. A security attack on VANETs can have severe harmful or fatal consequences to legitimate users. Consequently, ensuring secure vehicular communications is a must before any VANET application can be put into practice. A well-recognized solution to secure VANETs is to deploy Public Key Infrastructure (PKI), and to use Certificate Revocation Lists (CRLs) for managing the revoked certificates. In PKI, each entity in the network holds an authentic certificate, and every message should be digitally signed before its transmission. A CRL, usually issued by a Trusted Authority (TA), is a list containing all the revoked certificates. In a PKI system, the authentication of any message is performed by first checking if the sender's certificate is included in the current CRL, i.e., checking its revocation status, then, verifying the sender's certificate, and finally verifying the sender's signature on the received message. The first part of the authentication, which checks the revocation status of the sender in a CRL, may incur long delay depending on the CRL size and the employed mechanism for searching the CRL. Unfortunately, the CRL size in VANETs is expected to be large for the following reasons: (1) To preserve the privacy of the drivers, i.e., to abstain the leakage of the real identities and location information of the drivers from any external eavesdropper [1]-[3], each OBU should be preloaded with a set of anonymous digital certificates, where the OBU has to periodically change its anonymous certificate to mislead attackers [4]. Consequently, a revocation of an OBU results in revoking all the certificates carried by that OBU leading to a large increase in the CRL size; (2) The scale of VANET is very large. According to the United States Bureau of Transit Statistics, there are approximately 251 million OBUs in the Unites States in 2006 [5]. Since the number of the OBUs is huge and each OBU has a set of certificates, the CRL size will increase dramatically if only a small portion of the OBUs is revoked. To have an idea of how large the CRL size can be, consider the case where only 100 OBUs are revoked, and each OBU has $25,000$ certificates [6]. In this case, the CRL contains 2.5 million revoked certificates. According to the employed mechanism for searching a CRL, the Wireless Access in Vehicular Environments (WAVE) standard [7] does not state that either a non-optimized search algorithm, e.g., linear search, or some sort of optimized search algorithm such as binary search, will be used for searching a CRL. In this paper, we consider both non-optimized and optimized search algorithms.

According to the Dedicated Short Range Communication (DSRC) [8], which is part of the WAVE standard, each OBU has to broadcast a message every $300\ msec$ about its location, velocity, and other telematic information. In such scenario, each OBU may receive a large number of messages every $300\ msec$, and it has to check the current CRL for all the received certificates, which may incur long authentication delay depending on the CRL size and the number of received certificates. The ability to check a CRL for a large number of certificates in a timely manner leads an inevitable challenge to VANETs.

To ensure reliable operation of VANETs and increase the amount of authentic information gained from the received messages, each OBU should be able to check the revocation status of all the received certificates in a timely manner. Most of the existing works overlooked the authentication delay resulting from checking the CRL for each received certificate. In this paper, we introduce an expedite message authentication protocol[1] (EMAP) which replaces the CRL checking process by an efficient revocation checking process using a fast and secure $HMAC$ function. EMAP is suitable not only for VANETs but also for any network employing a PKI system. To the best of our knowledge, this is the first solution to reduce the authentication delay resulting from checking the CRL in VANETs.

The remainder of the paper is organized as follows. The related works are discussed in section II. Section III introduces some preliminaries. The proposed EMAP is presented in section IV. Security analysis and performance evaluation are given in section V and section VI, respectively. Section VII concludes the paper.

II. Related Work

In VANETs, the primary security requirements are identified as entity authentication, message integrity, non-repudiation, and privacy preservation. The Public Key Infrastructure (PKI) is the most viable technique to achieve these security requirements [4],[10]. PKI employs Certificate Revocation Lists (CRLs) to efficiently manage the revoked certificates. Since the CRL size is

[1] Part of this work was presented at IEEE Globecom'09 [9].

expected to be very large, the delay of checking the revocation status of a certificate included in a received message is expected to be long.

In [10], Hubaux *et al.* identify the specific issues of security and privacy challenges in VANETs, and indicate that a Public Key Infrastructure (PKI) should be well deployed to protect the transited messages and to mutually authenticate network entities. In [4], Raya *et al.* use a classical PKI to provide secure and privacy preserving communications to VANETs. In this approach, each vehicle needs to pre-load a huge pool of anonymous certificates. The number of the loaded certificates in each vehicle should be large enough to provide security and privacy preservation for a long time, e.g., one year. Each vehicle can update its certificates from a central authority during the annual inspection of the vehicle. In this approach, revoking one vehicle implies revoking the huge number of certificates loaded in it.

In [11], Studer *et al.* propose an efficient authentication and revocation scheme called TACK. TACK adopts a hierarchy system architecture consisting of a central trusted authority and regional authorities (RAs) distributed all over the network. The authors adopted group signature where the trusted authority acts as the group manager and the vehicles act as the group members. Upon entering a new region, each vehicle must update its certificate from the RA dedicated for that region. The vehicle sends a request signed by its group key to the RA to update its certificate, the RA verifies the group signature of the vehicle and ensures that the vehicle is not in the current Revocation List (RL). After the RA authenticates the vehicle, it issues short-lifetime region-based certificate. This certificate is valid only within the coverage range of the RA. It should be noted that TACK requires the RAs to wait for some time, e.g., 2 seconds, before sending the new certificate to the requesting vehicle. This renders the vehicle not able to send messages to neighboring vehicles within this period, which makes TACK not suitable for the safety applications in VANETs as the WAVE standard [7] requires each vehicle to transmit beacons about its location, speed, and direction every $100 \sim 300$ msec. Also, TACK requires the RAs to completely cover the network, otherwise, the TACK technique may not function properly. This requirement may not be feasible especially in the early deployment stages of VANETs. Although TACK eliminates the CRL at the vehicles level, it requires the RAs to verify the revocation status of the vehicles upon requesting new certificates. To check the revocation status of a vehicle, the RA has to verify that this vehicle is not in the current Revocation List (RL) by preforming a check against all the entries in the RL. Each check requires three pairing operations. Consequently, checking the revocation status of a vehicle may be a time-consuming process. The authors suggested to use an optimized search method to remedy the computationally expensive RL check. The proposed method can reduce the RL checking to two pairing operations. However, this solution is based on fixing some parameters in the group signature attached to every certificate request, which reduces the privacy preservation of TACK and renders the tracking of a vehicle possible.

There are some works addressing the problem of distributing the large-size CRL in VANETs. In [12], Raya *et al.* introduce RC2RL (Revocation using Compressed Certificate Revocation Lists), where the traditional CRLs, issued by the TA, are compressed using Bloom filters to reduce its size prior to broadcasting. Papadimitratos *et al.* [13] propose to partition the CRL into small pieces and distribute each piece independently. Laberteaux *et al.* [14] use car to car communication to speed up the CRL broadcasting. Haas *et al.* [6] develop a mechanism to reduce the size of the broadcast CRL by only sending a secret key per revoked vehicle. On receiving the new CRL, each OBU uses the secret key of each revoked vehicle to re-produce the identities of the certificates loaded in that revoked vehicle, and construct the complete CRL. It should be noted that although the broadcast CRL size is reduced, the constructed CRL at each OBU, which is used to check the revocation status of other entities, still suffers from the expected large size exactly as that in the traditional CRLs where all the identities of the certificates of every revoked OBU are included in the broadcast CRL. Also, the authors propose using bloom filter, which is some kind of lookup hash tables, to perform CRL checking for the received certificates. To minimize the false-positives in the bloom filter, the authors proposed that each vehicle has to check before sending its certificate whether this certificate will trigger a false positive or no. If yes, then it uses another certificate. The authors proposed to upload each vehicle with additional certificates to compensate for those ones which will trigger a false-positive. Although this solution can minimize the false positives, it cannot to completely prevent them, which limits their advantages, especially, in safety-related VANETs applications.

The probabilistic approach is a promising technique for the key management in ad hoc networks [15], [16]. Zhu *et al.* introduce the GKMPAN protocol [17], which adopts a probabilistic key distribution approach based on pre-deployed symmetric keys. The GKMPAN is efficient and scalable for wireless mobile networks, because it takes the node mobility into consideration. In [18], a probabilistic random key distribution is proposed to achieve efficient privacy-preserving group communication protocol for VANETs. Employing a probabilistic random key distribution and a secret key sharing threshold scheme, an efficient distributed revocation protocol for VANET is designed in [19].

In this paper, we propose an Expedite Message Authentication Protocol (EMAP) to overcome the problem of the long delay incurred in checking the revocation status of a certificate using a CRL. EMAP employs keyed Hash Message Authentication Code $(HMAC)$ in the revocation checking process, where the key used in calculating the $HMAC$ for each message is shared only between unrevoked OBUs. In addition, EMAP is free from the false positive property which is common for lookup hash tables as it will be indicated in the next section.

III. Preliminaries

In this section, we introduce the bilinear pairing, hash chains, and search algorithms that can be employed for checking a CRL.

A. Bilinear Pairing

The bilinear pairing [20] is one of the foundations of the proposed protocol. Let $\mathbb{G}_1$ denote an additive group of prime order q, and $\mathbb{G}_2$ a multiplicative group of the same order. Let P be a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear mapping with the following properties:

1) Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, for all $P, Q \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q$.
2) Non-degeneracy: $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3) Symmetric: $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in \mathbb{G}_1$.
4) Admissible: the map $\hat{e}$ is efficiently computable.

The bilinear map $\hat{e}$ can be implemented using the Weil [21] and Tate [22] pairings on elliptic curves.

The security of the proposed protocol depends on solving the following hard computational problem:

- **Elliptic Curve Discrete Logarithm Problem (ECDLP):** Given a point P of order q on an elliptic curve, and a point Q on the same curve. The ECDLP problem [23] is to determine the integer l, $0 \leq l \leq q-1$, such that $Q = lP$.

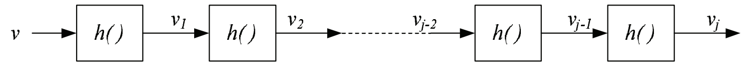


Fig. 1. Hash chain

B. Hash Chains

A hash chain [24] is the successive application of a hash function $h : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ with a secret value as its input. A hash function is easy and efficient to compute, but it is computationally infeasible to invert. Fig. 1 shows the application of a hash chain to a secret value v, where $v_0 = v, v_i = h(v_{i-1}) \; \forall \; 1 \leq i \leq j$.

C. Search Algorithms

The WAVE standard does not consider a specific mechanism for searching CRLs to check the revocation status of certificates. The most common search algorithms [25] include non-optimized search algorithms such as linear search algorithm, and optimized search algorithms such as binary search algorithm and lookup hash tables. The basic concept of each algorithm is as follows.

1) Linear Search Algorithm: In the linear search algorithm, the revocation status of a certificate is checked by comparing the certificate with each entry in the CRL. If a match occurs, the certificate is revoked and vice versa.

2) Binary Search Algorithm: The binary search algorithm works only on sorted lists. Consequently, upon receiving a new CRL, each OBU has to maintain a sorted (with respect to the certificate's identity) database of the revoked certificates included in previous CRLs and the recently received CRL. The main idea of the binary search algorithm is to cancel out half of the entries under consideration after each comparison in the search process. In the binary search, the revocation status of a certificate is checked by comparing the identity of the certificate with middle value (which in this case will be the median value) of the sorted database. If the identity of the certificate is greater than the median value, the right half of the database will be considered in the next comparison process and vice versa. This process continues until a match is found, i.e., the certificate is revoked, or the process is finished without finding a match which means that the certificate is unrevoked.

3) Lookup Hash Tables: In this approach, the set of all possible certificates $(\mathcal{U})$ is mapped using a hash function into a table of n entries. To check the revocation status of a certificate, the hash of the certificate's identity is the index of the entry in the lookup table which should be checked to determine the revocation status of the certificate. If nil is found in that entry, the certificate under consideration is unrevoked and vice versa. Since VANETs scale is very large and each OBU has a set of certificates, the size of $\mathcal{U}$ will be huge compared to the size (n) of the lookup table. Consequently, the probability of hash collisions will be high, which directly translates to a high probability of false positives. Here, a false positive means that the certificate of an innocent OBU is falsely considered revoked which results in rejecting all the messages containing the certificate of that OBU. The rejected messages may include a warning from dangerous situations. Hence, rejecting these messages may deprive the recipient OBU from taking the appropriate countermeasures to ensure its safety. Accordingly, lookup hash tables may not be practical for VANETs. Hence, lookup hash tables will not be considered in this paper. It should be noted that hash functions which map an input to one entry of possible n entries used in the lookup tables, are different from cryptographic hash functions which map an input to a unique output. Throughout the rest of the paper, the considered hash functions are cryptographic hash functions.

IV. Expedite Message Authentication Protocol

The proposed EMAP uses a fast $HMAC$ function and novel key sharing scheme employing probabilistic random key distribution.

A. System Model

As shown in Fig. 2, the system model under consideration consists of the followings.

- A Trusted Authority (TA), which is responsible for providing anonymous certificates and distributing secret keys to all OBUs in the network;
- Roadside units (RSUs), which are fixed units distributed all over the network. The RSUs can communicate securely with the TA;
- On-Board Units (OBUs), which are embedded in vehicles. OBUs can communicate either with other OBUs through V2V communications or with RSUs through V2I communications.

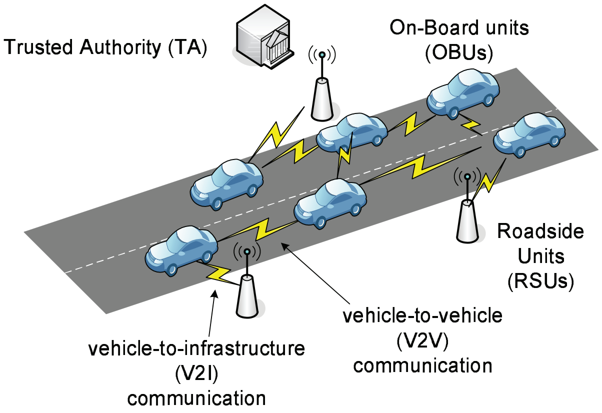

Trusted Authority (TA)
On-Board units (OBUs)
Roadside Units (RSUs)
vehicle-to-vehicle (V2V) communication
vehicle-to-infrastructure (V2I) communication

Fig. 2. The system model

According to the WAVE standard [7], each OBU is equipped with a Hardware Security Module (HSM), which is a tamper-resistant module used to store the security materials, e.g., secret keys, certificates, etc., of the OBU. Also, the HSM in each OBU is responsible for performing all the cryptographic operations such as signing messages, verifying certificates, keys updating, etc. We consider that legitimate OBUs cannot collude with the revoked OBUs as it is difficult for legitimate OBUs to extract their security materials from their HSMs. Finally, we consider that a compromised OBU is instantly detected by the TA.

B. System Initialization

The TA initializes the system by executing Algorithm 1. In step (20), it should be noted that: PK_u^i denotes the i^{th} public key for OBU_u, where the corresponding secret key is SK_u^i; PID_u^i denotes the i^{th} pseudo identity for OBU_u, where the TA is the only entity that can relate PID_u^i to the real identity of OBU_u; $sig_{TA}(PID_u^i||PK_u^i)$ denotes the TA signature on the concatenation $(||)$ of PID_u^i and PK_u^i; and C is the number of certificates loaded in each OBU.

After the system is initialized, the TA has the followings:

- A secret key pool $U_s = \{K_i^- = k_i Q | 1 \leq i \leq l\}$;
- The corresponding public key set $U_p = \{K_i^+ = \frac{1}{k_i} P | 1 \leq i \leq l\}$;
- A master secret key s and the corresponding public key $P_\circ$;
- The secret key K_g;

Algorithm 1 System initialization

```
Select two generators $P,\ Q \in \mathbb{G}_1$ of order $q$,
for $i \leftarrow 1, l$ do
    Select a random number $k_i \in \mathbb{Z}_q^*$
    Set the secret key $K_i^- = k_i Q \in \mathbb{G}_1$
    Set the corresponding public key $K_i^+ = \frac{1}{k_i} P \in \mathbb{G}_1$
end for
Select an initial secret key $K_g \in \mathbb{G}_2$        ▷ to be shared between all the non-revoked OBUs
Select a master secret key $s \in \mathbb{Z}_q^*$
Set the corresponding public key $P_\circ = sP$
Choose hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$
Select a secret value $v \in \mathbb{Z}_q^*$ and set $v_\circ = v$
for $i \leftarrow 1, j$ do                          ▷ to obtain a set $V$ of hash chain values
    Set $v_i = h(v_{i-1})$
end for
for all $OBU_u$ in the network, TA do
    for $i \leftarrow 1, m$ do
        Select a random number $a \in [1, l]$
        Upload the secret key $K_a^- = k_a Q$ and the corresponding public key $K_a^+ = \frac{1}{k_a} P$ in
        $HSM_u$ which is the $HSM$ embedded in $OBU_u$
    end for
    Generate a set of anonymous certificates $CERT_u = \{cert_u^i(PID_u^i,$
    $PK_u^i, sig_{TA}(PID_u^i || PK_u^i)) | 1 \leq i \leq C\}$        ▷ for privacy-preserving authentication
    Upload $CERT_u$ in $HSM_u$ of $OBU_u$
end for
Announce $H,\ h,\ P,\ Q,$ and $P_\circ$ to all the OBUs
```

- A set of hash chain values $V = \{v_i | 0 \leq i \leq j\}$, where j is large enough to accommodate with the number of revocation processes occur during the life-time of the network;
- The public parameters H, h, P, and Q.

Also, each OBU will have the followings:

- A set of anonymous certificates ($CERT_u$) used to achieve privacy-preserving authentication;
- A set of secret keys RS_u consisting of m keys randomly selected from U_s, i.e., $RS_u \subset U_s$;
- The set of the public keys RP_u corresponding to the keys in RS_u, i.e., $RP_u \subset U_p$;
- The secret key K_g, which is shared between all the legitimate OBUs;
- The hash function H, h, P, Q, and the public key $P_\circ$.

Note that the system model under consideration is mainly a PKI system, where each OBU_u has a set of anonymous certificates $(CERT_u)$ used to secure its communications with other entities in the network. In specific, the public key PK_u, included in the certificate $cert_u$, and the secret key SK_u are used for verifying and signing messages, respectively. Also, each OBU_u is pre-loaded with a set of asymmetric keys (secret keys K^-'s in RS_u and the corresponding public keys K^+'s in RP_u). Those keys are necessary for generating and maintaining a shared secret key K_g between unrevoked OBUs.

C. Message Authentication

Since we adopt a generic PKI system, the details of the TA signature on a certificate and an OBU signature on a message are not discussed in this paper for the sake of generality. We only focus in how to accelerate the revocation checking process, which is conventionally performed by checking the CRL for every received certificate. The message signing and verification between different entities in the network are performed as follows.

1) Message Signing: Before any OBU_u broadcasts a message $\mathcal{M}$, it calculates its revocation check REV_{check} as $REV_{check} = HMAC(K_g, PID_u || T_{stamp})^*$, where T_{stamp} is the current time stamp, and $HMAC(K_g, PID_u || T_{stamp})$ is the hash message authentication code on the concatenation of PID_u and T_{stamp} using the secret key K_g. Then, OBU_u broadcasts $(\mathcal{M} || T_{stamp} || cert_u(PID_u, PK_u, sig_{TA}(PID_u || PK_u)) || sig_u(\mathcal{M} || T_{stamp}) || REV_{check})$, where $sig_u(\mathcal{M} || T_{stamp})$ is the signature of OBU_u on the concatenation of the message $\mathcal{M}$ and T_{stamp}.

*It should be noted that throughout the rest of the paper the superscript i will be removed from PID_u^i and PK_u^i for the ease of presentation.

2) Message Verification: Any OBU_y receiving the message $(\mathcal{M}||T_{stamp}||cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})||REV_{check})$ can verify it by executing Algorithm 2.

Algorithm 2 Message verification

```
Require: $(\mathcal{M}||T_{stamp}||cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})|| REV_{check})$
    and $K_g$
 1: Check the validity of $T_{stamp}$
 2: if invalid then
 3:     Drop the message
 4: else
 5:     Check $REV_{check} \overset{?}{=} HMAC(K_g, PID_u||T_{stamp})$
 6:     if invalid then
 7:         Drop the message
 8:     else
 9:         Verify the TA signature on $cert_{OBU_u}$
10:         if invalid then
11:             Drop the message
12:         else
13:             Verify the signature $sig_u(\mathcal{M}||T_{stamp})$ using $OBU_u$ public key $(PK_u)$
14:             if invalid then
15:                 Drop the message
16:             else
17:                 Process the message
18:             end if
19:         end if
20:     end if
21: end if
```

In step (5), OBU_y calculates $HMAC(K_g, PID_u||T_{stamp})$ using its K_g on the concatenation $PID_u||T_{stamp}$, and compares the calculated $HMAC(K_g, PID_u||T_{stamp})$ with the received REV_{check}.

D. Revocation

The revocation is triggered by the TA when there is an OBU_u to be revoked. The certificates of OBU_u must be revoked. In addition, the secret key set RS_u of OBU_u and the current secret key K_g are considered revoked. Hence, a new secret key $\tilde{K_g}$ should be securely distributed to all the non-revoked OBUs. Also, each non-revoked OBU should securely update the compromised keys in its key sets RS and RP [17]. The revocation process is as follows.

1) The TA searches its database to determine the identity (M) of the non-compromised secret key $K_M^- = k_M Q$ that is shared by the majority of the non-revoked OBUs, and finds the corresponding public key $K_M^+ = \frac{1}{k_M}P$. The TA then selects a random number $t \in \mathbb{Z}_q^*$, and calculates the intermediate key $K_{im} = tK_M^+ = \frac{t}{k_M}P \in \mathbb{G}_1$, and the new secret key $\tilde{K_g}$ as follows

$$\begin{aligned} \tilde{K_g} &= \hat{e}(K_M^-, K_{im}) \\ &= \hat{e}(k_M Q, \frac{t}{k_M}P) \\ &= \hat{e}(Q,P)^{k_M \cdot \frac{t}{k_M}} \\ &= \hat{e}(Q,P)^t \end{aligned} \tag{1}$$

   Also, it selects the value v_{j-ver} of the hash chain values, where v_j is the last value in the hash chain as shown in Fig. 1, and ver is an integer indicating the revocation version, i.e., the number of the revocation processes performed since the network initialization. The value v_{j-ver} is used by all the OBUs to update their compromised secret keys and the corresponding public keys. After that, the TA prepares a key update message $Kmsg = (ver||M||IDrev_{key}||K_{im}||enc_{\tilde{K_g}}(v_{j-ver}))$, where $IDrev_{key}$ is a list of the identities of the revoked keys, and $enc_{\tilde{K_g}}(v_{j-ver})$ is the symmetric encryption of v_{j-ver} using the key $\tilde{K_g}$. Finally, the TA broadcasts the following message $REV_{msg} = (CRL||Kmsg||sig_{TA}(CRL||Kmsg))$, where CRL is a list of the certificates of the revoked OBUs, and $sig_{TA}(CRL||Kmsg) = sH(CRL||Kmsg)$ is the TA signature on $CRL||Kmsg$;
2) After receiving the message REV_{msg}, each OBU_y executes Algorithm 3;
3) In Algorithm 3 step (1), OBU_y verifies the signature $sig_{TA}(CRL||Kmsg)$ by checking that

$\hat{e}(sig_{TA}(CRL||Kmsg), P) \overset{?}{=} \hat{e}(H(CRL||Kmsg), P_{\circ})$. This check follows since

$$\begin{aligned}\hat{e}(sig_{TA}(CRL||Kmsg), P) &= \hat{e}(sH(CRL||Kmsg), P)\\ &= \hat{e}(H(CRL||Kmsg), sP)\\ &= \hat{e}(H(CRL||Kmsg), P_{\circ})\end{aligned}$$

Algorithm 3 Processing revocation messages

```
Require: $REV_{msg} = (CRL||Kmsg||sig_{TA}(CRL||Kmsg))$ and $P_{\circ}$
Verify $sig_{TA}(CRL||Kmsg)$ by checking $\hat{e}(sig_{TA}(CRL||Kmsg), P) \overset{?}{=} \hat{e}(H(CRL||Kmsg), P_{\circ})$
if invalid then
    Exit
else
    Run Algorithm 4 to get $\tilde{K_g}$ and $v_{j-ver}$
    Run Algorithm 5 to update the key set of $OBU_y$
end if
Store $ver$ and $IDrev_{key}$
Erase $K_{im}$, the hash chain values, and the original compromised secret and public keys.
```

Algorithm 4 Obtaining $\tilde{K_g}$ and v_{j-ver}

```
1:  if $K_M^-$ exists in $RS_y$ then
2:      Set the new secret key $\tilde{K_g} = \hat{e}(K_M^-, K_{im})$
3:      Decrypt $enc_{\tilde{K_g}}(v_{j-ver})$ using $\tilde{K_g}$ to get $v_{j-ver}$
4:  else
5:      Broadcast a signed request and $cert_y(PID_y, PK_y, sig_{TA}(PID_y||PK_y))$ to get $\tilde{K_g}$ from
        neighboring OBUs
6:      Start a timer $T_1$
7:      Any neighboring OBU of $OBU_y$ having $\tilde{K_g}$ verifies the signature and certificate of
        $OBU_y$, ensures that $cert_y$ is not in the recent CRL, uses the public key $(PK_y)$ of $OBU_y$
        included in $cert_y$ to encrypt $\tilde{K_g}$, and sends the encrypted $\tilde{K_g}$ to $OBU_y$
8:      if the encrypted $\tilde{K_g}$ is received then
9:          Decrypt $\tilde{K_g}$ using the secret key corresponding to $PK_y$
10:         Decrypt $enc_{\tilde{K_g}}(v_{j-ver})$ using $\tilde{K_g}$ to get $v_{j-ver}$
11:     else
12:         if $T_1$ is timed out then
13:             Go to 5
14:         end if
15:     end if
16: end if
```

Algorithm 5 Updating the key sets of OBU_y

```
Require: $\tilde{K_g}$ and $v_{j-ver}$
if not previously missing any revocation message then
    if possesses compromised secret keys $\{K_i^-\} = \{k_i Q\}$ in $IDrev_{key}$ then
        Update the secret key $K_i^-$ as $\tilde{K_i^-} = v_{j-ver} K_i^- = v_{j-ver} k_i Q$
        Update the corresponding pubic keys $\tilde{K_i^+} = \frac{1}{v_{j-ver}} K_i^+ = \frac{1}{v_{j-ver} k_i} P$
    else
        Exit
    end if
else
    Set $n = ver$
    while $n \neq v_{ver_{last}}$ do        ▷ $ver_{last}$ is the last received revocation version
        Set $v_{j-n+1} = h(v_{j-n})$
        Set $n = ver + 1$
    end while        ▷ this loop outputs $\{v_{j-ver+1}, v_{j-ver+2}, \cdots, v_{ver_{last}-1}\}$
    Broadcast a signed request to the neighboring OBUs requesting $ver_{|_{missed}}$ and
      $IDrev_{key|_{missed}}$ for all the missed revocation processes
    for each received signed value of $ver_{|_{missed}}$ do
        Verify the signature and certificate of the sender and, ensures that the certificate of
        the sender is not in the recent CRL
        Find the value of $v_{j-ver_{|_{missed}}}$ from $\{v_{j-ver+1}, v_{j-ver+2}, \cdots, v_{ver_{last}-1}\}$
        for each possessed key $K_i^- = k_i Q \in IDrev_{key|_{missed}}$ do
            Update the secret key $K_i^-$ as $\tilde{K_i^-} = v_{j-ver_{|_{missed}}} K_i^- = v_{j-ver_{|_{missed}}} k_i Q$
            Update the corresponding public key as $\tilde{K_i^+} = \frac{1}{v_{j-ver_{|_{missed}}}} K_i^+ = \frac{1}{v_{j-ver_{|_{missed}}} k_i} P$
        end for
    end for
end if
```

4) OBU_y has to execute Algorithm 4 to get $\tilde{K_g}$ and v_{j-ver}. If OBU_y has K_M^-, it can independently calculate $\tilde{K_g}$ according to step (2). Otherwise, OBU_y gets $\tilde{K_g}$ from its neighboring OBUs as indicated in steps (5-15);

5) In Algorithm 4, the revoked OBUs cannot compute $\tilde{K}_g$ since they do not have K_M^-. Also, they cannot receive $\tilde{K}_g$ from other OBUs since the recent CRL sent in REV_{msg} contains the certificates of the revoked OBUs, which stops others from forwarding $\tilde{K}_g$ to them;
6) OBU_y has to execute Algorithm 5 to update its key sets RS_y and RP_y. If OBU_y does not miss any previous revocation messages, it updates its key sets as indicated in steps (3-4). If OBU_y missed a number of previous revocation messages, it can update its key sets as indicated in steps (9-22). It should be noted that in step (14), $ver|_{missed}$ and $IDrev_{key}|_{missed}$ denote the revocation version and the list of identities of the revoked keys of a missed revocation process, respectively;
7) It should be noted that in Algorithm 4 step (7) and in Algorithm 5 step (16) one of the communicating parties do not have the new key $\tilde{K}_g$. Accordingly, the OBUs must use the CRL to check that the certificates of the communicating parties are not previously revoked.

Remarks

- An important feature of the proposed EMAP is that it enables an OBU to update its compromised keys corresponding to previously missed revocation processes provided that it picks one revocation process in the future. To the best of our knowledge, this is the first work to propose a rekeying mechanism capable of updating compromised keys corresponding to previously missed rekeying processes.
- Note that EMAP has a modular feature, which makes it integrable with any PKI system. In other words, EMAP does not require any modification to the core of the PKI architecture. It only needs a key distribution module to be added to the TA during the system initialization.
- EMAP is suitable for not only VANETs but also any type of networks employing PKI.
- Algorithms 3-5 are executed through the HSM module in each OBU.

1) Renewing the Hash Chain Values: The values of the hash chains are continuously used in the revocation processes, and hence, the TA can consume all the hash chain values. As a result, there should be a mechanism to replace the current hash chain with a new one as follows. After using the last value $v_\circ$ in the current hash chain, the TA generates a new hash chain $\tilde{V} = \{\tilde{v}_i | 0 \leq i \leq j\}$. In the upcoming revocation messages where the new hash chain values will be used, the TA will always broadcast the last value of the old hash chain $v_\circ$ and the current value $\tilde{v}_{j-ver}$ of the new hash chain. Having the last value of the old hash chain $v_\circ$ and the current value $\tilde{v}_{j-ver}$ of the new hash chain, any OBU missed revocation messages corresponding to some values of the old hash chain and some values in the new hash chain can regenerate all the values of the old hash chain and the values of the new hash chain up to $\tilde{v}_{j-ver}$ and consequently, that OBU can update its compromised keys as indicated in the previous subsection.

V. SECURITY ANALYSIS

In this section, we analyze the security of the proposed protocol against some common attacks.

2) Resistance to forging attacks: To forge the revocation check $REV_{check} = HMAC(K_g, PID_u|| T_{stamp})$ of any OBU_u, an attacker has to find the current K_g, which is equivalent to finding t in the following ECDLP problem: given $K_{im} = tK_M^+ = \frac{t}{k_M}P$ and $K_M^+ = \frac{1}{k_M}P$, find t such that $K_{im} = tK_M^+$. Similar analogy applies to finding the TA secret key s from the TA message signature $sgn_{Kmsg} = sH(Kmsg)$. Since ECDLP is a hard computational problem [23], i.e., it cannot be solved in a sub-exponential time, the revocation check and the TA message signature sgn_{Kmsg} are unforgeable. Similarly, finding the TA secret value s from $P_\circ = sP$ is ECDLP problem, which makes it unforgeable. From the aforementioned discussion, it is concluded that EMAP is resistant to forging attacks.

3) Forward secrecy: Since the values of the hash chain included in the revocation messages are released to non-revoked OBUs starting from the last value of the hash chain, and given the fact that a hash function is irreversible, a revoked OBU cannot use a hash chain value $v_{j-ver+1}$ received in a previous revocation process to get the current hash chain value v_{j-ver}. Consequently, a revoked OBU cannot update its secret key set (RS). Accordingly, a revoked OBU can neither get K_M^- necessary to independently calculate the new secret key $\tilde{K_g}$ nor get $\tilde{K_g}$ from the neighboring OBUs since the certificates of the revoked OBUs are in the up-to-date CRL which prevents unrevoked OBUs from forwarding $\tilde{K_g}$ to the revoked OBUs. As a result, the proposed EMAP guarantees forward secrecy.

4) Resistance to replay attacks: Since in each message an OBU includes the current time stamp in the revocation check value $REV_{check} = HMAC(K_g, PID_u||T_{stamp})$, an attacker cannot record REV_{check} at time T_i and replay it at a later time T_{i+1} to pass the revocation checking process as the receiving OBU compares the current time T_{i+1} with that included in the revocation check. Consequently, EMAP is secure against replay attacks.

5) Resistance to colluding attacks: For a colluding attack, a legitimate OBU colludes with a revoked OBU by releasing the current secret key $\tilde{K_g}$ such that the revoked vehicle can use this key to pass the revocation check process by calculating the correct HMAC values for the transmitted messages. All the security materials of an OBU are stored in its tamper-resistant HSM. In addition, all the keys update processes in Algorithms 3-5 are executed in the HSM, which means that the new secret key $\tilde{K_g}$ is stored in the HSM, and it cannot be transmitted in clear under any circumstances. Note that in Algorithm 4 step (7) the HSM only sends $\tilde{K_g}$ encrypted with the public key included in the certificate of the OBU requesting $\tilde{K_g}$ after checking that the certificate of that OBU is not in the CRL. Accordingly, only that OBU is the entity that can decrypt and obtain $\tilde{K_g}$ using its secret key which is exclusively known to itself. Since it is infeasible to extract the security materials from the tamper-resistant HSM, an unrevoked OBU cannot collude with a revoked OBU by passing the new secret key $\tilde{K_g}$ to the revoked OBU. Hence, EMAP is secure against colluding attacks.

VI. Performance Evaluation

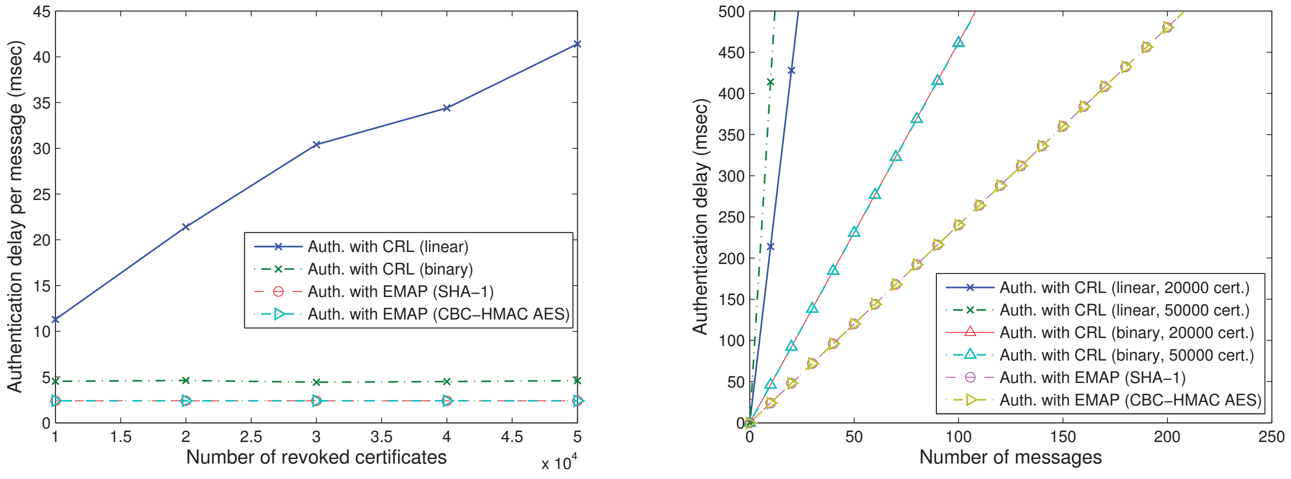
A. Computation Complexity of Revocation Status Checking

We are interested in the computation complexity of the revocation status checking process which is defined as the number of comparison operations required to check the revocation status of an OBU. Let N_{rev} denote the total number of revoked certificates in a CRL. To check the revocation status of an OBU_u using the linear search algorithm, an entity has to compare the certificate identity of OBU_u with every certificate of the N_{rev} certificates in the CRL, i.e., the entity performs one-to-one checking process. Consequently, the computation complexity of employing the linear search algorithm to perform a revocation status checking for an OBU is $O(N_{rev})$. In the binary search algorithm, the certificate identity of OBU_u is compared to the certificate identity in the middle of the sorted CRL. If the certificate identity of OBU_u is greater than that of the entry in the middle, then half of the CRL with identities lower than that of OBU_u are discarded from the upcoming comparisons. If the certificate identity of OBU_u is lower than that of the entry in the middle, then half of the CRL with identities higher than that of OBU_u are discarded. The checking process is repeated until a match is found or the CRL is finished. It can be seen that at each step in the binary search method half of the entries considered in the search is discarded. Thus, the computation complexity of the binary search algorithm to perform a revocation status checking for an OBU is $O(\log N_{rev})$ [25]. In EMAP, the revocation checking process requires only one comparison between the calculated and received values of REV_{check}. As a result, the computation complexity of EMAP is $O(1)$, which is constant and independent of the number of revoked certificates. In other words, EMAP has the lowest computation complexity compared with the CRL checking processes employing linear and binary search algorithms.

B. Authentication Delay

We compare the message authentication delay employing the CRL with that employing EMAP to check the revocation status of an OBU. As stated earlier, the authentication of any message is performed by three consecutive phases: checking the sender's revocation status, verifying the sender's certificate, and verifying the sender's signature. For the first authentication phase which checks the revocation status of the sender, we employ either the CRL or EMAP. For EMAP, we adopt the Cipher Block Chaining Advanced Encryption Standard (CBC-HMAC AES) [26] and Secure Hash Algorithm 1 SHA-1 [27] as the $HMAC$ functions. We consider the pseudo identity (PID) of OBU and the time stamp (T_{stamp}) having equal lengths of 8 bytes. We adopt the Crypto++ library [28] for calculating the delay of the $HMAC$ functions, where it is compiled on Intel Core2Duo 2 GHz machine. The delay incurred by using CBC-HMAC AES and SHA-1 to calculate the revocation check $(REV_{check} = HMAC(K_g, PID_u||T_{stamp}))$ is 0.23 μsec and 0.42 μsec, respectively. Also, we have simulated the linear and binary CRL checking process using C++ programs compiled on the same machine. The linear CRL checking program performs progressive search on a text file containing the unsorted identities of the revoked certificates, while the binary CRL checking program performs a binary search on a text file containing the sorted identities of the revoked certificates. For the second and third authentication phases, we employ Elliptic Curve Digital Signature Algorithm (ECDSA) [29] to check the authenticity of the certificate and the signature of the sender. ECDSA is the digital signature method chosen by the WAVE standard. In ECDSA, a signature verification takes $2T_{mul}$, where T_{mul} denotes the time required to perform a point multiplication on an elliptic curve. Consequently, the verification of a certificate and message signature takes $4T_{mul}$. In [30], T_{mul} is found for a supersingular curve with embedding degree $k = 6$ to be equal to 0.6 $msec$.

Fig. 3(a) shows a comparison between the authentication delay per message using EMAP, linear CRL checking process, and binary CRL checking process vs. the number of the revoked



(a) Authentication delay per message

(b) Total authentication delay vs. the number of the received messages

Fig. 3. Authentication delay

certificates, where the number of the revoked certificates is an indication of the CRL size. It can be seen that the authentication delay using the linear CRL checking process increases with the number of revoked certificates, i.e., with the size of the CRL. Also, the authentication delay using the binary CRL checking process is almost constant. This can be explained as follows: the number of revoked certificates in the conducted simulation ranges from 10000 to 50000 revoked certificates; This is respectively corresponding to 14 to 16 comparison operations. Since the range of the number of the comparison operations is very small, the authentication delay is almost constant. The authentication delay using EMAP is constant and independent of the number of revoked certificates. Moreover, the authentication delay using the EMAP outperforms that using the linear and binary CRL checking processes. For example, the authentication delay per message using the linear CRL checking process, the binary CRL checking process, and EMAP (SHA-1) for a CRL including 20000 revoked certificates are 21.4 $msec$, 4.62 $msec$, and 2.4004 $msec$, respectively. Consequently, EMAP (SHA-1) expedites the message authentication by 88.78% and 48.04% compared to that using the linear and binary CRL checking processes, respectively. Fig. 3(b) shows the total authentication delay in $msec$ vs. the number of messages to be authenticated using EMAP and the linear and binary CRL checking processes. It can be seen that as the CRL size increases the number of messages that can be verified within a specific

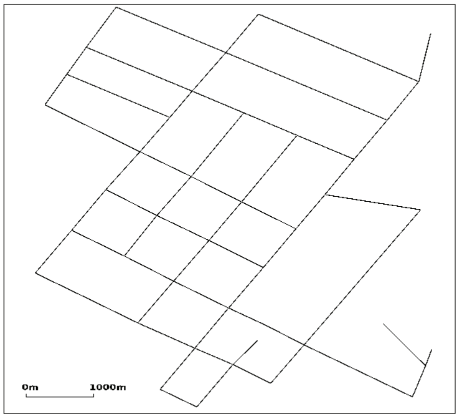


Fig. 4. A city street simulation scenario

TABLE I

NS-2 SIMULATION PARAMETERS

Simulation area	$7.4\ Km \times 7.4\ Km$
Simulation time	30 sec
Max. OBU speed	60 Km/h
OBU transmission range	300 m
OBU information dissemination interval	300 $msec$
MAC protocol	802.11a
Wireless channel capacity	6 Mbps

period is significantly decreased using the linear CRL checking process. Also, for a constant authentication delay, EMAP outperforms the linear and binary CRL checking processes. The maximum number of messages that can be verified simultaneously in $300\ msec$ is $14,\ 64,$ and 124 messages for message authentication employing linear CRL checking, binary CRL checking, and EMAP, respectively, where the considered CRL includes $20,000$ certificates. The number of messages that can be verified using EMAP within $300\ msec$ is greater than that using linear and binary CRL checking by 88.7% and 48.38%, respectively.

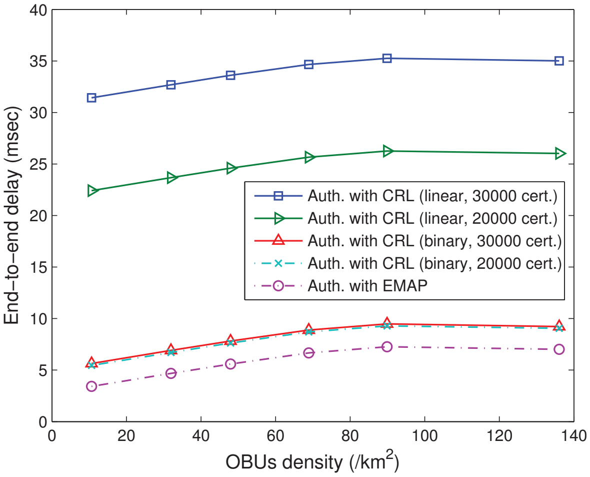


Fig. 5. End-to-end delay vs. OBUs density

C. *End-to-end delay*

To further evaluate EMAP, we have conducted ns-2 [31] simulation for the city street scenario shown in Fig. 4. The adopted simulation parameters are given in Table I. We select the dissemination of the road condition information by an OBU every 300 $msec$ to conform with the DSRC standards. The mobility traces adopted in this simulation are generated using TraNS [32]. We are interested in the end-to-end delay, which is defined as the time to transmit a message from the sender to the receiver. Fig. 5 shows the end-to-end delay in $msec$ vs. the OBUs density, by employing authentication using the proposed EMAP (SHA-1), the linear CRL checking, and binary CRL checking, respectively. In the simulation, we consider CRLs containing 20000 and 30000 revoked certificates, respectively, and the OBUs density as the number of OBUs per km^2. It can be seen that the end-to-end delay increases with the OBUs density because the number of the received packets increases with the OBUs density resulting in longer waiting time for the packets to be processed by the application layer in each OBU. In addition, the end-to-end delay tends to be constant for high OBUs densities as the number of received packets reaches the maximum number of packets an OBU can verify within a specific duration. The end-to-end delay also increases with the number of revoked certificates included in the CRL for the linear CRL checking process. However, the end-to-end delay is almost constant with the CRL size using the binary checking process as the number of comparison operations needed to check CRLs with

20000 and 30000 certificates is almost the same. From Fig. 5, employing the proposed EMAP in authentication reduces the end-to-end delay compared with that using either the linear or the binary CRL checking process.

D. Message Loss Ratio

The average message loss ratio is defined as the average ratio between the number of messages dropped every 300 $msec$, due to the message authentication delay, and the total number of messages received every 300 $msec$ by an OBU. It should be noted that we are only interested in the message loss incurred by OBUs due to V2V communications. According to DSRC, each OBU has to disseminate a message containing information about the road condition every 300 $msec$. In order to react properly and instantly to the varying road conditions, each OBU should verify the messages received during the last 300 $msec$ before disseminating a new message about the road condition. Therefore, we chose to measure the message loss ratio every 300 $msec$. Fig. 6 shows the analytical and simulated average message loss ratio vs. the average number of OBUs within the communication range of each OBU for message authentication employing CRL linear checking, CRL binary checking, and EMAP, respectively, for a CRL containing $20,000$ certificates. It can be seen that the simulated average message loss ratio closely follows the analytical message loss ratio which is calculated based on the maximum number of messages that can be authenticated within 300 $msec$. The difference between the analytical and simulations results stems from observing that some zones in the simulated area become more congested than other zones, thus, some OBUs experience higher message loss than other OBUs, which leads to that difference between the analytical and simulations results. It can also be seen that the message loss ratio increases with the number of OBUs within communication range for all the protocols under considerations. In addition, the message authentication employing EMAP significantly decreases the message loss ratio compared to that employing either the linear or binary CRL revocation status checking. The reason of the superiority of EMAP is that it incurs the minimum revocation status checking delay compared to the linear and binary CRL revocation checking processes.

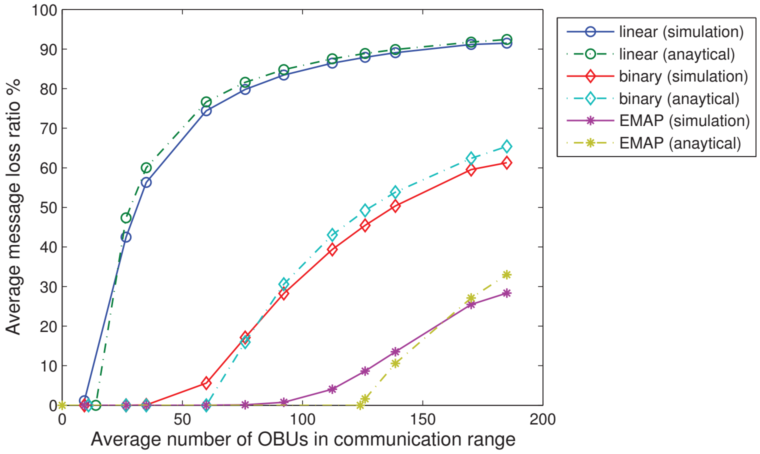


Fig. 6. Comparison between message loss ratio for different schemes

E. Communication Overhead

In EMAP, each OBU_u broadcasts a signed message on the form $(\mathcal{M}||T_{stamp}||\ cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})||REV_{check})$ to its neighboring OBUs. A signed message in the WAVE standard should include the certificate of the sender, a time stamp, and the signature of the sender on the transmitted message. Consequently, the additional communication overhead incurred in EMAP compared to that in the WAVE standard is mainly due to REV_{check}. The length of REV_{check} depends on the employed hash function. For example, when SHA-1 is employed in EMAP for calculating REV_{check}, this is corresponding to an additional overhead of 20 bytes [27]. The total overhead incurred in a signed message in the WAVE standard is 181 bytes [7]. Consequently, the total overhead in EMAP (SHA-1), assuming the same message format of the WAVE standard, is 201 bytes. In WAVE [7], the maximum payload data size in a signed message is 65.6 Kbytes. Accordingly, the ratio of the communication overhead in a signed message to the payload data size is 0.28% and 0.31% for the WAVE standard and EMAP, respectively. EMAP incurs 0.03% increase in the communication overhead compared to the WAVE standard, which is acceptable with respect to the gained benefits from EMAP.

F. Communication Cost of Updating the Secret Key (K_g)

We are interested in the communication cost of updating the secret key (K_g), which is the average number of messages an OBU has to transmit and receive after triggering the revocation

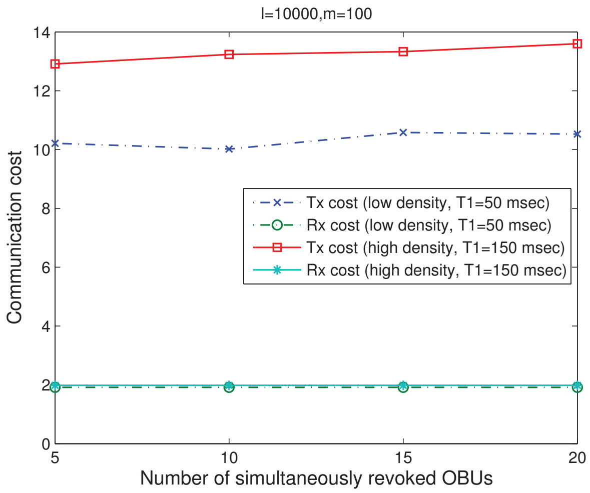


Fig. 7. Communication cost of updating K_g in EMAP

process to get the new secret key $(\tilde{K_g})$ and distribute $\tilde{K_g}$ to its unrevoked neighboring OBUs. We have conducted ns-2 simulation using the same parameters in Table I, for two scenarios: low and high OBUs densities corresponding to OBUs densities of $32.5\ /km^2$ and $91.5\ /km^2$, respectively. We consider the TA having a key pool of size $l = 10000$, and each OBU having a key set of size $m = 100$. In EMAP, an OBU_y not having K_M^- will send a request message to its neighboring OBUs to get the new secret key $(\tilde{K_g})$ and start timer T_1, where OBU_y will retry to get $\tilde{K_g}$ if T_1 is expired before getting $\tilde{K_g}$. In the conducted simulation, we set T_1 to be $50\ msec$ and $150\ msec$ for the low and high OBUs densities, respectively. Also, we only consider the case that an OBU without K_M^-, can get the new secret key $\tilde{K_g}$ from another OBU through a single hop.

Initially, the percentage of OBUs having the key K_M^- is 1.97% and 1.56% for the low and high OBUs densities, respectively. After the broadcast of the revocation message REV_{msg}, only the OBUs having K_M^- are able to independently calculate the new secret key $\tilde{K_g}$, and they will deliver $\tilde{K_g}$ to other OBUs through V2V communication. Fig. 7 shows the average communications cost vs. the number of simultaneously revoked OBUs. It can be seen that in each scenario, the communication cost (transmit or receive) is almost constant with respect to the number of simultaneously revoked OBUs. This is due to the fact that revoking the key sets of the revoked OBUs does not revoke the key K_M^- which is shared by the majority of OBUs. Consequently, the percentage of OBUs initially having K_M^- will not change. It can be seen that the communication cost is equal with respect to the number of received messages in both low and high OBUs densities. Also, the communication cost of the transmitted messages is higher than that of the received messages. This is due to the fact that a request broadcast by an OBU_y to get the new secret key $(\tilde{K_g})$ is received by all the neighboring OBUs, and each OBU of the neighboring OBUs will send $\tilde{K_g}$ to OBU_y. As a result, a number of OBUs, requesting $\tilde{K_g}$, in some geographic area, will cause all the neighboring OBUs to broadcast $\tilde{K_g}$ as many times as the number of OBUs requesting $\tilde{K_g}$ in that area. We have tried several values for the timer T_1, and the considered values for T_1 give the best results. If we select smaller T_1, the transmission cost will increase since each OBU not having K_M^- will send requests to get $\tilde{K_g}$ at a higher rate, and hence, more replies will be transmitted by the OBUs. It can also be seen that the low OBU density scenario incurs lower transmission communication cost than the high OBU density scenario since the number of OBUs in the low density scenario is lower than that in the high OBU density scenario.

G. *Incurred Delay to Obtain the New Secret Key* $(\tilde{K_g})$

We are interested in the average delay for an OBU without K_M^- to get the new secret key $\tilde{K_g}$ from its neighboring OBUs after the revocation message REV_{msg} is delivered to all the OBUs in the simulated area. We conducted ns-2 simulation for the low and high OBUs densities scenarios considered in the previous subsection. Initially, the percentage of OBUs having the key K_M^-, and capable of independently calculating $\tilde{K_g}$, is 1.97% and 1.56% for the low and high OBUs densities scenarios, respectively. Fig. 8 shows the average delay in $msec$, incurred by an OBU from the moment the revocation message REV_{msg} is received by all the OBUs in the simulated area until it gets the new secret key $\tilde{K_g}$, vs. the number of simultaneously revoked OBUs. It can be seen that the incurred delay to get $\tilde{K_g}$ is confined to a small range in each scenario. Also, the delay of obtaining $\tilde{K_g}$ in the high OBU density scenario is higher than that in the low OBU density scenario as the value of T_1 in the high OBU density scenario is higher than that in the low OBU density scenario. However, for both low and high OBU densities, the delay of getting $\tilde{K_g}$ is less than 1 sec, which indicates that EMAP is feasible and reliable.

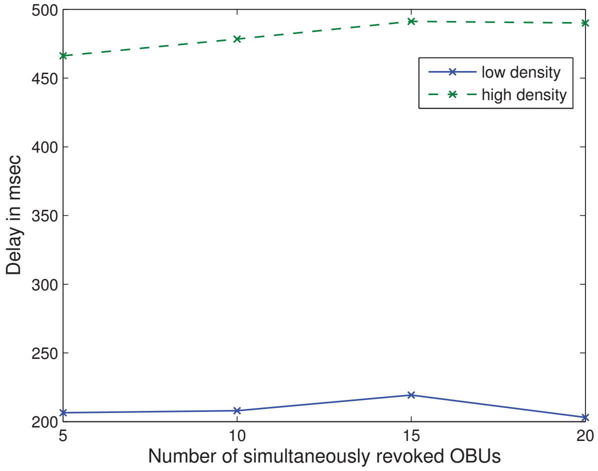


Fig. 8. Incurred delay to obtain $\tilde{K}_g$ in EMAP

VII. Conclusion

We have proposed EMAP for VANETs, which expedites message authentication by replacing the time-consuming CRL checking process with a fast revocation checking process employing $HMAC$ function. The proposed EMAP uses a novel key sharing mechanism which allows an OBU to update its compromised keys even if it previously missed some revocation messages. In addition, EMAP has a modular feature rendering it integrable with any PKI system. Furthermore, it is resistant to common attacks while outperforming the authentication techniques employing the conventional CRL. Therefore, EMAP can significantly decrease the message loss ratio due to message verification delay compared to the conventional authentication methods employing CRL checking. Our future work will focus on the certificate and message signature authentication acceleration.

Acknowledgment

The authors would like to thank Mr. Rongxing Lu for his valuable comments on this work.

References

[1] P. Papadimitratos, A. Kung, J. P. Hubaux, and F. Kargl, “Privacy and identity management for vehicular communication systems: a position paper,” *Proc. Workshop on Standards for Privacy in User-Centric Identity Management, Zurich, Switzerland*, July 2006.

[2] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, “CARAVAN: Providing location privacy for VANET,” *Proc. Embedded Security in Cars (ESCAR)*, November 2005.

[3] A. Wasef, Y. Jiang, and X. Shen, “DCS: An efficient distributed certificate service scheme for vehicular networks,” *IEEE Trans. on Vehicular Technology*, vol. 59, pp. 533–549, 2010.

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

[5] “US bureau of transit statistics.” [Online]. Available: http://en.wikipedia.org/wiki/Passenger_vehicles_in_the_United_States

[6] J. J. Haas, Y. Hu, and K. P. Laberteaux, “Design and analysis of a lightweight certificate revocation mechanism for VANET,” *Proc. 6th ACM international workshop on VehiculAr InterNETworking*, pp. 89–98, 2009.

[7] “IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages,” *IEEE Std 1609.2-2006*, 2006.

[8] “5.9 GHz DSRC.” [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[9] A. Wasef and X. Shen, “MAAC: Message authentication acceleration protocol for vehicular ad hoc networks,” *Proc. IEEE GLOBECOM’09*, 2009.

[10] J. P. Hubaux, “The security and privacy of smart vehicles,” *IEEE Security and Privacy*, vol. 2, pp. 49–55, 2004.

[11] A. Studer, E. Shi, F. Bai, and A. Perrig, “TACKing together efficient authentication, revocation, and privacy in VANETs,” *Proc. SECON ’09*, pp. 1–9, 2009.

[12] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, “Eviction of misbehaving and faulty nodes in vehicular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 25, pp. 1557–1568, 2007.

[13] P. P. Papadimitratos, G. Mezzour, and J. Hubaux, “Certificate revocation list distribution in vehicular communication systems,” *Proc. 5th ACM international workshop on VehiculAr Inter-NETworking*, pp. 86–87, 2008.

[14] K. P. Laberteaux, J. J. Haas, and Y. Hu, “Security certificate revocation list distribution for VANET,” *Proc. 5th ACM international workshop on VehiculAr Inter-NETworking*, pp. 88–89, 2008.

[15] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” *Proc. 2003 IEEE Symposium on Security and Privacy*, pp. 197–213, 2003.

[16] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” *Proc. ACM conference on Computer and communications security*, pp. 41–47, 2002.

[17] S. Zhu, S. Setia, S. Xu, and S. Jajodia, “GKMPAN: An efficient group rekeying scheme for secure multicast in ad-hoc networks,” *Journal of Computer Security*, vol. 14, pp. 301–325, 2006.

[18] A. Wasef and X. Shen, “PPGCV: Privacy preserving group communications protocol for vehicular ad hoc networks,” *Proc. ICC’08*, pp. 1458–1463, 2008.

[19] A. Wasef and X. Shen, “EDR: Efficient decentralized revocation protocol for vehicular ad hoc networks,” *IEEE Trans. on Vehicular Technology*, vol. 58, no. 9, pp. 5214 – 5224, 2009.

[20] D. Boneh and M. K. Franklin, “Identity-based encryption from the Weil pairing,” *Proc. 21st Annual International Cryptology Conference on Advances in Cryptology*, pp. 213–229, 2001.

[21] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil pairing,” *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[22] M. Scott, “Computing the Tate pairing,” *Topics in Cryptology, Springer*, pp. 293–304, 2005.

[23] N. Koblitz, A. Menezes, and S. Vanstone, “The state of elliptic curve cryptography,” *Designs, Codes and Cryptography*, vol. 19, no. 2, pp. 173–193, Mar. 2000.

[24] L. Lamport, “Password authentication with insecure communication,” *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, 2001.

[26] S. Frankel, R. Glenn, and S. Kelly, “The AES-CBC cipher algorithm and its use with IPsec,” *RFC3602*, Sept. 2003.

[27] D. Eastlake and P. Jones, “US secure hash algorithm 1 (SHA1),” *RFC 3174*, Sept. 2001.

[28] “Crypto++ library 5.5.2.” [Online]. Available: http://www.cryptopp.com/

[29] D. Johnson, A. Menezes, and S. Vanstone, “The elliptic curve digital signature algorithm (ECDSA),” *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

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

[31] “The network simulator - ns-2.” [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/User Information

[32] “Traffic and network simulation environment - TraNS.” [Online]. Available: http://trans.epfl.ch/

Albert Wasef received the Ph.D. degree (2011) from University of Waterloo (Canada) and B.Sc. (1998) degree and the M.Sc. (2003) from El Menoufia University (Egypt), both in electrical communications engineering. His research interest includes wireless network security, privacy preservation in vehicular networks, and group communications.

Xuemin (Sherman) Shen (IEEE M’97-SM’02-F’09) received the B.Sc. (1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen’s research focuses on mobility and resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a co-author of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering. Dr. Shen served as the Tutorial Chair for IEEE ICC’08, the Technical Program Committee Chair for IEEE Globecom’07, the General Co-Chair for Chinacom’07 and QShine’06, the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also serves as a Founding Area Editor for IEEE Transactions on Wireless Communications, Editor-in-Chief for Peer-to-Peer Networking and Application, Associate Editor for IEEE Transactions on Vehicular Technology, KICS/IEEE Journal of Communications and Networks, Computer Networks, ACM/Wireless Networks, and Wireless Communications and Mobile Computing (Wiley), etc. He has also served as Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier’s Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, and a Distinguished Lecturer of IEEE Communications Society.