MAAC: Message Authentication Acceleration Protocol for Vehicular Ad Hoc Networks

Albert Wasef and Xuemin (Sherman) Shen
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) to reliably secure the network. In any PKI system, the authentication of a received message is performed by checking that the certificate of the sender is not included in the current CRL, and verifying the authenticity of the certificate and signature of the sender. In this paper, we propose a Message Authentication Acceleration (MAAC) protocol for VANETs, which replaces the time-consuming CRL checking process by an efficient revocation check process. The revocation check process 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, the MAAC protocol uses a novel probabilistic key distribution, which enables non-revoked OBUs to securely share and update a secret key. By conducting security analysis and performance evaluation, the MAAC protocol is demonstrated to be secure and efficient.

I. INTRODUCTION

Vehicular ad-hoc networks (VANETs) have recently attracted extensive attentions as a promising technology for revolutionizing the transportation systems. VANETs consist of entities including On-Board Units (OBUs) and infrastructure Road-Side Units (RSUs). Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are two basic communication modes, which respectively allow OBUs to communicate with each other and with the infrastructure RSUs.

A well-recognized solution to secure VANETs is to deploy Public Key Infrastructure (PKI), and use Certificate Revocation Lists (CRLs) to manage the revoked certificates. In PKI, each entity in the network holds an authentic certificate, and every message should be digitally signed before it is transmitted. A CRL is a list containing all the revoked certificates, which is usually issued by a Trusted Authority (TA). In a PKI system, the authentication of any message is performed by first, checking that the sender certificate is not included in the current CRL, i.e., checking its revocation status, then, verifying the sender certificate, and finally, verifying the sender 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. 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 locations information of the drivers from any external eavesdropper [1]-[3], each OBU should be preloaded with a set of anonymous digital certificates, where it has to periodically change its anonymous certificate to mislead attackers [4]. Consequently, an OBU revocation results in revoking all the certificates carried by that OBU leading to a large increase in the CRL size; (2) The VANET network scale is very large. According to the United States Bureau of Transit Statistics, there are approximately 251 million OBUs in the United 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 big the CRL size can be, consider the case where 1% of the total number of the OBUs in the United States is revoked, and each OBU has one certificate. In this case, the CRL contains 2.51 million revoked certificates.

According to the Dedicated Short Range Communication (DSRC) [6], each OBU has to broadcast a message every 300 msec indicating its current position, speed, and the road conditions. 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. The ability to check a CRL for a large number of certificates in a timely manner forms an inevitable challenge to VANETs. Some works addressed the problem of distributing a large-size CRL in VANETs as in [7], where Raya et al. propose KC2RL (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. [8] propose to divide the CRL into small pieces and distribute each piece independently. Laberteaux et al. [9] propose to use car to car communication to speed up the CRL broadcasting. In [4], Raya et al. propose a PKI-based security and privacy protocol, where each vehicle needs to pre-load a huge pool of anonymous public/private keys, and the trusted authority also needs to store all the anonymous certificates of all the vehicles.

The probabilistic approach is a promising technique for the key management in ad hoc networks [10], [11]. Zhu et al. introduce the GKMPAN protocol [12], 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 [13], a probabilistic random key distribution is proposed to achieve efficient privacy-preserving group
communication protocol for 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 a message authentication acceleration (MAAC) protocol which replaces the CRL check process by an efficient revocation check process using a fast and secure HMAC function. The MAAC protocol 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. Section II introduces some preliminaries. The proposed MAAC protocol is presented in section III. Security analysis and performance evaluation are given in section IV and section V, respectively. Section VI concludes the paper.

II. PRELIMINARIES

In this section, we introduce the bilinear pairing [14], which is one of the foundations of the proposed protocol, the hash chains, and the adopted system model.

A. Bilinear Pairing

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 $\hat{\cdot} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear mapping with the following properties:

1) Bilinear: $\hat{\cdot}(aP, bQ) = \hat{\cdot}(P, Q)^{ab}$, for all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$.

2) Non-degeneracy: $\hat{\cdot}(P, Q) \neq 1_{\mathbb{G}_2}$.

3) Symmetric: $\hat{\cdot}(P, Q) = \hat{\cdot}(Q, P)$, for all $P, Q \in \mathbb{G}_1$.

4) Admissible: the map $\hat{\cdot}$ is efficiently computable.

The bilinear map $\hat{\cdot}$ can be implemented using the Weil and Tate 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 [15] is to determine the integer $l$, $0 \leq l \leq q - 1$, such that $Q = lP$.

B. Hash Chains

A hash chain [16] is the successive application of a hash function $h : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ to a secret value. 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 $v_0 = v, v_i = h(v_{i-1}) \forall 1 \leq i \leq j$.

Fig. 1. Hash chain

C. 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 can communicate either with other OBUs through V2V communications or with RSUs through V2I communications.

Note that the system model under consideration is mainly a PKI system, where each OBU has a set of anonymous certificates used to secure its communication with other entities in the network. Also, each OBU is pre-loaded with a set of keys. Those keys are necessary for generating a shared secret key between all the legitimate OBUs. Legitimate OBUs do not have sufficient incentives to disclose security materials, e.g., secret keys, certificates, etc., to the revoked OBUs, i.e., legitimate OBUs can not collude with the revoked OBUs. Finally, we consider that a compromised OBU is instantly detected by the TA.

III. THE PROPOSED MAAC PROTOCOL

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

A. System Initialization

The system is initialized as follows.

1) Initially, the TA selects two generators $P, Q \in \mathbb{G}_1$ of order $q$, and a set $U$ consisting of $l$ random numbers such that $k_i \in \mathbb{Z}_q^* \forall i \in \{1, 2, \cdots, l\}$, where $k_i$ is a random number. Then, the TA calculates its key pool consisting

Fig. 2. The system model
of l secret keys as \( K_i = k_i Q \in \mathbb{G}_1 \), where each key has a fixed identity \( i \in \{1, 2, \cdots, l\} \). Also, the TA calculates the public keys \( PK_{all} = \{PK_1, PK_2, \cdots, PK_l\} = \{\frac{1}{k_1} P, \frac{1}{k_2} P, \cdots, \frac{1}{k_l} P\} = \frac{1}{P} P \forall i \in \{1, 2, \cdots, l\} \) corresponding to the secret keys \( \{K_1, K_2, K_3, \cdots, K_l\} = \{k_1 Q, k_2 Q, k_3 Q, \cdots, k_l Q\} \). In addition, the TA selects a master secret key \( s \in \mathbb{Z}_q^* \) and an initial secret key \( K_y \in \mathbb{G}_2 \), where \( K_y \) is to be shared between all the non-revoked OBUs. Then, it calculates the corresponding public key \( P_o = sP \), and chooses a hash function \( H : \{0, 1\}^* \rightarrow \mathbb{G}_1 \). Finally, it selects another hash function \( h : \{0, 1\}^* \rightarrow \mathbb{Z}_q^* \), and a secret value \( v \in \mathbb{Z}_q^* \), and runs the hash chain shown in Fig. 1 for a large value of \( j \). It should be noted that the value of \( j \) should be large enough to accommodate with the number of revocation processes occur during the life-time of the network.

2) Each OBU in the network randomly picks from the key pool a key set \( (R) \) consisting of \( m \) distinct secret keys and the corresponding \( m \) public keys;

3) The TA issues for each OBU a set of anonymous certificates used to achieve privacy-preserving authentication;

4) The TA announces \( H, h, P, Q, \) and \( P_o \) to all the OBUs. After the system is initialized, each OBU should have the following information:
   - A set of anonymous certificates;
   - A set of \( m \) secret keys and the corresponding \( m \) public keys;
   - The secret key \( K_y \);
   - The hash function \( H, h, P, Q, \) and the public key \( P_o \).

B. 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 in the scope of this paper. We are only interested in how to accelerate the revocation check process, irrespective of the employed authentication mechanism.

C. Revocation

The revocation is triggered by the TA when there is an OBU to be revoked. The certificates of OBU must be revoked. In addition, the secret key set \( R_u \) of OBU and the current secret key \( K_y \) are considered revoked. Hence, a new secret key \( K_y' \) should be securely distributed to all the non-revoked OBUs. Also, each non-revoked OBU should securely update its compromised secret keys [12]. 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 \( PK_M = \frac{1}{m} P \). The TA then selects a random number \( t \in \mathbb{Z}_q^* \), and calculates the intermediate key \( K_{im} = tPK_M = \frac{1}{m} P \in \mathbb{G}_1 \), and the new secret key \( K_y' \) as follows

\[
K_y' = \hat{e}(K_M, K_{im}) = \hat{e}(k_M Q, t) = \hat{e}(Q, P)^{t k_M} = \hat{e}(Q, P)^t
\]

(1)

Also, it selects the value \( v_{j-ver} \) of the hash chain values, where \( v_i \) is the last value in the hash chain as shown in Fig. 1, and \( ver \) is the revocation version, i.e., it is an integer indicating 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 \( K_{msg} = (ver\{M\}||ID_{revkey}||K_{im}||enc_{K_y'}(v_{j-ver})) \).
where $ID_{rev\text{key}}$ is the identities of the revoked keys, and $enc_{K_y}(v_{j-\text{ver}})$ is the symmetric encryption of $v_{j-\text{ver}}$ using the key $K_y$. Finally, the TA broadcasts the following message $REV_{msg} = (CRL||K_{msg}||sgn_{K_{msg}})$, where CRL contains the certificates of the revoked OBUs, and $sgn_{K_{msg}} = sH(K_{msg})$ is the TA signature on the message $K_{msg}$. Note that the CRL is also signed by the TA;

2) After receiving the message $REV_{msg}$, each OBU verifies the TA signature on the CRL. Also, it verifies the signature $sgn_{K_{msg}}$ as follows $\hat{e}(sgn_{K_{msg}}, P) = \hat{e}(H(K_{msg}), P_{O})$. This verification follows since

$$\hat{e}(sgn_{K_{msg}}, P) = \hat{e}(sH(K_{msg}), P) = \hat{e}(H(K_{msg}), sP) = \hat{e}(H(K_{msg}), P_{O})$$

If $sgn_{K_{msg}}$ is correctly verified, the OBU checks if it has $K_M$ or not. If yes, the OBU uses $K_M$ and the intermediate key $K_{im}$ to calculate the new secret key $K_y = \hat{e}(K_M, K_{im})$ independently. Then, it decrypts $enc_{K_y}(v_{j-\text{ver}})$ using $K_y$ to get $v_{j-\text{ver}}$;

3) If an $OBU_v$ does not have the key $K_M$, it will not be able to calculate the new group key $K_y$ and obtain $v_{j-\text{ver}}$, and it must get $K_y$ from its neighboring OBUs. To get $K_y$, $OBU_u$ broadcasts its certificate and a request to get $K_y$, and starts its own timer $T_1$;

4) Any neighboring OBU of $OBU_v$ having $K_y$ uses the public key of $OBU_v$, included in its certificate, to encrypt $K_y$ and sends the encrypted $K_y$ to $OBU_v$;

5) If $OBU_v$ receives the encrypted $K_y$, it uses its secret key to decrypt $K_y$. Then, $OBU_v$ uses $K_y$ to decrypt and obtain $v_{j-\text{ver}}$ from the message sent by the TA. If the timer $T_1$ of $OBU_v$ is timed out without receiving the required data, then go to step 3;

6) The revoked OBUs can not compute $K_y$ since they do not have $K_M$. Also, they can not receive $K_y$ from other OBUs since the new CRL sent in $REV_{msg}$ contains the certificates of the revoked OBUs, which stops others from forwarding $K_y$ to them;

7) If any OBU, not previously missing any revocation process, possesses secret keys $\{K_i\} = \{k_iQ\}$ in the current $ID_{rev\text{key}}$, i.e., compromised keys, it updates the compromised secret keys as follows $K_i = v_{j-\text{ver}}K_i = v_{j-\text{ver}}\hat{e}(v_{j-\text{ver}}, k_iQ)$ and the corresponding public keys as follows $PK_i = \frac{1}{v_{j-\text{ver}}}PK_i = \frac{1}{v_{j-\text{ver}}k_i}P$;

8) If an $OBU_y$ missed a number of previous revocation messages and having the current $K_y$ and $v_{j-\text{ver}}$, it can update its compromised keys corresponding to the previous revocation processes by first calculating the hash chain values corresponding to the missed revocation processes starting from the current hash chain value $(v_{j-\text{ver}})$ as $v_{j-n+1} = h(v_{j-n}) \forall n \in \{\text{ver,ver} + 1, \cdots, \text{ver}y - 2\}$, where $\text{ver}_y$ is the version of the last revocation process received by $OBU_y$, i.e, $OBU_y$ generates the following hash chain values $\{v_{j-\text{ver}+1}, v_{j-\text{ver}+2}, \cdots, v_{j-y+1}\}$. Then, $OBU_y$ sends a request to the neighboring OBUs requesting $\text{ver}_y$ and $ID_{rev\text{key}}$ for all the missed revocation processes, where $\text{ver}_y$ and $ID_{rev\text{key}}$ are the revocation version and the identities of the revoked keys of a missed revocation process, respectively. It should be noted that $\text{ver}_y$ and $ID_{rev\text{key}}$ are sent in clear since they do not contain any sensitive data. After receiving the necessary data, for each missed revocation process, starting from the oldest process till the current revocation process, $OBU_y$ updates each key $K_i = k_iQ$ in $ID_{rev\text{key}}$ as $K_i = v_{j-\text{ver}y}K_i = v_{j-\text{ver}y}k_iQ$, and the corresponding public key as $PK_i = \frac{1}{v_{j-\text{ver}y}}PK_i = \frac{1}{v_{j-\text{ver}y}k_i}P$. Note that $v_{j-\text{ver}y}$ belongs to $\{v_{j-\text{ver}+1}, v_{j-\text{ver}+2}, \cdots, v_{j-y+1}\}$;

9) After performing the key set update, each OBU erases $K_{im}$, the hash chain values, and the original compromised secret and public keys. Also, each OBU stores $\text{ver}_y$ and $ID_{rev\text{key}}$.

Remarks

- Note that if an OBU missed a revocation message, it is still able to share in the upcoming revocation processes since only the compromised keys are updated. Hence, it can use its non-compromised secret key(s) in the future to get the new secret key $K_y$.
- An important contribution of the proposed MAAC protocol 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, our work is the first to propose a rekeying mechanism capable of updating compromised keys corresponding to previously missed rekeying processes.
- Note that the MAAC protocol has a modular feature, which makes it integrable with any PKI system. In other words, MAAC 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.
- The MAAC protocol is suitable for not only VANETs but also any type of networks employing PKI.

IV. Security Analysis

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

1) Resistance to forging attacks: To forge the revocation check $REV_{\text{check}} = HAMC(K_y, ID_{\text{key}}||T_{\text{stamp}})$ of any $OBU_u$, an attacker has to find the current $K_y$, which is equivalent to finding $t$ in the following ECDLP problem: given $K_{im} = tPK_M = \frac{1}{k_{im}}P$ and $PK_M = \frac{1}{k_{im}}P$, find $t$ such that $K_{im} = tPK_M$. Similar analogy applies to finding the TA secret key $s$ from the TA message signature $sgn_{K_{msg}} = sH(K_{msg})$. Since ECDLP is a hard computational problem, i.e., it can not be solved in a sub-exponential time, the
revocation check and the TA message signature $sgn_{K_{msg}}$ are unforgeable. Similarly, finding the TA secret value $s$ from $P_0 = sP$ is ECDLP problem, which makes it unforgeable. From the aforementioned discussion, it is concluded that the MAAC protocol is resistant to forging attacks.

2) 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 can not use its a hash chain value $v_{j-\text{ver}+1}$ received in a previous revocation process to get the current hash chain value $v_{j-\text{ver}}$. Consequently, a revoked OBU can not update its secret key set, and hence, it can not get the new secret key $K_g$. As a result, the proposed MAAC protocol guarantees forward secrecy.

3) Resistance to replay attacks: Since in each message an OBU includes the current time stamp in the revocation check value $REV_{\text{check}} = HAMC(K_g, ID_u || T_{\text{stamp}})$, an attacker can not record $REV_{\text{check}}$ at time $T_i$ and replay it at a later time $T_{i+1}$ to pass the revocation check process as the receiving OBU compares the current time $T_{i+1}$ with that included in the revocation check. Consequently, the MAAC protocol is secure against replay attacks.

4) Resistance to bogus message attacks: Since each OBU checks the information correctness of the received message with respect to the previously received messages, a bogus message will fail to pass the information correctness test, and it will be eventually dropped by the OBU. Therefore, the MAAC protocol is resistant to bogus message attacks.

V. PERFORMANCE EVALUATION

A. Authentication Delay

In this section, we compare the message authentication delay employing the CRL and MAAC protocol, respectively. As stated earlier, the authentication of any message is performed by three consecutive phases: checking the sender revocation status, verifying the sender certificate, and verifying the sender signature. For the first authentication phase, which checks the revocation status of the sender, we employ either the CRL or the MAAC protocol. For the MAAC protocol, we adopt the Cipher Block Chaining Advanced Encryption Standard (CBC-HMAC AES) [17] and Secure Hash Algorithm 1 SHA-1 [18] as the HAMC functions. We consider the OBU identity (ID) and the time stamp ($T_{\text{stamp}}$) having equal lengths of 4 bytes. We adopt the Crypto++ library [19] for calculating the delay of the HAMC functions, where it is compiled on Intel Core2Duo 2 GHz machine. We found the delay of using CBC-HMAC AES and SHA-1 to calculate the revocation check ($REV_{\text{check}} = HAMC(K_g, ID_u || T_{\text{stamp}})$) to be 0.11 $\mu$s and 0.21 $\mu$s, respectively. Also, we simulated the CRL check process using C++ program compiled on the same machine. The CRL check program performs progressive search on a text file containing the revoked certificates identities. For the second and third authentication phases, we employ ECDSA [20] to check the authenticity of the certificate and the signature of the sender. ECDSA is the digital signature method chosen by the VANET standard IEEE1609.2. In ECDSA, a signature verification takes $2T_{\text{mul}}$, where $T_{\text{mul}}$ denote the time required to perform a point multiplication on an elliptic curve. Consequently, the verification of a certificate and signature takes $4T_{\text{mul}}$. In [21], $T_{\text{mul}}$ is found for a supersingular curve with embedding degree $k = 6$ to be equal to 0.6 msec.

Fig. 3 shows a comparison between the authentication delay per message using the CRL and the MAAC protocol vs. the number of revoked 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 CRL increases with the number of revoked certificates, i.e., with the size of the CRL. Also, the authentication delay using MAAC protocol is constant and independent of the number of revoked certificates. In addition, authentication using the MAAC protocol outperforms that using the CRL. For example, the authentication delay per message using the CRL and MAAC (SHA-1) for 50,000 revoked certificate are 41.4 msec and 2.4002 msec, respectively. Consequently, the MAAC (SHA-1) protocol accelerates the message authentication by 94.2% compared to that using the CRL.

B. End-to-end delay

To further evaluate the MAAC protocol, we have conducted ns-2 [22] simulation for the city street scenario shown in Fig. 4. The adopted simulation parameters are given in Table
TABLE I
NS-2 SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation area</td>
<td>7.4 Km x 7.4 Km</td>
</tr>
<tr>
<td>Simulation time</td>
<td>30 sec</td>
</tr>
<tr>
<td>Max. OBU speed</td>
<td>60 Km/h</td>
</tr>
<tr>
<td>OBU transmission range</td>
<td>300 m</td>
</tr>
<tr>
<td>OBU information dissemination interval</td>
<td>300 msec</td>
</tr>
</tbody>
</table>

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

1. We select the road condition information dissemination by an OBU to be every 300 msec to conform with the DSRC VANET standards. The mobility traces adopted in this simulation are generated using TraNS [23]. We are interested in the end-to-end delay, which is defined as the time taken by a message since it is generated at the application layer of the sender till it is received at the application layer of the receiver. Fig. 5 shows the end-to-end delay in msec vs. the OBUs density employing authentication using the proposed MAAC (SHA-1) protocol and the conventional CRL. In the conducted simulation, we consider CRLs containing 30000, 50000, and 70000 revoked certificates, respectively. Also, we consider the OBUs density as the number of OBUs per km². 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. From Fig. 5, employing the proposed MAAC protocol in authentication substantially reduces the end-to-end delay compared to that using the conventional CRL.

VI. CONCLUSION

We have proposed MAAC protocol for VANETs which accelerates message authentication by replacing the time-consuming CRL check process with a fast revocation check process employing HMAC function. The proposed MAAC protocol 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, MAAC protocol has a modular feature making it integrable with any PKI system. Furthermore, it is resistant to common attacks while outperforming the authentication techniques employing the conventional CRL. Our future work will focus on certificate and message signature authentication acceleration.

REFERENCES