ON-THE-FLY KEY GENERATION 
FOR AN EFFICIENT ROAD SIDE UNITS BASED 
AUTHENTICATION IN VEHICULAR NETWORKS 

1Ashraph Sulaiman, 2S.V. Kasmir Raja and 3Sung Han Park 

1Department of Computer Science, National University of Rwanda, Huye, Rwanda 
2Department of Computer Engineering, SRM University, Chennai, India 
3Department of Computer Engineering, Hanyang University, Seoul, South Korea 

Received 2013-05-26; Revised 2013-06-21; Accepted 2013-07-04 

ABSTRACT 

Obtaining efficient security without compromising privacy is a primary issue in vehicular communication. Though many counterparts proposed solutions in this regard, accommodating scalability, security and traceability altogether is a difficult task due to the contradictions between these qualities. Some of the previous studies suggests RSU based authentication to address the above issues, while others propose independent OBU authentication. In either scheme, any one of the entities is overloaded during key generation and verification processes. The proposed scheme addresses these issues, by distributing the workload between OBUs and RSUs to outperform other protocols. We propose a novel scheme, in which OBUs generate short-lived public keys on the fly and other vehicles can verify them with the help of RSUs. This protocol also admits certificate-less authentication, in addition to aggregated signature verification. Therefore, the total verification time can be drastically reduced in the proposed scheme. We analyze the proposed protocol significantly to demonstrate its efficiency. 

Keywords: VANET, Privacy, Security, Traceability, Pseudo-Id, Signature Verification 

1. INTRODUCTION 

Vehicular Networks (VANETs) are established to enhance road safety, traffic management and infotainment facilities. In VANET, each vehicle is equipped with an On Board Units (OBUs) to communicate with other vehicles, Road Side Units (RSUs) that are located on the roads and the Trusted Authority (TA) to register RSUs and OBUs. According to (USDT, 2006) OBUs frequently broadcasts routine traffic related messages with information about its position, current time, direction, speed, acceleration/deceleration, traffic events. This helps the vehicle to be warned with critical situations such as accidents, traffic jams. 

Through this communication helps the driver community, it has a critical side effect of privacy. Some studies (Raya and Hubaux, 2005; 2007) proposed pseudonym based approach to solve this problem. Generation of pseudonyms by the TA or RSUs is not an issue with their high computation and storage capacity. However, the computation cost of OBUs grows linearly with the traffic density. Some studies suggests RSU based authentication to reduce the burden of vehicles, while others propose independent OBU authentication. Both the schemes suffer with scalability and message loss problems, as any one entity (OBU or RSU) is solely responsible for key generation and/or verification. The proposed scheme addresses these issues by allowing both OBUs and RSUs to contribute in authentication process.

2. RELATED WORKS 

Many studies have been reported on the security and privacy-preservation issues for VANETs proposed by several authors (Raya and Hubaux, 2007; Lin et al., 2007; Zhang et al., 2008a; Ren et al., 2006; Lu et al., 2007).
2008). They can be grouped into three categories. First category is based on a huge number of pseudo-anonymous key based (HAB) protocols proposed by several authors (Raya and Hubaux, 2007; Lin et al., 2008; Mak et al., 2005; Xu et al., 2007; Xi et al., 2007; 2008). Though this is a straightforward solution, this method requires each OBU has to take large storage space to store a number of anonymous key pairs.

The second category is based on Group Signature (GSB), which was first introduced by Chaum and VenHeyst (1991). This allows a group member to sign messages anonymously on behalf of the group. In case of a dispute, the group manager can identify the identity of a signer. According to Xiong et al. (2010) although the group signature can achieve anonymity on conditional privacy preservation, the time for message verification grows linearly with the number of revoked vehicles. Lin et al. (2008) propose an efficient security protocol called GSIS. With this protocol, only a private key and group public key are stored in the vehicle and the messages are signed according to the group signature scheme without revealing any identity information to the public. However, the verification of group signature requires at least two pairing operations, which may not be scalable when the density of traffic is increased. Finally, Calandrillo et al. (2007) proposed a hybrid approach by combining the pseudonym and the group signature schemes. However, this approach suffers with the same drawbacks.

The third category employs the RSUs to assist message authentication. Lu et al. (2008) proposed a protocol called ECPP, in which the RSU issues only an ephemeral certificate for valid vehicles at the time of authentication to eliminate the certificate requirement and the RL. In RAISE, Zhang et al. (2008a) employed RSUs to authenticating messages. Compared to the solutions previously mentioned, this scheme enables lower computation and communication overheads for each vehicle. Also, Zhang et al. (2008b) introduced IBV scheme, in which multiple signatures can be batch verified instead of one by one. Therefore, the signature verification speed improved significantly and alleviated the computational workload of the RSUs. By generating distinct pseudo identities and the corresponding private keys for signing each message with a tamper-proof device, privacy regarding user identity and location of the vehicles can be protected. However, this scheme requires additional hardware to be installed on OBUs to generate pseudo identities.

However, the verification process of most of the protocols solely depends on either on OBUs or on RSUs, which leads to scalability issues when the traffic density goes high. In order to address this downside, we propose this scheme to employ both RSUs and OBUs to work together for the key generation and verification processes, in order to distribute the workload between the two. Thus, this scheme achieves a better performance comparatively to other counterparts even in a high traffic situation.

3. SYSTEM MODELS AND PRELIMINARIES

3.1. System Model

VANET architecture consists of three entities as in Fig. 1: (1) the Trusted Authority (TA), who is in-charge for the registration of RSUs and OBUs, (2) the RSUs at the roadside, that act upon the commands of TA and (3) the vehicles equipped with OBUs in order to communicate with other vehicles.

3.2. System Requirements

As any other VANET system, we assume that our system fulfills the following requirements:

- Anonymous Authentication: From the message senders’ perception, leaking their privacy information such as Real ID (RID) of the vehicle is unacceptable
- Unlink ability and Traceability: Any recipient cannot link two or more messages sent by a vehicle to other vehicles. On the other hand, the authorities should be able to trace the sender of the message by mapping the message with the real identity of the sender in case of any liability investigation
- Scalability and Low overhead: Any application of the vehicular networks must be scalable to a large network. The computation and communication overhead increases linearly with the number of vehicles in the network

3.3. Bilinear Pairing

Since bilinear maps are the basis of our scheme, we briefly introduce them here. Let $G_1$, $G_2$ be the cyclic additive and multiplicative groups of same prime order $q$. Let $P$ be the generator of $G_1$. An admissible bilinear map is a map $ê: G_1 \times G_1 \rightarrow G_2$ satisfying the following properties:

- Bilinearity: $\forall a, b \in G_1$ and $\forall a, b \in \mathbb{Z}_q$, $ê(g^a, h^b) = ê(g, h)^{ab}$
- Nondegeneracy: $ê(1,1) \neq 1_{G_2}$
- Computability: $\exists an algorithm to ê(a,b), \forall(a,b) \in G_1$
Table 1. Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Secret between RSU and its OBU's</td>
</tr>
<tr>
<td>$V_x$</td>
<td>The x-th vehicle</td>
</tr>
<tr>
<td>R</td>
<td>The RSU</td>
</tr>
<tr>
<td>$PK_x$</td>
<td>Long term public key of $V_x$</td>
</tr>
<tr>
<td>$sK_x$</td>
<td>Corresponding private key of $PK_x$</td>
</tr>
<tr>
<td>$T_{exp}$</td>
<td>Time expiry</td>
</tr>
<tr>
<td>$Cert_{TA}[PK_x]$</td>
<td>TA’s certificate on $PK_x$</td>
</tr>
<tr>
<td>${PID}_x$</td>
<td>Short-lived pseudo-id of $V_x$</td>
</tr>
<tr>
<td>$PK_R$</td>
<td>Public key of RSU</td>
</tr>
<tr>
<td>$sk_R$</td>
<td>Corresponding private key of $PK_R$</td>
</tr>
<tr>
<td>$K_{ss}$</td>
<td>Session key between $V$ and RSU</td>
</tr>
<tr>
<td>$h(.)$</td>
<td>A one way hash function such that $SHA-1$ (Eastlake and Jones, 2001)</td>
</tr>
<tr>
<td>$H(.)$</td>
<td>Hash function such as $H : {0, 1} \rightarrow \mathbb{G}$ (Sweeney, 2002)</td>
</tr>
</tbody>
</table>

Such bilinear map $\hat{e}$ can be constructed by modified Weil (Boneh and Franklin, 2001) or Tate pairing (Miyaji et al., 2001) on elliptic curves.

4. PROPOSED SCHEME

4.1. System Initialization

All the OBUSs and RSUs must register themselves with the TA before they join in the VANET. The TA is in-charge of checking the vehicle’s identity and to provide a long-term public/private key pair for each vehicle and to set up the system parameters \{G_1, G_2, q, P\} for RSUs and OBU. Rests of the notations are listed in Table 1.

4.2. Short-Lived Key Pair Generation

Firstly, RSU (hereafter we say R) randomly chooses $s \in \mathbb{Z}_q$ as a common secret between the vehicles in its range and computes $Q = sP$. Also, RSU is responsible to choose a distinct Pseudo-ID (PID) for each vehicle when it comes into its communication range. The detailed working of our protocol is as follows.

At regular intervals, R broadcasts hello messages. When $V_x$ enters into R’s proximity, it detects the hello message. Immediately, $V_x$ sends its $Cert_{TA}[PK_x]$, signed by TA and a random number $r_1 \in \mathbb{Z}_q$ to R, to initiate the mutual authentication process. After authenticating $Cert_{TA}[PK_x]$, R chooses $r_2$ as its share to establish a shared session key between $V_x$ and itself. This process can be achieved through Diffie and Hellman (1976) key agreement protocol. Besides, R chooses a unique PID for $V_x$ and sends \{r_2||(PK_x ||T_{exp}||s||Q||2)E_{K_{ss}}\}, where $E_{K_{ss}}$ is encryption using $K_{ss}$.
With this PID’s, Vx can now generate anonymous short-lived key pairs on the fly in order to send traffic related messages to other vehicles. In our scheme, OBU’s generate these key pairs (U and v respectively) randomly from the given Pseudo-IDs (PID), based on ID based cryptography (Sha et al., 2006). Each U is composed of U1 and U2. This U1 and U2 are the cipher texts of Elgamal (1985) encryption algorithm. Similarly, each private key v consists of v1 and v2. Generation of these keys pairs can be detailed in algorithm 1.

Algorithm 1: On-the-fly generation of short-lived public/private keys by the OBUs

Input: PID’s obtained from RSU

Output: Short-lived anonymous key pairs U’s and v’s

i. Computes the short-lived public key U’ as:

\[ U’_x = \text{PID}^{v_x} \oplus P \]
\[ U’_x = h\left(\text{PID}^{v_x}\right) \oplus H\left(\text{PID}^{aQ}\right) \]

where, a is a random nonce, \( \oplus \) is an XOR operation

ii. Computes the corresponding public key vas:

\[ v’_x = sU’_x \]
\[ v’_x = sH\left(U’_1 \parallel U’_2\right) \]

In order to generate unique key pairs, Vx changes the random nonce each time it generates a short-lived public/private key.

4.3. Signature Generation

When a vehicle Vx wants to send message Mx, it generates a short-lived key pair as in algorithm 1. It then computes a signature ‘\( \sigma’ \) on M using the short-lived private key v’s = (v’1, v’2) in such a way that\( \sigma’ = v’_1 \oplus M + v’_2 \). After that Vx sends {U’s || M||TS||{Uvx||M||TS} σ’s} to other vehicles. For sending subsequent messages Vx changes its short-lived key pairs by choosing a distinct random nonce ‘a’.

4.4. Aggregated List of Pseudo-IDs Agree

Meanwhile, R periodically broadcasts an aggregated list of issued pseudo-id hashes haggr = \{h(PID1), h(PID2) ... h(PIDn)\} to the vehicles in its communication range. This list eliminates the certificate overhead. For this purpose, R first hashes the PIDs that are not expired, aggregates them all and signs the aggregated PIDs using sk and sends out the signed list haggr||haggr||haggr. Each time R issues a Pseudo-ID (PID) to a new vehicle, it appends the new PID in its aggregated list. Similarly, when a PID reaches Texp, it will be cut off from the list. In case a vehicle remains in same R even after the Texp of its PID, it can continue participating in the communication by requesting a new PID from R using Ksy.

4.5. Verification

4.5.1. Aggregated Hash Verification

When a vehicle receives messages sent by the other vehicles, the receiver verifies the authenticity of the short-lived public keys from the aggregated pseudo-id hashes published by the RSU’s periodically. For this ground, the receiver first computes the pseudo-id hash h(\{PID’s\}) of a short-lived public key U’s = U’1 + U’2 as follows Equation 1:

\[
= h\left(\text{PID}^{v_x}\right) \oplus H\left(\text{PID}^{aQ}\right) \oplus H\left(s\text{PID}^{aP}\right)
= h\left(\text{PID}^{v_x}\right) \oplus H\left(\text{PID}^{aQ}\right) \oplus H\left(\text{PID}^{aQ}\right)
\tag{1}
\]

= h\left(\text{PID}^{v_x}\right)

After extorting the pseudo-id hash h(\{PID’s\}) from the short-lived public key U’s, it compares the h(\{PID’s\}) in the RSU list for its existence.

4.5.2. Batch Signature Verification

Once the receiver confirms the genuineness of the pseudo-ids of all received messages through the aggregated list of pseudo-id hashes, it undergoes verification of signatures for the corresponding short-lived public keys. The authentication of a signature in a message can be carried out using the short-lived public key U of the sender attached in the message. With the system public parameters \{G, Gz, q, p\} assigned by the TA and the parameters \{s, Q\} obtained from RSU, the receiving vehicle verifies the signature of the sender Vx as below Equation 2:

\[
\mathcal{e}\left(\sigma’_1, P\right)
= \mathcal{e}\left(v’_1, M + v’_2\right)
= \mathcal{e}\left(v’_1, M\right) \mathcal{e}\left(v’_2, P\right)
= \mathcal{e}\left(sU’_1, M\right) \mathcal{e}\left(sH\left(U’_1 \parallel U’_2\right), P\right)
= \mathcal{e}\left(sP, U’_1, M\right) \mathcal{e}\left(sP, H\left(U’_1 \parallel U’_2\right)\right)
= \mathcal{e}\left(Q, U’_1, M + H\left(U’_1 \parallel U’_2\right)\right)
\tag{2}
\]
Since we employ batch verification in our scheme, a receiver can collectively verify n distinct messages from n distinct vehicles once in every 300 ms. If the receiver receives $\sigma_1, \sigma_2 \ldots \sigma_n$, the signature on the messages $M_1, M_2 \ldots M_n$ with their public keys $U_1, U_2 \ldots U_n$ then, those signatures are valid if the following Equation 3 holds:

$$\varepsilon \left( \sum_{i=1}^{n} \sigma_i \right)$$

(3)

4.6. Additional Storage Requirement

Considering the storage requirement, our protocol requires each OBU to store the aggregated list of pseudo-id hashes published by the RSU periodically. However, this may require a small amount of storage capacity, as this list would not grow long, since the expired pseudo-id will be erased incessantly from the list by the RSU.

5. ANALYSIS AND EVALUATION

5.1. Security Analysis

Claim 1: Privacy Preservation and Anonymous Authentication is Achieved

Proof

The RSU can authenticate vehicle $V_i$ through its long-term public key $PK_{V_i}$, since it is signed by TA’s private key. By this way, the real identity of the vehicle is preserved within TA. The short-lived public keys that are used for sending messages are generated from a pseudo-id given by the RSU, that has no trace of this long-term public key. Even if the RSU is hacked by any high level attacks, the real id of a vehicle cannot be revealed from the RSU. In terms of anonymous authentication, RSUs periodically broadcast the aggregated list of valid pseudo-ids signed by its private key $sk^k$ to the vehicles in its range. Therefore, a vehicle can trust a public key if its pseudo-id hash extracted from its public key is present in the aggregated pseudo-id hash of RSU. This shows that, claim 1 is correct.

Claim 2: The Anonymity of the Message Originator and Traceability by the Authorities is Assured

Proof

The short-lived anonymous public key $U$ is computed in such a way that $U_1 = PIDaP, U_2 = h(PID)@H(PIDaQ)$ where, ‘a’ is a random number which would be changed by the vehicle for every different messages. This guarantees a unique short-lived public key each time. Moreover, the pseudo-id of a vehicle PID cannot be retrieved from its hash $h(PID)$ because of the irreversible property of one-way hash chain. Therefore, a receiver cannot link any two short-lived public keys that are generated from the same PID. In case of any dispute, the RSU first fetch the pseudo-id hash in the accused message in order to find the real PID value of the message sender. Later, it extracts the long-term public key of the responsible vehicle and submits it to the TA for penalty. Therefore, claim 2 is correct.

Claim 3: Scalability and Low Verification Overhead is Guaranteed

Proof

In the proposed protocol, a public key certificate is not required as the public keys can be authenticated form the list of aggregated pseudo-id hashes published by the RSU. Though this requires the RSU’s signature in the list to be verified, it is one signature shared for n messages. Therefore, our protocol dramatically reduces verification overhead and improves the scalability of the system. This confirms that claim 3 is correct.

5.2. Performance Evaluation

5.2.1. Verification Delay

We evaluated and compared our protocol with the following schemes. ECDSA proposed by Boneh et al. (2001) BLS proposed by Boneh et al. (2003) and GSIS proposed by Choi et al. (2011). Here, ECDSA is the traditional PKI based scheme, BLS and GSIS are group based, group and identity based signature schemes respectively. Considering the time to perform one pairing operation $T_{pair}$, one point multiplication over elliptic curve cryptography $T_{mul}$; we used the experiment of Scott (2007) with an MNT curve of embedding degree k = 6 and 160 bit q simulated on an Intel Core 2 Duo @ 3 GHz machine and attained the values for $T_{pair} = 4.5$ and $T_{mul} = 0.6$ ms.

As depicted in Table 2, we calculated the time to sign and verify a single message and n messages. ECDSA uses one $T_{mul}$ operations to sign and 4 times $T_{mul}$ operations to verify a single message. During message verification, for n messages, it requires n times operations as that of single message verification. BLS uses one $T_{mul}$ one $T_{pair}$ per signing and 4 times pairing and 2 times point multiplication operations to verify a single message. While on the other hand, for verifying n messages, it requires $(2n+2)T_{pair}+2nT_{mul}$ operations. This is because BLS performs aggregated verification. GSIS is closer to BLS but does not use $T_{imp}$ operation. Rather, $3T_{pair}+9T_{mul}$ is required during signing a single message and two additional pairing and one reduced multiplication operations are needed for verifying a single message.
For n messages, signing and verification in GSIS requires an equivalent of n times operations with its single message signing and verification time respectively. Our protocol requires similar time as that of ECDSA in signing messages. For verification, our protocol requires \(4T_{pair} \) in addition to one \(T_{mul} \) and one \(T_{mtp} \) operations. This is because, in the total \(4T_{pair} \) operations, \(2T_{pair} \) remains the same for batch verifying n signatures and another \(2T_{pair} \) is for the verification of the ECDSA signature of the RSU that comes along with the list of aggregated pseudo-id hashes, which is one for n messages.

Figure 2 illustrates the message verification delay of the various schemes ECDSA, GSIS and BLS compared with our protocol. The verification delay of GSIS and BLS is higher, when compared to the verification time of ECDSA and our protocol. GSIS starts losing the messages, when the number of messages increases over 10 within 300ms. BLS takes a similar time, as GSIS when number of messages are 50 to verify. ECDSA verifies around 125 messages in 300ms. Our protocol verifies closely twice the messages as verified by ECDSA within the same 300ms time interval.

### 5.2.2. Communication Overhead

To measure the communication overhead of our protocol, we evaluated our protocol using ns-2 the network simulator, simulation with the parameters shown in Table 3.

Our protocol is compared with the ECDSA, BLS and GSIS for the communication overhead occurred due to the cryptographic operations used in the schemes. ECDSA, BLS and GSIS schemes use a certificate of 125 bytes along with their signature costs. 181, 146 and 184 are the additional communication overhead for the above schemes respectively. Since our protocol uses a certificate less communication, it requires an additional overhead of only 42+21+(56/n)bytes, in which 42 bytes are for the short-lived public key and 21 bytes is for its corresponding signature. An additional overhead of 56 bytes is required for the RSU’s ECDSA signature in the list of pseudo-id hashes that are periodically by the RSU. However, these 56 bytes would be shared for n messages, as the RSU aggregates all the pseudo-id hashes into one list.

Therefore, a total of \(63+(56/n) \) bytes are required as the compared schemes with our protocol. We observe that our protocol has the lowest message loss ratio when compared to the other schemes. Since we deploy the batch verification and avoided the certificate verification, our protocol is able to verify more messages than the other compared counterparts.

### Table 2. Comparison of signing and verification speed

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Signing</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 message</td>
<td>n messages</td>
</tr>
<tr>
<td></td>
<td>( T_{mul} )</td>
<td>( n T_{mul} )</td>
</tr>
<tr>
<td>ECDSA</td>
<td>( T_{pair} + 9T_{mul} )</td>
<td>( 3nT_{pair} + 9nT_{mul} )</td>
</tr>
<tr>
<td>GSIS</td>
<td>( T_{pair} + 9T_{mul} )</td>
<td>( 3nT_{pair} + 9nT_{mul} )</td>
</tr>
<tr>
<td>Our Protocol</td>
<td>( T_{pair} + 9T_{mul} )</td>
<td>( 3nT_{pair} + 9nT_{mul} )</td>
</tr>
</tbody>
</table>

Communication overhead of BLS is slightly less than ECDSA and GSIS schemes and consumes slightly over 8 MB, when 300 vehicles in range. Our protocol requires less than the half of the overhead of BLS when the communication is between 300 vehicles and a RSU.

### 5.2.3. Message Loss Ratio

We evaluated the message loss ratio of our protocol using ns-2 with the parameters shown in Table 3 and compared it with the other studied protocols. The average message loss ratio is defined as the average ratio between the number of messages dropped every 300 ms due to cryptographic delays and the total number of messages received in every 300 ms. This can be calculated from the maximum number of signatures and certificates that can be verified by a protocol in 300 ms.

Figure 4 shows the average message loss ratio between the compared schemes with our protocol. We observe that our protocol has the lowest message loss ratio when compared to the other schemes. Since we deploy the batch verification and avoided the certificate verification, our protocol is able to verify more messages than the other compared counterparts.
Fig. 2. Message verification delay of compared schemes

Fig. 3. Communication overhead of compared schemes

Fig. 4. Message loss ratio of compared schemes
Table 3. NS-2 Simulation Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation area</td>
<td>7.5×7.5 Km</td>
</tr>
<tr>
<td>Simulation time</td>
<td>30000 ms</td>
</tr>
<tr>
<td>Maximum speed of vehicles</td>
<td>60 Km/h</td>
</tr>
<tr>
<td>OBU transmission range</td>
<td>300 m</td>
</tr>
<tr>
<td>MAC protocol</td>
<td>802.11a</td>
</tr>
<tr>
<td>OBU data dissemination interval</td>
<td>300 ms</td>
</tr>
<tr>
<td>Wired channel capacity</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>Wireless channel capacity</td>
<td>6 Mbps</td>
</tr>
<tr>
<td>Distribution of RSUs</td>
<td>Uniform</td>
</tr>
</tbody>
</table>

6. CONCLUSION

In this study, we propose a novel RSU based on-the-fly anonymous short-lived public key generation. With our protocol, RSUs are responsible to provide a pseudo-identity to the OBUs, which come into its proximity. The OBUs can generate on-the-fly short-lived public keys using the pseudo-id. This protocol considerably reduces the verification overhead, as it does not require any public key certificate for the authentication of the short-live public keys. This is because; the RSUs periodically publish the valid pseudo-id hashes, which would be used by the vehicles to compare the pseudo-id hashes of the received messages for its trustworthiness. Extensive simulation has been conducted to demonstrate the low overhead and high performance of our protocol.

For future research, we will contribute to reduce the signature verification cost for vehicle-to-vehicle communication when the fixed infrastructure such as RSUs is absent in the network.

7. REFERENCES


