Efficient Privacy Preservation Protocol Using Self-certified Signature For VANETS

Bidi Ying[1,2], Dimitrios Makrakis[1], Hussein T. Mouftah[1]
Broadband Wireless & Internetworking Research Laboratory,
School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada[1]
Zhejiang Gongshang University, Hangzhou, China[2]
{byiung, dimitris, mouftah} @site.uottawa.ca

***Abstract*—Privacy and security are two important issues in vehicular networks. Traditional authentication methodologies such as public/private keys & their corresponding certificates are not feasible to protect location privacy and security due to large computing overhead. In this paper, we propose an Efficient Privacy Preservation (EPP) protocol in vehicular ad-hoc networks, which, uses smart card functionality to authenticate users and employs bilinear pairings method to generate the public and private key. The public key is derived from the signature of the user's pseudonym identity and private key signed by a trust authority and roadside units, hence, users can verify signatures without their corresponding certificates. Performance analysis shows the proposed EPP protocol can authenticate vehicular users and data messages with low time complexity and preserve users' privacy.**

***Keywords*-Vehicular networks, self-certified, privacy**

I. INTRODUCTION

Vehicular ad-hoc network (VANET) is a promising technology expected to play an important role in road safety, traffic management, and information dissemination to drivers and passengers [1]. A VANET mainly consists of On-Board Units (OBUs) and Roadside Units (RSUs) [2]. OBUs are installed on vehicles while RSUs are deployed to act as base stations providing connectivity to properly equipped vehicles located in their area of coverage.

Authentication is an important feature in a VANET as the source of the information should be verified to ensure the legitimacy of the data communicated [3-4]. Compare to wired network applications, VANET applications typically have more stringent authentication requirements. First of all, authentication should be done in short time in VANETs to ensure enough time for the drivers to take action. For e.g. a message update slower than once every 500 msec is probably too slow. Driver reaction time to stimuli like brake lights can be of the order of 0.7 Sec and higher [5]. Thus if updates come in slower than every 500msec, the driver may realize something is wrong before the safety system. This would make the driver think the safety system is not effective. Second, a certain degree of anonymity is typically required to ensure privacy of drivers, and the authentication model must ensure that this anonymity is maintained. For example, a rouge user could target and succeed to collect messages generated by other vehicular users and obtain sensitive information such as the driver's name, license plate, speed, location of vehicle, route of travel without successful security and privacy guarantee mechanisms in place. Third, VANETs are highly mobile and the mobility should be considered when designing the authentication protocol for possible network partitioning. For instance, if two cars drive in opposite directions with 90 Km/h each, and if we assume a theoretical wireless transmission range of 300meters, communication is only possible for 12 seconds.

Most of the existing security proposals for secure VANETs are based on the use of an asymmetric algorithm [1, 3-4]. For example, an algorithm using public/private keys and their corresponding anonymous certificates to authenticate messages requires larger storage of a huge number of keys and larger computing overhead. A short-time signature using bilinear pairings was proposed in [6] to use in the electronic cash system, which allows a user to get a signature without giving the signer any information about the actual message or the resulting signature. However, this short-time signature is impractical for vehicular networks due to high mobility. Girault [7] first introduced self-certified public key, where the private key of each user is only known to the user himself, while the corresponding public key is derived from the signature of the user's identity and private key. This self-certified method can implicitly validate the user's public keys, and cannot need extra corresponding certificates. Hence, it can reduce storage space. Shuo [8] proposed a further research about Self-Certified Signature (SCS) where users can choose their private keys and the actual public key consists of the public key of a Trust Authority (TA) and the partial public key chooses by the user, along with the identity of the user explicitly. However, when sending the signed message together with the public key Y_{TA}, its partial public key Y_{ID} and its identify *ID*, an attacker still can link its *ID* to the user by monitoring messages. Thus, it can reveal private information regarding the activities of the user. Besides, real-time data from users is to be accessed directly by an external party (eg. attacker), which will leak sensitive information to the external party.

In order to solve above problems, we introduce an efficient privacy preservation protocol, which is based on bilinear pairings. EPP scheme uses smart card to authenticate users and RSUs before allowing the user to access data; and employs bilinear pairings to generate the partial private key. The actual private key including two partial private keys is used to sign a data message, and the corresponding public key derived from the user's pseudo ID/ partial public key and the TA's public key is used to verify the signature. This strategy helps to avoid some active attacks such as forgery attack and guess attack, as will be shown in the security analysis section. Compared to the SCS scheme, the time complexity of EPP scheme remains lower, as will be shown in the time complexity analysis section.

The remainder of the paper is organized as follows. Section II presents background. In Section III, the EPP scheme is described in detail. Section IV provides performance analysis. In Section V, the related work is surveyed. Finally, Section VI concludes the paper.

II. BACKGROUND

A. Definition of Bilinear Pairings

Bilinear pairing is an important cryptographic primitive and has recently been applied in many positive applications in cryptography [9].

Let G_1 and G_2 be two cyclic groups of the same prime order *q*. We write the group laws of G_1 and G_2 additively and multiplicatively, respectively. Let *P* be a generator of G_1, assume that the discrete logarithm problems in G_1 and G_2 are hard. An efficient admissible bilinear map *e*: $G_1 \times G_1 \rightarrow G_2$ with the following properties: ①Bilinear: for all $P, P' \in G_1$ and $a, b \in Z_q^*$, $e(aP, bP') = e(P, P')^{ab}$;②Non-degenerate: there exist $P, P' \in G_1$ such that $e(P, P') \neq 1$; ③Computable: there is an efficient algorithm to compute $e(P, P')$ for any $P, P' \in G_1$.

We review related underlying mathematics problems [9] in G_1, which will server as the basis for our proposed EPP scheme.

① *Bilinear Diffie-Hellman (BDH) parameter generator:* a randomized algorithm *IG* is a BDH parameter generator if *IG* takes a security parameter *k*>0, runs in time polynomial in *k*, and outputs the description of two groups G_1 and G_2 of the same large prime order *q* and the description of an admissible pairing *e*: $G_1 \times G_1 \rightarrow G_2$

②*Discrete Logarithm (DL) Problem:* Given $P, P' \in Q_1$, for unknown $n \in Z_q^*$, compute $P' = nP$.

③*Computational Diffie-Hellman (CDH) Problem:* Given $P, aP, bP \in G_1$, for unknown $a, b \in Z_q^*$, compute $abP \in G_1$.

④Decisional *Diffie-Hellman (DDH) Problem:* Given $P, aP, bP, cP \in G_1$, for unknown $a, b, c \in Z_q^*$, decide whether *c*=*ab* mod *q*. It is know that DDH problem in G_1 is easy and can be solved in polynomial time according to $e(P, P)^{ab} = e(P, P)^c$ [9].

⑤ *Gap Diffie-Hellman (GDH) Problem:* If DDH problem in G_1 is easy and CDH problem in G_1 is hard, call G_1 is GDH.

B. Self-certified Signature by Bilinear Pairings

Shuo [8] proposed the SCS scheme using bilinear pairings, which includes KeyGen, Extract, Sign, and Verify. ①*KeyGen:* It takes a security parameter *k* as input and returns system parameters. The system parameters include two cryptographic hash functions *H* and H_1. The TA chooses a master-key *s* and computes the corresponding public key Y_{TA}. Each user chooses partial private key x_{ID} and computes the corresponding partial public key Y_{ID}. ②*Extract:* The TA generates the partial private key d_{ID} by input the system parameters, the master-key *s*, the partial public key Y_{ID} and an arbitrary $ID \in \{0,1\}^*$, the infinite set of all binary strings. Then TA sends d_{ID} securely to the user. The user can get the actual private key $< x_{ID}$, $d_{ID} >$ and the actual public key $< Y_{TA}$, *ID*, $Y_{ID} >$. ③*Sign:* An user signs any message *M* with its actual private key $< x_{ID}$, $d_{ID} >$. ④*Verify:* Any verifier can validate the signed *M* by checking the verification equation with respect to the actual public key $< Y_{TA}$, *ID*, $Y_{ID} >$.

In the SCS method, the actual public key $< Y_{TA}$, *ID*, $Y_{ID} >$ is verified implicitly through the subsequent use of the public key Y_{TA}, hence, two users can directly use the partial public keys and the corresponding partial private keys to work. It can decrease much storage space and reduce authentication message time. However, when sending the signed message together with the public key Y_{TA}, its partial public key Y_{ID} and its identify *ID*, an attacker still can link its *ID* to the user by monitoring messages. Thus, it can reveal private information regarding the activities of the user.

III. EFFICIENT PRIVACY PRESERVATION PROTOCOL USING SELF-CERTIFIED SIGNATURES

A. System Formulation

Fig.1 illustrates the network architecture, which consists of three entities: the top Trusted Authority (TA), the RSUs located at the road side, and the OBUs located on the vehicles. The TA is responsible for the registration of RSUs and OBUs and is assumed that is having sufficient computation and storage capabilities. RSUs are assumed to connect with the TA by wire or wireless links. Wireless access between vehicles as well as vehicles and RSUs is conducted through networks complying with the IEEE 802.11p standard [5].

Made assumptions are the following.

1) TA is fully trusted by all parties in the system, and is not possible for an adversary to compromise it.
2) RSUs are semi-trusted entity. The TA can inspect all RSUs at the high level. Once an RSU is compromised in one time slot, the TA can detect and take action to recover it in the next time slot [11].

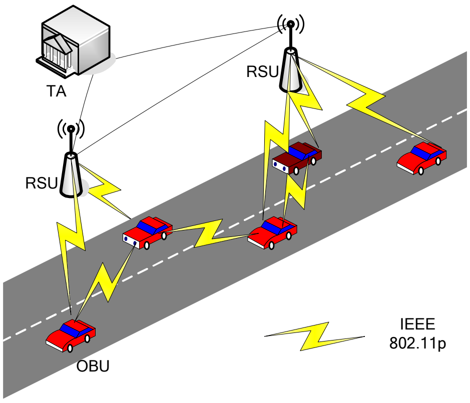


Figure 1. System model

B. EPP scheme

We assume that RSUs and OBUs are not trust. Hence, before entering into the VANET, RSUs and OBUs should be registered to the TA, and then OBUs obtain their actual public through the TA and RSUs. Therefore, the proposed EPP protocol consists of four parts shown in Fig. 2.

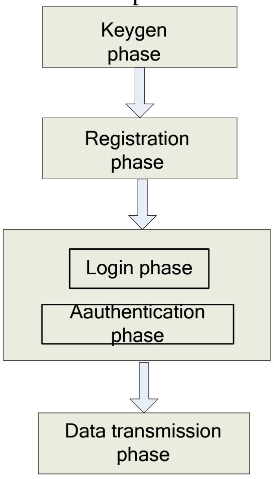


Figure 2. EPP scheme

①Keygen phase, which generates system parameters for the self-certified signatures; ②Registration phase, where RSUs & users register into TA and obtain their smart cards; ③Login/authentication phase, where a vehicular user should be authenticated before entering into the vehicular network and obtain its actual public key from the TA and RSUs and its actual private key generated by the TA and RSUs; ④Data transmission phase: where each vehicular user signs any messages by its actual private key, receivers verify the signature by using its actual public key.

1) *Keygen phase*

Let k be the security parameter and a randomized algorithm *IG* be a BDH parameter generator satisfying the GDH Problem. Let G_1 and G_2 be two cyclic groups of the same prime order *q*. We write the group laws of G_1 and G_2 additively and multiplicatively, respectively. Let P be a generator of G_1, assume that the discrete logarithm problems in G_1 and G_2 are hard. An efficient admissible bilinear map is e: $G_1 \times G_1 \to G_2$ .The TA first generates the bilinear parameters (*q*, G_1, G_2, *e*, *P*) by running the randomized algorithm *IG*. Then, the TA chooses two cryptographic hash functions: $H, H_1 : \{0,1\}^* \to Z_q^*$, and random selects $s \in Z_q^*$ as its private key, and computes $Y_{TA} = sP$ as its public key. The system parameters will be published, which include {*q*, G_1, G_2, *e*, *P*, *H*, H_1, Y_{TA} }.

2) *Registration phase*

When a RSU or vehicle submits its identity to the TA for registering itself, the TA will do the following function:

① For a vehicle: V_i submits its identity (ID_{V_i}) and password (PW_i) to the TA. Upon receiving the registration request, TA will compute vehicle V_i ’ pseudonym $PVID_{V_i} = h(ID_{V_i})$, pseudo password $\alpha_i = h(PW_i)$, $\gamma_i = \alpha_i \oplus x_s$, $N_i = h(PVID_{V_i} \| x_s)$. Note that we use γ_i and N_i to hide the parameter x_s . Then it sends a smart card with $< h(\bullet)$, $PVID_{V_i}$, α_i, γ_i, N_i >to V_i. Here, $h(\bullet)$ is a Hash function; x_s is a secret parameter generated securely by the TA; $\oplus$ is XOR operation.

② For the RSU: R_i submits its identity (ID_{R_i}) and location (L_i) to the TA. After receiving the registration request, the TA computes $\Phi_i = h(ID_{R_i} \| L_i)$ and $\Psi_i = h(L_i) \oplus h(ID_{R_i}) \oplus x_b$, then sends a smart card with $< h(\bullet)$, $h(ID_{R_i})$, Φ_i, $h(L_i)$, $\Psi_i >$ to R_i. Note that we use Ψ_i to hide the parameter x_b. Here, $h(\bullet)$ is a Hash function; x_b is a secret parameter generated securely by the TA.

3) *Login/authentication phase*

Login and authentication phase is invoked when V_i wants to enter into this network. The login and authentication phase is as follows.

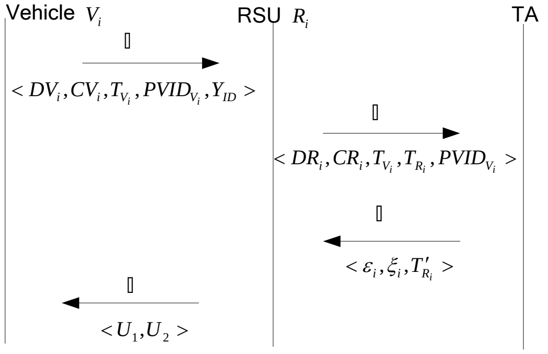
① **Login phase:**

V_i inserts its smart card to a terminal and ID_{V_i} and PW_i . The smart card validates ID_{V_i} and PW_i with the stored ones in it by the functions by the functions $h(ID_{V_i})^* \overset{?}{=} PVID_{V_i}$ and $h(PW_i)^* \overset{?}{=} \alpha_i$, and then computes $x_s{}^* = \gamma_i \oplus h(PW_i)^*$ and verifies $h(h(ID_{V_i}) \| x_s^*) \overset{?}{=} N_i$. If it is not true, the smart card terminates the process and send rejection message to V_i .

R_i inserts its smart card to a terminal and ID_{R_i} and L_i . The smart card validates ID_{R_i} and L_i with the stored ones in it by the function $h(L_i)^* \overset{?}{=} h(L_i)$ and $h(ID_{R_i})^* \overset{?}{=} h(ID_{R_i})$. If it is not true, the smart card terminates the process and send rejection message to R_i . Finally, R_i can obtain x_b^* by computing $x_b^* = \Psi_i \oplus h(L_i) \oplus h(ID_{R_i})$.

② **Authentication phase**

Authentication phase is invoked when V_i wants to enter into this network. The authentication phase is shown as Fig.3. The details are as follows.


Vehicle V_i
RSU R_i
TA
▯
$< DV_i, CV_i, T_{V_i}, PVID_{V_i}, Y_{ID} >$
▯
$< DR_i, CR_i, T_{V_i}, T_{R_i}, PVID_{V_i} >$
▯
$< \varepsilon_i, \xi_i, T'_{R_i} >$
▯
$< U_1, U_2 >$

▯ : $DV_i = h(\alpha_i \| PVID_{V_i}) \oplus h(T_{V_i} \| x_s^*)$ $CV_i = h(N_i \| \gamma_i \| T_{V_i})$

▯ : $DR_i = h(DV_i \| CV_i \| \Phi_i) \oplus h(x_b^* \| T_{R_i} \| T_{V_i})$

$CR_i = h(h(\Psi_i \| DV_i \| CV_i) \| x_b^* \| T_{R_i} \| T_{V_i})$

▯ : $\varepsilon_i = s \oplus h(ID_{R_i} \| L_i \| T'_{R_i} \| PVID_{V_i})$

$\xi_i = h(s \oplus h(ID_{R_i} \| L_i))$

▯ : $U_1 = rP$ $U_2 = d_{ID} + rY_{ID}$

Figure 3. Authentication phase

Step 1: The smart card in V_i picks a random $x_{ID} \in Z_q^*$ as its partial private key and sets $Y_{ID} = x_{ID}P$, then will perform

·Compute V_i ' dynamic login identity

$$DV_i = h(\alpha_i \| PVID_{V_i}) \oplus h(T_{V_i} \| x_s^*)$$

·Compute $CV_i = h(N_i \| \gamma_i \| T_{V_i})$

·Send $< DV_i, CV_i, T_{V_i}, PVID_{V_i}, Y_{ID} >$ to the RUS R_i , where T_{V_i} is the current timestamp when sending the message.

Step 2: After receiving the login message, R_i will perform as follows:

·Verify $(T - T_{V_i}) \le \Delta T$, if it dose not hold, abort the process. Otherwise, store $PVID_{V_i}$ and Y_{ID} , where T is the current timestamp when receiving the message, ΔT denotes the expected time interval for the transmission delay.

·Compute R_i ' dynamic login identity

$DR_i = h(DV_i \| CV_i \| \Phi_i) \oplus h(x_b^* \| T_{R_i} \| T_{V_i})$, where T_{R_i} is the current timestamp when sending the message.

·Compute $CR_i = h(h(\Psi_i \| DV_i \| CV_i) \| x_b^* \| T_{R_i} \| T_{V_i})$

·Send $< DR_i, CR_i, T_{V_i}, T_{R_i}, PVID_{V_i} >$ to the TA.

Step 3: Upon receiving the message, the TA will perform as follows:

·Verify $(T - T_{R_i}) \le \Delta T$, if it dose not hold, abort the process.

·Compute

$$DV_i^* = h(h(PW_i) \| PVID_{V_i}) \oplus h(T_{V_i} \| x_s)$$

·Compute

$$CV_i^* = h(h(PVID_{V_i} \| x_s) \| (h(PW_i) \oplus x_s) \| T_{V_i})$$

·Compute

$$h(DV_i^* \| CV_i^* \| h(ID_{R_i} \| L_i))^* = DR_i \oplus h(x_b \| T_{R_i} \| T_{V_i})$$

·Compute

$$CR_i^* = h(h(DV_i^* \| CV_i^* \| (x_b \oplus h(ID_{R_i}) \oplus h(L_i))) \| x_b \| T_{R_i} \| T_{V_i})$$

·Verify $CR_i^* \overset{?}{=} CR_i$, if it is not true, reject the message.

·Compute $\varepsilon_i = s \oplus h(ID_{R_i} \| L_i \| T'_{R_i} \| PVID_{V_i})$

·Compute $\xi_i = h(s \oplus h(ID_{R_i} \| L_i))$

·Send message $< \varepsilon_i, \xi_i, T'_{R_i} >$, where T'_{R_i} is the current timestamp when sending the message.

Step 4: Upon receiving the message, R_i will perform as follows:

·Verify $(T - T'_{R_i}) \le \Delta T$, if it dose not hold, abort the process.

·Compute $s^* = \varepsilon_i \oplus h(ID_{R_i} \| L_i \| T'_{R_i} \| PVID_{V_i})$

·Compute $\xi_i^* = h(s^* \oplus \Phi_i)$

·Verify $\xi_i^* \overset{?}{=} \xi_i$, if it dose not hold, abort the process.

·Compute $H_{ID} = H(Y_{TA}, PVID_{V_i}, Y_{ID}) \in G_1^*$, and sets the partial private key $d_{ID} = s^* H_{ID}$.

·Choose random integer $r \in Z_q^*$ and compute $U_1 = rP$, $U_2 = d_{ID} + rY_{ID}$

·Send message $< U_1, U_2 >$ to V_i .

Step 5: After receiving the message, V_i will perform as follows:

·Compute $d_{ID}^* = U_2 - x_{ID}U_1$

·Verify $e(d_{ID}^*, P) \overset{?}{=} e(H(Y_{TA} \| PVID_{V_i} \| Y_{ID}), Y_{TA})$, if it dose not hold, abort the process. Otherwise, d_{ID}^* is the secret certificate of the TA's public key Y_{TA} , the partial public key Y_{ID} and the vehicle V_i ' $PVID_{V_i}$. Thus, the V_i obtains his actual private key (x_{ID}, d_{ID}^*). Hence, the actual public key ($Y_{TA}, PVID_{V_i}, Y_{ID}$) is used as the private key for signing.

4) Data transmission phase

Step 1: To sign a message M_j, V_i randomly chooses an integer $a \in Z_q^*$ and performs

·Compute $H_{ID} = H(Y_{TA} \| PVID_{V_i} \| Y_{ID})$

·Compute $R = aY_{TA}$

·Compute $f = H_1(M_j \| R \| H_{ID} \| PVID_{V_i})$

·Compute $\beta = fad_{ID} + x_{ID}H_{ID}$

·Send the message

$$< \beta, R, PVID_{V_i}, Y_{ID}, M_j, T_1 >$$

Step 2: To verify the signature (R, β), the verifier V_{i+1} will perform

·Verify $(T - T_1) \le \Delta T$, if it dose not hold, abort the process.

·Compute $H_{ID} = H(Y_{TA} \| PVID_{V_i} \| Y_{ID})$

·Compute

$e(\beta, P) \overset{?}{=} e(H_1(M_j \| R \| H_{ID} \| PVID_{V_i})R + Y_{TA}, H_{ID})$, if it dose not hold, abort the process.

Hence, if two vehicles follow this protocol, the verifier will always accept the signature (R, β) and be convinced of the authenticity of the partial public key of V_i.

IV. PERFORMANCE ANALYSES

A. Security Analysis

1) Resilience to stolen smart attack

Assume that the smart card of a vehicle is stolen or lost, then the attacker can extract the secret information {$h(\bullet)$, $PVID_{V_i}$, α_i, γ_i, N_i} from it by side channel attacks and invasive attacks [12]. However, even though an adversary taking control of the smart can obtain γ_i, it is practically infeasible for the adversary to know x_s by inferring γ_i or N_i, because of the one-way property of $h(\bullet)$. Therefore, the attacker cannot generate a valid login message DV_i' ($DV_i' = h(\alpha_i \| PVID_{V_i}) \oplus h(T_{V_i} \| x_s)$).

For the RSU R_i, the attacker can obtain secret information {$h(\bullet)$, $h(ID_{R_i})$ Φ_i, $h(L_i)$, Ψ_i} from the smart card, however, he cannot forger a fake message DR_i' ($DR_i' = h(DV_i \| CV_i \| h(ID_{R_i} \| L_i)) \oplus h(x_b^* \| T_{R_i} \| T_{V_i})$) without knowing x_b.

2) Resilience to guessing attack

Guessing attack is a crucial concern to any password-protected system [13]. Our scheme can resist the guessing attack, since the communication units within vehicles do not contain password and IDs. The attacker might try different passwords in its effort to construct DV_i', however, the probability of failing is very high, because dose not have knowledge of x_s.

3) Resilient to replay attack

Assume that the attacker intercepts a valid login message $< DV_i, CV_i, T_{V_i} >$ and tries to login to the RSUs by replaying the same message. The verification of this login message fails because the interval $(T' - T_{V_i}) > \Delta T$ (ΔT denotes the expected time interval for the transmission delay), where T' is RSU' s system time when receiving the replayed message.

We also include a timestamp in each data packet in order to verify during the data transmission phase the message's validity. Thus, the replay attack is also prevented.

4) Validity

In the data transmission phase, the receiver can compute

$$\begin{aligned} e(\beta, P) &= e(H_1(M_j \| R \| H_{ID} \| PVID_{V_i})ad_{ID} + x_{ID}H_{ID}, P) \\ &= e((H_1(M_j \| R \| H_{ID} \| PVID_{V_i})as + x_{ID})H_{ID}, P) \\ &= e((H_1(M_j \| R \| H_{ID} \| PVID_{V_i})as + x_{ID})P, H_{ID}) \\ &= e(H_1(M_j \| R \| H_{ID} \| PVID_{V_i})R + Y_{TA}, H_{ID}) \\ &= e(H_1(M_j \| R \| H_{ID} \| PVID_{V_i})R + Y_{TA}, H(Y_{TA} \| PVID_{V_i} \| Y_{ID})) \end{aligned}$$

Hence, the proposed EPP is validity.

5) Resilient to forgery attack

Theorem1. The proposed EPP scheme is unforgeable under the assumption of the *DL* problem.

Proof. If the attacker wants to forge a signature (R', β'), he has to make sure the following verification correct:

$$e(\beta', P) \overset{?}{=} e(H_1(M_j \| R \| H_{ID} \| PVID_{V_i})R + Y_{TA}, H(Y_{TA} \| PVID_{V_i} \| Y_{ID}))$$

If the attacker knows $(H_1(M_j \| R \| H_{ID} \| PVID_{V_i})R + Y_{TA})$ be the discrete logarithm problem in *P*, given $r \in Z_q^*$, then assume that $\beta = rH_{ID}$, where $r = (H_1(M_j \| R \| H_{ID} \| PVID_{V_i})as + x_{ID})$. The attacker knows *a* and computes $H_1(M_j \| R \| H_{ID} \| PVID_{V_i})$, however, he dose not know *s* and x_{ID} by computing $Y_{TA} = sP, Y_{ID} = x_{ID}P$ because of the discrete logarithm problem in G_1 and G_2.

B. Time Complexity Analysis

Assume that T_{pmul} represents the time for one point multiplication computation in G_1; T_{padd} represents the time for one point addition computation in G_1; T_{pair} denotes the time for one pairing computation, and T_{hash} denotes that the time for one hash function. Note that the time complexity for other computation operations, such as the multiplication in Z_q^*, are ignored, since they are much smaller than T_{pmul}, T_{padd}, T_{pair} and T_{hash} [10]. Table I shows the time complexity of the proposed EPP scheme and the SCS scheme [8]. Compared to the SCS scheme in signing a message, the proposed EPP scheme reduces the time of ($T_{pair} + T_{padd}$). From the paper [10], we can see that T_{pair} is much larger than other operations.

TABLE I TIME COMPLEXITY

Scheme	Sign Message	Verify Message
SCS scheme	T_{pair} +3 T_{pmul} +2 T_{padd} +2 T_{hash}	2 T_{pair} +2 T_{pmul} + T_{padd} +2 T_{hash}
EPP scheme	3 T_{pmul} + T_{padd} +2 T_{hash}	2 T_{pair} + T_{pmul} + T_{padd} +2 T_{hash}

V. RELATED WORK

User authentication is very important feature in a VANET as the source of the information should be verified to ensure the legitimacy of the communicated data [3-5]. To address such issues in VANETs, Raya et al. [1] introduced a security protocol for VANETs by installing a large number of private keys and their corresponding anonymous certificates to each vehicle. Instead of taking any real identity information of the drivers, these anonymous certificates are generated by taking the pseudo IDs of the vehicles. Existing PKI-based security schemes are prohibitively inefficient due to their computational complexity and obviously cannot scale to large vehicle populations.

A possible approach to reduce the overhead of the PKI-based security schemes is to improve the verification efficiency. We can do so by using the short group signatures method [14, 15], since it can quickly verify a large number of signatures simultaneously instead of sequentially by decreasing the number of some principal time-consuming operations. However, these methods assume that all verified signatures are authentic, and therefore, they need to be optimized for realistic applications, where bogus signatures commonly exist.

A short-time signature using bilinear pairings was proposed in [6] to use in the electronic cash system, which allows a user to get a signature without giving the signer any information about the actual message or the resulting signature. Liu et al. [11] proposed an efficient conditional private preservation protocol which authenticates users & RSUs by using Wei or Tate pairings on the elliptic curves and authenticates data messages by the short-time anonymous keys within certificates. However, the computing overhead to authenticate users & RSUs is still much high (the execution time (computing time) of authenticating users & RSUs is about 34.8msec in paper [11]). Liu et al. [16] proposed a secure protocol based on group signature and identity-based signature techniques by bilinear pairings. A signature based on identity is adopted in the RSUs to digitally sign each message by the RSUs, which reduces signature overhead.

VI. CONCLUSIONS

In this paper, we presented an efficient privacy preservation protocol by using smart card functionality and bilinear pairings method for secure communications in vehicular network. Since users sign messages by their actual private keys and verify these messages only with their actual public keys no corresponding certificates, the EPP protocol has been identified to be not only capable of providing the conditional privacy preservation that is critically demanded in the VANET applications, but also achieves high efficiency in terms of time complexity.

ACKNOWLEDGMENT

This work was supported by the Government of Ontario under the ORF-RE WISENSE project (3074600) and the Natural Sciences and Engineering Research Council (NSERC) of Canada under NSERC Grant 193961-2006.

REFERENCES

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

[2] Y. Peng, Z. Abichar, and J. M. Chang, “Roadside-aided routing (RAR) in vehicular networks,” in Proc. IEEE ICC 2006, Istanbul, Turkey, vol. 8, pp. 3602-3607, June 2006.

[3] K. Ren, W. Lou, R. H. Deng, and K. Kim, “A novel privacy preserving authentication and access control scheme in pervasive computing environments,” IEEE Transaction on Vehicular Technology, vol. 55, no. 4, pp.1373-1384, July 2006.

[4] H. Moustafa, G. Boudron, and Y. Gourhand, “AAA in vehicular communication on highways with ad hoc networking support: a proposed architecture,” in Proc. International Workshop on Vehicular Ad Hoc Networks (VANET), Germany, 2005.

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

[6] D. Chaum, “Blind signatures for untraceable payments,” in Proc. Advances in Cryptology - Crypto’82, Santa Barbara, California, USA, pp. 199-203, Aug. 1982.

[7] M. Girault, “Self-certified public keys,” in Proc. Advances in Cryptology- Eurocrypt’91, LNCS 1440, Springer-Verlag, Brighton, UK, pp. 491-497, 1991.

[8] Z.H. Shao, “Self-certified signature scheme from parings,” The Journal of Systems and Software, 2007, 80 (2) : 388 - 395.

[9] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil paring,” in Proc. Advances in Cryptology - Asiacrypt’01, LNCS 2248, Springer-Verlag, Gold Coast, Australia, pp. 514-532, 2001.

[10] “Crypto++ 5.6.0 Benchmarks,” http://www.cryptopp.com/benchmarks. html.

[11] R. X. Liu, et al., “ECPP: efficient conditional privacy preservation protocol; for secure vehicular communications,” The 27th Conference on Computer Communications. IEEE, 2008, pp. 1229-1237.

[12] K. Markantonakis, et al., “Attacking smart card systems: theory and practice,” Information Security Technical Report, Vol. 14, Issue 2, May 2009, pp. 46-56.

[13] M.L. Das, “Two-factor user authentication in wirelesssensor networks,” IEEE Trans. Wireless Comm. 2009, 8, pp. 2450-2459.

[14] D. Chaum and E. van Heyst, “Group signatures,” in Advances in Cryptology - EUROCRYPT 1991, LNCS 547, pp. 257-265, Springer-Verlag, 1991.

[15] D. Boneh, X. Boyen, and H. Shacham, “Short group signatures,” in Advances in Cryptology - CRYPTO 2004, LNCS 3152, pp. 41-55, Springer-Verlag, 2004.

[16] X. D.Liu, et al., “GSIS: a secure and privacy-preserving protocol for vehicular communications,” IEEE Transactions on Vehicular Technology, 56(6), pp. 3442-3456, 2007.