# RPD: Reusable Pseudo-id Distribution for a Secure and Privacy Preserving VANET

**Sulaiman Ashraph**
**Department of Computer Science, National University of Rwanda**
**Butare, Rwanda**

**S.V. Kasmir Raja**
**Dean of Directorate of Research, SRM University**
**Chennai, India**

**and**

**Sung Han Park**
**Dean of Faculty of Applied Sciences, National University of Rwanda**
**Butare, Rwanda**

## ABSTRACT

In any VANET, security and privacy are the two fundamental issues. Obtaining efficient security in vehicular communication is essential without compromising privacy-preserving mechanisms. Designing a suitable protocol for VANET by having these two issues in mind is challenging because efficiency, unlinkablity and traceability are the three qualities having contradictions between them. In this paper, we introduce an efficient Reusable Pseudo-id Distribution (RPD) scheme. The Trusted Authority (TA) designating the Road Side Units (RSUs) to generate *n* reusable pseudo ids and distribute them to the On Board Units (OBUs) on request characterizes the proposed protocol. RSUs issue the aggregated hashes of all its valid pseudo-ids along with a symmetric shared key and a particular pseudo-id to each vehicle that enters into its coverage range. Through this the certificates attached to the messages can be eliminated and thus resulting in a significantly reduced packet size. The same anonymous keys can then be re-distributed by the RSUs episodically to other vehicles. We analyze the proposed protocol extensively to demonstrate its merits and efficiency.

**Keywords**: VANET, privacy, security, unlinkability, traceability, pseudonyms.

## 1. INTRODUCTION

Vehicular communication (VC) systems are developed as a means to enhance road safety, traffic management and infotainment facilities for drivers and passengers. In vehicular ad hoc networks, each vehicle is equipped with a communication device known as On Board Units (OBUs) that facilitate them to communicate with other vehicles, RSUs located on the road at different points and the TA (trusted authority) as well. In general, OBUs frequently broadcasts routine traffic related messages [1] with information about its position, current time, direction, speed, acceleration/deceleration, traffic events, etc. This helps the vehicle to be warned with critical situations such as accidents, traffic jams and so on, in addition with predicting the movements of the nearby vehicles.

Though this communication helps the driver community, it has a critical side effect of privacy. An attacker can easily track the physical location of a vehicle using these messages just by eavesdropping the communication. Tracking the movements of a vehicle such as “Big brother syndrome” is another case. One approach to solve this problem is that the vehicles broadcast their messages under pseudonyms that they change with some frequency [2]. The pseudonym-based approach that has been proposed by [3],[ 4] is an idea to help the vehicles exchange their communications without revealing their real identity. Many studies have contributed for this approach. One of them is Baseline Pseudonyms (BP) approach that stores a huge number of pseudonyms in the OBU [4], [5]. Another is the Hybrid approach (HP) that combines both BP and Group Signature (GS) approaches [6]. In this approach, pseudonyms are generated on board and used for sending messages by attaching a group certificate.

In all the pseudonym-based approaches that are previously discussed, pseudonyms generated are discarded soon after their lifetime. This cause the pseudonym providers to generate pseudonyms every now and then upon the request from the vehicles. Though generation of pseudonyms by the TA or RSUs is not an issue with their high computation and storage capacity, the computation cost of OBUs on signature and certificate generation and verification grows linearly with the traffic density, since every message comprises of a public key, a signature using its private key and a certificate on the public key essentially. In order to address this problem we propose a reusable anonymous key distribution scheme in which the pseudo-ids are generated in bulk by the RSUs and issued to the OBUs in its coverage zone by attaching a token with it. This token contains a hashed value of the given pseudo-id sealed with the long-term public key of the vehicle that receives the pseudo-id. RSU also disclose the aggregated hashes of all its valid pseudo-ids generated by it to the vehicles in its range in order to facilitate them knowing the authentication of the pseudo-ids the messages are sent from. Session keys are generated by the RSUs to communicate with vehicles to share this information. Therefore, the proposed scheme avoids the attachment of certificate with every message and by this way cuts down the cost of verification.

On the other hand, while sending safety messages the vehicles will attach the token provided along with the pseudo-id. This token is embedded with messages merely for future traceability.

## 2. OUR APPROACH

### 2.1 System Model

cryptographic materials for each OBUs and RSUs such as $q$, G, $G_T$, ê, $P_1$, $P_2$. TA randomly selects a master secret key $s \in Z_q$ and computes $U_1 = sP_1$ and $U_2 = sP_2$ as its public keys. TA also chooses a cryptographic hash functions H: $(0, 1)^* \rightarrow G$. Each RSU and vehicle are preloaded with the public parameters $q$, G, $G_T$, ê, $P_1$, $P_2$, $U_2$, H.

| Notation | Description |
|---|---|
| $s$ | TA's master secret key |
| $U_1$,$U_2$ | TA's public keys |
| $V_i$ | the $i$-th vehicle |
| R | the RSU |
| $RID_v$ | real ID of the vehicle |
| $RID_R$ | real ID of the RSU |
| $PK_{vi}$ | long term public key of $V_i$ |
| $sk_{vi}$ | corresponding private key of $PK_{vi}$ |
| $T_{exp}$ | time expiry |
| $Cert_{TA}[PK_{vi}]$ | TA's certificate on the public key of $V_i$ |
| $LOC_R$ | Location of RSU |
| $LID_R$ | Location ID of RSU |
| $PID_i$ | $i^{th}$ short term pseudo-id |
| $PK_R$ | public key of RSU |
| $sk_R$ | corresponding private key of RSU, |
| $K_s$ | shared session key between V and RSU |
| $E_x$ | encryption using the key x |
| $D_x$ | decryption using the key x |
| $T_{issue}$ | issue time |
| $T_{return}$ | return time |
| $\tau$ | token |
| $ack_{termination}$ | acknowledgement message for termination request |
| $h(.)$ | a one way hash function such that SHA-1[24] |

*Table 1: Notations*

The proposed protocol could be explained in five stages: registration and anonymous key generation, distribution of aggregated hashes, message generation, message validation and id traceability and revocation list. The key generation and mutual authentication between RSUs and vehicles of this protocol is based on [21]. For easy understanding the notations used throughout this paper are listed in table1.

### 2.2. Registration and Anonymous key generation

**a) Key generation by TA:** All the vehicles and RSUs must register themselves with the TA before they join in the VANET. Each vehicle is assigned with a real identity $RID_{vi} \in G$. We assume that the TA is in-charge of checking the vehicle's identity, generating a long-term public/private key pair for each vehicle and loading it into its OBU. TA chooses a random private key $sk_{vi} \in Z_q$ for the vehicle and computes $PK_{vi} = sk_{vi}P_1$ as its long-term public key.. It stores $RID_v$, $PK_{vi}$, and $sk_{vi}$ and in its database for future traceability and returns $PK_{vi}$ , $sk_{vi}$ , $Cert_{TA}[PK_{vi}]$ to the vehicle.

The registration of RSUs with TA is very similar to that of vehicle registration. Firstly, the RSU sends its real-id $RID_R$ and its location information $LOC_R$ to the TA. TA selects a public/private key pair $PK_R$, $sk_R$ for the RSU also stores $LID_R$, $PK_R$, $sk_R$ in its database for reference.

**b) Key generation by RSUs:** In this phase, the RSU is responsible for generating '*n'* number of pseudo-ids $PID_1$, $PID_2$….. $PID_n$ for the vehicles. The PID is encrypted here using ElGamal encryption algorithm [22] over the Elliptic curve Cryptography [23]. In each pseudo-id, a random nonce and is changed to guarantee a distinct PID.

### 2.3. Distribution of token and aggregated hashes of pseudo-ids

The proposed RPD protocol comprises of four phases: pre-authentication phase, mutual authentication phase, key distribution phase and token return phase as illustrated in figure 3. The detailed explanation of the proposed protocol is as follows:

**1) Pre-Authentication Phase***:* In this phase, the RSU generates '*n'* pseudo-ids and stores them in its pseudo-id table as shown in table2. It also computes the hash values of all the pseudo-ids generated by it and aggregates all the hashes, i.e., $h_{aggr} = h(PID_1) \| h(PID_2)….. \| h(PID_n)$.

| pseudo_id | $T_{gen}$ | Status |
|---|---|---|
| $PID_1$ | $t_1$ | 1 |
| $PID_2$ | $t_1$ | 0 |
| ⋮ | ⋮ | ⋮ |
| $PID_n$ | $t_1$ | |

*Table 2: pseudo-id table*

**2) Authentication Phase***:* At regular intervals, RSU broadcasts a *hello* message M, its real id $RID_R$ and its public key $PK_R$ by signing them using its private key $sk_R$. R computes its signature $\sigma_R$ using (ω, Q) on the *hello* message M as follows.

$Q = nP_2$ , $n \in Z_q$
$\omega = sk_R - nH(Q) \in Z_q$
$msg_1 = (RID_R, PK_R, M, \sigma_R(M), Q, \omega, T_s)$

When the vehicle $V_i$ enters into the communication range of the RSU R, it detects the public key $PK_R$ of R through this message. Note that $V_i$ uses this message only at the first time to obtain the symmetric key with R, other vehicles that are already inside the RSU range ignores the message. $V_i$ verifies the location information attached by default with the RSU message by matching the location information of R through GPS. If both are matching then, $V_i$ checks the public key of R for its trustworthiness.

Once the RSU is authenticated by $V_i$, $V_i$ generates a random number $r_1 \in Z_q$ and computes $r_1P_1 \in G$ as its share for the session key $K_s$ , $H(msg_1)sk_{vi}$ as its signature $\sigma_{vi}$ and forms a request for the session key and pseudo-id, signs them using its private key. $V_i$ then submits its credentials that include its long-term public key obtained from the TA and its request to R after encrypting them using the public key $PK_R$ of R.

$msg_2 = E_{PK_R}(V_{id} \| N \| T_s)$

where, N = ($r_1P_1$,$req_1$, $\sigma_{vi}$ ($r_1P_1$ || $req_1$)) , $r_1 \in Z_q$ and $V_{id}$ is the vehicle's credential (see phase II of RPD protocol) and $req_1$ is the pseudo-id request.

The authentication on the other hand is as follows: RSU *R* scans the revocation list each time a new vehicle tries to associate with it. Thus, on the reception of $msg_2$ from $V_i$, R decrypts $msg_2$ using its delegate private key $sk_R$ and checks $V_i$'s public key in the revocation list and the freshness of the timestamp attached with the message. If the public key is not revoked, *R* checks whether the signature $\sigma_{vi}$ of $V_i$ is legitimate.

**3) Key Distribution Phase:** After authenticating $V_i$ , R randomly picks a pseudo-id (whose status is 0) from the pseudo-id table, and chooses $r_2$ for the selection of session key $K_s$. R then computes a token Ʈ = H($PID_i$ || $PK_{vi}$|| $T_{issue}$) to bind the long term public key of $V_i$ with the pseudo-id $PID_i$ temporarily. R stores the token, pseudo-id, $V_i$'s public key along with the token issue time as shown in the first four columns of table3. Note that, the records of the token table are wiped out after a certain period of time (may be once in a week or two), in order to avoid the table growing linearly. Then, *R* encrypts the pseudo-id, token and the aggregated hashes of all its pseudo-ids by using the shared session key and sends to $V_i$. Once $V_i$ receives the message from *R* it calculates the session key $K_s$ and decrypts the message using it. Vehicle $V_i$ now holds the pseudo-id and uses it for sending messages to other vehicles.

| Token | pseudo_id | V's public key | $T_{issue}$ | $T_{return}$ |
|---|---|---|---|---|
| $Ʈ_1$ | $PID_1$ | $PK_{vi}$ | $t_1$ | $t_1+t$ |
| $Ʈ_2$ | $PID_2$ | $PK_{vj}$ | $t_2$ | $t_2+t$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $Ʈ_n$ | $PID_n$ | $PK_{vn}$ | $t_n$ | $t_n+t$ |

*Table 3: token table*

**4) Token return Phase:** Since the long-term public key of the vehicle is bound with the pseudo-id it is provided, vehicle $V_i$ must return the token to the RSU after its usage. In this phase a vehicle may pass two types of request to the RSU. *req1* is a request for new pseudo-id, which must be sent to the RSU when $V_i$ wants to change its pseudo-id. In such case the RSU extracts the vehicle's public key from the old pseudo-id and rebinds the public key with another pseudo-id to construct a new token and issues the new id along with its token to the vehicle by encrypting it using the shared session key. *req2* is another type of request the vehicle sends to RSU when it goes out of the range of the RSU or when it receives a *hello* message from another RSU. This request can be called as a handover request to make itself free from bonds with that RSU. In either case, the RSU will respond by giving a new pseudo-id or a handover acknowledgement message based on the type of the request it received.

*Key Reusability* is the main advantage of the proposed scheme when compared to other studies as the RSU's burden on continuous pseudo-id generation is considerably reduced because of reusing the same key for many vehicles. Upon receiving *req1,* the RSU uses the token to extract the public key of the vehicle, resets the status of the corresponding pseudo-id of token (i.e status= 0 for the corresponding pseudo-id in table 2). This pseudo-id can be reused later, by binding with another vehicle's public key upon request.

***I. Pre-Authentication Phase:***

R : computes $PID_i$ where i={1…n}
R : computes $h_{aggr} = h(PID_1)|| h(PID_2)….. || h(PID_n)$,

***II. Mutual Authentication Phase:***

R : computes Q= $nP_2$ , $n \in Z_q$
R : computes $\omega = sk_R - nH(Q) \in Z_q$
R : computes $\sigma_R$ as (Q, ω)
R : broadcasts $msg_1$ = (M ,$PK_R$ ,$\sigma_R$ (M || $PK_R$), Q, ω, $T_s$)
$V_i$ : checks $\sigma_R$ to authenticate R

$V_i$ : computes $V_{id}$ = ($PK_{vi}$ || $WR_{PKvi}$|| $Cert_{TA}[PK_{vi}]$)
$V_i$ : computes $\sigma_{vi}$ as H($msg_1$)$sk_{vi}$
$V_i$ : computes N = ($r_1P_1$ , $req_1$ , $\sigma_{vi}$ ($r_1P_1$ || $req_1$)) , $r_1 \in Z_q$
$V_i$ : computes $msg_2 = E_{PK_R}$ ($V_{id}$ || N || $T_s$)

$V_i$ → R : $msg_2$
R : $D_{skR}(msg_2)$
R : verifies $PK_{vi}$ and authenticates $V_i$

***III. Key Distribution Phase:***

R : computes session key $K_S = r_1r_2P_1$
R : picks $PID_i$ from pseudo-id table
*(where, 1≤ i ≤n and status($PID_i$)=0)*
R : sets status($PID_i$)=1
R : computes O = ($PID_i$, $T_{issue}$ , $r_2$)
R : sets Ʈ = $h$($PID_i$ || $PK_{vi}$|| $T_{issue}$) where $k \in j$
R : computes $msg_3 = E_{Ks}$(O, Ʈ, $h_{aggr}$ )
R→ $V_i$ : $msg_3$ , $r_2$
$V_i$ : computes $K_s = r_1r_2P_1$
$V_i$ : computes $D_{Ks}$ ($msg_3$)
$V_i$ : holds $PID_i$ and $h_{aggr}$

***IV. Token return Phase:***

$V_i$ : computes $msg_4 = E_{Ks}$ ($req_i$,Ʈ)
$V_i$ → R : ($PID_i$ , $msg_4$)
R : if $req_i = req_1$ then
map Ʈ in table 3 and set status($PID_i$)=in table 2 for the corresponding Ʈ
sets a new Ʈ = $h$($PID_l$ || $PK_{vi}$ || $T_{issue}$) for $V_i$ , where $l \in n$
$msg_5 = E_{Ks}$ ($PID_l$, Ʈ)
else if $req_i = req_2$ then
map Ʈ in table 3 and set status($PID_i$)=0 in table 2 for the corresponding Ʈ
$msg_5 = ack_{termination}$
R→ $V_i$ : $msg_5$
*(note: $req_1$ = new pseudo-id request ; $req_2$ = termination request)*

*Figure 1: Reusable Pseudo-id Distribution (RPD) Protocol*

**2.4. Message Generation**

Once the vehicle $V_x$ obtains the pseudo-id $PID_i$ and its token Ʈ from the RSU R, $V_x$ uses $PID_i$ to send messages to other vehicles. Each message will be composed of the pseudo-id and its bound token along with the message payload. With *M* as the message payload, the message format is:

$$msg_{v2v} = M, PID_{ik}, \sigma_M, Ʈ, TS_M$$

where, $\sigma_M$ is the ECDSA signature of the OBU on message *M*, Ʈ = $h$($PID_i$ || $PK_{vx}$ || $T_{issue}$) the token and is uniquely embedded with the pseudo-id $PID_i$ at time T. Note, in any circumstances the vehicles cannot recognize or open the token Ʈ coming along with each message, as the hash is computed by the RSU and known only by it. The reason why the token is embedded with the message is, no pseudo-id could be reused unless it is free from binding with any vehicle in the range. Also, the token facilitates the TA to revoke any misbehaved vehicle with the help of the RSU that issued that token. $TS_M$ is a timestamp chosen by the vehicle to prevent message replay attack.

Since each participant in the network is assumed to be aware of the identity privacy, they would request for the change of pseudo-ids sporadically. In such case, it is obviously the responsibility of the user (driver in this case) to handover the current token back to the RSU that generated the pseudo-id. This is because, as every pseudo-id is encapsulated with the long-term public key of the vehicle, the vehicle that obtained the particular pseudo-id is accountable for any message sent from that id. Therefore, with in the communication range of the RSU R, $V_i$ can make a pseudo-id request or a handover request. In either case, $V_x$ has to communicate with R using its shared symmetric key.

### 2.5. Message Verification

Once a message is received, the receiving vehicle generates the hash of the pseudo-id in the message and matches it with the stored aggregated hashes of pseudo ids that are obtained from the RSU. It accepts the message if a match is identified and ignores otherwise as explained in algorithm 1.

***Algorithm 1: Verification of safety messages***

***Input:*** PID of the received message
***Output:*** message acceptation or denial

*Step 1:* start
*Step 2:* computes $h(PID_i)$ for the $PID_i$ of *msg*, where $msg = M, PID_i, Ʈ$
*Step 3:* scan $h_{aggr}$ for $h(PID_i)$ *(i.e)* is $h(PID_i)$ in $\{h(PID_1) \| h(PID_2) \ldots \| h(PID_n)\}$ ?
*Step 4:* if '*yes*'
accept '*msg*'
else
drop '*msg*'
*Step 5:* stop

### 2.6. ID traceability and Revocation list

An identity disclosure is performed only when solving a dispute. In case of any dispute concerning a message, the TA first verifies the pseudo-id of the accused message to identify the RSU who issued the pseudo-id. The RSU then uses the token Ʈ in the message in order to fetch the long-term public key of the responsible vehicle in that and to verify the token issuance time. If the time in the reported message is within the time window of token issuance and handover, the corresponding public key will be added in the revocation list or any penalty is charged based on the legal considerations.

## 3. RELATED WORK

The IEEE 802.11p task group is working on the Dedicated Short Range Communications (DSRC) standards, which aims to enhance the 802.11 protocol to support wireless data communications for vehicles and road side infrastructure [7]. Many studies have been reported on the security and privacy-preservation issues for VANETs [3, 4, 8-12]. The privacy and security issues for VANET can mainly be classified into three categories.

First is based on a huge number of pseudo-anonymous key based (HAB) protocols [3, 4, 8]. Though this is a simple and straight forward solution, there found three main disadvantages [12] in HAB: (a) each OBU has to take large storage space to store a number of anonymous key pairs; (b) very time consuming for the authority to track for any problematic certificate due to the long revocation list; (c) once some OBUs' anonymous keys are revoked, it takes a long time for each OBU to update the certificate revocation list.

The second one is based on group signature (GSB) which was first introduced in [14] which allows a group member to sign messages anonymously on behalf of the group. The group manager can still reveal the identity of a signer in case of a dispute. Although the group signature can achieve anonymity on conditional privacy preservation, the time for message verification grows linearly with the number of revoked vehicles [15]. Worse, the unrevoked have to update their private keys and group public keys with the group manager when the number of revoked vehicles surpasses some predefined threshold. In [16], application of the short group signatures is suggested. Authors in [8] propose an efficient security protocol called GSIS, which is based on the group signature scheme. 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 each group signature requires at least two pairing operations, which might not be scalable when the density of traffic is increased. Finally, a hybrid pseudonym based approach [5] has been proposed by combining the baseline pseudonym scheme [3] and the group signature scheme [8] together. However this approach is also categorized as GSB, since it suffers with the same drawbacks.

The third one employs the RSUs to assist with message authentication [12, 13]. Authors in [17] propose an authentication algorithm called Group-ID Tree. In this protocol the vehicle is able to connect the RSU after proving its membership in a group. However, this leads to additional overhead in managing group membership. The protocol proposed by the authors of [18] elects a group leader who then communicates with the RSU on behalf of the group. This protocol also suffers with the disadvantage of the overhead associated with the Revocation List (RL) management required to authenticate group membership. ECPP (Efficient Conditional Privacy Preservation) [12] protocol was proposed to solve the storage requirements by using the RSU to manage the vehicle's certificate. In this protocol the RSU issues only an ephemeral certificate for valid vehicles at the time of authentication to eliminate the need for the vehicles to manage the certificates and the RL.

In [13] the authors introduced a RSU aided message authentication scheme called RAISE. RAISE is responsible for verifying the authenticity of the messages sent from the vehicles and for notifying the results back to the vehicles. They also adopted *k-anonymity* [19] to protect user identity privacy where the RSUs assign a common pseudo id to *k*-vehicles. Our work complements the RAISE and ECPP works by providing another protocol to furnish a conditional privacy preserving and a secure VANET environment.

## 4. EVALUATION

In this section, we use the ns-2 simulator 2.34 to evaluate the performance of our RPD protocol. Since the proposed protocol focuses on the signing and verification overhead, we are more concerned in the system performance of RPD in terms of throughput, message loss ratio and average end-to-end 1message delay. We simulate a traffic scenario with high vehicle density of 30-180 vehicles. The ECDSA and the group signature verification delays are 3.87ms and 11 ms respectively [25]. The simulation script is written in TCL using DSDV protocol. The traces are recorded and analyzed using *awk* utility in Linux (Fedora 14).

| **Simulation Setup** | |
|---|---|
| Physical and MAC model | IEEE 802.11a standard |
| Nominal bit rate | 2Mbps |
| Transmission Range | 300m |
| Number of nodes | 30-180nodes |
| Simulation duration | 1000 seconds |
| Simulation area | 1500m x 300m |
| Traffic Type | CBR |
| Routing Protocol | DSDV |
| Packet Size for OBU message | 208 |

*Table 4: NS-2 Simulation Parameters*

### a) Throughput

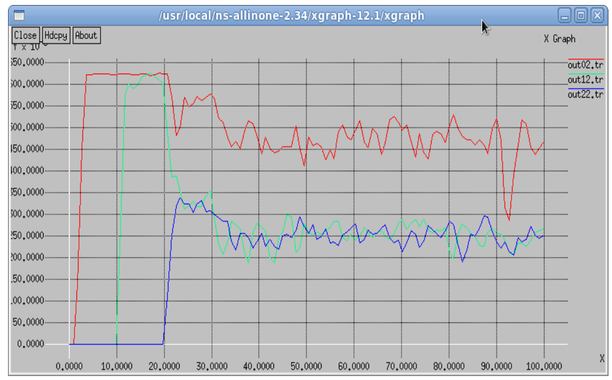
Throughput is the average rate of successful message delivery over a communication channel. Throughput is usually measured in bits/sec or data packets/sec. The throughput of a protocol varies based on the cryptographic operations involved in securing the message and the transmission overhead. Mainly the length of the authority certificate on the public key possesses the additional overload to every signed message.

In ECDSA [20] which is accepted as the most appropriate candidate for VANET in terms of packet overload and verification delay, the total length of a signed packet is around 281 bytes, in which the additional overhead for each message is 181 bytes due to the cryptographic operations. With the group signature based scheme GSIS [8] the additional communication overhead is 184 bytes [26]. According to IBV scheme [9] a pseudo-id $PID_i = PID_{i1} + PID_{i2}$ possesses a total length of 42 bytes. With RPD the total message length is calculated as follows:

$$
\begin{aligned}
L_{msg} &= L_M + L_{PID} + L_{Sign} + L_T + L_{TS} \\
&= 100 + 42 + 42 + 20 + 4 \\
&= 208 \text{ bytes}
\end{aligned}
$$

According to [1], the message M occupies 100 bytes. Therefore, the additional overhead is only 42+42+20 bytes, which is comparatively lower for the OBU. In addition, RPD does not require the revocation list stored in OBUs, which makes the protocol free from increase in its storage overhead with the increase in number of revoked public keys. On the other hand, the additional transmission overhead on RSU is $(20*n)/m$ bytes along with the parameters for mutual authentication, where 20 B is the length of a $h(PID_i)$ sent by the RSU which is multiplied by $n$ for $n$ aggregated PIDs. This $20*n$ are shared by $m$ messages, because in RPD the $n$ pseudo-ids are hashed and sent as an aggregated hash ($h_{aggr}$) only once for an RSU range during the symmetric key establishment and thus it is considered as negligibly small.

Figure 2 shows the throughputs of Group based, PKI and RPD schemes over a period of 100 sec with a traffic density of 50 vehicles. We can see that when compared to the traditional PKI based ECDSA scheme and the group signature based scheme, RPD has very high throughput. This is because; the certificate attached dominates the length of the overhead and thus reduces the throughput. The advantage gained by the proposed scheme is obvious, since no certificate is attached with the message. (*x-axis: Time in seconds, y-axis: Data Packets in bits)*




*out02.tr=RPD Scheme*
*out12.tr=PKI Based Scheme*
*out22.tr=Group Based Scheme*

*Figure 2: Protocol throughputs of Group Based, PKI and RPD schemes(100secs)*

### b) Message Loss ratio

One among the main performance metrics considered is the average message loss ratio, which is to be denoted as $MSG_{L\_ratio}$. A message is lost only if the queue of messages is full when the message verification rate is much lower than the message arrival rate. As defined in [25] the $MSG_{L\_ratio}$ can be expressed as,

$$MSG_{L\_ratio} = \frac{1}{N}\sum_{n=1}^{N}\frac{M_{cons}^{i}}{M_{recv}^{i}}$$

Where $N_s$ represents the total number of vehicles in the simulation and $N_c$ represents the number of vehicles in one hop communication range of the vehicle *i*. $M_{recv}^{i}$ represents the total number of messages received by the vehicle *i* in the medium access control layer, $M_{cons}^{i}$ represents the total number of messages received by the vehicle *i* in the application layer. Here, we only consider the message loss incurred by the security protocol rather than the loss caused by the wireless communication between the RSU and vehicles.

Figure 3 shows the relationship between the PLratio and the number of vehicles, which is represented for the traffic load. We can observe that the message loss ratio of the three schemes increases as the traffic load increases. The group signature based scheme has the highest PLratio, the PKI based scheme grades the second place, whereas RPD has the lowest PLratio. This is because, the message verification rate is absolutely based on the $h_{aggr}$ comparison computation cost is neglected when compared to the PKI based signature scheme [3].

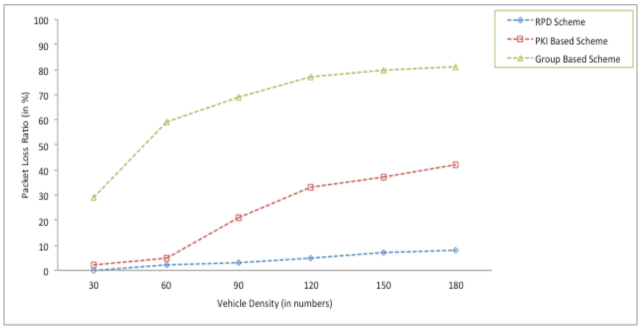



*Figure 3: Message Loss Ratio vs Vehicle Density*

### c) Average end to end message delay

The average end to end message delay which we denote $\mathrm{MSG_{delay}}$ as can be defined as the difference between the time $V_i$ sends the $m^{th}$ message and the time $V_j$ receives it. Considering N as the total number of vehicles in the simulation, M as the number of messages sent by the vehicle, and J as the number of adjacent vehicles within the communication range of vehicle $V_i$. If $T_{send}^{i\_j\_m}$ represents the time instant $V_i$ in the application layer sends the $m^{th}$ message to $V_j$ and $T_{recv}^{i\_j\_m}$ represents the instant $V_j$ in the application layer receives the $m^{th}$ message then, according to [8], the average message delay is expressed as follows:

$$\mathrm{MSG_{delay}} = \frac{1}{(N.M_n.J_n)}\sum_{j=0}^{J_n}\sum_{m=1}^{M_n} x\,(T_{sign}^{i\_m} + T_{transmission}^{i\_j\_m} + T_{verify}^{i\_j\_m})\ x\,(L_{i\_j\_m} + 1)$$

Where $T_{sign}^{i\_m}$, $T_{transmission}^{i\_j\_m}$ and $T_{verify}^{i\_j\_m}$ denotes, the time taken by the $i^{th}$ vehicle to sign the $m^{th}$ message, the time taken for the $m^{th}$ message to get transmitted from $i^{th}$ vehicle to $j^{th}$ vehicle and the time taken by $j^{th}$ vehicle to verify $m^{th}$ message respectively. $M_n$ is the number of messages sent by $V_i$ and $J_n$ is the number of vehicles within the one hop communication range of $V_i$. Since RPD does not require the message to be signed and the verification can be neglected as the $h_{aggr}$ comparison computation is very fast, the message end to end delay is exclusively depends on the transmission delay which does not vary a lot with the increase of traffic load such like for a city scenario of 20 to 150 vehicles the message end to end delay is around 22ms[8] which is smaller than the maximum allowable message end to end transmission latency of 100ms[7]. $L_{i\_j\_m}$ (denotes the queue length in $V_j$ when $m$ from $V_i$ is received) is neglected in RPD as well, for the above said reason. Therefore, the message delay for RPD can be reformulated as follows:

$$\mathrm{MSG_{delay}} = \frac{1}{(N.M_n.J_n)}\sum_{j=0}^{J_n}\sum_{m=1}^{M_n} x\,(\ T_{transmission}^{i\_j\_m})$$

$$= \frac{1}{(N.M_n.J_n)}\sum_{j=0}^{J_n}\sum_{m=1}^{M_n} x\,(\ T_{recv}^{i\_j\_m} - T_{send}^{i\_j\_m})$$

Figure 4 shows the relationship between the $\mathrm{MSG_{delay}}$ and the traffic load. We can see that group signature scheme has the highest $\mathrm{MSG_{L\_ratio}}$ due to the high verification delay whereas RPD yields the minimum $\mathrm{MSG_{delay}}$. This demonstrates the effectiveness of the proposed protocol.

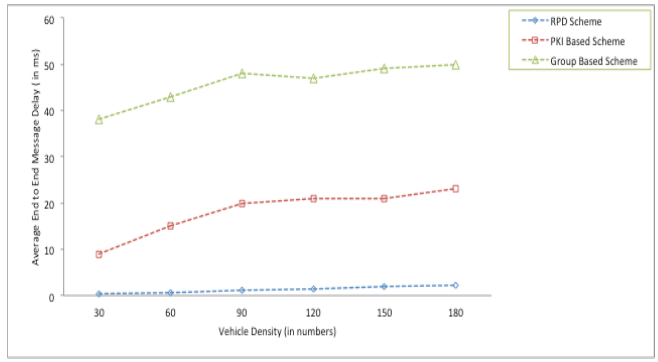



*Figure 4: Average End-End Message Delay vs Vehicle Density*

## 5. CONCLUSION

In this paper, a novel reusable pseudo-id distribution (RPD) scheme has been proposed. With RPD, RSUs are responsible to generate the anonymous ids in bulk and issue them one at a time to the requesting vehicles. The token, which binds the long-term public key of the vehicle with the given pseudo-id, facilitates traceability. Also this makes the vehicle accountable for messages from the pseudo-id and insists the token return to get a new pseudo-id. The RPD protocol has many advantages because of the cost cut down in verifying messages. Extensive simulation has been conducted to demonstrate the quite low transmission delay, message loss ratio and the message end-to-end delay. For future research, we will contribute to reduce the signature verification cost for vehicle-to-vehicle communication when the fixed infrastructures such as RSUs are absent in the network.

## 5. REFERENCES

[1] U.S. Dept. Transp., Nat. Highway Traffic Safety Admin., **Vehicle Safety Communications Project**, 2006. Final Rep.

[2] Raya, M. Hubaux, J.P.: *In:* **Proc. of Third ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2005)**, Alexandria, November 2005.

[3] M. Raya and J,P, Hubaux, **Securing Vehicular Ad Hoc Networks, Journal of Computer Security**, *Special Issue on Security of Ad Hoc and Sensor Networks,* Vol. 15, pp. 36-68 , January 2007.

[4] X. Lin, X. Sun, X.Wang, C. Zhang, P.–H. Ho and X. Shen, TSVC: Timed Efficient and Secure Vehicular Communications with Privacy Preserving, **IEEE Transactions on Wireless Communications**, vol. 7, no, 12, pp. 4987-4998, December 2008.

[5] G.Calendrillo, P. Papadiitratos, J.-P. Hubaux, and A. Lioy. Efficient and Robust Pseudonymous Authentication in VANET. **In proceedings of the fourth ACM international workshop on Vehicular Ad hoc Networks (VENET)**, pp. 19-28, New York, NY, USA, 2007, ACM press.

[6] G.Calendrillo, P. Papadiitratos, J.-P. Hubaux, and A. Lioy. On the Performance of Secure Vehicular Communication Systems, **IEEE Transactions on Dependable and Secure Computing**, 2010.

[7] National highway traffic safety administration, U.S. Department of Transportation, **Vehicle Safety Communications Project – Final Rep.**, April 2006.

[8] X. Lin, X. Sun, P.–H. Ho and X. Shen, GSIS: A secure and privacy preserving protocol for vehicular communications, **In proc. IEEE INFOCOM**, Phoenix, AZ, 2008, pp: 1229-1237.

[9] C. Zhang, R. Lu, X. Lin, P.–H. Ho and X. Shen, An efficient identity based batch verification scheme for vehicular sensor

networks, **In proc. IEEE INFOCOM**, Phoenix, AZ, 2008, pp: 246-250.

[10] K. Ren, W. Lou, R.H. Deng, and K. Kim, A novel privacy preserving authentication and access control scheme in pervasive computing environments, **IEEE Trans. Veh. Techonol.**, vol. 55, no. 4, pp. 1373-1384, July 2006.

[11] Y. Xi, K. Sha, W. Shi, L. Schewiebert, and T. Zhang, Enforcing privacy using symmetric random key-set in vehicular networks, **In proc. ISADS**, 2007, pp. 344-351.

[12] R. Lu, X. Lin, H. Zhu, P.–H. Ho and X. Shen, ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications, **In proc. IEEE INFOCOM**, Phoenix, AZ, 2008, pp: 1229-1237.

[13] C. Zhang, X. Lin, R. Lu, and P.–H. Ho, RAISE: An efficient RSU aided message authentication scheme in vehicular communication networks. **IEEE International Conference on Communications (ICC '08)**, Beijing, China, May 2008.

[14] D. Chaum and E. Ven Heyst, Group Signatures, **In Advances in Cryptology – EUROCRYPT 1991**, LNCS 547, pp. 257-265, Springer-Verlag, 1991.

[15] Hu. Xiong, Konstantin Beznosov, Zhiguang Qin, Matei Ripeanu, Efficient and Spontaneous privacy preserving protocol for secure vehicular communications, **In proc. IEEE ICC**, Cape Town, May 2010, pp. 1-6.

[16] D. Boneh and X. Boyen, and H. Shacham, Short Group Signatures, **In Advances in Cryptology – EUROCRYPT 2004**, LNCS 3152, pp. 41-55, Springer-Verlag, 2004.

[17] K. Sha, Y. Xi, W. Shi, L. Schewiebert, and T. Zhang, Adaptive privacy-preserving authentication in vehicular networks, **In proc. of the International Workshop on Vehicular Communication and Application**, pp. 1-8, October 2006.

[18] K. Sampigethaya, M. Li. L. Huang, and R. Poovendran, AMOEBA: robust location privacy scheme for VANET, **IEEE Journal on Selected Areas in Communications**, vol. 25, no. 8, pp. 1569-1589, 2007.

[19] L. Sweeney, K-ANONYMITY: a model for protecting privacy, **International Journal on uncertainty, Fuzziness and Knowledge Based Systems**, vol. 10, no. 5, pp. 557-570, 2002

[20] IEEE Trial-use Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages. **IEEE Standard 1609.2-2006**, July 2006.

[21] Hyoung-Kee Choi, In-Hwan Kim and Jae-Chern Yoo, Secure and Efficient Protocol for Vehicular Ad Hoc Network with Privacy Preservation, **EURASIP Journal on Wireless Communications and Networking**, vol. 2011.

[22] T. Elgamal, A public-key cryptosystem and a signature scheme based on discrete logarithms. **IEEE Transaction on InformationTheory**, Vol. 31, Bo. 4, pp. 469-472, 1985.

[23] V. S. Miller, Use of elliptic curves in cryptography, **in proc. of Advance in Cryptology**, *pp. 417-426, Aug. 1985.*

[24] D. Eastlake and P.Jones, US Secure hash algorithm 1 (SHA 1), **IETFRFC 3174**, 2001.

[25] Chenxi Zhang, Xiagong Lin, RongXing Lu, Pin-Han Ho, Xuemin Shen, An efficient message authentication scheme for vehicular communications, *in* **IEEE Transaction on Vehicular Technology**, vol, 57, No. 6, Nov 2008.

[26] X. Sun, Anonymous secure and efficient vehicular communications, **M.S. thesis, University of Waterloo**, ON, Canada, 2007.