Secure Public Key Regime (SPKR) in vehicular networks

Tan Heng Chuan, Jun Zhang, Ma Maode, Peter Han Joo Chong, Houda Labiod

Secure Public Key Regime (SPKR) in Vehicular Networks

Tan Heng Chuan[1], Jun Zhang [2], Ma Maode[1], Peter Han Joo Chong[1], Houda Labiod[2],
[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
{htan005, emdma, ehjchong} @ntu.edu.sg
[2]INFRES, Telecom ParisTech, 46, rue Barrault, 75634, Paris cedex 13, France
{Labiod,jun.zhang}@telecom-paristech.fr

***Abstract*— Public Key Regime (PKR) was proposed as an alternative to certificate based PKI in securing Vehicular Networks (VNs). It eliminates the need for vehicles to append their certificate for verification because the Road Side Units (RSUs) serve as Delegated Trusted Authorities (DTAs) to issue up-to-date public keys to vehicles for communications. If a vehicle's private/public key needs to be revoked, the root TA performs real time updates and disseminates the changes to these RSUs in the network. Therefore, PKR does not need to maintain a huge Certificate Revocation List (CRL), avoids complex certificate verification process and minimizes the high latency. However, the PKR scheme is vulnerable to Denial of Service (DoS) and collusion attacks. In this paper, we study these attacks and propose a pre-authentication mechanism to secure the PKR scheme. Our new scheme is called the Secure Public Key Regime (SPKR). It is based on the Schnorr signature scheme that requires vehicles to expend some amount of CPU resources before RSUs issue the requested public keys to them. This helps to alleviate the risk of DoS attacks. Furthermore, our scheme is secure against collusion attacks. Through numerical analysis, we show that SPKR has a lower authentication delay compared with the Elliptic Curve Digital Signature (ECDSA) scheme and other ECDSA based counterparts.**

Keywords— Schnorr signature; certificate-less PKI; Denial of Service Attacks; Collusion Attacks

I. INTRODUCTION

Public key cryptography is used to secure communications in Vehicular Networks (VNs). It guarantees message authentication, integrity and non-repudiation with digital signatures. The private key is used to generate a digital signature whereas the public key is used for verifying the digital signature. If the hash value of the message matches the hash value of the received signature after decryption, it proves that the message is not altered (integrity property) and that the message was created by a known sender such that the sender cannot deny having send the message (authentication and non-repudiation property). However, since the public keys are published publicly, it has the problem ascertaining if a public key truly belongs to the purported owner. An attacker can create a private/public key pair and then announces it to the entire network that the public key he publishes belongs to another user, for example, Bob. Therefore, when other nodes sends confidential messages encrypted using Bob's public key, the attacker decrypts and reads the message instead. To solve this issue related to the ownership of public key, Public Key Infrastructure (PKI) is used. In North America, the IEEE Wireless Access in Vehicular Environments (WAVE) standard [1] describes the IEEE standard 1609.2 [2] that specifies that a PKI to be employed to secure WAVE messages. In Europe, European Telecommunications Standards Institutes group for Intelligent Transport Systems (ETSI ITS G5) establishes the ETSI TS 102 940 standard [3] that mandates the use of a PKI for secure communication.

In a PKI, a Certificate Authority (CA) issues digital certificate to attest the credentials of the entities taking part in a communication protocol. Since the CA is trusted by all entities in the network, any node that receives a digital certificate that is signed by the CA can be ensured of the authenticity of the public key. The digital certificate contains binding between the public key and one or more attributes related to the user's identity. Unfortunately, employing a PKI in VNs has several limitations as highlighted in [4-6]. First of all, each vehicle is required to store large amount of certificates (Long Term Certificates/Pseudonyms Certificate) for PKI support which leads to storage overhead issue. At the same time, each vehicle needs to append its certificate signed by the CA to the message during transmission which increases the message overhead. This results in wastage of resources and valuable communication bandwidth. Another issue is the need to maintain a huge Certificate Revocation List (CRL) which does not scale well with increasing network size. Furthermore, the freshness of public keys cannot be guaranteed because CRL is disseminated only periodically by the CA. A vehicle may unknowingly accept a public key that has already been revoked. After receiving the certificate, vehicle needs to (1) check the expiry date of the certificate, (2) check the received certificate against a recent issued Certification Revocation List (CRL) to verify its status and (3) verify the CA's digital signature on the received signed certificate. This introduces much latency which is undesirable in most critical VNs applications that have strict delay requirements. For example, forward collision warning requires a frequency of 10 messages per second with a maximum latency of 100ms [7].

To address the inefficiency of certificate based PKI, a Public Key Regime (PKR) scheme [5] was proposed. It is designed based on the public file concept where each Road Side Unit (RSU) in the network is given a public key directory by the Trusted Authority (TA). This public key directory

contains the vehicles' IDs and their corresponding public keys already certified by a Trusted Authority (TA). To start a communication in the network, a vehicle has to broadcast its identity to the other communicating vehicle and the receiving vehicle sends a public key query to a nearby RSU to request for the sender's public key. Evaluation results show that the transmission overhead of the PKR scheme is lower than the certificate based PKI because the certificate is eliminated from the transmission. It also has a lower latency compared with the certificate based PKI because time-consuming certificate verifications are eliminated and there is no need to download huge CRL making it very scalable in a large scale environment.

However, PKR scheme is designed without security considerations. Since the RSU needs to service many query requests from the vehicles in a region, this may result in a bottleneck at the RSU leading to a Denial of Service (DoS) attack, compromising on the availability of service. Moreover, in the scheme, all the RSUs maintaining the public key directory are assumed to be trusted which exposes a security risk. The RSU can misbehave to launch a colluding attack with a malicious vehicle to issue a signed public key belonging to a colluding vehicle. Consequently, messages encrypted using a session key derived from the public key of the colluding vehicle will be compromised.

In this paper, we examine these two threats and propose several extensions to the PKR scheme. Our proposed solution is called the Secure Public Key Regime (SPKR) scheme. To mitigate the DoS attacks, the key idea is to let the requesting vehicle expends equal amount of computing power when it requests for public keys from the RSU. In this way, it discourages internal attackers from launching DoS attacks. To achieve this, we introduce a mutual pre-authentication scheme between a vehicle and an RSU whereby two communicating parties have to exchange signatures for verification before the requested public key is issued. This involves both parties to commit some CPU resources. External DoS attackers are also prevented since the unauthorized vehicles will not pass the authentication process. To solve collusion attacks, several changes are made to the stored public key directory held by each RSU.

The rest of the paper is organized as follows. Section II reviews the related works. Section III provides an overview of the PKR scheme and describes the two security threats. Section IV describes the system model. Section V discusses the SPKR scheme in details. Section VI analyses the security and performance of SPKR scheme. Section VII concludes the paper.

II. Related Works

The issues of certificate based PKI are widely known and discussed. Besides the PKR scheme [5], several other works have been proposed to make certificate based PKI more efficient. In [8], Wasef and Shen propose a Message Authentication Acceleration Protocol (MAAC) to accelerate the traditional CRL checking process. It uses a keyed Hash Message Authentication Code (HMAC) together with a secret key shared only among unrevoked vehicles to check that a vehicle is not previously revoked before proceeding to check the certificate against the CRL In this manner, MAAC avoids the long delay in checking the vehicle's certificate against a huge CRL list. Other approaches [9] [10] uses the ID-based cryptography to simplify the certificate management process. In an ID based system, the vehicle's identity is used as a public key for signing and verifying messages which greatly reduces the communication and computation overhead.

To mitigate external DoS attacks, Wasef et al. [4] propose to append a Hash based Message Authentication Code (HMAC) to all the outgoing signed messages when the number of invalid signatures to the number of received messages exceeds a certain threshold. The HMAC is generated using a common group key shared among all unrevoked vehicles. When other vehicles receive the message attached with the HMAC, each of them will verify the HMAC before verifying the digital signature. As long as the group key is not disclosed and unforgeable, this scheme is secure against external attackers. However, the disadvantage of employing a threshold value is that it will take several invalid signatures to detect an outsider attackers and this scheme only focuses on external DoS attackers. Other DoS mitigation schemes include [11] which is a modification of [4]. A pre-authentication process is introduced before the signature verifying process where a chain value of the one way hash chain is appended instead of the HMAC over the entire message. It proves that this method is computationally more efficient. But similar to [4], this scheme can only defend against external DoS attackers.

III. Overview of PKR Scheme

In this scheme, each RSU registers with the TA to receive a read-only copy of the public key directory and the TA's signing key. This public key directory contains the registered vehicles IDs and their corresponding public keys certified by the root TA. When a vehicle wishes to communicate to another vehicle, it sends a public key request message to any nearby RSU to request for it. The exact operation is detailed as follows.

Step 1: Periodic Announcements

Vehicle 1 and vehicle 2 periodically broadcast a beacon message that contains its identity to the other neighboring vehicles within its communication range.

Step 2: Public key query

If vehicle 1 (vehicle 2) wants to talk to vehicle 2 (vehicle 1), vehicle 1 (vehicle 2) sends an unencrypted signed public key request to RSU 1, requesting for a copy of vehicle 2's (vehicle 1's) public key.

Step 3: Verification of public key query

RSU 1 then searches for the sender's (vehicle 1/vehicle 2) public key in the public key directory and verifies the signature on the sender's message to confirm its authenticity and message integrity.

Step 4: Issuing of requested public key reply

If the verification is successful, RSU 1 searches for the public key of the requested vehicle ID in the public key directory and generates a reply message. This reply message is encrypted using the sender's (vehicle 1's/vehicle 2's) public key and its message content is signed using the TA's signing key (which is given to the RSU in the registration phase). When the sender (vehicle 1/vehicle 2) receives the message from RSU 1, it will decrypt the message using its own private key to retrieve the requested public key and then verifies the TA's signature on the public key using the TA's public key.

Step 5: Session key agreement

Vehicle 1 and vehicle 2 then proceed to use each other public key to negotiate a session key and subsequently, use the common session key for message encryption.

A. *Security Threats*

As mentioned earlier, PKR scheme is vulnerable to DoS attacks and collusion attacks. In this sub-section, we describe the two attacks.

1) *Denial of Service (DoS) Attacks*

In the PKR scheme, the RSU has to verify many signed public key request messages that originate from the vehicles in the RSU's communication range. This generates a huge amount of computation and communication overheads on the RSU side which creates a bottleneck. Exploiting this weakness, an external or internal attacker can launch a DoS attack by generating many invalid signed public key queries to deplete the resources of the RSU. As a result, legitimate vehicles are denied access to the service and therefore unable to communicate in the network. Situation can be worse if the legitimate vehicles are law enforcement vehicles that are deprived of communications. This could lead to dire consequences in life critical situations.

2) *Collusion Attacks*

A misbehaved RSU may collude with another vehicle to issue the public key of the colluding vehicle such that subsequent confidential messages from the requesting vehicle are read by the colluding vehicle. This is possible because RSU has the TA's signing key which allows it to issue any valid signature using any public key stored in the public key directory. The effects of this attack are critical in the network as message confidentiality and privacy are compromised.

IV. SYTEM MODEL

We consider an infrastructure-based vehicular network consisting of 3 entities which we briefly describe below.

1) *Transport Authority (TA)*

TA is in charge of the management of the vehicles and Road side Units (RSUs) in a geographical area. It is responsible for the registration of vehicles and RSUs, creation of the public file directory as well as updating and dissemination of the directory. It is assumed trusted by all entities (vehicles and RSUs) in the deployment area and is equipped with advanced security mechanisms to safeguard any information leakage. It therefore, has the highest security level and cannot be compromised.

2) *Road Side Units (RSUs)*

The RSUs are devices that are statically and strategically deployed in the network to bridge the communication between vehicles or among RSUs in the network. They are installed at traffic lights, lamp posts or road signs etc. They are connected to the RTA via a secure network and are equipped with multiple interfaces for interoperability with different access technologies. RSUs have no resource constraints in terms of storage and computing power. Each RSU is given a copy of the public key directory where it assists in the request and issuance of public key. We assume majority of the RSUs are trusted but not all of them.

3) *Vehicles*

Vehicles can be static or mobile, travelling in the network with varying speeds. Vehicles are installed with Onboard Unit (OBU) to support communications with other vehicles and RSUs. They are also equipped with a Hardware Security Module (HSM) responsible for storing and physically protecting the cryptographic information. Any attempt to retrieve any information from the HSM will cause it to self-destruct. We assume that majority of the vehicles are trusted but not all.

V. SPKR SCHEME

Our scheme makes use of the Schnorr signature to mutually authenticate each other before the RSU issues the requested public key. The benefits of using Schnorr signature are twofold. Firstly, a DoS attack is mitigated. Each communicating party has to prove to each other that it has committed some CPU resources to verify digital signature by providing a knowledge proof of a discrete logarithm. By doing so, vehicles have no incentive to misbehave as it is also going to cost them huge computational costs. Secondly, both parties are properly authenticated and no external unauthorized party can take part in the communication process. To facilitate the Schnorr based authentication process, we propose that each vehicle also stores a public key directory similar to the one kept by each RSU. In addition, the RSU in the SPKR scheme is not given the TA's signing key to prevent collusion attacks.

The SPKR scheme consists of four steps namely initialization, vehicles/RSU registration, dissemination of public key directory and authentication. More details of the scheme are elaborated below.

A. *Initialization*

TA chooses a large prime p such that the discrete logarithm problem in Z_p^* is intractable and chooses another prime q that divides $p-1$. Next, TA selects a generator g of an order q subgroup of Z_p^* such that $g^q = 1\,(mod\ p)$ and let $h: \{0,1\}^* \rightarrow Z_p$ be a secure hash function. All system

parameters (p, q, g, h) are public to all the vehicular network users.

B. Vehicles/RSUs Registration

Each vehicle i chooses a private key $a_i \in Z_p^*$ and constructs its own public key $k_i^+ = g^{-a_i} (mod\ p)$. After generating the public key, each vehicle registers its public key with the TA. TA first generates a unique ID for the vehicle, then binds the public key with the corresponding vehicle's ID and stores this entry inside a public file directory. To prevent collusion attacks, another column of information is created in the public key directory to store the signature created from the vehicle's ID and its public key which is signed by the TA's signing key. Since the RSU is not in possession of the TA's signing key, RSU is not able to issue a forged public key reply using another vehicle's public key different from the one being requested. The role of the RSU is thus reduced to only issuing public keys. The new database format is shown in Fig. 1.

Each RSU i also chooses a private key $a_i \in Z_p^*$ and constructs its own public key as $k_i^+ = g^{-a_i} (mod\ p)$. Similarly, TA generates a unique ID for each registered RSU, binds the RSU's public key with its corresponding ID and stores it in another public key directory. To differentiate between the two public key directories, the directory that stores only the vehicle's ID, its public key and TA's signature over the vehicle's ID and public key binding is called the Vehicle Public Key Directory (VPKD) whereas directory that stores only RSU's ID and its associated public key is called the RSU Public Key Directory (RSU-PKD). The format of the RSU-PKD is shown in Fig. 2.

C. Dissemination of VPKD and RSU-PKD

After successful registration, TA disseminates read-only copy of the VPKD to each RSU in the network. On the other hand, read-only copy of the RSU-PKD is given to each successfully registered vehicle. Since the vehicle is not responsible for issuing public keys, it does not need to contain the TA's signature column. Therefore, the size of the RSU-PKD is smaller than VPKD. The purpose of the RSU-PKD in each vehicle is only to serve as a lookup table to reference the RSU's public key for verification of signature during the authentication stage. Any changes to the VPKD will be updated and disseminated in real-time to the RSUs via a secure network according to the implementation in [5]. For RSU-PKD, vehicles can receive the updates from Original Equipment Manufacturers (OEM) whenever there are changes in the RSU-PKD.

Vehicle ID	Public Key	ID and Public key binding signed by TA's signing key
1	k_1^+	$Sig_{k_{RTA}^-}\{1, k_1^+\}$
2	k_2^+	$Sig_{k_{RTA}^-}\{2, k_2^+\}$
9	k_9^+	$Sig_{k_{RTA}^-}\{9, k_9^+\}$
⋮	⋮	⋮

Fig. 1. VPKD

RSU ID	Public Key
5	k_5^+
4	k_4^+
100	k_{100}^+
⋮	⋮

Fig. 2. RSU-PKD

D. Mutual Pre-authentication

Before an RSU issues a public key to the vehicle, both vehicle and RSU take turns to generate a Schnorr signature that contains a pre-generated commitment value. The authentication process is shown in Fig. 3 and consists of the following steps.

Step 1 and Step 2: Vehicle 1 and RSU 5, each select two random values, $r_i, k_i \in Z_p^*$ where i stands for the ID of the nodes and computes R_i and K_i according to the formula in step 2 of Fig. 3. The R_i (R_1 and R_5) values serve as nonce values to prevent replay attack. The K_i (K_1 and K_5) are commitment values used to prove the knowledge of their private keys.

Step 3: Suppose vehicle 1wants to communicate with vehicle 3, it constructs a public key query message m containing its own ID and the target vehicle's ID i.e. vehicle 3 and sends it to RSU 5. This message to RSU 5 also contains the random nonce value R_1.

Step 4: Upon receiving (R_1, m) from vehicle 1, RSU 5 generates a challenge, e_5 and a response, S_5 according to the formula given in step 4 of Fig. 3 and sends the tuple (e_5, S_5, R_5) to vehicle 1. The challenge contains the hash of the message, the random nonce values (R_1, R_5) and the commitment value, K_5 of RSU 5 used to proof that RSU 5 has the knowledge of its private key.

Step 5: When vehicle 1 receives (e_5, S_5, R_5) from RSU 5, it first retrieves the RSU 5's public key from the RSU-PKD. Then, vehicle 1 re-computes the RSU 5's commitment value as $\bar{K}_5 = g^{S_5} k_5^{+e_5} (mod\ p)$ based on the received S_5, e_5 and RSU 5's public key, k_5^+. Next, vehicle 1 calculates its version of the challenge denoted as $\bar{e_5}$ by taking the hash of the message, m concatenated with both random nonce values (R_1, R_5) and the calculated value of $\bar{K}_5$ computed earlier. If the calculated hash value matches the received e_5, it implies that RSU 5 is authenticated successfully. If there is a mismatch, the session terminates at this stage. The proof of correctness is shown in (1) and (2). Once the challenge $\bar{e_5}$ is verified true, vehicle 1 prepares its own signature consisting of a challenge e_1 and a response S_1. At the same time, vehicle 1 needs to compute v_1 which is a hash of the two random nonce values and the $\bar{K}_5$ calculated earlier. Message to RSU 5 contains (v_1, e_1, S_1).

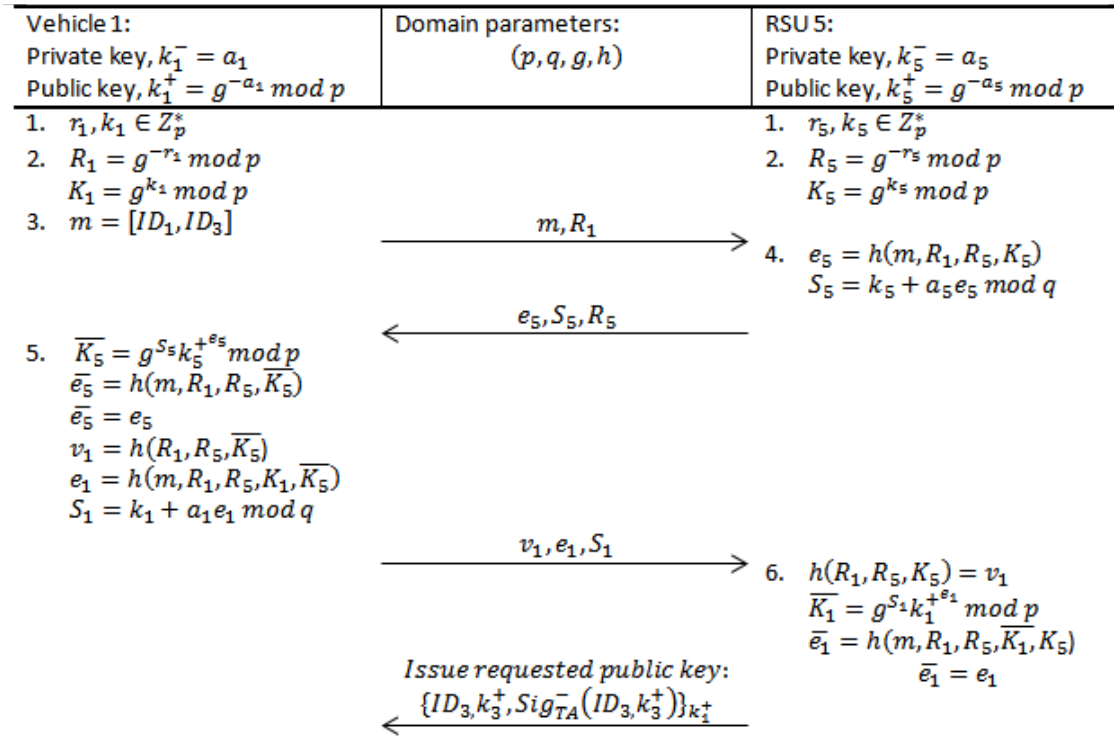

Fig. 3. Pre-authentication process

Proof of Correctness:

$$\overline{K_1} = g^{S_1} k_1^{+e_1} (mod\ p) = g^{k_1 + a_1 e_1} \cdot g^{-a_1 e_1}\ (mod\ p) = g^{k_1} \bmod p \quad (1)$$

$$\bar{e}_1 = h(m, R_1, R_5, \overline{K_1}, K_5) \quad (2)$$

If $\bar{e}_1 = e_1$ and the signature pair (e_1, s_1) from vehicle 1 passes verification.

Step 6: Before RSU 5 proceeds to verify the signature pair (e_1, S_1) from vehicle 1. It first checks that the hash value of v_1 is correct by computing $h(R_1, R_5, K_5)$ using its own commitment value, K_5 generated in step 2. The signature pair (e_1, S_1) needs to be verified only if the hash value matches. The verification procedure of (e_1, S_1) is the same in step 5. RSU 5 computes the commitment value of vehicle 1 using $\overline{K_1} = g^{S_1} k_1^{+e_1} (mod\ p)$ and then verifies if $\bar{e}_1 = h(m, R_1, R_5, \overline{K_1}, K_5)$ matches the received e_1.

Step 7: RSU 5 issues the requested public key only if $\bar{e}_1 = e_1$, Otherwise, RSU 5 terminates at this step. The reply message to vehicle 1 contains the requested public key and TA's signature over the public key and the entire message is encrypted using vehicle 1's public key to ensure confidentiality. The TA's signature is to ensure the integrity and authenticity of the sent message.

VI. Security and Performance Analysis

In this section, we show that SPKR is robust against DoS and collusion attacks. After that, we analyze the storage overhead requirements.

A. Denial of Service (DoS) Attack

DoS attack is mitigated in the SPKR scheme because RSU 5 has to check if vehicle 1 has the correct hash value, $h(R_1, R_5, K_5)$ which contains the commitment value K_5 generated by RSU 5 in step 2. This implies that vehicle 1 has to spend expensive modular exponentiations to verify the signature pair given by RSU 5 in step 4. These expensive verifications are required to reveal the correct K_5 in order to obtain the correct hash value, v_1. Using this approach, it discourages any malicious but rational attackers from launching DoS attacks. On the other hand, RSU 5 verifies vehicle 1's signature only when $h(R_1, R_5, K_5)$ is evaluated to be true which is inexpensive to RSU 5.

The malicious vehicle may attempt to skip the computational intensive verification of the RSU's signature by trying to forge a valid v_1 which is a function containing K_5. This is not possible due to the property of the one-way hash function and the hardness of the discrete logarithm problem. The RSU may misbehave to cause requesting vehicles to waste their resources without issuing the requested public key. However, the impact of such misbehavior from the RSU side is less destructing since vehicles are mobile and they can request from other well-behaved RSUs in the region. Moreover, a reputation system can be implemented to measure the trust

level of the RSUs so that vehicles are better informed about the reliability of the RSUs before they request public keys from them. This, however, is beyond the scope of this paper and is left as a future work.

Next, we analyze the authentication delay of the SPKR scheme and compared with those of the Elliptic Curve Digital Signature (ECDSA) algorithm which is the de facto standard adopted in [2, 12] and the ECDSA-HMAC mechanism proposed by Wasef et al.[4]. The comparison is in terms of the following aspects: the authentication delay incurred by the RSU to process a public key query request from a vehicle.

For the SPKR, the authentication delay consists of the signature generation time, hashing time and verification time. When there is a DoS attack, the authentication delay consists of only the signature generation time and the hashing time because if there is a mismatch in the hash value, the RSU will skip the signature verification process. For the ECDSA, the authentication delay consists of the signature verification time regardless of whether there is DoS and no DoS attack. If ECDSA-HMAC[4] is used and when there is a DoS attack, the authentication delay consists of the signature verification time and the HMAC verification time.

To calculate the authentication delay, we implemented the SPKR and the ECDSA schemes in C using Openssl library to estimate the signature generation and verification times. Simulation was carried out for 100000 times on an Intel Core i7-2620M@2.70Ghz workstation with 2 GB RAM running on 32 bit Debian Linux operating system. Table 1 shows the running times of both schemes in milliseconds. Similarly, we ran the HMAC program for 100000 simulation runs and estimated the running time to be 0.008ms. Fig. 6 shows the authentication delay of all the schemes under no DoS and DoS attack.

We observe that when ECDSA is employed, the introduction of invalid messages (10% and 30% of the number of valid messages) increases the authentication delay greatly. On the other hand, using the Schnorr signature scheme as the mutual pre-authentication mechanism in SPKR, two observations can be made. First, the authentication delay is significantly lower than the ECDSA based schemes (ECDSA and ECDSA-HMAC) under no DoS attack. Second, according to the left inset figure in Fig. 6, the performance of the SPKR scheme when subject to 10% and 30% invalid messages has little effect on the authentication delay and is also significant lower than the two ECDSA based schemes under DoS attacks. SPKR scheme is more efficient because the signature generation time is much lower than the ECDSA counterparts due to that computation of modular exponentiations in step 1 and 2 can be done off-line. Furthermore, RSU only checks the vehicle's signature when the proof provided by the vehicle is correct. In this manner, RSU can differentiate well-behaved vehicles from DoS attackers. This explains why SPKR scheme has a lower authentication delay. A lower authentication delay indicates a higher availability to service the requests of other vehicles. Although ECDSA-HMAC scheme does not contribute much to the authentication delay under DoS attacks based on the right inset figure in figure 6, our scheme is still far more superior than the ECDSA-HMAC scheme. Based on these observations, we conclude that the SPKR scheme can effectively detect invalid messages and mitigate DoS attackers.

TABLE I. DELAY PERFORMANCE FOR SCHNORR AND ECDSA

Algorithm / Key length	Schnorr	ECDSA
	1024	160
Signing in ms	0.013	0.996
Verify in ms	1.506	1.887
Hashing in ms	0.009	-

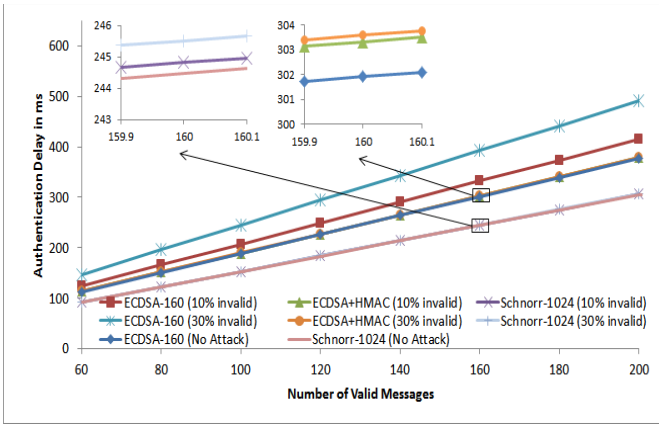

Fig. 1. Authentication delay under no DoS and DoS attacks

B. *Collusion Attack:*

In the original PKR scheme, when a vehicle requests for a target vehicle's public key, the RSU searches for it in the public key directory and then signs it using the TA's signing key. Since the RSU has possession of the TA's signing key, it is able to fake the signature to make sure that the signature always passes with the issued public key even if the public key is not the one requested by a vehicle. In the SPKR scheme, each entry $\{ID, public\ key\}$ in the public key directory is attached with a signature generated by the TA's signing key during registration phase and the RSUs are not given the TA's signing key. Therefore, there is no way for the RSU to collude with another vehicle to manipulate and issue a valid signed public key reply. Another advantage of storing the TA's signature for every entry in the VPKD is that RSU do not need to generate a signature when it issues the requested public key. It can simply retrieve the TA's signature from the stored entry and send it back to the requesting vehicle thereby reducing the workload on the RSU. However, the storage size of the repository will increase which we analyze next.

C. *Storage overhead:*

We estimate the storage size required by each RSU and vehicle to store the VPKD and RSU-PKD respectively. Assume that the public key has a key size of 1024 bits and 4 bytes are allocated to store the node ID and further, the size of the TA's signature on the ID and public key pair is 40 bytes. If there are 1 million vehicles in the network, the size of the

VPKD would be $1\,million\,x\,(128+4+40)bytes = 172\,Mbytes$ which is an increase of about 30.3% compared to the original PKR scheme. If there are 100,000 RSUs deployed in the network, the size of RSU-PKD will be equal to $100{,}000\,x\,(128+4) = 13.2Mbytes$ which is still reasonable given that the typical storage capacity of an OBU is at least 256M bytes. This additional storage requirement is a trade-off for enhanced security against DoS and collusion attacks. To reduce the storage requirement, SPKR scheme can be realized with smaller key size using Elliptic Curve Cryptography (ECC) which is left as a future work.

VII. Conclusions

In this paper, we analyze the PKR scheme and show that it is vulnerable to DoS and collusion attacks. We improved on the scheme by introducing a mutual pre-authentication scheme using Schnorr signature and modified the way information is stored in the public key directories. Although each vehicle has to store a public key directory, we show that the total storage size is manageable given the current storage capacity of the OBUs in the vehicles. The results of these extensions prove that SPKR scheme alleviates the risk of DoS attacks and that it has a lower authentication delay compared to the ECDSA and ECDSA-HMAC schemes under DoS attack. It is also secure against collusion attacks. In terms of future work, we plan to apply the SPKR scheme using ECC. Furthermore, we are interest to validate the security properties of the SPKR scheme using AVISPA or Proverif.

References

[1] R. Uzcategui and G. Acosta-Marum, "WAVE: a tutorial," *Communications Magazine, IEEE,* vol. 47, pp. 126-133, 2009.

[2] "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," *IEEE Std 1609.2-2006*, pp. 0_1-105, 2006.

[3] E. T. S. Institute, "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management," *Tech Rep. ETSI TS 102 940 V1.1.1,* June 2012 2012.

[4] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]," *Wireless Communications, IEEE,* vol. 17, pp. 22-28, 2010.

[5] P.-Y. Shen, V. Liu, M. Tang, and C. William, "An efficient public key management system: an application in vehicular ad hoc networks," in *Pacific Asia Conference on Information Systems (PACIS)*, 2011, p. 175.

[6] A. Slagell, R. Bonilla, and W. Yurcik, "A survey of PKI components and scalability issues," in *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, 2006, pp. 10 pp.-484.

[7] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE,* vol. 46, pp. 164-171, 2008.

[8] A. Wasef and S. Xuemin, "MAAC: Message Authentication Acceleration Protocol for Vehicular Ad Hoc Networks," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 2009, pp. 1-6.

[9] S. Jinyuan, Z. Chi, Z. Yanchao, and F. Yuguang, "An Identity-Based Security System for User Privacy in Vehicular Ad Hoc Networks," *Parallel and Distributed Systems, IEEE Transactions on,* vol. 21, pp. 1227-1239, 2010.

[10] Z. Chenxi, L. Rongxing, L. Xiaodong, H. Pin-Han, and S. Xuemin, "An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008.

[11] H. Li and Z. Wen Tao, "Mitigating DoS attacks against signature-based authentication in VANETs," in *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, 2012, pp. 261-265.

[12] E. T. S. Institute, "Intelligent Transport Systems (ITS); Security; Security header and certificate formats," *ETSI TS 103 097 v1.1.1,* 2013.