Two-factor Verification for Mitigating Sybil Attack Launched by the Compromised Certificate Authority (LTCA or PCA)

Keyao Huang
KTH Royal Institute of Technology
Stockholm, Sweden
Keyao@kth.se

***Abstract*—Vehicle Public Key Infrastructure (VPKI) provides certificates for vehicles participating in the Vehicular Communication (VC) system, ensuring the authenticity and integrity of data transmission in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). SECMACE+ provides a viable solution for VPKI. However, it does not consider the Sybil attack launched by a compromised CA. The pseudonym resolution process by the Resolution Authority (RA) in SECMACE+ is unable to detect this malicious behavior. The author intends to prevent this attack based on the smallest possible changes to the original architecture, by introducing group signatures as a two-factor verification that allows RA to detect Sybil attacks initiated by a compromised CA.**

I. Introduction

Vehicular Communication (VC) systems are designed to facilitate the exchange of information between traffic entities, avoid potential road hazards, and improve safety when vehicles are in motion. Common models of communication technologies include vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). Vehicle Public Key Infrastructure (VPKI) provides certificates to entities in VC systems, primarily through public-private key pairs, to ensure the confidentiality, integrity, and authenticity of communications, and to protect against unauthorized data transmission and data tampering.

There are many designs and schemes for VPKI, and this report focuses on the SECMACE architecture [1] and intends to address the shortcomings therein. SECMACE considers multi-domains and each domain consists of a Long-term Certification Authority (LTCA), several Pseudonym Certification Authority (PCA), and a Resolution Authority (RA). The workflow is as follows: (1) Vehicle registers with the LTCA to obtain a long-term certificate (LTC). (2) The vehicle receives a ticket from the LTCA. (3) The vehicle obtains pseudonyms from the PCA by using the ticket. Besides, RA can interact with CAs for pseudonym resolution when it receives a report of suspicious pseudonyms.

SECMACE+ allows for accountability for Sybil attacks caused by illegal vehicles and relies heavily on the honest LTCA and PCA to check for duplicate requests, but when one of these CAs is compromised, RA is unable to detect a Sybil attack initiated by it. The proposed scheme is intended to make as few changes as possible to the SECMACE architecture and communication protocol, which can effectively mitigate this issue without introducing excessive overheads. The proposed solution is to use group signature scheme to provide two-factor verification.

II. Problem Statement and Adversary Model

A. Problem Statement

In SECMACE [1] and SECMACE+ [2], Sybil attacks initiated by illegal vehicles have been mitigated, specifically by the LTCA and the PCA checking whether the same request has already been processed upon receiving a ticket/pseudonym request. However, this is based on the premise that LTCA and PCA are honest. If the LTCA or the PCA is compromised (*one of them*), Sybil attacks can be launched and such malicious behavior cannot be detected and accounted for based on the current resolving protocols initiated by RA in both papers. Although it is possible for both the LTCA and the PCA to be compromised and collaborate in launching a Sybil attack, in this case the VPKI would be defunct, the identities of all vehicles would be revealed and any vehicle could be admitted to the system, Sybil attacks would no longer be a major threat. Therefore, this report will focus on the Sybil attack initiated by a single compromised CA (LTCA or PCA).

B. Adversary Model

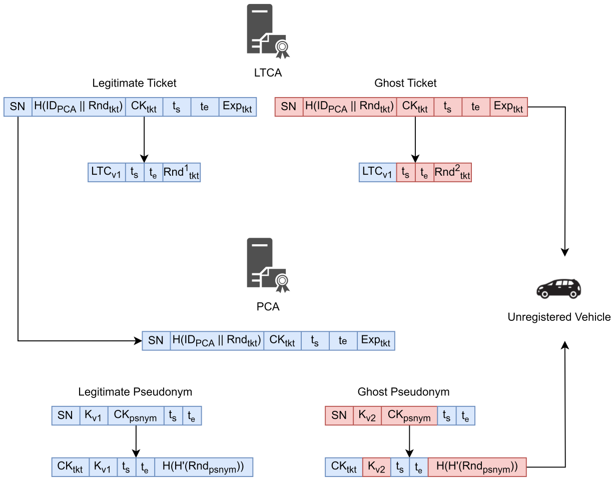
LTCA
Legitimate Ticket
SN | $H(ID_{PCA} \| Rnd_{tkt})$ | CK_{tkt} | t_s | te | Exp_{tkt}
LTC_{v1} | t_s | t_e | Rnd^1_{tkt}
Ghost Ticket
SN | $H(ID_{PCA} \| Rnd_{tkt})$ | CK_{tkt} | t_s | te | Exp_{tkt}
LTC_{v1} | t_s | t_e | Rnd^2_{tkt}
PCA
SN | $H(ID_{PCA} \| Rnd_{tkt})$ | CK_{tkt} | t_s | te | Exp_{tkt}
Unregistered Vehicle
Legitimate Pseudonym
SN | K_{v1} | CK_{psnym} | t_s | t_e
CK_{tkt} | K_{v1} | t_s | t_e | $H(H'(Rnd_{psnym}))$
Ghost Pseudonym
SN | K_{v2} | CK_{psnym} | t_s | t_e
CK_{tkt} | K_{v2} | t_s | t_e | $H(H'(Rnd_{psnym}))$

Fig. 1. Sybil Attack Pattern

Scenario 1: Compromised LTCA and honest PCA: Assume a legal vehicle (V_1) has been registered with LTCA and has received an LTC_{v_1} from the LTCA. Upon receiving a ticket request from V_1, the LTCA first generates a random number Rnd^1_{tkt} and a commitment key $CK^1_{n-tkt} = H(LTC_{v_1}||t_s||t_e||Rnd^1_{tkt})$. Then it generates the legitimate ticket $(SN, H(Id_{PCA}||Rnd_{tkt}), CK^1_{tkt}, t_s, t_e, Exp_{tkt})$. A compromised LTCA can copy the LTC_{v_1} and pick another random number Rnd^2_{tkt}, then generate an illegal commitment key $CK^2_{n-tkt} = H(LTC_{v_1}||t_s||t_e||Rnd^2_{tkt})$ and a ghost ticket $(SN, H(Id_{PCA}||Rnd_{tkt}), CK^2_{tkt}, t_s, t_e, Exp_{tkt})$. Though both the commitment keys are based on the same LTC, the random numbers are different, so the commitment keys obtained after hashing are also different. Since the PCA can only check if pseudonyms have been issued for the same commitment key before, the PCA will accept that CK^2_{n-tkt} is legal and thus issue pseudonyms for an unregistered vehicle.

Scenario 2: Compromised PCA and honest LTCA: Assume that a legitimate vehicle (V_1) acquires a legitimate ticket $tkt_{\sigma_{LTCA}} = (SN, H(Id_{PCA}||Rnd_{tkt}), CK_{tkt}, t_s, t_e, Exp_{tkt})$ and uses it to obtain a legitimate pseudonym $P^i_{v_1} = (SN^i, K^i_{v_1}, CK_{P^i_{v_1}}, t^i_s, t^i_e)$ from a compromised PCA, where $CK_{P^i_{v_1}} = H(CK_{tkt}||K^i_{v_1}||t^i_s||t^i_e||H'(H^i(Rnd_{psnym})))$. The compromised PCA can record this legitimate ticket, extract the commitment key CK_{tkt}, and uses it to generate ghost pseudonym for an unregistered vehicle (V_2). It first generates a fake pseudonym commitment key $CK_{P^i_{v_2}} = H(CK_{tkt}||K^i_{v_2}||t^i_s||t^i_e||H'(H^i(Rnd_{psnym})))$ with a legitimate ticket received before, then uses this key to generate a ghost pseudonym $P^i_{v_2} = (SN^i, K^i_{v_2}, CK_{P^i_{v_2}}, t^i_s, t^i_e)$. Therefore, a compromised PCA can generate a set of illegal pseudonyms based on a *legitimate ticket*. When RA initiates the pseudonym resolution process, it gets $(SN_{P^i}, n - tkt_{\sigma_{LTCA}}, H'(H^i(Rnd^i_v))$ from the PCA. First, it checks LTCA's signature on $n - tkt$, then checks $H(CK_{tkt}||K^i_{v_2}||t^i_s||t^i_e||H'(H^i(Rnd_{psnym}))) \overset{?}{=} CK_{P^i_{v_2}}$ (Protocol 5 in SECMACE+ [2]). Since the PCA uses a legitimate ticket and a legitimate commitment key (CK_{tkt}) to generate an illegal pseudonym, and RA will only check if a pseudonym is generated based on a legitimate ticket, RA cannot detect PCA's Sybil malicious behavior.

RA's resolving of ghost pseudonyms or ghost tickets will all end up locating a legitimate LTC and the holder of that LTC will be penalised by RA, whereas unregistered Sybil vehicles will not be punished, meaning that a Sybil attack initiated by a CA cannot be traced and has no cost.

III. Related Work

A. VPKI

AKI [3] and ARPKI [4] introduce log server to record certificates, which can increase transparency in the issuance of certificates. However, synchronizing certificates through multiple log servers causes great latency for the user to obtain the certificates, which is unacceptable in VC environment where the vehicle needs to obtain pseudonyms quickly.

ACMS [5] also utilizes log server to ensure the transparency of the certificate. It uses multiple LTCAs to prevent the issue of a single LTCA being compromised. Validator enrolls a vehicle after confirming signatures of multiple LTCAs. However, the authors do not discuss the issue of the validator being compromised, where a malicious validator can forward the registration request with multiple LTCA signatures that are less than the threshold to the Log server and allow the vehicle to complete the registration.

IOTA-VPKI [6] applies distributed ledger technology with SECMACE scheme [1] to increase the transparency of CA-signed certificates. It records $S = (Sign(LK_{CA}, H(crt)), Id_{CA})$ on the distributed ledger. However, there is no checking mechanism to verify the behavior of the LTCA in generating LTC and tickets, so LTCA's malicious behavior cannot be detected.

SCMS [7] focuses on V2X communication security and proposes a novel approach called butterfly key expansion that allows onboard equipment (OBE) to only send a single request to obtain multiple pseudonyms, avoiding the need for the vehicle to generate thousands of public keys and send them to the VPKI. However, this scheme is vulnerable to Sybil attack [8].

B. Group Signature

The paper [9] measures the time to sign and verify by both ECDSA-256 and group signature (GS) as well as proposes a hybrid scheme combining based-scheme and GS scheme to improve the speed of verifying pseudonyms by vehicles. Another paper [10] uses group signature to authenticate users in crowd sensing system.

IV. Proposed Solution

A. Preliminaries & Assumptions

In SECMACE [1] and SECMACE+ [2], RA, LTCA and PCA are assumed to be honest-but-curious. This report inherits the assumption of RA, which is honest-but-curious. The LTCA or the PCA can be compromised and initiate Sybil attack by issuing a ghost ticket or a ghost pseudonym for unregistered vehicle. Regarding the compromised LTCA, assume that the registration process of the LTCA is honest and the attacker only focuses on creating ghost tickets for unregistered vehicles. Assume that the entities in VPKI transfer information through end-to-end authenticated Transport Layer Security (TLS) channels [11], which can guarantee authenticity, and confidentiality of messages. Finally, assume that RA is honest and follows the protocol of resolution. The notations later used in the text are listed in Table I.

B. Architecture & Workflow

Initialization: The proposed solution maintains the SECMACE architecture with one RA, one LTCA, and one or several PCAs within a domain [1]. The architecture of the proposed scheme is presented in Fig. 2. During system initialization, RA generates group keys, including multiple group signing keys $(gsk_1, gsk_2, \ldots, gsk_i)$ and a group public key

TABLE I
NOTATIONS

CK	Commitment Key
(LK, Lk)	Long-term public key / private key
gsk, gpk	Group Signing Key, Group Public Key
PK, sk	RSA public/private key pair
P_v^i	Pseudonym i of the vehicle
$Sign(Lk_{LTCA}, msg)$	Signing a message with the private key of the LTCA
$Verify(Lk_{LTCA}, msg_{\sigma_{LTCA}})$	Verifying a message's signature with Lk_{LTCA}
H	Hash function
Rnd	A random number
$t_s/t_e, t_{now}$	Starting/ending, current time
ξ	Temporary Variable
$Reveal((P_v^i)_{\sigma_{gsk_i}})$	Revealing the private signing key of P_v^i
$Enc(PK, msg)$	Encrypting message with public key LK
$Dec(sk, msg)$	Decrypting message with private key Lk
$GenRnd()$	Generating a random number
V	Vehicle
SN, CN	Serial Number, Common Name

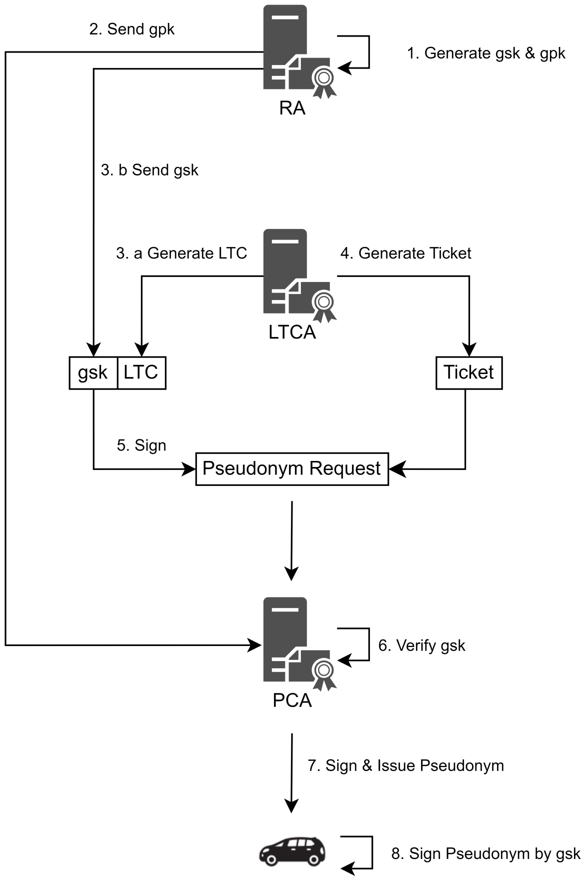

Fig. 2. Workflow of Proposed Solution

(gpk), which will be sent to the PCA immediately. The group represents registered vehicles, so the group signature scheme must be dynamic (unfixed number of group participants) [10].

Vehicle Registration Process (Protocol 1): Before registration, the vehicle needs to generate another public-private key pair for encryption and decryption (PK and sk), and the PK will be sent to the LTCA along with the registration request (Step 1.1). After deciding to register the vehicle as a member, the LTCA signs LK_v and sends it along with the vehicle's encryption public key (PK) to RA, who randomly selects a group signing key gsk_i and encrypts it with PK, then sends C_{gsk} ($C_{gsk} \leftarrow Enc(PK, gsk_i)$) back to the LTCA (Steps 1.2-1.8). Encrypting gsk_i prevents the LTCA from obtaining group signing keys. Otherwise, the LTCA can log the group signing key and send it to an unregistered vehicle, which can sign the ghost ticket (will be mentioned in the Pseudonym Issuance Process section) and pretend to be a legitimate vehicle to gain the trust of the PCA. Upon receiving the response from RA, the LTCA first generates the commitment key (CK_{LTC}) by $CK_{LTC} \leftarrow H(VRegID_{\sigma_{LK_{oem}}} || LK_v || Rnd_{CK_{LTC}} || C_{gsk})$, where $VRegID_{\sigma_{LK_{oem}}}$ is the signature of Original Equipment Manufacturer (OEM), $Rnd_{CK_{LTC}}$ represents the random number, and LK_v is the long-term public key of the vehicle. LTCA sends C_{gsk} along with LTC ($LTC \leftarrow (SN, CN, CK_{LTC}, LK_v, t_s, t_e)$) to the vehicle (Steps 1.9-1.12). Upon successful registration, the vehicle will decrypt C_{gsk} with its private key sk and obtain gsk_i. Therefore, a legitimate member has a group signing key (gsk_i) and a long-term identity (LTC_v), both of which can prove that the vehicle has completed the registration.

Protocol 1 Registration Process

```
1:  $V \rightarrow LTCA : VRegID_{\sigma_{Lk_{oem}}}, (LK_v)_{\sigma_{Lk_v}}, PK, N, t_{now}$

2:  $LTCA : Verify(LK_{oem}, VRegID_{\sigma_{Lk_{oem}}})$
3:  $LTCA : Verify(LK_v, (LK_v)_{\sigma_{Lk_v}})$
4:  $LTCA : (LK_v)_{\sigma_{LTCA}} \leftarrow Sign(Lk_{LTCA}, LK_v)$
5:  $LTCA \rightarrow RA : (LK_v)_{\sigma_{LTCA}}, PK$
6:  $RA : Verify(LTC_{LTCA}, (LK_v)_{\sigma_{LTCA}})$
7:  $RA : C_{gsk} \leftarrow Enc(PK, gsk_i)$
8:  $RA \rightarrow LTCA : C_{gsk}$
9:  $LTCA : Rnd_{CK_{LTC}} \leftarrow GenRnd()$
10: $LTCA : CK_{TLC} \leftarrow H(VRegID_{\sigma_{Lk_{oem}}} || LK_v ||$
        $Rnd_{CK_{LTC}} || C_{gsk})$
11: $LTCA : (LTC)_v \leftarrow (SN, CN, CK_{LTC}, LK_v, t_s, t_e)$
12: $LTCA \rightarrow V : (LTC_v)_{\sigma_{LTCA}}, C_{gsk}, N+1, t_{now}$
13: $V : gsk_i \leftarrow Dec(sk, C_{gsk})$
```

Ticket Issuance Process: The LTCA issues a ticket ($ticket \leftarrow (SN, H(id_{PCA}), CK_{n-tkt}, t_s, t_e, Exp_{n-tkt})$) for the vehicle following protocols 2 and 3 in SECMACE+ [2].

Pseudonym Issuance Process: Upon receiving the ticket, the vehicle must sign the pseudonym request with gsk_i ($tkt_{\sigma_{gsk_i}}$) before sending it to the PCA. After receiving the pseudonym request, according to protocol 4 in SECMACE+ [2], the PCA first verifies the LTCA's signature of the ticket ($Verify(LTC_{LTCA}, (tkt)_{\sigma_{LTCA}})$) to ensure its authenticity and validity. Next, it verifies $tkt_{\sigma_{gsk_i}}$ ($Verify(gpk, tkt_{\sigma_{gsk_i}})$) with the group public key (gpk). This step is a two-factor verification of whether the vehicle has completed registration. If the group signature verification fails, it means that the pseudonym request is using a ghost ticket. Otherwise, it means that the vehicle has a legitimate ticket and has completed the registration, the PCA will trust the pseudonym request.

The adoption of group signatures as two-factor authentication utilizes the properties of group signature scheme, where the PCA knows that it was a member in the group that signed it, but does not know which exact member did this, meaning that the PCA cannot link the group signature and the ticket to a specific vehicle, guaranteeing the privacy of the vehicle. After that, the PCA generates pseudonyms following protocol 4 in SECMACE+ [2]. The vehicle need to sign all pseudonyms after receiving the set of pseudonyms with gsk and broadcast the signature along with the pseudonym and beacons. However, this group signature will not be verified by other vehicles. The use of the group key to sign the pseudonym is for RA to verify that the vehicle is indeed a legitimate member (has a valid group signing key) during the resolution process.

C. RA Resolution Process

Protocol 2 Pseudonym Resolution

```
1: $V : P_v^i \leftarrow (SN^i, K_v^i, CK_{P_v^i}, t_s^i, t_e^i)$
2: $V : (P_v^i)_{\sigma_{P_j^i}} \leftarrow Sign(P_j^i, P_v^i)$
3: $V \rightarrow RA : (Id_{req}, (P_v^i)_{\sigma_{P_j^i}}, (P_v^i)_{\sigma_{gsk_i}}, t_{now})$
4: $RA : Verify(P_j^i, (P_v^i)_{\sigma_{P_j^i}})$
5: $RA : Verify(gpk, (P_v^i)_{\sigma_{gsk_i}})$
6: ...
7: $RA : H(CK_{tkt}||K_{v_i}^i||t_s^i||t_e^i||H'(H^i(Rnd_{psnym}))) \overset{?}{=} CK_{P_{v_i}^i}$
```

Pseudonym Resolution Process (Protocol 2): The report of suspicious behavior submitted by entities (V) includes suspicious pseudonym (P_v^i) and the group signature of that pseudonym ($(P_v^i)_{\sigma_{gsk_i}}$) (Steps 2.1-2.3). Upon receiving a report, RA first verifies the signature of the entity to ensure the authenticity of the report (Step 2.4), and second verifies the group signature of the pseudonym with the group public key (gpk) (Step 2.5). If the group signature is verified, it means that the vehicle was not using a ghost pseudonym (this prevents Sybil attacks Launched by a compromised PCA). After that, RA interacts with the PCA to obtain pseudonym information following protocol 5 in SECMACE+ (Step 2.6) [2]. Finally, the RA checks the commitment key $CK_{P_{v_i}^i}$ of the pseudonym to ensure that this pseudonym was generated based on a ticket (Step 2.7). The PCA who issued the pseudonym would be immediately held accountable and penalized once RA detects a ghost pseudonym.

Ticket Resolution Process (Protocol 3): After performing pseudonym resolution, RA can continue to interact with the LTCA to resolve the real identity of the vehicle. Upon receiving the relevant ticket information from the LTCA, RA first checks whether the ticket is generated based on real vehicle information and then checks the generation of the LTC to confirm the vehicle's real identity (Steps 3.6-3.8). With C_{gsk}, RA can obtain the group signing key bound to the vehicle from the local database, and since RA is the group manager, it can reveal the identity of the signer. Therefore, RA reveals the group signature of the pseudonym ($(P_v^i)_{\sigma_{gsk_i}}$)) to confirm the group signing key used by the signer. Then, RA compares it with the key bound to the LTC. If these two keys are the same, it means that the reported vehicle is indeed registered, and is not joining the system using another vehicle's LTC (Step 3.9)

Protocol 3 Ticket Resolution

```
1: ...
2: $LTCA : \xi \leftarrow (Id_{req}, LTC_v, Rnd_{CK_{n-tkt}}, Rnd_{CK_{LTC}},$
          $VReqID_{\sigma_{Lk_{oem}}}, N+1, t_{now})$
3: $LTCA : (\xi)_{\sigma_{LTCA}} \leftarrow Sign(Lk_{LTCA}, \xi)$
4: $LTCA \rightarrow RA : (\xi)_{\sigma_{LTCA}}$
5: $RA : Verify(LTC_{LTCA}, (\xi)_{\sigma_{LTCA}})$
6: $RA : H(LTC_v||t_s||t_e||Rnd_{CK_{n-tkt}}) \overset{?}{=} CK_{n-tkt}$
7: $RA : Verify(LTC_{oem}, VReqID_{\sigma_{Lk_{oem}}})$
8: $RA : H(VRegID_{\sigma_{LK_{oem}}}||LK_v||Rnd_{CK_{LTC}}||C_{gsk}) \overset{?}{=} CK_{LTC}$
9: $RA : Reveal((P_v^i)_{\sigma_{gsk_i}}) \overset{?}{=} gsk$
```

V. Analysis

A. How this proposed scheme addresses the issue?

The issue addressed in this project is that a compromised LTCA or PCA can intercept legitimate ticket or pseudonym request, thus generating a ghost ticket or ghost pseudonym that can escape checking in the RA's pseudonym resolution process. The mitigation approach is to use group signatures as two-factor authentication, where RA resolves the pseudonym by verifying the group signature of the pseudonym to determine whether the vehicle was using a ghost ticket or ghost pseudonym to join the system.

Ghost Ticket: The reason for ghost ticket situation is that the commitment key contained in the ticket (CK_{tkt}) is a hash value of LTC, which can prevent the PCA from obtaining the true identity of the vehicle. However, the PCA cannot distinguish whether the LTCA issued multiple concurrently valid tickets for the same LTC. To address this issue, RA issues a group signing key gsk for registered vehicles, the pseudonym request should be signed with this gsk. This allows the PCA to verify whether the vehicle is a legitimate member as the second factor. Due to the properties of group signature, the PCA can verify the legitimacy of the signer while not being able to trace its real identity. An unregistered vehicle cannot get a valid group signing key even if it gets a ghost ticket from the compromised LTCA, because RA encrypts the key (gsk) before sending it to the registered vehicle, and the LTCA cannot decrypt it.

Ghost Pseudonym: The ghost pseudonym appeared because the PCA can obtain tickets from legitimate vehicles, thus obtaining the commitment key (CK_{n-tkt}) in that ticket, and generating illegal pseudonyms with the legal CK_{n-tkt}. When RA performs pseudonym resolution, the PCA can return a valid ticket and prove that the pseudonym (ghost) was generated based on a legitimate ticket, so RA cannot detect PCA's malicious behavior. To address this issue, vehicles need to sign the pseudonyms with their group signing key, and broadcast the group signature along with pseudonym and beacons. This allows RA to verify the group signature of the pseudonym to confirm that the vehicle is a legitimate member. If the group signature validation fails, RA will hold the PCA that issued that pseudonym accountable.

The proposed scheme addresses either the Sybil attack launched by the compromised LTCA (ghost ticket) or the Sybil attack launched by the compromised PCA (ghost pseudonym). Although the proposed scheme can also solve the case of both LTCA and PCA being compromised at the same time, but in this case, the VPKI will lose it's usefulness and all the vehicle identities and pseudonyms can be released freely. Secondly, if the compromised LTCA registers vehicles freely, all illegal vehicles can join the system. Therefore, Sybil attack is launched on the premise that VPKI has not lost its proper role and the registration process is in accordance with the protocol (process of getting LTC).

B. *Overheads*

The extra overhead introduced by this proposed solution is mainly caused by group signatures. According to the paper [9], the time required to sign and verify the group signature scheme will be greater than the ECDSA signature. Assume the time for communication between VPKI entities is T_{C2C}, the time for group signature signing is $T_{G_{Sign}}$; verifying is $T_{G_{Verify}}$. The time for encryption is T_E, while that for decryption is T_D. The number of pseudonyms is N. Therefore, the extra time for obtaining pseudonyms by vehicle is:

$$T_{GetP} = \ 2 \times T_{C2C} + T_E + T_D + T_{G_{Sign}} + T_{G_{Verify}} \tag{1}$$

The extra time for signing pseudonyms is:

$$T_{SignP} = \ N \times T_{G_{Sign}} \tag{2}$$

The total extra overhead is:

$$T_{Total} = (1) + (2) \tag{3}$$

The process of generating the group keys by RA is done during the initialization of the VPKI and does not increase the latency of obtaining pseudonyms by the vehicle.

VI. Evaluation

A. *Implementation*

To implement the prototype of the proposed scheme, I used four virtual machines to simulate the vehicle, LTCA, RA, and PCA respectively, Python to implement the interactions between them, and evaluated the time required and the extra latency introduced by the scheme. I mainly used Python's *cryptograph* library to implement the public and private keys for ECDSA p-256 and RSA encryption & decryption, this library can also be used for signing and verifying signatures. For communication between VMs I used *socket* and *SSL* library. To simplify the interaction, the server's certificate is not verified when establishing communication. I used the JSON format for the message in communication. Since there is no well-established implementation of the group signature scheme, so I replaced the steps that require group signatures with the use of ECDSA-256. Then, evaluated the total delay based on the the signing and verification speeds of ECDSA-256 and the experimental results in the paper [9]. Finally, estimated the time required for group signatures in this proposed scheme. The implementation code can be found in the GitHub repository (https://github.com/Keyao7/ANSS-Project).

B. *Results*

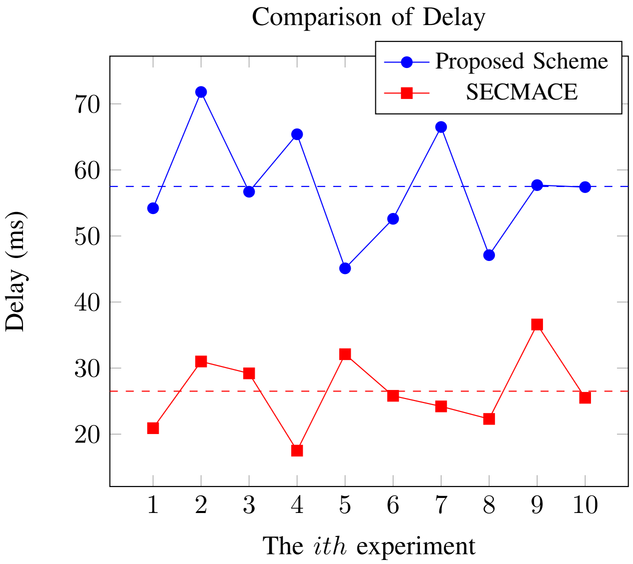

Figure above shows the results of ten experiments. According to the article [9], the time required to sign with the group signature (GS) scheme is approximately 6.5 times of ECDSA-256, and that for verification is about 12 times of ECDSA-256 verification. The blue dashed line indicates the average delay of the proposed scheme, which is **57.5ms**, while the red one represents the average delay of SECMACE, which is **26.5ms**. Therefore, the extra delay for a vehicle to get pseudonyms is:

$$T_{GetP} = 57.5ms - 26.5ms = 31ms \tag{4}$$

VII. Conclusion

This project focuses on mitigating the issue of ghost ticket and ghost pseudonym issued by a compromised CA (LTCA or PCA). The proposed scheme uses group signatures as two-factor verification, which allows RA to detect and evict ghost vehicles while punishing the malicious CA. The author implements the proposed scheme and measures the extra overhead and latency introduced by the scheme. Since there is no well-established implementation of group signatures yet, the author uses ECDSA-256 as a substitute and estimates the time required for group signatures. Finally, the authors evaluated an additional delay of 31ms for the vehicle to obtain pseudonyms, which is acceptable.

References

[1] M. Khodaei, H. Jin, and P. Papadimitratos, "Secmace: Scalable and robust identity and credential management infrastructure in vehicular communication systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1430–1444, 2018.

[2] M. Khodaei, H. Noroozi, and P. Papadimitratos, "Secmace+: Upscaling pseudonymous authentication for large mobile systems," *IEEE Transactions on Cloud Computing*, 2023.

[3] T. H.-J. Kim, L.-S. Huang, A. Perrig, C. Jackson, and V. Gligor, "Accountable key infrastructure (aki) a proposal for a public-key validation infrastructure," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 679–690, 2013.

[4] D. Basin, C. Cremers, T. H.-J. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "Design, analysis, and implementation of arpki: an attack-resilient public-key infrastructure," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 393–408, 2016.

[5] S. Khan, L. Zhu, X. Yu, Z. Zhang, M. A. Rahim, M. Khan, X. Du, and M. Guizani, "Accountable credential management system for vehicular communication," *Vehicular Communications*, vol. 25, p. 100279, 2020.

[6] A. Tesei, L. Di Mauro, M. Falcitelli, S. Noto, and P. Pagano, "Iota-vpki: A dlt-based and resource efficient vehicular public key infrastructure," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–6, IEEE, 2018.

[7] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for v2x communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.

[8] S. Khan, F. Luo, Z. Zhang, M. A. Rahim, M. Ahmad, and K. Wu, "Survey on issues and recent advances in vehicular public-key infrastructure (vpki)," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1574–1601, 2022.

[9] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "On the performance of secure vehicular communication systems," *IEEE transactions on dependable and secure computing*, vol. 8, no. 6, pp. 898–912, 2010.

[10] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "Security, privacy, and incentive provision for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 839–853, 2016.

[11] E. Rescorla, "The transport layer security (tls) protocol version 1.3," tech. rep., 2018.