# Revocation Protocol for Group Signatures in VANETs: A Secure Construction

**Nur Fadhilah Mohd Shari[1], Amizah Malip[1*] and Wan Ainun Mior Othman[1]**
[1] Institute of Mathematical Sciences, Faculty of Science, University of Malaya
50603, Kuala Lumpur, Malaysia
[e-mail: fadhilahshari@siswa.um.edu.my, amizah.malip@um.edu.my, wanainun@um.edu.my]
*Corresponding author: Amizah Malip

*Received February 17, 2019; revised May 2, 2019; accepted July 22, 2019; published January 31, 2020*

## Abstract

Vehicular ad hoc networks (VANETs) enable wireless communication between vehicles and roadside infrastructure to provide a safer and more efficient driving environment. However, due to VANETs wireless nature, vehicles are exposed to several security attacks when they join the network. In order to protect VANETs against misbehaviours, one of the vital security requirements is to revoke the misbehaved vehicles from the network. Some existing revocation protocols have been proposed to enhance security in VANETs. However, most of the protocols do not efficiently address revocation issues associated with group signature-based schemes. In this paper, we address the problem by constructing a revocation protocol particularly for group signatures in VANETs. We show that this protocol can be securely and efficiently solve the issue of revocation in group signature schemes. The theoretical analysis and simulation results demonstrate our work is secure against adversaries and achieves performance efficiency and scalability.

***Keywords:*** Revocation, group signature, vehicular ad hoc networks, security, cryptographic protocols

This work was supported by University of Malaya under Grant No. BK016-2015

**http://doi.org/10.3837/tiis.2020.01.017** **ISSN : 1976-7277**

## 1. Introduction

The growing concern in road safety and traffic efficiency has drawn a significant interest towards the development of vehicular ad hoc networks (VANETs) [1-7]. In VANETs, vehicles can communicate with each other (V2V) and with the roadside units, known as infrastructure (V2I) to announce safety messages [8]. The communication scenario is illustrated in **Fig. 1**. With the safety information, drivers can anticipate any harmful situations and take actions accordingly. However, safety can only be achieved if messages broadcasted are trustworthy [9]. To evaluate trustworthiness of a message, receiving vehicle performs verification check on the message received. This arises an issue of privacy as such message verification may reveal some information about the sender, leading to profiling of a vehicle by an adversary. Profiling is an act of classifying messages into pre-defined profiles, thus gaining useful information about the sender [10]. The presence of adversaries is a common assumption in VANETs [6,9,11-14]. They can be categorised as internal and external adversaries. An internal adversary is a legitimate user who possesses credentials in VANETs while an external adversary does not possess one [14]. The internal adversary can be further broken down into a revoked adversary who is a legitimate user that has been revoked in the system but still owns its initial keys. This type of adversary may misuse his legitimacy to mislead any other vehicle by manipulating the content of a message, impersonating other vehicle's identity and intercepting communication between two parties. In our work, we assume the presence of internal adversaries since most of external attacks can be prevented by means of authentication and privacy protection. Moreover, this is a common assumption in the literature [12-14].

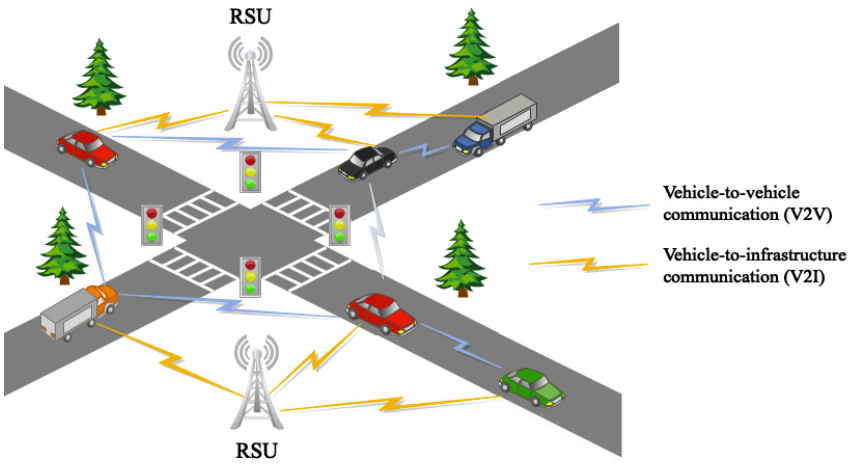


**Fig. 1.** Communication in VANETs

Vehicles would be less likely to participate in VANETs if the system is vulnerable to attacks. The system vulnerability may render the technology to be unutilized. In order to protect VANETs against misbehaviours, addressing a secure and efficient revocation is indispensable in the network [15]. Revocation is vital to ensure these misbehaved vehicles are held accountable for their own actions and to prevent them from further participation in the network.

At the same time, privacy should be preserved in VANETs. Group signatures [7,11,12,16-22] and pseudonymous mechanisms [4,8,13,14,23,24] are the two common techniques to provide privacy in VANETs. Some may favour group signatures over pseudonymous public key due to the storage efficiency and ease of certificate management [11]. In order to make VANETs beneficial to the users, a privacy-preserved scheme such as group signatures should address a practical revocation protocol in the construction.

There are two conditions to be considered for a practical revocation protocol. First, the revocation procedure should be integrated with other security requirements. Second, an efficient revocation procedure is required as delay in revoking the misbehaved vehicles may open up the possibility for them to continue jeopardizing the safety of other vehicles. To address these conditions, we propose a secure and efficient revocation protocol particularly for group signature schemes in VANETs. Our proposed protocol can integrate with the requirement of privacy provided by group signature schemes. The identity of an adversary will only be revealed when it has been found misbehaved. The efficiency of our revocation protocol is comparable to other revocation techniques presented in other group signatures schemes in the literature. Furthermore, our work efficiently addresses the issue of revocation in Wu et al.'s scheme [11] where revocation was discussed but no explicit mechanism was presented. While our revocation protocol is proposed to address the gap in [11], it is adaptable to other group signature schemes of similar setup and construction.

The remainder of this paper is organised as follows. We discuss related work in this research topic in Section 2. We then briefly review the group signature scheme proposed in [11] in Section 3. In Section 4, we introduce our revocation construction. We analyse the security and performance efficiency of our protocol in Section 5 and Section 6 respectively. Finally, we conclude the paper in Section 7.

## 2. Related Work

A number of revocation protocols have been proposed to address the issue of revocation in group signature schemes [7,11,12,16-22]. The most common way to revoke misbehaved vehicles in group signatures is by using Verifier-Local Revocation (VLR) [12,16-18]. VLR was introduced by Boneh and Shacham in 2004 [25] where revocation information stored in the Revocation List (RL) is distributed to verifiers for check-up operation. Calandriello et. al [17] and Studer et. al [18] depend on VLR to achieve revocation. However, VLR method is efficient only if there are a few revoked vehicles in the network. This method becomes computationally inefficient when a large number of revoked vehicles exist in the list.

Some other group signatures schemes such as in [19,20] replaced the RL checking process during message verification phase with a Hash Message Authentication Code (HMAC) checking. A detailed mechanism of HMAC can be found in [26]. Adopting HMAC method requires low storage space as the size of the HMAC is smaller than the size of the certificate. However, the involvement of Road Side Units (RSUs) in these two schemes [19,20] is needed to filter out revoked vehicles who enter their communication range using the revocation lists, so that only unrevoked vehicles are able to obtain group keys and announce messages with HMAC values attached. The same role of RSU is also needed in [7,21,22] where revocation check is performed by the RSU before issuing a group key for each legitimate vehicle that passes by its domain.

Meanwhile, in other group signature schemes where no reliance is placed on RSUs during message broadcast phase such as in Lin et al's [16] and Chen et al.'s [12], VLR is

used in conjunction with an additional method for a more efficient system. In Lin et al.'s scheme called Group Signature and Identity-based Signature (GSIS) [16], VLR plays its role when the number of revoked vehicles is below a certain threshold. When it exceeds the threshold, updating the key pairs of unrevoked vehicles is required. Thus, the revoked vehicles who are unable to obtain their keying materials updated will automatically be excluded from the network. Meanwhile, in Chen et al.'s scheme, called Threshold Anonymous Announcement (TAA) [12], its revocation construction is almost similar to GSIS scheme, only that TAA requires both Trusted Parties (TPs) and unrevoked vehicles key pairs to be updated. This is necessary in [12] since the TPs key is used in the verification phase by the verifiers. To update the keys, interval communication between vehicles and the TP may be required since TAA does not entirely assume the availability of RSUs. Vehicles may also interact with the TP during regular maintenance visit or at VANET service points. In GSIS, although the role of RSU is not needed during message broadcast between vehicles, its involvement is assumed only to relay information such as to announce key update in executing revocation.

The revocation protocols discussed earlier may not be a suitable solution to be implemented in [11] due to inefficiency of adopting VLR alone in [17,18], zero reliance on RSUs to broadcast message in [11] as opposed to [7,19-22] and challenges to update credentials compared to [12,16] as vehicles generate their own tracing information that was sent to the TP during registration phase. Our main goal is to provide a promising solution for revocation, which is imperative in VANETs. The contributions of this paper are as follows:

- We highlight the flaw in [11], which shows that the scheme does not achieve its security objectives when revocation protocol is not presented in the scheme.
- We design a generic abstraction of revocation protocols for group signature-based schemes which aims to provide the basis for designation of revocation protocol in group signatures.
- We propose a new revocation protocol that completes the scheme in [11] which combines the use of VLR with an additional technique of updating credentials and tracing information.
- We run a simulation of our revocation protocol. We provide an analysis and present experimental results that validate the efficiency and scalability of our protocol.

## 3. Wu et al.'s Scheme

In this section, we provide an overview of Wu et al.'s scheme [11], called Message-Linkable Group Signature (MLGS). There are three different roles of authorities in this scheme, which are a vehicle manufacturer ($\mathcal{VM}$), a group registration manager ($\mathcal{RM}$), and a tracing manager ($\mathcal{TM}$). To enroll into a VANET system, each vehicle, $\mathcal{V}$ signs a contract with the ($\mathcal{VM}$) to confirm the vehicle ownership and generates its own public key $Y = U_1^y$ for a random value $y \in \mathbb{Z}_p^*$, where $y$ is the secret key. Then, $\mathcal{V}$ registers to the $\mathcal{RM}$ to become a legitimate group member by sending its public-private key pair $(Y, y)$. During registration, $\mathcal{V}$ also sends a tracing information $T = g_2^y$ to the $\mathcal{TM}$ so that $\mathcal{TM}$ can trace the vehicle if in case of dispute. When $\mathcal{V}$ has successfully registered to the system, $\mathcal{V}$ will receive a sign on its public key from the $\mathcal{RM}$ and use the signature as a group certificate to announce safety messages. $\mathcal{VM}$, $\mathcal{RM}$, and $\mathcal{TM}$ are assumed as semi-trusted parties since they have no access

to the private key of vehicles. **Table 1** shows the lists of some notations related to our work which was adopted from MLGS [11] to ease the reading throughout this paper.

**Table 1.** Notations and Descriptions [11]

| Notation | Description |
|---|---|
| $\mathcal{VM}$ | Vehicle manufacturer |
| $\mathcal{RM}$ | Registration manager |
| $\mathcal{TM}$ | Tracing manager |
| $\mathcal{V}$ | Vehicle |
| $T = g_2^y$ | Tracing information of $\mathcal{V}$ |
| $(Y, y)$ | $\mathcal{V}$'s public-private key pair |
| $(A, Z)$ | $\mathcal{RM}$'s public-private key pair |
| $m$ | A message |
| $\sigma$ | A signature on message $m$ |
| $\sigma_i$ | The $i$-th component of $\sigma$ |
| $M = (m, \sigma)$ | A message appended with a signature |
| $H_1(\,)$ | A cryptographic hash function from $\{0,1\}^*$ to $\mathbb{G}_1$ |
| $\mathbb{G}_i (i = 1,2,3)$ | Finite cyclic group of prime order $p$ |
| $g_i$ | A random generator of $\mathbb{G}_i$ |
| $U_2, h_2 \in \mathbb{G}_2$ | Public system parameters |
| $\emptyset$ | An isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ |
| $U_1 = \phi(U_2)$ | Public system parameter |
| $h_1 = \phi(h_2)$ | Public system parameter |
| $K_v = (K_1, K_2)$ | $\mathcal{V}$'s group certificate |

The goal of MLGS scheme is to provide an efficient trustworthy system with a balanced public safety and vehicle privacy. Threshold authentication is used to satisfy the trustworthiness property. (Say $n$) MLGS signatures are generated by $n$ distinct registered vehicles on messages of the same content. A receiver verifies $n$ signatures using the $\mathcal{RM}$'s public key, $A$ to validate the group certificates. If these $n$ signatures are valid and if $n$ satisfies the threshold, the message is considered trustworthy. To protect the privacy of vehicles, this scheme allows each vehicle to generate only one message-link identifier $\sigma_4 = H_1(m)^y$ for the same message. This approach enables a vehicle to remain anonymous if it generates one signature on each message but this vehicle can be traced once it produces two signatures on the same message as the two signatures share the same component $\sigma_4$. Thus, anonymity is preserved as long as the vehicle does not misbehave by generating two signatures on the same message. However, user safety is compromised in MLGS when no revocation technique was proposed. This leads to system vulnerabilities when adversaries are not accountable for their attacks. In the next section, we show our construction of a revocation scheme that can be efficiently deployed in MLGS.

## 4. Revocation Construction

### 4.1 Generic Revocation Construction

Based on our analysis of different revocation protocols proposed in existing group signature schemes (presented in Section 2), we formulate a generic abstraction of revocation protocols

for group signature-based schemes. To the best of our knowledge, this is the first generic revocation construction exists in the literature that systematically studies and generalise revocation protocols of group signatures schemes in VANETs.

In this section, we present our generalisation of the protocols demonstrated in **Fig. 2**. The network consists of a trusted party ($\mathcal{TP}$), and vehicles ($\mathcal{V}$). Before describing the abstraction, we review the main role of each entity as follows:

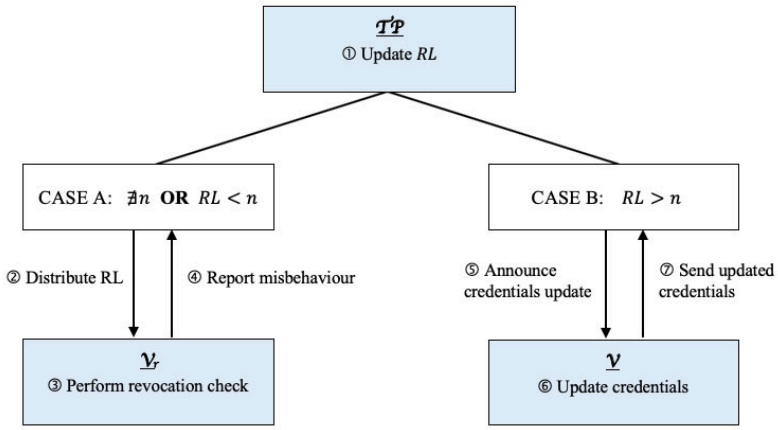


**Fig. 2.** Generic Revocation Construction

**Trusted Party** ($\mathcal{TP}$). A $\mathcal{TP}$ is responsible for the distribution and management of the revocation list (RL). This $\mathcal{TP}$ is commonly known as a Trusted Authority ($\mathcal{TA}$) [18-20], a Tracing Manager ($\mathcal{TM}$) [7,11,16,22], a Certificate Authority ($\mathcal{CA}$) [17,21], and an Issuer ($\mathcal{I}$) [12].

**Vehicle** ($\mathcal{V}$). $\mathcal{V}$ has two roles in VANETs:

- Sending vehicle $\mathcal{V}_s$ sends messages in the network. $\mathcal{V}_s$ is further divided into two categories; unrevoked vehicles $V_{s_u}$ and revoked vehicles $\mathcal{V}_{s_r}$.
- Receiving vehicle $\mathcal{V}_r$ receives and verifies the messages. In some schemes [7,19-22], revocation check is not performed during message verification phase, but it is performed during authentication phase by RSUs whenever a vehicle sends a message request to acquire a short-term group key. In such situation, we refer RSUs as $\mathcal{V}_r$. We do not distinguish between revoked and unrevoked vehicles in $\mathcal{V}_r$ since the task of receiving messages could cost no harm to the network.

### 4.1.1 Description of the Revocation Construction

The abstraction of revocation shown in Figure 2 is composed of the following steps:

①Firstly, $\mathcal{TP}$ updates the revocation list (RL). There are two cases to be considered after updating the RL.

- CASE A: Either a threshold method is not adopted in the mechanism, denoted by $\nexists n$ in Figure 1 (such as RSU reliance revocation, VLR without threshold or traditional

distribution of RL) or if it does, the number of revoked vehicles $\mathcal{V}_{s_r}$ in the RL is less than a predefined threshold $n$, denoted by $RL < n$.

- CASE B: The number of revoked vehicles $\mathcal{V}_{s_r}$ in the RL exceeds the threshold $n$, denoted by $RL > n$.

**For CASE A:**

②$\mathcal{TP}$ distributes the updated RL to $\mathcal{V}_r$ via the wireless channel.

③$\mathcal{V}_r$ uses the received RL to perform revocation check in order to verify if a vehicle is revoked or not. If there is an identity matched, $\mathcal{V}_r$ rejects the message. If not, the message is accepted, provided the message originates from a legitimate sender.

④If $\mathcal{V}_r$ experiences any misbehaviours, it may lodge a report and send it to $\mathcal{TP}$ via the wireless channel.

**For CASE B:**

⑤TP announces a credential update via wireless channel. If the credential update is performed by $\mathcal{TP}$ the process is simple as $\mathcal{TP}$ only updates unrevoked vehicle, $V_{s_u}$'s credential and makes it available to $V_{s_u}$. However, if the credential update is performed by vehicles; $\mathcal{V}_s$ and $\mathcal{V}_r$, thus step ⑥ and ⑦ follow.
(Note that, there is a circumstance where $\mathcal{TP}$'s credential also should be updated. Such update will be performed by $\mathcal{TP}$.)

*These last two steps can be discarded if the credential update was performed by the $\mathcal{TP}$.*

⑥$\mathcal{V}_s$ and $\mathcal{V}_r$ update its own credential using a unique value distributed by $\mathcal{TP}$.

⑦$\mathcal{V}_s$ and $\mathcal{V}_r$ send to $\mathcal{TP}$ its updated credential for authentication or tracing purposes.

### 4.2 Our Proposed Construction

Our proposed revocation protocol has minimal reliance on RSUs. The involvement of the RSUs is only needed to relay information and to provide a gateway between a trusted party and vehicles. Furthermore, we utilize the generic abstraction presented in Section 4.1 to define the structure of our revocation protocol.

#### 4.2.1 VLR Adoption

The construction begins with the adoption of VLR method [25]. This is a common approach for group signature schemes in VANETs. VLR is used in the revocation check during the verification of a signature. According to Bringer in [27], the verification phase should be divided into two parts; `revocation check' and `signature check'. The former is to verify if the signing vehicle has been revoked or not whereas the latter is to check if the signing vehicle is a legitimate member in the network. When analysing the MLGS scheme, we found its verification phase only applies `signature check' without being curious to identify if the vehicles have been revoked from the system or not. Hence, the adoption of VLR is applicable in MLGS since there is no `revocation check' being implemented. Nevertheless, due to the downside of VLR discussed in Section 2, we choose to apply VLR when a number of revoked vehicles in RL is below a certain threshold, and credential update when the number of revoked vehicles in RL exceeds the threshold.

The detailed mechanism of VLR adoption is described according to the generic revocation structure in Section 4.1 as below:

①$\mathcal{TM}$ updates the Revocation List, $RL = \{Y_{v_1}, \ldots, Y_{v_i}\}$.

②$\mathcal{TM}$ distributes the revocation list to $\mathcal{V}$ when $i < n$ is a predefined threshold.

③Upon receiving a message $m$ that contains a signature $\sigma_{v_i}$, $\mathcal{V}$ first performs revocation check operation by checking $\sigma_2 = K_2\{(h_1 Y_i)\}^s$ for each $Y_i$ in the RL. If there is a matched $Y_i$, the message will be discarded. If not, the message is considered as valid and $\mathcal{V}$ continues to perform signature check operation to validate the signature.

④$\mathcal{V}$ also lodges a revocation report to $\mathcal{TM}$ when repetition of $\sigma_4$ is found as it indicates an attempt of misbehavior in MLGS.

### 4.2.2 Credentials Update

We apply an additional revocation mechanism when the number of revoked vehicles exceeds the predefined threshold. Similar technique was also adopted in [12,16]. In MLGS, since the $\mathcal{RM}$'s key is used by the verifiers during the verification phase, updating both its $\mathcal{RM}$'s key and vehicles' credentials is necessary for revocation. However, updating vehicles' credentials is an issue in MLGS since the vehicles generate their own tracing information which was sent to the $\mathcal{TM}$ during the registration phase. Our proposed revocation protocol includes the process of updating the tracing information, thus solving the issue in MLGS.

Since we adopt a threshold method for the size of revoked vehicles in the revocation list, the revocation construction follows until the last step of the generic abstraction presented in Section 4.1:

**Algorithm 1**: Register Manager ($\mathcal{RM}$)

**Initially:** $(pk_{\mathcal{RM}}, sk_{\mathcal{RM}} = (A, Z) = (e(Z, g_2), Z)$

**Update:** $\left(\dot{pk}_{\mathcal{RM}}, \dot{sk}_{\mathcal{RM}}\right) = \left(e\left(\dot{Z}, g_2\right), \dot{Z}\right)$ where $\dot{Z} \in \mathbb{Z}_p^*$

**Publish:** $\dot{A}$, while $\dot{Z}$ is kept secret

**Fig. 3.** $\mathcal{RM}$'s Credential Update

⑤$\mathcal{TM}$ announces a credential update to be performed by $\mathcal{V}$. At the same time, $\mathcal{RM}$ updates its credential. The $\mathcal{RM}$ has public-private key pair $(A, Z)$ where $A = e(Z, g_2)$. To update its key, the $\mathcal{RM}$ first updates its private key $Z$ to a new value $\dot{Z} \in \mathbb{Z}_p^*$. The $\mathcal{RM}$ then updates its public key $A$ to $\dot{A} = e(\dot{Z}, g_2)$. The $\mathcal{RM}$ can now publish its new public key $\dot{A}$ to be used across the network via the RSU while its secret key is kept private. The detailed algorithm is illustrated in **Fig. 3**.

⑥$\mathcal{V}$ updates its credential (shown in **Fig. 4**) in which tracing information is also updated in this process. Since the credential update is performed by $\mathcal{V}$ in MLGS, the $\mathcal{TM}$ distributes a new value $x \in \mathbb{Z}_p^*$ to $\mathcal{V}$ in the system. $\mathcal{V}$'s public-private key pair is $(Y, y)$ where $Y = U_1^y$. By having the new value $x$, $\mathcal{V}$ updates its private key $y$ first to $\dot{y} = y^x$. Then, $\mathcal{V}$ updates its public key $Y$ to $\dot{Y} = \left(U_1^y\right)^x$. Now, $\mathcal{V}$ has its new key pair $\left(\dot{Y}, \dot{y}\right) = \left(\left(U_1^y\right)^x, y^x\right)$ and can use $\dot{Y}$ across the network. Using the new key, $\mathcal{V}$ computes a new tracing information $\dot{T} = \left(g_2^y\right)^x$.

⑦$\mathcal{V}$ sends its new tracing information together with the old signature of $\mathcal{VM}$ on $Y$ and the new $\dot{Y}$ to $\mathcal{TM}$. Upon receiving the information, $\mathcal{TM}$ first verifies the legitimacy of $\mathcal{V}$ by

checking the signature and the new key. Then, $\mathcal{TM}$ verifies the traceability of $\mathcal{V}$ in case of dispute by checking the new tracing information if $e(\dot{Y}, g_2) = e(U_1, \dot{T})$ where $e(\left(U_1^y\right)^x, g_2) = e(xU_1, x\dot{T})$. If both checks hold, $\mathcal{TM}$ generates a signature on $\dot{Y}$ and sends it to $\mathcal{V}$. Then $\mathcal{V}$ sends the received signature to $\mathcal{RM}$ to acquire a new group certificate $K_\mathcal{V} = (K_1, K_2)$. Upon receiving the signature $\mathcal{RM}$ checks its validity. If the check holds, $\mathcal{RM}$ generates $K_v$ to $\mathcal{V}$.

**Algorithm 2**: Vehicle ($\mathcal{V}$)

**Initially:** $(pk_\mathcal{V}, sk_\mathcal{V} = (Y, y) = (U_1^y, y)$

**Update:** $\left(p\dot{k}_\mathcal{V}, s\dot{k}_\mathcal{V}\right) = \left(\left(U_1^y\right)^x, \mathrm{y}^{\mathrm{x}}\right)$ where $x \in \mathbb{Z}_p^*$

**Publish:** $\dot{Y}$ while $\dot{y}$ is kept secret

**For tracing purpose:**
$\mathcal{V}$ computes $\dot{T} = \left(g_2^y\right)^x$

$\mathcal{V} \xrightarrow{\sigma_{\mathcal{VM}}(Y), \dot{Y}, \dot{T}} \mathcal{TM}$

$\mathcal{TM}$ verifies:

- $\sigma_{\mathcal{VM}}(Y)$
- $\dot{Y}$
- $e\left(\dot{Y}, g_2\right) = e(U_1, T)$ where $e\left(\left(U_1^y\right)^x, g_2\right) = e\left(xU_1, x\dot{T}\right)$

$\mathcal{TM} \xrightarrow{\sigma_{\mathcal{TM}}(\dot{Y})} \mathcal{V} \xrightarrow{\sigma_{\mathcal{TM}}(\dot{Y})} \mathcal{RM}$

$\mathcal{RM}$ verifies:

- $\sigma_{\mathcal{TM}}(\dot{Y})$

$\mathcal{RM} \xrightarrow{K_\mathcal{V} = (K_1, K_2)} \mathcal{V}$

**Fig. 4.** $\mathcal{V}$'s Credential Update

$\mathcal{TM}$ will be able to identify a revoked vehicle if it attempts to update its credential by sending its tracing information. $\mathcal{TM}$ will not sign its key nor will it validate the tracing information sent. This prevents revoked vehicles from further contacting $\mathcal{RM}$ to acquire a new group certificate, thus no longer be able to continue its future participation in the network.

# 5. Security Analysis

## 5.1 Fulfilment of Accountability Requirement

Some schemes in the literature [6,8,9,11,12] highlighted the following three security requirements as critical concerns to be met towards VANETs deployment:

- **Trustworthiness.** To view a message as trustworthy, it must be sent unmodified by a legitimate vehicle. Moreover, the message sent must reflect the actual event.
- **Privacy.** The identity of the sending vehicle should be protected unless it misbehaved. Furthermore, if two different messages are generated by the same sender, they cannot be linked to each other.
- **Accountability.** If misbehaviour arise, the misbehaved vehicles can be traceable. Moreover, they must satisfy non-repudiation, that is, the assurance that they are the message originator. The misbehaved vehicle can then be revoked from the network.

We show that our construction completes the security requirement of accountability in MLGS. The requirement of trustworthiness and privacy are not discussed here as these properties have been addressed in [11].

Accountability can only be achieved if it satisfies the properties of traceability, non-repudiation and revocation. Traceability is satisfied in MLGS when a malicious vehicle who produces two or more signatures on the same message can be traceable since $\mathcal{V}$ can only generate one identifier, indicated by $\sigma_4$ for the same message. Non-repudiation is attained as each vehicle generates its own secret key $y$ without being known by other entities, including semi-trusted entities ($\mathcal{VM}, \mathcal{TM}, \mathcal{RM}$). However, revocation is not supported in MLGS since there is no explicit revocation mechanism presented in [11]. Thus, the requirement of accountability is not achieved in MLGS. By adopting our proposed revocation protocol presented in this paper, the accountability requirement is now satisfied.

**Table 2.** Comparison of Accountability Analysis

| | Traceability | Non-repudiation | Revocation |
|---|---|---|---|
| TAA [12] | √ | √ | √ |
| GSIS [16] | √ | X | √ |
| Hybrid [17] | √ | X | √ |
| TACK [18] | √ | X | √ |
| HMAC [19] | √ | X | √ |
| HMAC v2 [20] | √ | X | √ |
| CMAP [21] | √ | X | √ |
| Signcryption [22] | √ | X | √ |
| MLGS [11] | √ | √ | X |
| Our work | √ | √ | √ |

We compare the functionalities of accountability requirement with other group signature schemes in VANETs (illustrated in **Table 2**). All schemes satisfy the traceability property. However, only MLGS and TAA achieve non-repudiation since the vehicle in both of these schemes is the sole holder of its secret key. Lastly, looking at the revocation column in **Table 2**, MLGS is the only scheme that does not achieve revocation property, but with our proposed revocation construction, the issue is solved.

### 5.2 Robustness against Attacks

We analyze the robustness of our revocation protocol in the presence of adversaries which we classify into three categories; external adversaries, internal adversaries, and revoked adversaries. External adversaries are outsiders who did not register in the network. Internal adversaries are registered vehicles who are in possession of all valid keys. Meanwhile, revoked adversaries who are also registered vehicles, do not possess updated keys but still keeping the initial keys.

In this paper, we consider the attacks originated from registered vehicles; internal adversaries and revoked adversaries since the purpose of revocation is to remove registered misbehaved vehicles from the network. We then define the probable attacks in vehicular networks namely impersonation attack, sybil attack, and man-in-the-middle attack below. We then present the robustness of our work against these attacks in the next subsection.

1. **Impersonation attack.** An adversary masquerade as other legitimate vehicles by impersonating their identities and use them to manipulate revocation report.
2. **Sybil attack.** An internal adversary reports a false revocation and endorse the report by computing as many signatures as required to make the trusted party believe that the report is generated by different vehicles.
3. **Man-in-the-middle attack.** An adversary eavesdrops, intercepts, and may modify the revocation report sent by the vehicle to the trusted party, who believe they are directly communicating with each other.

We adopt proof technique similar to that used in [11] since the construction of protocol is based on similar computational assumptions i.e. the Diffie-Hellman Knowledge (DHK) assumption. The DHK is described as follows. Let $\mathbb{G}$ be a finite cyclic group of prime order $p$ and $g$ the generator of $\mathbb{G}$. Given $(g, g^a) \in \mathbb{G}^2$ where $a \in \mathbb{Z}_p^*$, any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ has only negligible probability of creating a Diffie-Hellman tuple $(g, g^a, g^r, g^{ar})$ without knowing $r$.

#### 5.2.1 Robustness against Internal Adversaries

1. Robustness against Impersonation Attacks

**Claim 1:** *An internal adversary is not able to impersonate other legitimate vehicle's identity to generate false revocation reports.*

To execute this attack, the internal adversary has to impersonate a vehicle's certificate and make a false revocation report in the name of the vehicle. However, the protocol is able to detect the forgery, hence, the internal adversary cannot impersonate an identity, let alone make false accusation to deceive other vehicles.

**Proof.** We consider an internal adversary $\mathcal{B}$ attempts to steal identity of a vehicle $\mathcal{V}$. We show that if $\mathcal{B}$ can steal a valid group signature, then the DHK assumption in $\mathbb{G}_1$ of pairing groups $\Upsilon$ holds.

Given that $\mathcal{B}$ can request the public system parameters. We run the DHK challenger in pairing groups $\Upsilon = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e) \leftarrow PGen(1^\lambda)$ where the DHK assumptions hold in $\mathbb{G}_1$. Let $\psi$ is an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ such that $\psi(g_2) = g_1$. Let $h_2, U_2$ be randomly chosen from $\mathbb{G}_2$ and hence, we can efficiently compute $\psi(h_2) = h_1,\ \psi(U_2) = U_1$. We receive a challenge $(g_2, h_2) = \left(g_2, g_2^\beta\right) \in \mathbb{G}_2^2$ where $\beta \in \mathbb{Z}_p^*$ is undisclosed. Then, we compute $(g_1, h_1)$ where $h_1 = \psi(h_2)$. Note that $\beta$ is unknown to us such that $h_1 = g^\beta$. We

choose two hash functions $H_1(.):\{0,1\}^* \rightarrow \mathbb{G}_1$ and $H(.):\{0,1\}^* \rightarrow \mathbb{Z}_p$ at randoms where $H$ and $H_1$ are modeled as random oracles which makes the outputs of both hash functions are also random. $\mathcal{B}$ has to query the oracles in order to compute its output. Then, we have system parameters

$$\mu = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e; h_2, h_1; U_2, U_1; H_1, H \rangle \quad (1)$$

When $\mathcal{B}$ wants to enter the network, $\mathcal{B}$ requests the system parameters $\mu$ and thus, we forward $\mu$ to $\mathcal{B}$. Then, we generate $A$ and send $A$ to $\mathcal{B}$ where $A$ is $\mathcal{RM}$'s public key. When $\mathcal{B}$ requests a group certificate and the signature of $Y$ from $\mathcal{TM}$, we can generate the group certificate provided we know the secret key of $\mathcal{B}$, $y$ of the registered public key, $Y$ and $\mathcal{RM}$'s secret key, $Z$ to satisfy $e(Z, g_2)$. $Z$ is known to us, but not $y$. To obtain $y$, we run a zero-knowledge proof $ZK\{y|Y = g_1^y\}$ with $\mathcal{B}$ by invoking $\mathcal{B}$ twice according to Forking Lemma in [28].

Now, $\mathcal{B}$ impersonates legitimate vehicle's identity which has a group signature $\sigma = (\sigma_1, \ldots, \sigma_6)$. Due to the impersonation, the tracing information equation

$$e(\sigma_2, g_2) = e(\sigma_1, T) \quad (2)$$

does not hold since the illegal activity is valid. At this point, we proceed with the verification procedure assuming revocation check is not able to locate $\mathcal{B}$. The two verification equations

$$e(\sigma_2, g_2) = Ae(\sigma_1, h_2)e(\sigma_3, g_2) \quad (3)$$

$$\sigma_5 = H(m||\sigma_1||\sigma_2||\sigma_3||\sigma_4||H_1(m)^{\sigma_6}\sigma_4^{\sigma_5}||\sigma_1^{\sigma_6}\sigma_3^{\sigma_5}) \quad (4)$$

hold as the impersonation is valid. We note that the second verification equation implies that $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ is a group signature of a message $m$ under one-time public key $\sigma_3 = \sigma_1^{\hat{y}}$ and $\sigma_4 = H_1(m)^{\hat{y}}$ which is generated by zero-knowledge proof to keep $\hat{y}$ hidden. Since we can extract $\hat{y}$ by invoking $\mathcal{B}$ twice according to Forking Lemma in [28], we obtain

$$\begin{aligned} e(\sigma_2, g_2)e(\sigma_1, h_2)e(\sigma_1^{\hat{y}}, U_2) &= A \\ e(\sigma_2, g_2) &= Ae(\sigma_1^{-1}, h_2)e\left(\sigma_1^{-\hat{y}}, U_2\right). \end{aligned} \quad (5)$$

We know $g_1, h_1 \in \mathbb{G}_1$ and recall $h_1 = g_1^{\beta}$. Thus, there exists $\beta, \hat{k} \in \mathbb{Z}_p^*$ such that $\sigma_1 = g_1^{\hat{k}}$ and $h_1 = g_1^{\beta}$, where we do not know $\beta, \hat{k}$ as $h_1$ produces from the challenge. We also recall the isomorphism $\psi(h_2) = h_1, \psi(g_2) = g_1, \psi(U_2) = U_1$. Then, we obtain $h_2 = g_2^{\beta}$ and $U_2 = g_2^u$. It follows that

$$\begin{aligned} e(\sigma_2, g_2) &= Ae(\sigma_1^{-1}, h_2)e\left(\sigma_1^{-\hat{y}}, U_2\right) \\ &= e(Z, g_2)e\left(g_1^{-\hat{k}}, g_2^{\beta}\right)e\left(\left(g_1^{\hat{k}}\right)^{-\hat{y}}, g_2^u\right) \\ &= e(Z, g_2)e\left(\left(g_1^{\beta}\right)^{-\hat{k}}, g_2\right)e\left(\left(g_1^{u\hat{y}}\right)^{-\hat{k}}, g_2\right) \\ &= e(Z, g_2)e\left(h_1^{-\hat{k}}, g_2\right)e\left(\sigma_1^{-u\hat{y}}, g_2\right) \\ &= e\left(Zh_1^{-\hat{k}}\sigma_1^{-u\hat{y}}, g_2\right). \end{aligned} \quad (6)$$

Thus, we have $\sigma_1 = g_1^{\hat{k}}$, $\sigma_2 = Zh_1^{-\hat{k}}\sigma_1^{-u\hat{y}}$. Since we can extract $\hat{y}$, we have $\hat{\sigma}_2 = \sigma_2\sigma_1^{u\hat{y}} = Zh_1^{-\hat{k}}$ where $\hat{k}$ is an unknown value to us. We note that $(\sigma_1, \hat{\sigma}_2)$ is an ElGamal ciphertext of $Z$ encrypted under the public key $h_1$. Hence, we have a Diffie-Hellman tuple

$$(g_1, h_1, \sigma_1, \hat{\sigma}_2) = \left(g_1, h_1, g_1^{\hat{k}}, Z h_1^{-\hat{k}}\right)$$
$$(g_1, h_1, \sigma_1, {}^{Z}/_{\hat{\sigma}_2} = (g_1, h_1, g_1^{\hat{k}}, h_1^{\hat{k}}) \quad (7)$$

We obtain an accompanied Diffie-Hellman tuple $(g_2, h_2, \sigma_1, {}^{Z}/_{\hat{\sigma}_2})$ where we can use to answer the challenge and subsequently contradicts the DHK assumption in $\mathbb{G}_1$. This contradiction completes the proof. ■

2. Robustness against Sybil Attacks

**Claim 2:** *An internal adversary is not able to produce multiple signatures to sign a false accusation report for personal benefits.*

To commit this attack, an internal adversary produces at least two signatures on the same revocation report. When an internal adversary signs the same report more than once, the trusted party can trace the identity of the adversary and reject the report. Thus, sybil attack can be prevented in our protocol.

**Proof.** We consider an internal adversary $\mathcal{C}$ generates two signatures to endorse a false revocation report. Assuming the trusted party performs verification procedure similar to message authentication when receiving the report, the attempt of sybil attack can be identified when the two signatures share

$$\sigma_4 = H_1(m)^y. \quad (8)$$

We recall $\sigma_3 = \sigma_1^y$ and $\sigma_4 = H_1(m)^y$ is a one-time public key of the group signature, $\sigma$ which specifies that the knowledge of $y$ is hidden in $(\sigma_3, \sigma_4)$. We can say $y$ is the secret key of some group member since we have proved that impersonation is not valid in Claim 1. Thus, the trusted party is able to use the tracing information $T = g_2^y$ to trace the group member by checking

$$e(\sigma_3, g_2) = e(\sigma_1, T). \quad (9)$$

If credentials update is performed, the trusted party will use $\dot{T} = \left(g_2^y\right)^x$ and check

$$e(\left(U_1^y\right)^x, g_2) = e(xU_1, x\dot{T}). \quad (10)$$

This enables the trusted party to trace $\mathcal{C}$ when the same component of $\sigma_4$ is identified upon receiving the report. The trusted party will then discard the report and put $\mathcal{C}$ in the revocation list so that $\mathcal{C}$ will no longer be able to participate in the network. ■

3. Robustness against Man-in-the-middle Attacks

**Claim 3:** *An internal adversary is not able to launch man-in-the-middle attack on the revocation report delivered from other vehicle to the trusted party.*

To launch this attack, an internal adversary generates a valid report replacing the actual report generated by other vehicle. Since the protocol requires verification procedure between two communicating parties, it is not possible for the man-in-the-middle to pass the verification and further intercept with the communication.

**Proof.** We consider an internal adversary $\mathcal{D}$ performs man-in-the-middle attack. $\mathcal{D}$ captures the revocation report generated by other vehicle. We assume $\mathcal{D}$ is able to modify the report and the trusted party performs verification procedure similar to message authentication when receiving the report. Since it is impossible to forge a report according to Claim 1, the modified report cannot satisfy the two verification equations

$$e(\sigma_2, g_2) = Ae(\sigma_1, h_2)e(\sigma_3, g_2) \tag{11}$$

$$\sigma_5 = H(m||\sigma_1||\sigma_2||\sigma_3||\sigma_4||H_1(m)^{\sigma_6}\sigma_4^{\sigma_5}\ ||\sigma_1^{\sigma_6}\sigma_3^{\sigma_5}) \tag{12}$$

when the trusted party received the report. The modified revocation report will be rejected and thus, the protocol is secured against man-in-the-middle attack. ■

### 5.2.2 Robustness against Revoked Adversaries

1. Robustness against Impersonation Attacks

**Claim 4:** *Our protocol is robust against a revoked adversary conducting impersonation attack that aims to falsify revocation reports*.

The revoked adversary who has been found guilty in the past attempts to reenter the network. However, being a revoked user would not pass the verification procedure due to our proposed revocation check process. Thus, the revoked adversary is not able to impersonate other vehicle's identity to make a false revocation report.

**Proof.** We consider a revoked adversary $\mathcal{E}$ with its public key, $Y_i$ performs the impersonation attack. We recall the proof in Claim 1, where we assume revocation check procedure is not able to locate the adversary since we want to show the contradiction of DHK challenge. In contrast, here we assume revocation check procedure is able to locate $\mathcal{E}$ (since $\mathcal{E}$ is a revoked adversary) by checking

$$\sigma_2 = k_2(h_1 Y_i)^s \tag{13}$$

for each $Y_i$ in the revocation list. Once identifying $Y_i$, the revocation report will not be accepted and thus $\mathcal{E}$ is discarded from the network. This shows our protocol is robust against impersonation attack. ■

2. Robustness against Sybil Attacks

**Claim 5:** *Our protocol is robust against a revoked adversary attempting sybil attack to lodge a false revocation report*.

The revoked adversary who is able to lodge a revocation report using its non-updated key attempts to produce multiple signatures on the same report in order to make it look reliable when the report is generated by multiple vehicles. However, the trusted party can identify this activity since the generation of two or more signatures (either by using updated keys or revoked keys) on the same report produce similar component of $\sigma_4$. Since the repetition of $\sigma_4$ is present, the proof runs similarly to the proof in Claim 2.

3. Robustness against Man-in-the-Middle Attacks

**Claim 6:** *Our protocol is robust against a revoked adversary performing man-in-the-middle attack in order to alter revocation reports.*

To execute this attack, the revoked adversary intercepts a revocation report from other vehicle and modify the content before relaying it to the trusted party. Since revoked certificate is no longer be able to use its revoked key in the network, this attack is preventable.

**Proof.** We consider an adversary $\mathcal{F}$ whose certificate has been revoked performs man-in-the-middle attack. $\mathcal{F}$ intercepts the communication between other vehicle and a trusted party by

capturing the report from the vehicle, inject a new report using its own credentials and sends it to the trusted party. Upon receiving the report, we assume the trusted party performs revocation check during verification procedure. Since $\mathcal{F}$ is a revoked user, there are two possible situations. First, if $\mathcal{F}$ was revoked by VLR, the trusted party will find $\mathcal{F}$'s certificate in the revocation list . If $\mathcal{F}$ was revoked by credential update procedure, $\mathcal{F}$ will not be verified correctly under $\mathcal{RM}$'s public key $A = e(Z, g_2)$ which has been updated to $\dot{A} = e(\dot{Z}, g_2)$. Thus, $\mathcal{F}$ is unable to proceed with the attack. Consequently, our protocol provides strong robustness against man-in-the-middle attacks. ■

## 6. Performance Analysis

This section presents the performance of our proposed revocation protocol. We compare the computational cost and computation time of our protocol with other revocation protocols designed for VANETs. We then evaluate our protocol by conducting a simulation.

### 6.1 Computational Cost and Computation Time

We compare the performance efficiency of our work with revocation protocols adopted in GSIS [16], and TAA [12] schemes. We do not compare our work with revocation protocols in [7,19-22] since the schemes rely on RSU during message broadcast between vehicles. Meanwhile, revocation protocols in [17-18] only adopted VLR mechanism which is known to be inefficient if a large number of revoked vehicles exists in the revocation list. Thus, only GSIS and TAA are suitable schemes for a comparison. Moreover, we do not compare our work with pairing-based work as our construction is based on pairing in group signature.

We only evaluate the performance of verification phase because this is the phase where `revocation check' and `signature check' are being conducted. **Table 3** summarizes the comparison of the performance efficiency for $t = 1$ as GSIS does not support a threshold method. In this table, $r.\mathbb{G}_1$ indicates $r$ scalar multiplications in $\mathbb{G}_1$, $s.P$ indicates $s$ pairing operations and $n$ in the fifth column denotes the size of the revocation list. Specifically, we conduct our comparison in two categories: computational cost and computation time.

**Table 3.** Comparison of Performance Analysis

| | Computational Cost | | Computation Time | |
|---|---|---|---|---|
| | **Revocation Check** | **Signature Check** | **Revocation Check (ms)** | **Signature Check (ms)** |
| GSIS [16] | $2.P$ | $5.\mathbb{G}_1 + 1.P$ | $9.0 \times n$ | 7.5 |
| TAA [12] | $1.\mathbb{G}_1$ | $7.\mathbb{G}_1 + 5.P$ | $0.6 \times n$ | 26.7 |
| Our work | $1.P$ | $6.\mathbb{G}_1 + 1.P$ | $4.5 \times n$ | 8.1 |

- **Computational cost.** We consider the two most expensive operations, particularly scalar multiplication and pairing evaluation. If exponentiation is used, it will be changed into scalar multiplication to ease the comparison. According to [12], in usual implementation, one exponentiation in $\mathbb{G}_T$ ($\mathbb{G}_3$ in MLGS) costs about 4 scalar multiplication in $\mathbb{G}_1$. We use this trick to transform our observation of operation used in [11,12,16] into the operation presented in computational cost column of **Table 3**. In addition, a multi-base pairing is similar to a single-base pairing as they almost have the same overhead [29]. Now, we add both `revocation check' and `signature check' operations to get a full

operation of verification phase. We then have $5.\mathbb{G}_1 + 3.P$ for GSIS, $8.\mathbb{G}_1 + 5.P$ for TAA, and $6.\mathbb{G}_1 + 2.P$ for the improved MLGS. Since the cost of pairing evaluation is more expensive compared to scalar multiplication [30], we see that the computational cost for our work is better than GSIS and more costly efficient compared to TAA as our work has the least pairing operation compared to others. Our work requires two pairings, while GSIS and TAA require three and five pairings respectively.

- **Computation time.** According to the experiment in [31], to calculate the time taken to perform a pairing operation and a scalar multiplication, we observe the processing time for an MNT elliptic curve running on an Intel Pentium IV 3.0 GHZ machine. We set the curve with embedding degree $k = 6$ and $q = 160$ bits to achieve security level of 80 bits. Then, we obtain the following results: one pairing evaluation and one scalar multiplication in $\mathbb{G}_1$ can be done within 4.5 ms and 0.6 ms respectively. Using this information, we calculate the computation time of operations tabulated in the computational cost column of **Table 3**. For instance, we take `revocation check' operation in our work, i.e $1.P$, then we multiply it by 4.5 ms $\times$ $n$, where $n$ is the length of the revocation list to obtain 4.5 ms $\times$ $n$. Similarly for the `signature check', i.e. $6.\mathbb{G}_1$ and $1.P$, we multiply each of them with 0.6 ms and 4.5 ms respectively before adding them up together to obtain 8.1 ms as a total. We present the rest of the calculation result in Computation Time column of **Table 3**. We observe that our revocation check is twice as fast as GSIS but it is slower than TAA. However, TAA takes three times significantly longer to perform signature check. Even though GSIS outperforms in term of signature check, its revocation check is the longest compare with other schemes. We note that the difference of signature check between GSIS and our work differ by less than 1 ms.

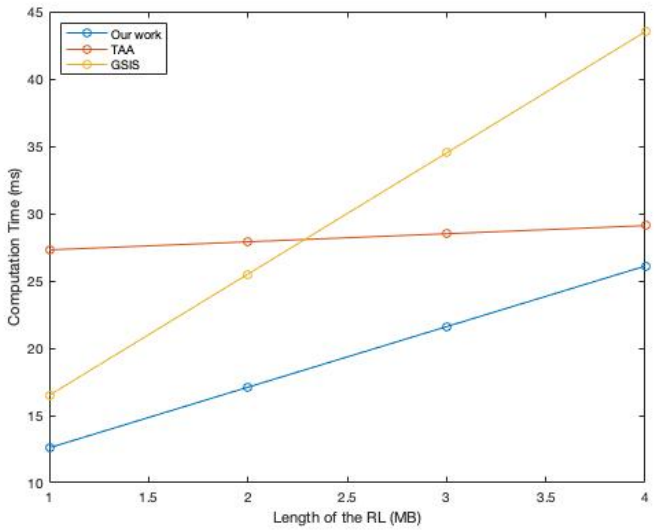


**Fig. 5.** Increase of computation time when $RL < 5$ MB

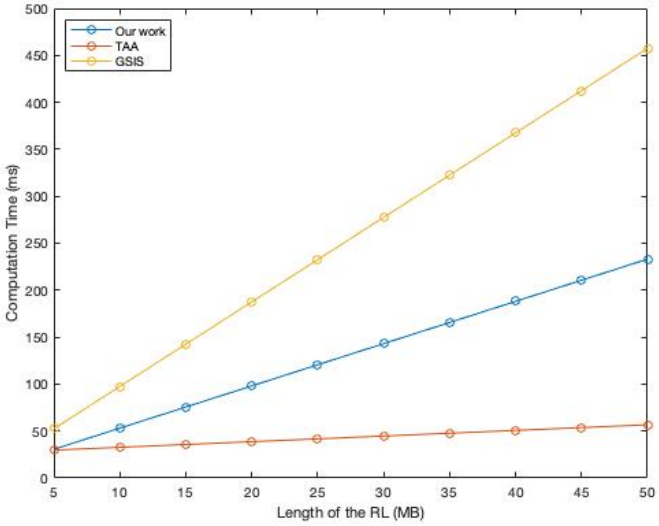


**Fig. 6.** Increase of computation time when $RL \geq 5$ MB

We depict **Fig. 5** and **Fig. 6** for a clear comparison of the computation time by calculating the value of $n$ (the length of the revocation list). **Fig. 5** shows that our work requires the least computation time when $n < 5$ MB. Meanwhile, when $n \geq 5$ MB illustrated in **Fig. 6**, our work is more efficient than GSIS but slower than TAA. However, we note that our work proposes credentials update procedure when the size of the revocation list is larger than a certain threshold. Thus, when the threshold is set at 5 MB, **Fig. 6** will not be applicable as revocation will be carried out by credentials update procedure.

From the above analysis, our work shows better performance compared to GSIS in both computational cost and computation time and achieves comparable performance to TAA. TAA competes in terms of computation time when the size of revocation list is 5 MB and above, but this is not applicable when credentials update takes it roll. In addition, TAA's computation cost is the most expensive compared to others. In this comparison, we note that our work shows better performance which requires the least cost and operates at comparable pace.

### 6.2 Simulation

We evaluate the performance of our revocation protocol by using the MATLAB platform. This MATLAB platform has also been used in other previous works [32-36]. We use the freeway mobility model that is of size 10000 $m^2$ with 4 lanes, 3 entry ramps, 3 exit ramps, and vehicles move in one direction. The simulation road scenario is shown in **Fig. 7**. In this model, we allow a vehicle to change lanes and overtake the leading vehicle only if there is no vehicle within the vehicle's safety distance, $d_s$ of 10 m. We note that this freeway mobility setting which represents a common scenario encountered in Europe and North America has also been used in other work such as in [37-39]. We selected 500 vehicles randomly distributed on the freeway lanes with each vehicle has a transmission range of 300m. This is a common range used in [9,16,18,19] and it follows the IEEE 802.11p standard, that is, the transmission range can be up to 1000m [40]. The adopted simulation

parameters are listed in **Table 4**. The simulations run for 5 trials for 10 minutes duration each. We simulate our protocol from the following aspects:

- Revocation delay versus the number of revocation reports: The time taken to perform cryptographic operations to verify revocation reports.
- Revocation delay versus the number of revoked vehicles: The increment of revocation delay due to the increasing number of revoked vehicles in the certificate revocation lists.

**Table 4.** Simulation Parameters

| Parameters | Value |
|---|---|
| Mobility model | Freeway |
| Simulation region | 10000 $m^2$ |
| Number of lanes | 4 lanes |
| Max. vehicles speed | 90 km/h |
| Min. vehicles speed | 70 km/h |
| Safety distance | 10 m |
| Transmission range | 300 m |
| Road traffic density | 500 |

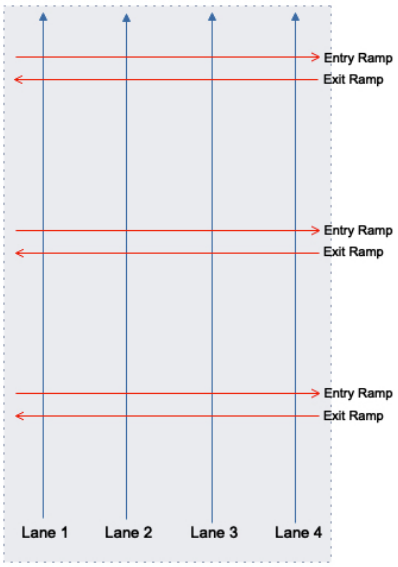


**Fig. 7.** Road scenario for simulation

Revocation delay is an important issue in revocation protocols in VANETs as it evaluates the efficiency of a protocol to revoke the misbehaved vehicles from the network. The longer the delay times, the lower the number of misbehaved vehicles can be removed from the network. We classify the issue of revocation delay into two categories: 1) the time taken to perform the cryptographic operations in order to verify the validity of a revocation report, and 2) the time taken to perform the cryptographic operations when the number of revocation reports increases. We note that the cryptographic operations are the scalar multiplication and pairing operation as they are the most time-consuming operations for a revocation protocol [41]. The first issue has been addressed in Section 6.1 where we have shown that our protocol achieves

better cryptographic performance than the other two schemes [12,16]. Meanwhile, the second issue which focuses on scalability matter will be evaluated as follows.

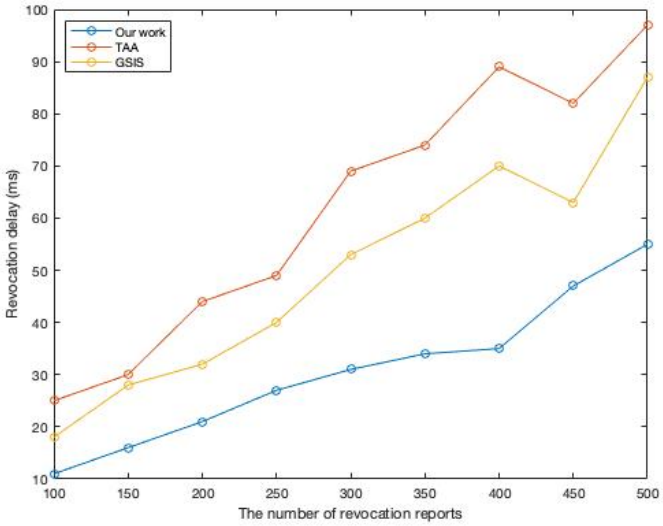


**Fig. 8.** Revocation delay versus the number of revocation reports

We recall in **Table 3** that our work requires $6.\mathbb{G}_1 + 2.P$, GSIS [16] requires $5.\mathbb{G}_1 + 3.P$, and TAA [12] requires $8.\mathbb{G}_1 + 5.P$ to verify a revocation report. We now use this information to evaluate the revocation delay of our work when the increasing number of revocation reports is considered and we compare the result with TAA and GSIS schemes to assess overall performance. We also assume that each vehicle may broadcast one revocation report and thus 500 vehicles may broadcast a total of 500 reports in this experiment. **Fig. 8** shows the simulation result of the revocation delay with respect to the number of revocation reports. The results of the experiments show that the revocation delay increases when the number of revocation reports increases. This is natural since if there are more revocation reports to be verified, there will be more cryptographic operations to be performed and thus resulting in longer revocation delay. We observe our work has the lowest revocation delay, followed by GSIS and TAA schemes. This is due to the fact that our proposed protocol has the least pairing operation. Our protocol uses $2.P$, GSIS uses $3.P$ and TAA uses $5.P$. As discussed in Section 6.1, pairing operation takes longer computation, that is, 4.5 ms compared to scalar multiplication which only takes 0.6 ms for computation. This proves that our proposed protocol significantly outperforms the other schemes in terms of revocation delay when the number of revocation reports increases in the network.

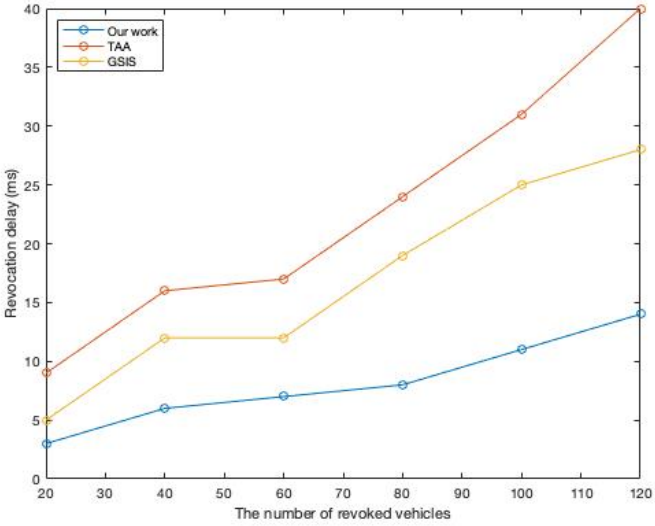


**Fig. 9.** Revocation delay versus the number of revoked vehicles

**Fig. 9** shows the simulation results of the revocation delay with respect to the number of revoked vehicles in the revocation lists. In this experiment, we set the threshold 0.25 for the number of revoked vehicles in the revocation list as in [42] to evaluate the effectiveness of the protocol. In other words, the number of revoked vehicles must be within one fourth from the selected 500 vehicles. If the number of revoked vehicles appears to be above the said threshold, credential updates operation takes over the situation. The results of the experiment show that the revocation delay increases when the number of revoked vehicles increases. This is reasonable, as when the size of the revocation list grows longer due to an increase of the number of revoked vehicles in the network, the time taken to look up for the revoked vehicles in the list also becomes longer. From **Fig. 9**, it is clear that our work has the least revocation delay with respect to the number of revoked vehicles, followed by GSIS and TAA schemes. The reason for the least delay in our work is because of the efficiency of the proposed cryptographic operations. The same operation is used for the revocation look up operation, that is, $2.P$ in our work, $3.P$ in GSIS [16] and $5.P$ in TAA [12]. Hence, our protocol is faster and outperforms [12,16].

The simulation results demonstrate the practicability and efficiency of our proposed revocation protocol when evaluated using real VANET environment. Our protocol requires the least pairing operation which performs significantly faster than the existing schemes of [12,16] and this allows our protocol to scale well with the increase in the number of revocation reports and revoked vehicles in the network. Hence, our protocol provides better efficiency and scalability.

## 7. Conclusion

In this paper, we have presented a secure and efficient revocation protocol for group signature schemes in VANETs. Our protocol adopted VLR technique and credentials update procedure to remove misbehaved vehicles from the network. We have shown that our revocation protocol can solve the revocation issue for group signature schemes that proposes

vehicles to generate its own tracing information in the system. Meanwhile, our generic abstraction may assist to provide guidelines to design future revocation protocol based on group signatures. As far as we are aware of, this is the first generic abstraction for revocation in group signatures exists in the literature. Our proposed revocation protocol not only provides the desired level of security but also performs better than the other group signature-based schemes of GSIS [16] and TAA [12].

For future work, it might be of interest to explore other cryptographic techniques that can vastly improve the performance efficiency for a secure revocation protocol. Moreover, it would be interesting to design revocation protocol based on other cryptographic primitives.

## References

[1] C2CC. The car-to-car communication consortium, 2011. Available online: http://www.car-to-car.org (accessed on 20 June 2018).
[2] NoW. Network on wheels, 2004. Available online: http://www.network-on-wheels.de (accessed on 20 June 2018).
[3] R. Kroh, A. Kung, and F. Kargl, "VANETs security requirements final version, Technical report," *Secure Vehicle Communication (SeVeCom)*, 2006.
[4] H. Artail and N. Abbani, "A pseudonym management system to achieve anonymity in vehicular ad hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 106-119, 2016. Article (CrossRef Link).
[5] D. He, S. Zeadally, B. Xu and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681-2691, 2015. Article (CrossRef Link).
[6] A. Malip, S. L. Ng, Q. Li, "A certificateless anonymous authenticated announcement scheme in vehicular ad hoc networks," *Security and Communication Networks*, vol. *7*, no. 3, pp. 588–601, 2014. Article (CrossRef Link).
[7] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1711–1720, 2016. Article (CrossRef Link).
[8] M. Raya and J. P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007. Article (CrossRef Link).
[9] Q. Li, A. Malip, K. M. Martin, S. L. Ng, and J. Zhang, "A reputation-based announcement scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 4095–4108, 2012. Article (CrossRef Link).
[10] S. Díaz-Santiago and D. Chakraborty, "Encryption schemes secure against profiling adversaries," in *Proc. of E-Business and Telecommunications, ICETE 2012, M.S. Obaidat and J. Filipe, Eds.; Springer: Berlin, Heidelberg*, pp. 172–191, 2014. Article (CrossRef Link).
[11] Q. Wu, J. Domingo-Ferrer, and Ú. González-Nicolás, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 559–573, 2010. Article (CrossRef Link).
[12] L. Chen, S. L. Ng, and G. Wang, "Threshold anonymous announcement in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605–615, 2011. Article (CrossRef Link).
[13] P. Golle, D. H. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. of the First International Workshop on Vehicular Ad Hoc Networks*, pp. 29–37, 2004. Article (CrossRef Link).
[14] P. Papadimitratos, L. Buttyán, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J. P. Hubaux, "Secure vehicular communication systems: design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100-109, 2008. Article (CrossRef Link).

[15] B. Liu, J. T. Chiang, and Y. C. Hu, "Limits on revocation in VANETs," in *Proc. of the 8th International Conference on Applied Cryptography and Network Security (ACNS 2010)*, pp. 38–52, 2010.
[16] X. Lin, X. Sun, P. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007. Article (CrossRef Link).
[17] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proc. of the Fourth International Workshop on Vehicular Ad Hoc Networks*, pp. 19–28, 2007. Article (CrossRef Link).
[18] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing together efficient authentication, revocation, and privacy in VANETs," in *Proc. of the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 1–9, 2009. Article (CrossRef Link).
[19] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 907–919, 2014. Article (CrossRef Link).
[20] X. Zhu, S. Jiang, L. Wang, H. Li, W. Zhang, and Z. Li, "Privacy-preserving authentication based on group signature for VANETs," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pp. 4609–4614, 2013. Article (CrossRef Link).
[21] Y. Hao, Y. Cheng, C. Zhou, and W. Song, "A distributed key management framework with cooperative message authentication in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 616–629, 2011. Article (CrossRef Link).
[22] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A scalable robust authentication protocol for secure vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1606–1617, 2010. Article (CrossRef Link).
[23] J. Choi and S. Jung, "A security framework with strong non-repudiation and privacy in VANETs," in *Proc. of 6th IEEE Consumer Communications and Networking Conference*, pp. 1–5, 2009. Article (CrossRef Link).
[24] A. Wasef, Y. Jiang, and X. Shen, "ECMV: efficient certificate management scheme for vehicular networks," in *Proc. of the Global Communications Conference*, pp. 639–643, 2008. Article (CrossRef Link).
[25] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation" in *Proc. of the 11th ACM Conference on Computer and Communications Security*, pp. 168–177, 2004. Article (CrossRef Link).
[26] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: keyed-hashing for message authentication," *RFC*, vol. 2104, pp. 1–11, 1997. Article (CrossRef Link).
[27] J. Bringer and A. Patey, "Backward unlinkability for a VLR group signature scheme with efficient revocation check," *IACR Cryptology ePrint Archive*, pp. 376, 2011.
[28] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptology*, vol. 13, no. 3, pp. 361–396, 2000. Article (CrossRef Link).
[29] X. Boyen and B. Waters, "Compact group signatures without random oracles," in *Proc. of Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 427–444, 2006. Article (CrossRef Link).
[30] N. B. Gayathri, G. Thumbur, P. V. Reddy, and Z. U. R. Muhammad, "Efficient pairing-free certificateless authentication scheme with batch verification for vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 31808–31819, 2018. Article (CrossRef Link).
[31] M. Scott, "Efficient implementation of cryptographic pairings," Available online: http://ecrypt-ss07.rhul.ac.uk/Slides/Thursday/ mscott-samos07.pdf (accessed on 10 April 2019).
[32] A. Wasef and X. S. Shen, "REP: location privacy for VANETs using random encryption periods," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 172-185, 2010. Article (CrossRef Link).

[33] A. Zhou, J. Li, Q. Sun, C. Fan, T. Lei, and F. Yang, "A security authentication method based on trust evaluation in VANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, pp. 59, 2015. Article (CrossRef Link).
[34] H. Dok, H. Fu, R. Echevarria, and H. Weerasinghe, "Privacy issues of vehicular ad-hoc networks," *International Journal of Future Generation Communication and Networking*, vol. 3, no. 1, pp. 17-32, 2010.
[35] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.P. Hubaux, "Mix-zones for location privacy in vehicular networks," in *Proc. of ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, 2007.
[36] C. H. Kim and I. H. Bae, "A misbehavior-based reputation management system for VANETs," *Embedded and Multimedia Computing Technology and Service, Springer, Dordrecht*, pp. 441-450, 2012. Article (CrossRef Link).
[37] P. Gokulakrishnan and P. Ganeshkumar, "Road accident prevention with instant emergency warning message dissemination in vehicular ad-hoc network," *PloS one*, vol. 10, no. 12, pp. e0143383, 2015. Article (CrossRef Link).
[38] H. S. Dawood and Y. Wang, "An efficient emergency message broadcasting scheme in vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, pp. 232916, 2013. Article (CrossRef Link).
[39] P. Tyagi and D. Dembla, "Performance analysis and implementation of proposed mechanism for detection and prevention of security attacks in routing protocols of vehicular ad-hoc network (VANET)," *Egyptian informatics journal*, vol. 18, *no.* 2, pp. 133-139, 2017. Article (CrossRef Link).
[40] B. S. Gukhool and S. Cherkaoui, "IEEE 802.11p modelling in NS 2," in *Proc. of 33nd IEEE Conference on Local Computer Networks (LCN 2008) Montreal*, 2008. Article (CrossRef Link).
[41] A. Wasef and X. Shen, "EDR: efficient decentralized revocation protocol for vehicular ad hoc networks," *IEEE Trans. Vehicular Technology*, vol. 58, no. 9, pp. 5214-5224, 2009. Article (CrossRef Link).
[42] G. Arboit, C. Crépeau, C. R. Davis, and M. Maheswaran, "A localized certificate revocation scheme for mobile ad hoc networks," *Journal of Ad Hoc Networks*, vol. 6, no. 1, pp. 17-31, 2008. Article (CrossRef Link).

**Nur Fadhilah Mohd Shari** received the B.Sc. (first class) degree from the Purdue University, U.S.A in 2015. She is currently pursuing the M.Sc. degree in Mathematics from University of Malaya, Malaysia. Her current research interests are cryptographic protocols, wireless and network security.

**Amizah Malip** received the M.Sc. degree in mathematics of cryptography and communications and Ph.D. degree in information security, both from the Royal Holloway, University of London, Surrey, UK. She is currently a Senior Lecturer with the Institute of Mathematical Sciences in the University of Malaya, Malaysia. Her main research interests include privacy, network security and cryptographic applications.

**Wan Ainun Mior Othman** received the Ph.D. degree in mathematics from the University of Science, Malaysia. She is currently an Associate Professor with the Institute of Mathematical Sciences in University of Malaya, Malaysia. Her research interests include algorithms and cryptology.