Noname manuscript No.
(will be inserted by the editor)

Efficient Group Signatures for Privacy-Preserving Vehicular Networks

Lukas Malina · Arnau Vives-Guasch · Jordi Castellà-Roca · Alexandre Viejo · Jan Hajny

Received: date / Accepted: date

Abstract In this paper, we deal with efficient group signatures employed in secure and privacy-preserving vehicular networks. Our solution aims to minimize the impact of several common attacks like denial of services or replay attacks on the efficiency of privacy-preserving security solutions in vehicular networks. Due to advanced properties like a short-term linkability and a categorized batch verification, our solution based on group signatures ensures privacy, security and the efficiency of vehicular networks which can be attacked by malicious parties. We outline the proposed communication pattern of vehicular networks, our security solution in detail, a formal security analysis and the experimental implementation of our solution. In addition, we evaluate and compare our solution with related works. Our group signature scheme is more efficient and secure in the signing phase and in the verification phase than related schemes.

Keywords Authenticity · Cryptography · Group Signatures · Privacy · Security · Vehicular Networks

1 Introduction

Vehicular ad hoc Networks (VANET) or vehicular networks can be useful in many ways, from increasing a driver safety to reducing traffic congestions. VANET applications can work in short distances, e.g. monitoring collision warnings, change lanes, break alerts and so on. The data processing and communication of these applications should be as fast as possible for the safe and on-time responses of drivers. The sending period of beacon messages should last less than 300 ms [11]. These messages are sent via the Vehicle to Vehicle communication (V2V). On the other hand, there are applications which work in wide areas to distribute useful VANET messages, e.g., accident warnings, traffic jam warnings or weather monitoring. These messages can be sent via V2V or via the communication model called Vehicle to Infrastructure (V2I) and then these messages are broadcasted to other users in a specific area (V2I-I2V). Considering the communication latency due to longer distances, we assume that the sending period of these messages should last seconds. In addition to offering efficient communication and data processing, it is also important to provide security, given the potential abuses and attacks.

1.1 Vehicular Network Security

Vehicular network security plays a key role in situations such as the generation of bogus and/or malicious messages, misusing at roads, eavesdropping etc. Common solutions, e.g., [10], [19] guarantee the message integrity, authentication and non-repudiation. Furthermore, privacy is required due to the possibility of drivers being tracked by malicious observers. VANETs can serve in a urban traffic where hundreds of vehicles communicate following the V2V or V2I paradigms, so that the security overhead and computation time are minimal. There are a lot of solutions in VANETs that are

L. Malina · J.Hajny
Dept. of Telecommunications, Brno University of Technology,
Technicka 12, 616 00 Brno, Czech Republic
Tel.: +42054114 6963
E-mail: {malina, hajny}@feec.vutbr.cz

Arnau Vives-Guasch · Jordi Castellà-Roca · Alexandre Viejo
Department of Computer Engineering and Mathematics,
Universitat Rovira i Virgili,
Av. Paisos Catalans 26, Tarragona, Catalonia (Spain)
E-mail: {arnau.vives, jordi.castella, alexandre.viejo}@urv.cat

secure and keep users' privacy. Nevertheless, privacy-preserving solutions can be vulnerable against several denial of service attacks. The following scenario demonstrates the current security problems which affect the solutions that provide user privacy in VANETs.

Scenario 1: A driver, Alice (A), with the car no. 2, which is depicted in Fig.1, records special events (accidents, traffic jams, roads under construction etc.). Depending on the type of event, A immediately broadcasts a warning message through the wireless V2V communication to all the cars which form the VANET. In this scenario, an accident is depicted in Fig.1. Let us assume that another driver, Bob (B), with car no. n-1, who is in range and coming closer to A, receives this message. B also receives more messages from other cars in the area. Moreover, other messages can contain contradictory warnings or malicious/bogus information. In a short time, B must consider the validity of these messages and quickly decide changing the route (from planed I. to II.). If B makes the right decision, he can avoid the situation referenced by the first warning message. It is obvious that the decision must come in real time and as soon as possible. Nevertheless, the received messages are from anonymous nodes so B may wonder which messages are coming from honest sources and which are not. Our solution to this problem is based on the employment of a group signature scheme which adds new properties, namely, the short-term linkability and the categorized batch verification. Due to these properties, A can sort out known honest and malicious messages and perform a verification process faster.

The paper is organized as follows: The next section presents the related work, which is focused on the security and privacy protection in VANETs, and in which we outline our contribution. Section 3 presents preliminaries that describe parties in our solution, a communication pattern, requirements and main cryptographic techniques used in our proposal. Further, section 4 introduces our solution and the phases of our scheme are described. Section 5 contains the security analysis of our solution. Section 6 describes our experimental implementation, and the important phases like signing and verification are evaluated and compared with related solutions in section 7. Finally, a conclusion is presented.

2 Related Work and Our Contribution

This section outlines the related work and our contribution.

2.1 Related Work

Privacy in VANETs can be achieved in many ways. For example, in [4], the authors deal with privacy and security in VANETs with a safe distance-based location privacy scheme called SafeAnon. The scheme uses a safe distance measurement technique to determine the maximum obfuscation radius for preserving location privacy while maintaining traffic safety. The SafeAnon scheme fights against a Global Passive Adversary (GPA) that can locate and track any vehicle in an area of interest by eavesdropping on broadcast messages. Nevertheless, this protection can be only employed in several VANET applications based on a short distance among vehicles, e.g., collision detection. Our solution also aims at VANET applications used in medium and long distances, e.g., the detection of traffic jams, accidents and so on. On the other hand, Horng et al. [9] propose a private V2V communication mode that can be used in wide areas. Nevertheless, the main drawback of the proposed private V2V scheme is the restriction of privacy which can be kept only in the specific group of users, where users know the public keys of other participants and can build their profiles. The second disadvantage is a presence of a session key establishment subphase which can slow the communication process.

Using pseudonyms in VANETs is proposed in [8] and [7]. Raya and Hubaux [18] use anonymous certificates which are stored in vehicles (usually in a tamper-proof device). This approach uses a set of short-lived pseudonyms, and privacy among vehicles is provided by changing these certified public keys. Nevertheless, in large urban VANETs, this approach is burdened by preloading and storing a large number of anonymous certificates with pseudonyms.

Group signatures (GS) in VANETs provide user anonymity by signing a message on behalf of a group. GS guarantee the unlinkability of honest users and the traceability of misbehaving users. The scheme [13], called GSIS, uses the combination of a group signature based on [3] with a hybrid membership revocation mechanism in the V2V communication, and Identity Based Group Signature (IBGS) in the V2I communication. The hybrid membership revocation with the list of revoked members (RL) works with a threshold value T_τ. In case $|\text{RL}| < T_\tau$, the scheme uses a revocation verification algorithm. Otherwise, the scheme updates the public/private group keys of all non-revoked members. For efficient verification, the authors of [26] propose a GS with batch verification in V2I, which takes three pairing operations. This scheme, called IBV, has several drawbacks such as using tamper proof devices, being thus vulnerable to tracking or impersonation attacks.

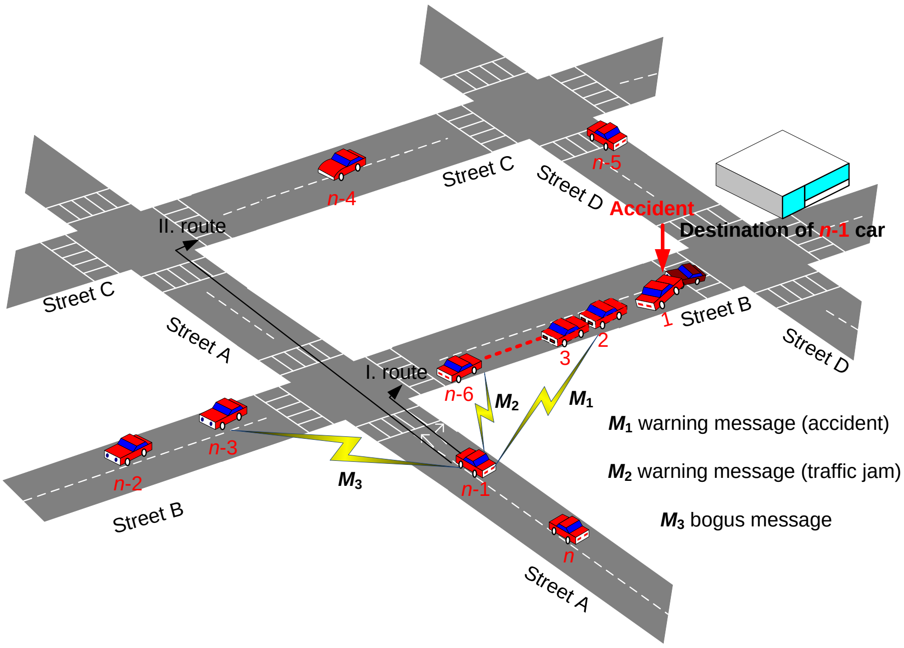

Fig. 1: The VANETs in urban traffic - Scenario 1.

See [5] for a complete description. Schemes proposed in [27] and [23] can efficiently verify a large number of messages in V2V. These schemes use short group signatures with fast batch verification (only two pairing operations are used instead of 5 n, where n is the number of messages). Nevertheless, the performance of batch verification degrades in dense V2V communication with bogus messages. The On Board Units (OBUs) must process the messages quickly (they have between 100 ms and 300 ms to process a message [11]). Thus, the computation of expensive pairing and exponentiation on limited On Board Units (OBUs) is a hard requirement to meet because of the short response time. This fact limits the VANETs in practice. Qin et al. [17] employs identity based group signature with the batch verification, provides a scalable management of large VANETs and an efficient revocation of members, but suffers from more expensive signing and verification phases than GS.

2.2 Our Contribution

Related works focus on providing security and privacy protection and they also try to offer efficient signing and verification processes. In addition to those features, we also aim at the protection against denial of services attacks, which is not usually covered in the literature.

– In the V2V communication, our solution provides efficient signing with short-term linkability. Our proposal uses the modified scheme of Wei et al. (WLZ scheme) [24]. Nevertheless, our solution adds short-term linkability obtaining a more efficient signing phase than in the WLZ scheme. Moreover, the WLZ scheme is focused on the V2V communication and does not describe the registration and join phases in detail. The short-term linkability is demanded for several applications [20] and can protect against Sybil and Denial of Service attacks. Due to this, our solution can provide efficient categorized batch verification with this short-term linkability. Generally, in group signatures, the batch verification of n messages is more efficient than individual verification, but the complexity of batch computation with bogus messages increases from O(1) to O(ln n). In [6], the authors claim that if $\geq 15\%$ of the signatures are invalid, then batch verification is not more efficient than individual verification. Our proposal modifies the WLZ scheme [24], where the batch verification costs only 2 pairings and $11n$ exponentiations. But the WLZ scheme and related solutions use uncategorized batch verification which can cause less efficient verification if bogus messages appear during attacks like the Sybil attack, the Denial of Services (DoS) attack etc. However, our solution applies categorized batch verification which sort potential honest messages to the first batch, and potential untrusted messages to the second or third batch with lower

priorities, so the verification phase can be more efficient and strong against Sybil and DoS attacks.
- In V2I communication, our scheme uses probabilistic cryptography for keeping long-term unlinkability and privacy protection of drivers. The join or registration phase takes only two messages (request /response) and the scheme does not need tamper-proof devices. Moreover, we avoid the inefficient linear growth of revocation list with the secret keys of members. Certified pseudonyms are valid to expiration date and, after the expiration date, certified pseudonyms are automatically revoked. Vehicles do not have to deal with a Revocation List (RL). Instead, our proposal uses only a Group Temporary Revocation List (GTRL) to deny malicious members accessing the group of VANET members.

3 Preliminaries

In this section, we describe the basic parties in our security scheme, a communication pattern, requirements and the cryptography background of our scheme.

3.1 Parties in Our Security Scheme

Our security scheme consists of a Trusted Authority (TA), a Group Manager (GM), a user (U) and a Vehicle (V).

- **TA** issues certified member pseudonyms and generates all public cryptographic parameters in our solution. TA is a fully trusted entity in our model and can reveal the real identity of a member (ID) in the revocation phase. TA is securely connected with all group managers (e.g. via Transport Layer Security) and manages the registration of all members.
- **GM** is an entity which generates group secret keys to members in the join phase. In our proposal, we assume that GM is managed by a service provider. GM broadcasts messages in the I2V communication. These messages are signed by GM. GM can also trace and open the malicious messages in its own area but it cannot reveal the user ID.
- **U** is a user with ID. After the registration of the user in TA, U obtains the certified pseudonym. Then, U can join the VANET with a vehicle. Furthermore, U can report a bogus message through the V2I communication to GM.
- **V** is a vehicle representing a user (driver) and user devices (e.g. smartphones, navigations, vehicle's OBU, ...). After joining the GM's area through V2I communication, the vehicle can broadcast and receive messages through the V2V communication or V2I-I2V communication. These messages are signed by a group signature key and verified by a batch or simple verification.

3.2 Communication Pattern

In our communication pattern (see Figure 2), a user U (specifically his/her vehicle V) can broadcast signed messages to other users/vehicles by intra-vehicle communication V2V using short/medium distance communication technologies e.g. Wimax, ZigBee IEEE 802.15.4 or Bluetooth IEEE 802.15.1. See more details in [25]. We assume that the user owns an On Board Unit (OBU) ensuring mainly wireless communication in the V2V connection. The electronic element used to process data and interact with OBU can be an external user personal device such as a smartphone or a navigation device. These devices usually have enough computational power for basic modular arithmetic, pairing and cryptographic operations. The use of these elements Using reduce the overall costs of the VANET architecture.

Furthermore, U can send signed messages via infrastructure connection V2I, ensuring a long-distance mobile radio communication technology e.g. GSM, 3G/4G mobile networks using Internet connection IPv4/IPv6. Road Side Units (RSU) are substituted by existing Base Transceiver Stations (BTS) in GSM or nodes B in 3G networks. Several VANET applications operating with long distances, e.g. monitoring traffic congestion or accidents, send signed messages via a V2I-I2V connection. For better efficiency of the V2I-I2V connection and fast switching of areas, we can adapt the mechanism of data aggregation and data dissemation, described in [21], into a central switch server. These mechanisms are ensured by a service provider that issues VANET applications and navigation services. The service provider manages several group managers for specific areas. GMs are securely connected to a shared database. GMs may act as routers for incoming messages transmitted via the V2I-I2V connection. Every GM is able to verify messages received via the V2I connection while maintaining user privacy. Then, GMs send these messages to vehicles in certain areas. These messages can be signed by a GM private key and easily verified by a GM public key.

Every GM controls a specific area and releases one group public key (gpk) for this area. If a vehicle crosses different boundaries and receives messages from the neighbouring area, then the vehicle determines which messages are sent from a neighboring area due to the fingerprint of gpk in these messages. The vehicle can use the group public key of the neighbouring area that

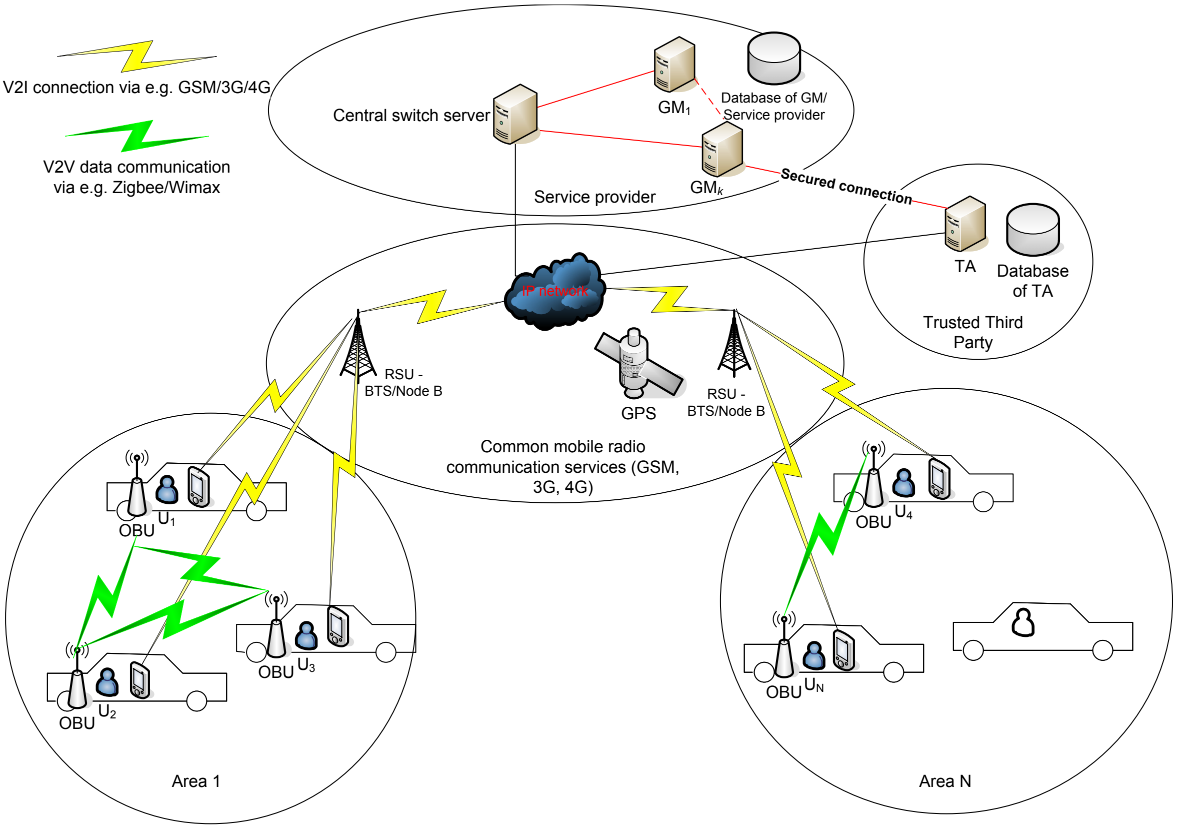

Fig. 2: The communication pattern of our solution.

is stored in a device memory. The group public keys of the area and neighboring areas are obtained if the vehicle enters a new area.

3.3 Requirements

Our scheme is designed to satisfy the following security and practical requirements:

- **Privacy (Revocable Anonymity)**. Our scheme protects driver's privacy in the long-term. An honest driver U with a VANET device and OBU can use the pseudonym signed by TA to obtain group parameters and keys from GM. Then, its OBU can sign every message on behalf of the group members and keep drivers' anonymity. Every malicious driver can be revealed by the collaboration of GM and TA. If some member breaks the rules, his/her messages can be opened by GM and his/her pseudonym is sent to TA, which can extract the member's ID. Next time, when an adversary requests a new pseudonym with a fresh time stamp (e.g. via IETF RFC 3161), TA checks if his/her ID appears in the list of globally revoked members.
- **Non-repudiation, Message Integrity and Authenticity**. In the V2V communication, the group signature ensures that a message is signed by a vehicle which holds the right and fresh group key pair (authenticity). The system must verify the received messages, i.e., the messages that have not been modified once they have been sent (integrity). Members stay private but can not deny that they created the signed messages (non-repudiation).
- **Short-term Linkability**. In several VANET applications like the safe changing of road lanes and the short-term mapping of vehicle movements, the short-term linkability is a desirable property [20]. In a short period, i.e., every 100÷300 ms, broadcasted V2V beacon messages are used to trace the vehicle's position and direction. The current proposals which use group signatures cannot link related messages from one vehicle sent in a short interval. Our scheme balances the privacy of drivers and the linkability of messages, which is available only for a short interval. On the other hand, long-term unlinkability is ensured by using the probabilistic encryption and by changing the pseudonyms in the group signature, e.g., in the V2I-I2V communication.

Table 1: Notations used in our solution.

A_i	the part of a member secret key
α	a random element $\in Z_p^*$
β	a random element $\in Z_p^*$
c	a hash value in the group signature / self-challenge $c \overset{R}{\leftarrow} Z_q$
cer_{U_i}	users' certificate signed by TA
δ	a commitment value in a signature
$e()$	a pairing operation
$enc_{pk_{TA}}$	a ElGamal encryption by TA
$enc_{pk_{U_i}}$	a ElGamal encryption by U
f	the fingerprint of a group public key
g_1	a generator of G_1
g_2	a generator of G_2
G_1	a multiplicative cyclic group of a prime order p
G_2	a multiplicative cyclic group of a prime order p
$gmsk_{GM_k}$	a group manager secret key
gpk_{GM_k}	a group public key
GRL	Global Revocation List
gsk_{V_i}	a group member secret key
GTRL	Group Temporary Revocation List
γ	a random element $\in Z_p^*$
h	a random element $\in G_1^*$
H	a hash function
ch	a challenge $c \overset{R}{\leftarrow} Z_q$
ID_{U_i}	a user ID
k	a counter value
l	the security length of parameters
M	a message
μ	a commitment value in a signature
π_{U_i}	the user certificate issued by TA
p_i	a temporary result of the pairing
pk_{GM_k}	an ElGamal public key of GM
pk_{TA}	an ElGamal private key of TA
pk_{U_i}	an ElGamal private key of a user
r	random elements $\in Z_p^*$
R_i	a commitment value in a signature
s	elements in signature$\in Z_q$
sig_{GM_k}	an ECDSA private key of GM
sig_{TA}	an ECDSA private key of TA
sk_{GM_k}	an ElGamal private key of GM
sk_{TA}	an ElGamal private key of TA
sig_{U_i}	an ECDSA private key of a user
sk_{U_i}	an ElGamal private key of a use
σ	the product of a group signature
T_i	pseudonyms in a signature
TL	Temporary List
T_l	a time stamp
θ	random elements $\in Z_p$
u	the element of a group public key
v	the element of a group public key
ver_{GM_k}	an ECDSA public key of GM
ver_{TA}	an ECDSA public key of TA
ver_{U_i}	an ECDSA public key of a user
w	the element of a group public key
W	a validity value
x_i	the element of a group member secret key
Z_p	the (set of) p-adic integers
Z_q	the (set of) q-adic integers

3.4 Cryptography Background

Our solution employs the Elliptic Curve Digital Signature Algorithm (ECDSA) [12] as a signature scheme with the public/private keys of TA, GM, V. Petit and Mammeri [16] investigate the authentication algorithm ECDSA in vehicular networks, and processing delay of verification takes around 5 ms for ECDSA with P-256 bit curves measured on a Pentium D 3.4GHz workstation. Additionally, we use a probabilistic ElGamal encryption/decryption during the join of members. The modified short group signature WLZ scheme [24], based on the BBS04 scheme [3] is used in the V2V communication. This scheme uses bilinear maps and it is based on the q-SDH problem and the Decision Linear problem, which have been studied in [3].

We follow the notation of [3] for the concept of bilinear maps: G_1, G_2 and G_T are multiplicative cyclic groups of a prime order p. Then, g_1 is a generator of G_1; g_2 is a generator of G_2; and ψ is an isomorphism from G_2 to G_1 that $\psi(g_2) = g_1$. So e is a computable bilinear map $e : G_1 \times G_2 \rightarrow G_T$ with the following properties:

- Bilinearity: for all $u \in G_1, v \in G_2$ and $a, b \in Z_p, e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: $e(g_1, g_2) \neq 1_{G_T}$.

The q-Strong Diffie-Hellman problem is a hard computational problem where (q+2)-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{\gamma^2}$, $..., g_2^{\gamma^q})$ is the input and a pair $(g_1^{\frac{1}{x+\gamma}}, x)$ is the output.

The Decision Linear Diffie-Hellman problem. Given $u, v, h, u^a, v^b, h^c \in G_1$ as input, the output is *yes* if $a + b = c$ and *no* otherwise. This is detailed in [3].

4 Our Solution

In this section, we describe our solution. The notations used are described in Table 1. We focus on the practical registration and join of VANET members and the efficient signing/verification of V2V and V2I-I2V messages. Our solution consists of seven phases: Setup, Registration, Join, Signing, Categorized Verification, Trace, Revocation.

4.1 Setup $\mathbf{Set}(0,1)^l \rightarrow parameters$

In the first part, TA chooses parameters $(G_1, G_2, g_1, g_2, \psi, e)$ and generates an ECDSA key pair sig_{TA}/ver_{TA}, an ElGamal private key sk_{TA} and a public key pk_{TA}. It then releases the public keys and parameters. GMs generate group signature keys, ElGamal private sk_{GM_k}, an ECDSA key pair sig_{GM}/ver_{GM} and public pk_{GM_k} keys for the secure V2I communication and publish public keys. Every GM$_k$ randomly selects $r_1, r_2 \in Z_p^*, h \in G_1^*$ and sets u, v such that $u^{r_1} = v^{r_2} = h$. Then, GM$_k$ selects random $\gamma \in Z_p^*$ and computes $w = g_2^{\gamma}$. The group public key is $gpk_{GM_k} = (g_1, g_2, u, v, w, h)$ and the group manager secret key is $gmsk_{GM_k} = (r_1, r_2)$.

4.2 Registration $\mathbf{Reg}(ID_{Ui}) \rightarrow \pi_{Ui}$

In the registration phase, the i-th user (member) U$_i$ using a vehicle V$_i$ with OBU, requests a valid certified pseudonym π_{U_i} from TA. First, the user follows an off-line registration step to get the signed certificate cer_{U_i}. After this process, U$_i$ owns her cer_{U_i} and she can perform the on-line registration step to get her pseudonym, which has an expiration time.

4.2.1 Off-line Registration

For the first time, TA must physically verify the driver's real ID, his/her driving license and OBU's ID number. U$_i$ then creates an ECDSA key pair sig_{U_i}/ver_{U_i}, gives the public key to TA, which stores (ID_{U_i}, ver_{U_i}) in the database, and the signed certificate $cer_{U_i} = sig_{TA}(ID_{U_i}, ver_{U_i})$ is given to V$_i$.

4.2.2 On-line Registration

After a successful off-line registration process, the driver can request his/her pseudonym online. Assuming that U$_i$ has pk_{TA}, ver_{TA}, the two-message of the registration phase consists of these steps:

1. U$_i$ self-generates ElGamal key pair (sk_{U_i}/pk_{U_i}) and sends the encrypted request $enc_{pk_{TA}}(pk_{U_i}||ID_{U_i}|| sig_{U_i}(pk_{U_i}))$ to TA.

2. TA decrypts the request and checks if ID_{U_i} is not revoked in Global Revocation List (GRL), the certificate cer_{U_i} and the user's signature, which ensures user's authenticity and commits the pk_{U_i} in the certificate with new ElGamal key pair. Then, TA generates a challenge $ch \overset{R}{\leftarrow} Z_q$, a time stamp T_l and sends the encrypted response $enc_{pk_{U_i}}(enc_{pk_{TA}}(ID|| \; ver_{U_i}||ch)||T_l|| sig_{TA}(T_l||enc_{pk_{TA}}(ID|| \; ver_{U_i}||ch)||pk_{U_i}))$ back to U$_i$. Finally, U$_i$ checks the signature by TA and composes the pseudonym $\pi_{U_i} = pk_{U_i}||enc_{pk_{TA}}(ID||ver_{U_i}||ch)||T_l|| sig_{TA}(T_l||enc_{pk_{TA}}(ID||ver_{U_i}|| \; ch)||pk_{U_i})$ and stores it.

4.3 Join $\mathbf{Join}(\pi_{U_i}) \rightarrow gsk_{V_i}, gpk_{GM_k}$

A vehicle V$_i$ with the user U$_i$ entering the k-th GM$_k$ area for the first time, requests the group public key and his/her group member secret key. Let $H()$ be a hash function and let the two-message join phase consist of these steps:

1. V$_i$ sends $\pi_{U_i} = pk_{U_i}||enc_{pk_{TA}}(ID||ver_{U_i}||ch)||T_l|| sig_{TA}(T_l||enc_{pk_{TA}}(ID||ver_{U_i}||ch)||pk_{U_i})$,which is encrypted using pk_{GM_k}, to GM$_k$.

2. GM$_k$ decrypts π_{U_i} using sk_{GM_k}, verifies π_{U_i}, which is signed by TA and controls if $enc_{pk_{TA}}(ID||ver_{U_i}||ch)$ is not in Group Temporary Revocation List (GTRL) and the validity of the time stamp T_l. If π_{U_i} is fine, GM creates $gsk_{V_i} = (x_i, A_i)$, where $x_i \;=\; H(enc_{pk_{TA}}(ID|| ver_{U_i}||ch)||T_l||\gamma), A_i = g_1^{\frac{1}{x_i+\gamma}}$, and stores $(enc_{pk_{TA}}(ID|| ver_{U_i}||ch), A_i, T_l)$ to the join table and sends ver_{GM_k}, gpk_{GM_k}, the group public keys of neighboring areas and gsk_{V_i} encrypted using pk_{U_i} to V$_i$.

We note that ElGamal encryption/decryption is probabilistic. Due to this fact, an observer can not link two or more encrypted messages if V$_i$ requests gsk_{V_i} for the second time.

4.4 Signing $\mathbf{Sig}(M, gsk_{V_i}, gpk_{GM_k}) \rightarrow \sigma$

The signing phase applies the modified short group signature WLZ scheme [24], which is based on the BBS04 scheme [3]. We include a counter k in the OBUs, a member secret key $gsk_{V_i} = (x_i, A_i)$ and a group public key $gpk_{GM_k} = (g_1, g_2, h, u, v, w)$. OBU signs a message $M \in$ (0,1)* and outputs the signature of knowledge $\sigma \;=\; (T_1, T_2, T_3, R_2, R_3, R_5, c,\; s_\alpha, s_\beta, s_x, s_\delta, s_\mu)$.
If $k = 0$, V$_i$ generates $\alpha, \beta, r_\alpha, r_\beta, r_x, r_\delta, r_\mu \in Z_p^*$, and computes

$$\begin{aligned} T_1 \;=\; u^{\alpha}, T_2 \;=\; v^{\beta}, T_3 \;=\; A_i h^{\alpha+\beta}, \\ \delta \;=\; \alpha x, \mu \;=\; \beta x. \end{aligned} \tag{1}$$

$$p_1 \;=\; e(T_3, g_2), p_2 \;=\; e(h, w), p_3 \;=\; e(h, g_2). \tag{2}$$

stores $T_1, T_2, T_3, \delta, \mu, p_1, p_2, p_3$, and computes

$$\begin{aligned} R_1 = u^{r_\alpha}, R_2 = v^{r_\beta}, R_3 \;=\; p_1^{r_x} \cdot p_2^{-r_\alpha - r_\beta} \cdot p_3^{-r_\delta - r_\mu}, \\ R_4 = T_1^{r_x} u^{-r_\delta}, R_5 = T_2^{r_x} v^{-r_\mu}, \end{aligned} \tag{3}$$

$$c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \tag{4}$$

$$\begin{aligned} s_\alpha = r_\alpha + c\alpha, s_\beta = r_\beta + c\beta, s_x = r_x + cx, \\ s_\delta = r_\delta + c\delta, s_\mu = r_\mu + c\mu. \end{aligned} \tag{5}$$

Finally, V$_i$ increases the counter k++, computes the fingerprint f of the group public key by the hash function (e.i. SHA-256) and sends the message M with the signature $\sigma = (T_1, T_2, T_3, R_2, R_3, R_5, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu, f)$.

If α and β are unchanged every n messages, the short-term linkability is kept because the pseudonyms of group signature T_1, T_2, T_3 are also unchanged. Thus, for n messages, when $1 \leq k \leq n-1$, V$_i$ does not need to compute equations 1, 2, contrary the WLZ scheme, but only generates random $r_\alpha, r_\beta, r_x, r_\delta, r_\mu \in Z_p^*$ and computes equations 3, 4 and 5. This reduces all 3 bilinear operations to 0, 10 exponentiations to 9, and 14 multiplications to 9. This mode is suitable for the fast V2V communication where the short-term linkability is demanded. The concrete VANET application can decide when to fix the counter $k = 0$ and V$_i$ generates new α and β and recomputes the equations 1 and 2. This mode is suitable for the V2I or V2I-I2V communication, where user privacy is more imported than the efficiency of signing. It is worth mentioning that pairing equations p_2, p_3 are fixed and can be precomputed only once.

4.5 Categorized Verification

Our solution uses a categorized verification which sorts the incoming signed messages to three levels of credibility. Due to the short-term linkability, V$_i$ can keep the Temporary List (TL) of known vehicles. Firstly, the received message M_j is checked by V$_i$ if it contains a valid time stamp, real and consistent data. The precise value of the time stamp, respectively time window, depends on a concrete VANET application, used communication technology, distance with specific latency etc. Furthermore, V$_i$ has to check the fingerprint f of the group public key in every received signature so that all received signed messages are from one area with gpk_{GM_k}. Received messages with signatures that contain different fingerprints f have to be verified by the different and appropriate group public keys.

After that, the message with the group signature containing T_3 is checked if T_3 is in TL. If it holds, the recorded T_3 with previous validity (W=1) is included and sorted in the first batch. The validity W can be a boolean value which indicates valid (W=1) or invalid (and unknown, W=0) signatures. If T_3 is not in TL, the signed message with the unknown T_3 is sorted to the second batch which is verified after the first batch verification. This category is formed by the messages sent via the V2I-I2V communication. If OBU has enough time for message validation, the rest of signed messages with T_3 linked with W=0 are verified in the third batch at the end of verification. This behaviour limits the effectiveness of Denial of Service attacks where malicious cars try to use eavesdropped T_3 and generate a lot of invalid signatures with known T_3. This approach improves the efficiency of the batch verification process and helps when an attacker, who is out of the group, generates unsigned or corrupted messages.

4.5.1 Batch Verification $\boldsymbol{Ver}(M, gpk_{GM_k}, \sigma) \rightarrow$ *valid/invalid*

Batch verification is investigated in [6], and it verifies n messages in one batch. V$_i$ uses $gpk_{GM_k} = (g_1, g_2, h, u, v, w)$ to verify messages $\sigma_j = (T_{j1}, T_{j2}, T_{j3}, R_{j2}, R_{j3}, R_{j5}, c_j, s_{j\alpha}, s_{j\beta}, s_{jx}, s_{j\delta}, s_{j\mu})$ for $j = 1, ..., n$.

V$_i$ restores $\overline{R}_{j1} = u^{s_{j\alpha}} T_{j1}^{-c}, \overline{R}_{j4} = u^{-s_{j\delta}} T_{j1}^{s_x}$, computes a new control hash c'_j from received parameters $c'_j = H(M_j, T_{j1}, T_{j2}, T_{j3}, \overline{R}_{j1}, R_{j2}, R_{j3}, \overline{R}_{j4}, R_{j5})$, and checks if $c'_j = c_j$. If yes, then V$_i$ continues with verification. Otherwise, the message with the signature is inconsistent and it is refused.

V$_i$ randomly selects $\theta_1, \theta_2, ..., \theta_n \in Z_p$ with l_b bit (the Small Exponent Test [2]),

checks batch if

$$\prod_{j=1}^{j=n} R_{j3}^{\theta_j} = e(\prod_{j=1}^{j=n} (T_{j3}^{s_{jx}} h^{-s_{j\delta}-s_{j\mu}} g_1^{-c_j})^{\theta_j}, g_2) \\ e(\prod_{j=1}^{j=n} (T_{j3}^{c_j} h^{-s_{j\alpha}-s_{j\beta}})^{\theta_j}, w) \tag{6}$$

and if

$$1_{G_1} = (R_{j5} R_{j2})^{-\theta_j} T_{j2}^{\theta_j s_{jx} - \theta_j c_j} v^{(s_{j\beta}-s_{j\mu})\theta_j}. \tag{7}$$

The signed message is valid if equations 6 and 7 hold. All T_3s from new valid signed messages are added to TL with W=1. In case that the batch verification fails, the divide-and-conquer approach is used to identify the invalid signatures that were added to TL with W=0. The honest messages keep the mark W=1.

4.5.2 Individual Verification $\boldsymbol{Ver}(M, gpk_{GM_k}, \sigma) \rightarrow$ *valid/invalid*

At the end of the divide-and-conquer approach, the final two messages are individually verified.
V$_i$ restores $\overline{R}_1 = u^{s_\alpha} T_1^{-c}, \overline{R}_4 = u^{-s_\delta} T_1^{s_x}$, computes new control hash c' from received parameters
$c' = H(M, T_1, T_2, T_3, \overline{R}_1, R_2, R_3, \overline{R}_4, R_5)$,
and checks if $c' = c$. If it is equal, V$_i$ then continues with the verification. Otherwise, the message is inconsistent and it is refused.

Then, V$_i$ checks if

$$R_3 = e(T_3, g_2)^{s_x} e(h, w)^{(-s_\alpha - s_\beta)} e(h, g_2)^{(-s_\delta - s_\mu)} \\ (e(T_3, w) e(g_1, g_2)^{-1})^c \tag{8}$$

and

$$1_{G_1} = (R_5 R_2)^{-1} T_2^{s_x - cx} v^{(s_\beta - s_\mu)}. \tag{9}$$

The signed message is valid if equations 8 and 9 hold.

We can see from equations 6 and 8 that individual verification has a cost of 5 pairing operations per one message but batch verification costs only 2 pairing operations per n messages. This is the main reason why we avoid individual verification and propose to use the categorized batch verification.

In some long-distance VANET applications, GMs may act as routers for incoming messages transmitted via V2I-I2V communication. In this case, GM$_k$ receives the messages and verifies their signatures, signs the valid ones using its own private ECDSA key sig_{GM_k}, and finally submits them to all the users in a certain k-area. Then, these users can easily verify the signature issued by GM$_k$ using the public ECDSA key ver_{GM_k}.

4.6 Trace $\mathbf{Trace}(M, \sigma, gmsk_{GM_k}) \rightarrow gsk_{V_i}, \pi_{U_i}$

Every bogus signed message can be opened by GM$_k$ using the group manager secret key $gmsk_{GM_k} = (r_1, r_2)$. Bogus messages are messages with correct signatures that carry malicious content which can cause problems in traffic. GM$_k$ extracts the part of the member secret group key $gsk_{V_i} \rightarrow A_i = T_3/(T_1^{r_1} \cdot T_2^{r_2})$ and searches the record $(enc_{pk_{TA}}(ID||ver_{V_i}||c), A_i, T_l)$ in the database. The part of the member pseudonym can be sent to TA for revocation.

4.7 Revocation $\mathbf{Rev}(\pi_{U_i}) \rightarrow ID_{V_i}$

When there are serious circumstances, e.g., an accident, a malicious member is revoked globally by the cooperation of GM$_k$ and TA. GM$_k$ is able to open a message and extract the member pseudonym that is sent to TA. TA broadcasts $rev = (enc_{pk_{TA}}(ID||ver_{V_i}||c), T_l)||$ $sig_{TA}(rev)$ to other active GMs which check the signature and store rev to own GTRLs until the lifetime of this pseudonym expires. TA extracts ID_{V_i} and adds it to GRL so that the malicious member can not refresh his/her pseudonym in the next registration phase.

5 Security Analysis

We next detail the adversary model and the possible attacks the proposed scheme has to be robust against. These attacks are related to the security requirements which must be fulfilled by our scheme, and which were introduced in Section 2.3: *revocable anonymity*, *message integrity* and *message authenticity.*

5.1 Adversary Model

Our attacker model considers an adversary who can control vehicles and can also access communication lines to capture, modify and retransmit messages. In this way, she can be a purely external attacker and also an internal one. In any case, her computational power does not permit the adversary to break current computationally secure cryptosystems.

Regarding the other entities of the proposed system, the Trusted Authority (TA) is managed by some governmental organization such as the traffic authority of each country. Therefore, this entity is fully trusted. Regarding the Group Managers (GMs), these elements are assumed to be managed by some company that participates in the system as a service provider. In this way, GMs are expected to follow the proposed protocol in an honest way (i.e., they will not tamper with messages, drop them, etc) but they may try to retrieve the real identities of the users who use the VANET. Gathering real identities and other personal data may report significant economical benefits to the company in charge [1] and it is an explicit privacy threat. Therefore, they are covered in the proposed adversary model as passive attackers that uniquely try to break the privacy of the legitimate users. In this way, they will not participate in any other kind of attack. Moreover, the RSUs which are used by the GMs to communicate with the vehicles of the VANET are assumed to be tamper-proof elements which cannot be compromised by external attackers.

We next summarize the attacks that can be performed by the considered adversaries. They can be broadly divided into *passive* and *active* attacks:

- *Passive attacks.* They only require the attacker to have access to the communication lines. Their main purpose is to jeopardize the privacy of the users by compromising the confidentiality and/or unlinkability of the submitted messages. Specifically, those attacks are:
  - Eavesdrop messages transmitted between V_i and TA in the *Online Registration step.*
  - Eavesdrop messages transmitted between V_i and GM in the *Join step.*
  - Eavesdrop messages transmitted between TA and GM.
  - Trace the V2V/V2I messages sent by a certain user.
  - Retrieve the real identity of a certain user in the *Join step.*
- *Active attacks.* These attacks are based on tampering with valid messages, submitting fake ones, etc; Their main purpose is to get some benefit or simply disrupt the normal execution of the proposed

scheme. This kind of attacks generally compromise the integrity and/or the authenticity of the submitted messages. Specifically, our system should be strong against:

- Tamper with messages transmitted between V_i and TA in the *Online Registration step.*
- Tamper with messages transmitted between V_i and GM in the *Join step.*
- Tamper with V2V/V2I messages sent by legitimate vehicles.
- Tamper with messages transmitted between TA and GM.
- Generate a fake but valid pseudonym.
- Allow unauthorized users to generate fake but valid V2V/V2I messages.
- Launch a DoS attack against the vehicles of the VANET.
- Reuse former messages to perform replay attacks.
- Use the anonymity provided by the scheme to misbehave without being traced.

5.2 System's Behaviour against the Considered Attacks

We next explain how the proposed scheme deals with the attacks which have been introduced above. Note that some of these attacks may be covered together in the same subsection.

5.2.1 Eavesdrop Messages Transmitted during the Different Steps of the Protocol

First, we focus on the messages transmitted between V_i and TA in the *Online Registration step.* In this case, V_i sends a *request* $(enc_{pk_{TA}}(pk_{U_i}||ID_{U_i}||sig_{U_i}(pk_{U_i})))$ in order to get a new pseudonym and TA answers with a *response* $(enc_{pk_{U_i}}(enc_{pk_{TA}}(ID||ver_{U_i}||ch)||T_l||sig_{TA}(T_l ||enc_{pk_{TA}}(ID|| ver_{U_i}||ch)||pk_{U_i})))$. Both messages are encrypted using ElGamal cryptosystem (nowadays this cryptosystem is considered to be secure [22]) and, hence, the attacker is unable to decrypt them and get the transmitted data because decryption requires knowledge of the secret keys sk_{U_i} and sk_{TA}. These keys are only known by the legitimate user and the trusted authority, respectively.

Similarly, the attacker cannot get the data transmitted between V_i and GM in the *Join step* because these messages are also encrypted using ElGamal cryptosystem. In this case, the secret keys that are needed to obtain the sensitive information are sk_{U_i} and sk_{GM_k}. Both keys are only known by the legitimate user and the contacted group manager. Note that, as explained previously, group managers are controlled by a service provider and, hence, they are expected to behave honestly.

Finally, the attacker cannot disclose any information from the messages sent between the trusted authority and the different group managers due to the fact that these communications are always secured using TLS.

5.2.2 Trace the V2V/V2I Messages Sent by a Certain User

Vehicles apply the modified short group signature WLZ scheme [24] to sign the V2V/V2I messages that they submit. Group signatures generated under this scheme contain the group members' pseudonyms T_1, T_2, T_3 which are a linear encryption of members' secret key A_i and random α and β. The short-term linkability property of the messages does not violate the drivers' privacy. When the counter k is set to 0 and V_i generates new values for α and β, the new generated signatures are unlinkable with the former ones because they contain new values for T_1, T_2 and T_3.

5.2.3 Retrieve the Real Identity of a Certain User in the Join step

This attack is based on retrieving the real identity of a certain user from its pseudonym π_{U_i} in the *Join step.* Note that this attack can only be performed by the GM that is expected to receive the message because π_{U_i} is encrypted using its ElGamal public key pk_{GM}.

Pseudonym π_{U_i} contains the identity of the user (ID) encrypted with the ElGamal public key pk_{TA}, which is only known by TA. Therefore, GM cannot retrieve the real ID. Nevertheless, GM is capable of linking all the request messages that contain the same π_{U_i}. In order to minimize this issue, the user should update π_{U_i} with a certain frequency (following the *Online Registration step*).

5.2.4 Tamper with Messages Transmitted during the Different Steps of the Protocol

Focusing on the messages transmitted between V_i and TA in the *Online Registration step*, message integrity and authenticity are ensured by the ECDSA signature scheme. The *request message* contains the member public key pk_{U_i} signed with the ECDSA signature key sig_{U_i}. Assuming that both the ECDSA signature scheme and the hash function in use are secure, if the request message is modified in any way, the ECDSA verification process will detect this situation.

Messages transmitted between V_i and GM in the *Join step* also ensure integrity and authenticity. First,

V_i submits its pseudonym, which is signed by the TA using the ECDSA signature scheme. Then, GM sends to V_i its assigned group member secret key $gsk_{V_i} = (x_i, A_i)$. The use of a hash function to compute gsk_{V_i} together with the use of ElGamal cryptosystem to encrypt the message provide integrity and authenticity.

Regarding the V2V/V2I messages sent by the vehicles, those elements are signed and verified employing the modified short group signature WLZ scheme [24]. This approach ensures message authenticity and integrity to those messages.

Finally, the attacker cannot tamper with the data exchanged between the trusted authority and the different group managers due to the integrity and authenticity properties provided by the use of TLS.

5.2.5 Generate a Fake but Valid Pseudonym

If the attacker wants to create a valid pseudonym π_{U_i}, she needs the ECDSA private key sig_{TA}. This secret key is only known by the TA and, hence, the attacker cannot obtain it to launch this attack.

It is worth mentioning that if an illegal π_{U_i} is sent to a legitimate user, she can use the TA's public ECDSA key ver_{TA} to verify its validity.

5.2.6 Allow Unauthorized Users to Generate Fake but Valid V2V/V2I Messages

The attacker can launch this attack by signing a new fake message on behalf of a group of legitimate users or by modifying a message signed and submitted by a legitimate user.

The signing and verification phases employ short group signatures with the short-term linkability to ensure message authenticity and integrity. As explained previously, our scheme applies the modified short group signature WLZ scheme [24] and inherits all its security features. As a result, only the group manager GM_i and the valid group members U_i can sign a message on behalf of the group.

If an attacker without the valid $gsk_{V_i} = (A_i, x_i)$ is willing to modify a certain message, she must recompute the hash c and some signature parts. Assuming that the hash function is secure and that the Discrete Logarithm problem holds, computing ($s_{j\alpha}$, $s_{j\beta}$, s_{jx}, $s_{j\delta}$, $s_{j\mu}$) without knowing x_i is considered unfeasible. If this proof of knowledge is incorrectly computed, equations 6, 7 and 8, 9 will not hold during the verification step.

5.2.7 Launch a DoS Attack against the Vehicles of the VANET

The attacker can launch a DoS attack by broadcasting a large number of bogus messages containing fake pseudonyms and signatures. This attack can be more effective if several attackers collaborate on this purpose (note that, a Sybil attack can be considered to achieve this).

As a result of this attack, legitimate users will be flooded with a large amount of messages and they will not be able to process all of them. The straightforward solution for this situation is to discard some of the received messages (or all of them). The problem of this approach is that some of these discarded messages can be legitimate warnings of some dangerous situation. In order to prevent it, our scheme implements a *categorized batch verification* step.

In this way, a honest user has a Temporary List (TL) of other known and honest drivers, which uses the short-term linkability property that keeps the pseudonym T_3 of each signed message unchanged for a certain period of time. This user receives messages and checks the TL to put the messages containing a known T_3 in the first batch of verification (the one with the highest priority). Messages with an unknown pseudonym are stored in the second batch. Finally, potentially untrusted messages (e.g., with validity $W = 0$) are verified in the third batch only if Bob's OBU has free time and computational capacity to do it.

5.2.8 Reuse Former Messages to Perform Replay Attacks

Submitted messages contain a time stamp with current time and date. Before being verified, the time stamp of each received message is checked. If an attacker without the valid $gsk_{V_i} = (A_i, x_i)$ is willing to reuse an old message with a valid signature, she must refresh the time stamp and then recompute the hash c_j and the signature $(s_{j\alpha}, s_{j\beta}, s_{jx}, s_{j\delta}, s_{j\mu})$. Note that obtaining valid values for $s_{jx}, s_{j\delta}$ and $s_{j\mu}$ without knowing x_i is unfeasible under the Discrete Logarithm problem.

5.2.9 Use the Anonymity Provided by the Scheme to Misbehave without Being Traced

The proposed scheme provides anonymity and unlinkability for drivers in front of other vehicles and GMs. Nevertheless, this protection can be revoked if the GM of the area and the TA collude. Since both entities are honest, this will be assumed to happen only if the driver misbehaves.

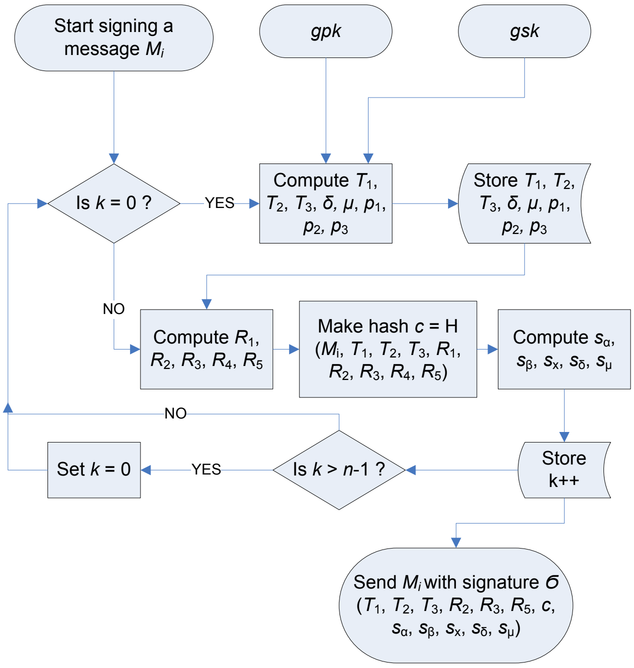
Start signing a message M_i
gpk
gsk
Is k = 0 ?
YES
Compute T_1, T_2, T_3, δ, μ, p_1, p_2, p_3
Store T_1, T_2, T_3, δ, μ, p_1, p_2, p_3
NO
Compute R_1, R_2, R_3, R_4, R_5
Make hash c = H (M_i, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)
Compute s_α, s_β, s_x, s_δ, s_μ
NO
Set k = 0
YES
Is k > n-1 ?
Store k++
Send M_i with signature σ (T_1, T_2, T_3, R_2, R_3, R_5, c, s_α, s_β, s_x, s_δ, s_μ)

Fig. 3: The process flowchart of signing.

If this is the case, each correct message submitted by a malicious member can be opened by the GM using its group manager secret key $gmsk_{GM_k}$. In this way, the GM extracts the part of the member secret group key $gsk_{V_i} \to A_i = T_3/(T_1^{r_1} \cdot T_2^{r_2})$ and searches the record $(enc_{pk_{TA}}(ID||ver_{V_i}||c), A_i, T_l)$ in the database. Finally, the part of the member pseudonym can be sent to the TA for retrieving the real ID.

6 Experimental Implementation

We have implemented our scheme as a proof-of-concept in JAVA (PC) and the Android platform (smartphones). The main core of our experimental implementations is formed by the group signature scheme that uses the Java Pairing Based Cryptography (jPBC) Library[1] in both test scenarios (jPBC on Java for the PC version and wrapped jPBC on Android for the smartphone version). The implementation employs the MNT curves type D with the embedding degree $k = 6$, the 171-bit order of curves and the pre-generated parameters d840347-175-161.param.

The registration and join phases use the ECDSA signature scheme and ElGamal cryptosystem that are provided by the Bouncy castle Library [2]. All ECDSA and ElGamal keys can be inherited from class org.bouncycastle.jce.provider.JDKKeyFactory. We used the 1024-bit ElGamal encryption and the 256-bit ECDSA scheme with the SHA-1 hash function.

In the signing phase, Fig. 3, a string of a message M_i, counter k, a member secret key gsk and a group public key gpk input to the signing process. There are two modes of signing: an initial signing mode and a normal signing mode. The initial mode of signing is performed if $k = 0$. The signature algorithm then computes $T_1, T_2, T_3, \delta, \mu, p_1,\ p_2, p_3$, where 3 pairing operations are computed. This mode is used in the V2I-I2V communication, respectively, in the long distance VANET applications.

The normal mode of signing is performed if $1 \leq k \leq n-1$ and the signature algorithm uses the stored parameters $T_1, T_2, T_3, \delta, \mu, p_1,\ p_2, p_3$. M_i is signed and the signature σ with elements $T_1, T_2,\ T_3, R_2, R_3, R_5, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu)$ is produced. Then the message M_i containing the signature σ is sent. The normal mode is used in the fast V2V communication,in the short distance VANET application respectively. The proposed signing phase is depicted as a flowchart in more detail in Fig. 3.

A receiver (verifier) receives the M_i and checks the time stamp and consistency of the message. Then, the receiver checks the validity of elements R_1, R_4, c' and saves the incoming M_i to an input buffer. Messages are sorted out into three categories, and 3 buffers, respectively. The sorting process is based on knowing the T_3 of incoming messages and the validity indicator W (a boolean type). Depending on the permitted number of received messages and the maximal time limit of the verification phase, the verifier starts to do the batch verification.

The categorized verification process outcomes the list of valid messages and upgrades a temporary list with the elements T_3 and W. If M_i is valid, then W is set to true. Otherwise, if M_i is invalid, then W is set to false. The proposed categorized batch verification is depicted in more detail as a flowchart in Fig. 4. At the end of the verification process, the valid messages are sent to VANET applications depending on their time priority. The performance results of the implemented signing and categorized verification phases are outlined in the following section.

7 Evaluation of Our Solution

We outline a theoretical evaluation and comparison of our scheme with the related VANET schemes which use group signatures, GSIS [13], Zhang et al. [27] , Ferrara et al. [6] and the scheme of Wei et al. (WLZ scheme) [24]. This evaluation is independent from the used ma-

[1] (avail. on http://gas.dia.unisa.it/projects/jpbc/index.html)
[2] (avail. on http://www.bouncycastle.org/resources.html)

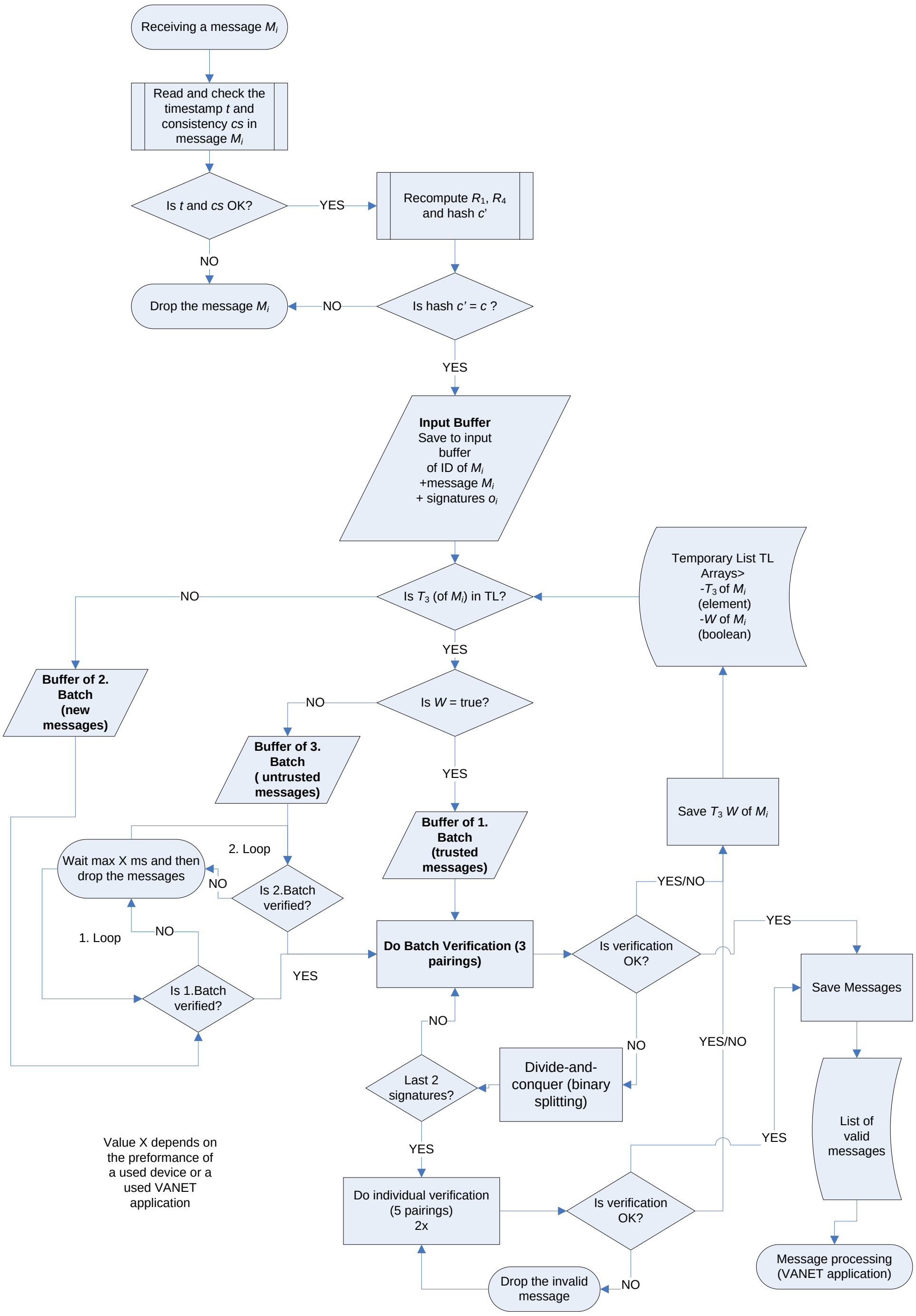
Receiving a message M_i
Read and check the timestamp t and consistency cs in message M_i
Is t and cs OK?
YES
Recompute R_1, R_4 and hash c'
NO
Drop the message M_i
NO
Is hash $c' = c$?
YES
Input Buffer
Save to input buffer of ID of M_i +message M_i + signatures o_i
Temporary List TL Arrays> -T_3 of M_i (element) -W of M_i (boolean)
NO
Is T_3 (of M_i) in TL?
YES
Buffer of 2. Batch (new messages)
NO
Is W = true?
Buffer of 3. Batch (untrusted messages)
YES
Save T_3 W of M_i
Buffer of 1. Batch (trusted messages)
2. Loop
Wait max X ms and then drop the messages
NO
Is 2.Batch verified?
YES/NO
YES
1. Loop
NO
Do Batch Verification (3 pairings)
Is verification OK?
YES
Is 1.Batch verified?
Save Messages
NO
NO
YES/NO
Last 2 signatures?
Divide-and-conquer (binary splitting)
List of valid messages
YES
Value X depends on the preformance of a used device or a used VANET application
YES
Do individual verification (5 pairings) 2x
Is verification OK?
Message processing (VANET application)
Drop the invalid message
NO

Fig. 4: The process flowchart of categorized batch verification.

chine. In addition, we compare the experimental implementation of our scheme and the implementation of BBS group signature scheme used in the related works. The implementation of our solution runs on two platforms, namely JAVA (PC) and the Android platform (smartphones).

7.1 Theoretical Evaluation and Comparison

Generally, the time of bilinear pairing T_p is considered the most expensive operation (ten times more expensive than exponentiation operation T_e) and exponentiation is more expensive than multiplication T_m. The modular arithmetic operations like addition and subtraction can be computed more efficiently than multiplication

V2V scheme:	Our scheme	WLZ scheme [24]	GSIS [13]	Zhang et al. [27]	Ferrara et al. [6]
Batch:	yes	yes	no	yes	yes
Short-term linkability:	yes	no	no	no	no
Length of signature:	$5G_1, G_T, 5Z_p, f$ (2636 bits)	$5G_1, G_T, 5Z_p$ (2380 bits)	$3G_1, 6Z_p$ (1500 bits)	$7G_1, G_T, 5Z_p$ (2570 bits)	$3G_1, G_T, 6Z_p$ (2032 bits)
Performance of batch verification					
Pairings	**2**	**2**	5n	2	2
Exponentiation	**11n**	**11n**	12n	14n	13n
Multiplication	11n+1	11n+1	**8n**	17n	10n+1
Performance of individual verification					
Pairings	5	5	5	5	5
Exponentiation	**10**	**10**	12	12	12
Multiplication	9	9	8	8	8
Performance of initial mode signing / normal mode signing					
Pairings	3 / **0**	3 / 3	3 / 3	3 / 3	3 / 3
Exponentiation	12 / **9**	10 / 10	12 / 12	12 / 12	12 / 12
Multiplication	12 / **9**	14 / 14	12 / 12	12 / 12	12 / 12

Table 2: The comparison of verification and signing.

and exponentiation. See results in [15]. Consequently, we omit these fast operations in this performance evaluation. In our cryptographic scheme, the initial signing mode takes $3T_p + 12T_e + 12T_m$ and the normal signing mode takes only $9T_e + 9T_m$. The computation complexity of verification is linear and depends on the number n of received messages. The verification takes $2T_p + 11nT_e + (11n + 1)T_m$ in the batch verification mode, and $5T_p + 10T_e + 9T_m$ in the individual verification mode.

The signing phase of our scheme costs less exponentiations than the signing phase of the related schemes. Moreover, during the normal mode signing of x messages with short-term linkability, all operations are significantly reduced to pairing ($3 \Rightarrow 0$), exponentiation ($10 \Rightarrow 9$) and multiplication ($14 \Rightarrow 9$).

Our scheme based on the group signature scheme BBS04 [3] reaches more efficient batch verification ($2\ T_p + 11n\ T_e$) and individual verification ($5\ T_p + 10\ T_e$) than the compared schemes (see Table 2). But the related solutions like Zhang et al. [27], Ferrara et al. [6], the WS2010 scheme [23] and also the WLZ scheme [24] use uncategorized batch verification that can be negatively affected by malicious messages ($\geq 15\%$ from all messages). To our best knowledge, our proposal applies the categorized batch verification with the short-term linkability in VANET for the first time. Our categorized batch verification with the temporary list of known vehicles reaches the high correctness of the important first batch in case the bogus or damaged signed messages appear in the V2V communication. In case a malicious driver Eve (E) starts the Sybil attack, which is a special kind of the DoS attack, then she broadcasts bogus messages that contain fake pseudonyms and signatures. Meanwhile, the honest drivers (C, D, F,...) send messages that contain valid pseudonyms and signatures announcing an accident (sent by D) or a traffic jam (sent by C). If existing solutions are used, E can flood the uncategorized batch verification process and paralyze drivers who must discard some messages.

Our solution uses categorized batch verification. Driver Bob (B) has a Temporary List (TL) of honest drivers. We suppose that Bob's TL keeps the list of known and honest drivers like D, F,... using the property of the short-term linkability, which keeps the pseudonym T_3 unchanged for a short time. If B receives all messages, he checks the TL and collects the messages containing known T_3 to the first batch, and then B verifies them. Therefore, the warning message referencing the accident from driver D is verified in time. The messages with unknown pseudonyms like those from driver C are collected to the second batch. The potentially untrusted messages from driver E with validity W=0 are verified in the third batch only if Bob's OBU has free time and computational capacity for this. If Eve tries to replay recent valid pseudonyms together with false signatures, then the recomputed hash c'_j is not equal to received hash c_j due to time stamps in messages. For this reason, Eve is not able to mount a successful DoS attack against the batch verification of signatures.

7.2 Practical Comparison and Results

We have tested our JAVA implementation on a PC with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram, Windows 7 Professional. The Android implementation has been tested on two smartphones: Google Nexus S with CPU Cortex-A8 @ 1 GHz and 512 MB Ram, and Samsung Galaxy S3 with CPU 4xCortex-A9 @ 1.4 GHz and 1024 MB Ram.

7.2.1 Results of JAVA Implementation

We compare the signing phase of our scheme and the BBS scheme [3], which is also used in the GSIS scheme [13] & Zhang et al. scheme[27] & Ferrara et al. scheme [6] and the scheme of Wei et al. [24] (see Figures 5 and 6). The normal mode of our signing phase takes approximately 55 ms per 1 signing. This is more efficient than the compared schemes based on the BBS scheme [3] taking approx. 165 ms per 1 signing, because our solution reduced 3 pairing operations to 0 pairings for n messages in the signing phase. The initial mode of our signing phase takes approx. 165 ms due to the same number of operation as BBS scheme. This slowed mode is used in the long distance VANET applications where the privacy must be kept and the time of data processing is not critical.

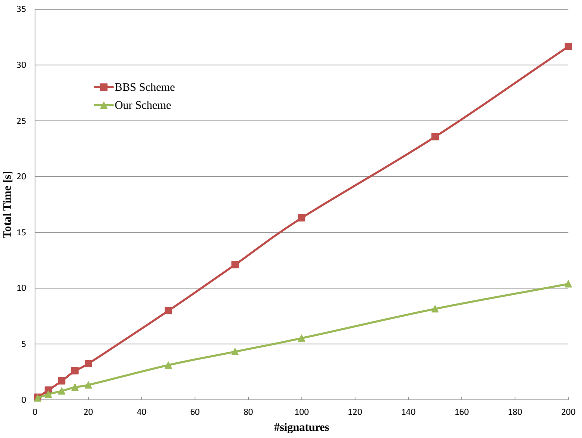

Fig. 5: The performance of Signature phase on the machine.

The performance of the Verification phase in our solution is more efficient than related BBS schemes (see Figures 7 and 8). The verification of a single signature takes approx. 207 ms using our scheme, and approx. 224 ms using related schemes based on the BBS04 scheme. Figure 8 demonstrates the efficiency of the batch verification. If the batch verification is employed, then the verification of one signature takes only approx. 50 ms on average so the batch verification of 10 signatures takes approx. 500 ms. Then, the verification of 6 signed messages takes approx. 300 ms. In the short distance VANET applications, e.g. break alerts, the vehicle controls the nearest vehicles only in the front of its direction. With the measured numbers and used hardware, our scheme can monitor and verify the 6 signed messages from 6 vehicles that are in front of the receiving vehicle. Assuming that the device that supports optimized cryptographic operations like exponentiation, multiplication and pairings is used, then our scheme is able to monitor and verify tens of cars in close distances. Moreover, due to the short-linkability, the receiver sorts out the known and potentially honest signed messages.

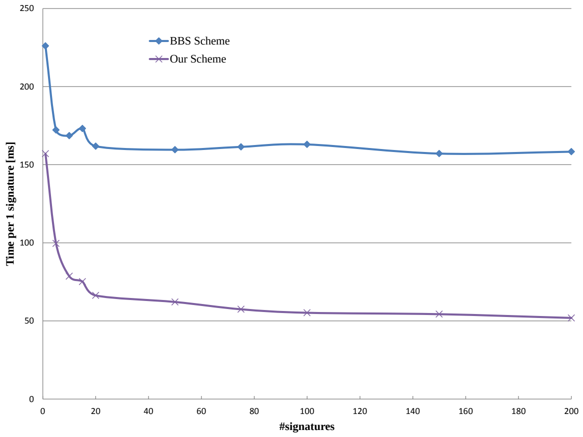

Fig. 6: The performance of Signature phase (per 1 signature) on the machine.

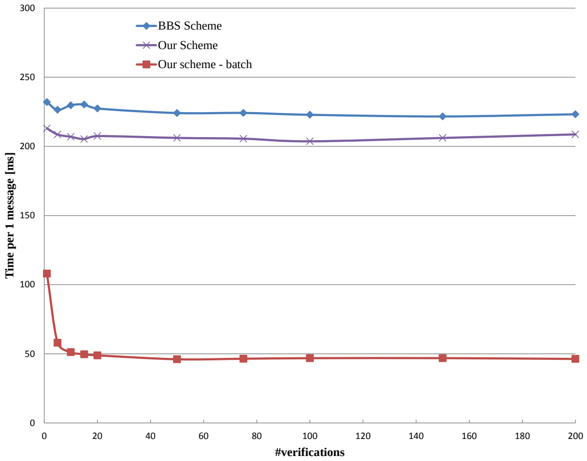

Fig. 7: The performance of Verification phase (per 1 verification) on the machine.

7.2.2 Results of Android Implementation

Moreover, we have tested our solution using two smartphones, Google Nexus S and Samsung Galaxy S3, which use the Android platform and support the jPBC Library. Figures 9, 10, 11 and 12 show their performance when signing and verifying messages. These results reflect that our scheme can effectively monitor events re-

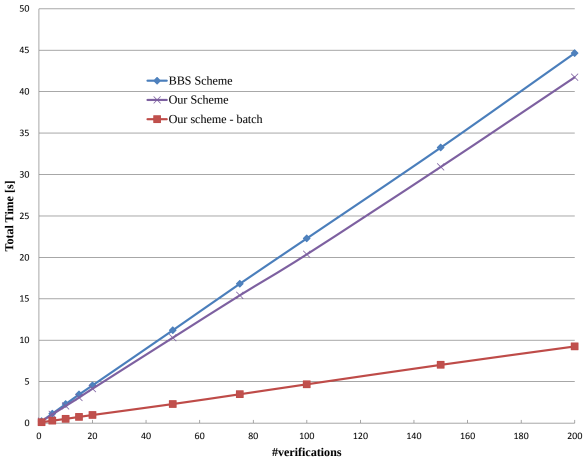

Fig. 8: The performance of Verification phase on the machine.

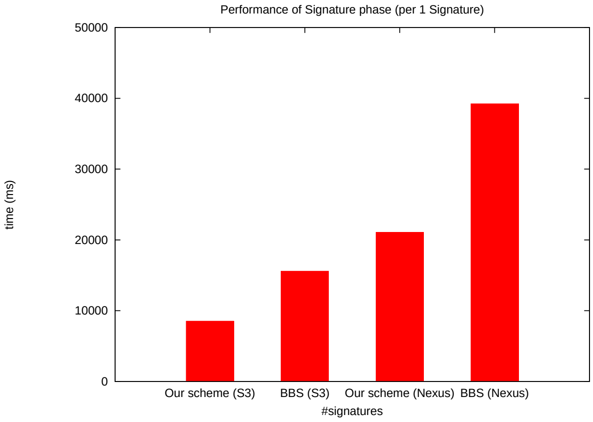

Fig. 9: The performance of Signature phase (per 1 Signature) on the smartphones.

lated to long distance VANET applications, such as traffic jams, accidents, on-road weather reports etc. Note that these messages are transmitted via V2I-I2V connection.

Furthermore, a remarkable difference can be observed between the execution time achieved by the two smartphones (Nexus S and Galaxy S3). The newer device has more computation power and, hence, it computes all the operations faster. This is especially helpful to perform the signing step in a realistic period of time and, hence, enable our proposal to be deployed in real environments. Regarding the batch verification step, although we obtain a reasonable performance at this point, we think that this step should be executed in the OBUs for practicability.

It should be stressed that the Divide-and-conquer process can be used in those cases when the batch verification fails due to the presence of a fake message.

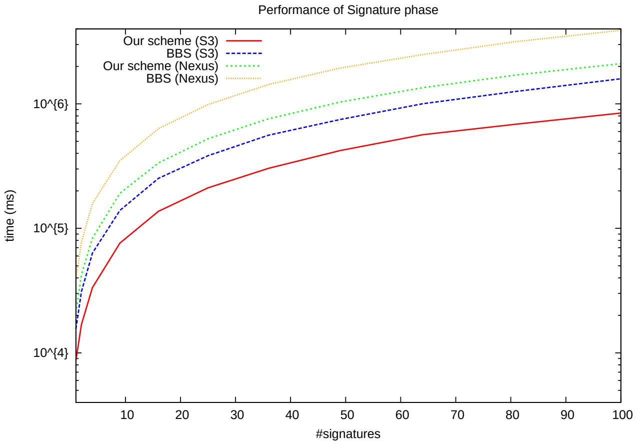

Fig. 10: The performance of Signature phase on the smartphones.

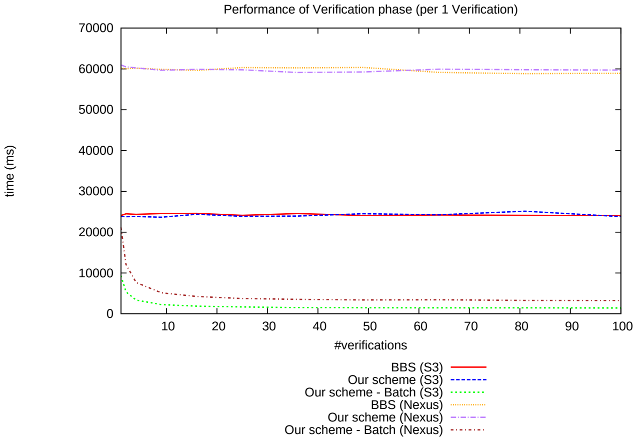

Fig. 11: The performance of Verification phase (per 1 Verification) on the smartphones.

Therefore, this algorithm can be used to split and process the messages until the fake one is found. Moreover, aggregated messages can be computed and stored to avoid recomputing them again if the verification fails. The cost of this process is logarithmic ($log_2 N$) and it still improves the cost of individual verification, which is $(N+1)/2$. Ferrara et al. show in [6] that the batch verification step of the BBS scheme can be more efficient if fake messages are less than 15 %. The categorized verification process which is proposed in our scheme minimizes the rate of fake messages in the first priority batch.

8 Conclusion

This paper presents a comprehensive security solution of vehicular networks that protects the driver's privacy. Our security solution focuses on users' privacy while

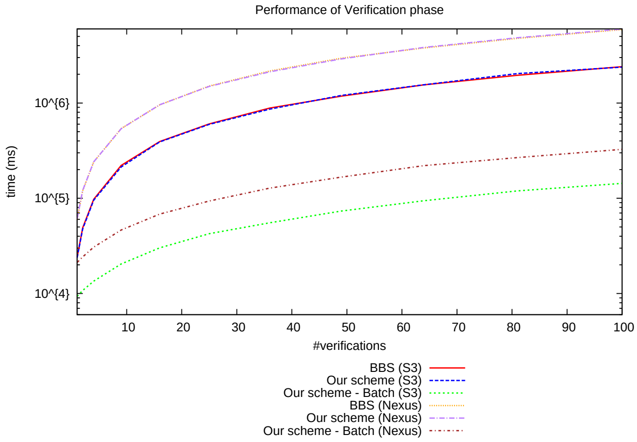

Fig. 12: The performance of Verification phase on the smartphones.

messages are transmitted between vehicles and between users and the infrastructure. We assume that the infrastructure is maintained by a group manager. Furthermore, the proposed scheme prevents the denial of service attacks, which are a current problem of many secure and privacy-preserving proposals in vehicular networks. The proposed verification is categorized. Thus, we can detect and remove some fake messages in the first stage and the second stage processes less messages.

The results of our experimental implementation on the PC point to the fact that our security solution with batch verification can be used in the short distance VANET applications which demand a fast message verification. Smartphones have lower computational power than PCs, so they could be used for processing long-distance VANET applications because a small computational delay would not cause difficulties. We assume that GM has a greater computational power than OBU or smartphones, and it can take the responsibility of verifying the signatures transmitted via V2I-I2V communication. Moreover, our solution is three times faster in signing than related schemes due to the short-term linkability. In long distance VANET application, our security solution keeps users' privacy, guaranteeing that nobody can create a profile of them.

Acknowledgements This work is partially supported by project SIX CZ.1.05/2.1.00/03.0072; the Technology Agency of the Czech Republic projects TA-02011260 and TA03010818; the Ministry of Industry and Trade of the Czech Republic project FR-TI4/647; the Spanish Ministry of Science and Innovation (through projects eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ICTW TIN2012-32757, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004 and Audit Transparency Voting Process IPT-430000-2010-31), by the Spanish Ministry of Industry, Commerce and Tourism (through projects eVerification2 TSI-020100-2011-39 and SeCloud TSI-020302-2010-153) and by the Government of Catalonia (under grant 2009 SGR 1135). Authors also wish to thank the reviewers for their useful and constructive comments. A preliminary short version of this work has been presented at the 5th International Symposium on Foundations and Practice of Security (FPS 2012) [14].

References

1. Acquisti, A.: The economics of personal data and the economics of privacy. In: texte de la conférence donnée en décembre (2010)
2. Bellare, M., Garay, J., Rabin, T.: Fast batch verification for modular exponentiation and digital signatures. Advances in CryptologyEUROCRYPT'98 pp. 236–250 (1998)
3. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Proc. Adv. Cryptology-Crypto 04, ser. LNCS 3152, pp. 41–55. Springer-Verlag (2004)
4. Chen, Y.M., Wei, Y.C.: Safeanon: a safe location privacy scheme for vehicular networks. Telecommunication Systems pp. 1–16. URL http://dx.doi.org/10.1007/s11235-010-9408-x. 10.1007/s11235-010-9408-x
5. Chim, T.W., Yiu, S.M., Hui, L.C.K., Li, V.O.K.: Specs: Secure and privacy enhancing communications schemes for vanets. Ad Hoc Networks **9**(2), 189–203 (2011)
6. Ferrara, A.L., Green, M., Hohenberger, S., Pedersen, M.Ø.: Practical short signature batch verification. In: Topics in Cryptology - The Cryptographers' Track at the RSA Conference, vol. 5473, pp. 309–324. Springer (2009)
7. Fonseca, E., Festag, A., Baldessari, R., Aguiar, R.: Support of anonymity in vanets - putting pseudonymity into practice. In: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong (2007)
8. Gerlach, M., Festag, A., Leinmuller, T., Goldacker, G., Harsch, C.: Security architecture for vehicular communication. In: The 5th International Workshop On Intelligent Transportation (2007)
9. Horng, W.B., Lee, C.P., Peng, J.W.: Privacy preservation in secure group communications for vehicular ad hoc networks. Telecommunication Systems pp. 1–11. URL http://dx.doi.org/10.1007/s11235-010-9409-9. 10.1007/s11235-010-9409-9
10. Hu, Y., Laberteaux, K.: Strong vanet security on a budget. In: Proceedings of Workshop on Embedded Security in Cars (ESCAR) (2006)
11. Hussain, R., Kim, S., Oh, H.: Towards privacy aware pseudonymless strategy for avoiding profile generation in vanet. In: H. Youm, M. Yung (eds.) Information Security Applications, *LNCS*, vol. 5932, pp. 268–280 (2009)
12. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ecdsa). International Journal of Information Security **1**(1), 36–63 (2001)
13. Lin, X., Sun, X., han Ho, P., Shen, X.: Gsis: A secure and privacy preserving protocol for vehicular communications. In: IEEE Transactions on Vehicular Technology, vol. 56, pp. 3442–3456 (2007)
14. Malina, L., Castellà-Roca, J., Vives-Guasch, A., Hajny, J.: Short-term linkable group signatures with categorized batch verification. In: Foundations and Practice of Security, pp. 244–260. Springer (2013)
15. Malina, L., Hajny, J.: Accelerated modular arithmetic for low-performance devices. In: the 34th International Conference on Telecommunications and Signal Processing (TSP), pp. 131 –135 (2011)

16. Petit, J., Mammeri, Z.: Authentication and consensus overhead in vehicular ad hoc networks. Telecommunication Systems pp. 1–14. URL http://dx.doi.org/10.1007/s11235-011-9589-y. 10.1007/s11235-011-9589-y
17. Qin, B., Wu, Q., Domingo-Ferrer, J., Zhang, L.: Preserving security and privacy in large-scale vanets. In: Proceedings of the 13th international conference on Information and communications security, ICICS'11, pp. 121–135. Springer-Verlag (2011)
18. Raya, M., Hubaux, J.P.: Securing vehicular ad hoc networks. J. Comput. Secur. **15**, 39–68 (2007)
19. Raya, M., Papadimitratos, P., Hubaux, J.P.: Securing vehicular communications. Wireless Communications, IEEE **13**(5), 8 –15 (2006). DOI 10.1109/WC-M.2006.250352
20. Studer, A., Shi, E., Bai, F., Perrig, A.: Tacking together efficient authentication, revocation, and privacy in vanets. In: SECON, pp. 1–9. IEEE (2009)
21. Tsai, H.W.: Aggregating data dissemination and discovery in vehicular ad hoc networks. Telecommunication Systems pp. 1–11. URL http://dx.doi.org/10.1007/s11235-010-9404-1. 10.1007/s11235-010-9404-1
22. Tsiounis, Y., Yung, M.: On the security of elgamal based encryption. In: Public Key Cryptography, pp. 117–134. Springer (1998)
23. Wasef, A., Shen, X.S.: Efficient group signature scheme supporting batch verification for securing vehicular networks. In: IEEE International Conference on Communications (ICC) (2010)
24. Wei, L., Liu, J., Zhu, T.: On a group signature scheme supporting batch verification for vehicular networks. In: International Conference on Multimedia Information Networking and Security, pp. 436–440. IEEE C. S., Los Alamitos, CA, USA (2011)
25. Zeadally, S., Hunt, R., Chen, Y.S., Irwin, A., Hassan, A.: Vehicular ad hoc networks (vanets): status, results, and challenges. Telecommunication Systems pp. 1–25. URL http://dx.doi.org/10.1007/s11235-010-9400-5. 10.1007/s11235-010-9400-5
26. Zhang, C., Lu, R., Lin, X., Ho, P.H., Shen, X.: An efficient identity-based batch verification scheme for vehicular sensor networks. In: INFOCOM, pp. 246–250. IEEE (2008)
27. Zhang, L., Wu, Q., Solanas, A., Domingo-Ferrer, J.: A scalable robust authentication protocol for secure vehicular communications. In: IEEE Transactions on Vehicular Technology 59(4), pp. 1606–1617 (2010)