

Harmonized Group Mix for ITS

Mirja Nitschke^a, Christian Roth^b, Christian Hoyer and Doğan Kesdoğan

University of Regensburg, Regensburg, Germany

Keywords: V2V Communication, Mix, Privacy, k-Anonymity, ITS.

Abstract: Vehicle-to-Vehicle (V2V) communication is crucial for almost all future applications in the context of smart traffic, such as autonomous driving. However, while current standards like WAVE provide a technical platform for communication and management, they lack aspects of privacy for their participants. In this paper, we introduce a Harmonized Group Mix (HGM), an architecture suited to exchange information in ITS, compatible with current standards. HGM does not rely on expensive Road-Side-Units (RSUs) or complex organizational relationships to introduce a trust anchor but is built on the concept of peer-to-peer networks. Hence, our proposal does not require any changes to current environments and is eventually easy to deploy in the real world. Our proposed method provides k-anonymity using group signatures and splits trust between multiple parties. At the same time, the integrity of the system is preserved. We evaluate our approach using the simulation framework Veins. Our experiments show that HGM is feasible from a performance and privacy perspective in the given context.

1 INTRODUCTION

In the EU and USA, the potential of Vehicle-to-Vehicle (V2V) communication has been identified years ago. With the creation of the CAR 2 CAR Communication Consortium and the Transportation Systems Committee efforts are being made to standardize communication in this field, crucial for global success. Existing standards like WAVE provide a modern platform for V2V communication allowing real time communication and thus enabling modern use cases, such as autonomous driving. Data needed for this kind of application has to be precise, up-to-date, and most importantly integer. The exact location of every user must be known at all times in order to provide a safe and secure environment. This may conflict with interests of privacy of participants. For instance, location traces are sensitive because they give insight into a user's behavior and daily routine. Current standards treat privacy interest as an orphan, hence, privacy has become an issue of interest to science and industry.

Our contribution is an architecture extension for V2V communication that provides anonymous and yet authenticated broadcast messages by using the idea of mix technology to harmonize the appearance of different messages of miscellaneous users. The

Harmonized Group Mix (HGM) is designed to be compatible to current standards. The architecture has several advantages. First, the organizational structure is simple, beside the users we only need one other entity. Second, this entity is merely semi-trusted. Third, we limit cryptographic overhead by only using group signatures. Fourth, a sender cannot easily be identified because the collaboration of several tracing authorities is needed to decrypt the group signature, providing maximum privacy. We follow the classical approach of the triple bottom line of security “algorithm, adversary and evaluation”.

The remainder of this paper is organized as follows. After a review of related work in Section 2, we discuss the system model in Section 3. We present our approach *Harmonized Group Mix (HGM)* to provide anonymous and yet authenticated broadcast Messages in the context of the ITS in Section 4. In Section 5 we evaluate the proposed approach and discuss the results. Section 6 concludes the paper and names possible extensions of our approach. A list of notation can be found in Appendix A.

2 RELATED WORK

The topic of privacy in vehicular ad-hoc networks (VANETs) has been investigated in various works,

^a  <https://orcid.org/0000-0002-2527-6340>

^b  <https://orcid.org/0000-0002-1668-5441>

for example, (Papadimitratos et al., 2007) focus on management of identities and cryptographic keys, (Sampigethaya et al., 2005) concentrate on the possibility of tracking of broadcast communications. (Calandriello et al., 2007) investigate in their architecture into pseudonym-based authentication, while (Khodaei and Papadimitratos, 2015) direct their attention towards Vehicular Public Key Infrastructure. Other interesting approaches are (Alexiou et al., 2013; Plossl et al., 2006; Laurendeau and Barbeau, 2007). (Brecht et al., 2018) propose a security credential system that uses certificates to balance privacy and anonymity. However, they rely on a complex structure of multiple organizations in contrast to our work. (Guo et al., 2007) suggest a superficial group signature based security framework in which the identity of an individual can be uncovered using only one other entity. (Verheul, 2016) use short time certificates where multiple deactivated certificates are given to a vehicle during a setup phase. Activation can be then performed once a central authority dispatches activation keys for a specific certificate, however, the vehicle must be connected to the network to receive the codes. The certificate holder can change the certificate to sign off messages at a given time to protect himself against location based attacks.

There is also a great deal of literature on architectures that enable anonymous V2V communication (Sun et al., 2007; Hao et al., 2008; Xiaonan et al., 2007; Sampigethaya et al., 2005; Hu et al., 2011; Zhang et al., 2010). However most of these papers rely on Road-Side-Units (RSU), which are either autonomous or controlled by a single trusted third party.

We focus in our work on unlinkability between messages, as previously mentioned. We try to enhance the idea of linking single messages not to a specific (pseudonymized) entity but to a group of people. Thus we combine the Mix idea with group signatures to provide location aware message while not disclosing a specific identity.

3 SYSTEM MODEL

Our system model is compatible with the typical scenario of an Intelligent Transportation System (ITS). Vehicles (i.e. users $u \in \mathcal{U}$) drive along a road network in different trajectories and eventually pass other vehicles, most of the time quite quickly. Each vehicle is inseparably equipped with communication devices called On-Board-Units (OBUs; see (Papadimitratos et al., 2007)) that enable them to exchange information in an authenticated and unforgeable way. It can be preloaded with cryptographic material.

In typical ITS scenarios, vehicles have to exchange information, thus silent periods are not acceptable and the ability to send messages must be in place at all times. Messages are broadcasted within the network and can be relayed by all participants \mathcal{U} and/or external (static) entities such as Road-Side-Units. The set \mathcal{M} describes all messages $m \in \mathcal{M}$ sent.

Each user u is uniquely identifiable via his vehicle ID $id(obu)$ which is inextricably linked to his OBU. The OBU is responsible for signing off messages, thus allowing each m to be traced back to u via $id(obu)$. The information in general is arbitrary, but is always enriched with a timestamp and a GPS location due to the nature of an ITS. Since every message is traceable to a specific location it is obvious that such messages are a threat for a participant's privacy. A trajectory of several messages traceable to a single user provides information about a user's behavior and is thus a sensitive asset that needs to be protected. Therefore, each participant has an interest in hiding their true identity in combination with the originating messages.

3.1 Group Signatures

A feasible approach to provide anonymity is the usage of group signatures, which were first presented in (Chaum and Van Heyst, 1991) and were extended by (Chen and Pedersen, 1995; Boneh et al., 2004; Lin et al., 2007; Camenisch, 1997). Group signatures allow members of a group to sign messages on behalf of the entire group. They are verifiable with a single public verification key associated to the group as a whole. Afterwards, nobody can identify the originator of that signature or link signed messages of the same originator, except for a tracer authority holding a special opening key. A second role, called group leader, distributes signing keys along all members. Furthermore, next to anonymity and traceability, group signatures provide non-frameability and unforgeability, essential building blocks for a vehicular communication protocol. In particular, we select a scheme providing the following functionality:

- Users may join a group after it has been created (dynamic groups).
- A group leader must not be the only tracer (two authorities).
- The ability to identify the originator of a message has to be split across multiple entities (distributed tracers).
- No central key distributing agency or any other similar trusted third party.

- CCA-full-anonymous (Fischlin, 2005) properties against insider attacks since every vehicle can be a potential attacker.
- Very short signatures due to limited packet size (around 1024 byte) while providing enough space left for payload.

Such a protocol can be realized using the group signature scheme by (Blömer et al., 2016) which is based on work from (Boneh et al., 2004) and extends it to provide distributed traceability. Furthermore, we adapt (Delerablée and Pointcheval, 2006) to feature dynamic groups.

3.2 Mix

A viable approach to anonymize the communication between two users is the Mix Network introduced in (Chaum, 1981). Presented simply, a Mix is a chain of nodes that a message has to pass from a user to its destination. A Mix collects several messages from different people which have been multi-layer encrypted by the originator using known asymmetric keys of all Mix nodes in the chain. Then each Mix node decrypts the respective layer of the message using its private key and thus changes the message's appearance. It now issues all messages in changed order and the next Mix node performs the same operation until the message reaches its destination. Thereby an eavesdropper can not link incoming to outgoing messages in the system due to changes in shape and order. We pick up this idea, by harmonizing the appearance of several messages from different users by using group signature encryption. Because the users act themselves as mix, see Figure 1, nobody sees the incoming messages and the mix does not have to collect various messages. Instead an attacker can only link a message to a group of users and not to a single user.

3.3 Attacker

All participants are considered to be honest-but-curious, i.e. they respect each protocol step but want to gain more knowledge about others. Given that, the attacker in our system is typically anyone who can actively interact with the system like replaying messages. Thus we require CCA full-anonymity. As a consequence each user is able to collect messages distributed in the system and store them in a message log $\hat{\mathcal{M}}$. The attacker's main goal is to link multiple messages m to a user u , however a user may use different pseudonyms π at the same time to craft messages m . In the end, the attacker can then use the log of collected messages $\hat{\mathcal{M}}$ and his knowledge about his own

behavior along with π and to build a full location trajectory of a user u .

For our setup, we assume that the attacker knows all system parameters. This results in an even more severe threat to stress our algorithm. He starts his attack to link messages to user and derive information about that participant's driven path. However, as mentioned, it is very resource intensive for an attacker to gain access to multiple OBUs, hence, we assume that he is unable to control a big part of the network. We do not focus on approaches to validate a message's content (i.e. the substance) in this work. One can, for example, rely on (Chen et al., 2011) for that. However, we provide a method to reveal the originator of a message.

3.4 WAVE

This section presents a brief derivation of requirements for our approach on the basis of the IEEE 802.11p standard and the IEEE 1609 extension. Together they are collectively called WAVE (Wireless Access for Vehicular Environments) (IEEE 1609 Working Group, 2017). In order to exchange information, participants form a WAVE Basic Service Set (WBSS, c.f. IEEE 1609.3) for the organization of our anonymity groups (see (Uzcategui et al., 2009)).

The IEEE 1609.4 standard features multichannel operation over WAVE PHY and MAC layers. It provides a control channel (CCH) and 6 service channels (SCH) usable by different applications. Each channel has different characteristics regarding maximum transmit power or frequency depending on the application requirements. Depending on the speed of a vehicle, packet loss and bandwidth may vary (Bilgin and Gungor, 2013). Due to its safety-critical property, the control channel is best suited for stable and longer range communication in the urban context (Gräfling et al., 2010). Thus, we derive the following requirements: communication exchange on SCHs should be minimal and must be completed while vehicles are in close range. The protocol shall be as efficient and robust as possible.

4 HARMONIZED GROUP MIX

We now introduce *Harmonized Group Mix* (HGM) for ITS. Our approach combines ideas from Mixes to change the shape of a message, i.e. harmonizing them, with group signatures to dynamically build new Mix groups.

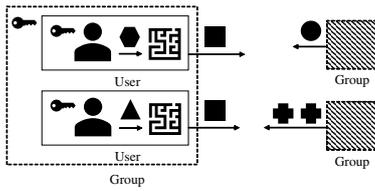


Figure 1: White- and Blackbox view on the Mix groups each producing harmonized messages unlinkable to the originator. Because every user operates his own Mix node the group does not need to communicate after exchanging the group keys.

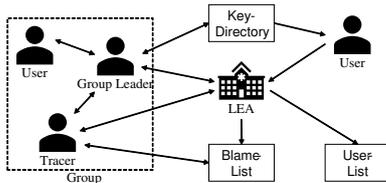


Figure 2: An organizational overview of *HGM*'s entities and roles.

4.1 High Level Overview

HGM only relies on two basic entities to balance anonymity and integrity and is therefore significantly less complex than other approaches. Next to all users, a semi-trusted third party called *Law Enforcement Agency (LEA)*, managing a userlist containing \mathcal{U} , is present. Figure 2 illustrates all entities and relationships. Each user can own multiple roles in the system.

In general, there is no need to attach personally identifiable information to a message or restrict any access to it at all. However, in real world scenarios it is likely that dishonest users try to attack the system by e.g. flooding it with bogus information. This is in particular true for the open nature of V2V architectures where dishonest users are eventually present. Thus, countermeasures to exclude such users from a system are needed; potential threats are discussed later on in section 4.4.

HGM uses separation of duties to isolate a user's activity from his identity. Messages are not directly linkable to a user u or his respective pseudonym π_i but to arbitrary groups $g_i \in \mathcal{G}$. A group is managed by another user (called group leader $l(g_i) = \pi_x$ with π_x being the group leader's pseudonym) who is responsible for creating groups or adding new members while preventing the need for a trusted third party. All messages from a group are *harmonized*, i.e. identifiable information to a user is obfuscated. Hence, every member of a group may be the originator of a message making it hard for an adversary to gain any additional information about a user. To further distribute power this obfuscation can only be removed if the majority

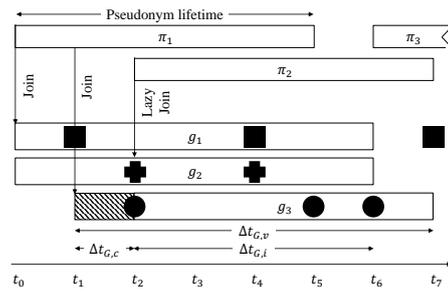


Figure 3: Process flow of *HGM*. A user holds multiple pseudonyms π which he uses to participate in different groups g . Each message (rectangle, circle, plus) is only linkable to a group but not the pseudonym. A user can select a pseudonym to join multiple groups at once even his own enhancing anonymity.

of a user group (called tracers) agrees. After sending a message, it is impossible for a receiver to tell the originator of a message since all messages have been *mixed* within the group. Messages are crafted locally in a user's domain using their private group key (see Figure 1) because only the local vehicle is assumed to be fully trusted.

In contrast to existing approaches, no silent periods are needed when changing from one π to another since a user holds multiple π s and is always member in multiple groups. Figure 3 shows an excerpt of the lifetime of a single user u . He can use every $\pi \in \Pi$ during its pseudonym lifetime to join groups at any time as long as they are accepting new users (up to $\Delta t_{G,i}$ after creation). He can then randomly use each joined group to craft messages breaking up a potential location trajectory. Groups also have a specific lifetime of $\Delta t_{G,v}$ to ensure that expired or blocked pseudonyms are not able to send messages anymore. As a consequence messages from such groups are discarded upon reception. A new group can be formed at any time but has to be completed within $\Delta t_{G,c}$.

4.2 Roles

Our approach uses different entities to separate knowledge and control. Communication between these different entities should be minimized, not only because of the performance aspect but also because of anonymity.

Law Enforcement Agency (LEA): Managed, for instance, by the government, LEA keeps track of all \mathcal{U} and acts as a doorkeeper. She is a semi-trusted party since all users trust authenticated entities from her. LEA is not considered and needed to be fully trusted, since she can only detect that users want to participate, but is unable to gain any knowledge about their activity. This is important since this entity rep-

resents a global attacker in the honest-but-curious attacker model. For example, the LEA is interested in the GPS trajectory of a specific OBU – something *HGM* targets to protect against.

Participant: A user becomes a participant when he has an entitled OBU allowing him to interact with the system. It is the most basic role in our system. Participants can use multiple signed pseudonyms π to enroll in dynamically created anonymity groups. Only participants in a group can create messages. A user's permission to be a participant is controlled by LEA.

Group Leader: Since anonymity in our system is achieved by organizing multiple and unique participants in groups, there has to be a managing entity called group leader. A group leader is responsible for managing groups including key distribution, although, he is not needed during crunch time. Furthermore, the group leader does not gain any additional knowledge about his group participants except knowing which pseudonym has joined with what role. He is unable to tell if different signed pseudonyms point to the same $id(obu)$.

Tracer: Because any group member can create valid messages on the behalf of his anonymity group, the whole group is to blame in the case of, for example, forged messages. Therefore the system allows to reveal the real identity of the originator. We split the ability to map $m \rightarrow \pi$ between multiple, randomly chosen group members, called tracers. A specific percentage of these tracers has to work together to reveal a message originator. Thus, they are accountable for the integrity of the whole group including their own messages.

4.3 Protocol

We now present an in depth look into the protocol. Communication between users is done via V2V communication channels (e.g. WAVE) while information between LEA is exchanged and directory lookups are performed out-of-band, for instance via LTE supporting higher ranges and bandwidth. In Figures 4 to 6 we denote out-of-band communication using - before the entity.

Notation. c, d denote a public private key pair used supporting encryption+decryption and signing+verifying. A group g of n members is denoted as $u_1, \dots, u_n \in g$ where g holds a subset \mathcal{U}_g of all users \mathcal{U} and all groups \mathcal{G} . $|\mathcal{U}_g|$ defines the number of members in that group. As mentioned in Section 3.1, we use group signatures by (Blömer et al., 2016) with the stated extension. For the sake of simplicity, we do not

list all parameters of that group signature scheme but only name additions needed for *HGM*. If stated, \mathcal{X} and \mathcal{Y} always denote parameters for the group signature scheme.

4.3.1 Initialization of the System

HGM provides anonymity i.a. through pseudonyms. These pseudonyms are derived from a users $id(obu)$ in a reversible way to protect integrity. However, unveiling those shall only be possible by LEA. Requesting pseudonyms may be upstreamed so that problems with unreliable long-range communication can be mitigated.

To request an authenticated pseudonym, a user sends a `PSEUDOSIGNREQUEST`(Δ) to LEA using a user generated signing payload $\Delta = \{id(obu) \parallel \{0, 1\}^k\}^{c_{LEA}}, zk$ with k random bits. LEA is then responsible for signing the Δ and thus confirming the rightful participation of the user. LEA responds with a `PSEUDOSIGNSUCCESS`($\{\Delta, ts\}^{d_{LEA}}$) using the input from the `PSEUDOSIGNREQUEST`. We call the payload of that response π created at timestamp ts . However, LEA also can reject a signing request if the user's vehicle ID is not in the user list or blacklisted due to several reasons. In that case a `PSEUDOSIGNDENIAL` is sent and the user is unable to participate.

Since our system uses the WAVE protocol, each message is broadcasted, therefore such π s are easily interceptable. Thus, a user must prove his rightful possession of that pseudonym. This is done by performing a zero knowledge proof of knowledge when showing π to other users. Public parameters zk are therefore included.

4.3.2 Managing Groups

Each participant can become a group leader once it is necessary. There are two different strategies when a member decides to offer a new group for other participants. A user may enter an area where he can admittedly connect to others but none of these users within reach intends to be a group leader. As a consequence, the new user decides to open his own group according to our protocol. In order to prevent situations where not enough groups are available, a user decides with probability ζ to open a new group.

HGM requires a setup phase for new groups according to the group signature scheme's step *Setup* (Blömer et al., 2016), however, no central key creating instance is used but specific steps are distributed. We separate the sign ability from the trace ability according to our constraints. Since this process requires all participants to be able to communicate and exchange information, data size and number of commu-

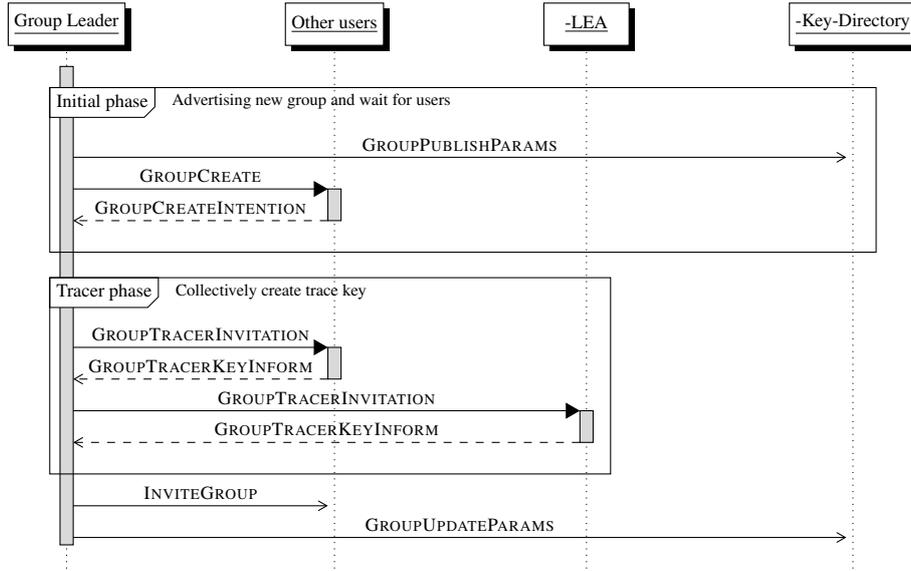


Figure 4: Sequence of GROUPCREATE.

nication is minimized. The setup phase has two steps (cf. Figure 4) but has to be completed within $\Delta t_{G,c}$ seconds.

Firstly, the group leader signals, during the initial phase, to create a new group g_k by updating $\text{GROUPPUBLISHPARAMS}$ the central key directory with required information of his group ($(id(g_k), ts, \mathcal{X})$ with ts being the creating timestamp) followed by a GROUPCREATE broadcast, where he invites users to join his group. Hence, he broadcasts a $\text{GROUPCREATE}(\pi, id(g_k), SCH_RC, ts_{exp})$ message on the Control Channel (using an unused $\pi \in \Pi$ and a group id $id(g_k) \equiv \mathcal{H}(\pi \parallel \{0, 1\}^k)$). In our scheme, each group is organized in a WBSS (Wave Basic Service Set) using one of the multiple channels (the selected one is SCH_RC) to minimize packet collisions across different groups. At least θ^P users need to state their intention to join ($\text{GROUPCREATEINTENTION}$). Otherwise the group setup phase may fail due to a timeout. This is a precaution because during the setup phase no user is able to send any payload messages (using this group) and is only passively waiting for response from $l(g_k)$.

Assuming θ^P members are found, the group leader starts the tracer phase by selecting $\theta^T - 1 < \theta^P$ users using a direct message $\text{GROUPTRACERINVITATION}$. Hopefully, $\theta^T - 1$ members respond with $\text{GROUPTRACERKEYINFORM}$ providing each information about their secret key part $sk_{0, \dots, \theta^T - 1}$, the group leader requests the remaining tracer part sk_i from LEA as well. She always responds. Eventually, the group leader can generate the verification key VK and the public key PK (see (Blömer et al., 2016)).

The setup is now complete and the group is ready to accept members ($\mathcal{U}_{g_k} = \emptyset$), broadcasting $\text{INVITEGROUP}(\pi, id(g_k))$ for $\Delta t_{G,i}$ seconds. At the same time, he deposits the membership list, public group key and verification key in an online directory for others to download once they want to verify a signature (GROUPUPDATEPARAMS).

4.3.3 Joining an Existing Group

A participant u_i uses one $\pi_j \in \Pi_{u_i}$ to join a group g_k (equals *Join* from (Blömer et al., 2016) with the extension of (Delerablée and Pointcheval, 2006)). In order to find a group he observes the Control Channel for INVITEGROUP messages sent by $l(g_k)$, afterwards he checks the key directory, which it can cache locally, for all required information. (cf. Figure 5) Once a user finds such a message indicating that a group leader accepts new members, a user ask in a two way phase to join the group using $\text{GROUPSETUPJOIN1,2}(\pi_j, \mathcal{X}, id(g_k))$. A successful join finishes once the group leader answers with a $\text{GROUPSETUPACCEPT1,2}(id(g_k), \mathcal{J})$ and executes $\mathcal{U}_{g_k} \leftarrow u_i$.

A group leader is not able to tell if two distinct pseudonyms point to the same user. Therefore, a group leader only checks the received pseudonym π_j for a correct signature from the LEA. He also uses π 's included timestamp ts to enforce particular joining constrains.

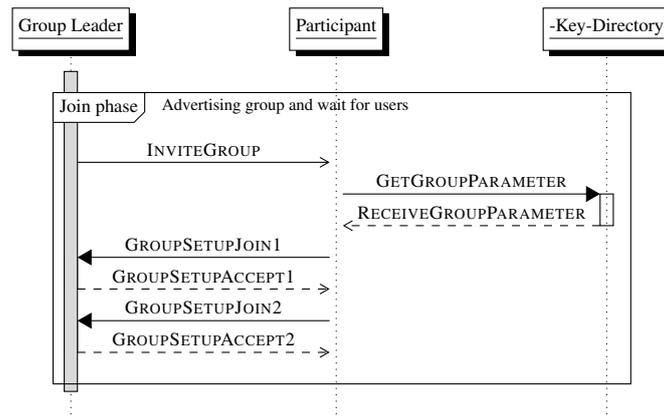


Figure 5: Sequence of GROUPJOIN.

4.3.4 Sending Messages

A user u_i can send messages at any given time, but usually every $\Delta t_{M,b}$ seconds, as long as $u_i \in \mathcal{U}_{g_k}$ (equals $Sign$ from (Blömer et al., 2016)). They then send messages on behalf of that group. Consequently, if a user wants to send some payload \square , he uses his private part of the group key to craft a message $m = (\square, ts, id(g_k), \sigma)$. σ is the signature of the message m . To save space, $\mathcal{H}(\sigma)$ is the message ID. Also, $id(g_k)$ is included in the message. This fulfills two purposes. Firstly, a user receiving a message can validate the signature by just acquiring the group’s public key from the online directory and then use this key to verify the signature. Secondly, $id(g_k)$ is used for the blame process explained in the next section.

By design, a user should be a member of different groups at the same time. Therefore he can send messages using any of his private group keys. This allows him to become untraceable by interrupting any message trajectories. As a note, any other identifying information (e.g. MAC address) must be changed.

Messages are relayed by other members receiving them. However, they always verify a message’s signature first¹ (using *SignatureVerify* from (Blömer et al., 2016)) before they resend the message without changing it. Furthermore, users do not relay messages if the payload’s geographical location is too far away from the user’s current position. This is done to reduce load on the network. Also, messages are discarded if they are too old.

4.3.5 Blaming a Message

Our system allows users to send any payload (including sensitive GPS-coordinates) without revealing the

¹Required information can be looked up in the key directory and be cached locally.

identity, yet all messages are authenticated making *HGM* an integer system. Keeping that state is done by a deliberate backdoor. A user who suspects a message m from group g_k can start a blame process (cf. Figure 6) which potentially reveal m ’s originator (i.e. find $m \rightarrow \pi$). This can only be done with the support of the tracers of the message’s group. LEA itself is unable to find out any more information about the message conforming our requirements. We recall a message $m = (\square, ts, id(g_k), \sigma)$ holds all the required information, and in particular the group ID $id(g_k)$.

Blaming one m requires a valid π . Any user owning π_b sends a *MESSAGEBLAMEREQ*(m, π_b) to LEA. She validates the blame request by extracting the $id(obu)_b$ from the pseudonym π_b using her private key d_{LEA} . LEA keeps a record of all blame request for each $id(obu)_b$ to prevent abusive use of blaming. A unsuccessful blame request (i.e. a blame request denied by tracers) increases $id(obu)_b$ ’s penalty score on the userlist.

LEA posts a valid *MESSAGEBLAMEREVEAL*(m) to a blame list. For maximum transparency, a blockchain based list can be used. Every tracer has subscribed to that list and listens for changes (*MESSAGEBLAMEUPDATE*). If a new entry is added, each tracer first checks if he was tracer for that specific group using the $id(g_k)$ field. If so, he decides on its own if he wants to use his part of the tracer key to reveal his part of the user’s real identity (*ShareOpen* from (Blömer et al., 2016)). The process has to be completed within $\Delta t_{G,b}$, otherwise fails.

- *MESSAGEBLAMEACCEPT*: A tracer also doubts that the message’s payload is true because, e.g. he has seen otherwise or he tries to compromise a user.
- *MESSAGEBLAMEDENIAL*: A confirmation of the payload ultimately leads to this behavior of a tracer, but it is also possible that an adverse tracer

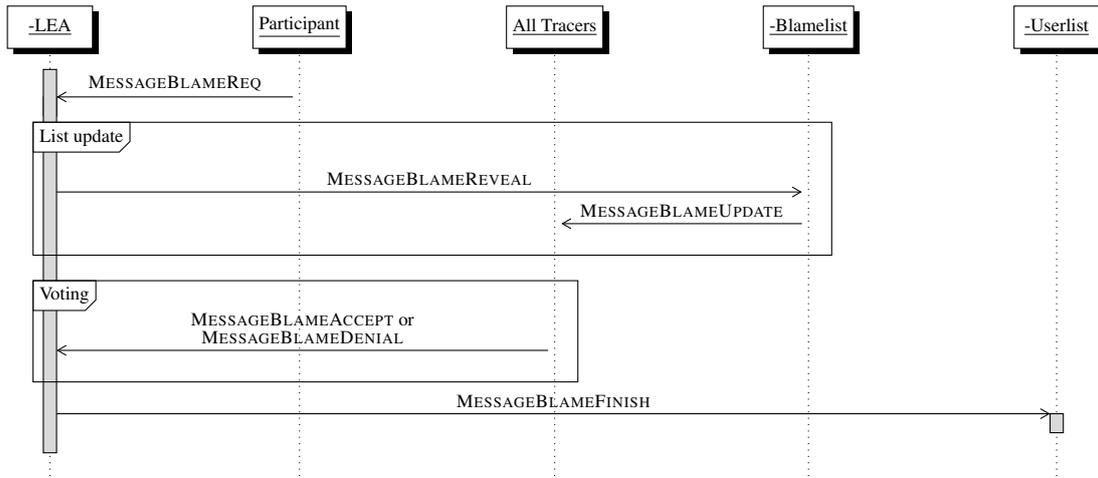


Figure 6: Sequence of MESSAGEBLAME.

is shielding the originator.

Since we are in a highly dynamic network it may occur that a proportion of former tracers leaves the network or does not answer the blame request. Therefore, *HGM* does not require all θ^T tracers to accept an blame request. Instead only $\lambda * \theta^T < \theta^P$ tracers need to accept a request. This is sufficient to reveal the real identity (i.e. the π used in that group) of the message originator. Once, λ percent of $\mathcal{T}(g_k)$ cooperated to reveal the π , LEA can perform *ShareVerify* in combination with *ShareCombine* from (Blömer et al., 2016) and then extract the $id(obi)$ and add a penalty point to it which may lead to putting it on the blacklist if a penalty point threshold is exceeded (MESSAGEBLAMEFINISH).

4.4 Security & Anonymity

LEA is responsible for authenticating the pseudonyms of the specific users. Only she can match different pseudonyms to single users. Thus, she is able to limit the amount of pseudonyms a user can have at the same time and is essential for the protection against sybil attacks. She sends a PSEUDOSIGNDENIAL once a user already exceeds the limit of valid pseudonyms allowed at a particular time. A pseudonym is considered valid as long it is not expired. LEA is unable to tell if a pseudonym is in active use, therefore each π given to a user counts towards a limit θ^π .

HGM is build on the idea that people hold multiple pseudonyms at once which sounds counterintuitive at first. Table 1 discusses how *HGM* protects against multiple attacks.

It is also important to consider the ratio of θ^T and θ^P , i.e. the minimum number of tracers and users

during group setup phase (and in particular GROUP-TRACERKEYNEGOTIATION). If $\theta^T = \theta^P$ then the group leader is forced to select every member of the group as a tracer resulting in predictability to become a tracer. An adversary user may circle the group leader with all of his θ^π pseudonyms knowing that he will later hold θ^π parts of the tracer key. Therefore, it is recommended to set $\theta^T \ll \theta^P$.

5 EVALUATION

We now evaluate *HGM* using a state-of-the-art vehicular network simulation framework called Veins (Sommer et al., 2011). Veins has support for IEEE 802.11p and WAVE module. In the following sections, we will describe the setup and parametrization of our simulation model. We then present our results and show that *HGM* is feasible from a privacy and performance perspective.

An urban area around the city of Regensburg, Germany (about 5 km^2 in size) was extracted from OpenStreetMap and prepared using *netconvert*, *randomTrips*, and *polyconvert* for SUMO (Krajzewicz et al., 2012) and Veins. The chosen area includes all the different aspects of an urban area. To present realistic results, we use SimpleObstacleShadowing included in Veins to simulate radio interference due to obstacles. For the sake of simplicity, we disabled WAVE channel switching in our simulation, although, *HGM* is designed to work with different channels.

A number of parameters influence the system. Relevant parameters for simulation and evaluation are $\theta^\pi = 3$, $\theta^P = 5$, $\theta^T = 3$, $\rho_{U,b} = 0$, $\rho_{G,o} = 0$, $\rho_{G,j} = 1$, $\Delta t_{G,c} = 20\text{ s}$, $\Delta t_{G,v} = 80\text{ s}$, $\Delta t_{G,i} = 60\text{ s}$, and $\Delta t_{M,b} = 1\text{ s}$.

Table 1: List of security threats and how *HGM* protects against them.

	Role		
	<i>Participant</i>	<i>Group Leader</i>	<i>Tracer</i>
<i>Alter</i>	σ prevents to change \square or τ s. Pointing to another group by changing $id(g)$ points to other group key.	Forging any key parameters to track an individual during group join is easily detectable by a user via the public group key directory which shows the same attributes for each group.	Altering any MESSAGEBLAMEREQ is not possible since these requests are always initiated by LEA.
<i>Suppress</i>	No effect since messages are relayed by all participants (cf. Mesh networks).	Group creation or join fails automatically once a timeout is exceeded.	Not responding to a blame MESSAGEBLAMEREQ is implicitly a MESSAGEBLAMEDENIAL.
Attack <i>Replay</i>	Intended since each message is unique via its $\mathcal{H}(\sigma)$ and <i>HGM</i> uses relaying techniques.	Using another user's π to join another group is not possible because of a zero knowledge proof. Resending key parameters during create or join is impossible since recipient may not know the respective secrets.	Each tracer only has one vote (either MESSAGEBLAMEACCEPT or MESSAGEBLAMEDENIAL), successive responses may be discarded.
<i>Inject</i>	Blame of bogus messages will eventually remove them.	Group leader can violate θ^P and/or θ^T . He can either track a user by performing an $n - 1$ attack or create a group without LEA as a tracer or any other independent tracers. As a consequence each blame request will eventually fail. Such a rogue group can be easily identified by LEA with the group leader being ultimately responsible for it.	Sending either MESSAGEBLAMEACCEPT or MESSAGEBLAMEDENIAL without a proper MESSAGEBLAMEREQ has no effect.

If not otherwise stated, the defaults are used. To ensure that enough vehicles are in the simulation, a warm-up period of 10min is preceded. Key statistics are tracked after the warm-up period for a duration of another 10min. Every second a vehicle spawns and drives along a random path. To ensure the accuracy of the results, every parameter configuration is run at least five times and afterwards the mean of the results is calculated.

5.1 Privacy

We measure privacy using two metrics mentioned by (Wagner and Eckhoff, 2018). Of interest is the *anonymity set size*, which describes "the set of users that the adversary cannot distinguish from u " (Wagner and Eckhoff, 2018). Furthermore, the *size of uncertainty region* illustrates "the minimal size of the region R_U to which an adversary can narrow down the position of a target user u " (Wagner and Eckhoff, 2018).

In theory, the *anonymity set size* is at least θ^P . Therefore, it would be wise to choose θ^P as big as

possible. However, from a performance perspective, the upper bound of θ^P is limited since larger groups are more unlikely to be successfully created. Thus, our algorithm allows subsequent group joins, subsequently increasing the anonymity set size. The simulation helps to balance these conflicting interests. To extract feasible values, we track the sender of a payload message (called safety beacons in the simulation) and his group affiliation. On that, we derive the ratio of messages sent by a user in comparison to all messages sent by the affiliated group called multitude of messages. Also, we count the average number of users within each group ($|\mathcal{U}_g|$). As Figure 7 illustrates, there is a direct correlation between $\Delta t_{M,b}$ and these parameters indicating that low values for $\Delta t_{M,b}$ are to prefer.

To gain insight into the *size of uncertainty region*, we collect payload messages sent by users \mathcal{U}_{g_i} of a group g_i . With the help of the Gauss's area formula, we calculate the region span of \mathcal{U}_{g_i} which sent a safety beacon message at the same time. However, this yields just the minimum of the uncertainty region because not all $u \in \mathcal{U}_{g_i}$ may have send a safety bea-

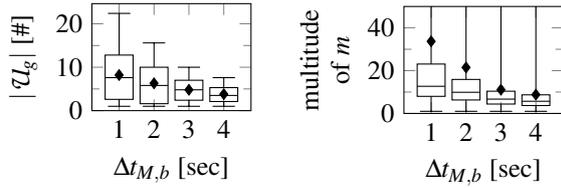


Figure 7: Anonymity set size defined by the user who send a safety beacon message under the same group ID.

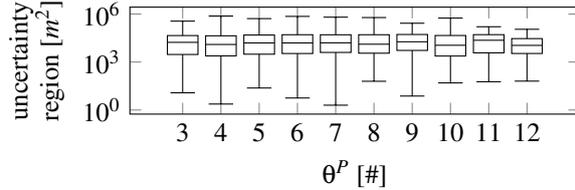


Figure 8: Boxplot of the uncertainty region.

con message using that specific group g_i . Figure 8 shows that we can choose θ^P small because the uncertainty region stays more or less constant with a decrease of θ^P .

Figure 9 plots the influence of parameters $\Delta t_{G,c}$, $\Delta t_{G,i}$, and $\Delta t_{G,v}$, on the percentage of the time² a user is able to sent messages on behalf of a group and the uncertainty region. We can see that there is no correlation between the uncertainty region and the percentage of active time. To determine the parameters which have an impact on the uncertainty region, we calculated the correlation coefficient between all relevant parameters in our simulation and the uncertainty region. We reason that none of the considered parameters have a nonambiguous effect on the uncertainty region. The movement of vehicles may have a more significant impact than the presented parameters.

5.2 Performance

To depict the overhead of *HGM*, we list in Table 2 the distribution of messages types in relation to $\Delta t_{M,b}$. Most received messages are payload messages. With bigger $\Delta t_{M,b}$ decreases their percentage because the group setup phases take more time and the users can send fewer payload messages.

Figure 10 shows the time for group creation Δt in relation to different parameters. It shows that increasing $\Delta t_{M,b}$, $\Delta t_{G,c}$, or θ^P also increases Δt . However, this does not come as a surprise, since for instance searching for more members depends on the density of cars which was constant during our simulation.

Figure 11 illustrates *HGM*'s robustness because even when 90% of the vehicles in the simulation are

²In relation to the user's overall simulation lifetime.

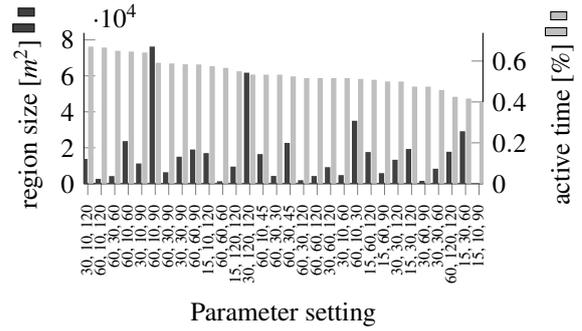


Figure 9: The influence of $\Delta t_{G,c}$, $\Delta t_{G,i}$, $\Delta t_{G,v}$ on the percentage of the users active group time, and the uncertainty region in m^2 .

Table 2: Distribution of messages types in relation to $\Delta t_{M,b}$.

		$\Delta t_{M,b}$			
		1 [%]	2 [%]	3 [%]	4 [%]
received	payload	39.28	50.87	43.18	29.26
	broadcast	43.97	34.73	37.76	45.19
	direct	0.99	0.87	0.71	0.49
send	payload	3.70	3.57	2.32	1.04
	broadcast	1.26	0.97	0.99	0.99
	LEA	6.71	4.49	8.05	12.68

blocked, groups with $\theta^P = 5$ are created and on average at least 5 participants send a message under the same $id(g)$.

6 CONCLUSION AND FUTURE WORK

In this paper we presented a privacy enhanced approach for V2V communication which can be used with current standards like WAVE. *HGM* uses a state-of-the-art group signature scheme feasible for the tight requirements of ITSs. It only relies on a semi-trusted entity called LEA and apart from that, is fully

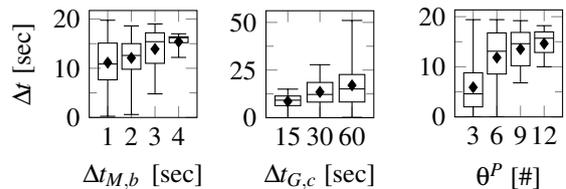


Figure 10: Time for group creation Δt is correlated to multiple parameters.

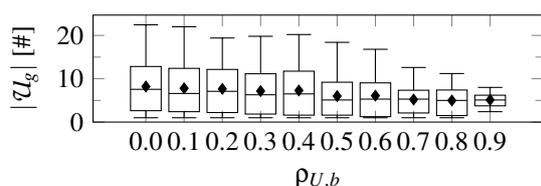


Figure 11: The anonymity set size in dependence of the probability of blocked users.

decentralized. We tackle to balance both requirements, privacy and integrity.

Our evaluation was done with Veins showing that *HGM* improves the privacy of users by hiding them along with other members in regions with uniform probability. Among other things, the results also show that our approach is robust in terms of dishonest participants who suppress messages. We also show that the overhead related to *HGM* is acceptable.

In future work we would like to extend our analysis to a more realistic scenario such as (Codeca et al., 2016). We want to increase performance using WAVE's channel switching in the simulation. In addition, we want to strengthen our attacker by including external knowledge such as derived user behavior and analyze tracers.

ACKNOWLEDGEMENTS

This research was partly funded by the German Federal Ministry of Education and Research (BMBWF) with grant number: 16KIS0367K.

REFERENCES

- Alexiou, N., Laganà, M., Gisdakis, S., Khodaei, M., and Papadimitratos, P. (2013). VeSPA. In *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy - HotWiSec '13*, page 19, New York, New York, USA. ACM Press.
- Bilgin, B. E. and Gungor, V. C. (2013). Performance Comparison of IEEE 802.11p and IEEE 802.11b for Vehicle-to-Vehicle Communications in Highway, Rural, and Urban Areas. *International Journal of Vehicular Technology*, 2013:1–10.
- Blömer, J., Juhnke, J., and Löken, N. (2016). Short group signatures with distributed traceability. In *Mathematical Aspects of Computer and Information Sciences. MACIS 2015. Lecture Notes in Computer Science*, volume 9582, pages 166–180. Springer, Cham.
- Boneh, D., Boyen, X., and Shacham, H. (2004). Short group signatures. In Franklin, M., editor, *Advances in Cryptology – CRYPTO 2004*, pages 41–55, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Brecht, B., Theriault, D., Weimerskirch, A., Whyte, W., Kumar, V., Hehn, T., and Goudy, R. (2018). A security credential management system for V2X communications. *CoRR*, abs/1802.05323.
- Calandriello, G., Papadimitratos, P., Hubaux, J.-P., and Lioy, A. (2007). Efficient and robust pseudonymous authentication in VANET. In *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks, VANET '07*, pages 19–28, New York, NY, USA. ACM.
- Camenisch, J. (1997). Efficient and generalized group signatures. In *Advances in Cryptology – EUROCRYPT '97. EUROCRYPT 1997. Lecture Notes in Computer Science*, volume 1233, pages 465–479. Springer, Berlin, Heidelberg.
- Chaum, D. and Van Heyst, E. (1991). Group signatures. In *Advances in Cryptology – EUROCRYPT '91. EUROCRYPT 1991. Lecture Notes in Computer Science*, volume 547 LNCS, pages 257–265. Springer, Berlin, Heidelberg.
- Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90.
- Chen, L., Ng, S.-L., and Wang, G. (2011). Threshold anonymous announcement in vanets. *IEEE Journal on Selected Areas in Communications*, 29:605–615.
- Chen, L. and Pedersen, T. P. (1995). New group signature schemes. In De Santis, A., editor, *Advances in Cryptology – EUROCRYPT '94*, pages 171–181. Springer, Berlin, Heidelberg.
- Codeca, L., Frank, R., and Engel, T. (2016). Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research. In *IEEE Vehicular Networking Conference, VNC*, volume 2016-January, pages 1–8. IEEE Computer Society.
- Delerablée, C. and Pointcheval, D. (2006). Dynamic Fully Anonymous Short Group Signatures. In *VIETCRYPT 2006*, pages 193–210.
- Fischlin, M. (2005). Communication-Efficient Non-interactive Proofs of Knowledge with Online Extractors. In Shoup, V., editor, *Advances in Cryptology – CRYPTO 2005*, pages 152–168. Springer Berlin Heidelberg.
- Gräffling, S., Mähönen, P., and Riihijärvi, J. (2010). Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications. In *ICUFN 2010 - 2nd International Conference on Ubiquitous and Future Networks*, pages 344–348.
- Guo, J., Baugh, J. P., and Wang, S. (2007). A group signature based secure and privacy-preserving vehicular communication framework. In *2007 Mobile Networking for Vehicular Environments*, pages 103–108. IEEE.
- Hao, Y., Cheng, Y., and Ren, K. (2008). Distributed key management with protection against RSU compromise in group signature based VANETs. In *GLOBECOM - IEEE Global Telecommunications Conference*, pages 4951–4955. IEEE.
- Hu, W., Xue, K., Hong, P., and Wu, C. (2011). Atcs: A novel anonymous and traceable communication scheme for vehicular Ad hoc networks. *International Journal of Network Security*, 13(2):71–78.

- IEEE 1609 Working Group (2017). IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. *IEEE Std 1609.0-2013*, pages 1–78.
- Khodaei, M. and Papadimitratos, P. (2015). The Key to Intelligent Transportation: Identity and Credential Management in Vehicular Communication Systems. *IEEE Vehicular Technology Magazine*, 10(4):63–69.
- Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.
- Laurendeau, C. and Barbeau, M. (2007). Secure Anonymous Broadcasting in Vehicular Networks. In *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, pages 661–668. IEEE.
- Lin, X., Sun, X., Ho, P. H., and Shen, X. (2007). GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on Vehicular Technology*, 56(6 D):3442–3456.
- Papadimitratos, P., Buttyan, L., Hubaux, J.-P., Kargl, F., Kung, A., and Raya, M. (2007). Architecture for Secure and Private Vehicular Communications. In *2007 7th International Conference on ITS Telecommunications*, pages 1–6. IEEE.
- Plossl, K., Nowey, T., and Mletzko, C. (2006). Towards a security architecture for vehicular ad hoc networks. In *First International Conference on Availability, Reliability and Security (ARES'06)*, pages 8 pp.–381. IEEE.
- Sampigethaya, K., Huang, L., Li, M., Poovendran, Radha Matsuura, K., and Sezaki, K. (2005). CARAVAN: Providing location privacy for VANET. *Embedded Security in Cars*, pages 1–15.
- Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15.
- Sun, J., Zhang, C., and Fang, Y. (2007). An ID-based framework achieving privacy and non-repudiation in vehicular ad hoc networks. In *Proceedings - IEEE Military Communications Conference MILCOM*, pages 1–7. IEEE.
- Uzcategui, R. A., De Sucre, A. J., and Acosta-Marum, G. (2009). Wave: A tutorial. *IEEE Communications Magazine*, 47(5):126–133.
- Verheul, E. R. (2016). Activate later certificates for v2x - combining its efficiency with privacy. *IACR Cryptology ePrint Archive*, 2016:1158.
- Wagner, I. and Eckhoff, D. (2018). Technical Privacy Metrics. *ACM Computing Surveys*, 51(3):1–38.
- Xiaonan, L., Zhiyi, F., and Lijun, S. (2007). Securing vehicular ad hoc networks. In Ning, P. and Du, W., editors, *2007 2nd International Conference on Pervasive Computing and Applications, ICPCA'07*, volume 15, pages 424–429. IOS Press.
- Zhang, L., Wu, Q., Solanas, A., and Domingo-Ferrer, J. (2010). A scalable robust authentication protocol for secure vehicular communications. *IEEE Transactions on Vehicular Technology*, 59(4):1606–1617.

APPENDIX

A Notation

u	Known user in the system
\mathcal{U}	Set of all users
g	Existing group in the system
$\hat{\mathcal{G}}$	Set of all groups with $id(g)$
\mathcal{U}_g	Users in group g
l	Group leader
t	Tracer
\mathcal{T}	Set of tracers
π	Pseudonym signed by LEA
Π	Set of all signed pseudonyms
\mathcal{H}	Hash function
m	Message
\mathcal{M}	Set of all messages
σ	Signature of a message m
zk	Public parameters of a zero knowledge proof
θ^T	Minimum number of tracers
θ^P	Minimum number of group users
θ^π	Maximum number of concurrent pseudonyms of a single user
λ	Proportion of tracers needed to reveal
k	Length of randomness
α	Percentage of dishonest users in the system
$\Delta t_{G,c}$	Group creation time
$\Delta t_{G,v}$	Group valid time
$\Delta t_{G,i}$	Group invitation time
$\Delta t_{G,b}$	Blame duration time
$\Delta t_{M,b}$	Message safety beacon time interval
$\rho_{G,o}$	Probability of opening a group
$\rho_{G,j}$	Probability of joining a group
$\rho_{U,b}$	Probability of being blocked