Road-to-Vehicle Communications with Time-Dependent Anonymity: A Light Weight Construction and its Experimental Results

Keita Emura* Takuya Hayashi*

April 9, 2015

Abstract

This paper describes techniques that enable vehicles to collect local information (such as road conditions and traffic information) and report it via road-to-vehicle communications. To exclude malicious data, the collected information is signed by each vehicle. In this communications system, the location privacy of vehicles must be maintained. However, simultaneously linkable information (such as travel routes) is also important. That is, no such linkable information can be collected when full anonymity is guaranteed through the use of cryptographic tools such as group signatures. Similarly, continuous linkability (via pseudonyms, for example) may also cause problem from the viewpoint of privacy.

In this paper, we propose a road-to-vehicle communication system with relaxed anonymity by considering time-dependent linking properties via group signatures with time-token dependent linking (GS-TDL). These techniques are used to construct an anonymous time-dependent authentication system via GS-TDL. Briefly, a vehicle is unlinkable unless it generates multiple signatures at the same time period.

In addition, we describe vulnerability in the anonymous authentication system proposed by Wu, Domingo-Ferrer and González-Nicolás (IEEE T. Vehicular Technology 2010), where an unauthorized individual can create a valid group signature without using signing key. Moreover, our GS-TDL scheme supports verifier-local revocation (VLR), which maintains constant signing and verification costs by using the linkable part of signatures. These appear to be related to independent interests.

Finally, we provide our experimental results (using the TEPLA library) and confirm that our system is feasible in practice.

1 Introduction

Location privacy is widely recognized as an important issue, especially in the case of motor vehicles. For example, Rouf et al. [57] mentioned that location privacy could be compromised via a tire pressure monitoring system, and Xu et al. [70] proposed a secure communication protocol as a countermeasure against this vulnerability. However, local information is important and useful, e.g., information about traffic and road conditions is highly indispensable for maintaining urban operations. Therefore, collecting local information without infringing privacy is an important issue that requires a resolution. For example, consider a situation in which vehicles collect such local information and report it via road-to-vehicle communications. To exclude malicious data, this collected information must be signed by each vehicle. In this case, it seems undesirable, from the viewpoint of privacy, to link location information (obtained from collected information and signatures) to a particular person. To establish road-to-vehicle communication systems with effective privacy safeguards, vehicle ad-hoc network (VANET) systems with anonymity were proposed. These systems enhance privacy using various cryptographic tools, such as applying group signatures [23] or ring signatures [56]; other similar systems have been proposed in numerous studies [24, 36, 67, 55, 46, 61, 62, 69, 68].

*National Institute of Information and Communications Technology (NICT), 4-2-1, Nukui-kitamachi, Koganei, Tokyo, 184-8795, Japan. {k-emura,takuya.hayashi}@nict.go.jp

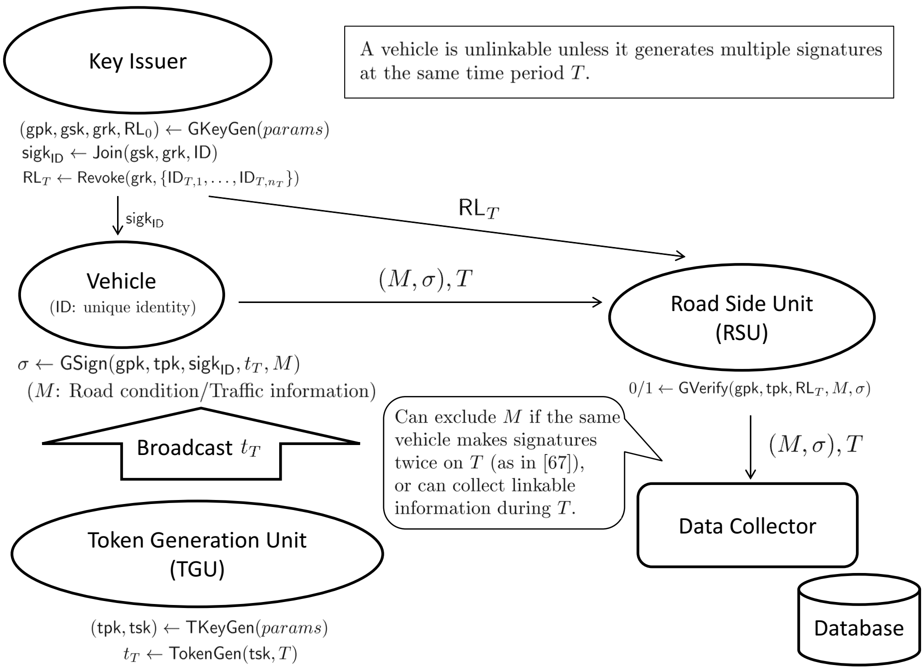

Figure 1: Brief Description of Our Road-to-Vehicle Communication System Based on GS-TDL

In particular, Wu, Domingo-Ferrer and González-Nicolás [67] proposed an anonymous threshold authentication scheme, in which messages are accepted when more than t vehicles send the same message (signed by the vehicles). The core cryptographic technique in their system utilizes message-linkable group signature (MLGS), where two group signatures become linkable if a signer generates a group signature for the same message twice. Thus, double voting is protected even in an anonymous environment.

When defining secure road-to-vehicle systems, it should be decided whether full anonymity (or more precisely, unlinkability) must be guaranteed or not when local information is collected, since simultaneously linkable information (such as travel routes) is also important. That is, no such linkable information can be collected when full anonymity is guaranteed through the use of cryptographic tools such as group signatures. For example, even if linkable information (such as travel routes) would like to be collected, no such linkable information can be collected when full anonymity, where it is not possible to distinguish whether the same signer generated the two signatures or not, is guaranteed. Moreover, system efficiency may be drastically improved if unlinkability is not required, because the theoretical gap between a group signature with unlinkability and one without unlinkability is significantly large.[1] For example, Baldimtsi and Lysyanskaya proposed a lightweight version of anonymous credentials [11], where no pairing computation is required under a relaxed anonymity definition. Conversely, continuous linkability (via pseudonyms, for example) is problematic from the viewpoint of privacy. For example, a vehicle with continuous linkability is tracked from the time the vehicle is acquired up until the time it is sold, and parking spaces, which may include the driver's home and work place, are revealed. Therefore, suitably defining "moderate" anonymity with practical efficiency in a road-to-vehicle communications context is an important issue that must be resolved.

Our Contribution: In this paper, we propose a road-to-vehicle communication system with relaxed anonymity by considering time-dependent linking properties. We assume that a Token Generation Unit (TGU) generates and broadcasts a time-dependent token. In our system, a vehicle computes signatures using its signing key and a time-dependent token generated by the TGU. As a result, our system guarantees

[1] Group signature without unlinkability can be constructed from one-way functions, whereas group signature with unlinkability implies chosen-ciphertext secure public key encryption [53].

that: *A vehicle is unlinkable unless it generates multiple signatures at the same time period T.* Our system's core cryptographic tools utilize group signatures with time-token dependent linking (GS-TDL), which we propose in this paper. In GS-TDL, nobody can distinguish whether two signatures were generated by the same signer or not if $(\mathsf{ID}, T) \neq (\mathsf{ID}', T')$ for identities ID and ID' and time periods T and T'. Moreover, our GS-TDL supports verifier-local revocation (VLR) [44, 40, 52, 51, 20], where no signer is involved in the revocation procedure. In particular, our GS-TDL achieves backward unlinkability [51, 52], which prevents adversaries from breaking anonymity, even after the challenge users are revoked. Our time-dependent linking properties enable us to achieve *constant verification costs*, whereas those of previous schemes are $O(r)$, where r is the number of revoked users. This appear to be related to independent interest. Briefly, TGU generates a time-dependent token t_T at a time T, and broadcasts t_T (via GPS systems in the road-to-vehicle communications context, for example). Each signer (vehicle), who uses a unique identity ID, to generate a signature σ on a message M (local information)[2] by using its own signing key $\mathsf{sigk}_{\mathsf{ID}}$ and t_T. A verifier (we assume a Road Side Unit (RSU) in the road-to-vehicle communications context) checks whether σ is a valid signature or not. We give a brief description of our system in Figure 1.

Next, we construct an anonymous time-dependent authentication system by using GS-TDL, in which two group signatures become linkable if a signer simultaneously generates a group signature twice. Time-dependent linking appears to be more suitable for our system than message-dependent linking [67] in which the vehicle is always linkable if it generates group signatures on the same message, and this situation might occur when a vehicle is used for work trips and uses the same road each day.[3] We note that no formal security definition for MLGS is provided in [67], and therefore the security proofs are informal. As a result, we can show an attack against the MLGS scheme of Wu et al., where anyone can generate a valid-but-untraceable group signature without using a secret key (See Section 6). In contrast, our GS-TDL scheme is provably secure.

Our GS-TDL is secure in the Random Oracle Model[4] (because we pursue a light-weight implementation of the system) under the q-Strong Diffie-Hellman (q-SDH) assumption [18] and the Strong Diffie-Hellman Inversion (SDHI) assumption [22, 29]. The group signature size in our system is shorter than that of previous schemes owing to time-dependent linkability. Specifically, a signature contains only 6 group elements, whereas that of the short group signature scheme [19] contains 9 group elements, that of the short controllable linkable group signature scheme [37] contains 8 group elements, and that of the controllable linkable group signature scheme (for dynamic group setting) [38] contains 12 group elements.[5] Moreover, our linking algorithm does not require cryptographic computations (i.e., comparisons to determine two elements are the same).

Finally, we provide the experimental results of our road-to-vehicle communication system, and show that our system is feasible in practice. To implement GS-TDL, we use the TEPLA library [5]. We note that we employ asymmetric pairing settings ((type III) Barreto-Naehrig (BN) curves [13]) with 254-bit order due to the recent novel works for solving the discrete logarithm problem over certain elliptic curves with symmetric pairing settings [33, 12].

Related Work: Since car security has been recognized as a real threat, several organizations have been launched, e.g., Preserve (EU) [4], ITS Info-communications Forum (Japan) [2], IntelliDrive (USA) [1], etc., and car security is researched in several papers. To name a few, Wetzels [66] reported security and privacy concerns regarding RFID-based car key applications. Busold et al. [21] proposed a security framework for secure smartphone-based immobilizers. Tillich and Wójcik [64] analyzed an open car immobilizer protocol stack and shows several attacks. Meiklejohn et al. [48] pointed out driving payment systems proposed

[2]Here, we need to assume that no personal information is revealed from a content M (or even its statistical values). Though we may apply privacy preserving techniques for modifying contents or its statistical values, e.g., [63, 45, 41, 65, 32, 26], we leave it as a future work of this paper.

[3]Even if a random nonce is included as a part of signed message, no linking algorithm works and this leads to a wag-the-dog situation. Even if a time T is included, e.g., sign $M||T$ by using a MLGS scheme, anyone can manipulate T and such a signer-driven anonymous system must be avoided because vehicles have incentive to hide identity. On the contrary, in GS-TDL, time T is authorized by TGU and no vehicle can manipulate T.

[4]We also give a GS-TDL scheme secure in the standard model in Appendix.

[5]We remark that these schemes [19, 37, 38] also achieve only CPA-anonymity (i.e., no opening oracle access is allowed in anonymity game) as in ours.

in [10, 54] reveal to drivers the locations of spot-checking road side cameras, and showed that colluding drivers can select roads for avoiding payment. Then, Meiklejohn et al. proposed a new system which they call Milo.

Bellare, Shi, and Zhang (BSZ) [16] show an extension of the Bellare-Micciancio-Warinschi (BMW) model (for dynamic group setting), and Sakai et al. [58] further extended the BSZ model to prevent signature hijacking attack. Nakanishi et al. proposed linkable group signature [50], where anyone can determine whether two signatures were made by the same signer or not. As a difference from GS-TDL, no time-dependent token is required for linking. That is, two group signatures made by the same signer are always linkable. A group signature with a relaxed anonymity for VANET has been considered in [46, 47]. But the link algorithm is not publicly executable, and an authority called Link Manager is introduced. That is, two group signatures made by the same signer are always linkable from the viewpoint of Link Manager. Moreover, pairing computations are required for linking. Abe et al. [6] proposed double-trapdoor anonymous tags which can generally construct traceable signatures [39]. Since a signer is always linkable after the corresponding token is broadcasted, we cannot use traceable signatures instead of GS-TDL. As a special case of traceable signatures, group signatures with controllable linkability has been proposed [38, 37], where a link key is defined for the linking procedure. However, pairing computations are required for linking, which lead to inefficiency.

We note that MLGS [67] is essentially the same as unique group signature proposed by Franklin and Zhang [29], and formal security definitions are given in [29]. That is, we may be able to apply unique group signature to construct an anonymous threshold authentication. However, our time-dependent linking seems suitable for the system rather than message-dependent linking as explained before. Moreover, the Franklin-Zhang model supports the open algorithm and therefore it considers CCA anonymity. Since we customize the syntax to be suitable for light-weight realization and exclude the open algorithm, in this paper we do not directly apply the Franklin-Zhang unique group signature scheme to our system.

2 Preliminaries

In this section, we give the definitions of bilinear groups, complexity assumptions, and digital signature as follows.

Complexity Assumptions: Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that takes a security parameter λ as input and generates a parameter $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ of bilinear groups, where p is a λ-bit prime, $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of order p, e is a bilinear map from $\mathbb{G}_1 \times \mathbb{G}_2$ to $\mathbb{G}_T$, and g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Here we use the asymmetric setting, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$.

Definition 2.1 (SDDHI assumption [22]). *We say that the SDDHI (Strong Decisional Diffie-Hellman Inversion) assumption holds if for all PPT adversaries $\mathcal{A}$, $|\Pr[x\ (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \xleftarrow{\$} \mathcal{G}(1^\lambda);\ \xleftarrow{\$} \mathbb{Z}_p;$ $(T, st) \leftarrow \mathcal{A}^{\mathcal{O}_x}(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, g_1^x);\ \tau_0 = g_1^{\frac{1}{x+T}};\ \tau_1 \xleftarrow{\$} \mathbb{G}_1;\ b \xleftarrow{\$} \{0,1\};\ b' \leftarrow A^{\mathcal{O}_x}(y_b, st):\ b = b'] - \frac{1}{2}|$ is negligible, where $\mathcal{O}_x$ is an oracle which takes as input $z \in \mathbb{Z}_p^* \setminus \{T\}$, outputs $g_1^{\frac{1}{x+z}}$.*

We remark that the underlying bilinear group must not be symmetric.

Definition 2.2 (q-SDH assumption [18]). *We say that the q-SDH (q-Strong Diffie-Hellman) assumption holds if for all PPT adversaries $\mathcal{A}$, $\Pr[(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \xleftarrow{\$} \mathcal{G}(1^\lambda);\ \gamma \xleftarrow{\$} \mathbb{Z}_p;\ (x, g_1^{\frac{1}{x+\gamma}}) \leftarrow \mathcal{A}(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_1^\gamma, \ldots, g_1^{\gamma^q}, g_2, g_2^\gamma);\ x \in \mathbb{Z}_p^* \setminus \{-\gamma\}]$ is negligible.*

Digital Signature: Let $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be a digital signature scheme. The key generation algorithm Gen takes as input a security parameter λ, and outputs a pair of verification/signing key $(\mathsf{vk}, \mathsf{sigk})$. The signing algorithm Sign takes as input sigk and a message to be signed $M \in \mathcal{M}$, where $\mathcal{M}$ is a message space, and outputs a signature Σ. The verification algorithm Verify takes as input vk, Σ and M, and outputs 0/1. We require the following correctness property: for all $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $M \in \mathcal{M}$, $\Pr[\mathsf{Verify}(\mathsf{vk}, \mathsf{Sign}(\mathsf{sigk}, M), M) = 1] = 1$ holds. Next, we define existential unforgeability against chosen message attack (EUF-CMA) as follows. Let $\mathcal{C}$ be the challenger, and $\mathcal{A}$ be an adversary. $\mathcal{C}$ runs $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and gives vk to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue signing queries M. $\mathcal{C}$ runs $\Sigma \leftarrow \mathsf{Sign}(\mathsf{sigk}, M)$ and returns Σ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs (Σ^*, M^*). We say that a digital signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ is EUF-CMA if the probability, that $\mathsf{Verify}(\mathsf{vk}, \Sigma^*, M^*) = 1$ and $\mathcal{A}$ did not send M^* as a signing query, is negligible.

3 Definitions of GS-TDL

In this section, we give the syntax and security definitions of GS-TDL by adding the above time-dependent linkability to the Bellare-Micciancio-Warinschi (BMW) model [15] (which is recognized as a de-facto standard for group signature).

Design Principle: We note that our overall goal is to apply GS-TDL to road-to-vehicle communications. Therefore, in addition to security, we attach great importance to the efficiency of the system. Because we pursued a lightweight implementation of the system, there is room for discussion about whether the open functionality should be utilized. In the open functionality, an authority (called an opener) can determine the identity of the actual signer by using a secret opening key. For example, the open functionality is implemented by using public key encryption (PKE) or non-interactive zero-knowledge proof of knowledge, and could be an efficiency bottleneck. It has been reported that the signature size of the Furukawa-Imai group signature scheme [30] can be reduced by 50% if the open functionality is removed in [27]; it has also been reported that implementing the open functionality without using PKE leads to a short group signature scheme at the expense of the signature opening costs [17]. Given the above facts, we do not consider the open functionality (we only consider the linking functionality). Moreover, we assume that the signing key of a vehicle is embedded in a device during the setup phase, and therefore we also removed an interactive join algorithm from our syntax. Finally, we considered the revocation functionality, especially verifier-local revocation (VLR) where no signer is involved in revocation procedures.

Definition 3.1 (Syntax of GS-TDL)**.** *A group signature scheme with time-token dependent linking $\mathcal{GS}\text{-}\mathcal{TDL}$ consists of the algorithms* $(\mathsf{Setup}, \mathsf{GKeyGen}, \mathsf{TKeyGen}, \mathsf{Join}, \mathsf{TokenGen}, \mathsf{GSign}, \mathsf{Revoke}, \mathsf{GVerify}, \mathsf{Link})$ *as follows:*

Setup**:** *The setup algorithm takes as input a security parameter λ, and outputs a public parameter params.*

$\mathsf{GKeyGen}$**:** *The group key generation algorithm takes as input params, and outputs a group public key* gpk*, a group master key* gsk*, an initial revocation storage* $\mathsf{grs} := \emptyset$ *and an initial revocation list* $\mathsf{RL}_0 := \emptyset$*.*

$\mathsf{TKeyGen}$**:** *The token key generation algorithm takes as input params, and outputs a public key* tpk *and a secret key* tsk*.*

Join**:** *The join algorithm takes as input* gsk, grs *and a unique identity* ID*, and outputs a signing key* $\mathsf{sigk}_{\mathsf{ID}}$ *and updated revocation storage. We remark that this algorithm is not required to be interactive.*

$\mathsf{TokenGen}$**:** *The token generation algorithm takes as input* tsk *and a time $T \in \mathcal{T}$, and outputs a token t_T, where $\mathcal{T} := \mathsf{poly}(\lambda)$ is the time space.*

GSign**:** *The signing algorithm takes as input* gpk, tpk, t_T, $\mathsf{sigk}_{\mathsf{ID}}$*, and a message M, and outputs a signature σ.*

Revoke**:** *The revocation algorithm takes as input* gpk, grs*, and a set of revoked users at a time T* $\{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\}$*, and outputs* RL_T*. Here, n_T is the number of users that are additionally revoked on T.*

$\mathsf{GVerify}$**:** *The verification algorithm takes as input* gpk, tpk, RL_T*, σ, and M, and outputs 0* (invalid) *or 1* (valid)*.*

Link: *The linking algorithm takes as input* gpk, tpk, *and* RL_T, *and two signatures and messages* (σ_0, M_0, T_0) *and* (σ_1, M_1, T_1), *and outputs 1 if two signatures are made by the same signer, and 0 otherwise. We remark that the* Link *algorithm outputs 0 does not guarantee two signatures are made by the different signers. For example, if a signature is invalid, then the algorithm outputs 0.*

We require the following correctness, where any honestly generated signatures are valid, and the Link algorithm correctly links two signatures if these are generated by the same signing key with the same token, unless the corresponding signer is not revoked. Moreover, we require that a signature is invalid if the corresponding signer is revoked.[6]

Definition 3.2 (Correctness)**.** *For any probabilistic polynomial time (PPT) adversary* $\mathcal{A}$ *and the security parameter* $\lambda \in \mathbb{N}$, *we define the experiment* $\mathrm{Exp}^{corr}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$ *as follows.*

$$
\begin{aligned}
&\mathrm{Exp}^{corr}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda):\\
&\quad params \leftarrow \mathsf{Setup}(1^\lambda);\ (\mathsf{gpk}, \mathsf{gsk}, \mathsf{grs}, \mathsf{RL}_0) \leftarrow \mathsf{GKeyGen}(params)\\
&\quad (\mathsf{tpk}, \mathsf{tsk}) \leftarrow \mathsf{TKeyGen}(params);\ \mathsf{GU} := \emptyset\\
&\quad (\mathsf{ID}^*, T^*, M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{AddU}(\cdot),\mathsf{Revoke}(\mathsf{grs},\cdot)}(\mathsf{gpk}, \mathsf{tpk})\\
&\quad \mathsf{ID}^* \in \mathsf{GU};\ t_{T^*} \leftarrow \mathsf{TokenGen}(\mathsf{tsk}, T^*)\\
&\quad \sigma_0 \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_{T^*}, \mathsf{sigk}_{\mathsf{ID}^*}, M_0);\ \sigma_1 \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_{T^*}, \mathsf{sigk}_{\mathsf{ID}^*}, M_1)\\
&\quad \textit{Return } 1 \textit{ if the following holds}:\\
&\qquad \big[\mathsf{ID}^* \notin \mathsf{RL}_{T^*} \wedge \big((\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, M_0, \sigma_0) = 0\\
&\qquad \vee\ \mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, M_1, \sigma_1) = 0)\\
&\qquad \vee\ \mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, (M_0, \sigma_0, T^*), (M_1, \sigma_1, T^*)) = 0\big)\big]\\
&\quad \vee \big[\mathsf{ID}^* \in \mathsf{RL}_{T^*} \wedge \big((\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, M_0, \sigma_0) = 1\\
&\qquad \vee\ \mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, M_1, \sigma_1) = 1\big)\big]\\
&\quad \textit{Otherwise return } 0
\end{aligned}
$$

AddU: *The add user oracle allows an adversary* $\mathcal{A}$ *to add honest users to the group. On input an identity* ID, *this oracle runs* $\mathsf{sigk}_{\mathsf{ID}} \leftarrow \mathsf{Join}(, \mathsf{gsk}, \mathsf{grs}, \mathsf{ID})$. ID *is added to* GU.

Revoke: *Let* $T-1$ *be the time that the oracle is called. The revocation oracle allows an adversary* $\mathcal{A}$ *to revoke honest users. On input identities* $\{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\}$, *this oracle runs* $\mathsf{RL}_T \leftarrow \mathsf{Revoke}(\mathsf{gpk}, \mathsf{grs}, \{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\})$. *We remark that* T^* *is the challenge time that* $\mathcal{A}$ *outputs* $(\mathsf{ID}^*, M_0, M_1)$.

$\mathcal{GS}$-$\mathcal{TDL}$ *is said to be satisfying correctness if the advantage* $\mathrm{Adv}^{corr}_{\mathsf{GS},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{corr}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda) = 1]$ *is negligible for any PPT adversary* $\mathcal{A}$.

Next, we give our anonymity definition which guarantees that no adversary who has tsk can distinguish whether two signatures are generated by the same signer or not, if the corresponding linkable signatures are not generated. In contrast to the BMW model, $\mathcal{A}$ is not allowed to obtain signing keys of challenge users (selfless anonymity). This is a reasonable setting since $\mathcal{A}$ can trivially break anonymity if $\mathcal{A}$ obtains such signing keys. For example, let $\mathcal{A}$ have $\mathsf{sigk}_{\mathsf{ID}_{i_0}}$. Then, $\mathcal{A}$ can make a signature σ on T_0 using $\mathsf{sigk}_{\mathsf{ID}_{i_0}}$ (with arbitrary message M), and can check whether $\mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T_0}, (M_0, \sigma^*, T_0), (M, \sigma, T_0)) = 1$ or not, where σ^* is the challenge signature. Instead, $\mathcal{A}$ is allowed to access the GSign oracle in our definition. Moreover, we consider backward unlinkability, where no adversary can break anonymity even after the challenge signers are revoked.

[6]As a remark, the case that an adversary generates a valid signature using a revoked user's signing key cannot be captured by unforgeability since the open algorithm is not defined. Instead, we consider the case that a signature is invalid when the corresponding signer is revoked in correctness, though it might be additionally defined such as revocation soundness.

Definition 3.3 (Anonymity). *For any PPT adversary $\mathcal{A}$ and a security parameter $\lambda \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{\textit{anon-tg-b}}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$ *as follows.*

$$\begin{aligned}
&\mathrm{Exp}^{\textit{anon-tg-b}}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda):\\
&\quad \mathit{params} \leftarrow \mathsf{Setup}(1^\lambda)\\
&\quad (\mathsf{gpk},\mathsf{gsk},\mathsf{grs},\mathsf{RL}_0) \leftarrow \mathsf{GKeyGen}(\mathit{params});\\
&\quad (\mathsf{tpk},\mathsf{tsk}) \leftarrow \mathsf{TKeyGen}(\mathit{params});\ \mathsf{GU} := \emptyset;\ \mathsf{STSet} := \emptyset\\
&\quad d \leftarrow \mathcal{A}^{\mathsf{AddU}(\cdot),\mathsf{Revoke}(\mathsf{grs},\cdot),\mathsf{GSign}(\cdot,\cdot,\cdot),\mathsf{Ch}(b,\cdot,\cdot,\cdot,\cdot,\cdot)}(\mathsf{gpk},\mathsf{tpk},\mathsf{tsk})\\
&\quad \mathit{Return}\ d
\end{aligned}$$

AddU: *The same as before.*

Revoke: *The same as before. We remark that if $T_0 \neq T_1$ and assume that $T_0 < T_1$, then ID_{i_0} and/or ID_{i_1} can be revoked after T_1. If $T_0 = T_1$, then ID_{i_0} and/or ID_{i_1} can be revoked after T_0.*

GSign: *The signing oracle takes as input ID, t_T, and a message M. We assume that t_T is a valid token which means that the $\mathsf{GVerify}$ algorithm outputs 1 for all honestly generated signatures with t_T, even though this is made by $\mathcal{A}$. If $\mathsf{ID} \notin \mathsf{GU}$, then the oracle runs $\mathsf{AddU}(\mathsf{ID})$. The oracle returns $\sigma \leftarrow \mathsf{GSign}(\mathsf{gpk},\mathsf{tpk},t_T,\mathsf{sigk}_{\mathsf{ID}},M)$ and stores (ID,T) in STSet.*

Ch: *The challenge oracle takes as input ID_{i_0}, ID_{i_1}, t_{T_0}, t_{T_1}, M_0^*, and M_1^* where $\mathsf{ID}_{i_0} \neq \mathsf{ID}_{i_1}$ and $\mathsf{ID}_{i_0}, \mathsf{ID}_{i_1} \in \mathsf{GU}$. Return signature(s) according to the following cases:*

- $T_0 = T_1$: *If $(\mathsf{ID}_{i_0},T_0),(\mathsf{ID}_{i_1},T_1) \notin \mathsf{STSet}$, then compute $\sigma^* \leftarrow \mathsf{GSign}(\mathsf{gpk},\mathsf{tpk},t_{T_b},\mathsf{sigk}_{\mathsf{ID}_{i_b}},M^*)$, and return σ^*. Without loss of generality, we set $M^* = M_0^* = M_1^*$.*
- $T_0 \neq T_1$: *If $(\mathsf{ID}_{i_0},T_0),(\mathsf{ID}_{i_1},T_1),(\mathsf{ID}_{i_0},T_1) \notin \mathsf{STSet}$, then compute $\sigma_0^* \leftarrow \mathsf{GSign}(\mathsf{gpk},\mathsf{tpk},t_{T_0},\mathsf{sigk}_{\mathsf{ID}_{i_0}},M_0^*)$ and $\sigma_1^* \leftarrow \mathsf{GSign}(\mathsf{gpk},\mathsf{tpk},t_{T_1},\mathsf{sigk}_{\mathsf{ID}_{i_b}},M_1^*)$, and return σ_0^* and σ_1^*.*

Moreover, we assume that t_{T_0} and t_{T_1} are valid tokens even though these are made by $\mathcal{A}$, which means that the $\mathsf{GVerify}$ algorithm outputs 1 for all honestly generated signatures with t_{T_0} or t_{T_1}.[7]

$\mathcal{GS\text{-}TDL}$ is said to be satisfying anonymity if the advantage $\mathrm{Adv}^{\textit{anon-tg}}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda) := |\Pr[\mathrm{Exp}^{\textit{anon-tg-1}}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda) = 1] - \Pr[\mathrm{Exp}^{\textit{anon-tg-0}}_{\mathsf{GS},\mathcal{A}}(\lambda) = 1]|$ *is negligible for any PPT adversary $\mathcal{A}$.*

When $T_0 = T_1$, our definition guarantees that two different vehicles are unlinkable even if they generate signatures at the same time period. We note that if $\mathcal{A}$ obtains two signatures even though $T_0 = T_1$, then $\mathcal{A}$ can break anonymity by using the Link algorithm. Therefore, $\mathcal{A}$ is allowed to obtain one challenge signature σ^* only. When $T_0 \neq T_1$, our definition guarantees that a vehicle is still unlinkable if the vehicle respectively generates two signatures on different time periods. That is, when $\mathcal{A}$ obtains σ_0^*, which is generated by ID_{i_0} at a time T_0, and σ_1^*, which is generated by ID_{i_b} at a time $T_1 \neq T_0$, no $\mathcal{A}$ can distinguish whether two signatures are respectively made by the same user ID_{i_0} or different users ID_{i_0} and ID_{i_1}. In order to prevent a trivial linking attack, $\mathcal{A}$ is not allowed to obtain a signature for (ID_{i_0},T_1) in this case.

We note that we do not have to consider the case $\mathsf{ID}_{i_0} = \mathsf{ID}_{i_1}$ and $T_0 \neq T_1$, since time T is an input of the verification algorithm. That is, $\mathcal{A}$ can easily break anonymity in this case: $\mathcal{A}$ just obtains $\sigma^* \leftarrow \mathsf{GSign}(\mathsf{gpk},\mathsf{tpk},t_{T_b},\mathsf{sigk}_{\mathsf{ID}_{i_0}},M^*)$ and checks whether $\mathsf{GVerify}(\mathsf{gpk},\mathsf{tpk},\mathsf{RL}_{T_0},M^*,\sigma^*) = 1$ or not.

As a remark, the above definition is anonymity against Token Generator, and the other definition, where anonymity against Key Issuer, can also be defined as follows: An adversary (who has gsk) is allowed to issue token queries, except for the challenge time. However, we need to restrict that the adversary is not allowed to obtain tokens for the challenge time (if not, the adversary can easily break anonymity since the adversary can generate signing keys for all identities, and the Link algorithm is publicly executable), and this setting is

[7] This condition must be required to exclude the trivially-broken case, e.g., $\mathcal{A}$ honestly generates t_{T_0} and sets t_{T_1} as arbitrary value. Then, $\mathcal{A}$ can check whether σ^* is valid or not. If yes, then $b = 0$ and $b = 1$ otherwise.

far from the real situation we considered (i.e., tokens are "broadcasted" by Token Generator). If we assume an interactive join process, and a secret value known by a vehicle only is prepared, then such a definition might make sense since Key Issuer cannot generate a group signature. However, any interactive join process is hard to be considered in the vehicle context. So, we do not consider the anonymity against Key Issuer case, and leave it as a future work of this paper since it might be interesting in other settings.

Next, we define unforgeability which guarantees that nobody who does not have a signing key or does not have a token can generate a valid signature.

Definition 3.4 (Unforgeability)**.** *For any PPT adversary $\mathcal{A}$ and security parameter $\lambda \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{unf}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$ *as follows, where* $\mathcal{O} := (\mathsf{AddU}(\cdot), \mathsf{Revoke}(\mathsf{grs}, \cdot), \mathsf{TokenGen}(\mathsf{tsk}, \cdot), \mathsf{SetToken}(\cdot), \mathsf{GSign}(\cdot,\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{TSK}(\cdot))$.

$$
\begin{aligned}
&\mathrm{Exp}^{unf}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda):\\
&\quad params \leftarrow \mathsf{Setup}(1^\lambda)\\
&\quad (\mathsf{gpk}, \mathsf{gsk}, \mathsf{grs}, \mathsf{RL}_0) \leftarrow \mathsf{GKeyGen}(params);\ \ (\mathsf{tpk}, \mathsf{tsk}) \leftarrow \mathsf{TKeyGen}(params)\\
&\quad \mathsf{GU} := \emptyset;\ \ \mathsf{TSet} := \emptyset;\ \ \mathsf{SSet} := \emptyset\\
&\quad (M, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{gpk}, \mathsf{tpk})\\
&\quad \mathit{Return}\ 1\ \mathit{if}\ (1) \wedge (2) \wedge ((3) \vee (4))\ \mathit{hold}:\\
&\quad\quad (1)\ \mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, M, \sigma) = 1\\
&\quad\quad (2)\ (T^*, M, \sigma) \notin \mathsf{SSet}\\
&\quad\quad (3)\ T \notin \mathsf{TSet} \wedge \mathsf{TSK}(\cdot)\ \mathit{has\ not\ been\ called}\\
&\quad\quad (4)\ \mathsf{TSK}(\cdot)\ \mathit{has\ been\ called\ with\ non\text{-}}\bot\ \mathit{output}\\
&\quad \mathit{Otherwise\ return}\ 0
\end{aligned}
$$

AddU**:** *The same as before.*

Revoke**:** *The same as before. We note that T^* is the challenge time that $\mathcal{A}$ outputs (M, σ).*

$\mathsf{TokenGen}$**:** *The token generation oracle takes as input a time T. This oracle runs $t_T \leftarrow \mathsf{TokenGen}(\mathsf{tsk}, T)$, stores T in TSet, and returns t_T.*

$\mathsf{SetToken}$**:** *The token setting oracle takes as input t_T, and sets t_T as the token at a time T. Without loss of generality, we assume that if the $\mathsf{TokenGen}$ oracle is called, the $\mathsf{SetToken}$ oracle is also called right after calling the $\mathsf{TokenGen}$ oracle. We remark that $\mathcal{A}$ can set arbitrary value as t_T via this oracle.*

GSign**:** *The signing oracle takes as input ID, T, and a message M. If $\mathsf{ID} \notin \mathsf{GU}$, then the oracle runs $\mathsf{AddU}(\mathsf{ID})$. If t_T is not generated via the $\mathsf{TokenGen}$ oracle, then call the oracle $\mathsf{TokenGen}(\mathsf{tsk}, T)$ and the $\mathsf{SetToken}$ oracle. The oracle returns $\sigma \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_T, \mathsf{sigk}_{\mathsf{ID}}, M)$ and stores (T, M, σ) in SSet.*

USK**:** *The user key reveal oracle takes as input ID. If the TSK oracle was called before, then return $\bot$. If $\mathsf{ID} \notin \mathsf{GU}$, then the oracle runs $\mathsf{AddU}(\mathsf{ID})$. Return $\mathsf{sigk}_{\mathsf{ID}}$.*

TSK**:** *The token key reveal oracle returns $\bot$ if the USK oracle was called before and at least one identity is not revoked.*[8] *Otherwise, return tsk.*

$\mathcal{GS}$-$\mathcal{TDL}$ is said to be unforgeable if the advantage $\mathrm{Adv}^{unf}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{unf}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda) = 1]$ *is negligible for any PPT adversary $\mathcal{A}$.*

Finally, we define linking soundness which guarantees that the Link algorithm does not return 1 when two valid signatures are made by either different signers or different time tokens.

[8] That is, the TSK oracle returns tsk if all identities input in the USK oracle were revoked.

Definition 3.5 (Linking Soundness)**.** *For any PPT adversary $\mathcal{A}$ and security parameter $\lambda \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{snd}_{\textsf{GS-TDL},\mathcal{A}}(\lambda)$ *as follows.*

$$
\begin{aligned}
&\mathrm{Exp}^{snd}_{\textsf{GS-TDL},\mathcal{A}}(\lambda):\\
&\quad params \leftarrow \mathsf{Setup}(1^\lambda)\\
&\quad (\mathsf{gpk}, \mathsf{gsk}, \mathsf{grs}, \mathsf{RL}_0) \leftarrow \mathsf{GKeyGen}(params)\\
&\quad (\mathsf{tpk}, \mathsf{tsk}) \leftarrow \mathsf{TKeyGen}(params);\ (\mathsf{ID}_0, \mathsf{ID}_1, T_0, T_1, M, st) \leftarrow \mathcal{A}(\mathsf{gpk}, \mathsf{tpk})\\
&\quad (\mathsf{ID}_0, T_0) \neq (\mathsf{ID}_1, T_1);\ \mathsf{sigk}_{\mathsf{ID}_0} \leftarrow \mathsf{Join}(\mathsf{gsk}, \mathsf{grs}, \mathsf{ID}_0);\ \mathsf{sigk}_{\mathsf{ID}_1} \leftarrow \mathsf{Join}(\mathsf{gsk}, \mathsf{grs}, \mathsf{ID}_1)\\
&\quad t_{T_0} \leftarrow \mathsf{TokenGen}(\mathsf{tsk}, T_0);\ t_{T_1} \leftarrow \mathsf{TokenGen}(\mathsf{tsk}, T_1)\\
&\quad \sigma_0 \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_{T_0}, \mathsf{sigk}_{\mathsf{ID}_0}, M)\\
&\quad (M^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Revoke}(\mathsf{grs},\cdot)}(st, \mathsf{sigk}_{\mathsf{ID}_1}, t_{T_1}, \sigma_0)\\
&\quad \textit{Return } 1 \textit{ if } \mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T_1}, (M, \sigma_0, T_0), (M^*, \sigma^*, T_1)) = 1\\
&\quad \textit{Otherwise return } 0
\end{aligned}
$$

Revoke**:** *The same as before.*

A GS-TDL scheme is said to be satisfying linking soundness if the advantage $\mathrm{Adv}^{snd}_{\textsf{GS-TDL},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{snd}_{\textsf{GS-TDL},\mathcal{A}}(\lambda) = 1]$ *is negligible for any PPT adversary $\mathcal{A}$.*

4 Proposed GS-TDL Scheme

In this section, we give our GS-TDL scheme. Since we mainly pursue a *light-weight realization* of the system, here we do not employ structure preserving signatures [7, 8] and Groth-Sahai proofs [35] which are typically used for constructing group signature schemes secure in the standard model, e.g., [34, 9, 42, 43] (we will give a GS-TDL scheme secure in the standard model in Appendix). Instead, we employ the Fiat-Shamir transformation [28] which converts a 3-move Σ protocol to non-interactive zero-knowledge (NIZK) proof, as in group signature schemes secure in the random oracle model, e.g., [30, 19, 25, 17].

The Basic Idea: Our GS-TDL scheme is based on the Furukawa-Imai group signature scheme [30] which is recognized as one of the most efficient group signature schemes. First, we exclude the open functionality from the Furukawa-Imai group signature scheme as in [27]. Next, for the linking property, we apply the Franklin-Zhang technique [29], where a group signature contains Belenkiy et al.'s verifiable random function (VRF) [14]. Concretely, the value $\tau = g^{\frac{1}{x+T}}$ is contained in a signature at a time T, where x is a (part of) signing key. Then, if a signer computes two or more group signatures at a time T, then the value τ is the same, and can be linked without any cryptographic operation. Whereas, τ itself can be seen as a random value (under the SDDHI assumption), and therefore a signer is still anonymous unless the signer computes two or more group signatures at the same time. For (verifier-local) revocation, we also apply τ such that τ is added in a revocation list. Note that the verification cost of VLR-type group signatures schemes [20, 44, 52] is $O(|\mathsf{RL}_T|)$, especially, $|\mathsf{RL}_T|$-times pairing computations are required. In order to avoid such an inefficiency, we use the linkable part τ for revocation and this setting requires no cryptographic operation.

Construction 1 (Proposed GS-TDL scheme)**.**

$\mathsf{Setup}(1^\lambda)$**:** *Let* $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ *be a bilinear group with prime order p, where* $\langle g_1 \rangle = \mathbb{G}_1$, $\langle g_2 \rangle = \mathbb{G}_2$, *and* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *be a bilinear map. Output* $params = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$.

$\mathsf{GKeyGen}(params)$**:** *Choose* $\gamma \xleftarrow{\$} \mathbb{Z}_p$, *and* $h \xleftarrow{\$} \mathbb{G}_1$, *and compute* $W = g_2^\gamma$. *Output* $\mathsf{gpk} = (params, h, W, e(g_1, g_2), e(h, W), e(h, g_2), H)$, $\mathsf{gsk} = \gamma$, *where* $H : \{0,1\}^* \to \mathbb{Z}_p$ *is a hash function modeled as a random oracle,* $\mathsf{grs} := \emptyset$ *and* $\mathsf{RL}_0 := \emptyset$.

$\mathsf{TKeyGen}(params)$**:** *Let* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *be a digital signature scheme. Run* $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, and output* $\mathsf{tpk} := \mathsf{vk}$ *and* $\mathsf{tsk} := \mathsf{sigk}$*.*

$\mathsf{Join}(\mathsf{gsk}, \mathsf{grs}, \mathsf{ID})$**:** *Choose* $x, y \xleftarrow{\$} \mathbb{Z}_p$*, compute* $A = (g_1 h^{-y})^{\frac{1}{\gamma+x}}$*, output* $\mathsf{sigk}_{\mathsf{ID}} = (x, y, A)$*, and update* $\mathsf{grs} := \mathsf{grs} \cup \{(\mathsf{ID}, x)\}$*.*

$\mathsf{TokenGen}(\mathsf{tsk}, T)$**:** *Assume that* $T \in \mathbb{Z}_p$*. Compute* $W_T = g_2^T$ *and* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{sigk}, W_T)$*, and output* $t_T = (T, W_T, \Sigma)$*.*

$\mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_T, \mathsf{sigk}_{\mathsf{ID}}, M)$**:** *Let* $\mathsf{sigk}_{\mathsf{ID}} = (x, y, A)$ *and* $t_T = (T, W_T, \Sigma)$*. If* $\mathsf{Verify}(\mathsf{vk}, W_T, \Sigma) = 0$*, then output* $\perp$*. Otherwise, choose* $\beta \xleftarrow{\$} \mathbb{Z}_p$*, set* $\delta = \beta x - y$*, and compute* $C = Ah^\beta$ *and* $\tau = g_1^{\frac{1}{x+T}}$*. Choose* $r_x, r_\delta, r_\beta \xleftarrow{\$} \mathbb{Z}_p$*, and compute*

$$R_1 = \frac{e(h, g_2)^{r_\delta} e(h, W)^{r_\beta}}{e(C, g_2)^{r_x}}, \ R_2 = e(\tau, g_2)^{r_x}$$
$$c = H(\mathsf{gpk}, \mathsf{tpk}, C, \tau, R_1, R_2, M)$$
$$s_x = r_x + cx, s_\delta = r_\delta + c\delta, \text{and } s_\beta = r_\beta + c\beta,$$

and output $\sigma = (C, \tau, c, s_x, s_\delta, s_\beta)$*. This proves that (1)* (x, y, A) *is a valid Boneh-Boyen signature under* gpk *(i.e.,* $\mathsf{sigk}_{\mathsf{ID}}$ *is issued by Key Issuer) and (2)* τ *is computed by the same* x*.*

Pairing-free Variant: *We remark that* $e(h, g_2)$ *and* $e(h, W)$ *are pre-computable and can be contained in* gpk*. Moreover* $e(C, g_2)^{r_x}$ *and* $e(\tau, g_2)^{r_x}$ *can be represented as* $e(A, g_2)^{r_x} e(h, g_2)^{\beta r_x}$ *and* $e(g_1, g_2)^{\frac{r_x}{x+T}}$*, respectively. Then,*

$$R_1 = \frac{e(h, g_2)^{r_\delta - \beta r_x} e(h, W)^{r_\beta}}{e(A, g_2)^{r_x}} \text{ and } R_2 = e(g_1, g_2)^{\frac{r_x}{x+T}}.$$

So, by assuming that $e(A, g_2)$ *is pre-computable (we can simply assume that* $e(A, g_2)$ *is contained in* $\mathsf{sigk}_{\mathsf{ID}}$*), we can remove any pairing computation from the signing algorithm, instead of adding two exponentiations over* $\mathbb{G}_T$*.*

$\mathsf{Revoke}(\mathsf{gpk}, \mathsf{grs}, \{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\})$**:** *If there exists* $\mathsf{ID} \in \{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\}$ *that is not joined to the system via the* Join *algorithm, then output* $\perp$*. Otherwise, extract* $(\mathsf{ID}_{T,1}, x_{T,1}), \ldots, (\mathsf{ID}_{T,n_T}, x_{n_T})$ *from* grs*. Output* $\mathsf{RL}_T := \{(\mathsf{ID}_{T,1}, g_1^{\frac{1}{x_{T,1}+T}}), \ldots (\mathsf{ID}_{T,n_T}, g_1^{\frac{1}{x_{T,n_T}+T}})\}$*.*

$\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_T, M, \sigma)$**:** *Assume that* $\mathsf{Verify}(\mathsf{vk}, W_T, \Sigma) = 1$ *(if not, output* $\perp$*). Parse* $\sigma = (C, \tau, c, s_x, s_\delta, s_\beta)$*. If* τ *is contained in* RL_T *such that* $(\mathsf{ID}, \tau) \in \mathsf{RL}_T$ *for some* ID*, then output 0. Otherwise, compute*

$$R_1' = \frac{e(h, g_2)^{s_\delta} e(h, W)^{s_\beta}}{e(C, g_2)^{s_x}} \Big(\frac{e(C, W)}{e(g_1, g_2)}\Big)^{-c} \text{ and } R_2' = e(\tau, g_2)^{s_x} \Big(\frac{e(g_1, g_2)}{e(\tau, W_T)}\Big)^{-c},$$

and output 1 if $c = H(\mathsf{gpk}, \mathsf{tpk}, C, \tau, R_1', R_2', M)$ *holds, and 0 otherwise. We remark that* $e(h, g_2)$*,* $e(h, W)$ *and* $e(g_1, g_2)$ *are pre-computable and contained in* gpk*.*

$\mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_T, (M_0, \sigma_0, T_0), (M_1, \sigma_1, T_1))$**:** *Parse* $\sigma_0 = (C_0, \tau_0, c_0, s_{x,0}, s_{\delta,0}, s_{\beta,0})$ *and* $\sigma_1 = (C_1, \tau_1, c_1, s_{x,1}, s_{\delta,1}, s_{\beta,1})$*. If either* $T \neq T_0$ *or* $T \neq T_1$*, then output* 0*. Else if either* $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T_0}, M_0, \sigma_0) = 0$ *or* $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T_0}, M_1, \sigma_1) = 0$*, then output 0. Otherwise, output 1 if* $\tau_0 = \tau_1$*, and 0 otherwise.*

Since τ just depends on T and x, and does not contain any randomness, we can directly use τ for revocation. Since we do not have to run any cryptographic operation, we can achieve the (almost) constant verification cost by using hash tables which are made in the Revoke algorithm.

As a remark, the open algorithm, where an authority can identify the actual signer, also can be implemented (though we do not use it) as follows: let (ID, g_2^x) be preserved in the join phase, and the open algorithm checks whether $e(\tau, g_2^x g_2^T) = e(g_1, g_2)$ or not. If the equation holds, then ID is the identity of the corresponding signer. This open algorithm is essentially the same as that of the Bichsel et al. scheme [17].

5 Security Analysis

In this section, we give security proofs of our scheme.

Theorem 5.1. *The proposed GS-TDL scheme has anonymity in the random oracle model under the SDDHI assumption, where H is modeled as a random oracle.*

Proof. We define the following two games: Game 0 is the same as $\mathrm{Exp}^{anon\text{-}tg\text{-}b}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$. Game 1 is the same as Game 0, except τ^* contained in σ^* is randomly chosen, and σ^* is generated by the simulation of NIZK. Here, we show that there exist an algorithm $\mathcal{B}$ that breaks the SDDHI problem by using $\mathcal{A}$ as follows.

Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ be a bilinear group, and (g_1, g_2, g_1^x) is an instance of the SDDHI problem. Let q be the number of AddU queries. $\mathcal{B}$ chooses $i^* \in [1, q]$ and set x is a part of signing key of the user. $\mathcal{B}$ chooses $\gamma \xleftarrow{\$} \mathbb{Z}_p$, and $h \xleftarrow{\$} \mathbb{G}_1$, computes $W = g_2^{\gamma}$, and sets $\mathsf{gpk} = (params, h, W, e(g_1, g_2), e(h, W), e(h, g_2), H)$, where $H : \{0,1\}^* \to \mathbb{Z}_p$ is a hash function modeled as a random oracle. $\mathcal{B}$ also runs $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^{\lambda})$, and sets $\mathsf{tpk} := \mathsf{vk}$ and $\mathsf{tsk} := \mathsf{sigk}$. $\mathcal{B}$ sends gpk, tpk, and tsk to $\mathcal{A}$.

In the i-th AddU query (with input ID), where $i \neq i^*$, $\mathcal{B}$ chooses $x, y \xleftarrow{\$} \mathbb{Z}_p$, computes $A = (g_1 h^{-y})^{\frac{1}{\gamma+x}}$, sets $\mathsf{sigk}_{\mathsf{ID}} = (x, y, A)$, and adds ID to GU. In the i^*-th AddU query (with input ID^*), $\mathcal{B}$ adds ID^* to GU.

For a GSign query with input (ID, t_T, M), if $\mathsf{ID} \notin \mathsf{GU}$, then $\mathcal{B}$ runs the simulation of the AddU oracle. If $\mathsf{ID} \neq \mathsf{ID}^*$, then $\mathcal{B}$ computes a group signature σ as in the actual GSign algorithm, returns σ to $\mathcal{A}$, and adds (ID, T) to STSet. Let $\mathsf{ID} = \mathsf{ID}^*$. $\mathcal{B}$ sends T to $\mathcal{O}_x$, and obtains $\tau = g_1^{\frac{1}{x+T}}$. $\mathcal{B}$ chooses $s_x, s_\delta, s_\beta, c \xleftarrow{\$} \mathbb{Z}_p$ and $C \xleftarrow{\$} \mathbb{G}_1$, computes

$$R_1 = \frac{e(h, g_2)^{s_\delta} e(h, W)^{s_\beta}}{e(C, g_2)^{s_x}} \Big(\frac{e(C, W)}{e(g_1, g_2)}\Big)^{-c}$$

and

$$R_2 = e(\tau, g_2)^{s_x} \Big(\frac{e(g_1, g_2)}{e(\tau, W_T)}\Big)^{-c},$$

and patches H such that $c := H(\mathsf{gpk}, \mathsf{tpk}, C, \tau, R_1, R_2, M)$. $\mathcal{B}$ returns $\sigma = (C, \tau, c, s_x, s_\delta, s_\beta)$ to $\mathcal{A}$.

In the challenge phase, $\mathcal{A}$ sends $(\mathsf{ID}_{i_0}, \mathsf{ID}_{i_1}, t_{T_0}, t_{T_1}, M_0^*, M_1^*)$ to $\mathcal{B}$. $\mathcal{B}$ chooses $b \xleftarrow{\$} \{0,1\}$. If $\mathsf{ID}_{i_b} \neq \mathsf{ID}^*$, then $\mathcal{B}$ aborts. Let $\mathsf{ID}_{i_b} = \mathsf{ID}^*$ (this holds with the probability at least $1/q$). Next, we consider the following two cases:

$T_0 = T_1$: Let (T, W_T) be contained in both t_{T_0} and t_{T_1}. $\mathcal{B}$ sends $T := T_0 = T_1$ to the challenger of the SDDHI problem, and obtains τ^*. We remark that T was not sent to $\mathcal{O}_x$. $\mathcal{B}$ chooses $s_x^*, s_\delta^*, s_\beta^*, c^* \xleftarrow{\$} \mathbb{Z}_p$ and $C^* \xleftarrow{\$} \mathbb{G}_1$, computes

$$R_1^* = \frac{e(h, g_2)^{s_\delta^*} e(h, W)^{s_\beta^*}}{e(C, g_2)^{s_x^*}} \Big(\frac{e(C, W)}{e(g_1, g_2)}\Big)^{-c^*}$$

and

$$R_2 = e(\tau^*, g_2)^{s_x^*} \Big(\frac{e(g_1, g_2)}{e(\tau^*, W_T)}\Big)^{-c^*},$$

and patches H such that $c^* := H(\mathsf{gpk}, \mathsf{tpk}, C^*, \tau^*, R_1^*, R_2^*, M^*)$. $\mathcal{B}$ returns $\sigma^* = (C^*, \tau^*, c^*, s_x^*, s_\delta^*, s_\beta^*)$ to $\mathcal{A}$.

$T_0 \neq T_1$: Let (T_0, W_{T_0}) and (T_1, W_{T_1}) be contained in t_{T_0} and t_{T_1}, respectively. $\mathcal{B}$ sends T_0 to $\mathcal{O}_x$, and obtains $\tau_0^* = g_1^{\frac{1}{x+T_0}}$. $\mathcal{B}$ chooses $s_{x,0}^*, s_{\delta,0}^*, s_{\beta,0}^*, c_0^* \xleftarrow{\$} \mathbb{Z}_p$ and $C_0^* \xleftarrow{\$} \mathbb{G}_1$, computes

$$R_{1,0}^* = \frac{e(h, g_2)^{s_{\delta,0}^*} e(h, W)^{s_{\beta,0}^*}}{e(C_0^*, g_2)^{s_{x,0}^*}} \Big(\frac{e(C_0^*, W)}{e(g_1, g_2)}\Big)^{-c_0^*}$$

and

$$R_{2,0}^* = e(\tau_0^*, g_2)^{s_{x,0}^*} \Big(\frac{e(g_1, g_2)}{e(\tau_0^*, W_{T_0})}\Big)^{-c_0^*},$$

and patches H such that $c_0^* := H(\mathsf{gpk}, \mathsf{tpk}, C_0^*, \tau_0^*, R_{1,0}^*, R_{2,0}^*, M_0^*)$. Moreover, $\mathcal{B}$ sends T_1 to the challenger of the SDDHI problem, and obtains τ_1^*. We remark that T_1 was not sent to $\mathcal{O}_x$. $\mathcal{B}$ chooses $s_{x,1}^*, s_{\delta,1}^*, s_{\beta,1}^*, c_1^* \xleftarrow{\$} \mathbb{Z}_p$ and $C_1^* \xleftarrow{\$} \mathbb{G}_1$, computes

$$R_{1,1}^* = \frac{e(h, g_2)^{s_{\delta,1}^*} e(h, W)^{s_{\beta,1}^*}}{e(C_1^*, g_2)^{s_{x,1}^*}} \left(\frac{e(C_1^*, W)}{e(g_1, g_2)}\right)^{-c_1^*}$$

and

$$R_{2,1}^* = e(\tau_1^*, g_2)^{s_{x,1}^*} \left(\frac{e(g_1, g_2)}{e(\tau_1^*, W_{T_1})}\right)^{-c_1^*},$$

and patches H such that $c_1^* := H(\mathsf{gpk}, \mathsf{tpk}, C_1^*, \tau_1^*, R_{1,1}^*, R_{2,1}^*, M_1^*)$. $\mathcal{B}$ returns $\sigma_0^* = (C_0^*, \tau_0^*, c_0^*, s_{x,0}^*, s_{\delta,0}^*, s_{\beta,0}^*)$ and $\sigma_1^* = (C_1^*, \tau_1^*, c_1^*, s_{x,1}^*, s_{\delta,1}^*, s_{\beta,1}^*)$ to $\mathcal{A}$.

Finally, $\mathcal{A}$ outputs b'. If $\tau^* = g^{\frac{1}{x+T}}$ (or $\tau_1^* = g^{\frac{1}{x+T_1}}$), then $\mathcal{B}$ simulates Game 0, and if τ^* (or τ_1^*) is a random value, then $\mathcal{B}$ simulates Game 1. In Game 1, no information of the challenge bit b is revealed from σ^*, σ_0^*, and σ_1^*. So, $\mathcal{B}$ desides the challenge is a random value if $b' \neq b$, and it is not a random value, otherwise, and solves the SDDHI problem. We remark that $\mathcal{B}$ can revoke ID^* at a time $T' > T$ (or $T' > T_1$) using the $\mathcal{O}_x$ oracle. This concludes the proof. □

Theorem 5.2. *The proposed GS-TDL scheme has unforgeability in the random oracle model if the q-SDH assumption holds and* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *is EUF-CMA, where q is the number of signers and H is modeled as a random oracle.*

Proof. We consider the following two cases. The first one is $\mathcal{A}$ produces a valid signature although $\mathcal{A}$ does not have t_T ($(1) \wedge (2) \wedge (3)$ in the definition), and the second one is $\mathcal{A}$ produces a valid signature although $\mathcal{A}$ does not have a signing key ($(1) \wedge (2) \wedge (4)$ in the definition).

First Case: We construct an algorithm $\mathcal{B}$ that breaks EUF-CMA security of the underlying signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$. The challenger of the signature scheme runs $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^\lambda)$, and sends vk to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{tpk} := \mathsf{vk}$, runs $params \leftarrow \mathsf{Setup}(1^\lambda)$ and $(\mathsf{gpk}, \mathsf{gsk}) \leftarrow \mathsf{GKeyGen}(params)$, and sends $(\mathsf{gpk}, \mathsf{tpk})$ to $\mathcal{A}$. For a $\mathsf{TokenGen}$ query T, $\mathcal{B}$ computes $W_T = g_2^T$, sends W_T to the challenger as a signing query, and obtains Σ. $\mathcal{B}$ sets $t_T = (T, W_T, \Sigma)$, and sends t_T to $\mathcal{A}$. Since $\mathcal{B}$ has gsk, $\mathcal{B}$ can respond all AddU, GSign, and USK queries. We remark that $\mathcal{A}$ does not access the TSK oracle. Finally, $\mathcal{A}$ outputs (T, M, σ). Since σ is a valid group signature, there exist (Σ, W_T) such that W_T is used in the verification algorithm, and Σ is a valid signature under vk. That is, $\mathcal{A}$ produces $t_T = (T, W_T, \Sigma)$, and sets it via the $\mathsf{SetToken}$ oracle. Since W_T is not sent to $\mathcal{B}$ as a $\mathsf{TokenGen}$ query, $\mathcal{B}$ outputs (Σ, W_T) as a forgery of the signature scheme.

Second Case: We construct an algorithm $\mathcal{B}$ that breaks the q-SDH problem as follows. Let $(g_1, g_1^\gamma, \ldots, g_1^{\gamma^q}, g_2, g_2^\gamma)$ be an SDH instance. Here, q be the number of AddU queries. $\mathcal{B}$ runs $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^\lambda)$, and sets $\mathsf{tpk} := \mathsf{vk}$. $\mathcal{B}$ chooses $x_1, \ldots, x_q, y_1, \ldots, y_q \xleftarrow{\$} \mathbb{Z}_p$ and $\alpha, \theta \xleftarrow{\$} \mathbb{Z}_p^*$. Let define

$$f(X) = \prod_{i=1}^{q}(X + x_q) := \sum_{i=0}^{q} \alpha_i X^i$$

and

$$f_i(X) := f(X)/(X - x_i) = \prod_{j=1, j\neq i}^{q}(X + x_i) := \sum_{i=1}^{q-1} \beta_i X^i,$$

and set $g_1' = (\prod_{i=0}^{q}(g_1^{\gamma^i})_i^\alpha)^\theta = g_1^{\theta f(\gamma)}$. Then, for each $i \in [1, q]$ $(\prod_{j=0}^{q-1}(g_1^{\gamma^i})^{\beta_i})^\theta = g_1^{\theta f_i(\gamma)} = {g_1'}^{\frac{1}{\gamma + x_i}}$ hold. Set $h := {g_1'}^\alpha$. For each $i \in [1, q]$, $\mathcal{B}$ computes $A_i := ({g_1'}^{\frac{1}{\gamma + x_i}})^{1 - y_i\alpha} = (g_1' h^{-y})^{\frac{1}{\gamma + x_i}}$. $\mathcal{B}$ sets $W := g_2^\gamma$,

$params = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1', g_2)$, and $\mathsf{gpk} = (params, h, W, e(g_1', g_2), e(h, W), e(h, g_2), H)$, and gives $(\mathsf{gpk}, \mathsf{tpk})$ to $\mathcal{A}$.

For an AddU query, $\mathcal{B}$ chooses unselected $x \in \{x_1, \ldots, x_q\}$ and sets the corresponding (x, y, A) as the signing key. Since $\mathcal{B}$ has tsk, $\mathcal{B}$ can respond $\mathsf{TokenGen}$ and TSK queries. Moreover, $\mathcal{B}$ can respond GSign and Revoke queries since $\mathcal{B}$ has all signing keys (x_i, y_i, A_i) for each $i \in [1, q]$.

Finally, $\mathcal{A}$ outputs a forge group signature $\sigma = (C, \tau, c, s_x, s_\delta, s_\beta)$. $\mathcal{B}$ rewinds $\mathcal{A}$ and obtains $\sigma' = (C, \tau, c', s_x', s_\delta', s_\beta')$ where $c \neq c'$ with non-negligible probability (due to the forking lemma). Set

$$\tilde{x} := \frac{s_x - s_x'}{c - c'}, \tilde{y} := \frac{(s_x - s_x')(s_\beta - s_\beta') - (s_\delta - s_\delta')(c - c')}{(c - c')^2}, \text{ and } \tilde{\beta} := \frac{s_\beta - s_\beta'}{c - c'}.$$

Then,

$$\frac{e(C, W)}{e(g_1', g_2)} = \frac{e(h, g_2)^{\tilde{\beta}\tilde{x} - \tilde{y}} e(h, W)^{\tilde{\beta}}}{e(C, g_2)^{\tilde{x}}}$$

and

$$e(\tau, g_2)^{\tilde{x}} = \frac{e(g_1', g_2)}{e(\tau, W_T)}$$

hold. That is, $(\tilde{x}, \tilde{y}, \tilde{A})$ can be extracted. If $1 - \tilde{y}\alpha = 0$, then $\mathcal{B}$ aborts. Moreover, if $\tilde{x} \in \{x_1, \ldots, x_q\}$, then $\mathcal{B}$ aborts. Since α and all x are randomly chosen, the aborting probability is at most q/p, and is negligible. From now on, we assume that $1 - \tilde{y}\alpha \neq 0$ and $\tilde{x} \notin \{x_1, \ldots, x_q\}$. Since $\tilde{A}$ can be represented as $\tilde{A} = (g_1' h^{-\tilde{y}})^{\frac{1}{\gamma + \tilde{x}}}$, $\mathcal{B}$ can compute $\tilde{A}^{\frac{1}{1 - \tilde{y}\alpha}} = g_1'^{\frac{1}{\gamma + \tilde{x}}} = (g_1^{\theta f(\gamma)})^{\frac{1}{\gamma + \tilde{x}}}$. Next, $\mathcal{B}$ computes $F(X)$ and $\gamma_* \in \mathbb{Z}_p^*$ which satisfy $f(X) = (X + \tilde{x})F(X) + \gamma_*$. Finally, $\mathcal{B}$ computes

$$\left(((g_1^{\theta f(\gamma)})^{\frac{1}{\gamma + \tilde{x}}})^{\frac{1}{\theta}} \prod_{i=0}^{q-1} (g_1^{x^i})^{-F_i} \right)^{\frac{1}{\gamma_*}} = g_1^{\frac{1}{\gamma + \tilde{x}}},$$

where $F(X) := \sum_{i=0}^{q-1} F_i X^i$, and outputs $(\tilde{x}, g_1^{\frac{1}{\gamma + \tilde{x}}})$ as a solution of the SDH problem.

$\square$

Theorem 5.3. *The proposed GS-TDL scheme has linking soundness.*

Proof. Let $(\mathsf{ID}_0, \mathsf{ID}_1, T_0, T_1, M)$ and (M^*, σ^*) be the output of $\mathcal{A}$, where $(\mathsf{ID}_0, T_0) \neq (\mathsf{ID}_1, T_1)$. Let x_0 be contained in $\mathsf{sigk}_{\mathsf{ID}_0}$ and x_1 be contained in $\mathsf{sigk}_{\mathsf{ID}_1}$, respectively. If $\mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_T, (M, \sigma_0, T_0), (M^*, \sigma^*, T_1)) = 1$, then $g_1^{\frac{1}{x_0 + T_0}} = g_1^{\frac{1}{x_1 + T_1}}$ and $T = T_0 = T_1$ holds. Then, $x_0 = x_1$ holds. Since x_0 and x_1 are randomly chosen, this equation holds with probability at most $1/p$. This concludes the proof. $\square$

6 The WDG Construction and its Vulnerability

Wu, Domingo-Ferrer and González-Nicolás (WDG) [67] proposed message-linkable group signature (MLGS), where if a signer generates a group signature for the same message twice, then two group signatures become linkable. From MLGS, they constructed an anonymous threshold authentication for Vehicle-to-Vehicle communications, where if more than t vehicles send the same message (signed by vehicles) then this information is accepted. In their system, four entities are defined: a vehicle $\mathcal{V}$, the vehicle manufacturer $\mathcal{VM}$, the registration manager $\mathcal{RM}$, and the group tracing manager $\mathcal{TM}$. In this section, we give an attack against their MLGS scheme denoted by the WDG scheme. Briefly, we show that anyone can make a valid group signature without knowing a signing key.

The WDG scheme is described as follows: Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, h_1, h_2, U_1, U_2, H_1, H_2)$ be public parameters, where $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ are hash functions, and for a computable isomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$, $g_1 = \phi(g_2)$, $h_1 = \phi(h_2)$, and $U_1 = \phi(U_2)$ hold. A vehicle $\mathcal{V}$ has a secret key $y \in \mathbb{Z}_p$ and a public key $Y = U_1^y$. In the vehicle registration phase, $\mathcal{V}$ computes $T = g_2^y$ and sends (T, Y) to the group tracing manager $\mathcal{TM}$. $\mathcal{TM}$ checks the signature of Y and whether $e(Y, g_2) = e(U_1, T)$ holds or not. Then, $\mathcal{TM}$ saves (T, Y) into a local database. Moreover, $\mathcal{V}$ sends Y (and its signature) to the registration manager $\mathcal{RM}$, and runs a zero-knowledge protocol for $y = \log_{U_1} Y$. Then, $\mathcal{RM}$ chooses $k \leftarrow \mathbb{Z}_p$ and computes $K_1 = g_1^k$ and $K_2 = Z(h_1 Y)^{-k}$, where Z (and $A = e(Z, g_2)$) is a public key of $\mathcal{RM}$. $\mathcal{V}$ obtains $K_v = (K_1, K_2)$ as a secret signing key. In a signing phase for a message m, $\mathcal{V}$ selects $s, r_y \leftarrow \mathbb{Z}_p$ and computes $\sigma_1 = K_1 g_1^s$, $\sigma_2 = K_2(h_1 Y)^{-s}$, $\sigma_3 = \sigma_1^y$, $\sigma_4 = H_1(m)^y$, $\sigma_5 = H_2(m||\sigma_1||\sigma_2||\sigma_3||\sigma_4||H_1(m)^{r_y}||\sigma_1^{r_y})$, and $\sigma_6 = r_y - \sigma_5 y$, and outputs a group signature $\sigma = (\sigma_1, \ldots, \sigma_6)$. That is, this proves that the discrete logarithm of σ_3 and that of σ_4 (i.e., y) are the same. In the verification phase, check $e(\sigma_2, g_2)e(\sigma_1, h_2)e(\sigma_3, U_2) = A$ and $\sigma_5 = H_2(m||\sigma_1|| \sigma_2||\sigma_3||\sigma_4||H_1(m)^{\sigma_6}\sigma_4^{\sigma_5}||\sigma_1^{\sigma_6}\sigma_3^{\sigma_5})$.

The problem of this construction is the thing that no $\mathcal{RM}$'s verification key is involved in the signing and verification phases. Before showing our attack, we explain a typical methodology for constructing group signature introduced in [15, 16] as follows: A key issuer ($\mathcal{RM}$ in the MLGS context) has a verification/signing key pair (vk, sk) of a signature scheme, and an opener ($\mathcal{TM}$ in the MLGS context) has a public/secret key pair (pk, dk) of an encryption scheme. Then, the key issuer generates a signature for a user ($\mathcal{V}$ in the MLGS context) as a signing key (say $cert$). In the signing phase, a user encrypts $cert$ by using pk, and computes a non-interactive zero-knowledge proof that proves "encrypted $cert$ is a valid signature under vk". Since the WDG scheme lacks to involve vk, anyone can make a valid group signature.

The concrete attack is described as follows: let $\mathcal{A}$ be an adversary. $\mathcal{A}$ chooses $y, k, s, r_y \leftarrow \mathbb{Z}_p$ and computes $Y = U_1^y$, $\sigma_1 = g_1^k g_1^s$, $\sigma_2 = Z(h_1 Y)^{-k}(h_1 Y)^{-s}$, $\sigma_3 = \sigma_1^y$, $\sigma_4 = H_1(m)^y$, $\sigma_5 = H_2(m||\sigma_1|| \sigma_2||\sigma_3||\sigma_4||H_1(m)^{r_y}||\sigma_1^{r_y})$, and $\sigma_6 = r_y - \sigma_5 y$, and outputs a group signature $\sigma_{\text{forge}} = (\sigma_1, \ldots, \sigma_6)$. Then, σ_{forge} is a valid group signature though $\mathcal{A}$ does not have a signing key. Moreover, since $\mathcal{A}$ does not register Y, no $\mathcal{TM}$ can trace the signer of σ. This breaks traceability.

7 Anonymous Time-dependent Authentication and its Experimental Results

In this section, we construct an anonymous time-dependent authentication system via GS-TDL, and provide its experimental results.

7.1 System Architecture

In the setting described by Wu et al., [67], a threshold value t is defined and "*When t vehicles wish to endorse some message, they can independently generate an message-linkable group signature (MLGS) on that message. A verifying vehicle trusts the message after validating t MLGSs on it*". However, it seems difficult to assume that t vehicles generates signatures on the same message when local information is measured. Therefore, in our system we consider a situation in which each vehicle can send a message M (with its signature σ) once at a time T, and messages may be different from each other. Later, statistics of messages can be computed. Then, if a vehicle attempts to maliciously include multiple messages (for manipulating statistical information, for example), these messages would not be included in the statistics.

We define five entities as follows: a vehicle $\mathcal{V}$, the vehicle manufacturer $\mathcal{VM}$, the Road Side Unit $\mathcal{RSU}$, the Token Generation Unit $\mathcal{TGU}$, and a Data Collector $\mathcal{DC}$. We can consider multiple $\mathcal{RSU}$s but we assume only one $\mathcal{TGU}$ in this paper. We assume that $params \leftarrow \mathsf{Setup}(1^\lambda)$ has been honestly run, and all entities share $params$. First, $\mathcal{VM}$ runs $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grs}, \mathsf{RL}_0) \leftarrow \mathsf{GKeyGen}(params)$ and $\mathcal{TGU}$ runs $(\mathsf{tpk}, \mathsf{tsk}) \leftarrow \mathsf{TKeyGen}(params)$. When a vehicle $\mathcal{V}$ is sold, $\mathcal{VM}$ runs $\mathsf{sigk}_{\mathsf{ID}} \leftarrow \mathsf{Join}(\mathsf{gsk}, \mathsf{grs}, \mathsf{ID})$, and $\mathcal{V}$ preserves $\mathsf{sigk}_{\mathsf{ID}}$. In each time period T, $\mathcal{TGU}$ runs $t_T \leftarrow \mathsf{TokenGen}(\mathsf{tsk}, T)$ and broadcasts t_T. Moreover, $\mathcal{VM}$ updates the revocation list, and sends RL_T to all $\mathcal{RSU}$. A vehicle $\mathcal{V}$ generates a group signature on local information M such that $\sigma \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_T, \mathsf{sigk}_{\mathsf{ID}}, M)$. We assume that $\mathcal{V}$ generates σ and sends (M, σ, T) to an $\mathcal{RSU}$ when $\mathcal{V}$ enters a certain range of the $\mathcal{RSU}$. Let $\mathsf{S}_T := \{(M_1, \sigma_1), \ldots, (M_\ell, \sigma_\ell)\}$ be the storage at a time T managed by an $\mathcal{RSU}$ where $\ell \geq 0$. If (M, σ, T) where $(M, \sigma) \in \mathsf{S}_T$ is not a valid signature

Table 1: The number of operations for each algorithms.

Algorithm	Operations
GSign	2 Mul ($\mathbb{G}_1$) + 1 Mul ($\mathbb{G}_2$) + 2 Exp ($\mathbb{G}_T$) + 2 Pairing + Verify
GSign (Pairing free)	2 Mul ($\mathbb{G}_1$) + 4 Exp ($\mathbb{G}_T$) + Verify
$\mathsf{TokenGen}$	1 Mul ($\mathbb{G}_2$) + Sign
$\mathsf{GVerify}$	6 Exp ($\mathbb{G}_T$) + 4 Pairing + Verify
Revoke	$\lvert\mathsf{RL}_T\rvert$ Mul ($\mathbb{G}_1$)

on T, then the $\mathcal{RSU}$ excludes it from S_T. After T has passed, the $\mathcal{RSU}$ sends S_T to $\mathcal{DC}$. $\mathcal{DC}$ runs the $\mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_T, (M, \sigma, T), (M_i, \sigma_i, T))$ for each $i \in [1, \ell]$, and runs statistical calculations over collected M.[9]

7.2 Experimental Results

Here, we show experimental results of our prototype implementations and the practicality of our GS-TDL scheme. Our implementation uses TEPLA library [5] for elliptic curve operations and the pairing operation, OpenSSL[10] for standard signing and verifying, and GLib[11] for the hash table for (almost) constant-time searching.

We give the number of operations for each algorithms in Table 1. In the table, Mul ($\mathbb{G}_1$), Mul ($\mathbb{G}_2$) and Exp ($\mathbb{G}_T$) denote a scalar multiplication on $\mathbb{G}_1$, a scalar multiplication on $\mathbb{G}_2$ and an exponentiation on $\mathbb{G}_T$, respectively. Verify and Sign denote standard verifying and signing. We use RSA signing algorithm for them because of its efficiency in the verification. We remark that costs of all algorithms do not depend on the number of vehicles, therefore our GS-TDL scheme has good scalability. We also remark that the Revoke algorithm depends on the number of revoked vehicles $\lvert\mathsf{RL}_T\rvert$. However, since the Revoke algorithm is computed by the vehicle manufacturer $\mathcal{VM}$ periodically, like per day, the dependence does not reduce the practicality of our scheme.

Next, we give running time of basic operations of TEPLA library in Table 2. Even on Raspberry Pi, a cheap and constrained computational power device, the operations can be performed in practical running time.

Table 2: Basic Operations on BN curves [13] of 254-bit order. Operations are run over PC (Core i7-4770 with TurboBoost) and Raspberry Pi (ARM1176JZF-S) respectively.

Operation	PC (msec)	Raspberry Pi (msec)
Mul ($\mathbb{G}_1$)	0.330	9.030
Mul ($\mathbb{G}_2$)	0.540	16.620
Exp ($\mathbb{G}_T$)	2.840	78.580
Pairing	2.690	77.330

Finally, we give our experimental results of our GS-TDL scheme in Table 3. We evaluate these results as follows:

(Almost) Constant-Time Verification: First of all, we should highlight that cryptographic operations in the $\mathsf{GVerify}$ algorithm do not depend on the number of revoked vehicles (i.e., scalable) due to our time-dependent linkability, i.e., τ is deterministic, though we employ VLR-type revocation. In our implementation, a table preserves (ID, x) in the Join algorithm is regarded as grs, and ID is set as a

[9] We remark that a vehicle may send two signatures to two $\mathcal{RSU}$s at the same T. $\mathcal{DC}$ can exclude /include such signatures according to the statistical calculations.

[10] `https://www.openssl.org`

[11] `https://wiki.gnome.org/Projects/GLib`

Table 3: Benchmarks: The signing algorithms (3072-bit RSA sign, 3072-bit DSA sign, 256-bit ECDSA (prime256v1 curve) sign, and GSign) are run over Raspberry Pi (CPU: ARM1176JZF-S), and other algorithms are run over PC (Core i7-4770 CPU with TurboBoost) respectively. We use OpenSSL for RSA, DSA, and ECDSA. Operations are run over PC (Core i7-4770 with TurboBoost) and Raspberry Pi (ARM1176JZF-S). RSA sign/verify, DSA sign/verify and ECDSA sign/verify are performed with 3072-bit, 3072-bit and 256-bit on prime256v1 curve, respectively. The total number of vehicles is 10,000,000, and the number of revoked vehicles is specified in parentheses () in the $\mathsf{GVerify}$ algorithm and the Revoke algorithm. We employ BN curves [13] with 254-bit order for efficient pairings, and the hash table for (almost) constant-time searching. We also employ 3072-bit RSA as $(\mathsf{TokenGen}, \mathsf{Sign}, \mathsf{Verify})$ used in our GS-TDL scheme since the verification cost (which is run by vehicles in the GSign algorithm) is faster than that of DSA and ECDSA. We remark that the Link algorithm does not require any cryptographic operation, and RSA, DSA, and ECDSA does not support anonymity.

Algorithm	PC (msec)	Raspberry Pi (msec)	Entity
GSign	(12.573)	408.943	Vehicle
GSign (Pairing free)	(12.105)	400.302	Vehicle
RSA sign	(3.427)	233.511	Vehicle
DSA sign	(1.082)	75.135	Vehicle
ECDSA sign	(0.335)	11.702	Vehicle
$\mathsf{TokenGen}$	3.763	-	Token Generation Unit
$\mathsf{GVerify}$(1,000)	17.990	-	Road Side Unit
$\mathsf{GVerify}$(10,000)	17.997	-	Road Side Unit
$\mathsf{GVerify}$(100,000)	17.953	-	Road Side Unit
$\mathsf{GVerify}$(1,000,000)	18.049	-	Road Side Unit
RSA verify	0.072	5.043	Road Side Unit
DSA verify	1.283	87.913	Road Side Unit
ECDSA verify	0.382	13.719	Road Side Unit
Revoke(1,000)	299.829	-	Vehicle Manufacturer
Revoke(10,000)	3023.363	-	Vehicle Manufacturer
Revoke(100,000)	30270.951	-	Vehicle Manufacturer
Revoke(1,000,000)	301716.554	-	Vehicle Manufacturer

searching key (i.e., the table as takes as input ID, and outputs the corresponding x). In the Revoke algorithm, an array (ID, x, τ) is made, and τ contained in all arrays are updated on T such that $\mathsf{RL}_T := \{(\mathsf{ID}_{T,1}, g_1^{\frac{1}{x_{T,1}+T}}), \dots (\mathsf{ID}_{T,n_T}, g_1^{\frac{1}{x_{T,n_T}+T}})\}$, and the corresponding hash table is generated for (almost) constant-time searching. Therefore, the cost of the Revoke algorithm depends on the number of revoked vehicles (but we emphasize that this procedure is run by the vehicle manufacturer $\mathcal{VM}$ and is not related to vehicles). In the $\mathsf{GVerify}$ algorithm, the Road Side Unit $\mathcal{RSU}$ can easily check whether τ is contained in RL_T or not by using the hash tables without any cryptographic operation.

Practically Efficient Signing: In a usual situation, a vehicle $\mathcal{V}$ has a constrained computational power, and moreover $\mathcal{V}$ needs to generate signatures in several times. In our implementation result, the signing cost is still handled millisecond order and just twice as that of the 3072-bit RSA signing algorithm, though our system additionally supports anonymity. If $\mathcal{V}$ has a standard computational power (as in the PC), then the GSign algorithm can be run at 12.573 msec (and 12.105 msec for its pairing free version). This result shows that our system is feasible in practice.

For constructing an actual system, we need to find a suitable time period interval (e.g., 1 sec, 1 hour, 1 day, etc.). Moreover, we need to decide when vehicles generate signatures and send them to RSUs (e.g., a vehicle does when it comes within a range of a RSU, a vehicle broadcasts local info with a signature at fixed intervals, etc.). We need to investigate these values by simulation, e.g., using ns-3 [3], and by considering de-anonymization techniques, e.g., [49, 60, 31, 59]. We leave these as future works of this paper.

Acknowledgement

We would like to thank Ryo Nojima for his helpful comments and suggestions.

References

[1] IntelliDrive Program. Available at `http://www.intellidrive.org`.

[2] ITS Info-communications Forum. Available at `http://www.itsforum.gr.jp/E_index.html`.

[3] ns-3: Discrete-event Network Simulator for Internet Systems. Available at `http://www.nsnam.org/`.

[4] PRESERVE: Preparing Secure Vehicle-to-X Communication Systems. Available at `http://www.preserve-project.eu/`.

[5] TEPLA: University of Tsukuba Elliptic Curve and Pairing Library. Available at `http://www.cipher.risk.tsukuba.ac.jp/tepla/index_e.html`.

[6] M. Abe, S. S. M. Chow, K. Haralambiev, and M. Ohkubo. Double-trapdoor anonymous tags for traceable signatures. *Int. J. Inf. Sec.*, 12(1):19–31, 2013.

[7] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *CRYPTO*, pages 209–236, 2010.

[8] M. Abe, K. Haralambiev, and M. Ohkubo. Signing on elements in bilinear groups for modular protocol design. *IACR Cryptology ePrint Archive*, 2010:133, 2010.

[9] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai. A revocable group signature scheme from identity-based revocation techniques: Achieving constant-size revocation list. In *ACNS*, pages 419–437, 2014.

[10] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens. PrETP: Privacy-preserving electronic toll pricing. In *USENIX Security Symposium*, pages 63–78, 2010.

[11] F. Baldimtsi and A. Lysyanskaya. Anonymous credentials light. In *ACM Conference on Computer and Communications Security*, pages 1087–1098, 2013.

[12] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *EUROCRYPT*, pages 1–16, 2014.

[13] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, pages 319–331, 2005.

[14] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. Compact e-cash and simulatable VRFs revisited. In *Pairing*, pages 114–131, 2009.

[15] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629, 2003.

[16] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, pages 136–153, 2005.

[17] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. In *SCN*, pages 381–398, 2010.

[18] D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[19] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[20] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177, 2004.

[21] C. Busold, A. Taha, C. Wachsmann, A. Dmitrienko, H. Seudie, M. Sobhani, and A.-R. Sadeghi. Smart keys for cyber-cars: secure smartphone-based NFC-enabled car immobilizer. In *CODASPY*, pages 233–242, 2013.

[22] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clone wars: efficient periodic n-times anonymous authentication. In *ACM Conference on Computer and Communications Security*, pages 201–210, 2006.

[23] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[24] B. K. Chaurasia, S. Verma, and S. M. Bhasker. Message broadcast in VANETs using group signature. In *WCSN*, pages 131–136, 2009.

[25] C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *VIETCRYPT*, pages 193–210, 2006.

[26] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[27] K. Emura, A. Kanaoka, S. Ohta, and T. Takahashi. Building secure and anonymous communication channel: Formal model and its prototype implementation. In *ACM Symposium on Applied Computing*, pages 1641–1648, 2014.

[28] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[29] M. K. Franklin and H. Zhang. Unique group signatures. In *ESORICS*, pages 643–660, 2012.

[30] J. Furukawa and H. Imai. An efficient group signature scheme from bilinear maps. *IEICE Transactions*, 89-A(5):1328–1338, 2006.

[31] S. Gambs, M. Killijian, and M. N. del Prado Cortez. De-anonymization attack on geolocated data. In *TrustCom 2013*, pages 789–797, 2013.

[32] M. Götz, S. Nath, and J. Gehrke. Maskit: privately releasing user context streams for personalized mobile applications. In *ACM SIGMOD*, pages 289–300, 2012.

[33] R. Granger, T. Kleinjung, and J. Zumbrägel. Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot1223}}$ and $\mathbb{F}_{2^{12\cdot367}}$). In *CRYPTO (2)*, pages 126–145, 2014.

[34] J. Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT*, pages 164–180, 2007.

[35] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[36] J. Guo, J. P. Baugh, and S. Wang. A Group Signature Based Secure and Privacy-Preserving Vehicular Communication Framework. In *Mobile Networking for Vehicular Environments*, pages 103–108, 2007.

[37] J. Y. Hwang, S. Lee, B.-H. Chung, H. S. Cho, and D. Nyang. Short group signatures with controllable linkability. In *LightSec*, pages 44–52, 2011.

[38] J. Y. Hwang, S. Lee, B.-H. Chung, H. S. Cho, and D. Nyang. Group signatures with controllable linkability for dynamic membership. *Inf. Sci.*, 222:761–778, 2013.

[39] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT*, pages 571–589, 2004.

[40] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *Public Key Cryptography*, pages 345–361, 2014.

[41] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *ICDE*, pages 106–115, 2007.

[42] B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. In *CRYPTO*, pages 571–589, 2012.

[43] B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In *EUROCRYPT*, pages 609–627, 2012.

[44] B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *CANS*, pages 498–517, 2009.

[45] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.

[46] M. S. I. Mamun and A. Miyaji. Group signature with relaxed-privacy and revocability for VANET. *IACR Cryptology ePrint Archive*, 2013:804, 2013.

[47] M. S. I. Mamun and A. Miyaji. Secure VANET applications with a refined group signature. In *PST*, pages 199–206, 2014.

[48] S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham. The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion. In *USENIX Security Symposium*, 2011.

[49] T. Murakami and H. Watanabe. Location prediction attacks using tensor factorization and optimal defenses. In *IEEE International Conference on Big Data*, pages 13–21, 2014.

[50] T. Nakanishi, T. Fujiwara, and H. Watanabe. A linkable group signature and its application to secret voting. *JIP*, 40(7):3085–3096, 1999.

[51] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In *ASIACRYPT*, pages 533–548, 2005.

[52] T. Nakanishi and N. Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In *IWSEC*, pages 17–32, 2006.

[53] G. Ohtake, A. Fujii, G. Hanaoka, and K. Ogawa. On the theoretical gap between group signatures with and without unlinkability. In *AFRICACRYPT*, pages 149–166, 2009.

[54] R. A. Popa, H. Balakrishnan, and A. J. Blumberg. VPriv: Protecting privacy in location-based vehicular services. In *USENIX Security Symposium*, pages 335–350, 2009.

[55] B. Qin, Q. Wu, J. Domingo-Ferrer, and L. Zhang. Preserving security and privacy in large-scale VANETs. In *ICICS*, pages 121–135, 2011.

[56] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565, 2001.

[57] I. Rouf, R. D. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *USENIX Security Symposium*, pages 323–338, 2010.

[58] Y. Sakai, J. C. N. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *Public Key Cryptography*, pages 715–732, 2012.

[59] R. Shokri, G. Theodorakopoulos, J. L. Boudec, and J. Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*, pages 247–262, 2011.

[60] R. Shokri, G. Theodorakopoulos, C. Troncoso, J. Hubaux, and J. L. Boudec. Protecting location privacy: optimal strategy against localization attacks. In *ACM Conference on Computer and Communications Security*, pages 617–627, 2012.

[61] Y. Sun, Z. Feng, Q. Hu, and J. Su. An efficient distributed key management scheme for group-signature based anonymous authentication in VANET. *Security and Communication Networks*, 5(1):79–86, 2012.

[62] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE T. Vehicular Technology*, 59(7):3589–3603, 2010.

[63] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[64] S. Tillich and M. Wójcik. Security analysis of an open car immobilizer protocol stack. In *INTRUST*, pages 83–94, 2012.

[65] T. M. Truta, A. Campan, and P. Meyer. Generating microdata with p-sensitive k-anonymity property. In *Secure Data Management, 4th VLDB Workshop, SDM 2007, Vienna, Austria, September 23-24, 2007, Proceedings*, pages 124–141, 2007.

[66] J. Wetzels. Broken keys to the kingdom: Security and privacy aspects of RFID-based car keys. *CoRR*, abs/1405.7424, 2014.

[67] Q. Wu, J. Domingo-Ferrer, and Ú. González-Nicolás. Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications. *IEEE T. Vehicular Technology*, 59(2):559–573, 2010.

[68] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang. Probabilistic adaptive anonymous authentication in vehicular networks. *J. Comput. Sci. Technol.*, 23(6):916–928, 2008.

[69] Y. Xi, W. Shi, and L. Schwiebert. Mobile anonymity of dynamic groups in vehicular networks. *Security and Communication Networks*, 1(3):219–231, 2008.

[70] M. Xu, W. Xu, J. Walker, and B. Moore. Lightweight secure communication protocols for in-vehicle sensor networks. In *CyCAR*, pages 19–30, 2013.

Appendix: A GS-TDL Scheme Secure in the Standard Model

In this section, we briefly introduce a GS-TDL scheme secure in the standard model. We apply the Abe-Haralambiev-Ohkubo (AHO) signature [7, 8] and Groth-Sahai proofs [35] as in group signature schemes secure in the standard model, e.g., [34, 9, 42, 43].

First, we introduce the AHO signature scheme [8]. Let $\mathsf{pp} = ((\mathbb{G}, \mathbb{G}_T), g)$ and $n \in \mathbb{N}$ be an upper bound on the number of group elements that can be signed altogether. In our group signature, we set $n = 1$.

$\mathsf{KeyGen}(\mathsf{pp}, n)$: Choose $G_r, H_r \xleftarrow{\$} \mathbb{G}$, $\gamma_z, \delta_z \xleftarrow{\$} \mathbb{Z}_p$, and $\gamma_i, \delta_i \xleftarrow{\$} \mathbb{Z}_p$ for $i = 1, \ldots, n$. Compute $G_z = G_r^{\gamma_z}$, $H_z = H_r^{\delta_z}$, $G_i = G_r^{\gamma_i}$, and $H_i = H_r^{\delta_i}$ for $i = 1, \ldots, n$, and compute $\alpha_a, \alpha_b \xleftarrow{\$} \mathbb{Z}_p$, $A = e(G_r, g^{\alpha_a})$, and $B = e(H_r, g^{\alpha_b})$. Output $pk = (G_r, H_r, G_z, H_z, \{G_i, H_i\}_{i=1}^n, A, B) \in \mathbb{G}^{2n+4} \times \mathbb{G}_T^2$ and $sk = (\alpha_a, \alpha_b, \gamma_z, \delta_z, \{\gamma_i, \delta_i\}_{i=1}^n)$.

$\mathsf{Sign}(sk, (M_1, \ldots, M_n))$: Choose $\zeta, \rho, \tau, \nu, \omega \xleftarrow{\$} \mathbb{Z}_p$, and output a signature $\theta = (\theta_1, \ldots, \theta_7)$ where $\Big(\theta_1 = g^{\zeta}, \theta_2 = g^{\rho - \gamma_z \xi} \cdot \prod_{i=1}^{n} M_i^{-\gamma_i}, \theta_3 = G_r^{\tau}, \theta_4 = g^{(\alpha_a - \rho)/\tau}, \theta_5 = g^{\nu - \delta_z \xi} \cdot \prod_{i=1}^{n} M_i^{-\delta_i}, \theta_6 = H_r^{\omega}, \theta_7 = g^{(\alpha_b - \nu)/\omega}\Big)$.

$\mathsf{Verify}(pk, \theta, (M_1, \ldots, M_n))$: Check the equations $A = e(G_z, \theta_1)e(G_r, \theta_2)e(\theta_3, \theta_4)\prod_{i=1}^{n} e(G_i, M_i)$ and $B = e(H_z, \theta_1)e(H_r, \theta_5)e(\theta_6, \theta_7)\prod_{i=1}^{n} e(H_i, M_i)$. If both equations hold, then output 1, and 0 otherwise.

The AHO signature scheme is existential unforgeable under the q-SFP (Simultaneous Flexible Pairing) assumption. We remark that the AHO signature scheme supports a re-randomization algorithm ReRand, where for an AHO signature θ, let $\{\theta'_i\}_{i=1}^{7} \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}, \theta)$ be a result of re-randomization. Then, $\{\theta'_i\}_{i\in\{3,4,6,7\}}$ are independent of the corresponding signed message, and therefore $\{\theta'_i\}_{i\in\{3,4,6,7\}}$ can be directly included into a part of a group signature.

Next, we introduce Groth-Sahai proof systems [35] as follows. Let A, B be equal-dimension vectors or matrices containing group elements. Then $A \odot B$ denotes their entry-wise product. Let $\mathbf{f} := (\vec{f_1}, \vec{f_2}, \vec{f_3}) \in \mathbb{G}^3 \times \mathbb{G}^3 \times \mathbb{G}^3$ be a common reference string (CRS) s.t. $\beta_1, \beta_2, \xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $f_1 = g^{\beta_1}$, $f_2 = g^{\beta_2}$, $\vec{f_1} = (f_1, 1, g)$ and $\vec{f_2} = (1, f_2, g)$. In the perfectly sound proof setting, $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$ where $\xi_1, \xi_2 \in \mathbb{Z}_p^*$. To commit a group element $X \in \mathbb{G}$, compute commitments $\vec{C} = (1, 1, X) \odot \vec{f_1}^{r} \odot \vec{f_2}^{s} \odot \vec{f_3}^{t}$ with $r, s, t \xleftarrow{\$} \mathbb{Z}_p^*$, which is a ciphertext of the Boneh-Boyen-Shacham linear encryption scheme. In the witness indistinguishability (WI) setting, $\vec{f_1}, \vec{f_2}, \vec{f_3}$ are linearly independent. Then, $\vec{C}$ is a perfectly hiding commitment. To commit a scalar $x \in \mathbb{Z}_p$, compute $\vec{C} = \vec{\varphi}^{x} \odot \vec{f_1}^{r} \odot \vec{f_2}^{s}$ with $r, s \xleftarrow{\$} \mathbb{Z}_p^*$. In the perfectly sound proof setting, $\vec{\varphi} = \vec{f_3} \odot (1, 1, g)$ where $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$ for $\xi_1, \xi_2 \in \mathbb{Z}_p^*$. Then $\vec{\varphi}, \vec{f_1}, \vec{f_2}$ are linearly independent. In the WI setting, $\vec{\varphi} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$ for $\xi_1, \xi_2 \in \mathbb{Z}_p^*$.

Groth-Sahai proofs prove that the committed values satisfy pairing-product equations $\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \cdot \prod_{i=1}^{n} \cdot \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{i,j}} = t_T$ for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, constants $t_T \in \mathbb{G}_T$, $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathbb{G}$, $a_{i,j} \in \mathbb{Z}_p$ for $i, j \in \{1, \ldots, n\}$. Groth-Sahai proofs also follow multi-exponentiation equations $\prod_{i=1}^{m} \mathcal{A}_i^{y_i} \cdot \prod_{i=1}^{n} \mathcal{X}_j^{b_j} \cdot \prod_{i=1}^{m} \cdot \prod_{i=1}^{n} \mathcal{X}_j^{y_i \gamma_{ij}} = T$ for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$, $y_1, \ldots, y_m \in \mathbb{Z}_p$, and constants $T, \mathcal{A}_1, \ldots, \mathcal{A}_m \in \mathbb{G}$, $b_1, \ldots, b_n \in \mathbb{Z}_p$ and γ_{ij} for $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. Proofs for quadratic equations require 9 group elements, proofs for linear equations require 3 group elements, and proofs for linear multi-exponentiation equations require 2 group elements.

Next, we give our GS-TDL scheme secure in the standard model as follows. This scheme is secure under the q-SFP, decision linear (DLIN), q-SDH, and SDDHI assumptions. Briefly, a signature contains Groth-Sahai commitments and proofs, and a part of AHO signature (which is independent of a user-related value), and $\tau = g^{\frac{1}{x+T}}$ which can be seen as a random value under the SDDHI assumption. Moreover, as in the proof of Theorem 5.2, unforgeability holds if the underlying signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ and the AHO signature scheme are unforgeable. Linking soundness clearly holds. The security proofs will be available in the full version of this paper.

Construction 2 (Our GS-TDL Scheme Secure in the Standard Model)**.**

$\mathsf{Setup}(1^\lambda)$**:** *Run* $(p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathcal{G}(1^\lambda)$. *Output* $params = (\mathbb{G}, \mathbb{G}_T, e, g)$.

$\mathsf{GKeyGen}(params)$**:** *Generate a key pair* $(sk_{\mathsf{AHO}}, pk_{\mathsf{AHO}})$ *for the AHO signature in order to sign one group element, where* $pk_{\mathsf{AHO}} = (G_r, H_r, G_z, H_z, G_1, H_1, A, B)$ *and* $sk_{\mathsf{AHO}} = (\alpha_a, \alpha_b, \gamma_z, \delta_z, \gamma_1, \delta_1)$. *Select a CRS for non-interactive witness indistiguishable (NIWI) proof system:* $\mathbf{f} := (\vec{f_1}, \vec{f_2}, \vec{f_3}) \in \mathbb{G}^3 \times \mathbb{G}^3 \times \mathbb{G}^3$ *s.t.* $\beta_1, \beta_2, \xi_1, \xi_2 \xleftarrow{\$} \mathbb{Z}_p^*$, $f_1 = g^{\beta_1}$, $f_2 = g^{\beta_2}$, $\vec{f_1} = (f_1, 1, g)$, $\vec{f_2} = (1, f_2, g)$, *and* $\vec{f_3} = \vec{f_1}^{\xi_1} \odot \vec{f_2}^{\xi_2}$. $\vec{\varphi} = \vec{f_3} \odot (1, 1, g)$ *is also defined. Output* $\mathsf{gpk} = (params, pk_{\mathsf{AHO}}, \mathbf{f}, \vec{\varphi})$, $\mathsf{gsk} = sk_{\mathsf{AHO}}$, $\mathsf{grs} := \emptyset$ *and* $\mathsf{RL}_0 := \emptyset$.

$\mathsf{TKeyGen}(params)$**:** *Let* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ *be a digital signature scheme. Run* $(\mathsf{vk}, \mathsf{sigk}) \leftarrow \mathsf{Gen}(1^\lambda)$, *and output* $\mathsf{tpk} := \mathsf{vk}$ *and* $\mathsf{tsk} := \mathsf{sigk}$.

$\mathsf{Join}(\mathsf{gsk}, \mathsf{grs}, \mathsf{ID})$**:** *Choose* $x \xleftarrow{\$} \mathbb{Z}_p$, *compute* $X = g^x$, *and generate an AHO signature* $\theta = (\theta_1, \ldots, \theta_7)$ *on* X *by using* sk_{AHO}. *Output* $\mathsf{sigk}_{\mathsf{ID}} = (x, \theta)$, *and update* $\mathsf{grs} := \mathsf{grs} \cup \{(\mathsf{ID}, x)\}$.

$\mathsf{TokenGen}(\mathsf{tsk}, T)$: *Assume that* $T \in \mathbb{Z}_p$. *Compute* $W_T = g^T$ *and* $\Sigma \leftarrow \mathsf{Sign}(\mathsf{sigk}, W_T)$, *and output* $t_T = (T, W_T, \Sigma)$.

$\mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_T, \mathsf{sigk}_{\mathsf{ID}}, M)$: *Let* $\mathsf{sigk}_{\mathsf{ID}} = (x, \theta)$ *and* $t_T = (T, W_T, \Sigma)$. *If* $\mathsf{Verify}(\mathsf{vk}, W_T, \Sigma) = 0$, *then output* $\perp$. *Otherwise, compute* $\{\theta'_i\}_{i=1}^7 \leftarrow \mathsf{ReRand}(pk_{\mathsf{AHO}}, \theta)$. *Compute Groth-Sahai commitments* com_X *and* $\{com_{\theta'_i}\}_{i\in\{1,2,5\}}$, *and compute a NIWI proof* π_θ *which provides evidence that* $A = e(G_z, \theta'_1)e(G_r, \theta'_2)e(\theta'_3, \theta'_4)e(G_1, X)$ *and* $B = e(H_z, \theta'_1)e(H_r, \theta'_5)e(\theta'_6, \theta'_7)e(H_1, X)$. *Since* $\{\theta'_i\}_{i\in\{3,4,6,7\}}$ *are constants, the above equations are both linear and require 3 elements each. That is,* π_θ *contains 6 group elements. Next, compute* $\tau = g^{\frac{1}{x+T}}$, *and compute a NIZK proof* π_X *which provides evidence that* $e(\tau, XW_T) = e(g, g)$. *Since this equation is quadratic,* π_X *requires 9 group elements. Output* $\sigma = (\tau, com_X, \{com_{\theta'_i}\}_{i\in\{1,2,5\}}, \pi_\theta, \pi_X)$ *which contains 28 group elemets.*

$\mathsf{Revoke}(\mathsf{gpk}, \mathsf{grs}, \{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\})$: *If there exists* $\mathsf{ID} \in \{\mathsf{ID}_{T,1}, \ldots, \mathsf{ID}_{T,n_T}\}$ *that is not joined to the system via the* Join *algorithm, then output* $\perp$. *Otherwise, extract* $(\mathsf{ID}_{T,1}, x_{T,1}), \ldots, (\mathsf{ID}_{T,n_T}, x_{n_T})$ *from* grs. *Output* $\mathsf{RL}_T := \{(\mathsf{ID}_{T,1}, g^{\frac{1}{x_{T,1}+T}}), \ldots (\mathsf{ID}_{T,n_T}, g^{\frac{1}{x_{T,n_T}+T}})\}$.

$\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_T, M, \sigma)$: *Assume that* $\mathsf{Verify}(\mathsf{vk}, W_T, \Sigma) = 1$ *(if not, output* $\perp$*). Parse* $\sigma = (\tau, com_X, \{com_{\theta'_i}\}_{i\in\{1,2,5\}}, \pi_\theta, \pi_X)$. *If* τ *is contained in* RL_T *such that* $(\mathsf{ID}, \tau) \in \mathsf{RL}_T$ *for some* ID, *then output 0. Otherwise, return 1 if all proofs properly verify. Otherwise, return 0.*

$\mathsf{Link}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_T, (M_0, \sigma_0, T_0), (M_1, \sigma_1, T_1))$: *Parse* $\sigma_0 = (\tau_0, com_{X,0}, \{com_{\theta'_i,0}\}_{i\in\{1,2,5\}}, \pi_{\theta,0}, \pi_{X,0})$ *and* $\sigma_1 = (\tau_1, com_{X,1}, \{com_{\theta'_i,1}\}_{i\in\{1,2,5\}}, \pi_{\theta,1}, \pi_{X,1})$. *If either* $T \neq T_0$ *or* $T \neq T_1$, *then output* 0. *Else if either* $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T_0}, M_0, \sigma_0) = 0$ *or* $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T_0}, M_1, \sigma_1) = 0$, *then output 0. Otherwise, output 1 if* $\tau_0 = \tau_1$, *and 0 otherwise.*