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

Keita Emura[†] Takuya Hayashi[*]

July 6, 2016

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 via group signatures with time-token dependent linking (GS-TDL). Briefly, a vehicle is unlinkable unless it generates multiple signatures at the same time period. We provide our experimental results (using the RELIC library on a cheap and constrained computational power device, Raspberry Pi), and simulate our system by using a traffic simulator (PTV), a radio wave propagation analysis tool (RapLab), and a network simulator (QualNet). Though a similar functionality of time-token dependent linking was proposed by Wu, Domingo-Ferrer and González-Nicolás (IEEE T. Vehicular Technology 2010), we can show an attack against the scheme where anyone can forge a valid group signature without using a secret key. In contrast, our GS-TDL scheme is provably secure.

In addition to time-dependent linking property, our GS-TDL scheme supports verifier-local revocation (VLR), where a signer (vehicle) is not involved in the revocation procedure. It is particularly worth noting that no secret key or certificate of a signer (vehicle) needs to be updated whereas Security Credential Management System (SCMS) needs to update certificates frequently for vehicle privacy. Moreover, our technique maintains constant signing and verification costs by using the linkable part of signatures. These appear to be related to independent interests.

1 Introduction

Location privacy is widely recognized as an important issue, especially in the case of motor vehicles. For example, Rouf et al. [61] mentioned that location privacy could be compromised via a tire pressure monitoring system, and Xu et al. [72] proposed a secure communication protocol as a countermeasure against this vulnerability. However, simultaneously 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.

Let 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

[*]An extended abstract appears in the 4th International Workshop on Lightweight Cryptography for Security and Privacy (LightSec 2015) [27]. This is the full version. In the conference version, we provided the GS-TDL part, and used the TEPLA library [4] for estimating the efficiency of our GD-TDL scheme. In this submitted version, we use the RELIC library and reconsider the pairing equations in our algorithm. Moreover, we simulate our system using PTV, RapLab, and QualNet.

[†]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

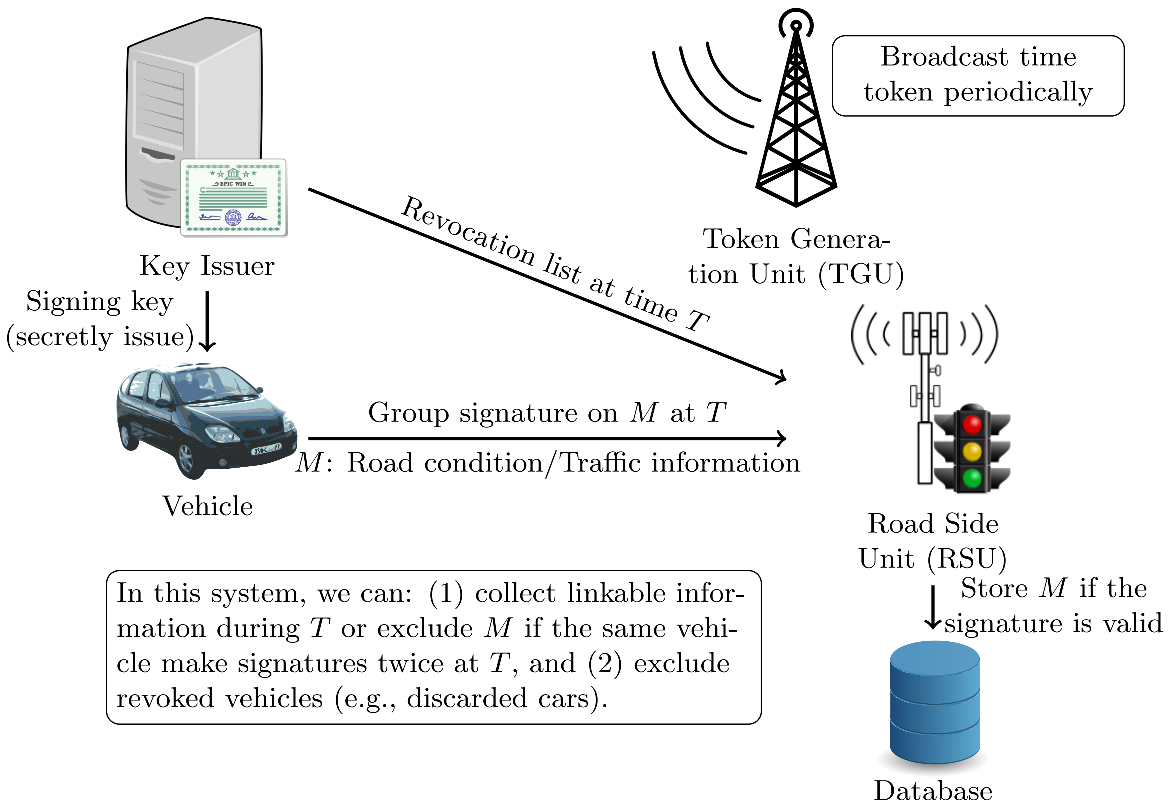
Broadcast time token periodically
Key Issuer
Revocation list at time T
Token Generation Unit (TGU)
Signing key (secretly issue)
Group signature on M at T
M: Road condition/Traffic information
Vehicle
Road Side Unit (RSU)
Store M if the signature is valid
In this system, we can: (1) collect linkable information during T or exclude M if the same vehicle make signatures twice at T, and (2) exclude revoked vehicles (e.g., discarded cars).
Database

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

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. As a countermeasure, Whyte et al. provided Security Credential Management System (SCMS) [68], where pseudonym certificate authority issues certificates to vehicles frequently to prevent linking vehicles. However, since vehicles are required to update their certificates frequently, it is hard to manage securely in practice. Applying group signatures [22] could be a solution where a verifier can anonymously verify a signer (vehicle), and there have been numerous studies, e.g., [23, 36, 69, 60, 52, 62, 63, 71, 70]. In particular, Wu, Domingo-Ferrer and González-Nicolás [69] 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.

Our Target: 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 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. 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.

One may think that the Wu et al. MLGS scheme [69] could be applicable for constructing a privacy-preserving system, however, we note that no formal security definition for MLGS is provided in [69] and the security proofs are informal. As a result, we can show an attack against the MLGS scheme of Wu et al., where anyone can forge a valid group signature without using a secret key (See Section 3). Moreover, we need to consider revocation in order to exclude discarded cars, cars with malicious behaviors, and so on.

Our Contribution: In this paper, we propose a road-to-vehicle communication system with relaxed anonymity by considering time-dependent linking properties, where a vehicle is unlinkable unless it generates multiple signatures at the same time period T. We assume that a Token Generation Unit (TGU) generates a time-dependent token t_T at a time T, and broadcasts t_T. Each signer (vehicle), who has a unique identity ID, generates a signature σ on a message M (local information) 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. As a remark, no secure channel is required for issuing a token t_T, e.g., via GPS systems or more simply via RSU in the road-to-vehicle communications context. We give a brief description of our system in Fig 1. Time-dependent linking appears to be more suitable for our system than message-dependent linking [69] where a vehicle is always linkable if it generates group signatures on the same message twice, and this situation might occur when a vehicle is used for work trips and uses the same road each day.[1]

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, a signer is unlinkable unless it generates multiple signatures at the same time period T. If T is set as short as possible, then GS-TDL comes close to usual group signatures, i.e., a signer is always unlinkable, whereas if T is set as long as possible, then GS-TDL comes close to usual digital signatures, i.e., a signer is always linkable. That is, GS-TDL can controll how long linkable information can be corrected by setting T (in our network simulation (Section 5.2) we set T as 10 minutes). Our GS-TDL supports verifier-local revocation (VLR) [48, 43, 57, 56, 18], where no signer is involved in the revocation procedure. In particular, our GS-TDL achieves backward unlinkability [56, 57], which prevents adversaries from breaking anonymity, even after vehicles are revoked. It is particularly worth noting that no secret key or certificate of a signer (vehicle) needs to be updated whereas SCMS [68] needs to update certificates frequently for vehicle privacy. Moreover, 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.

We also 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 RELIC library [7] [2]. We also simulate our system by using a traffic simulator (PTV) [3], a radio wave propagation analysis tool (RapLab) [4], and a network simulator (QualNet) [5].

Our GS-TDL is secure in the random oracle model, because we pursue a light-weight implementation of the system, under the q-strong Diffie-Hellman (q-SDH) assumption [16] and the strong decisional Diffie-Hellman inversion (SDDHI) assumption [21, 30]. The group signature size in our scheme 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 [17] contains 9 group elements, that of the short controllable linkable group signature scheme [38] contains 8 group elements, and that of the controllable linkable group signature scheme (for dynamic group setting) [39] contains 12 group elements. Recently, though a short dynamic controllable linkable group signature scheme is proposed [37], a signature contains 8 group elements.[6] In addition to the signature size, our linking

[1]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.

[2]We used the TEPLA library [4] in the previous paper [27]. We reconsider the pairing equations in our algorithm according to the RELIC library.

[3]http://vision-traffic.ptvgroup.com/

[4]http://www.kke.co.jp/en/solution/theme/raplab.html

[5]http://web.scalable-networks.com/content/qualnet

[6]We remark that these schemes [17, 37, 38, 39] also achieve only CPA-anonymity (i.e., no opening oracle access is

algorithm does not require cryptographic computations (i.e., comparisons to determine two elements are the same).

Related Work: Since car security has been recognized as a real threat, several organizations have been launched, e.g., Preserve (EU) [3], ITS Info-communications Forum (Japan) [2], IntelliDrive (USA) [1], etc., and car security is researched in several papers. To name a few, Wetzels [67] reported security and privacy concerns regarding RFID-based car key applications. Busold et al. [20] proposed a security framework for secure smartphone-based immobilizers. Tillich and Wójcik [65] analyzed an open car immobilizer protocol stack and shows several attacks. Meiklejohn et al. [54] pointed out driving payment systems proposed in [9, 59] 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.

Nakanishi et al. proposed linkable group signature [55], 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 [52, 53]. 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. [5] proposed double-trapdoor anonymous tags which can generally construct traceable signatures [41]. 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 [39, 38], where a link key is defined for the linking procedure. However, pairing computations are required for linking, which lead to inefficiency.

Kumar et al. propose a group signature scheme with probabilistic revocation [42]. In this scheme, a token (which they call alias token) is contained in a group signature, and the same token is used when group signatures are generated in the same time period, i.e., these are linkable as in our GS-TDL. Though their scheme could be a candidate for constructing an anonymous time-dependent authentication system, there are many problems as follows. First of all, their security model does not consider revocation. That is, there is a possibility that, for example, anonymity might be broken if some other users are revoked. Moreover, their security model also does not consider time-dependent linking. Whereas we consider both revocation and time-dependent linking in our security model. Even if we turn blind eye to the problem since we cannot find any attack of this scheme, the next problem is efficiency. That is, a group signature of the Kumar et al. scheme contains 9 group elements, whereas that of our scheme is 6 group elements.

As an independent work, Liu et al. [49] propose an anonymous authentication roaming protocol supporting time-bound credentials, where a user secret key is bounded to an expiry time. Though this time-bound property might be applicable to time-dependent linking, their protocol has a disadvantage where the size of the common group public key, which is necessary for generating signatures, depends on the bit-length for representing a time period. In the road-to-vehicle communication context, each vehicle is required to have such a long-size pubic key. Whereas, the size of group pubic key of our scheme is constant. Malina, Hajny, and Zeman [51] also consider group signatures with time-bound membership. One drawback of their scheme is that a target group element (i.e., an element of $\mathbb{G}_T$) is contained in a group signature. Due to the embedded degree of elliptic curves, the size of such an element is much larger than that of a base group element (i.e., an element of $\mathbb{G}_1$ or $\mathbb{G}_2$). Whereas, in our scheme, a group signature contains base group elements or $\mathbb{Z}_p$ elements only.

Remark: Since we mainly focus on how to control anonymity in the group signature context, we do not discuss how to provide location privacy of messages to be signed. For example, anonymity is never guaranteed if a message M to be signed contains the identifier of a vehicle itself. Even if we avoid such an extreme case, some personal information might be infrenged from M even we guarantee that the corresponding group signature does not leak any personal information. Here, we need to assume that no

allowed in anonymity game) as in ours.

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., [64, 50, 44, 66, 32, 26], we regard it as an outscope of this paper.

2 Preliminaries: Cryptographic Definitions

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, for any $a, b \in \mathbb{Z}_p$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ holds. We use the asymmetric setting, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$.

Definition 2.1 (SDDHI Assumption [21]). *We say that the SDDHI (Strong Decisional Diffie-Hellman Inversion) assumption holds if for all PPT adversaries $\mathcal{A}$,* $|\Pr[(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \xleftarrow{\$} \mathcal{G}(1^\lambda);\ x \xleftarrow{\$} \mathbb{Z}_p;\ (T, st) \leftarrow \mathcal{A}^{\mathcal{O}_x}(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, g_1^x);\ \tau_0 = g_1^{\frac{1}{x+T}};\ \tau_1 \xleftarrow{\$} \mathbb{G}_1;\ b \xleftarrow{\$} \{0,1\};\ b' \leftarrow A^{\mathcal{O}_x}(\tau_b, st) :\ b = b'] - \frac{1}{2}|$ *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 [16]). *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, 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, $\Pr[\mathsf{Verify}(\mathsf{vk}, \mathsf{Sign}(\mathsf{sigk}, M), M) = 1] = 1$ 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 The WDG Construction and its Vulnerability

In IEEE T. Vehicular Technology 2010, Wu, Domingo-Ferrer and González-Nicolás (WDG) [69] 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 forge 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 $Y = U_1^y$, and Y is signed by the vehicle manufacturer $\mathcal{VM}$. 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 $K_2 = Z(h_1Y)^{-k}$, 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 $\sigma_1 = K_1 g_1^s$, $\sigma_2 = K_2(h_1Y)^{-s}$, $\sigma_3 = \sigma_1^y$, $\sigma_4 = H_1(M)^y$, $\sigma_5 = H_2(M||\sigma_1||\sigma_2||\sigma_3||\sigma_4||H_1(M)^{r_y}||\sigma_1^{r_y})$, and $\sigma_6 = r_y - \sigma_5 y$, and outputs a group signature $\sigma = (\sigma_1, \dots, \sigma_6)$. 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 [13, 14] 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 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 a public key of the opener, and computes a non-interactive zero-knowledge (NIZK) proof that proves "encrypted *cert* is a valid signature under vk".

Since the WDG scheme lacks to involve vk, anyone can forge 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 $Y = U_1^y$, $\sigma_1 = g_1^k g_1^s$, $\sigma_2 = Z(h_1Y)^{-k}(h_1Y)^{-s}$, $\sigma_3 = \sigma_1^y$, $\sigma_4 = H_1(M)^y$, $\sigma_5 = H_2(M||\sigma_1||\sigma_2||\sigma_3||\sigma_4||H_1(M)^{r_y}||\sigma_1^{r_y})$, and $\sigma_6 = r_y - \sigma_5 y$, and outputs a group signature $\sigma_{\text{forge}} = (\sigma_1, \dots, \sigma_6)$. 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.

4 GS-TDL: Definitions and Our Construction

In this section, we give the syntax, security definitions, and construction of GS-TDL.

4.1 Definitions of GS-TDL

Design Principle: Because we pursued a light-weight implementation of the system[7], there is a room for discussion about whether the open functionality should be utilized or not. 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 [31] can be reduced by 50% if the open functionality is removed [28]. It also has 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 [15]. 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 signer is embedded in a device during the setup phase, and therefore we remove an interactive join algorithm from our syntax. Finally, we consider the revocation functionality, especially verifier-local revocation (VLR) where no signer is involved in revocation procedures.

[7]We note that MLGS [69] is essentially the same as unique group signature proposed by Franklin and Zhang [30], and formal security definitions are given in [30]. That is, we may be able to apply unique group signature to construct a road-to-vehicle communication system. 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 considers more stronger anonymity (so called 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.

Definition 4.1 (Syntax of GS-TDL). *A group signature scheme with time-token dependent linking $\mathcal{GS}$-$\mathcal{TDL}$ 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 inputs* tsk *and a time T, and outputs a token t_T.*
- GSign*: The signing algorithm takes as inputs* gpk, tpk, t_T, $\mathsf{sigk}_{\mathsf{ID}}$, *and a message M to be signed, and outputs a group signature σ.*
- Revoke*: The revocation algorithm takes as inputs* 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 revoked users on T.*
- $\mathsf{GVerify}$*: The verification algorithm takes as inputs* gpk, tpk, RL_T, σ, *and M, and outputs* 0 (invalid) *or* 1 (valid).
- Link*: The linking algorithm takes as inputs* 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 correctness, anonymity, unforgeability, and linking soundness. Briefly, correctness requires that 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. Anonymity 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. Unforgeability guarantees that nobody who does not have a signing key or does not have a token can generate a valid signature. Linking soundness guarantees that the Link algorithm does not return 1 when two valid signatures are made by either different signers or different time tokens.

Next, we give security definitions of GS-TDL by adding the above time-dependent linkability to the Bellare-Micciancio-Warinschi (BMW) model [13] (which is recognized as a de-facto standard for group signature).[8] 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.[9]

[8]Recently, Bootle et al. [19] consider fully dynamic group signatures where users can join and leave at any time. They point out that previous definitions are weak in the sense that these definitions allow members who joined at recent time periods to sign messages w.r.t earlier time periods during which they were not members of the group. Though this situation may be a problem in some applications (e.g., sign a document), however, in our usage, it seems easy to ignore such signatures by a verifier (Road Side Unit) if signatures sent from vehicles are generated in a past time period.

[9]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 4.2 (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.*

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

- 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 $\{\mathsf{ID}_{T,1}, \dots, \mathsf{ID}_{T,n_T}\}$, this oracle runs $\mathsf{RL}_T \leftarrow \mathsf{Revoke}(\mathsf{gpk}, \mathsf{grs}, \{\mathsf{ID}_{T,1}, \dots, \mathsf{ID}_{T,n_T}\})$. 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.

Definition 4.3 (Anonymity)**.** *For any PPT adversary $\mathcal{A}$ and a security parameter $\lambda \in \mathbb{N}$, we define the experiment $\mathrm{Exp}^{anon\text{-}tg\text{-}b}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$ as follows.*

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

- 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}.[10]

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

When $T_0 = T_1$, our definition guarantees that two different signers 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 signer is still unlinkable if the signer 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.

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 4.4 (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 $\mathcal{O} := (\mathsf{AddU}(\cdot), \mathsf{Revoke}(\mathsf{grs}, \cdot), \mathsf{TokenGen}(\mathsf{tsk}, \cdot), \mathsf{SetToken}(\cdot), \mathsf{GSign}(\cdot,\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{TSK}(\cdot))$.*

```
$\mathrm{Exp}^{unf}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$ :
    $params \leftarrow \mathsf{Setup}(1^\lambda)$
    $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grs}, \mathsf{RL}_0) \leftarrow \mathsf{GKeyGen}(params)$
    $(\mathsf{tpk}, \mathsf{tsk}) \leftarrow \mathsf{TKeyGen}(params)$
    $\mathsf{GU} := \emptyset;\ \mathsf{TSet} := \emptyset;\ \mathsf{SSet} := \emptyset$
    $(M, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{gpk}, \mathsf{tpk})$
```

[10]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.

```
Return 1 if (1) $\wedge$ (2) $\wedge$ ((3) $\vee$ (4)) hold:
  (1) $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{tpk}, \mathsf{RL}_{T^*}, M, \sigma) = 1$
  (2) $(T^*, M, \sigma) \notin \mathsf{SSet}$
  (3) $T \notin \mathsf{TSet} \wedge \mathsf{TSK}(\cdot)$ has not been called
  (4) $\mathsf{TSK}(\cdot)$ has been called with non-$\perp$ output
Otherwise return 0
```

- AddU*: The same as before.*
- Revoke*: The same as before. We note that T^* is the challenge time that $\mathcal{A}$ outputs (M, σ).*
- 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.*
- 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* TokenGen *oracle is called, the* SetToken *oracle is also called right after calling the* 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* AddU(ID)*. If t_T is not generated via the* TokenGen *oracle, then call the oracle* $\mathsf{TokenGen}(\mathsf{tsk}, T)$ *and the* 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 $\perp$. If* $\mathsf{ID} \notin \mathsf{GU}$*, then the oracle runs* AddU(ID)*. Return* $\mathsf{sigk}_{\mathsf{ID}}$*.*
- TSK*: The token key reveal oracle returns $\perp$ if the* USK *oracle was called before and at least one identity is not revoked.*[11] *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.

Definition 4.5 (Linking Soundness)**.** *For any PPT adversary $\mathcal{A}$ and security parameter $\lambda \in \mathbb{N}$, we define the experiment* $\mathrm{Exp}^{snd}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$ *as follows.*

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

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

- Revoke*:The same as before.*

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

4.2 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 (e.g., [6]) and Groth-Sahai proofs [35] which are typically used for constructing group signature schemes secure in the standard model, e.g., [34, 8, 45, 46, 47]. We employ the Fiat-Shamir transformation [29] which converts a three-move Σ protocol to non-interactive zero-knowledge (NIZK) proof as in group signature schemes secure in the random oracle model, e.g., [31, 17, 24, 15, 58, 25].

The Basic Idea: Our GS-TDL scheme is based on the Furukawa-Imai group signature scheme [31] 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 [28]. Next, for the linking property, we apply the Franklin-Zhang technique [30], where a group signature contains Belenkiy et al.'s verifiable random function (VRF) [12]. 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 [18, 48, 57] 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 $(\mathsf{ID}_{T,1}, x_{T,1}), \ldots, (\mathsf{ID}_{T,n_T}, x_{T,n_T})$ 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 [15].

We give the security proofs of the following theorems in Appendix.A.

Theorem 4.1. *The proposed GS-TDL scheme has anonymity in the random oracle model under the SDDHI assumption, where H is modeled as a random oracle.*

Theorem 4.2. *The proposed GS-TDL scheme has unforgeability in the random oracle model if the q-SDH assumption holds and* (Gen, Sign, Verify) *is EUF-CMA, where q is the number of signers and H is modeled as a random oracle.*

Theorem 4.3. *The proposed GS-TDL scheme has linking soundness.*

5 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.

5.1 Experimental Results of Our GS-TDL

Here, we explain experimental results of our GS-TDL scheme which shows the practicality of the scheme. Our implementation uses the RELIC library (ver.0.4.1) [7] for elliptic curve operations and the pairing operation, OpenSSL[12] (ver.1.0.2d) for standard signing and verifying, and GLib[13] (ver.2.47.1) for the hash table for (almost) constant-time searching. We note that we employ asymmetric pairing settings ((type III) Barreto-Naehrig (BN) curves [11]) 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, 10]. We use the parameter of "Nogami-Aranha" given in an IETF Internet-Draft [40]. Table 1 shows specifications of our machines we employed for experiments.

Table 1: Machine specification

Machine	x86_64 Server	Raspberry Pi (Model B)
CPU	Xeon E5-2660 v3 (2.60 GHz)	ARM1176JZF-S (700 MHz)
RAM	128GB	512 MB

Table 2 shows the running time of group operations. Here, Mul($\mathbb{G}_1$,Type), Mul($\mathbb{G}_2$,Type), and Exp($\mathbb{G}_T$,Type) are scalar multiplication on $\mathbb{G}_1$ and $\mathbb{G}_2$, and exponentiation on $\mathbb{G}_T$, respectively. If a base point is previously known, then Type is set as K, and U otherwise. We note that we always use Type U for operations on $\mathbb{G}_T$ since the RELIC library does not support Type K operation on $\mathbb{G}_T$.

Table 2: Group operations over 254-bit BN curves (μsec)

Operations		x86_64 Server	Raspberry Pi
Mul($\mathbb{G}_1$,U)		160	6,814
Mul($\mathbb{G}_1$,K)		51	2,081
Mul($\mathbb{G}_2$,U)		310	28,011
Mul($\mathbb{G}_2$,K)		67	5,388
Exp($\mathbb{G}_T$,U)		467	36,653
Pairing	Miller loop	466	35,477
	Final exp.	208	14,663

Table 3 shows the running time of singing and verification algorithms of OpenSSL 3072-bit RSA, 3072-bit DSA, and prime 256v1 ECDSA on our environment.

Optimization of GSign/GVerify: Originally, our GSign algorithm requires 2 Mul($\mathbb{G}_1$,K) + 4 Exp($\mathbb{G}_T$,U) + 2 Pairing (+Verify) and our GVerify algorithm requires 6 Exp($\mathbb{G}_T$,U) + 4 Pairing (+Verify) Here, we apply bilinearity of pairing in order to optimize the costs of algorithms by using Table 2 as follows. For GSign, we modify R_1 and R_2 as

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

[12]https://www.openssl.org

[13]https://wiki.gnome.org/Projects/GLib

Table 3: OpenSSL signing and verification costs (msec)

		x86_64 Server	Raspberry Pi
3072-bit RSA	Sign	2.844	236.736
	Verify	0.061	5.165
3072-bit DSA	Sign	0.942	75.864
	Verify	1.154	90.917
prime256v1 ECDSA	Sign	0.041	12.383
	Verify	0.104	14.716

Table 4: The number of operations for each algorithms

Algorithms	Operations
GSign (Original)	2 Mul($\mathbb{G}_1$,K) + 4 Exp($\mathbb{G}_T$) + 2 Pairing (+Verify)
GSign (Pairing-free)	2 Mul ($\mathbb{G}_1, K$) + 4 Exp($\mathbb{G}_T$) (+Verify)
GSign (Optimized)	4 Mul($\mathbb{G}_1$,K) + 2 Exp($\mathbb{G}_T$) + 1 Pairing (+Verify)
TokenGen	1 Mul ($\mathbb{G}_2$,K) + Sign
GVerify (Original)	6 Exp($\mathbb{G}_T$) + 4 Pairing (+ Verify)
GVerify (Optimized)	5 Mul($\mathbb{G}_2$,K) + 1 Exp($\mathbb{G}_T$) + 3 Miller loop + 2 Final exp. (+Verify)
Revoke	$\lvert\mathsf{RL}_T\rvert$ Mul($\mathbb{G}_1$,K)

and this modification allows us to run the GSign algorithm with 4 Mul($\mathbb{G}_1$,K) + 2 Exp($\mathbb{G}_T$,U) + 1 Pairing (+Verify) Moreover, for GVerify, we modify R_1' and R_2' as

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

and this modification allows us to run the GVerify algorithm with 5 Mul($\mathbb{G}_2$,K) + 1 Exp($\mathbb{G}_T$,U) + 3 Miller loop + 2 Final exp. (+Verify). As a remark, we use Mul($\mathbb{G}_2$,K) for computing W_T^c since W_T is fixed during the time period T. If T is frequently updated, Mul($\mathbb{G}_2$,U) should be used instead. We summarize the number of operations for each algorithms in Table 4. We can reduce the number of Exp($\mathbb{G}_T$,U) and pairings from the originals. In our environment, the running time of GSign (Optimized) is faster than that of GSign (Pairing-free). However, if the running time of 2 Exp($\mathbb{G}_T$,Type) is faster than that of 1 Pairing, then Pairing-free variant is more efficient.

Benchmarks of each algorithms: We give our experimental results of our GS-TDL scheme in Table 5. The total number of signers is 10,000,000, and the number of revoked signers is specified in parentheses () in the GVerify algorithm and the Revoke algorithm. We employ 3072-bit RSA for (TokenGen, Sign, Verify) which is used in our GS-TDL scheme since the verification cost (which is run by signers in the GSign algorithm) is faster than that of DSA and ECDSA in our environment.

We evaluate our results as follows.

- (Almost) Constant Verification Cost: First of all, we should highlight that cryptographic operations in the 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 searching key (i.e., the table takes ID as input, and outputs the corresponding x). In the Revoke algorithm, an array of (ID, x, τ) which corresponds to information of revoked users is constructed, and each τ contained in the array 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 of τ 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 key issuer,

Table 5: Benchmarks (msec)

Algorithms	x86_64 Server	Raspberry Pi
Join	0.146	-
TokenGen	2.92	238.553
GSign (Optimized)	2.078	142.472
GSign (Pairing-free)	2.212	161.616
GVerify(1,000)	2.889	210.883[†]
GVerify(10,000)	2.891	-
GVerify(100,000)	2.892	-
GVerify(1,000,000)	2.888	-
Revoke(1,000)	55.706	-
Revoke(10,000)	555.466	-
Revoke(100,000)	5,578.380	-
Revoke(1,000,000)	55,640.379	-

[†] The GVerify algorithm on Raspberry Pi does not contain the cost of serching on the revocation list.

Table 6: Simulation Environments

Software	CPU	RAM
PTV VISSIM 5.4	Core i7-4500U 1.8GHz	4GB
RapLab 8.0.4	Core i5-3470 3.2GHz	12GB
QualNet 7.4	Xeon E5-2643v3 3.40GHz $\times$ 2 (6 cores $\times$ 2)	256GB

who has plenty of computational resource.). In the GVerify algorithm, the verifier can easily check whether τ is contained in RL_T or not by using the hash table without any cryptographic operation.

- Practically Efficient Signing: In some situation, such as a road-to-vehicle communication system introduced in the next subsection, a verifier has a constrained computational power, and moreover the verifier needs to generate signatures frequently. In our experimental result, the GSign algorithm can be run at 142.472 msec even on Raspberry Pi, and it is comparable to RSA/DSA with similar security level, though our system additionally supports anonymity. If the verifier has a standard computational power (as in x86_64 server), then the GSign/GVerify algorithms can be run at 2.078 msec and 2.889 msec respectively. This result shows that our system is feasible enough in practice.

5.2 Network Simulation

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. Here, we give our simulation results for specifying the time T. We employ a traffic simulator PTV, a radio wave propagation analysis tool RapLab, and a network simulator QualNet. Simulation environments are shown in Table 6. The running time of PTV and RapLab is about 20 minutes in total, and that of QualNet is about 8 hours in total.

System Architecture: We define four entities as follows: a vehicle $\mathcal{V}$, the vehicle manufacturer $\mathcal{VM}$, the Road Side Unit $\mathcal{RSU}$, and the Token Generation Unit $\mathcal{TGU}$. 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 $\sigma \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{tpk}, t_T, \mathsf{sigk}_{\mathsf{ID}}, M)$.

Simulation Setup: We consider the following two cases:

- Method 1: A vehicle $\mathcal{V}$ generates σ every a few second, and broadcasts (M, σ, T). In our simulation, we set three seconds as its interval.
- Method 2: A vehicle $\mathcal{V}$ generates σ and sends (M, σ, T) to an $\mathcal{RSU}$ when $\mathcal{V}$ enters a certain range of the $\mathcal{RSU}$. In our simulation, the range is 239m.

We set T as 10 minutes, and prepare a "blank" time period where vehicles stop to generate signatures during the blank time[14]. We set 10 minutes as the blank. After the blank time period, the next T will start and vehicles generate signatures (according to the method). The communication protocol from a vehicle $\mathcal{V}$ to an $\mathcal{RSU}$ is IEEE 802.11p. The road network considered covers an area of 2.4 $\times$ 2.4 km^2. We selected two places for different visibility. One is Ginza area, which has low visibility due to tall buildings, and the other is Koganei area, which has high visibility. We set 2,000 vehicles and 1,000 vehicles in these areas, respectively. Since about 200,000 vehicles are discarded per a year in Japan[15], we set the number of revoked vehicles is 4,000,000 that covers the number of discarded vehicles during 20 years. The detailed simulation flow is as follows:

1. At the beginning of the time period T, $\mathcal{VM}$ generates the current revocation list, and each $\mathcal{RSU}$ receives it. In the simulation, each $\mathcal{RSU}$ waits a few minutes which are determined by the size of the revocation list and Table 5. We set it 278.2 seconds, and assume that there is a channel to send the revocation list to each $\mathcal{RSU}$ and ignore the sending time.
2. Right after T starts, each vehicle generates a group signature and send it to $\mathcal{RSU}$s (according to the method). We assume that each vehicle can obtain a time token immediately, and ignore this time. In the simulation, each vehicle waits 142.472 msec (see Table 5) for generating a group signature. If $\mathcal{RSU}$s wait to receive the revocation list, then $\mathcal{RSU}$s preserve the group signature, and verify it after the waiting time. In the simulation, each $\mathcal{RSU}$ waits 2.90 msec (see Table 5) for a group signature verification.
3. During the blank time, each vehicle does nothing. If $\mathcal{RSU}$s have group signatures that have not been verified, then $\mathcal{RSU}$s verify these signatures.

We display our simulation results over maps in Fig 2, 3, 4, and 5. We use OpenStreetMap.[16] Each dot indicates a group signature generated by a vehicle. In our simulation (60 minutes), the vehicle generates signatures during the first 10 minutes which are colored by red. After the blank time (10 minutes), these signatures are colored by black during 10 minutes. Again, after the blank time (10 minutes), these signatures are colored by brown during 10 minutes. Recall that, as the colors in the figures show, signatures generated in the same time period are linkable, and ones generated in different time periods are unlinkable. For the sake of clarity, we draw blue line as the route that the vehicle went through during the simulation.

Our simulation results indicate that Method 2 is more suitabe to correct linkable information since we can easily follow the track of the route that a vehicle passes during 10 minutes. Since we set $\mathcal{RSU}$s to cover the map, e.g., on lights of crossroad, a middle of a long road, and so on, we need to consider more adequate installation location of $\mathcal{RSU}$s in practice. We leave it as a future work of this paper.

[14]This blank time is necessary to cut linkage of location information which would be contained in messages. Suppose that a vehicle broadcasts a message and a corresponding signature periodically, e.g. in a second, and there is no such blank time. Then, the last message generated in the time period T and the first message generated by the next time period are linkable even though corresponding signatures are unlinkable, since locations indicated by these messages are too close to guess these are generated by the same vehicle.

[15]Japan Automobile Dealers Association (Japanese): `http://www.jada.or.jp/`

[16]OpenStreetMap: `https://www.openstreetmap.org/`

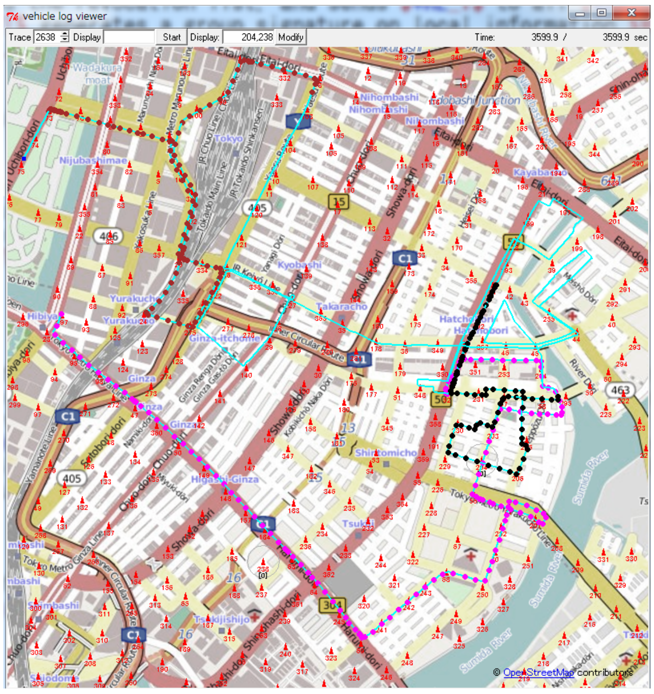
Figure 2: Method 1: Ginza, 2,000 vehicles

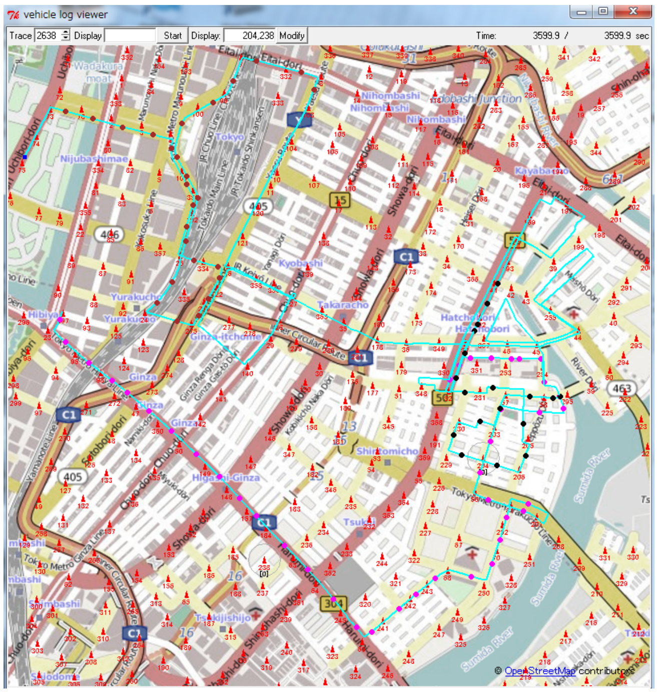
Figure 3: Method 2: Ginza, 2,000 vehicles

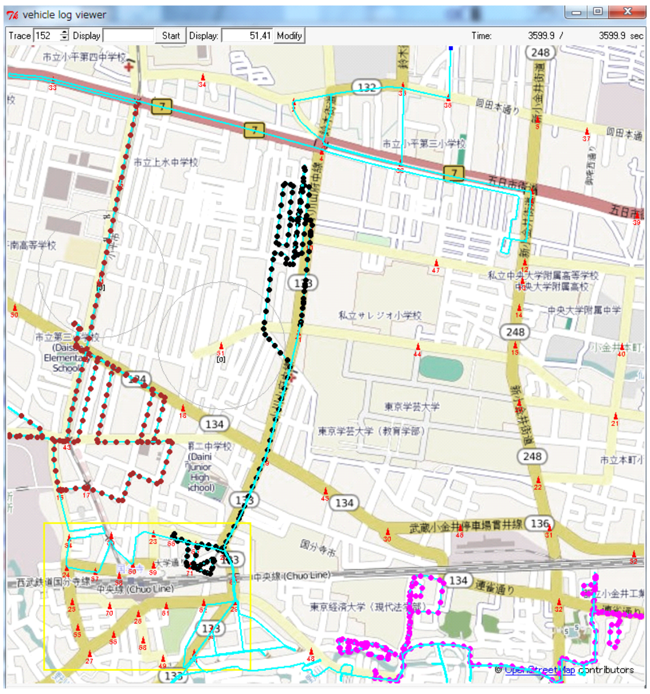
Figure 4: Method 1: Koganei, 1,000 vehicles

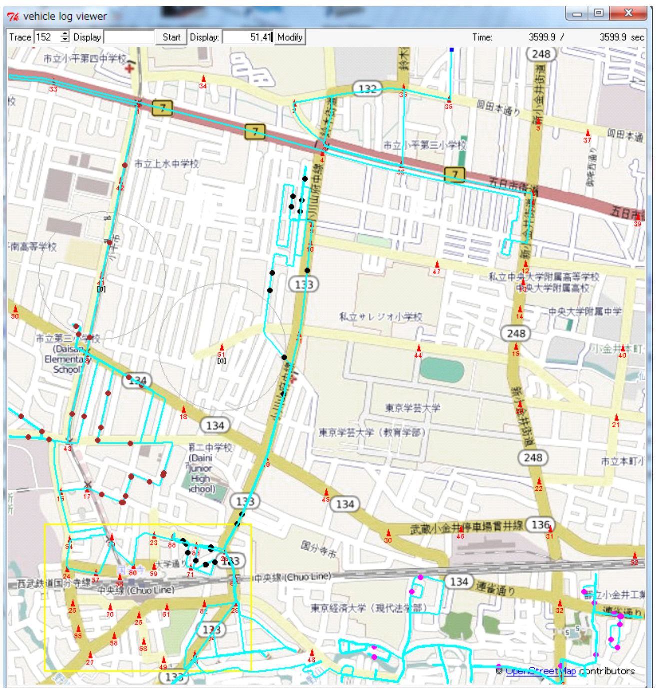
Figure 5: Method 2: Koganei, 1,000 vehicles

The number of stacked signatures on RSUs: Since each vehicle sends group signatures during $\mathcal{RSU}$s wait to receive the revocation list, these signatures are stacked in $\mathcal{RSU}$ to wait verification. Fig 6 shows the number of stacked signature in an $\mathcal{RSU}$. As the figure shows, the number of stacked signature is immediately reduced after the $\mathcal{RSU}$s start to verify the signatures. This result shows that our scheme has enough performance for road-to-vehicle communication systems even in a practical situation.

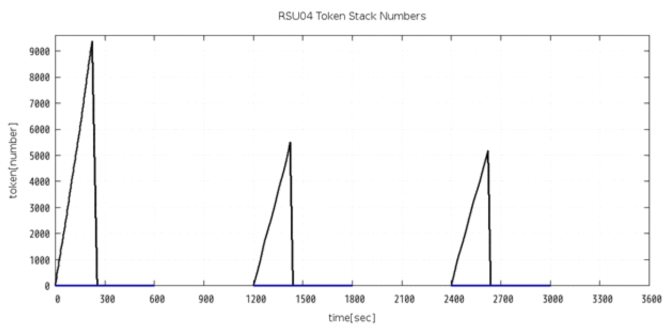

Figure 6: RSUstack

6 Conclusion

In this paper, we propose a road-to-vehicle communication system via group signatures with time-token dependent linking (GS-TDL), and show our network simulation whose results indicate our system has enough performance for road-to-vehicle communication systems even in a practical situation. Since our system has not only a good performance but also supports the revocation property which is indispensable in real systems. It is particularly worth noting that no vehicle is involved in the revocation procedure and the signing/verification costs do not depend on the number of revoked vehicles.

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

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

[5] 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.

[6] 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.

[7] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient LIbrary for Cryptography. `https://github.com/relic-toolkit/relic`.

[8] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai. Revocable group signature with constant-size revocation list. *Comput. J.*, 58(10):2698–2715, 2015.

[9] 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.

[10] 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.

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

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

[13] 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.

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

[15] 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.

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

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

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

[19] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. Foundations of fully dynamic group signatures. Cryptology ePrint Archive, Report 2016/368, 2016. `http://eprint.iacr.org/`.

[20] 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.

[21] 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.

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

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

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

[25] D. Derler and D. Slamanig. Fully-anonymous short dynamic group signatures without encryption. Cryptology ePrint Archive, Report 2016/154, 2016. `http://eprint.iacr.org/`.

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

[27] K. Emura and T. Hayashi. A light-weight group signature scheme with time-token dependent linking. In *LightSec*, pages 37–57, 2015.

[28] K. Emura, A. Kanaoka, S. Ohta, and T. Takahashi. Secure and anonymous communication technique: Formal model and its prototype implementation. *IEEE Transactions on Emerging Topics in Computing*, pages 1, PrePrints, doi:10.1109/TETC.2015.2400131, 2015.

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

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

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

[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, L. Chen, H. S. Cho, and D. Nyang. Short dynamic group signature scheme supporting controllable linkability. *IEEE Transactions on Information Forensics and Security*, 10(6):1109–1124, 2015.

[38] 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.

[39] 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.

[40] K. Kasamatsu, S. Kanno, T. Kobayashi, and Y. Kawahara. Barreto-naehrig curves. Internet-Draft draft-kasamatsu-bncurves-01, IETF Secretariat, July 2015. `http://www.ietf.org/internet-drafts/draft-kasamatsu-bncurves-01.txt`.

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

[42] V. Kumar, H. Li, J. J. Park, K. Bian, and Y. Yang. Group signatures with probabilistic revocation: A computationally-scalable approach for providing privacy-preserving authentication. In *ACM Conference on Computer and Communications Security*, pages 1334–1345, 2015.

[43] 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.

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

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

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

[47] B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *CRYPTO*, pages 296–316, 2015.

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

[49] J. K. Liu, C. Chu, S. S. M. Chow, X. Huang, M. H. Au, and J. Zhou. Time-bound anonymous authentication for roaming networks. *IEEE Transactions on Information Forensics and Security*, 10(1):178–189, 2015.

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

[51] L. Malina, J. Hajny, and V. Zeman. Light-weight group signatures with time-bound membership. *Security and Communication Networks*, 9(7):599–612, 2016.

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

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

[54] 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.

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

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

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

[58] D. Pointcheval and O. Sanders. Short randomizable signatures. In *CT-RSA*, pages 111–126, 2016.

[59] 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.

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

[61] 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.

[62] 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.

[63] 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.

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

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

[66] T. M. Truta, A. Campan, and P. Meyer. Generating microdata with p-sensitive k-anonymity property. In *Secure Data Management*, pages 124–141, 2007.

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

[68] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for V2V communications. In *IEEE Vehicular Networking Conference*, pages 1–8, 2013.

[69] 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.

[70] 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.

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

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

A. Security Analysis of Our GS-TDL

In this appendix, we give security proofs of our GS-TDL scheme.

Proof of Theorem 4.1

Proof. We define the following two games: Game 0 is the same as $\mathrm{Exp}^{\mathit{anon\text{-}tg\text{-}b}}_{\mathsf{GS\text{-}TDL},\mathcal{A}}(\lambda)$. 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 $\mathsf{gpk} = (\mathit{params}, h, W, e(g_1, g_2), e(h, W), e(h, g_2), H)$, 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
$$R_{1,1}^* = \frac{e(h, g_2)^{s_{\delta,1}^*} e(h, W)^{s_{\beta,1}^*}}{e(C_1^*, g_2)^{s_{x,1}^*}} \Big(\frac{e(C_1^*, W)}{e(g_1, g_2)}\Big)^{-c_1^*}$$
and
$$R_{2,1}^* = e(\tau_1^*, g_2)^{s_{x,1}^*} \Big(\frac{e(g_1, g_2)}{e(\tau_1^*, W_{T_1})}\Big)^{-c_1^*},$$
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. □

Proof of Theorem 4.2

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.

□

Proof of Theorem 4.3

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. $\square$