A Group Signature with relaxed-privacy and revocability for VANET

Mohammad Saiful Islam Mamun and Atsuko Miyaji

Japan Advanced Institute of Science and Technology (JAIST)
Ishikawa, Japan.
{mamun, miyaji}@jaist.ac.jp

Abstract. This paper adapts a new group signature (GS) scheme to the specific needs of certain application e.g., a vehicular ad hoc network (VANET). Groth GS is the first efficient GS scheme in the BSZ-model with security proofs in the standard model. We modify the Groth GS in order to meet a restricted, but arguably sufficient set of privacy properties. Although there are some authentication schemes using GS none of them satisfy all the desirable security and privacy properties. Either they follow GSs that rely on Random Oracle Model, or unable to satisfy potential application requirements. In particular, *link management* which allows any designated entities to link messages, whether they are coming from the same member or a certain group of members without revealing their identities; *opening soundness* that prevents malicious accusations by the opener against some honest member of the group; *revocation system* that privileges from fraudulent member like the traditional Public Key infrastructure (PKI). In order to achieve the aforementioned security properties together, we propose a new GS model where linkability, sound opening and revocability properties are assembled in a single scheme. The novelty of our proposal stems from extending the Groth GS by relaxing strong privacy properties to a scheme with a lightly lesser privacy in order to fit an existing VANET application requirements. In addition, we partially minimize the Groth GS scheme to expedite efficiency.

keywords: Group signature, Linking, Revocability, Opening soundness

1 Introduction

Although Groth GS scheme exhibits strong privacy properties, but sometimes stringent security and privacy policy prevents some reasonable case of application and bring performance overhead. For example, to guarantee privacy, VANET use pseudonym mechanisms [3] or GS scheme [7] (without additionally generating a pseudonym). Although complete untraceability (or, strong privacy) among the members is an important properties, many applications (e.g., VANET) demand diverse privacy requirement. Unfortunately standard GSs like Groth's, is unsuitable for such kind of application. Members might benefit from established trust relations among them in order to communicate private data in an unobservable manner [4,5]. Therefore, we refer to to introduce *Link Manager* (LM) where a designated members (e.g., Road Side Unit (RSU) in VANET) would operate as a semi-trusted entity. For instance, let an RSU intend to keep the record of the average number of emergency vehicles per hour passing through a certain junction without revealing the identity of the corresponding vehicles. Let a Group Manager (GM) decide to tolerate *certain number* of consecutive malicious signature from a certain member before it attempts to declare the member as *illegal*. Note that in a GS scheme, illegal signature is sent to *opener* to disclose its *identity*. Hence, we propose LM that can link a signature with any of the previously received signatures from the same member. This feature significantly adds a privacy hierarchy in any application. For instance, in VANET, vehicles (group members) are fully anonymous in the network, RSUs (LM) can link between signatures from vehicles but cannot circumvent anonymity, and Traffic Security Division (Opener) can crack full anonymity.

We address a security threat to the reliability of ownership of a signature on a certain message and provide a new application framework (e.g, Value Added Service Providers (VSPs) in VANET). Let a vehicle be registered to several VSPs for certain services. It is mandatory to ascertain that a VSP provides service to the right client to which it has agreement to. But lack of *opening soundness* may allow a malicious member to claim for illicit service (and to provide a *proof* thereof) as if it is an honest member [9]. Therefore, we suggest the *Opener* to issue *a ticket* (e.g., signature on a service name) and a resp. *token* to that ticket (a proof of ownership of the signature) to the member. In order to obtain services, a member must submit the *ticket* together with corresponding *token* to the VSP for justification.

However, a GS should have a mean to revoke members from the system, e.g., if they are declared by the *opener* as *illegal*, or if their private keys get compromised over time. Therefore, a revocation system is required to GS that improves key-update efficiency on the Key Issuer side (s.t., constant computation) while restraining efficiency for the individual member (s.t., constant signature size, no secure channel needed to update keys). In addition, it requires to satisfy *backward unlinkability*, that ensures signatures produced by a revoked vehicle cannot be linked to the post-revocation signatures prior to the recent revocation.

Related Work: Unlike traditional digital signature schemes, GS allows a member to create an *anonymous* (and *unlinkable*) signature that conceals the identity of the vehicle and hence preserves privacy [1,7]. Following the foundation of GS [2], a number of different security requirements have been proposed as primitives. Consequently, BSZ-model in [6], proposes the dynamic GS scheme with three security notions anonymity, traceability and non-frameability that implies all the previously proposed notions of security. They also separate the role of Group Manager (GM) into: *issuer* and *opener*. That is why, we exploit the GS proposed by Groth [10] which is secure in BSZ-model and yields the best fit to any application model (e.g., VANET). In addition, it allows a reasonable constant number of group elements for all parts of the group signature scheme including a *group public key* and a *group signature*. This property is a prerequisite to support scalability, where thousands or perhaps even millions of messages may be transferred at any time instance. Furthermore, Groth GS satisfies strong security requirements, in particular, not relying on weak random oracle model.

Note that all the aforementioned GS properties are not completely novel. Firstly, *linkability* feature is discussed in several traceable GS schemes such as [19,20,24] and very recently [21]. But all of them either do not support opening algorithm and hence do not allow anonymity revocation, or the security proof belongs to Random Oracle Model (ROM). Secondly, *revocability* properties for a GS was first explored in [13] and later followed by [12,26,27]. All the revocable GS schemes that have been proposed so far were either reluctant to backward unlinkability, constant signature size/ verification cost/ public key size, or rely on ROM. Recently, two scalable revocation approaches have been proposed from standard security model [17,18]. Since the revocation techniques are inspired by broadcast encryption tree, the cardinality of the group becomes fixed and more harshly their signature size is 6 times larger than that of our scheme which could cause performance bottleneck in a large scale application. Thirdly, we followed the *opening soundness* property described in [9] which protect the signature from getting hijacked by other member vehicles. We actually utilize the property to present a real life application, such as secure transaction between the VSPs and respected group members.

Main contributions: We introduce a GS scheme, based on pairing-based construction of Groth with additional properties: (1) linkability by a special party, (2) Opening soundness to introduce an application framework, (3) Revocability with constant computation.

To the best of our knowledge, there is no GS scheme proposed in the literature that satisfy all the aforementioned properties together. We accumulate the cited properties in a single scheme and this challenging effort helps to induce relaxation from a strong privacy to a scheme with a lesser but adaptive privacy hierarchy, and hence make the GS scheme applicable to certain application environment by being simplistic, yet efficient way. Moreover, for accelerating efficiency we use a simplified version of Groth GS that is CPA-secure, and later suggest applying certain batch verification technique described in [16] for signature verification.

2 Preliminary

2.1 Network model

We refer to a symbolic network model for VANET described in [1]. In this hierarchical model, vehicles are remained at the bottom level of the hierarchy (see Fig. 1) where they might be a member of several RSUs, On-demand service stations (VSPs), Electronic Payment point (ex. toll service). Vehicular groups could be formed: by region (ex. east region), social spots/services (ex. shopping mall, hospital area), category (ex. public service, emergency, personal vehicles) etc. Two types of communication may exist in the network: vehicle to vehicle (V2V) and vehicle to infrastructures s.t., RSUs, VSPs etc. (V2I). Each vehicle in the network must be equipped with an On Board Unit (OBU) consisting of Event Data Recorder (EDR) that records all the received messages, Tamper Proof Device (TPD) that implements cryptographic tools and ensures authenticated access control. Each GM consists of: an *issuer* for the purpose of registration and an *opener* to explore the identification of the members. Subsequently, all the RSUs would act as LMs In order to obtain services from VSPs, vehicles would require to submit a *token* issued by the *opener* to prove its identity with the credibility of the requested service and later VSP would justify by the *Judge* algorithm.

2.2 Security model with extended GS Properties

Link Manager. Unlike conventional digital signatures, GS scheme permits its members to anonymously sign a message without revealing its identity except in some inevitable events when the *opener* discloses the identity of the signer. Unfortunately, this is unsuitable for multifarious privacy settings like VANET. Consider a real life application in VANET, where each time hundreds or even thousands of messages be transferred between vehicles and RSUs. If the *opener* is called for every single suspicious message, this will convey a severe burden to the *opener*, while causing the revocation system to deteriorate over time.

We render a relaxed privacy with on-demand *limited traceability* in two steps: First, the case in which any suspicious member discovers a *doubtful* message arriving from another member of the same group. Here the message with corresponding signature would be forwarded to the LM (RSUs in VANET) instead of *opener*. By using the linking key, LM can check if two signatures are from the same individual member while preserving anonymity of the member (without revealing *identity*). Clearly, LM is delegated the link capability by the *opener* that introduces a fine-grained control on the anonymity of the members. Second, the case in when LM determines a specific member as *malicious*, the message together with the signature would again be forwarded to the *opener* to reveal the identity of the respected member. Note that an *opener* responds only to the privileged member such as RSUs in VANET. LM can be added voluntarily to decide whether the signature to be linked or not [22] [24]. For instance, an RSU collects statistics for future traffic development, e.g., pattern or frequency of certain types of vehicles that follow specific road, without revealing identities of the vehicles.

It is worth pointing out that *full anonymity* can not be achieved here since LM can trace/link certain member or group of members, and hence, *absolute privacy* is not guaranteed. We termed it as *relaxed privacy* where members could only ask for checking linkability of any suspicious message to the LM, but should not get any feedback results.

However, providing linking capability to a group signature is not novel. For example, direct anonymous attestation scheme in [19], Ring signature scheme in

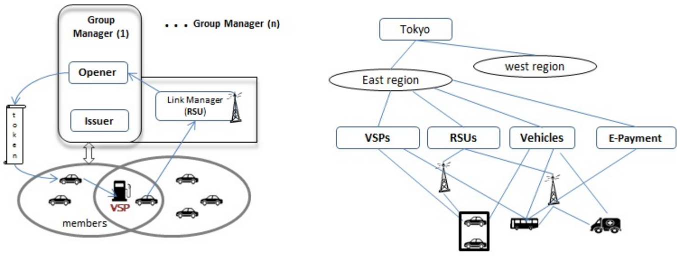

Fig. 1. An example demonstrating: vehicles' group formation (right) and communication among the GS members (left) in a traditional VANET Network.

[20] are special types of group signatures with linking capability. But these signature schemes do not support an opening algorithm, that means, the anonymity is not revocable. A recently proposed GS scheme in [21] is an example with both linking and opening capabilities, but the security proof is under ROM and it cannot be guaranteed whether the scheme supports *opening soundness* or not.

Opening soundness. Security threats should be considered as the threats to the system *reliability*. But in Groth's GS, signatures generated by a member are susceptible to be hijacked by a malicious member by forging the *proof of ownership* [9]. We present an application framework by utilizing this property. For instance, let a vehicle have an agreement with a third party VSP for a certain service. It would generate a signature citing the VSP's name and requested service and submit the message with the signature, which we termed as a *ticket*, to the Traffic Security Division (or, Opener) for attestation. After that, Opener will issue a *token* in order to bind a credential to its legitimate owner (see Fig. 1). Later when the vehicle requests for a service to the VSP, it would attach a *ticket* and its corresponding *token* issued by the Opener for justification. VSP would justify the credential of the vehicle, to ascertain that, the claiming vehicle is really worth of having the service.

Revocation. Current state-of-the-art GS allows members join the group dynamically, i.e., that supports growing the size of a group without updating group public keys or issuing certificates for the rest of the group members. Unfortunately, the revocation, where the size of the group shrinks, still remains a non-trivial problem. In traditional signature schemes in PKI, a Certificate Revocation List (CRL) is issued as a solution to the revocation. This is a very attractive from the signer's part since the signer need not be aware of any CRL changes, and on the opposite side, the verifier conveys the burden of checking or updating the revocation list. Nonetheless, this is not practical in the GS scheme, because if a verifier can link a single signature to the CRL entry, it can do the same to multiple signatures as well. Moreover, if a revoked member's signature is considered before revocation, *backward unlinkability* is not preserved [12].

Like standard PKIs, GS does not have any efficient revocation system in practice. Many existing solutions do not scale well due to either high overhead or tight operational requirements, such that, computational complexity belongs to $O(n)$ or $O(r)$, where n and r are group size and number of revoked members respectively. Revocation solution was first introduced in [13], where the signature size was linear to the number of revoked members. Authors in [14] proposed a forward secure revocation system with constant signature size. But, one of the features of this scheme was to use fixed time periods to revoke a member, which is in fact, impossible to implement in many application e.g., VANET. Schemes in [26] [27] have O(1)- cost for signing and verification time but O(N)-size (linear) group public keys.

Recently, two revocations approaches have been proposed, mainly based on the Naor-Naor-Lotspiech (NNL) Broadcast Encryption framework that yields a scalable revocable group signatures to obtain private keys of constant size in the standard model [17] [18]. Unfortunately, signature size of both the schemes are too large for practical deployment. They are approximately 3 and 6 times larger, respectively, than that of our scheme[1]. Moreover, since NNL is a tree-based technique, unlike ordinary dynamic GS schemes the maximal cardinality of the group would be fixed. Therefore, even though the revocation schemes are truly scalable, they cannot be used for application where larger signature size causes increased communication overhead and hence degrades overall performance.

We exploit the idea of [12] in our GS, where they offer a CRL-like revocation with constant length signature as well as constant computation for revocation, that means, the complexity is O(1) with respect to n and r. If a group member vehicle leaves the group or is judged as an *illegal* member, GM updates the *RList* accordingly. We propose not to update group public key (*gpk*) in every case when a new member leaves or is forcibly revoked from the group for the sake of efficiency. Instead, information regarding new/revoked group members can be accumulated between two successive revocation events.

3 The Proposal

Groth GS applies *certified signature* method based on the **DLIN** and the $q-\mathbf{U}$ assumption (see [10] for details) using Non-interactive Witness-indistinguishable (NIWI) proofs[8]. Note that we present a relaxed (CPA-secure) notion of Groth GS, e.g., allow no adversarial access to the *open* algorithm and add/modify some generic algorithm e.g., adding: SignLink(), Revoke() modifying: Keygen(), Registration(), Open() algorithms.

System Set-up: Consider a probabilistic polynomial time algorithm $\mathcal{G}$ that generates $gk := (p, \mathbb{G}, \mathbb{G}_{\mathbb{T}}, e, g) \leftarrow \mathcal{G}(1^k)$ such that: p is a k-bit prime, $(\mathbb{G}, \mathbb{G}_{\mathbb{T}})$

[1] Group signature size of [17] and [18] are comprised of 144 and 92 group elements respectively while our signature size consists of 28 group elements.

are cyclic group of order p. Let g generate $\mathbb{G}$ and e be a non-degenerate and efficiently computable bilinear map s.t., $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ so that $e(g, g)$ generates $\mathbb{G}_{\mathbb{T}}$, and $e(g^a, g^b) = e(g, g)^{ab}$ for any $a, b \leftarrow \mathbb{Z}_p$.

Key Generation GKg(1^k): Group Manager generates secret keys: ik for Issuer (Registration managers), ok for Opener (Traffic Security Division), lk for Link Manager (Designated RSUs) and public system parameters gpk. Let $(u, h, z, K, L) \leftarrow \mathbb{G}$, $(l, r, s) \leftarrow \mathbb{Z}_p$, $f = u^l$, $T := e(f, z)$, $xk := (\phi, \eta) \leftarrow gk, F := g^{\phi}, H := g^{\eta}$, $pk := (F, H, K, L)$, $R := g^r, S = g^s$, Hash $\leftarrow \mathcal{H}(1^k)$, $U := F^R$, $V := H^S$, $W = g^{R+S}; crs := (F, H, U, V, W)$. $lk := l$, $ik := z$, $ok := xk$, and $gpk := (gk, \text{Hash}, u, f, h, T, crs)$

Registration (User i : gpk, Issuer: gpk, ik): Group members with their identity i (e.g., vehicles, RSUs) need to complete registration with Issuer. Let total number of non-revoked vehicles be n in an instance. A member i and Issuer run a 5-move key generation protocol (described in [11]) in order to generate a key pair $\{(v_i, x_i),\ v_i\ \}$, where $v_i \leftarrow g^{x_i}$ Issuer then signs v_i to produce certificate certSign$_i$ $:= (a_i, b_i) \leftarrow (f^{-r_i},\ (v_i h)^{r_i} z)$, where $r_i \leftarrow \mathbb{Z}_p$.
Member i accepts the certificate certSign$_i$ if $e(a_i, hv_i)\ e(f, b_i) = T$. Finally, the Issuer maintains a database to store $reg[i] \leftarrow v_i$ for the *open*() and the *judge*() algorithm, and $rev[i] \leftarrow r_i$ for the *revocation*() algorithm and the member i stores group signing key $gsk[i] \leftarrow (x_i,$certSign$_i)$

Generating signature for authentication GSign($gpk, gsk[i], m$): In order to sign a message m a registered member i first generates a certified signature σ using her private key x_i. Then it produces a NIWI proof[2] π that consist of a commitment to σ. The detailed instantiation is as follows. Let a member i select $\rho \leftarrow \mathbb{Z}_n$ and compute $a := a_i f^{-\rho}$, $b := b_i(hv_i)^{\rho}$, $\varkappa := u^{-\rho}$ and $\sigma := g^{1/x_i+\mathcal{H}(m)}$. $\pi \leftarrow \mathrm{P_{NIWI}}\ (crs, (gpk, a, \mathcal{H}(m)), (b, v_i, \sigma))$ The resulting signature on a message m is: $\Sigma := (a, \varkappa, \pi, \sigma)$.

Message verification GVerify(gpk, m, Σ): To verify a signature Σ on message m, receiving member checks NIWI proof π:

```
IF $\mathrm{V_{NIWI}} \leftarrow (crs, (gpk, a, \mathcal{H}(m)), \pi) = true$;
    return 1
ELSE return 0
```

Identity Opening authority Open(gpk, ok, m, Σ): By accessing the registration table *reg*[3] generated by the Issuer, by using opening key ok it can revoke the signer's identity i of a valid signature Σ on message m. This algorithm can be used for two purposes: Firstly, it helps to exhibit the signer of a doubtable message/signature sender and later revoke the member from the group. Secondly, it promotes accountability of certain applications by providing proof of ownership

[2] To demonstrate that ciphertext contains a valid certified signature

[3] The opener has read access to the registration table *reg*

of a certain signature. Consider a member i that requires a credential regarding a service which is mentioned in the message m. It first generate a signature Σ on m and then request the Open() to provide a proof of ownership or token on m. Later a receiver (e.g., VSP) verifies the signature by using **GVerify** (gpk, m, Σ). If successful, then it extracts v of the corresponding member i and searches the registration table to find $v \overset{?}{=} v[i] \leftarrow reg[i]$.

$(b, v, \sigma) \leftarrow \text{Extract}_{ok}(crs, (gpk, a, \mathcal{H}(m)), \pi)$.

In order to generate proof of ownership, it randomly selects $(c, d) \leftarrow \mathbb{Z}_p$ and computes: $(y_1, y_2, y_3) := (F^c, H^d, v_i g^{c+d})$ and a Non Interactive Zero Knowledge (NIZK) proof $\theta \leftarrow (\theta_1, \theta_2)$ of corresponding member i where $\theta_1 := y_1^{1/\phi},\ \theta_2 := y_2^{1/\eta}$ and $(\phi, \eta) \leftarrow ok$. Finally, it issues a *proof of ownership* $(i, (\sigma, \theta))$ of a signer i on a certain message m.

Validating Ownership Judge $(gpk, i, v_i, m, \Sigma, (\sigma, \theta))$: This algorithm verifies whether the opening is correct or not. It returns 1 if the opening is correct. Say VSPs in VANET could use this algorithm to verify the beneficiary of a certain service.

```
IF $\Big(\textbf{GVerify}(gpk, m, \Sigma) = 1 \bigwedge (i \neq 0) \bigwedge e(\sigma, v_i g^{\mathcal{H}(m)}) = e(g, g) \bigwedge e(F, \theta_1) =$
  $e(y_1, g) \bigwedge e(H, \theta_2) = e(y_2, g) \bigwedge \sigma\theta_1\theta_2 = y_3\Big)$ return 1
ELSE return 0
```

Managing Linkability SignLink$((\Sigma_1, m_1), (\Sigma_2, m_2), lk)$: By using lk, the LM (designated member) tries to find a link among existing list of signatures with a new signature, or between two signatures whether they are generated from the same signer i. It returns 1 if successful. Let $a_1, \varkappa_1 \leftarrow \Sigma_1$ and $a_2, \varkappa_2 \leftarrow \Sigma_2$.

```
IF GVerify $(gpk, m_1, \Sigma_1) \bigwedge$ GVerify $(gpk, m_1, \Sigma_1)$
    IF $e(a_1, h)\ e(\varkappa_1, h^{lk})^{-1} = e(a_2, h)\ e(\varkappa_2, h^{lk})^{-1}$ Or,
        $e(a_1/a_2, h) = e(\varkappa_1/\varkappa_2, h^{lk})$
        return 1
ELSE return 0
```

Revocation Revoke$(gpk, RList)$: Revocation would be accomplished in two steps: Firstly, GM issues a new group public key gpk including all new parameters, termed as $\mathcal{R}$, and publish it for all the non-revoked members. Usually, the Issuer publishes a signed and time-stamped $\mathcal{R}$ in a publicly accessible bulletin board or server. Unlike ordinary GS schemes, in our scheme members do not need to contact the *issuer* privately (following interactive *join/issue* protocol) to update their certificates. Secondly, after getting the public parameters $\mathcal{R}$ for revocation, all the non-revoked member can update their certificates (a_i, b_i) with the newer one consequently. However, it is quite likely that no revoked members can update their certificates from the revocation information available in public. Moreover, all other non-revoked member need $O(1)$ operation to update, irrespective of the size of the revocation list or the group members.

This algorithm allows Issuer and all non-revoked member to update their keys according to the revoked users list *RList* provided by the GM. Let $t := \{\prod_{i=1}^{n} r_i, s.t.\ r_i \leftarrow rev[i]\}$ be known to all the last known non-revoked n group member. Note that, t considers of all the current non-revoked members including the *new* member that join between two consecutive revocation events.

Let m member be adjudged as *illegal* member between two successive revocation events, and $r_{ki} \leftarrow rev[i]$ be selected for the revoked members (m). Then, $RList := k_1, k_2 \cdots k_m$ where $m < n$; and $r_k = \prod_{i=1}^{m} r_{ki}$.

Issuer: update $rev[i]$ according to the new list of non-revoked member (n)
  $\tau \leftarrow \mathbb{Z}_n; \delta := \tau^l;\ u' := u \cdot \tau;\quad f' := f \cdot \delta;\quad h' = h \cdot \delta$
  $T' := e(f', z)$; and $\gamma := \delta^{\frac{t}{r_k}} \bmod n$
  new $gpk := (gk, \mathrm{Hash}, u', f', h', T', crs)$
  publish $\mathcal{R} \leftarrow (t, gpk, \gamma, r_k)$ for the non-revoked members.
Each group member ($i \neq k_i$) updates certificate $certSign_i(a_i, b_i)$ as follows:
  $gsk[i] := (x_i, {a_i}', {b_i}') \leftarrow (x_i, a_i, b_i)$
  set $s_i = \frac{r_i \cdot r_k}{t}$
  set $a'_i = a_i \cdot \gamma^{-s_i}$ and $b'_i = b_i \cdot \gamma^{s_i}$

4 Security Analysis

Some notations, definitions, security proof we use from [9,10] and hence omit due to space constraint. Interested readers are referred to [9,10] for further discussion. We define several oracles necessary for security notions:

AU(i): Add User oracle adds an honest user i to the set **HU** by using Registration protocol.
RU(i): Revoked User oracle adds a revoked user i to the set **RList**.
CU(i, I): Corrupt User oracle sets I as the public key of corrupted user i.
RR(i): Read Registration oracle retrieves the corresponding registration table entry $reg[i]$ in input i.
StoU(i, I): Send-to-User oracle sets public/secret key pair to a user i and add i to **HU** set. It allows the adversary to engage in **Join/Issue** protocol with information I. The response of the protocol is returned to adversary.
WR(i, I): Write Registration oracle modifies $reg[i]$ to I
StoI(i, I): Send-to-Issuer allows the adversary to engage in **Join/Issue** protocol on behalf of the corrupted user i. The response to the issuer is sent back to the adversary.
RS(i):Reveal-secret oracle discloses the secret keys (group and user) to the adversary.
Op(m, Σ): The Open oracle returns the opening result by *group* **Open** () algorithm of the signature Σ for the message m.
Ch$_b$(m, i_0, i_1): Challenge oracle returns a challenge Σ^* from **GSign:** () where (i_0, i_1) belongs to the set **HU**.

Sig(i, m): Signing oracle returns a signature Σ on the message m where i is under the set **HU**.
LU(i): Linked User oracle retrieves (m_i, Σ_i) from the set **LSet**.
Link$(m_i, \Sigma_i, m_j, \Sigma_j)$: It queries **SignLink**() to check whether two signatures are generated from the same user.

Definition 1. A GS is said to have ***CPA****-anonymity* if

$$\Pr[b \leftarrow \{0,1\}; b' \leftarrow \mathcal{A}^{StoU,WR,RS,CU,Ch_b}(gpk, ik)\text{: } b = b']\text{-}\frac{1}{2}$$

is negligible for any PPT adversary $\mathcal{A}$.

Definition 2. A GS is said to have *correctness* if

$$\Pr[(i,m) \leftarrow \mathcal{A}^{AU,RR}(gpk);\ \Sigma \leftarrow \mathbf{GSign}(gpk, gsk[i], m);\\ (j,\sigma,\theta) \leftarrow \mathbf{Open}(gpk, ok, m, \Sigma)\text{: } \mathbf{GVerify}(gpk, m, \Sigma) = 0 \bigvee i \neq j \bigvee\\ \mathbf{Judge}(gpk, i, v_i, m, \Sigma, \theta) = 1 \bigvee \mathbf{Judge}(gpk, j, v_j, m, \Sigma, \theta) = 1]$$

is negligible for any PPT adversary $\mathcal{A}$.

Definition 3. A GS is said to have *non-frameability* if

$$\Pr[(i, m, \Sigma, \theta) \leftarrow \mathcal{A}^{CU,RS,StoU,WR,RK,Sig}(gpk, ok, ik)\text{:}\\ \mathbf{GVerify}(gpk, m, \Sigma) = 1 \bigwedge \mathbf{Judge}(gpk, i, v_i, m, \Sigma, \theta) = 1 \bigwedge \mathcal{A} \text{ has no access to}\\ \mathbf{Sig}(i,m),\mathbf{RS}(i)]$$

is negligible for any PPT adversary $\mathcal{A}$.

Definition 4. A GS is said to have *traceability* if

$$\Pr[(m, \Sigma) \leftarrow \mathcal{A}^{\mathrm{CU,AU,StoU,RS,RR}}(gpk, ok);\ (i, \theta) \leftarrow \boldsymbol{Open}(gpk, ok, m, \Sigma)\text{:}\\ \boldsymbol{GVerify}(gpk, m, \Sigma) = 1 \bigwedge (i = 0 \bigvee \boldsymbol{Judge}(gpk, i, v_i, m, \Sigma, \theta) = 0)]$$

is negligible for any PPT adversary $\mathcal{A}$.

Definition 5. A GS is said to have limited *linkability* if

$$\Pr[(i, j, m_i, m_j, \Sigma_i, \Sigma_j) \leftarrow \mathcal{A}^{StoU,Sig,LU,Link,HU,CU,RS}(gpk, ik, lk)\text{:}\\ \mathbf{Judge}(gpk, v_i, m_i, \Sigma_i, \theta_i) = 1 \bigwedge \mathbf{Judge}(gpk, v_j, m_j, \Sigma_j, \theta_j) = 1 \bigwedge\\ i = j \bigwedge \mathbf{SignLink}((\Sigma_i, m_i, \Sigma_j, m_j), lk) = 0]$$

is negligible for any PPT adversary $\mathcal{A}$.

Lemma 1. *Modified Groth GS is linkable provided DL problem is hard.*

Proof: A valid signature Σ with NIWI proof entails the existance of a valid certified signature on $\mathcal{H}(m)$. Unlike anonymity-game, in the linkability-game $\mathcal{A}$ has access to the *linking* key lk in order to find a link among signatures from the same signer while not being aware of the real signers of the signatures. Consider $\mathcal{A}$ generates a pair of signature $\varSigma_0, \varSigma_1$. Let $a_i f^{-\rho}$ and $a_j f^{-\rho}$ denote the linkability tag in the two signatures respectively. However, the adversary $\mathcal{A}$ can compute a linking index: $e(a_i f^{-\rho}, h)$ associated with a signer i. Let LM create a database that is indexed by $e(a_i f^{-\rho}, h)$. We assume this index is singular and uniformly distributed from adversarial point of view. Clearly, this index is unique and independent of the signer's signing key $gsk[i] \leftarrow x_i$. Since $a_i \leftarrow certSign(a_i, b_i)$ for member i is randomized by ρ each time to generate a in **GSign**(), there would be no security compromise.

Let there exist a simulator $\mathcal{S}$ to solve DL-problem. $\mathcal{S}$ assigns DL problem instance (u, l) to the public parameter (f) and forwarded it to $\mathcal{A}$, say f^*. If $\mathcal{A}$ wins, it generates 2 signatures $(\varSigma_0, \varSigma_1)$ that are unlink. $\mathcal{A}$ is allowed to query at most one a_i ($\mathrm{certSign}_i \leftarrow \{a_i, b_i\}$) among the users of these two signatures. If $\mathcal{A}$ queries corresponding a_i of the f^*, $\mathcal{A}$ aborts.

A valid signature implies that DL of linking tag is equal to that of f^*. Since $(\varSigma_0, \varSigma_1)$ are unlinked, the linking tag $a_i f^{-\rho}$ from $\varSigma_0$ is not equal to the linking tag $a_j f^{-\rho}$ from $\varSigma_1$.

Assume two linkability tags ($a_i f^{-\rho_0}$ and $a_j f^{-\rho_1}$) where $\rho_0 \neq \rho_1$ and $gsk[i] \neq gsk[j]$ for any identity (i, j), $e(a_0/a_1, h) \neq e(\varkappa_0/\varkappa_1, h^{lk})$. But if both the tags from the same identity i where $\rho_0 \neq \rho_1$,

$$\begin{aligned} & e(a_0/a_1, h) \overset{?}{=} e(\varkappa_0/\varkappa_1, h^{lk}) \\ \Rightarrow\ & e(a_i f^{-\rho_0}/a_i f^{-\rho_1}, h) = e(u^{-\rho_0}/u^{-\rho_1}, h^l) \\ \Rightarrow\ & e(u, h)^{l(\rho_1-\rho_0)} = e(u, h)^{l(\rho_1-\rho_0)} \end{aligned}$$

Definition 6. A GS is said to have *revocability* if

$$\begin{gathered} \Pr[(i, j, m) \leftarrow \mathcal{A}^{AU,RU,RR}(gpk);\ \varSigma \leftarrow \mathbf{GSign}(gpk, gsk[i], m); \\ \varSigma' \leftarrow \mathbf{GSign}(gpk, gsk[j], m)\text{: } \mathbf{(GVerify}(gpk, m, \varSigma) = 0 \bigvee \\ \mathbf{GVerify}(gpk, m, \varSigma') = 1) \bigwedge j \leftarrow \mathbf{RList}\ \ \bigwedge i \neq j] \end{gathered}$$

is negligible for any PPT adversary $\mathcal{A}$.

Lemma 2. *Modified Groth GS satisfies revocability under the DL-assumption.*

Proof: Issuer publishes $\mathcal{R} \leftarrow (t, gpk, \gamma, r_k)$ that includes group public key gpk and other necessary parameters in public. Note that, all the updated gpk parameters (u, f, h, T) are randomized by δ, and $\gamma := \delta^{(t/r_k)}$ is published as part of $\mathcal{R}$. γ is calculated only from the non-revoked members r_i (from $rev[i]$ pre-stored to Issuer). In order to sign a message, a non-revoked member need to create a valid $certSign$ by following

$$(a_i', b_i') \leftarrow (a_i * \delta^{-r_i}, b_i * \delta_i^{r}) \text{ s.t., } \gamma^{s_i} = \delta^{(t/r_k)*(r_i*r_k)/t}$$

However, it is impossible for a revoked member to produce new $certSign$. Because it is hard to explore δ from γ under DL-assumption. Therefore, it is hard for a PPT adversary $\mathcal{A}$ to produce a colluding non-revoked member.

Let the adversary $\mathcal{A}$ be able to link signatures generated before and after a revocation phase. Thus, in order to break backward unlinkability, $\mathcal{A}$ needs to distinguish two signatures Σ_a (generated after revocation), Σ_b (generated before revocation). It appears that Groth GS scheme provides anonymity under DLIN assumption[4]. Moreover, during each signature generation, the parameters $(a, b, \varkappa)$ are randomized by ρ, and σ is independent of the updated parameters during revocation, since it is generated from the secret x_i. Furthermore, linkability from π is also infeasible, since it is a proof from NIWI that assures indistinguishability from the secrets/witnesses it possess, based on a variant of DDH assumption. □

Definition 7. A GS is said to have *opening soundness* if

$$\begin{gathered}\Pr[(m, \Sigma, i, \theta_i, j, \theta_j) \leftarrow \mathcal{A}^{\,CU, WR}(gpk, ok, ik): \\ (\mathbf{GVerify}(gpk, m, \Sigma) = 1 \bigwedge i \neq j \bigwedge \mathbf{Judge}(gpk, v_i, m, \Sigma, \theta_i) = 1 \\ \bigwedge \mathbf{Judge}\ (gpk, v_j, m, \Sigma, \theta_j) = 1]\end{gathered}$$

is negligible for a PPT adversary $\mathcal{A}$, where adversary uses oracle $\boldsymbol{CU}(i, I)$ to set i as corrupted user and access $reg[i]$ to get user public key v_i. Details security proof is given in [9]

5 Security and Performance comparison

We minimize and exploit a simpler variant of Groth GS [10]. Therefore, we provide construction for relaxed security notions (CPA anonymity) that removes the non-essential features of the main GS. Meanwhile, we extend the existing Groth GS to satisfy some essential security notions with minor performance overhead. However, ordinary CCA-anonymous Groth GS consist of 46 group elements in $\mathbb{G}$ and 1 field element in $\mathbb{Z}_p$ while the lighter version, where CPA-anonymity is sufficient and the adversary is not allowed to access *opening* oracle, the size of signature can be reduced to 22 group elements in $\mathbb{G}$. Still it supports dynamic member enrollment, constant number of group elements in *keys* and *group signatures*, opening soundness, feasible revocation, linkability to achieve relaxed privacy through LM. Considering length of a point in $\mathbb{G}$ is 22 bytes, Signature size will be 484 bytes (approx.). In [7], the authors show how efficiency degrades in relation to pairing computation in VANET environment and propose some solutions to speed up the signature verification process. In [16], the authors address this challenge for Groth signature and propose a batch verification system to reduce almost 90% of the pairing calculation. However, introducing batch verification for single signature has reduced expensive pairing equation per signature from 68 to 11 (for CPA anonymity). If the number of signature $n \geq 2$, it needs $4n + 7$ pairing calculation. In addition, introducing off-line signature

[4] A natural extension of DDH assumption

scheduling algorithm to find an optimum value of the batch size n, and paralleling partial pairing calculation using *thread*, as described in [7], can further optimize the final operation time for signature verification.

However, allowing LM to be used by designated member (e.g., RSU in VANET) can significantly improve signature verification. As the message with signature arrives to the LM, it will first search the local database whether the sending member is already known to it (by using LM key it can easily link the incoming signature with any previous record from the same member). If the sending member is enlisted already in the receiver's local database (e.g., second (or higher) message from the same sending member), expensive verification part (e.g., 11 pairing calculation) can be omitted. For instance, if a receiving RSU requires 11 pairing calculation for the first signature it has received from a vehicle i, it presumably need no pairing calculation from the second or any subsequent signatures coming from the vehicle i until no suspicious/deceitful message is claimed by the receiving vehicle.

Table 1. compares our GS scheme with some other recent GS schemes proposed for VANET in terms of security properties, security proof method, and performance etc.

Table 1. Comparison with related VANET schemes

	Ours	Hwang *et al.*$^{'12}$[21]	Qin *et al.*$^{'11}$ [25]	Mamun *et al.*$^{'12}$[7]	Zhang *et al.*$^{'12}$[23]	Malina *et al.*$^{'13}$[28]
Security Proof	Standard	ROM	ROM	ROM	ROM	ROM
Anonymity	CPA	CPA	CCA	CCA	CPA	CPA
Linkability	Yes	Yes	No	No	No	Yes
Revocability	Yes	Yes	No	No	Yes	Yes
Non-frameability	Yes	Yes	Yes	Yes	Yes	No
Opening Soundness	Yes	No	No	No	No	No
Batch verification	Yes	No	No	Yes	No	Yes
Signature length	484 B	171 B	845 B	542 B	362 B	300 B
Signature Verification	4n+7	1n+4n(exp)	11n + 19n(exp)	3+16n(exp)	5n+11n(exp)	11n+2

6 Conclusion

In this paper, we have presented a reliable and standard CPA-secure GS solution considering revocability, linkability and opening soundness together. We mainly focus on hierarchical privacy-preserving group signature that can be used for certain applications (e.g., VANET). We consider the lighter version of Groth GS to enhance efficiency while preserving optimal security with several essential properties. Using batch verification can even significantly improve the performance of signature verification that makes the solution applicable for real-world implementation.

References

1. 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, pp. 103-108, 2007.

2. D. Chaum and E. V. Heyst. Group signatures. In EUROCRYPT, volume 547 of Lecture Notes in Computer Science, pages 257-265, 1991.
3. P. Papadimitratos, L. Buttyan, J. Hubaux, F. Kargl, A. Kung, M. Raya. Architecture for Secure and Private Vehicular Communications. In: Intl. Conference on ITS Telecomm. , pp. 16 (2007)
4. Heen, O., Guette, G., Genet, T. On the unobservability of a trust relation in mobile ad hoc networks. In Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks (pp. 1-11). Springer Berlin Heidelberg, 2009.
5. Buttyan, L., Holczer, T., Weimerskirch, A., Whyte, W. SLOW: A practical pseudonym changing scheme for location privacy in vanets. In Vehicular Networking Conference (VNC),(pp. 1-8). IEEE 2009.
6. M. Bellare, H. Shi, C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136-153. Springer, Heidelberg 2005.
7. Mamun, M. S. I., Miyaji, A. An Optimized Signature Verification System for Vehicle Ad Hoc NETwork. The 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), pp.1-8, 2012.
8. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In EUROCRYPT, volume 4965 of Lecture Notes in Computer Science, pages 415-432, 2008.
9. Y. Sakai, J. C.N. Schuldt, K. Emura, H. Hanaoka, K. Ohta. On the security of Dynamic Group Signaturs: Preventing Signature Hijacking, LNCS 7293, pp. 715-732, PKC 2012.
10. J. Groth. Fully Anonymous Group Signatures Without Random Oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164180. Springer, Heidelberg (2007)
11. J. Groth. Fully anonymous group signatures without random oracles, Feb 15 (2012) (manuscript), http://www.cs.ucl.ac.uk/staff/J.Groth/CertiSignFull.pdf
12. G. Ateniese, G. Song, and G. Tsudik. Quasi-efficient revocation of group signatures, In Financial Crypto 2002, Lecture Notes in Computer Science (LNCS), 2002.
13. E. Bresson and J. Stern. Efficient Revocation in Group Signatures, In Proceedings of Public Key Cryptography (PKC'2001), Springer-Verlag, 2001.
14. D. Song. Practical Forward-Secure Group Signature Schemes, In Proceedings of ACM Symposium on Computer and Communication Security. November 2001.
15. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In CRYPTO, LNCS(3152), pages 41-55,2004.
16. O. Blazy, G. Fuchsbauer, M. Izabachene, A. Jambert, H. Sibert, and D. Vergnaud. Batch Groth-Sahai. In Proc. ACNS 2010, volume 6123 of LNCS, pages 218-235. Springer-Verlag,2010.
17. B. Libert, T. Peters, M. Yung. Group Signatures with Almost-for-free Revocation. CRYPTO2012, LNCS7417, pp. 571-589, 2012.
18. B. Libert, T. Peters, M. Yung. Scalable Group Signature with Revocation. Eurocrypt2012, LNCS7237, pp 609-627, 2012.
19. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In CCS 04, pages 132145, New York, NY, USA, 2004. ACM Press.
20. J. Liu, W.Susilo, D. Wong. Ring signature with designated linkability. IWSEC2006, LNCS4266, pp.104-119, 2006.
21. J. Hwang, S. Lee, B. Chung, H. Cho, D. Nyang. Short Group Signatures with Controllable Linkability. In IEEE LightSec2011, Pages: 44-52, 2011.

22. J. Liu, V. Wei, and D. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In ACISP04, pages 325335. Springer-Verlag,LNCS3108, 2004.
23. L. Zhang, Q. Wu, B. Qin, J. Ferrer. Practical Privacy for Value-Added Applications in Vehicular Ad Hoc Networks. In IDCS2012, LNCS(7646), pp 43-56, 2012.
24. Chow, S. S., Susilo, W., Yuen, T. H. Escrowed linkability of ring signatures and its applications. In Progress in Cryptology-VIETCRYPT 2006 (pp. 175-192). Springer Berlin Heidelberg,2006.
25. Bo Qin, Qianhong Wu, Josep Domingo-Ferrer, and Lei Zhang, Preserving Security and Privacy in Large-Scale VANETs, ICICS 2011, LNCS 7043, pp. 121-135, 2011
26. Libert, B., Vergnaud, D. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In Cryptology and Network Security (pp. 498-517). Springer Berlin Heidelberg, 2009.
27. Nakanishi, T., Fujii, H., Yuta, H., Funabiki, N. Revocable group signature schemes with constant costs for signing and verifying. IEICE transactions on fundamentals of electronics, communications and computer sciences, 93(1), 50-62, 2010.
28. Malina, Lukas, et al. Short-Term linkable group signatures with categorized batch verification. Foundations and Practice of Security. Springer Berlin Heidelberg, 2013.