Event-Oriented k-Times Revocable-iff-Linked Group Signatures

Man Ho Au[1], Willy Susilo[1], and Siu-Ming Yiu[2]

[1] Center for Information Security Research
School of Information Technology and Computer Science
University of Wollongong, Wollongong 2522, Australia
{mhaa456, wsusilo}@uow.edu.au
[2] Department of Computer Science
The University of Hong Kong
Pokfulam, Hong Kong
smyiu@cs.hku.hk

Abstract. In this paper, we introduce the notion of event-oriented k-times revocable if and only if linked group signatures (k-EoRiffL group signatures). In k-EoRiffL group signatures, signers can sign on behalf of a group anonymously and unlinkably up to a permitted number of times (k) per *event*. No party, even the group manager, can revoke the anonymity of the signer. On the other hand, everyone can identify the signer if he signs more than k times for a particular *event*. We then show that k-EoRiffL group signatures can be used for k-times anonymous authentication(k-TAA), compact e-cash, e-voting, etc.

We formally define security model for the new notion and propose constant-size construction, that is, size of our construction is *independent* of the size of the group and the number of permitted usage k. Our construction is secure based on the q-strong Diffie-Hellman assumption and the y-DDHI assumption.

Keywords: event-oriented, revocable anonymity, group signature, k-TAA.

1 Introduction

In the age of information technology, number of applications over the Internet continues to grow. These include messaging, voting, payments, commerce, etc. At the same time, people are concerned with their personal privacy and are aware of the protection of privacy.

Anonymity is an important form of privacy protection. This is especially true in case of group-oriented cryptography, where a group of users are involved. In schemes where participation of one or a proper subset of members is required to complete a process, anonymity refers to whether participants are distinguishable from non-participants. Users may prefer perfect anonymity, meaning that it is not possible to distinguish participants from non-participants so as to maintains their privacy in participating the process. In [3], anonymity can be divided

L. Batten and R. Safavi-Naini (Eds.): ACISP 2006, LNCS 4058, pp. 223–234, 2006.

into 4 different levels, namely, *No Anonymity*, *Revocable Anonymity*, *Linkable Anonymity* and *Full Anonymity* accordingly. Extending their ideas, we further refine levels of anonymity for group-oriented cryptography as follow, from highest level to lowest level (no anonymity).

1.1 Levels of Anonymity for Group–Oriented Cryptography

Full Anonymity. It means that identity of the participating user is indistinguishable from the non-participating users by *any* party. A prominent example is ring signature, formalized in [19]. Many ring signatures are then proposed subsequently and the constant-size construction (meaning the size of the signature is independent of the size of the group) first appeared in [10], followed by [16].

Linkable Anonymity. Users can participate in the process anonymously but their participation are linked, that is, everybody can tell if the underlying participant in two separate processes are the same. An example is linkable ring signature[13, 25, 24], where everybody can tell if two signatures are generated from the same signer. However, no one can tell who the actual signer is. A generalized notion is k-times linkable anonymity, meaning that suppose the user participate for k times or less, he enjoys full anonymity while if he participate for more than k times, at least two of his participations are linked.

Revocable-iff-linked Anonymity. Similar to linkable anonymity, users enjoy full anonymity if they only participate once. However, if they participate twice, everybody can reveal their identity. Some e-cash scheme [7, 2], tracing-by-linking (TbL) group signature scheme[26] are examples of this type. In [7, 2], no one (even the bank) could revoke the anonymity of the spender of the e-cash while in case someone spends twice, his identity is revealed. A more general notion is k-times Revocable-iff-Linked anonymity, in which user identity is revealed if he participate for more than k times. Examples include compact e-cash scheme[8], k-times anonymous identification (k-TAA)[21, 17].

Revocable Anonymity. Basically it means anonymity to everybody except an *Open Authority(OA)*. From user's standpoint, it can be regarded as a lower anonymity level than Revocable-iff-Linked anonymity since in the user must trust the OA not to abuse his power in comparison with Revocable-iff-Linked anonymity where users are anonymous unless they break the condition themselves. Group signature[1] is a famous example.

Linkable and Revocable Anonymity. As its name suggest, users enjoy linkable anonymity towards everybody except OA, where OA can always revoke the anonymity of the user. Systems where users are identified by pseudonym[12] with an authority knowing the corresponding identity of the user for each pseudonym belongs to this category. Many e-cash schemes[9, 23] in fact belongs to this category too. Should a user double-spends, everybody can detect it and the OA can then reveal the identity of the cheater.

Revocable-iff-Linked and Revocable Anonymity. Similarly, users enjoy revocable-iff-linked anonymity to everybody except OA. In fact, Linkable (resp.

Anonymity Level	Examples	Size	Event-Oriented	Ad-hoc
Full	Ring Sign[19]	O(n)	N/A	✓
	Anon Ident[10, 16]	O(1)	N/A	✓
Linkable	Linkable Ring[13]	O(n)	×	✓
	Eo-Linkable Ring[25]	O(n)	✓	✓
Revocable-iff-Linked				
2-times	E-Cash[7, 2],TbL[26]	O(1)	×	×
k-times	Compact E-Cash[8]	O(1)	×	×
	k-TAA[21]	O(k)	✓	×
	dynamic k-TAA[17]	O(k)	✓	✓
	constant-size K-TAA[22]	O(1)	✓	×
	this paper	O(1)	✓	×
Full+OA	Group Signatures	O(1)	×	×
Link+OA	Fair E-Cash[9, 23]	O(1)	×	×

Fig. 1. Examples of group-oriented cryptographic schemes with different levels of anonymity

Revocable-iff-Linked) and Revocable Anonymity can be achieved by adding an identity escrow to the schemes with linkable anonymity (resp. Revocable-iff-Linked anonymity).

No Anonymity. Identity-based signature[20] is an example of group-oriented cryptography with no anonymity. Multi-signatures[14] is another example if we assume that each user is in possession of one public key only.

As stated in [3], our goal is to decide schemes with carefully adjusted level of anonymity suitable for the application. For example, ring signature is perfect for secret leaking. In an e-voting scheme, linkable anonymity or revocable-iff-linked anonymity is essential for detection of double-vote. In e-voting, linkable anonymity may be acceptable since in the vote-counting stage, the party can disregard those who double-vote. People who double-vote thus would not gain any real benefit. On the other hand, in e-cash scheme, double-spender must be caught and thus revocable-iff-linked anonymity is a must. A work around is to use scheme with linkable and revocable anonymity so that when double-spender is caught, the OA could find out who the cheater is. The problem of this work around is that anonymity of honest spender is no longer assured and trust is placed on OA not to abuse its power.

1.2 Concept of Event in Linkable Anonymity

The concept of event-oriented linkability is introduced in [25]. *Event-oriented* linkable group/ring signatures means that one can tell if two signatures are linked if and only if they are signed for the same event, despite the fact that they may be signed on behalf of different groups. This considerably add flexibility to schemes with linkable (resp. revocable-iff-linked) anonymity since user needs not obtain new secret key for different events.

In group-oriented cryptography, other concerns include whether the group can be formed in an ad-hoc manner or users must register with some group

manager first. Ring signature and group signature are example of each type respectively. Order of computation and space complexity are other concerns. Figure 1.2 categorizes *some* of the schemes in existing literature according to their level of anonymity.

1.3 Related Works

Very recently and independently, Teranishi and Sako [22] proposed an k-TAA scheme with constant proving cost. Their construction is very similar to ours and is of similar performance. Our scheme can be thought of as the non-interactive version of theirs.

Our Contributions. We introduce a new notion, event-oriented k-times revocable-iff-linked group signatures, which belongs to the Revocable-iff-Linked Anonymity category. With the event-oriented feature, this new notion is flexible for many applications such as compact e-cash, e-voting, k-times anonymous identification, to name a few. Our notion is closely related to k-TAA if we treat each content provider in k-TAA as event. Specifically, we make the following contributions

- We introduce the notion of event-oriented k-times revocable-iff-linked group signatures.
- We propose constant-size construction.
- We show how to turn k-EoRiffL group signatures into compact e-cash and k-TAA. Our scheme can be used to construct k-TAA whose size is independent of the group and also independent of k.
- We formalize the security model for k-EoRiffL group signatures and present security arguments for our scheme.

Organization. We discuss related works and technical preliminary in the next section. Security model is shown in section 3. The construction of k-EoRiffL Group Signatures is shown in section 4, accompanied by security analysis. Finally we conclude the paper with applications and some discussions in section 5.

2 Preliminaries

2.1 Notations

Let $\hat{e}$ be a bilinear map such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic multiplicative groups of prime order p.
- each element of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ has unique binary representation.
- g_0, h_0 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively.
- $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, with $\psi(h_0) = g_0$.
- (Bilinear) $\forall x \in \mathbb{G}_1$, $y \in \mathbb{G}_2$ and a, $b \in \mathbb{Z}_p$, $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$.
- (Non-degenerate)$\hat{e}(g_0, h_0) \neq 1$.

$\mathbb{G}_1$ and $\mathbb{G}_2$ can be same or different groups. We say that two groups ($\mathbb{G}_1$, $\mathbb{G}_2$) are a bilinear group pair if the group action in $\mathbb{G}_1$, $\mathbb{G}_2$, the isomorphism ψ and the bilinear mapping $\hat{e}$ are all efficiently computable.

2.2 Mathematical Assumptions

Definition 1 (Decisional Diffie-Hellman). *The Decisional Diffie-Hellman (DDH) problem in $\mathbb{G}$ is defined as follow: On input a quadruple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$, output 1 if $c = ab$ and 0 otherwise. We say that the (t, ϵ)-DDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ over random guessing in solving the DDH problem in $\mathbb{G}$.*

Definition 2 (q-Strong Diffie-Hellman[5]). *The q-Strong Diffie-Hellman (q-SDH) problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follow: On input a $(q+2)$-tuple $(g_0, h_0, h_0^x, h_0^{x^2}, \cdots, h_0^{x^q}) \in \mathbb{G}_1 \times \mathbb{G}_2^{q+1}$, output a pair (A, c) such that $A^{(x+c)} = g_0$ where $c \in \mathbb{Z}_p^*$. We say that the (q, t, ϵ)-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no t-time algorithm has advantage at least ϵ in solving the q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$.*

Definition 3. y-Decisional Diffie-Hellman Inversion Assumption[11, 8]. *The y-Decisional Diffie-Hellman Inversion problem (y-DDHI) in prime order group $\mathbb{G}$ is defined as follow: On input a $(y+2)$-tuple $g, g^x, g^{x^2}, \cdots, g^{x^y}, g^c \in \mathbb{G}^{y+2}$, output 1 if $c = 1/x$ and 0 otherwise. We say that the (y, t, ϵ)-DDHI assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ over random guessing in solving the y-DDHI problem in $\mathbb{G}$.*

2.3 Building Blocks

Verifiable Random Function. One of the building blocks of our k-ERiffL group signatures is the verifiable random function (VRF) from [11]. The notion VRF was introduced by Micali, Rabin and Vadhan in [15]. Roughly speaking, an VRF is a pseudo-random function with non-interactive proof of correctness of its output. The VRF defined in [11] is described as follow. The function f is defined by a tuple $(\mathbb{G}_p, p, g, s)$, where $\mathbb{G}_T$ is a cyclic group of prime order p, g a generator of $\mathbb{G}_p$ and s is a seed in $\mathbb{Z}_p$. On input x, $f_{\mathbb{G}_p,p,g,s}(x) = g^{\frac{1}{s+x+1}}$. Efficient proof such that the output is correctly formed (with respect to s and x in some commitment scheme such as Pedersen Commitment [18]) exists and the output of f is indistinguishable from random elements in $\mathbb{G}_p$ if the y-DDHI assumption in $\mathbb{G}_p$ holds.

3 Security Model

3.1 Syntax

An event-oriented k-times revocalbe-iff-linked group signature is a tuple (GMSetup, UserSetup, Join, Sign, Verify, Link, Revoke) of seven polynomial time algorithms. The following enumerates the syntax.

- GMSetup On input an unary string 1^λ, where λ is a security parameter, the algorithm outputs GM secret key gsk and group public key gpk. All algorithms below have implicitly gpk as one of their inputs.

- UserSetup On input 1^λ, randomly outputs a key pair (pk, sk).
- Join Protocol. User with input (pk,sk) engage with GM with input (gsk). Finally the user obtain a *cert* which allow it to sign on behalf of the group.
- Sign User with input message $m \in \{0,1\}^*$, an event identifier $evt \in \{0,1\}^*$, pk, sk, *cert* output a signature σ.
- Verify Verifier with input message $m \in \{0,1\}^*$, event identifier $evt \in \{0,1\}^*$, signature σ output accept or reject.
- Link On input two signatures σ_1, σ_2, output link or unlink.
- Revoke On input two signatures σ_1, σ_2 such that link $\leftarrow$ Link(σ_1, σ_2), output pk^*.

A event-oriented k-times revocable-iff-linked group signature must satisfy

1. *Verification Correctness.* Signatures signed according to specification are accepted during verification, with overwhelming probability;
2. *Linking Correctness.* If two signatures are linked, they must be generated from the same signer. In addition, the output of Revoke of this two signature must be the actual signer if they two signatures are on different messages.

3.2 Security Notions

We first gives an informal description of the security requirement. A *secure* k-EoRiffL Group Signatures scheme should possess *linkability*, *anonymity* and *non-slanderability*, introduce as follows.

- *Linkability.* Roughly speaking, linkability means that a user cannot sign, per event, more than the allowable times without being linked. More precisely, we required that collusion of n users cannot produce more than nk valid signatures or in case they do produce $nk+1$ signature for a particular event, at least one of the colluder must be identified. A related notion is revocability, which means that from the linked signatures, identity of the actual signer must be revealed. It is straight forward to see that revocability is implied by the definition of linkability.
- *Anonymity.* It is required that no collusion of users and GM can ever guess who the actual signer is in a group signature with probability better than random guessing.
- *Non-slanderability.* It is required that an honest user cannot be accused of having sign more than k times, even with the help of GM.

In revocable-iff-linked group signatures, the standard notion of *unforgeability* is implied by *linkability* and *non-slanderability*. For if someone can forge a signature, either he can generate the signature without being *linked* or he successfully *slander* an honest user.

The capability of an adversary $\mathcal{A}$ is modeled as oracles.

- Join Oracle: $\mathcal{A}$ present a public key pk and engages in the join protocol as user and obtains a certificate. The oracle stores pk in a set $\mathbb{X}_A$.

- Signing Oracle: On input a message m, and event evt, the oracle return a signature σ on m and evt.
- Hash Oracle: $\mathcal{A}$ can ask for the values of the hash functions for any input.

We require that the answers from the oracles are indistinguishable from the view as perceived by an adversary in real world attack.

Definition 4 (Game Linkability)

- (Initialization Phase.) *The challenger $\mathcal{C}$ takes a sufficiently large security parameter λ and runs* `GMSetup` *to generate gpk and also a master secret key gsk. $\mathcal{C}$ keeps gsk to itself and sends gpk to $\mathcal{A}$.*
- (Probing Phase.) *The adversary $\mathcal{A}$ can perform a polynomially bounded number of queries to the oracles in an adaptive manner.*
- (End Game Phase.) *Let q_j be the number of queries to the Join Oracle. $\mathcal{A}$ submits an event evt∗, signatures σ_i on message m_i and evt^*, $i = 1, \cdots, kq_j + 1$ to $\mathcal{C}$.*

$\mathcal{A}$ wins the game if all the following holds:

1. *all σ_i are valid*
2. *none of the σ_i are the output of the Signing Oracle*
3. *None of the σ_i are linked or they are linked but* Revoke *cannot pointed to any of the users during the join protocol query.*
4. *$m_i \neq m_j$ if $i \neq j$.*

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

In the above game, if the condition such that each m_i are different is replaced by each σ_i are different, then we refer to the game as Game Strong Linkability.

Definition 5 (Game Anonymity)

- (Initialization Phase.) *The challenger $\mathcal{C}$ takes a sufficiently large security parameter λ and runs* `GMSetup` *to generate gpk and also the master secret key gsk. $\mathcal{C}$ gives* both *gpk and gsk to the user. Since $\mathcal{A}$ is in possession of gsk, only Hash oracle query is allowed in Game Anonymity.*
- (Challenge Phase.) *$\mathcal{C}$ runs the* Join *protocol with $\mathcal{A}$ acting as GM to obtain a certificate $cert_0$. $\mathcal{C}$ generate another certificate $cert_1$ by himself. $\mathcal{A}$ is then allowed to issue the following special signature query by submitting event evt_i, message m_i, bit $b_i = 0$ or 1 for the i-th special signature query. $\mathcal{C}$ return a signature on evt_i, m_i using $cert_{b_i}$. The only restriction is that for a particular event, the number of signature query for $cert_0$ or $cert_1$ does not exceed k. Finally, $\mathcal{A}$ gives evt^*, m^* to $\mathcal{C}$, $\mathcal{C}$ uses $cert_b$, where $b \in \{0, 1\}$ is the output of a fair coin, to sign on evt^*, m^* and return the signature to $\mathcal{A}$.*
- (End Game Phase.) *The adversary $\mathcal{A}$ decides $b = 0$ or 1.*

$\mathcal{A}$ wins the above game if it guesses correctly. The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins minus $\frac{1}{2}$.

Definition 6 (Game Non-Slanderability)

- (Initialization Phase.) *The challenger $\mathcal{C}$ takes a sufficiently large security parameter λ and runs* `GMSetup` *to generate gpk and also the master secret key gsk. $\mathcal{C}$ gives* both *gpk and gsk to the user. Since $\mathcal{A}$ is in possession of gsk, only Hash oracle query is allowed in Game Non-Slanderabiliy.*
- (Challenge Phase.) *$\mathcal{C}$ runs the* Join *protocol for q_j times with $\mathcal{A}$ acting as GM to obtain a certificate $cert_j$. $\mathcal{A}$ is then allowed to issue the following special signature query by submitting event evt_i, message m_i, $b_i = 1, \cdots, q_j$ for the i-th special signature query. $\mathcal{C}$ return a signature on evt_i, m_i using $cert_{b_i}$. The only restriction is that for a particular event, the number of signature query for any of the cert does not exceed k. $\mathcal{A}$ is also allowed to corrupt the user corresponding to $cert_j$.*
- (End Game Phase.) *$\mathcal{A}$ submits an event $evt*$, two signatures σ_0^*, σ_1^* on message m_0, m_1 and evt^*. $\mathcal{A}$ wins the game if σ_0^*, σ_1^* is linked and* Revoke *on the two linked signature is a user in one of the join query and is not corrupted.*

The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins.

A k-EoRiffL Group signature is *secure* if no PPT adversary can win in Game Linkability, Game Anonymity and Game Non-Slanderability with non-negligible advantage. It is *strongly secure* if it is secure and no PPT adversary can win in Game Strong Linkability.

4 Our Construction

4.1 Our *k*-EoRiffL Group Signature

GMSetup. Let λ be the security parameter. Let $(\mathbb{G}_1, \mathbb{G}_2)$ be a bilinear group pair with computable isomorphism ψ as discussed such that $|\mathbb{G}_1| = |\mathbb{G}_2| = p$ for some prime p of λ bits. Also assume $\mathbb{G}_p$ be a group of order p where DDH is intractable. Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_{evt}\{0,1\}^* \rightarrow \mathbb{G}_p$ be cryptographic hash functions. Let g_0, g_1, g_2, g_3 be generators of $\mathbb{G}_1$, h_0, h_1, h_2, h_3 be generators of group $\mathbb{G}_2$ such that $\psi(h_i) = g_i$ and u_0, u_1, u_2, u_3 be generators of $\mathbb{G}_p$ such that relative discrete logarithm of the generators are unknown. This can be done by setting the generators to be output of a hash function of some publicly known seed. The group manager (GM) also randomly selects $\gamma \in_R \mathbb{Z}_p^*$ and compute $w = {h_0}^{\gamma}$. The group public key is $gpk = (g_0, g_1, g_2, g_3, h_0, w, u_0, u_1, u_2, u_3, k)$ and the GM secret key is $gsk = \gamma$. k has to be much smaller than 2^{λ}.

UserSetup. We assume PKI, that is, each user is equipped with a discrete logarithm type public and private key pairs $({u_0}^x, x) \in \mathbb{G}_p \times \mathbb{Z}_p^*$.

Join Protocol. **Idea:** User with public key $y = {u_0}^x$ joins the group by obtaining a cert in the form of (A, e) such that $A^{e+\gamma} = g_0 g_1^s g_2^t g_3^x$ for some random number s, t unknown to GM.

Actual Protocol

1. User randomly selects $s' \in_R \mathbb{Z}_p^*$ and sends $C' = g_1^{s'} g_2^t g_3^x$, along with the proof $\Pi_0 = SPK\{(s', t, x) : C' = g_1^{s'} g_2^t g_3^x \wedge y = u_0^x\}$ to GM.
2. GM verifies that Π_0 is valid and randomly selects $s'' \in_R \mathbb{Z}_p^*$. It computes $C = C' g_1^{s''}$ and selects $e \in_R \mathbb{Z}_p^*$. It then computes $A = (g_0 C)^{\frac{1}{e+\gamma}}$ and sends (A, e, s'') to the user.
3. User computes $s = s' + s''$and checks if $\hat{e}(A, w h_0^e) = \hat{e}(g_0 g_1^s g_2^t g_3^x, h_0)$. It then stores (A, e, s, t).

Sign. **Idea:** For each event $evt \in \{0,1\}^*$, the user manages a counter J_{evt} which is the number of signatures he has generated. When the counter reaches k, user can no longer signs anonymously.

For a particular event evt^* and message m such that $R = H(evt^*, m)$ and $u_{evt^*} = H_{evt}(evt^*)$, user with (A, e, s, t) from GM and counter $J_{evt^*} \leq k$ produces a signature of knowledge by submitting $S = u_{evt^*}^{\frac{1}{J_{evt^*}+s+1}}, T = u_0^x u_{evt^*}^{\frac{R}{J_{evt^*}+t+1}}$ and proves, in zero-knowledge manner, (1) - (4).

1. $A^{e+\gamma} = g_0 g_1^s g_2^t g_3^x$ (This shows that the user possess a certificate from GM.)
2. $S = u_{evt^*}^{\frac{1}{J_{evt^*}+s+1}}$. ($S$ is called a *linkability tag* or simply *tag*. For each certificate (A, e, s, t) and event evt^*, user can generate k valid *tag*. Suppose a user generate more than k *tags* from the same certificate, duplicate *tags* must be used and can thus be detected.)
3. $0 \leq J_{evt^*} \leq k$ (The number of signings does not exceed k.)
4. $T = u_0^x u_{evt^*}^{\frac{R}{J_{evt^*}+t+1}}$ (Component for revealing identity of user using duplicated tags.)

Should a user attempt to sign more than the permitted number of times k for a particular event, he must have used duplicated *tag* and can thus be detected. Then two transcripts with the same *tag* together with different T reveals identity of user. Details are shown in the revoke algorithm. On the other hand, anonymity of honest signer is guaranteed. In short, the above can be represented by

$$\Pi_1 = SPK\{(A, e, s, t, x, J_{evt^*}) : A^{e+\gamma} = g_0 g_1^s g_2^t g_3^x \wedge S = u_{evt^*}^{\frac{1}{J_{evt^*}+s+1}}$$
$$\wedge T = u_0^x u_{evt^*}^{\frac{R}{J_{evt^*}+t+1}} \wedge 0 \leq J_{evt^*} \leq k\}(M)$$

Remarks: Two signature for the same event can be falsely 'linked' if $J_{evt^*} + s = J'_{evt^*} + s'$. However, the probability is negligible if $k << 2^\lambda$.

Details of Sign (instantiation of SPK Π_1) is shown in Appendix A.

Verify. The verifier verifies the SPK.

Link. For the same *event* evt^*, two signatures are from the same user (*linked*) if they share the same *tag* S.

Revoke. Given two signatures with same *tag* for the same event evt^* and different messages m, anyone can compute $u_o^x = (\frac{T^{R'}}{T'^R})^{((R'-R)^{-1})}$.

4.2 Security Analysis

Regarding our construction, we have the following theorem, whose proof can be found in the full version of the paper [4].

Theorem 1. *Our k-RiffL group signature is secure under the q-SDH assumption and the y-DDHI assumption in the random oracle model.*

We remark that our scheme does not possess strong linkability, meaning that two linked signatures of the same message from the same signer is not revocable. Thus, our scheme maybe best suit to be used in interactive protocols where (part of) the message maybe provided by another party to ensure its uniqueness. Examples include transaction information provided by the merchant in e-cash or a random seed provided by the content provider in k-TAA.

5 Applications and Discussions

5.1 Constant-Size k-TAA

If the verifier's identity is appended to the event *evt*, then we have a event-oriented k-times revocable-iff-linked group signatures which is verifier-specific, that is, the signatures for different verifiers will not be linked with one another. It is straight forward to show that k-TAA can be constructed from k-EoRiffL group signature by setting the event to be the identity of the verifier(content provider). Our k-EoRiffL group signature is the first k-TAA scheme which is of size O(1) (independent of group size and k).

On the other hand, if we treat the tags published by the content provider in k-TAA as event then any k-TAA can be turned into k-EoRiffL group signatures if the underlying k-TAA can be done in an non-interactive manner.

5.2 Compact E-Cash

In fact, compact E-Cash can be viewed as a k-times revocable-iff-linked group signature. To use the k-RiffL group signature as a compact e-cash scheme, the bank plays the role of GM in the scheme while the join protocol can be treated as users obtain a wallet from the bank (i.e. withdrawing k coins). Spending is done by using the certificate to sign. Since each certificate can be used to sign k-times, each wallet possesses k coins. When the wallet is used up, user need to obtain another certificate from the bank. Note that the concept of event is not used.

References

1. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO*, pages 255–270, 2000.
2. Man Ho Au, Sherman S. M. Chow, and Willy Susilo. Short e-cash. In *INDOCRYPT*, pages 332–346, 2005.

3. Man Ho Au, Joseph K. Liu, Patrick P. Tsang, and Duncan S. Wong. A Suite of ID-Based Threshold Ring Signature Schemes with Different Levels of Anonymity. Cryptology ePrint Archive, Report 2005/326, 2005. `http://eprint.iacr.org/`.
4. Man Ho Au, Willy Susilo, and Siu-Ming Yiu. Event-Oriented k-times Revocable-iff-Linked Group Signatures, 2006. Full version.
5. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *EUROCRYPT*, pages 56–73, 2004.
6. Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT*, pages 431–444, 2000.
7. Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In *CRYPTO*, pages 302–318, 1993.
8. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT*, pages 302–321, 2005.
9. Sébastien Canard and Jacques Traoré. On fair e-cash systems based on group signature schemes. In *ACISP*, pages 237–248, 2003.
10. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In *EUROCRYPT*, pages 609–626, 2004.
11. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *PKC 2005*, volume 3386 of *LNCS*, pages 416 – 431, 2005.
12. Wei-Bin Lee and Chang-Kuo Yeh. A new delegation-based authentication protocol for use in portable communication systems. *IEEE Trans. Wireless Commun.*, 4(1):57–64, January 2005.
13. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract). In *ACISP 2004*, volume 3108 of *LNCS*, pages 325–335. Springer-Verlag, 2004.
14. Silvio Micali, Kazuo Ohta, and Leonid Reyzin. Accountable-subgroup multisignatures: extended abstract. In *ACM Conference on Computer and Communications Security*, pages 245–254, 2001.
15. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *FOCS*, pages 120–130, 1999.
16. Lan Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292, 2005.
17. Lan Nguyen and Rei Safavi-Naini. Dynamic k-Times Anonymous Authentication. Cryptology ePrint Archive, Report 2005/168, 2005. `http://eprint.iacr.org/`.
18. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.
19. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565, 2001.
20. Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO 1984*, volume 196 of *LNCS*, pages 47–53, 1984.
21. Isamu Teranishi, Jun Furukawa, and Kazue Sako. k-Times Anonymous Authentication (Extended Abstract). In *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 308–322. Springer-Verlag, 2004.
22. Isamu Teranishi and Kazue Sako. k-times anonymous authentication with a constant proving cost. In *Public Key Cryptography*, pages 525–542, 2006.
23. Mårten Trolin. A universally composable scheme for electronic cash. In *INDOCRYPT*, pages 347–360, 2005.
24. Patrick P. Tsang and Victor K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC*, pages 48–60, 2005.

25. Patrick P. Tsang, Victor K. Wei, Tony K. Chan, Man Ho Au, Joseph K. Liu, and Duncan S. Wong. Separable Linkable Threshold Ring Signatures. In *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 384–398. Springer-Verlag, 2004.
26. Victor K. Wei. Tracing-by-linking group signatures. In *ISC*, pages 149–163, 2005.

A Detail of Sign Algorithm (Instantiation of Π_1)

Suppose $u_{evt^*} = H_{evt}(evt^*)$ and $J_{evt^*} < k$, $R = H(evt^*, m)$, the signer first computes the following quantities $A_1 = g_1^{r_1} g_2^{r_2}$, $A_2 = A g_2^{r_1}$, $A_3 = g_1^{J_{evt^*}} g_2^t g_3^{r_3}$, for $r_1, r_2, r_3 \in_R \mathbb{Z}_p^*$, in $\mathbb{G}_1$. Compute *tag* $S = u_{evt^*}^{\frac{1}{J_{evt^*}+s+1}}$, $T = u_0^x u_{evt^*}^{\frac{R}{J_{evt^*}+t+1}}$. The signer then computes a signature of knowledge (instantiation of Π_1) as follows.

$\Pi_2 = SPK\{(r_1, r_2, r_3, \delta_1, \delta_2, \delta_3, \delta_J, \delta_t, e, s, t, x, J_{evt^*}) : A_1 = g_1^{r_1} g_2^{r_2} \wedge A_1^e = g_1^{\delta_1} g_2^{\delta_2} \wedge \frac{e(A_2,w)}{e(g_0,h_0)} = e(g_1,h_0)^s e(g_2,h_0)^t e(g_3,h_0)^x e(g_3,h_0)^{\delta_1} e(g_2,w)^{r_1} e(A_2,h_0)^{-e} \wedge \frac{u_{evt^*}}{S} = S^{J_{evt^*}} S^s \wedge A_3 = g_1^{J_{evt^*}} g_2^t g_3^{r_3} \wedge A_3^x = g_1^{\delta_J} g_2^{\delta_t} g_3^{\delta_3} \wedge \frac{u_{evt^*}^R}{T} = T^{J_{evt^*}} T^t u_0^{-\delta_J} u_0^{-\delta_t} u_0^x \wedge 1 \leq J_{evt^*} \leq k\}(M)$ where $\delta_1 = r_1 e, \delta_2 = r_2 e, \delta_J = J_{evt^*} x, \delta_t = tx, \delta_3 = r_3 x$.

The range part $1 \leq J_{evt^*} \leq k$ require some attention. Secure and efficient exact proof of range is possible in groups of unknown order under factorization assumption [6]. Here, we make use of the fact that if we set $k = 2^t$ for some integer t, efficient range check, of order $O(logk)$, for J_{evt^*} could be achieved as follows.

Let g, h be two generators of a cyclic group $\mathbb{G}$ of order p whose relative discrete logarithm is unknown. To prove knowledge of a number J such that $0 < J \leq k$ in a commitment $C_J = g^J h^r$, let J_i be the i-th bit of J for $i = 1, \cdots t$. Compute $C_i = g^{J_i} h^{r_i}$ for some $r_i \in_R \mathbb{Z}_p^*$ for $i = 1, \cdots, t$. Compute the following SPK Π_{range}.

$\Pi_{range} = SPK\{(J, a, b, r, r_i) : C_J = g^J h^r \wedge C_J/g = g^a h^r \wedge \prod_{j=1}^t (C_j)^{2^j} = g^J h^b \wedge [C_i = h^{r_i} \vee C_i/g = h^{r_i}]_{i=1}^{i=t}\}(M)$ where $a = J - 1$, $b = \prod_{j=1}^t r_j 2^j$.

On the other hand, constant-size range proof is made possible as outlined in [22]. The GM has to publish k signatures $Sig(1), \cdots, Sig(k)$. In the proof, instead of proving $1 \leq J_{evt^*} \leq k$ (which has complexity $O(logk)$), the signer proves possession of signature on J_{evt^*} (which has complexity $O(1)$). This indirectly proves that J_{evt^*} is within the range. However, public key size of the GM is now linear in k, and user colluding with GM can be untraceable (since the malicious GM can issue several $Sig(J_{evt^*})$ for the user.