Anonymizable Signature and Its Construction from Pairings

Fumitaka Hoshino, Tetsutaro Kobayashi, and Koutarou Suzuki

NTT Information Sharing Platform Laboratories, NTT Corporation,
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
{hoshino.fumitaka,kobayashi.tetsutaro,suzuki.koutarou}@lab.ntt.co.jp

Abstract. We present the notion of *anonymizable signature*, which is an extension of the ring signature [RST01, BKM06]. By using an anonymizable signature, anyone who has a signed message can convert the signature into an anonymous signature. In other words, one can leave a signed message with an appropriate agent who will later anonymize the signature.

A relinkable ring signature [SHK09] is also an extension of the ring signature by which the ring forming ability can be separated from the signing ability. In the relinkable ring signature, an agent who has a special key given by the signer can modify the membership of existing ring signatures. However, the relinkable ring signature has two problematic limitations; a signer cannot select an agent according to the worth of the signature, because there exists the unique key to modify the membership for each public key, and we cannot achieve perfect anonymity even if the agent is honest.

The proposed anonymizable signature can free one from these limitations. In the anonymizable signature scheme, each signature can be anonymized without any secret but the signature itself. Thus, the signer can delegate signature anonymization to multiple agents signature by signature. Moreover, the anonymizable signature can guarantee unconditional anonymity and be used for anonymity-sensitive purposes, e.g., voting. After providing the definition of the anonymizable signature, we also give a simple construction methodology and a concrete scheme that satisfies perfect anonymity and computational unforgeability under the gap Diffie-Hellman assumption with the random oracle model.

1 Introduction

We present the notion of *anonymizable signature*, which is an extension of the ring signature [RST01, BKM06]. By using an anonymizable signature, anyone who has a signed message can convert the signature into an anonymous signature, i.e., one can leave a signed message with an appropriate agent who will later anonymize the signature.

For example, in the case of publication of a governmental document through a "freedom of information act", the governmental staff who publicize the document need to hide the personal information of the individuals in the document. To

M. Joye, A. Miyaji, and A. Otsuka (Eds.): Pairing 2010, LNCS 6487, pp. 62–77, 2010.

hide this information, they can use a sanitizing signature [SBZ01]. However, if the document is a contract between the government and an individual, the information on the signer cannot be hidden because readers must be convinced that the contract is valid.

To hide the information on the signer, it may seem to be effective for the signer to leave an additional ring signature with the document at the signing phase, which will replace his signature at the publishing phase. However, during the long term preservation of the document, there is a large probability that the ring signature will be made invalid by leakage of a signing key of a member in the ring, even if there is little probability that a member would leak his or her signing key. In such a case, the anonymizable signature could be effective. A signer can leave a signed message with the governmental staff who will convert it into an appropriate ring signature with a valid ring at the publishing phase.

If the agent is not trustworthy, the anonymizable signature is not suitable for an application that needs highly strong anonymity, such as whistleblowing. However, if the agent can be regarded as an ideal functionality, such a signature has many applications, e.g., server-aided computation of ring signatures, sanitizing of the signer in a signed documents, dynamic group management of ring signatures, as a countermeasure against the invalidation attack by the secret key exposure, as a gradually convertible ring signature [SHK09].

1.1 Related Works

An anonymizable signature is an extension of the ring signature [RST01, BKM06]. A ring signature is a kind of anonymous signature by which one can sign anonymously without a group setup or group manager. For each time of signing, a signer of a ring signature chooses a set of appropriate members, called a ring, then signs a message on behalf of the ring by using his or her own secret key and all of the public keys of the members in the ring. A signer can form any ring that includes him or herself as long as he or she has the public keys.

A relinkable ring signature [SHK09] is also an extension of the ring signature and has a similar functionality to the anonymizable signature. By using a relinkable ring signature, one can separate the ring-forming ability from the signing ability. Besides normal ring signature algorithms, the relinkable ring signature has a special algorithm called *relink*. An agent who has a special key given by the signer can derive a ring signature with a modified ring from an existing signed message by use of the relink algorithm. The agent who has the key to modify a ring signature can just modify the ring membership of an existing signed message; he or she cannot sign a new message. Thus, the signer can transfer the ring-forming ability to an agent without leaking the signing key. An anonymizable signature can be regarded as an extension of a relinkable ring signature, s.t., for each signing phase, the signer generates a new key, namely the signature.

An anonymizable signature also can be regarded as an extension of the convertible ring signature [LWH05], by which the signer can generate a key to revoke the anonymity of the ring signature. An agent who has an anonymizable signature can gradually decrease the anonymity of signed message by decreasing the

number of ring members of the signature. In an extreme case, the agent can generate a non-anonymous signature by making the ring members include only the signer. This is similar to a convertible ring signature [LWH05], although the agent never proof that the two ring signatures are generated from the same anonymizable signature.

Moreover an anonymizable signature can be regarded as an extension of the ID-based signature (IBS) [Sha84, BNN04]. The syntax of an anonymizable signature is similar to that of an IBS, although the roles of the participants are somewhat different.

1.2 Motivation

In the definition of a relinkable ring signature in [SHK09], a special secret called *relink key* is given by the signer at the key generation phase. One can modify the ring membership of an existing ring signature by use of the signer's relink key. Every public key has only one corresponding relink key. Once an agent has a relink key, it remains valid until the public key is revoked. This correspondence gives rise to many problematic limitations.

For example, a signer cannot select an agent according to the worth of the signature because any agent who has the relink key can always relink any signatures that the signer generates. Moreover, the signer cannot avoid the risk that an agent may relink unexpected ring signatures without his or her permission.

Furthermore, we cannot achieve a perfectly anonymous relinkable ring signature without strong limitations even if the agent is honest. To prohibit an agent from excluding the actual signer from the ring, the signer and the agent must share the information on the actual signer in the ring. If they share the information for each ring signature, they need large secrets whose size is proportional to the number of ring signatures. However, according to the relinkable ring signature, the only difference between the agent and the verifier is whether he or she has the relink key. It is impossible to hide the information perfectly beyond the size of the shared secret. Thus, according to their definition, it is impossible to achieve perfect anonymity without strong limitation of the number of signatures or giving some additional information to the agent.

Therefore, to construct an efficient relinkable ring signature, we must give up the idea of achieving perfect anonymity. Indeed, the concrete scheme proposed in [SHK09] just achieves a weaker notion of computational anonymity. Even though the agent is honest, an attacker with unexpected computational resources may derive the actual signer directly from a ring signature.

All of these problems are caused by the flaw in the definition of a relinkable ring signature.

1.3 Contributions

In this paper, we present an improved notion, which we call anonymizable signature. In the anonymizable signature scheme, the signing algorithm creates a signature on a message. The signature can be converted to an anonymous ring

signature, while the signed message cannot be changed. By using an anonymizable signature, the signer passes a message and a signature on the message to a proxy agent. The proxy agent can convert the signature into a ring signature afterward. We provide the definition of anonymizable signature, a simple construction methodology based on the non-interactive proof of knowledge of a signature, and an anonymizable signature scheme that can be proven to be unconditionally anonymous and computationally unforgeable under the GDH assumption in the random oracle model.

2 Definition

2.1 Notations

When X is a probabilistic Turing machine, $X(Y)$ denotes that X takes Y as an input. If X has an output value, $X(Y)$ also denotes the output value of X when X takes Y as an input. To simplify the description, we will omit some inputs that are not relevant, e.g., random tape, security parameter, and common reference string.

When Y is a fixed value, $X \leftarrow Y$ denotes that a value Y is assigned to a variable X. When Y is a set, $X \xleftarrow{\$} Y$ denotes that X is uniformly selected from Y. When Y is a probabilistic Turing machine, $X \xleftarrow{\$} Y()$ denotes that X is randomly selected from the output space of Y according to the distribution of Y's output when Y's random tape is uniformly selected. $X() \xrightarrow{\$} Y$ denotes that X is a probabilistic Turing machine that outputs Y. For any binary operator $\circ$, $X \xleftarrow{\circ} Y$ denotes that a new value $X \circ Y$ is assigned to the variable X. For instance, when Y is a set, $X \xleftarrow{\cup} Y$ denotes that a value $X \cup Y$ is assigned to the variable X. $\star$ denotes an appropriate string that is not relevant. $X \stackrel{?}{=} Y$ denotes a Boolean value equivalent to 1 if $X = Y$ and equivalent to 0 otherwise.

2.2 Syntax

Let $k \in \mathbb{N}$ be the security parameter. Let $N = \{0, 1, \ldots\}$ be the set of signers; we denote a subset of signers by $L \subset N$. We denote by x_i the secret key and by y_i the public key of member $i \in N$. We use notation $a_L = (a_i)_{i \in L}$.

The anonymizable signature scheme Σ consists of the following four algorithms: $\Sigma = (\mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Anonymize}, \mathtt{Verify})$.

Key Generation $\mathtt{KeyGen}(1^k) \xrightarrow{\$} (x, y)$: The key generation algorithm is a probabilistic poly-time algorithm that takes a security parameter k as input and outputs a secret key x and a public key y.

Signing $\mathtt{Sign}(x, m) \xrightarrow{\$} r$: The signing algorithm is a probabilistic poly-time algorithm that takes a secret key x and a message m as input and outputs a signature r.

Anonymization $\mathtt{Anonymize}(r, L, y_L, m) \xrightarrow{\$} \sigma / \perp$: The anonymization algorithm is a probabilistic poly-time algorithm that takes a signature r, ring $L \subset N$,

list y_L of public keys, and message m as input and outputs a ring signature σ or $\perp$ as rejection.

Verification $\texttt{Verify}(L, y_L, m, \sigma) \xrightarrow{\$} 0/1$: The verification algorithm is a probabilistic poly-time algorithm that takes a ring $L \subset N$, list y_L of public keys, message m, and ring signature σ as input and outputs a single bit $b \in \{0, 1\}$.

Note that we can always verify i's signature r of a message m as

$$\texttt{Verify}(\{i\}, y_i, m, \texttt{Anonymize}(r, \{i\}, (y_i), m)) \stackrel{?}{=} 1.$$

To avoid violating the signer's anonymity, we must treat the signature r as a secret between the signer and the agent, in contrast to the ring signature σ. A signature r can be regarded as a secret seed of a ring signature σ.

The syntax of an anonymizable signature is similar to that of an ID-based signature (IBS) [Sha84, BNN04], although the roles of the participants are somewhat different. We can regard the signing algorithm in an anonymizable signature as the key extraction in an IBS. The key extraction in an IBS generates a signing key for each ID, while the signing algorithm in an anonymizable signature generates a signing key, i.e., an anonymizable signature, for each message. In other words, a ring signature derived from an anonymizable signature is a special case of an ID-based signature such that the ID is identical to the message if it's ring consists of only the signer. Later in this paper, we will present a concrete scheme based on the BLS signature [BLS01], which can be seen as a scheme based on the ID-based Schnorr signature [SK03].

2.3 Security

The security of an anonymizable signature scheme is defined as follows. First, we prepare definitions of some oracles. Let $L^0 = \{1, ..., n\}$ be the set of indices of initially registered public keys, $L^{sk} \subset L^0$ be the set of indices for which secret key exposure oracle $\mathcal{O}_{sk}$ is called, and $L^{kr} = \{n+1, n+2, ...\}$ be the set of indices of public keys registered by adversary via key registration oracle $\mathcal{O}_{kr}$.

Signing Oracle $\mathcal{O}_s(i, m) \xrightarrow{\$} \mu$: The signing oracle takes a signer $i \in L^0$ and a message m as input and outputs a document index $\mu \in \mathbb{N}$ as follows:

1. if $i \in L^0$ and m is in valid domain, then set $r \xleftarrow{\$} \texttt{Sign}(x_i, m)$, otherwise set $r \leftarrow \perp$,
2. increment document counter $\hat{\mu} \in \mathbb{N}$ that is a state information, and set $\mu \leftarrow \hat{\mu}$
3. update list of signing oracle queries and answers as $Q_s \xleftarrow{\cup} \{(\mu, i, m, r)\}$,
4. return μ.

Anonymization Oracle $\mathcal{O}_a(\mu, L) \xrightarrow{\$} \sigma$: The anonymization oracle takes a document index $\mu \in \mathbb{N}$ and a list of ring members $L \subset L^0 \cup L^{kr}$ as input and outputs a ring signature σ as follows:

1. if μ is registered in Q_s and $L \subset L^0 \cup L^{kr}$, then find (i, m, r) s.t. $(\mu, i, m, r) \in Q_s$ and set $\sigma \xleftarrow{\$} \texttt{Anonymize}(r, L, y_L, m)$, otherwise set $\sigma \leftarrow \perp$,

2. update list of anonymization oracle queries and answers as $Q_a \xleftarrow{\cup} \{(\mu, i, m, r, L, \sigma)\}$,
3. return σ.

Signature Exposure Oracle $\mathcal{O}_e(\mu) \xrightarrow{\$} r$: The signature exposure oracle takes a document index $\mu \in \mathbb{N}$ as input and outputs a signature r as follows:

1. if μ is registered in Q_s, then find (i, m, r) s.t. $(\mu, i, m, r) \in Q_s$, otherwise set $r \leftarrow \perp$,
2. update list of anonymization oracle queries and answers as $Q_e \xleftarrow{\cup} \{(\mu, i, m, r)\}$,
3. return r.

Secret Key Exposure Oracle $\mathcal{O}_{sk}(i) \xrightarrow{\$} x_i$: The secret key exposure oracle takes a user index $i \in \mathbb{N}$ as input and outputs secret key x_i of i-th user.

1. if $i \in L^0$, then set $sk \leftarrow x_i$, otherwise set $sk \leftarrow \perp$ and return sk,
2. update the set of indices for which secret key exposure oracle is called, $L^{sk} \xleftarrow{\cup} \{i\}$,
3. return sk.

Key Registration Oracle $\mathcal{O}_{kr}(y) \xrightarrow{\$} i$: The key registration oracle takes a public key y as input, outputs a new user index i, and register y as the public key of the i-th user.

1. if y is in valid domain, then increment counter $\hat{i} \in \mathbb{N}$ that is a state information, set $i \leftarrow \hat{i}$, and register y as the public key of the i-th user, otherwise set $i \leftarrow \perp$ and return i,
2. update the set of indices of public keys registered by adversary via key registration oracle, $L^{kr} \xleftarrow{\cup} \{i\}$,
3. return i.

We say that anonymizable signature Σ is secure if it satisfies the following three properties.

Completeness

Correctly generated signatures are accepted with overwhelming probability. We say that anonymizable signature Σ satisfies *completeness*, if

$$\Pr[(x_j, y_j) \xleftarrow{\$} \mathtt{KeyGen}(1^k)\ (j \in L), r \xleftarrow{\$} \mathtt{Sign}(x_i, m),$$
$$\sigma \xleftarrow{\$} \mathtt{Anonymize}(r, L, y_L, m), \mathtt{Verify}(L, y_L, m, \sigma) = 0]$$

is negligible in k for any message $m \in \{0,1\}^*$, any ring $L \subset N$, and any signer $i \in L$.

Anonymity

We consider the following experiment $Exp^{\mathrm{anon}}_{k,\Sigma}(\mathcal{A})$, where adversary $\mathcal{A}$ try to distinguish the signer of a ring signature:

1. select random bit $b \xleftarrow{\$} \{0,1\}$,
2. at the beginning of the experiment, adversary $\mathcal{A}$ generates two pairs of secret and public keys $(x_0, y_0), (x_1, y_1)$, register two public keys y_0, y_1 by key registration oracle $\mathcal{O}_{kr}$, and outputs two secret keys x_0, x_1, and the experiment is aborted if the secret keys are not correct w.r.t. the public keys,
3. adversary $\mathcal{A}$ can access key registration oracle $\mathcal{O}_{kr}(y)$ adaptively during the experiment, and we denote the set of indices of all public keys registered by adversary $\mathcal{A}$ during the experiment by $L^{kr} = \{0, 1, ...\}$,
4. adversary $\mathcal{A}$ outputs (m^*, L^*) s.t. $0, 1 \in L^* \subset L^{kr}$,
5. generate ring signature $\sigma^* \xleftarrow{\$} \texttt{Anonymize}(\texttt{Sign}(x_b, m^*), L^*, y_{L^*}, m^*)$ using the secret key of signer b, and adversary $\mathcal{A}$ is given σ^*,
6. finally, adversary $\mathcal{A}$ outputs a bit $b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{kr}}(x_1, ..., x_n)$,
7. return 1 if $b = b'$, 0 otherwise.

We define the advantage Adv^{anon} of the adversary $\mathcal{A}$ as

$$Adv^{\text{anon}}_{k,\Sigma}(\mathcal{A}) = \left| \Pr\left[Exp^{\text{anon}}_{k,\Sigma}(\mathcal{A}) = 1\right] - \frac{1}{2} \right|,$$

where probability is taken over a random bit b, keys, random tapes of the oracles, random tapes of the adversary $\mathcal{A}$, and random tapes of `KeyGen`, `Sign`, `Anonymize`.

Definition 1 (computational anonymity w.r.t. adversarially-chosen keys). *An anonymizable signature Σ satisfies* computational anonymity w.r.t. adversarially-chosen keys *if $Adv^{anon}_{k,\Sigma}(\mathcal{A}_k)$ is negligible in k for any probabilistic poly-time adversary $\mathcal{A}_k$.*

In particular, we say that anonymizable signature Σ satisfies *perfect anonymity* if we have

$$\Pr[\texttt{Anonymize}(\texttt{Sign}(x_i, m), L, y_L, m) = \sigma] = \Pr[\texttt{Anonymize}(\texttt{Sign}(x_j, m), L, y_L, m) = \sigma]$$

for any security parameter k, any message m, any ring L, any valid ring signature σ with L, any $i, j \in L$, and any keys x_i, x_j.

Unforgeability

We consider the following experiment $Exp^{\text{unforge}}_{k,\Sigma}(\mathcal{A})$, where adversary $\mathcal{A}$ try to forge a valid ring signature without knowing the corresponding signing key or signature:

1. select random bit $b \xleftarrow{\$} \{0,1\}$, and generate secret and public keys $(x_i, y_i) \xleftarrow{\$} \texttt{KeyGen}(1^k)$ for $i \in L^0 = \{1, ..., n\}$,
2. at the beginning of the experiment, adversary $\mathcal{A}$ is given all public keys $y_1, ..., y_n$,
3. adversary $\mathcal{A}$ can access signing oracle $\mathcal{O}_s(i, m)$, anonymization oracle $\mathcal{O}_a(\mu, L)$ for $L \subset L^0 \cup L^{kr}$, and signature exposure oracle $\mathcal{O}_e(\mu)$ adaptively during the experiment,

4. adversary $\mathcal{A}$ can access secret key exposure oracle $\mathcal{O}_{sk}(i)$ adaptively during the experiment, and we denote the set of indices of all secret keys exposed by adversary $\mathcal{A}$ during the experiment by $L^{sk} \subset L^0$,
5. adversary $\mathcal{A}$ can access key registration oracle $\mathcal{O}_{kr}(y)$ adaptively during the experiment, and we denote the set of indices of all public keys registered by adversary $\mathcal{A}$ during the experiment by $L^{kr} = \{n+1, n+2, ...\}$,
6. finally, adversary $\mathcal{A}$ outputs $(m^*, L^*, \sigma^*) \stackrel{\$}{\leftarrow} \mathcal{A}^{\mathcal{O}_s, \mathcal{O}_a, \mathcal{O}_e, \mathcal{O}_{sk}, \mathcal{O}_{kr}}(y_1, ..., y_n)$,
7. return 1 if adversary $\mathcal{A}$ wins, 0 otherwise.

Here, we say that adversary $\mathcal{A}$ wins, if adversary $\mathcal{A}$ outputs forged signature (m^*, L^*, σ^*) and the following conditions hold:

1. $\texttt{Verify}(L^*, y_{L^*}, m^*, \sigma^*) = 1$,
2. $L^* \subset L^0 - L^{sk}$,
3. $\forall i \in L^*$, $(\star, i, m^*, \star) \notin Q_e$, i.e., adversary $\mathcal{A}$ never ask queries $\mathcal{O}_s(i, m^*) = \mu$ and $\mathcal{O}_e(\mu)$ for all $i \in L^*$, and
4. $(\star, \star, m^*, \star, L^*, \sigma^*) \notin Q_a$, i.e., adversary $\mathcal{A}$ never ask queries $\mathcal{O}_s(i, m^*) = \mu$ and $\mathcal{O}_a(\mu, L^*) = \sigma^*$.

We define the advantage Adv^{unforge} of the adversary $\mathcal{A}$ as

$$Adv^{\text{unforge}}_{k,\Sigma}(\mathcal{A}) = \Pr\left[Exp^{\text{unforge}}_{k,\Sigma}(\mathcal{A}) = 1\right],$$

where probability is taken over a random bit b, keys, random tapes of the oracles, random tapes of the adversary $\mathcal{A}$, and random tapes of `KeyGen`, `Verify`.

Definition 2 (unforgeability). *Anonymizable signature Σ satisfies* unforgeability *if $Adv^{unforge}_{k,\Sigma}(\mathcal{A}_k)$ is negligible in k for any probabilistic poly-time adversary $\mathcal{A}_k$.*

3 Proposed Scheme

In this section, we provide a very simple construction methodology of the anonymizable signature from any signature scheme, then according to our methodology we give a concrete scheme based on the BLS signature [BLS01].

3.1 Construction Methodology

We can construct an anonymizable signature scheme Σ from any signature scheme as follows.

(1) At the signing phase, the signer i generates a signature r of a message m by using the normal signature scheme that the anonymizable signature is based on. The signer passes the signature r to the proxy agent. The signature r must be treated as a secret between the signer and the agent.

(2) At the anonymizing phase, the agent makes a *"non-interactive proof of knowledge"* [FFS87, GMR85] σ_i which proves that he or she knows a valid signature r of the signer i on the message m.
(3) To anonymize the proof σ_i, the agent chooses an appropriate ring and simulates the proof of knowledge with respect to other members in the ring. By use of a *"proof of partial knowledge (or-proof)"* [CDS94], a ring signature σ can be composed of the proof of knowledge and it's simulations.

3.2 Concrete Scheme

We constructed an anonymizable signature scheme based on the BLS signature [BLS01] by which we can easily construct an efficient non-interactive proof of knowledge.

Let G and G_T be cyclic groups of prime order p, G^* be the set of the generators of G, $e : G \times G \to G_T$ be a non-degenerate bilinear map, $g \in G^*$ be a generator of G, and $H : \{0,1\}^* \to G^*$ and $H' : \{0,1\}^* \to \mathbb{Z}_p$ be mutually independent random oracles. We call the common reference string $\rho = (p, G, G_T, e, g, H, H')$ a system parameter. Let k be the security parameter. We assume that the system parameter can be determined in the polynomial time of k, ρ is encoded in the polynomial size of k, and all participants in our scheme use the same ρ.

Key Generation. Algorithm $\texttt{KeyGen}(1^k)$ takes a security parameter k, randomly chooses $x_i \in_U \mathbb{Z}_p$, and outputs secret and public keys $(sk_i = x_i, pk_i = y_i = g^{x_i})$ for signer $i \in N$.

Signing. Algorithm $\texttt{Sign}(x_i, m)$ takes i's secret key $sk_i = x_i$ and a message m, computes $h = H(\rho, m) \in G^*$, and outputs a signature $r = h^{x_i}$.

Anonymization. Algorithm $\texttt{Anonymize}(r, L, y_L, m)$ takes i's signature $r = h^{x_i}$, a ring $L \subset N$ s.t. $i \in L$, public keys y_L, and a message m, and outputs a ring signature σ as follows.

$h \leftarrow H(\rho, m) \in G^*$;
If $\exists i$ s.t. $e(g, r) = e(y_i, h), i \in L$, then perform the following steps, otherwise return $\perp$.
Generate a proof of the knowledge r s.t. $(\exists j \in L, r = h^{x_j})$ as the followings.
  $t \xleftarrow{\$} \mathbb{Z}_p$;
  $\tilde{a}_i \leftarrow e(g, h)^t \in G_T$;
  $\forall j \in L \setminus \{i\}$,
    $c_j \xleftarrow{\$} \mathbb{Z}_p,\ z_j \xleftarrow{\$} G$,
    $\tilde{a}_j \leftarrow e(g, z_j)e(h, y_j)^{c_j} \in G_T$;
  $c_i \leftarrow H'(\rho, L, m, y_L, \tilde{a}_L) - \sum_{j \neq i} c_j$;
  $z_i \leftarrow h^t r^{-c_i} \in G$;
Return $\sigma \leftarrow (c_L, z_L)$.

Verification. $\texttt{Verify}(L, y_L, m, \sigma)$ takes a ring $L \subset N$, public keys y_L, a message m, and a ring signature σ, and outputs a bit 0/1 as follows.

$(c_L, z_L) \leftarrow \sigma$;
$h \leftarrow H(\rho, m) \in G^*$;
$\forall j \in L,\ \tilde{a}_j \leftarrow e(g, z_j)e(h, y_j)^{c_j} \in G_T$;
return $\begin{cases} 1\ , & \text{if } H'(\rho, L, m, y_L, \tilde{a}_L) = \sum_{j \in L} c_j \\ 0\ , & \text{otherwise.} \end{cases}$

4 Security of the Proposed Scheme

In this section, we show that the proposed scheme satisfies computational unforgeability and perfect anonymity. We first refer to the GDH assumption [OP01] and the BLS signature [BLS01] that is used in the security proof of computational unforgeability.

4.1 Preliminary

Gap Diffie-Hellman (GDH) Assumption [OP01]

We refer to the GDH assumption where the adversary computes a CDH answer by use of a DDH oracle.

Let G_k be a cyclic group (family) with a security parameter k. We will omit the suffix k to simplify the description. For any $g_0, g_1, g_2, g_3 \in G$, we define a DDH oracle $\mathcal{O}_{\mathrm{ddh}}(g_0, g_1, g_2, g_3)$ as

$\mathcal{O}_{\mathrm{ddh}}(g_0, g_1, g_2, g_3) :=$
return $\begin{cases} 1\ , & \text{if } \log_{g_0} g_1 = \log_{g_2} g_3 \\ 0\ , & \text{otherwise.} \end{cases}$

For any algorithm $\mathcal{C}$, we define the GDH experiment Exp^{gdh} as

$Exp^{\mathrm{gdh}}_{k,G}(\mathcal{C}) :=$
$g \xleftarrow{\$} G,\ a \xleftarrow{\$} \mathbb{Z}_p,\ b \xleftarrow{\$} \mathbb{Z}_p$;
$h \xleftarrow{\$} \mathcal{C}^{\mathcal{O}_{\mathrm{ddh}}}(g, g^a, g^b)$;
return $\begin{cases} 1\ , & \text{if } h = g^{ab} \\ 0\ , & \text{otherwise.} \end{cases}$

We also define the advantage of $\mathcal{C}$, Adv^{gdh}, as

$$Adv^{\mathrm{gdh}}_{k,G}(\mathcal{C}) := \Pr\left[Exp^{\mathrm{gdh}}_{k,G}(\mathcal{C}) = 1\right],$$

where the sample space of the probability is the random tape of Exp^{gdh}.

Assumption 1 (the GDH assumption over G). *We say that the GDH assumption holds in G if, for any p.p.t. $\mathcal{C}_k$, $Adv^{gdh}_{k,G}(\mathcal{C}_k)$ is negligible in k.*

Hereafter, we assume that the above GDH assumption holds in pairing group G with pairing $e : G \times G \to G_T$.

BLS Signature [BLS01]

We refer to the BLS signature to which we reduce the computational unforgeability of the proposed scheme.

BLS signature $\Sigma_{\mathtt{b}} = (\mathtt{KeyGen}_{\Sigma_{\mathtt{b}}}, \mathtt{Sign}_{\Sigma_{\mathtt{b}}}, \mathtt{Verify}_{\Sigma_{\mathtt{b}}})$ is defined by the following algorithms.

Key Generation. Algorithm $\mathtt{KeyGen}_{\Sigma_{\mathtt{b}}}(1^k)$ takes 1^k as input, selects random $x \xleftarrow{\$} \mathbb{Z}_p$, computes $y \leftarrow g^x \in G$, and outputs secret and public keys (x, y)

Signing. Algorithm $\mathtt{Sign}_{\Sigma_{\mathtt{b}}}(x, m)$ takes secret key x and message m as input, computes $h \leftarrow H(\rho, m) \in G^*$ and $r \leftarrow h^x \in G$, where ρ is a system parameter, outputs signature r.

Verification. Algorithm $\mathtt{Verify}_{\Sigma_{\mathtt{b}}}(y, m, r)$ takes public key y, message m, and signature r as input, computes $h \leftarrow H(\rho, m) \in G$, where ρ is a system parameter, outputs 1 if $e(h, y) = e(g, r)$, 0 otherwise.

We define BLS signature oracle $\mathcal{O}_{\Sigma_{\mathtt{b}}}(m)$ as

$\mathcal{O}_{\Sigma_{\mathtt{b}}}(m) :=$
$\quad r \xleftarrow{\$} \mathtt{Sign}_{\Sigma_{\mathtt{b}}}(x, m)$;
$\quad Q_b \xleftarrow{\cup} (m, r)$;
$\quad$ return r ;

where x is a secret key of the signer. For any algorithm $\mathcal{B}$, we define a Turing machine $Exp^{\mathrm{bls}}_{k,\Sigma_{\mathtt{b}}}(\mathcal{B})$ as

$Exp^{\mathrm{bls}}_{k,\Sigma_{\mathtt{b}}}(\mathcal{B}) :=$
$\quad$ clear Q_b ;
$\quad (x, y) \xleftarrow{\$} \mathtt{KeyGen}_{\Sigma_{\mathtt{b}}}(1^k)$;
$\quad (m^*, r^*) \xleftarrow{\$} \mathcal{B}^{\mathcal{O}_{\Sigma_{\mathtt{b}}}, H}(y)$;
$\quad$ return $\begin{cases} 1\,, & \text{if } (m^*, r^*) \notin Q_b \ \wedge\ \mathtt{Verify}_{\Sigma_{\mathtt{b}}}(y, m^*, r^*) = 1 \\ 0\,, & \text{otherwise.} \end{cases}$

and the advantage of $\mathcal{B}$, Adv^{bls}, as

$$Adv^{\mathrm{bls}}_{k,\Sigma_{\mathtt{b}}}(\mathcal{B}) := \Pr\left[Exp^{\mathrm{bls}}_{k,\Sigma_{\mathtt{b}}}(\mathcal{B}) = 1\right]$$

where the sample space of the probability is the random tape of Exp^{bls}.

Lemma 1 ([BLS01]). *Let H be a random oracle. We define ϵ' and ϵ'' as*

$$\epsilon' := Adv^{bls}_{k,\Sigma_b}(\mathcal{B}),\ \epsilon'' := Adv^{gdh}_{k,G}(\mathcal{C}^{\mathcal{B}}).$$

Let q_e be the maximum number of queries to $\mathcal{O}_{\Sigma_b}$. For any real number $\xi \in [0,1]$, there exists a p.p.t. $\mathcal{C}$ that satisfies

$$(1-\xi)^{q_e}\xi\epsilon' \leq \epsilon''.$$

Corollary 1. *There exists a p.p.t. $\mathcal{C}$ that satisfies*

$$\epsilon' \leq 4(q_e+1)\epsilon''.$$

4.2 Unforgeability

The proposed scheme satisfies computational unforgeability. To show the unforgeability, we reduce the unforgeability of the proposed scheme to the unforgeability of the BLS signature. In a simulation, by the rewinding technique, we can obtain a forgery of the BLS signature.

Theorem 1. *The proposed scheme satisfies computational unforgeability under the GDH assumption in the random oracle model.*

Proof. In this proof, we construct a forger $\mathcal{B}$ against the BLS signature by use of a forger $\mathcal{A}$ against our scheme. The simulator executes the following steps.

$\mathcal{B}^{\mathcal{A},\mathcal{O}_{\Sigma_b},H}(y) :=$
clear ν, Q_s, Q_a, Q_e ;
$j \xleftarrow{\$} L^0$;
$\forall i \in L^0$, if $i = j$ then $u \xleftarrow{\$} \mathbb{Z}_p,\ y_i \leftarrow yg^u$;
else $(x_i, y_i) \xleftarrow{\$} \mathtt{KeyGen}(1^k)$;
$(L^*, m^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{S}_s,\mathcal{S}_a,\mathcal{S}_e,\mathcal{S}_{sk},\mathcal{S}_{kr}}(y_{L^0})$;
if $\mathcal{A}$ doesn't win the game then abort.
rewind $\mathcal{A}$ to the corresponding H'.
$\mathcal{A}$ outputs (L'^*, m'^*, σ'^*).
if $\mathcal{A}$ doesn't win the second game then abort.
if $(L^*, m^*) \neq (L'^*, m'^*) \ \vee\ \sigma^* = \sigma'^*$ then abort.
$(c_{L^*}, z_{L^*}) \leftarrow \sigma^*,\ \ (c'_{L^*}, z'_{L^*}) \leftarrow \sigma'^*$;
find $i \in L^*$ s.t. $c_i \neq c'_i$;
if $i \neq j$ then abort.
$r^* \leftarrow (z'_i/z_i)^{\frac{1}{c_i - c'_i}} (H(\rho, m^*))^{-u}$;
return (m^*, r^*).

The simulator executes the following signing oracle simulation $\mathcal{S}_s$.

$\mathcal{S}_s(i, m) :=$
$\mu \leftarrow \nu\mathtt{++}$;

if $i \in L^0$ and m is in valid domain then
  $r \xleftarrow{\$} \star$;
else $r \leftarrow \perp$;
$Q_s \xleftarrow{\cup} \{(\mu, i, m, r)\}$;
return μ ;

The simulator executes the following anonymization oracle simulation $\mathcal{S}_a$.

$\mathcal{S}_a(\mu, L) :=$
  if μ is registered in Q_s and $L \subset L^0 \cup L^{kr}$ then
    find $(i, m, \star)$ s.t. $(\mu, i, m, \star) \in Q_s$;
    $\sigma \xleftarrow{\$} \mathit{Simulate}(\mathtt{Anonymize}(\star, L, y_L, m))$;
  else $\sigma \leftarrow \perp$;
  $Q_a \xleftarrow{\cup} \{(\mu, L, \sigma)\}$;
  return σ ;

where $\mathit{Simulate}(\mathtt{Anonymize}(\star, L, y_L, m))$ is the following algorithm that produces a ring signature without a signature by manipulating the value of H'.

$\mathit{Simulate}(\mathtt{Anonymize}(\star, L, y_L, m)) :=$
  $h \leftarrow H(\rho, m)$;
  $\forall j \in L,\ c_j \xleftarrow{\$} \mathbb{Z}_p,\ z_j \xleftarrow{\$} G,$
    $\tilde{a}_j \leftarrow e(g, z_j)e(h, y_j)^{c_j} \in G_T$;
  if $H'(\rho, L, m, y_L, \tilde{a}_L)$ is defined then abort.
  else $H'(\rho, L, m, y_L, \tilde{a}_L) \leftarrow \sum_j c_j$;
  return (c_L, z_L) ;

Let q be the maximum number of queries to H' and α be the success probability that all $\mathit{Simulate}(\mathtt{Anonymize}(\star, L, y_L, m))$ does not abort through the execution of $\mathcal{B}^{\mathcal{A}}$. α is evaluated as the following, where $q \geq 1$, $p^n \gg q$, and $n = \#L^0$.

$$\alpha > \prod_{i=0}^{q-1} \left(1 - \frac{i}{p^n}\right) > \left(1 - \frac{q}{p^n}\right)^q > 1 - \frac{q^2}{p^n}$$

The simulator executes the following signature exposure oracle simulation $\mathcal{S}_e$.

$\mathcal{S}_e(\mu) :=$
  if μ is registered in Q_s then
    find $(i, m, \star)$ s.t. $(\mu, i, m, \star) \in Q_s$;
    if $i = j$ then $r \xleftarrow{\$} (\mathcal{O}_{\Sigma_{\mathfrak{b}}}(m))(H(\rho, m))^u$;
    else $r \xleftarrow{\$} \mathtt{Sign}(x_i, m)$;
  else $r \leftarrow \perp$;
  $Q_e \xleftarrow{\cup} \{(\mu, r)\}$;
  return r ;

The simulator executes the following secret key exposure oracle simulation $\mathcal{S}_{sk}$.

$\mathcal{S}_{sk}(i) :=$
  if $i \in L^0$ then
    if $i = j$ then abort.
    $x \leftarrow x_i$;
    $L^{sk} \overset{\cup}{\leftarrow} \{i\}$;
  else $x \leftarrow \perp$;
  return x ;

The simulator executes the key registration oracle simulation $\mathcal{S}_{kr}$ which is identical to the key registration oracle $\mathcal{O}_{kr}$ defined in section 2.3.

We define ϵ, ϵ', and ϵ'' as

$$\epsilon := Adv^{\mathrm{unforge}}_{k,\Sigma}(\mathcal{A}),\ \epsilon' := Adv^{\mathrm{bls}}_{k,\Sigma_{\mathrm{b}}}(\mathcal{B}^{\mathcal{A}}),\ \epsilon'' := Adv^{\mathrm{gdh}}_{k,G}(\mathcal{C}^{\mathcal{B}^{\mathcal{A}}}).$$

Let q_e be the maximum number of queries to $\mathcal{S}_e$, q be the maximum number of queries to H' and assume $q \geq 1$. Let $n = \#L^0$. According to the forking lemma [BN06],

$$\alpha\epsilon(\alpha\epsilon/q - 1/p)/n \leq \epsilon'.$$

Thus,

$$\begin{aligned}\epsilon &\leq (q/2p + \sqrt{(q/2p)^2 + qn\epsilon'})/\alpha \\ &\leq (q/2p + \sqrt{(q/2p)^2 + 4qn(q_e+1)\epsilon''})/\alpha \\ &\leq (q/2p + \sqrt{(q/2p)^2 + 4qn(q_e+1)\epsilon''})/(1 - q^2/p^n).\end{aligned}$$

We assume that the maximum number of queries to the random oracles, namely q and q_e, and the size of the ring n are bounded by some polynomial of security parameter k. Let $\mathcal{A}_k$ be a polynomial time algorithm of k that attacks the unforgeability of our scheme with success probability ϵ. Immediately we have a polynomial time algorithm $\mathcal{C}^{\mathcal{B}^{\mathcal{A}_k}}$ that wins the GDH game with success probability ϵ'', which satisfies the above inequality. Thus, if the GDH assumption over G holds, for any p.p.t. $\mathcal{A}_k$, $\epsilon'' = Adv^{\mathrm{gdh}}_{k,G}(\mathcal{C}^{\mathcal{B}^{\mathcal{A}_k}})$ is negligible. Therefore, for any p.p.t. $\mathcal{A}_k$, $\epsilon = Adv^{\mathrm{unforge}}_{k,\Sigma}(\mathcal{A}_k)$ is negligible by the above inequality. □

4.3 Perfect Anonymity

The proposed scheme satisfies perfect anonymity. To show the perfect anonymity, we prove that for any valid ring signature σ, for any member i in the ring L, there exists a unique way to produce the ring signature σ by use of i's secret key x_i.

Theorem 2. *The proposed scheme satisfies the perfect anonymity in the random oracle model.*

Proof. For any $\sigma^* = (c^*_L, z^*_L)$ s.t. $\mathtt{Verify}(L, y_L, m, \sigma^*) = 1$, for any $i \in L$, there exists a unique assignment of the random tape t, c_j, z_j in the `Anonymize` function that satisfies $\sigma^* = \mathtt{Anonymize}(\mathtt{Sign}(x_i, m), L, y_L, m)$.

$$t = x_i c^*_i + \log_{H(\rho,m)} z^*_i,\ \text{and } c_j = c^*_j,\ z_j = z^*_j,\ \forall j \neq i.$$ □

5 Conclusion

We presented a novel concept of a ring signature called anonymizable signature, by which one can convert a signature into an anonymous ring signature without any secret but the signature itself. By using an anonymizable signature, a signer can leave a signed message to a proxy agent who will convert the signature into a ring signature afterward. If the agent is not trustworthy, the anonymizable signature is not suitable for an application that needs highly strong anonymity. However, if the agent can be regarded as an ideal functionality, it has many applications. We also provided the definition of anonymizable signature, a simple construction methodology, and a concrete scheme that can be proven to be unconditionally anonymous and computationally unforgeable under the GDH assumption in the random oracle model.

References

[BKM06] Bender, A., Katz, J., Morselli, R.: Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 60–79. Springer, Heidelberg (2006)

[BLS01] Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)

[BN06] Bellare, M., Neven, G.: Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma. In: Proc. of the 13th ACM Conference on Computer and Communications Security (CCS), pp. 390–399 (2006)

[BNN04] Bellare, M., Namprempre, C., Neven, G.: Security Proofs for Identity-Based Identification and Signature Schemes. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 268–286. Springer, Heidelberg (2004)

[CDS94] Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994)

[FFS87] Feige, U., Fiat, A., Shamir, A.: Zero Knowledge Proofs of Identity. In: Proc. of STOC 1987, pp. 210–217 (1987)

[GMR85] Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-systems. In: Proc. of STOC 1985, pp. 291–304 (1985)

[LWH05] Lee, K.-C., Wen, H.-A., Hwang, T.: Convertible ring signature. IEE Proc. of Communications 152(4), 411–414 (2005)

[OP01] Okamoto, T., Pointcheval, D.: The Gap Problems: A New Class of Problems for the Security of Cryptographic Primitives. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)

[RST01] Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001)

[SBZ01] Steinfeld, R., Bull, L., Zheng, Y.: Content Extraction Signatures. In: Kim, K. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 285–304. Springer, Heidelberg (2001)

[Sha84] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1984)

[SHK09] Suzuki, K., Hoshino, F., Kobayashi, T.: Relinkable Ring Signature. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 518–536. Springer, Heidelberg (2009)

[SK03] Sakai, R., Kasahara, M.: ID based Cryptosystems with Pairing on Elliptic Curve. Cryptology ePrint Archive: 2003/054 (2003)