Relinkable Ring Signature

Koutarou Suzuki[1], Fumitaka Hoshino[2], and Tetsutaro Kobayashi[1]

[1] NTT Information Sharing Platform Laboratories, NTT Corporation,
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
suzuki.koutarou@lab.ntt.co.jp, kobayashi.tetsutaro@lab.ntt.co.jp
[2] IPA : Information-Technology Promotion Agency, Japan
2-28-8, Hon-Komagome, Bunkyo-ku, Tokyo, 113-6591 Japan
f-hoshi@ipa.go.jp

Abstract. In this paper, we propose the concept of a relinkable ring signature, which is a ring signature with ring reformation function, i.e., a signer can delegate ring reformation ability separately from signing ability to his/her proxy. The relinkable ring signature can be applicable to proxy ring reformation, anonymization of past-generated signature, or ring signature for dynamic group. We also propose a concrete relinkable ring signature scheme that uses pairing in the random oracle model.

Keywords: ring signature, anonymity, pairing.

1 Introduction

Ring signature, where a signer can sign anonymously on behalf of a group, the ring members, without a setup procedure or group manager, was introduced in [RST01]. The signer generates a ring signature for a message using his/her secret key and the public keys of all the ring members. Thus, by the moment of ring signature generation, the ring members need to be determined and their public keys need to be provided.

In this paper, we propose the concept of a *relinkable ring signature*: an extension of a ring signature where ring members do not need to be determined by the moment of ring signature generation and the ring members of the generated ring signature can be changed at a later point. Compared with the usual ring signature scheme, a relinkable ring signature scheme has a *relink algorithm* that can change the ring members of a given signature by using the relink key rk after the signature is generated using the signing key sk. The relink key rk is weaker than the signing key sk and can change the ring members but cannot change the message and the real signer, i.e., using relink key rk, one can create a new ring signature for the same message with different ring members that include the same real signer from the existing ring signature, but one cannot create a new ring signature for a new message and a new real signer. Thus, using the relinkable ring signature, one can separately select a message and ring members.

The relinkable ring signature provides restricted anonymity in comparison with the usual ring signature, i.e., it guarantees only computational anonymity,

J.A. Garay, A. Miyaji, and A. Otsuka (Eds.): CANS 2009, LNCS 5888, pp. 518–536, 2009.

while the usual ring signature guarantees unconditional anonymity. Moreover, using the relink key rk, one can check whether or not the real signer of a ring signature is the signer corresponding to the the relink key rk by changing the ring members to a set that consists of only the corresponding signer. Thus, the relinkable ring signature is not suitable for applications that highly require anonymity, e.g., voting or whistle-blowing. However, the restricted anonymity of the relinkable ring signature can be utilized for the following applications.

Proxy Ring Formation: By providing relink key rk to the signer's proxy, the signer can delegate the ring reformation ability to the proxy separately from the signing ability. This is useful for a signer with small computational resources. For instance, one can securely store the signing key sk in a tamper-resistant IC card that has only small computational resources and store the relink key rk in a PC that has large computational resources and access to PKI. The IC card computes a ring signature with ring members including only the signer by using the signing key sk, then the PC reforms the ring members of the ring signature by using the relink key rk and public keys of the other ring members from PKI. By this, one can isolate a ring formation process whose computational cost is heavy, i.e., is proportional to the number of ring members, and delegate it to a PC with large computational resources.

Anonymization of Past-generated Signature: In the case that one publicizes a document with a signature, he/she needs to ensure the privacy of the signer, e.g., in the case of publication of a governmental document by a "freedom of information act". To hide the content of the document, one can use a sanitizing signature [SBZ01]. To hide the signer, one can use the relinkable ring signature, i.e., one can anonymize a past-generated signature by using relink key rk. The signer submits the document, the relinkable ring signature on it with ring members including only the signer, and relink key rk. When the document is publicized, one can anonymize the signature by reforming the ring members using relink key rk.

Ring Signature for Dynamic Group: In a ring signature scheme [RST01], a signer can sign anonymously on behalf of the ring members and there is no group manager. However, this would not be suitable for a dynamic group whose members change, since after the public key of a member is removed, one can no longer verify stored past-generated ring signatures whose ring members include the removed member. The proposed relinkable ring signature can resolve this problem as follows. When a new member joins the group, the new member registers his/her public key to the PKI of the group and passes his/her relink key to the relink manager. When a member leaves the group, the PKI removes the public key of that member and the relink manager removes that member from the ring members of stored past-generated ring signatures by using that member's relink key.

Private Key Exposure Attack: The usual ring signature scheme that has unconditional anonymity is susceptible to a private key exposure attack, i.e., once a signer makes his/her private key public, all ring signatures whose ring members include that signer become meaningless because anyone can use the publicized private key to generate the signature. The relinkable ring signature is not susceptible to a private key exposure attack because it has computational anonymity and one can exclude the signer who exposed his/her private key from the ring members of a signature by using relink key rk.

Convertible Ring Signature: One can gradually decrease the anonymity of past-generated ring signatures by decreasing the number of ring members of the signature by using relink key rk. In an extreme case, one can convert a past-generated ring signature to non-anonymous signature by making the ring members include only the signer. This is similar to a convertible ring signature [LWH05], though one has a conversion key for each signature in the convertible ring signature.

Ring Signature and Group Signature: In a group signature scheme [CvH91], there is a group manager who can revoke the anonymity of signatures. In contrast, in a ring signature scheme [RST01], there is no group manager. As an intermediary between the ring and group signatures, the revocable ring signature [LLM+07] was invented. The relinkable ring signature can also be considered an intermediary between the ring and group signatures, where a signer can generate a signature without a setup procedure and the signer's proxy who has relink key rk can revoke anonymity.

The proposed relinkable ring signature scheme is secure in the random oracle model, and uses groups with efficiently computable pairing on a non-supersingular elliptic curve known as an MNT curve [MNT01], where there is no efficiently computable distortion map. More precisely, we use three assumptions in groups G_1, G_2, and G_3, where there is efficiently computable pairing $e : G_1 \times G_2 \to G_3$ but no efficiently computable distortion map $\psi : G_1 \to G_2$. These groups are studied by [SHUK03, GPS08] and used in some existing schemes [Sco02, ACdM05, BGdMM05, ACHdM05] where the XDH (eXternal Diffie-Hellman) assumption, i.e., the DDH problem in the group G_1 is intractable, is used.

In Section 2, we describe bilinear groups on a non-supersingular elliptic curve and three assumptions used in the proposed scheme. In Section 3, we define the relinkable ring signature. In Section 4, we show the proposed relinkable ring signature scheme using pairing, prove its security, and estimate its efficiency. In Section 5, we conclude the paper.

2 Bilinear Group and Assumptions

In this section, we describe groups with efficiently computable pairing but without an efficiently computable distortion map, we also describe three assumptions on these groups used in the proposed relinkable ring signature scheme.

We use the pairing on a non-supersingular elliptic curve known as an MNT curve [MNT01], where no distortion map is known. By using an MNT curve, we can construct cyclic groups G_1, G_2, and G_3 of prime order p, which are called a bilinear groups, and a polynomial-time computable bilinear non-degenerate map called pairing

$$e : G_1 \times G_2 \rightarrow G_3.$$

Let $g \in G_1$ be a generator of G_1 and $\hat{g} \in G_2$ be a generator of G_2. See Appendix B for a detailed construction.

On MNT curves [MNT01], no polynomial-time computable homomorphism $\psi : G_1 \rightarrow G_2$, called a distortion map, is known. See Appendix B for details.

These bilinear groups are studied by [SHUK03, GPS08], and used in some existing schemes [Sco02, ACdM05, BGdMM05, ACHdM05] where the XDH (eXternal Diffie-Hellman) assumption, i.e., the DDH problem in the group G_1 is intractable, is used.

We now state the three assumptions on these groups that are used in the proposed relinkable ring signature scheme as follows.

For adversary A, we define advantage

$$\mathrm{Adv}^{\mathrm{skewCDH}}(A) = \Pr[g \in_U G_1, \hat{g} \in_U G_2, \alpha \in \mathbb{Z}_p, A(g, g^{\alpha}, \hat{g}) = \hat{g}^{\alpha}],$$

where the probability is taken over the choices of $g, \hat{g}, \alpha$ and the coin tosses of A.

Definition 1 (Skew CDH Assumption from G_1 to G_2). *We assume that for all polynomial-time adversary A, advantage* $\mathrm{Adv}^{\mathrm{skewCDH}}(A)$ *is negligible in security parameter k.*

For adversary A, we define advantage

$$\mathrm{Adv}^{\mathrm{hintedCDH}}(A) = \Pr[g \in_U G_1, \hat{g} \in_U G_2, \alpha, \beta \in \mathbb{Z}_p, A(g, g^{\alpha}, g^{\beta}, \hat{g}, \hat{g}^{\alpha}, \hat{g}^{\beta}) = g^{\alpha\beta}],$$

where the probability is taken over the choices of $g, \hat{g}, \alpha, \beta$ and the coin tosses of A.

Definition 2 (Hinted CDH Assumption in G_1). *We assume that for all polynomial-time adversary A, advantage* $\mathrm{Adv}^{\mathrm{hintedCDH}}(A)$ *is negligible in security parameter k.*

We denote by $D_1 = \{(g, h, g', h') \in G_1^4 | \log_g h = \log_{g'} h'\}$ the set of DDH tuple, and by $D_0 = \{(g, h, g', h') \in G_1^4\}$ the set of random tuple. For adversary A, we define advantage

$$\mathrm{Adv}^{\mathrm{DDH}}(A) = |\Pr[b \in_U \{0, 1\}, X \in_U D_b : A(X) = b] - 1/2|$$

where the probability is taken over the choices of b, X and the coin tosses of A.

Definition 3 (DDH Assumption in G_1). *We assume that for all polynomial-time adversary A, advantage* $\mathrm{Adv}^{\mathrm{DDH}}(A)$ *is negligible in security parameter k.*

Notice that if there exists polynomial-time computable distortion map $\psi : G_1 \rightarrow G_2$, "Skew CDH Assumption from G_1 to G_2" and "DDH Assumption in G_1" are not true and "Hinted CDH Assumption in G_1" is equivalent to "CDH Assumption in G_1".

3 Relinkable Ring Signature

In this section, we provide the definition of relinkable ring signature. *Anonymity* means, informally, that adversary cannot distinguish test signature is generated by signer 0 or by signer 1, where the adversary knows all secret and relink keys except of signer 0 and 1. *Traceability* means, informally, that the real signer of signature generated by adversary can be determined uniquely, where the adversary knows all secret and relink keys. *Unforgeability* means, informally, that adversary cannot create new forged signature and cannot modify a signature from signing oracle, where the adversary does not know secret and relink keys. *Relinker unforgeability* means, informally, that adversary cannot create new forged signature and cannot modify message and real signer of a signature from signing oracle, where the adversary knows relink keys. Our definition of anonymity does not adopt *adversarially-chosen keys* and *full key exposure* [BKM06], since exposure of revoke keys trivially breaks anonymity. Our definitions of unforgeability and relinker unforgeability adopt *insider corruption* [BKM06].

3.1 Definition of Relinkable Ring Signature

We provide the definition of the relinkable ring signature scheme. In this scheme there are two secret keys: signing key sk by which signer can generate a ring signature, and relink key rk by which relinker can reform the ring member of generated ring signature.

We denote the set of signers $N = \{0, 1, ...\}$. We also denote subset of signers $L \subset N$ that is called ring.

Syntax. A relinkable ring signature scheme is a tuple of four algorithms $\Sigma = (\mathrm{Gen}, \mathrm{Sig}, \mathrm{Ver}, \mathrm{Rel})$, s.t.

- Gen, the key generation algorithm, is a probabilistic polynomial-time algorithm that takes security parameter $k \in \mathbb{N}$, and outputs secret, relink, and public key (sk, rk, pk):

  $$\mathrm{Gen}(k) \to (sk, rk, pk).$$

  We denote by (sk_i, rk_i, pk_i) the public, secret, and relink key of the i-th signer.
- Sig, the signing algorithm, is a probabilistic polynomial-time algorithm that takes secret key sk_i, ring $L \subset N$ s.t. $i \in L$, set of public keys of L, and message $m \in \{0, 1\}^*$, and outputs signature σ:

  $$\mathrm{Sig}(sk_i, L, (pk_j)_{j \in L}, m) \to \sigma.$$
- Ver, the signature verification algorithm, is a probabilistic polynomial-time algorithm that takes ring $L \subset N$, set of public keys of L, message $m \in \{0, 1\}^*$, and signature σ, and outputs a bit 0/1 that means reject/accept, respectively:

  $$\mathrm{Ver}(L, (pk_j)_{j \in L}, m, \sigma) \to 0/1.$$

- Rel, the relinking algorithm, is a probabilistic polynomial-time algorithm that takes relink key rk_i, rings $L, L' \subset N$ s.t. $i \in L, L'$, sets of public keys of $L \cup L'$, message $m \in \{0,1\}^*$, and signature σ, and outputs new signature σ':
$$\mathrm{Rel}(rk_i, L, L', (pk_j)_{j \in L \cup L'}, m, \sigma) \rightarrow \sigma'/''reject''.$$

A relinkable ring signature scheme satisfies the following correctness.

Correctness. For every $i \in N$, every $L_1, ..., L_J \subset N$ s.t. $i \in L_1, ..., L_J$, and every $m \in \{0,1\}^*$, if $\mathrm{Gen}(k) \rightarrow (sk_i, rk_i, pk_i)$, $\mathrm{Sig}(sk_i, L_1, (pk_l)_{l \in L_1}, m) \rightarrow \sigma_1$, $\mathrm{Rel}(rk_i, L_j, L_{j+1}, (pk_l)_{l \in L_j \cup L_{j+1}}, m, \sigma_j) \rightarrow \sigma_{j+1}$ for $j = 1, ..., J-1$, it holds with overwhelming probability that $\mathrm{Ver}(L_j, (pk_l)_{l \in L_j}, m, \sigma_j) = 1$ for $j = 1, ..., J$.

We first define the following three oracles called by adversary in the games of security definitions. We then define the following four security notions of relinkable ring signature: anonymity, unforgeability, relinker unforgeability, and traceability.

Key Registration Oracle $KO(i, rk_i, pk_i)$. Let $N = \{0, 1, ..\}$ be set of registered signers. Let $RK = \{rk_1, rk_2, ...\}$ and $PK = \{pk_1, pk_2, ...\}$ be set of registered relink keys and public keys. Adversary generates secret, relink, and public keys $(sk_i, rk_i, pk_i) \leftarrow Gen(k)$[1] and send index of key i, relink keyrk_i, and public key pk_i to key registration oracle KO. Key registration oracle KO registers the keys sent from the adversary, i.e., appends i to N, rk_i to RK, and pk_i to PK.

Signing Oracle $SO(i, L, m)$. Adversary sends index of signing key i, ring $L \subset N$, and message m to signing oracle SO. Signing oracle SO generates signature $\mathrm{Sig}(sk_i, L, (pk_j)_{j \in L}, m) \rightarrow \sigma$ and return signature σ.

Relink Oracle $RO(L, L', m, \sigma)$. Adversary sends ring $L, L' \subset N$, message m, and signature σ to relink oracle RO. If $\mathrm{Ver}(L, (pk_j)_{j \in L}, m, \sigma) \neq 1$, return "reject". Relink oracle RO finds i^* s.t. $\mathrm{Ver}(\{i^*\}, pk_i, m, \mathrm{Rel}(rk_i, L, \{i^*\}, (pk_j)_{j \in L}, m, \sigma)) = 1$. Here, such i^* is unique because of Traceability. If $i^* \in L, L'$ does not hold, return "reject". Relink oracle RO generates relinked signature $\mathrm{Rel}(rk_{i^*}, L, L', (pk_j)_{j \in L \cup L'}, m, \sigma) \rightarrow \sigma'$ and return relinked signature σ'.

Anonymity. We define the anonymity of a relinkable ring signature scheme Σ. We consider the following game of adversary D against Σ.

At the beginning of the game, simulator chooses a random bit $b \in \{0,1\}$, and generates secret, relink, and public keys $(sk_i, rk_i, pk_i) \leftarrow Gen(k)$ $(i = 0, 1)$ and registers (i, rk_i, pk_i) $(i = 0, 1)$. D takes pk_0, pk_1 as input, and performs the following steps.[2]

[1] To guarantee correct key generation, PKI may require zero-knowledge proof of correctness of keys, when user registers his/her keys.

[2] We assume that D tries to distinguish signature generated by 0 or 1 w.l.o.g.

D may make queries to key registration oracle KO, signing oracle SO, and relink oracle RO. D is allowed to execute these oracle calls polynomially many times at any moment.

D sends $L^* \subset N$ s.t. $\{0,1\} \subset L^*$ and m^* to the challenge oracle CO, and can obtain signature $\sigma^* \leftarrow \mathrm{Sig}(sk_b, (pk_j)_{j \in L^*}, m^*)$. D is allowed to execute this once at any moment.

Finally, D outputs a bit b'.

D cannot ask to relink oracle RO, if

- σ is σ^* or its (polynomially many times) relinked signatures,
- and $L' \cap \{0,1\} = \{0\}$ or $\{1\}$.

When the game is defined in the random oracle model, D may access the random oracle polynomially many times at any moment.

We define the advantage of D against Σ as

$$\mathrm{Adv}_{\Sigma}^{\mathrm{anon}}(D) = \left| \Pr\left[\begin{array}{l} b \in \{0,1\}, (sk_i, rk_i, pk_i) \leftarrow \mathrm{Gen}(k)\ (i = 0,1), \\ b' \leftarrow D^{KO,SO,RO,CO}(pk_0, pk_1) \end{array} : b = b' \right] - \frac{1}{2} \right|$$

where the probability is taken over the choice of bit b, keys (sk_i, rk_i, pk_i) $(i = 0,1)$, and the coin tosses of KO, SO, RO, CO and D.

Definition 4. *We say that relinkable ring signature scheme Σ is anonymous, if for every probabilistic polynomial-time adversary D the advantage $\mathrm{Adv}_{\Sigma}^{\mathrm{anon}}(D)$ is negligible in k.*

Traceability. We define the traceability of a relinkable ring signature scheme Σ. We consider the following game of adversary F against Σ.

F performs the following steps.

F may make queries to key registration oracle KO. F is allowed to execute these oracle calls polynomially many times at any moment.

Finally, F outputs (L^*, m^*, σ^*).

We say F wins the game if

- $\mathrm{Ver}(L^*, (pk_j)_{j \in L^*}, m^*, \sigma^*) = 1$,
- $\#\{i : i \in L^*, \mathrm{Ver}(\{i\}, pk_i, m^*, \mathrm{Rel}(rk_i, L^*, \{i\}, (pk_j)_{j \in L^*}, m, \sigma^*)) = 1\} \neq 1$.

When the game is defined in the random oracle model, F may access the random oracle polynomially many times at any moment.

We define the advantage of F against Σ as

$$\mathrm{Adv}_{\Sigma}^{\mathrm{trace}}(F) = \Pr\left[(L^*, m^*, \sigma^*) \leftarrow F^{KO}() : F \text{ wins.} \right]$$

where the probability is taken over the choice of the coin tosses of KO and F.

Definition 5. *We say that relinkable ring signature scheme Σ is traceable, if for every probabilistic polynomial-time adversary F the advantage $\mathrm{Adv}_{\Sigma}^{\mathrm{trace}}(F)$ is negligible in k.*

If a scheme is traceable, for given (L^*, m^*, σ^*) generated by probabilistic polynomial-time adversary F, there exists unique signer $i \in L^*$ s.t. $\mathrm{Ver}(\{i\}, pk_i, m^*, \mathrm{Rel}(rk_i, L^*, \{i\}, (pk_j)_{j \in L^*}, m, \sigma^*)) = 1$ except negligible probability. We denote the unique signer by $i^* \in L^*$.

Unforgeability. We define the unforgeability of a relinkable ring signature scheme Σ. We consider the following game of adversary F against Σ.

At the beginning of the game, simulator generates secret, relink, and public keys $(sk_i, rk_i, pk_i) \leftarrow Gen(k)$ $(i = 0, ..., n-1)$ and registers (i, rk_i, pk_i) $(i = 0, ..., n-1)$. F takes $(pk_j)_{j=0,\ldots,n-1}$ as input, and performs the following steps.

F may make queries to key registration oracle KO, signing oracle SO, and relink oracle RO. F is allowed to execute these oracle calls polynomially many times at any moment.

Finally, F outputs (L^*, m^*, σ^*).

We say F wins the game if

- $\mathrm{Ver}(L^*, (pk_j)_{j\in L^*}, m^*, \sigma^*) = 1$,
- $L^* \subset \{0, ..., n-1\}$,
- $((i^*, L^*, m^*), \sigma^*)$ never appears in oracle query and reply list of SO,
- and $((i^*, L, L^*, m^*, \sigma), \sigma^*)$ never appears in oracle query and reply list of RO for any L and σ.

When the game is defined in the random oracle model, F may access the random oracle polynomially many times at any moment.

We define the advantage of F against Σ as

$$\mathrm{Adv}_{\Sigma}^{\mathrm{unforge}}(F) = \Pr\left[\begin{matrix}(sk_i, rk_i, pk_i) \leftarrow \mathrm{Gen}(k)\ (i = 0, ..., n-1), \\ (L^*, m^*, \sigma^*) \leftarrow F^{KO,SO,RO}((pk_j)_{j=0,\ldots,n-1})\end{matrix} : F \text{ wins.}\right]$$

where the probability is taken over the choice of keys (sk_i, rk_i, pk_i) $(i = 0, ..., n-1)$ and the coin tosses of KO, SO, RO and F.

Definition 6. *We say that relinkable ring signature scheme Σ is unforgeable, if for every probabilistic polynomial-time adversary F the advantage* $\mathrm{Adv}_{\Sigma}^{\mathrm{unforge}}(F)$ *is negligible in k.*

Relinker Unforgeability. We define the relinker unforgeability of a relinkable ring signature scheme Σ. We consider the following game of adversary F against Σ.

At the beginning of the game, simulator generates secret, relink, and public keys $(sk_i, rk_i, pk_i) \leftarrow Gen(k)$ $(i = 0, ..., n-1)$ and registers (i, rk_i, pk_i) $(i = 0, ..., n-1)$. F takes $(rk_i, pk_i)_{i=0,\ldots,n-1}$ as input, and performs the following steps.

F may make queries to key registration oracle KO, signing oracle SO. F is allowed to execute these oracle calls polynomially many times at any moment.

Finally, F outputs (L^*, m^*, σ^*).

We say F wins the game if

- $\mathrm{Ver}(L^*, (pk_j)_{j\in L^*}, m^*, \sigma^*) = 1$,
- $L^* \subset \{0, ..., n-1\}$,
- and $((i^*, L, m^*), \sigma)$ never appears in oracle query and reply list of SO for any L and σ.

When the game is defined in the random oracle model, F may access the random oracle polynomially many times at any moment.

We define the advantage of F against Σ as

$$\mathrm{Adv}_{\Sigma}^{\mathrm{relink}}(F) = \Pr\left[\begin{array}{l}(sk_i, rk_i, pk_i) \leftarrow \mathrm{Gen}(k)\ (i = 0, ..., n-1), \\ (L^*, m^*, \sigma^*) \leftarrow F^{KO,SO}((rk_i, pk_i)_{i=0,...,n-1})\end{array} : F \text{ wins.}\right]$$

where the probability is taken over the choice of keys (sk_i, rk_i, pk_i) $(i = 0, ..., n-1)$ and the coin tosses of KO, SO and F.

Definition 7. *We say that relinkable ring signature scheme Σ is relinker unforgeable, if for every probabilistic polynomial-time adversary F the advantage $\mathrm{Adv}_{\Sigma}^{\mathrm{relink}}(F)$ is negligible in k.*

4 Proposed Relinkable Ring Signature Scheme

In this section, we propose a relinkable ring signature scheme using pairing, prove its security, and estimate its efficiency.

4.1 Intuition of the Proposed Scheme

The following scheme is the interactive protocol which our ring signature is based on.

1. Prover P and verifier V have common input $g, y = g^x, h, w \in G_1$ and $\hat{g} \in G_2$. Prover P has witness $\hat{y} = \hat{g}^x \in G_2$.
2. Prover P chooses random $r \in_U \mathbb{Z}_p$ and sends $a = e(g^r, \hat{g}) \in G_3$ and $b = e(h^r, \hat{g}) \in G_3$ to verifier V.
3. Verifier V chooses random $c \in_U \mathbb{Z}_p$ and sends it to prover P.
4. Prover P sends $\hat{z} = \hat{g}^r \hat{y}^{-c} \in G_2$ to verifier V.
5. Verifier V checks $a = e(g, \hat{z})e(y, \hat{g})^c$ and $b = e(h, \hat{z})e(w, \hat{g})^c$.

This protocol is a variant of well known Chaum-Pedersen's protocol [CP92]. By using the Chaum-Pedersen's protocol, the prover who knows x which satisfies $y = g^x \wedge w = h^x$ can give an interactive proof of knowledge about x, and he is able to prove that $(g, y, h, w) \in$ DDH-tuple. Instead of the discrete logarithm x, we employ the corresponding group element of G_2 as the knowledge to prove, that is the prover who knows $\hat{y} \in G_2$ such that $e(y, \hat{g}) = e(g, \hat{y}) \wedge e(w, \hat{g}) = e(h, \hat{y})$ can give an interactive proof of knowledge about $\hat{y}$. Therefore we can separate the ability to prove that $(g, y, h, w) \in$ DDH-tuple from the discrete logarithm x, namely we can use $\hat{y}$ as a relink key, and can use x as a signing key. Furthermore this protocol inherits honest verifier zero-knowledgeness, language soundness, and knowledge soundness which are closely related to the security of our ring signature scheme. Finally we combine this protocol with the Cramer-Damgård-Schoenmakers' standard technique (proof of partial knowledge) [CDS94] to construct our ring signature scheme.

4.2 Proposed Relinkable Ring Signature Scheme

The proposed relinkable ring signature scheme is as follows.

Let G_1, G_2, G_3 be a multiplicative cyclic group with prime order p. Let $g \in G_1$ and $\hat{g} \in G_2$ be generators of G_1 and G_2. Let $e : G_1 \times G_2 \rightarrow G_3$ be pairing. Let $k \in \mathbb{N}$ be a security parameter that is the bit length of group element. Let $H : \{0,1\}^* \rightarrow G_1$ and $H' : \{0,1\}^* \rightarrow \mathbb{Z}_p$ be distinct hash functions that are modeled as random oracles in the security statements below. We denote by $N = \{0, ..., n-1\}$ the set of n signers.

Key Generation. Gen takes security parameter k, randomly chooses $x_i \in_U \mathbb{Z}_p$, and outputs secret, relink, public keys $(sk_i = x_i, rk_i = \hat{y}_i = \hat{g}^{x_i}, pk_i = y_i = g^{x_i})$ for i-th signer.

Signing. Sig takes i-th secret key $sk_i = x_i$, ring $L \subset N$ s.t. $i \in L$, public keys $(pk_i = y_i)_{i \in L}$, and message m, and outputs signature σ as follows.

1. Choose random $r \in_U \{0,1\}^l$, compute $h = H(r, m) \in G_1$, $w = h^{x_i} \in G_1$.
2. Generate a (non-interactive) zero-knowledge proof for language $\{(g, (y_i)_{i \in L}, h, w) \mid \exists i \in L, \log_g y_i = \log_h w\}$ as follows.
   (a) For i, choose random $r_i \in_U \mathbb{Z}_p$, compute $a_i = e(g^{r_i}, \hat{g}), b_i = e(h^{r_i}, \hat{g}) \in G_3$.
   (b) For all $j \in L \setminus \{i\}$, choose random $c_j \in_U \mathbb{Z}_p$, $\hat{z}_j \in_U G_2$, compute $a_j = e(g, \hat{z}_j)e(y_j, \hat{g})^{c_j}, b_j = e(h, \hat{z}_j)e(w, \hat{g})^{c_j} \in G_3$.
   (c) Compute $c_i = H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L}) - \sum_{j \neq i} c_j \mod p$.
   (d) Compute $\hat{z}_i = \hat{g}^{r_i} \hat{y}_i^{-c_i} \in G_2$.
3. Output signature $\sigma = (r, w, (c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$.

Verification. Ver takes ring $L \subset N$, public keys $(pk_i = y_i)_{i \in L}$, message m, and signature $\sigma = (r, w, (c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$, and outputs bit 0/1 as follows.

1. Check $y_i, w \in G_1$, $c_i \in \mathbb{Z}_p$, $\hat{z}_i \in G_2$ for all $i \in L$, otherwise reject.
2. Compute $h = H(r, m)$.
3. Compute $a_i = e(g, \hat{z}_i)e(y_i, \hat{g})^{c_i}, b_i = e(h, \hat{z}_i)e(w, \hat{g})^{c_i} \in G_3$ for all $i \in L$.
4. Check that $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L}) = \sum_{i \in L} c_i \mod p$, otherwise reject.
5. Output accept if all checks above are passed, otherwise output reject.

Relinking. Rel takes i-th relink key $rk_i = \hat{y}_i$, rings $L, L' \subset N$ s.t. $i \in L, L'$, public keys $(pk_i = y_i)_{i \in L \cup L'}$, message m, and signature $\sigma = (r, w, (c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$, and outputs signature σ' as follows.

1. Verify signature σ, otherwise reject.
2. Compute $h = H(r, m)$.
3. Check $e(h, \hat{y}_i) = e(w, \hat{g})$, otherwise reject.

Table 1. The comparison of costs of the proposed relinkable ring signature scheme and ring signature scheme [AOS02]. Here, n is the number of group member, T_{exp} is the time to compute exponential in G, T_{pair} is the time to compute pairing, $L_G, L_{\mathbb{Z}_p}$ are the lengths of elements of $G, \mathbb{Z}_p$, respectively.

	proposed scheme	ring signature [AOS02]
Signing costs	$2nT_{exp} + 4nT_{pair}$	nT_{exp}
Verification costs	$2nT_{exp} + 4nT_{pair}$	$2nT_{exp}$
Relinking costs	$2nT_{exp} + 4nT_{pair}$	$-$
Signature size	$(n+1)(L_G + L_{\mathbb{Z}_p})$	$2nL_{\mathbb{Z}_p}$

4. Generate a (non-interactive) zero-knowledge proof for language $\{(g, (y_i)_{i \in L'}, h, w) \mid \exists i \in L', \log_g y_i = \log_h w\}$ as follows.
   (a) For i, choose random $r_i \in_U \mathbb{Z}_p$, compute $a_i = e(g^{r_i}, \hat{g}), b_i = e(h^{r_i}, \hat{g}) \in G_3$.
   (b) For all $j \in L' \setminus \{i\}$, choose random $c'_j \in_U \mathbb{Z}_p$, $\hat{z}'_j \in_U G_2$, compute $a_j = e(g, \hat{z}'_j)e(y_j, \hat{g})^{c'_j}, b_j = e(h, \hat{z}'_j)e(w, \hat{g})^{c'_j} \in G_3$.
   (c) Compute $c'_i = H'(L', h, w, (a_i)_{i \in L'}, (b_i)_{i \in L'}) - \sum_{j \neq i} c'_j \mod p$.
   (d) Compute $\hat{z}'_i = \hat{g}^{r_i} \hat{y}_i^{-c'_i} \in G_2$.
5. Output signature $\sigma' = (r, w, (c'_i)_{i \in L'}, (\hat{z}'_i)_{i \in L'})$.

4.3 Security

The proposed relinkable ring signature scheme satisfies anonymity, unforgeability, relinker unforgeability, and traceability.

Theorem 1. *The proposed scheme satisfies*

1. *anonymity if we assume DDH problem in G_1 is intractable and H and H' are random oracle,*
2. *unforgeability if we assume skew CDH problem from G_1 to G_2 is intractable and H and H' are random oracle,*
3. *relinker unforgeability if we assume hinted CDH problem in G_1 is intractable and H and H' are random oracle,*
4. *and traceability if we assume H and H' are random oracle.*

The proofs of theorem is provided in Appendix C.

4.4 Efficiency

The comparison of costs of the proposed relinkable ring signature scheme and existing discrete logarithm based ring signature scheme [AOS02] is provided in Table1. Although the proposed relinkable ring signature scheme has complexity of same order in n as existing ring signature scheme [AOS02], the proposed scheme needs more pairing operations and modular exponentiations.

5 Conclusion

In this paper, we proposed the concept of the relinkable ring signature, which is a ring signature with ring reformation, i.e., a signer can delegate ring reformation ability separately from signing ability. The security of relinkable ring signature is defined by anonymity, unforgeability, relinker unforgeability, and traceability. We also proposed a concrete relinkable ring signature scheme that uses pairing.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments.

References

[ACdM05] Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable RFID tags via insubvertible encryption. In: Atluri, V., Meadows, C., Juels, A. (eds.) ACM Conference on Computer and Communications Security, pp. 92–101. ACM, New York (2005)

[ACHdM05] Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. Cryptology ePrint Archive: 2005/385 (2005)

[AOS02] Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n Signatures from a Variety of Keys. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 415–432. Springer, Heidelberg (2002)

[BB04] Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)

[BGdMM05] Ballard, L., Green, M., de Medeiros, B., Monrose, F.: Correlation-resistant storage. Technical Report TR-SP-BGMM-050705, Johns Hopkins University, CS Dept, 2005 (2005)

[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)

[BLS02] Barreto, P.S.L.M., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 257–267. Springer, Heidelberg (2003)

[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)

[CP92] Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)

[CvH91] Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)

[DEM05] Dupont, R., Enge, A., Morain, F.: Building curves with arbitrary small mov degree over finite prime fields. J. Cryptology 18(2), 79–89 (2005)

[GPS08] Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)

[JN01] Joux, A., Nguyen, K.: Separating decision diffie-hellman from diffie-hellman in cryptographic groups. Cryptology ePrint Archive: 2001/003 (2001)

[LLM+07] Liu, D.Y.W., Liu, J.K., Mu, Y., Susilo, W., Wong, D.S.: Revocable ring signature. J. Comput. Sci. Technol. 22(6), 785–794 (2007)

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

[MNT01] Miyaji, A., Nakabayashi, M., Takano, S.: New explicit conditions of elliptic curve traces for fr-reduction. IEICE Transactions on Fundamentals E84-A(5), 1234–1243 (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)

[SB06] Scott, M., Barreto, P.S.L.M.: Generating more mnt elliptic curves. Des. Codes Cryptography 38(2), 209–217 (2006)

[SBZ01] Steinfeld, R., Bull, L., Zheng, Y.: Content extraction signatures. In: Kim, K. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 285–304. Springer, Heidelberg (2002)

[Sco02] Scott, M.: Authenticated id-based key exchange and remote log-in with simple token and pin number. Cryptology ePrint Archive: 2002/164 (2002)

[Sho97] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)

[SHUK03] Saito, T., Hoshino, F., Uchiyama, S., Kobayashi, T.: Candidate one-way functions on non-supersingular elliptic curves. Technical Report of IEICE, ISEC 2003-65 (2003)

[Ver01] Verheul, E.R.: Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 195–210. Springer, Heidelberg (2001)

A Security of Skew and Hinted CDH Assumptions

Security in Generic Group Model: First, we prove security of the skew CDH problem and the hinted CDH problem in generic groups in the sense of [Sho97]. Our proof is essentially the same as the proof of CDH problem in [Sho97] except the complicated group settings of the pairing. We employ the settings of [BB04] which gives a generic proof of the q-SDH assumption on the type 2 curve in [GPS08].

In the generic group model, elements of $G_1 = \langle g \rangle$, $G_2 = \langle \hat{g} \rangle$, and G_3 are encoded as unique random strings. Let $\xi_i : G_i \rightarrow \{0,1\}^*$ be a random encoding function of G_i for $i = 1, 2, 3$. The adversary $\mathcal{A}$ can make the following oracle calls

- the group operation in each of G_1, G_2 and G_3,
- the bilinear pairing $e : G_1 \times G_2 \to G_3$, and
- the projection $\phi : G_2 \to G_1$.

These oracles takes encoding(s) of the input element(s) and answers encoding of the output element. Notice that we consider projection just for generality.

We have the following propositions about security of skew and hinted CDH assumptions in the generic group model. Due to lack of space, we omit here the proofs of these propositions that only follow [Sho97, BB04].

Proposition 1. *Let $\mathcal{A}$ be an algorithm that solves the skew CDH problem in the generic group model, making a total of at most q queries to the oracles computing the group action in G_1, G_2, G_3, the oracle computing the projection ϕ, and the oracle computing the bilinear pairing e. If $a \in \mathbb{Z}_p$ and ξ_1, ξ_2, ξ_3 are chosen at random, then the probability ϵ that $\mathcal{A}(\xi_1(g), \xi_1(g^a), \xi_2(\hat{g}))$ outputs $\xi_2(\hat{g}^a)$, is bounded by $\epsilon \leq O(q^2/p)$*

Proposition 2. *Let $\mathcal{A}$ be an algorithm that solves the hinted CDH problem in the generic group model, making a total of at most q queries to the oracles computing the group action in G_1, G_2, G_3, the oracle computing the projection ϕ, and the oracle computing the bilinear pairing e. If $a, b \in \mathbb{Z}_p$ and ξ_1, ξ_2, ξ_3 are chosen at random, then the probability ϵ that $\mathcal{A}(\xi_1(g), \xi_1(g^a), \xi_1(g^b), \xi_2(\hat{g}), \xi_2(\hat{g}^a), \xi_2(\hat{g}^b))$ outputs $\xi_1(g^{ab})$, is bounded by $\epsilon \leq O(q^2/p)$*

Reduction Security: Second, we show reductions of the skew CDH problem and the hinted CDH problem to other known problems, assuming that there exists projection $\phi : G_2 \to G_1$, i.e., the type 2 curve in [GPS08].

Proposition 3. *If there exists probabilistic polynomial-time algorithm $A_s : G_1^2 \times G_2 \to G_2$ that solves the skew CDH problem on (G_1, G_2), there exists probabilistic polynomial-time algorithm $A_c : G_1^3 \to G_1$ that solves a variant of the CDH problem on G_1 (named chosen generator CDH).*

Proof. Let $g = \phi(\hat{g})$. $A_c(g, g^a, g^b)$ outputs $\phi(A_s(g, g^b, A_s(g, g^a, \hat{g})))$. □

Proposition 4. *If there exists probabilistic polynomial-time algorithm $A_h : G_1^3 \times G_2^3 \to G_1$ that solves the hinted CDH problem on (G_1, G_2), there exists probabilistic polynomial-time algorithm $A_b : G_2^4 \to G_3$ that solves the BDH problem on G_2.*

Proof. $A_b(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^c)$ outputs $e(A_h(\phi(\hat{g}), \phi(\hat{g}^a), \phi(\hat{g}^b), \hat{g}, \hat{g}^a, \hat{g}^b), \hat{g}^c)$. □

B Selection of Bilinear Group

Selection of G_1 and G_2: Let p be a prime and set $q = p^n$. Let $E[\ell]$ be ℓ-torsion points on elliptic curve E defined over $\mathbb{F}_q$. Let $\phi(x, y) = (x^q, y^q)$ the q-th power

Frobenius morphism. Let $e : E[\ell] \times E[\ell] \rightarrow \mu_\ell$ be Weil pairing, where $\mu_\ell \subset \overline{\mathbb{F}}_q$ is the group of the ℓ-th root of 1.

We consider the eigenspace decomposition of $E[\ell]$ w.r.t. ϕ. For simplicity, we assume that ℓ is a prime, $\#E(\mathbb{F}_q)[\ell] = \ell$, m is minimal integer s.t. $\#E(\mathbb{F}_{q^m})[\ell] = \ell^2$. Since ϕ is the identity map on $E(\mathbb{F}_q)$, $E(\mathbb{F}_q)[\ell]$ is an eigenspace of ϕ with eigenvalue $\lambda_1 = 1$. Let λ_2 be another eigenvalue, and we have $\lambda_2 = \lambda_1\lambda_2 = q \mod \ell$. We also assume that the eigenvalues of ϕ are non-degenerative, i.e., $t = \lambda_1 + \lambda_2 \neq 2 \mod \ell$.

Then, there exists the eigenspace corresponding to λ_2 in $E[\ell] \setminus E(\mathbb{F}_q)[\ell]$. Let Q be its generator, and P be a generator of $E(\mathbb{F}_q)[\ell]$. Thus, we have $E[\ell] = \langle P \rangle \oplus \langle Q \rangle$, and we use $G_1 = \langle P \rangle \subset E(\mathbb{F}_q)[\ell], G_2 = \langle Q \rangle \subset E(\mathbb{F}_{q^m})[\ell], G_3 = \mu_\ell \subset \mathbb{F}_{q^m}^*$, and Weil pairing $e : \langle P \rangle \times \langle Q \rangle \rightarrow \mu_\ell$. We can use also $G_2 = \langle R \rangle$ where $R = \alpha P + \beta Q$, $\alpha, \beta \in \mathbb{Z}/\ell\mathbb{Z}$, and $\beta \neq 0$.

The (normalized) trace map $tr = 1/m \sum_{i=0,\ldots,m-1} \phi^i : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q$ induces polynomial-time computable group isomorphism $tr : \langle R \rangle \rightarrow \langle P \rangle$.

The generator Q can be found by $Q = R - tr(R)$ from R. The generator R can be found with probability $(\ell - 1)/\ell$ if a point on $E[\ell]$ is chosen at random. A point on $E[\ell]$ can be found if a point on $E(\mathbb{F}_{q^m})$ is chosen at random, and multiplied by $\#E(\mathbb{F}_{q^m})/\ell^2$. A point on $E(\mathbb{F}_{q^m})$ can be found with probability of approximately $1/2$ if $x \in \mathbb{F}_{q^m}$ is chosen at random and $y \in \mathbb{F}_{q^m}$ is obtained by solving the curve equation.

The generator P of $E(\mathbb{F}_q)[\ell]$ can be found if a point on $E(\mathbb{F}_q)$ is chosen at random, and multiplied by $\#E(\mathbb{F}_q)/\ell$. A point on $E(\mathbb{F}_q)$ can be found with probability of approximately $1/2$ if $x \in \mathbb{F}_q$ is chosen at random and $y \in \mathbb{F}_q$ is obtained by solving the curve equation.

Selection of Elliptic Curve: To guarantee that our assumptions hold, we need to choose an elliptic curve that satisfies the following conditions.

On the the supersingular curve and the trace-2 curve, polynomial-time computable homomorphism from G_1 to G_2, called the distortion map, is known [JN01, Ver01]. Therefore, we should avoid the supersingular curve and the trace-2 curve.

On the other hand, it was shown that the distortion map on a non-supersingular non trace-2 curve cannot be described by any single rational map [Ver01]. Therefore, we choose the non-supersingular non trace-2 curve.

As in the case of a pairing-based cryptosystem, to guarantee that the CDH or DL problem is intractable, we need to choose an elliptic curve that satisfies the following two conditions.

- $\#E(\mathbb{F}_q)[\ell] = \ell$ is large enough s.t. the CDH or DL problem on $E(\mathbb{F}_q)[\ell]$ is intractable.
- $\#\mathbb{F}_{q^m}^*$ is large enough s.t. the CDH or DL problem on $\mathbb{F}_{q^m}^*$ is intractable.

We can choose in practice $\ell \sim q \geq 2^{160}$ and $q^m \geq 2^{1024}$.

Adding to the conditions above, we need to choose an elliptic curve with small q and m s.t. elliptic addition and pairing can be computed efficiently, i.e., polynomial-time computable.

Efficient methods to find non-supersingular non trace-2 pairing enabled secure curves are given in [MNT01, SB06, BLS02, DEM05].

C Proofs of Theorem 1

Anonymity: Let Σ be the proposed revocable ring signature scheme. Let D be a $(\tau, \epsilon, q_{SO}, q_H)$-adversary against Σ that requests signing oracle at most q_{SO} times and accesses random oracles at most q_H times in total and breaks the anonymity of Σ with advantage at least ϵ and running time at most τ. Let D' be a (τ', ϵ')-adversary against DDH assumption that breaks the assumption with advantage at least ϵ' and running time at most τ'. We construct adversary D' from adversary D as follows. Simulator maintains random oracle call lists H, H' and list Log, and performs followings.

At the beginning of the simulation, simulator D' is given instance of DDH problem $(g, g^\alpha, g^\beta, g^\gamma)$. Simulator selects random bit $b \in \{0,1\}$ and random $v \in \mathbb{Z}_p$, sets $pk_b = y_b = (g^\alpha)^v$, generates secret, relink, and public keys $(sk_{1-b}, rk_{1-b}, pk_{1-b}) \leftarrow Gen(k)$, and gives pk_0, pk_1 to adversary D.

Random Oracle $H(r,m)$. If $H(r,m)$ is already defined, return defined value. Otherwise, select random $u \in_U \mathbb{Z}_p$, define $Log(g,h) = u$, and define $H(r,m) = h = g^u$ and return it.

Random Oracle $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L})$. If $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L})$ is already defined, return defined value. Otherwise, select random $c \in_U \mathbb{Z}_p$, and define $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L}) = c$ and return it.

Key Registration Oracle $KO(i, \hat{y}, y)$. If y_i is already defined, reject. If $e(g, \hat{y}) \neq e(y, \hat{g})$, reject. Otherwise, define $\hat{y}_i = \hat{y}$ and $y_i = y$.

Signing Oracle $SO(i, L, m)$. Select random $r \in_U \{0,1\}^k$. Call random oracle $h = H(r,m)$, if $Log(g,h)$ is not defined, abort,otherwise set $u = Log(g,h)$ and $w = y_j^u$. Create simulated proof $((c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$ by setting $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L}) = \sum_{i \in L} c_i$. Return $(r, w, (c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$.

Relink Oracle $RO(L, L', m, \sigma)$. Verify signature $\sigma = (r, w, (c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$, otherwise reject. Call random oracle $h = H(r,m)$. Create simulated proof $((c'_i)_{i \in L'}, (\hat{z}'_i)_{i \in L'})$ by setting $H'(L, h, w, (a'_i)_{i \in L'}, (b'_i)_{i \in L'}) = \sum_{i \in L'} c'_i$. Return $(r, w, (c'_i)_{i \in L'}, (\hat{z}'_i)_{i \in L'})$.

Challenge Oracle $CO(L^*, m^*)$. Select random $r \in_U \{0,1\}^k$. If $H(r,m)$ is already defined, abort,otherwise select random $u \in_U \mathbb{Z}_p$, define $H(r,m) = h = (g^\beta)^u$ ($Log(g,h)$ can not be defined), and set $w = (g^\gamma)^{uv}$. Create simulated proof $((c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$ by setting $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L}) = \sum_{i \in L} c_i$. Return $(r, w, (c_i)_{i \in L}, (\hat{z}_i)_{i \in L})$.

Finally, D outputs bit $b' \in \{0,1\}$. Simulator D' outputs random bit if the simulation abort, bit 1 if $b = b'$, and random bit if $b \neq b'$.

The probability that the simulation does not abort is $Pr[\neg abort] \geq (1 - (q_H + q_S)/2^k)(1 - 1/2^k)^{q_S}$, since CO does not abort with probability at least $1 - (q_H + q_S)/2^k$ and SO does not abort with probability at least $(1 - 1/2^k)^{q_S}$.

In the case that the simulation aborts, we have $Pr[D'\ wins|abort] = 1/2$. Since the view of adversary is independent from b if the instance is not DDH-tuple, i.e., $\gamma \neq \alpha\beta$, we have $Pr[D'\ wins|\gamma \neq \alpha\beta|\neg abort] = 1/2$. Since the simulation is perfect if the instance is DDH-tuple, i.e., $\gamma = \alpha\beta$, we have $Pr[D'\ wins|\gamma = \alpha\beta|\neg abort] = Pr[D\ wins|\gamma = \alpha\beta|\neg abort]$.

Thus, we have $\epsilon' = |Pr[D'\ wins] - 1/2| = |Pr[D'\ wins|\neg abort] - 1/2| \cdot Pr[\neg abort] = |Pr[D'\ wins|\gamma = \alpha\beta|\neg abort] - 1/2| \cdot Pr[\gamma = \alpha\beta|\neg abort] \cdot Pr[\neg abort] = |Pr[D\ wins|\gamma = \alpha\beta|\neg abort] - 1/2| \cdot Pr[\gamma = \alpha\beta|\neg abort] \cdot Pr[\neg abort] \geq \epsilon \cdot 1/2 \cdot (1 - (q_H + q_S)/2^k)(1 - 1/2^k)^{q_S} \geq \epsilon \cdot 1/2 \cdot (1 - (q_H + 2q_S)/2^k)$.

Traceability: Let Σ be the proposed revocable ring signature scheme. Let F be a $(\tau, \epsilon, q_{SO}, q_H)$-adversary against Σ that requests signing oracle at most q_{SO} times and accesses random oracles at most q_H times in total and breaks the traceability of Σ with advantage at least ϵ and running time at most τ. Simulator maintains random oracle call lists H, H' and performs followings.

Random Oracle $H(r,m)$. If $H(r,m)$ is already defined, return defined value. Otherwise, select random $u \in_U \mathbb{Z}_p$, and define $H(r,m) = h = g^u$ and return it.

Random Oracle $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L})$. If $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L})$ is already defined, return defined value. Otherwise, select random $c \in_U \mathbb{Z}_p$, and define $H'(L, h, w, (a_i)_{i \in L}, (b_i)_{i \in L}) = c$ and return it.

Key Registration Oracle $KO(i, \hat{y}, y)$. If y_i is already defined, reject. If $e(g, \hat{y}) \neq e(y, \hat{g})$, reject. Otherwise, define $\hat{y}_i = \hat{y}$ and $y_i = y$.

Finally, F outputs L^*, m^*, σ^*. we write $\sigma^* = (r^*, w^*, (c_i)_{i \in L^*}, (\hat{z}_i)_{i \in L^*})$ and $h^* = H(r^*, m^*)$. By the second winning condition of adversary, $\forall i \in L^*$, (g, y_i, h^*, w^*) is not DDH-tuple, i.e., $x_i = \log_g(y_i) \neq x'_i = \log_{h^*}(w^*)$. Thus, the first winning condition of adversary holds with negligible probability $\epsilon \leq 1 - (1 - 1/p)^{q_H} \leq q_H/p$, since language soundness of zero-knowledge proof.

Unforgeability: Let Σ be the proposed revocable ring signature scheme. Let F be a $(\tau, \epsilon, q_{SO}, q_H)$-adversary against Σ that requests signing oracle at most q_{SO} times and accesses random oracles at most q_H times in total and breaks the unforgeability of Σ with advantage at least ϵ and running time at most τ. Let F' be a (τ', ϵ')-adversary against skew CDH assumption that breaks the assumption with advantage at least ϵ' and running time at most τ'. We construct adversary F' from adversary F as follows. Simulator maintains random oracle call lists H, H' and list Log, and performs followings.

At the beginning of the simulation, simulator F' is given instance of skew CDH problem $(g, g^{\alpha}, \hat{g})$. Simulator selects random $v_i \in \mathbb{Z}_p$, sets $pk_i = y_i = (g^{\alpha})^{v_i}$ for $i = 0, ...n-1$, and gives them to adversary F.

Random Oracle $H(r, m)$*.* If $H(r, m)$ is already defined, return defined value. Otherwise, select random $u \in_U \mathbb{Z}_p$, define $Log(g, h) = u$, and define $H(r, m) = h = g^u$ and return it.

Random Oracle $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L})$*.* If $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L})$ is already defined, return defined value. Otherwise, select random $c \in_U \mathbb{Z}_p$, and define $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L}) = c$ and return it.

Key Registration Oracle $KO(i, \hat{y}, y)$*.* If y_i is already defined, reject. If $e(g, \hat{y}) \neq e(y, \hat{g})$, reject. Otherwise, define $\hat{y}_i = \hat{y}$ and $y_i = y$.

Signing Oracle $SO(i, L, m)$*.* Select random $r \in_U \{0, 1\}^k$. Call random oracle $h = H(r, m)$, set $u = Log(g, h)$ and $w = y_j^u$. Create simulated proof $((c_i)_{i\in L}, (\hat{z}_i)_{i\in L})$ by setting $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L}) = \sum_{i\in L} c_i$. Return $(r, w, (c_i)_{i\in L}, (\hat{z}_i)_{i\in L})$.

Relink Oracle $RO(L, L', m, \sigma)$*.* Verify signature $\sigma = (r, w, (c_i)_{i\in L}, (\hat{z}_i)_{i\in L})$, otherwise reject. Call random oracle $h = H(r, m)$. Create simulated proof $((c'_i)_{i\in L'}, (\hat{z}'_i)_{i\in L'})$ by setting $H'(L, h, w, (a'_i)_{i\in L'}, (b'_i)_{i\in L'}) = \sum_{i\in L'} c'_i$. Return $(r, w, (c'_i)_{i\in L'}, (\hat{z}'_i)_{i\in L'})$.

Finally, F outputs L^*, m^*, σ^*. we write $\sigma^* = (r^*, w^*, (c_i)_{i\in L^*}, (\hat{z}_i)_{i\in L^*})$ and $h^* = H(r^*, m^*)$. By rewinding adversary F, F outputs $L^*, m^*, \sigma^{*'}$ where $\sigma^{*'} = (r^*, w^*, (c'_i)_{i\in L^*}, (\hat{z}'_i)_{i\in L^*})$ s.t. $\sum_{i\in L} c_i \neq \sum_{i\in L} c'_i$ with probability $1/q_H$, since adversary F uses same H' query for forgery with probability $1/q_H$. Find i s.t. $c_i \neq c'_i$ and compute $\hat{y}_i = (\hat{z}_i/\hat{z}'_i)^{1/(c'_i - c_i)}$. Simulator F' outputs $(\hat{y}_i)^{1/v_i}$ as guess of $\hat{g}^{\alpha}$. Using forking lemma, we have $\epsilon' \geq \epsilon(\epsilon/q_H - 1/p)$.

Relinker Unforgeability: Let Σ be the proposed revocable ring signature scheme. Let F be a $(\tau, \epsilon, q_{SO}, q_H)$-adversary against Σ that requests signing oracle at most q_{SO} times and accesses random oracles at most q_H times in total and breaks the relinker unforgeability of Σ with advantage at least ϵ and running time at most τ. Let F' be a (τ', ϵ')-adversary against hinted CDH assumption that breaks the assumption with advantage at least ϵ' and running time at most τ'. We construct adversary F' from adversary F as follows. Simulator maintains random oracle call lists H, H' and list Log, and performs followings.

At the beginning of the simulation, simulator F' is given instance of hinted CDH problem $(g, g^{\alpha}, g^{\beta}, \hat{g}^{\alpha}, \hat{g}^{\beta})$.

Simulator selects random $v_i \in_U \mathbb{Z}_p$ and random $\gamma_i \in_U \{0, 1\}$. Let $\hat{g}_i = (\hat{g}^{\alpha})^{\gamma_i}(\hat{g}^{\beta})^{1-\gamma_i}$ and $g_i = (g^{\alpha})^{\gamma_i}(g^{\beta})^{1-\gamma_i}$. Simulator sets $rk_i = \hat{y}_i = \hat{g}_i^{v_i}$ and $pk_i = y_i = g_i^{v_i}$, and give them to adversary F.

Random Oracle $H(r, m)$*.* If $H(r, m)$ is already defined, return defined value. Otherwise, select random $u \in_U \mathbb{Z}_p, \gamma \in_U \{0, 1\}$ define $H(r, m) = h = ((g^{\alpha})^{\gamma}(g^{\beta})^{1-\gamma})^u$ and define $Log((g^{\alpha})^{\gamma}(g^{\beta})^{1-\gamma}, h) = u$, and return h.

Random Oracle $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L})$. If $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L})$ is already defined, return defined value. Otherwise, select random $c \in_U \mathbb{Z}_p$, and define $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L}) = c$ and return it.

Key Registration Oracle $KO(i, \hat{y}, y)$. If y_i is already defined, reject. If $e(g, \hat{y}) \neq e(y, \hat{g})$, reject. Otherwise, define $\hat{y}_i = \hat{y}$ and $y_i = y$.

Signing Oracle $SO(i, L, m)$. Select random $r \in_U \{0, 1\}^k$. If $H(r, m)$ is already defined, abort, otherwise select random $u \in_U \mathbb{Z}_p$, define $H(r, m) = h = g_i^u$, define $Log(g_i, h) = u$, and set $w = h^{v_i}$. Create simulated proof $((c_i)_{i\in L}, (\hat{z}_i)_{i\in L})$ by setting $H'(L, h, w, (a_i)_{i\in L}, (b_i)_{i\in L}) = \sum_{i\in L} c_i$. Return $(r, w, (c_i)_{i\in L}, (\hat{z}_i)_{i\in L})$.

Finally, F outputs L^*, m^*, σ^*. we write $\sigma^* = (r^*, w^*, (c_i)_{i\in L^*}, (\hat{z}_i)_{i\in L^*})$ and $h^* = H(r^*, m^*)$. Find j s.t. $e(h^*, \hat{y}_j) = e(w^*, \hat{g})$, and set $v = v_j$. If $Log(g^\alpha g^\beta / g_j, h^*)$ is not defined, abort, otherwise set $u = Log(g^\alpha g^\beta / g_j, h^*)$. Simulator F' outputs $(w^*)^{1/uv}$ as guess of $g^{\alpha\beta}$. Since the simulation successes if SO doesn't select random r that is already queried by adversary F and $Log(g^\alpha g^\beta / g_j, h^*)$ is defined at the final step, we have $\epsilon' \geq Pr[Log(g^\alpha g^\beta / g_j, h^*) \ is \ defined | F' \ finds \ j | F \ wins | \neg abort \ in \ SO] \cdot Pr[F' \ finds \ j | F \ wins | \neg abort \ in \ SO] \cdot Pr[F \ wins | \neg abort \ in \ SO] \cdot Pr[\neg abort \ in \ SO] \geq 1/2 \cdot (1 - 1/p)^{q_H} \cdot \epsilon \cdot (1 - (q_H + q_S)/2^k)^{q_S} \geq \epsilon/2 \cdot (1 - q_H/p - (q_H + q_S)q_S/2^k)$.