IIE 2019 Lecture 1

Linkable Ring Signature Revisited

Scribe: Tian Qiu March 11, 2019

1 Introduction

Ring Signature

A ring signature scheme consists of a tuple of efficient algorithms ($\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify}$) for generating a public parameter, generating keys for users, signing messages, and verifying ring signatures, respectively.

$\mathsf{RSetup}(n)$: Generates public parameters pp which are made available to all users.

$\mathsf{RKgen}(pp)$ Generates a public key pk and the corresponding secret key sk.

$\mathsf{RSign}_{pp}(sk, M, R)$: Outputs a signature Σ on the message $M \in \{0,1\}^*$with respect to the ring $R = (pk_0, ..., pk_{N-1})$. It is required that (pk, sk) be a valid key pair produced by $\mathsf{RKgen}(pp)$ and that $pk \in R$.

$\mathsf{RVerify}_{pp}(M, R, \Sigma)$: Given a candidate signature Σ on a message M with respect to the ring of public keys R, this algorithm outputs 1 if Σ is deemed valid or 0 otherwise.

Definition 1 (Correctness)**.** *A ring signature* ($\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify}$) *is correct if any ring signature generated by a honest user legally is valid.* $pp \leftarrow \mathsf{RSetup}$*, any* $(pk, sk) \leftarrow \mathsf{RKgen}(pp)$*, any* R *such that* $pk \in R$*, any* $M \in \{0,1\}^*$ *we have*

$$\mathsf{RVerify}_{pp}(M, R, \mathsf{RSign}_{pp}(sk, M, R)) = 1$$

.

Definition 2 (Anonymity)**.** *A ring signature scheme* ($\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify}$) *provides statistical anonymity if it is infeasible to determine which ring member issued a particular ring signature even given all the signing keys.*

Given any (possibly unbounded) adversary $\mathcal{A}$*, its advantage in the following experiment is negligible in n:*

1. *The challenger* $\mathcal{C}$ *runs the* $\mathsf{RSetup}(n)$ *algorithm and gives the public parameter pp to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *runs the* $\mathsf{RKgen}(pp)$ *algorithm and obtains all the public key and secret key pairs* $\{pk_i, sk_i\}_{i=0}^{N-1}$*.*
3. $\mathcal{A}$ *outputs a message* M*, the ring* $R = (pk_0, pk_1, \ldots, pk_{N-1})$ *and two identities* $j_0, j_1 \in [N-1]$*.*

4. *$\mathcal{C}$ randomly picks a bit $b \xleftarrow{\$} \{0,1\}$ and feeds $\mathcal{A}$ with the signature $\Sigma \leftarrow \mathsf{RSign}_{pp}(sk_{j_b}, M, R)$.*

5. *$\mathcal{A}$ outputs a bit b' and succeeds if $b' = b$.(denoted as* Succ*)*

The advantage of $\mathcal{A}$ is $\Pr[\mathsf{Succ}] - 1/2$.

A ring signature is unforgeable with respect to insider corruption if it is infeasible to forge a ring signature without controlling one of the ring members.

Definition 3 (Unforgeability)**.** *A ring signature scheme* (RSetup, RKgen, RSign, RVerify) *provides unforgeability if it is infeasible for any PPT adversary $\mathcal{A}$ to win the following experiment with non-negligible probability.*

1. *The challenger $\mathcal{C}$ runs the* RSetup(n) *algorithm and runs the* RKgen(pp) *algorithm and returns all the public keys $\{pk_i\}_{i=0}^{N-1}$ to $\mathcal{A}$, where ring $R = (pk_0, pk_1, \ldots, pk_{N-1})$.*

2. *$\mathcal{A}$ may query the following oracles adaptively and in any order:*

   - *A* Corrupt *oracle that on input $i \in [N-1]$ returns sk_i and records the corresponding public key pk_i into the corrupted set $\mathcal{C}$.*
   - *A* Sign *oracle that on input i, M, R' outputs* RSign(sk_i, M, R').

3. *At some point, $\mathcal{A}$ outputs a message M^* and a signature Σ^* with a ring R^*. $\mathcal{A}$ wins when they meet the following conditions:*

   - $\mathsf{RVerify}_{pp}(M^*, R^*, \Sigma^*) = 1$
   - $R^* \cap \mathcal{C} = \emptyset$ $(R^* \subseteq R \setminus \mathcal{C})$
   - *$\mathcal{A}$ has never queried the oracle* Sign$(\cdot, M^*, R^*)$.

1.1 Linkable Ring Signature

Linkable ring signature means that two signatures generated by the same signer can be identified and linked.

Honest User v.s. Dishonest User

Same Signer v.s. Different Signers

A linkable ring signature scheme consists of a tuple of efficient algorithms (RSetup, RKgen, RSign, RVerify, RLin for generating a public parameter, generating keys for users, signing messages, verifying ring signatures, linking ring signatures, respectively.

RSetup(n): Generates public parameters pp which are made available to all users.

RKgen(pp) Generates a public key pk and the corresponding secret key sk.

$\mathsf{RSign}_{pp}(sk, M, R)$: Outputs a signature Σ on the message $M \in \{0,1\}^*$with respect to the ring $R = (pk_0, ..., pk_{N-1})$. It is required that (pk, sk) be a valid key pair produced by RKgen(pp) and that $pk \in R$.

$\mathsf{RVerify}_{pp}(M, R, \Sigma)$: Given a candidate signature Σ on a message M with respect to the ring of public keys R, this algorithm outputs 1 if Σ is deemed valid or 0 otherwise.

$\mathsf{RLink}_{pp}(M, R, \Sigma, M', R', \Sigma')$: Given two valid message-signature pairs $(M, \Sigma), (M', \Sigma')$ with respect to two rings of public keys $R, R' (R \cap R' \neq \emptyset)$, this algorithm outputs 1 if they are generated by the same user or 0 otherwise.

Correctness

1. Verification correctness. Signatures signed by honest users legally would definitely be accepted by the Verification algorithm.

2. Linking correctness. Any pair of signatures generated by the same honest signer should be linkable.

Definition 4 (Correctness)**.** *A linkable ring signature* $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify}, \mathsf{RLink})$ *is correct if any ring signature generated by a honest user legally is valid and any pair of signatures generated by the same signer is linkable.* $pp \leftarrow \mathsf{RSetup}$*, any* $(pk, sk) \leftarrow \mathsf{RKgen}(pp)$*, any* R, R' *such that* $pk \in R$ *and* $pk \in R'$*, any* $M \in \{0,1\}^*$ *we have*

$$\mathsf{RVerify}_{pp}(M, R, \mathsf{RSign}_{pp}(sk, M, R)) = 1$$

.

$$\mathsf{RLink}_{pp}(M, M', R, R', \mathsf{RSign}_{pp}(sk, M, R), \mathsf{RSign}_{pp}(sk, M', R')) = 1$$

.

Definition 5 (Anonymity)**.** *A linkable ring signature scheme* $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify}, \mathsf{RLink})$ *provides anonymity if it is infeasible to determine which ring member issued a particular ring signature as long as the adversary is not provided with the corresponding secret key or signatures generated by the same signer.*

Given any PPT adversary $\mathcal{A}$*, its advantage in the following experiment is negligible in* n*:*

1. *The challenger* $\mathcal{C}$ *runs the* $\mathsf{RSetup}(n)$ *algorithm and runs the* $\mathsf{RKgen}(pp)$ *algorithm and returns all the public keys* $\{pk_i\}_{i=0}^{N-1}$ *to* $\mathcal{A}$*. Let ring* $R = (pk_0, pk_1, \ldots, pk_{N-1})$*.*

2. $\mathcal{A}$ *may query the* $\mathsf{Corrupt}$ *oracle on input* $i \in [N-1]$ *and* $\mathcal{C}$ *answers this query by giving* $\mathcal{A}$ *the* sk_i *and records* pk_i *into the corrupted set* $\mathcal{C}$*.*

3. $\mathcal{A}$ *outputs a message* M*, a ring* $R^* \subseteq R$ *and two identities* $j_0, j_1 \in [N-1]$ *where* $j_0, j_1 \notin \mathcal{C}$*.*

4. $\mathcal{C}$ *randomly picks a bit* $b \xleftarrow{\$} \{0,1\}$ *and feeds* $\mathcal{A}$ *with the signature* $\Sigma \leftarrow \mathsf{RSign}_{pp}(sk_{j_b}, M, R^*)$*.*

5. $\mathcal{A}$ *outputs a bit* b' *and succeeds if* $b' = b$*.(denoted as* Succ*)*

The advantage of $\mathcal{A}$ *is* $\Pr[\mathsf{Succ}] - 1/2$*.*

A linkable ring signature is unforgeable if it is infeasible to forge a ring signature without controlling one of the ring members.

Due to the linking property, we can deem the linking algorithm as the opening operation in the group signature, where it requires that it is infeasible for any PPT adversary to generate a valid message-signature pair and it would not trace to a corrupt user.

In the linkable ring signature, we require that it is infeasible for any PPT adversary to generate a valid message-signature pair and it would not be linked with another ring signature generated by a corrupt user.

Definition 6 (Unforgeability)**.** *A ring signature scheme* $(\mathsf{RSetup}, \mathsf{RKgen}, \mathsf{RSign}, \mathsf{RVerify}, \mathsf{RLink})$ *provides unforgeability if it is infeasible for any PPT adversary* $\mathcal{A}$ *to win the following experiment with non-negligible probability.*

1. *The challenger* $\mathcal{C}$ *runs the* $\mathsf{RSetup}(n)$ *algorithm and runs the* $\mathsf{RKgen}(pp)$ *algorithm and returns all the public keys* $\{pk_i\}_{i=0}^{N-1}$ *to* $\mathcal{A}$*, where ring* $R = (pk_0, pk_1, \ldots, pk_{N-1})$.

2. $\mathcal{A}$ *may query the following oracles adaptively and in any order:*

   - *A* $\mathsf{Corrupt}$ *oracle that on input* $i \in [N-1]$ *returns* sk_i *and records the corresponding public key* pk_i *into the corrupted set* $\mathcal{C}$.
   - *A* Sign *oracle that on input* i, M, R' *outputs* $\mathsf{RSign}(sk_i, M, R')$.

3. *At some point,* $\mathcal{A}$ *outputs a message* M^* *and a signature* Σ^* *with a ring* R^*. $\mathcal{A}$ *wins when they meet the following conditions:*

   - $\mathsf{RVerify}_{pp}(M^*, R^*, \Sigma^*) = 1$
   - $\mathcal{A}$ *has never queried the oracle* $\mathsf{Sign}(\cdot, M^*, R^*)$.
   - *Denote* Σ_j *as the ring signature generated by* $\mathsf{RSign}(sk_j, M_j, R_j)$ *where* $pk_j \in R_j \subseteq R$. *Here we require that* $\mathcal{C} \cap R^* = \emptyset$*, or for any* $pk_j \in \mathcal{C} \cap R^*$, $\mathsf{RLink}(M^*, R^*, \Sigma^*, M_j, R_j, \Sigma_j) = 0$.

The linking algorithm can only be applied to two intersectant rings! ($pk_j \in \mathcal{C}$,$pk_j \in R_j$,$pk_j \in \mathcal{C} \cap R^*$,$pk_j \in R^*$, so $R_j \cap R^* \neq \emptyset$)

Anonymity

It should be infeasible for any adversary to identify the actual signer with probability greater than 1/n where n is the size of the ring.

Unforgeability

The unforgeability of a ring signature scheme is defined via the following game, denoted by $Game_{forge}$, between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

- Setup. The challenger C runs Setup with security parameter n and generates system parameter $param$. $\mathcal{C}$ sends $param$ to $\mathcal{A}$.

- Query. The adversary $\mathcal{A}$ may query $\mathcal{RO}$, $\mathcal{CO}$ and $\mathcal{SO}$ for a polynomial bounded number of times in an adaptive manner.

- Output. The adversary A outputs a forgery (m^*, $\sigma*$, R^*). A wins $Game_{forge}$ if

  - Verification(m^*, $\sigma*$, R^*) = accept;
  - (m^*, $\sigma*$, R^*) has not been queried by $\mathcal{A}$; and no public key in R^* has been input to $\mathcal{CO}$.

Remark: This requirement can be covered by our new defined unforgeability experiment above when $R^* \cap \mathcal{C} = \emptyset$.

Linkability

Link algorithm always outputs **linked** for two signatures generated by a same signer.

Remark: This property contains two aspect: honest signer or dishonest signer. The former situation is covered by the linking correctness and the later can be contented by the unforgeability.

If a malicious user with a single secret key generates two unlinked ring signatures, he can win the unforgeability experiment with non-negligible probability where $|\mathcal{C}| = 1$ and $\mathsf{RLink}(M_1, R_1, \Sigma_1, M_2, R_2, \Sigma_2) = 0$.

Nonslanderability A signer cannot frame other honest signers by generating a signature linked with those produced by the honest signer.
Remark: This property can be covered by the anonymity experiment. If a signer can generate a signature linked with those produced by the honest signer, he can choose this honest signer as one of the challenge identities i_0. As along as the challenge signature can be linked with his signature, he outputs 0, otherwise 1. In this way, he wins the anonymity experiment with the same probability in the Nonslanderability experiment.

In conclusion, our security model (Anonymity and Unforgeability **Linkability**) covers all previous requirements.