Monero's Building Blocks
Part 3 of 10 – *Introduction to Ring Signatures*

Bassam El Khoury Seguias

BTC: 3FcVvBZwTUkUrcqJd16RcjR42qT2tDWHWn

ETH: 0xb79Fb9194C8Cc6221368bb70976e18609Ab9AcA8

March 6, 2018

1 Introduction

This is a brief article that introduces the concept of a ring signature. In parts 4, 5, 6, and 7 we will look at specific instances of ring schemes – including those used in earlier and more recent versions of the Monero project – and analyze their security properties.

In 1991, Chaum and Van Heyst introduced a new class of signature schemes known as *group signatures* [2]. The core of the model consisted of a trusted entity known as the *group manager* that clusters a subset of users together into a group. The group manager provides each member of the group with a separate private key. The ingenuity of this structure lies in the fact that any member can sign messages in an anonymous fashion. This means that anybody who can access the signature, can also verify that it was created by one of the group members without knowing who specifically. The only entity that can identify the real signer is the trusted group manager. In group signature schemes, the anonymity of signers comes at the expense of relinquishing power to the group manager. Indeed, the trusted group manager is the only entity that:

- Decides who joins the group.
- Decides which member(s) get(s) banned from the group.
- Chooses the private key allocated to each member of the group.
- Identifies the real signer whenever a message is signed.

This setting works best if the group members agreed to cooperate beforehand . The group manager can then serve as the enforcer of this cooperation, revoking the membership of anyone trying to game the system.

The *anonymity* of group signatures paved the way to another class of *signer-ambiguous* shemes known as *ring signatures*. The expression *ring signature* was first coined by Rivest, Shamir, and Tauman [3]. Note that schemes fitting the definition of a ring signature have been proposed way before the publication of this paper. In a ring scheme, there does not exist a pre-defined group of users. As a consequence, there does not exist any omnipotent group manager. Instead, the actual signer defines a set of members of her choosing before she signs a message. This set is known as a *ring*. The only constraint is that the ring must include the actual signer. The signer creates a signature using her private key and all the other ring members' public keys. The ring can be arbitrary without the need to inform selected members of their participation – (all that is needed is access to their public keys which is usually common knowledge). The reason behind adopting the *ring* terminology is that *"rings are geometric regions with uniform periphery and no center"* [3].

2 Definition and Security analysis

A ring signature on a message m is a string that depends on:

- The message m.
- The public and secret keys of the signer.
- The public keys associated with an arbitrarily-specified ring of users.
- Some randomly chosen data.

Anyone can check whether it corresponds to a member of the ring but can not know the identity of the actual signer. More formaly, a generic digital signature scheme is defined as a set of 3 algorithms:

- **The key generation algorithm** $\mathcal{G}$. On input 1^k, where k is the security parameter, it produces a pair (pk, sk) of matching public and secret keys. The algorithm is modeled as a PPT Turing machine.
- **The ring signing algorithm** Σ. Suppose a user A_π decides to sign a message m on behalf of a ring of users $\{A_1, ..., A_n\} \ni A_\pi$. Σ takes three inputs including:

  1. The message m.
  2. The set $\{pk_1, ...pk_n\}$ of the public keys of the ring members
  3. The private key of the signer sk_π.

  It then outputs a ring signature $\sigma(m)$ on message m. The algorithm is modeled as a PPT Turing machine.
- **The ring verification algorithm** $\mathcal{V}$. Given a ring siganture σ, a message m, the set $\{pk_1, ...pk_n\}$ of public keys of the ring members, $\mathcal{V}$ is a boolean function that returns *True* if the signature is valid and *False* otherwise. $\mathcal{V}$ is a deterministic algorithm as opposed to probabilistic.

When we introduced digital signature schemes with only one user (part 1 of this series), we mandated two security measures:

1. **Correctness**: Any signature generated by Σ must pass the verification test with overwhelming probability.
2. **EFACM Unforgeability**: Even when an adversary can have access to valid signatures on any messages of his choosing – other than m –, the probability that he successfully forges a signature on message m must be negligible in k.

   For ring signatures, the EFACM mechanics differ slightly from the non-ring case described in previous parts. [1] defines various models for existential forgeability. The one we use in this work is known as *unforgeability against fixed-ring attacks.* It is defined as follows in [1]:

   - $\forall i \in \{1, .., n\}$, key pairs (sk_i, pk_i) are generated using $\mathcal{G}$, and the ring $L \equiv \{y_1, .., y_n\}$ is fed to the adversary.
   - The adversary can access any ROs applicable in the scheme, and can also access a signing oracle:

     $$SO : \{1, .., n\} \times \{0, 1\}^* \longrightarrow \text{Range of signatures.}$$

     SO takes an index $i \in \{1, .., n\}$ and a message $m \in \{0, 1\}*$, and outputs a signature authored by ring member i on message m. In other terms, $SO(i, m) \equiv \Sigma(L, m, x_i)$
   - The adversary outputs a valid forgery $\sigma_{forge}(L, m*)$ (i.e., one that passes $\mathcal{V}$'s test). Moreover, the adversary must have never queried SO on message $m*$.

   In a *fixed-ring attack*, the adversary can only query the signing oracle on the full ring L. That means that signatures issued by the signing oracle are created with respect to L. Moreover, $\mathcal{V}$'s verification algorithm is conducted with respect to L.

For ring signatures, we add a third requirement: **Anonymity**. In what follows, we introduce two definitions of *anonymity* also known as *signer-ambiguity.* The distinction between the two definitions has to do with the possibility that in a set of n ring members, a subset of them may be compromised (i.e., their private keys may be stolen or known).

- **Anonymity definition #1**: Given any subset $\mathcal{D}_t = \{\hat{sk}_1, ...\hat{sk}_t\}$, $t = 1, ...n$, of compromised members of a ring with n elements, an adversary can not do better than random guessing when trying to identify the real signer. This definition may seem counter-intuitive: if we know the private key of a member, we would think that we should be able to say whether she is the actual signer or not. However, this definition states that even if the private keys of all the members are revealed, no adversary can identify the real signer with probability better than random guessing. More formally, let $\mathcal{A}(\omega)$ be a PPT adversary with random tape ω that takes 4 inputs:
  - Any message m.

  - A ring L of the n public keys $\{pk_1, ...pk_n\}$ of the ring members. L includes the public key pk_π of the actual signer.
  - A list $\mathcal{D}_t \equiv \{\hat{sk}_1, ...\hat{sk}_t\}$ of compromised private keys of ring members ($0 \leq t \leq n$). Note that $\mathcal{D}_t$ can be empty. Also note that $\hat{sk}_i$ may be different than sk_i but we always have $\mathcal{D}_t \subseteq \{sk_1, ..., sk_n\}$
  - A valid signature $\sigma(m)$ on message m, with ring L and actual signer private key sk_π.

  $\mathcal{A}(\omega)$ outputs an index corresponding to the ring member in L that it thinks is the actual signer.

  This definition of signer-ambiguity mandates that for any polynomial $Q(k)$ in the security parameter k we have:

$$\tfrac{1}{n} - \tfrac{1}{Q(k)} \leq P[\mathcal{A}(\omega)(m, L, \mathcal{D}_t, \sigma(m)) = \pi \mid \sigma(m) \ is \ valid] \leq \tfrac{1}{n} + \tfrac{1}{Q(k)}$$

  Roughly speaking, that means that the probability of guessing the real signer is $\approx \frac{1}{n}$. The definition implies the *exculpability* of any user. This means that even if a signer is coerced or subpoenaed to release her private key, nothing can be done to prove that she is the real signer.

- **Anonymity definition #2**: This definition is a bit weaker than the first one in the sense that if the adversary knows t secret keys out n (excluding that of the signer), the best he can do is randomly guess over the remaining $(n - t)$ uncompromised members. In this definition, having access to a secret key can either help confirm or rule out whether the corresponding ring member is the actual signer. Contrary to definition #1, this definition does not ensure exculpability as previously defined. More formally, and using the same notation introduced earlier, definition #2 mandates that:

$$\tfrac{1}{n-t} - \tfrac{1}{Q(k)} \leq P[\mathcal{A}(\omega)(m, L, \mathcal{D}_t, \sigma(m)) = \pi \mid \sigma(m) \ is \ valid] \leq \tfrac{1}{n-t} + \tfrac{1}{Q(k)}$$
$$\text{if } sk_\pi \notin \mathcal{D}_t \text{ and } 0 \leq t < n - 1.$$

$$P[\mathcal{A}(\omega)(m, L, \mathcal{D}_t, \sigma(m)) = \pi \mid \sigma(m) \ is \ valid] > 1 - \tfrac{1}{Q(k)}$$
$$\text{if } sk_\pi \in \mathcal{D}_t \text{ or } t = n - 1.$$

References

[1] A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *Journal of Cryptology*, 22(1):114–138, 2008.

[2] D. Chaum and E. Van Heyst. Group signatures. *Advances in Cryptology - EUROCRYPT '91*, Lecture Notes in Computer science(547):257–265, 1991.

[3] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. *Advances in Cryptology - Asiacrypt 2001*, Lecture Notes in Computer Science(2248):552–565, 2004.