Dual-Ring Signature Report

Xiangyu Hui

20/03/2023

1 Introduction

This report provides an overview of Dual-Ring Signature[1] algorithm, which enables a member of a group to sign a message without revealing their identity.

2 Dual-Ring Algorithm Overview

2.1 Algorithm Description

2.1.1 SET- UP

First, we choose a base curve point G_0 to generate a set of public keys **pks**. For each public-secret key pair, the algorithm select a random $sk \in \mathbb{Z}_p$ and compute $pk = sk \cdot G_0$.

2.1.2 Dual-Ring Construction

The Dual-Ring algorithms involves the following 3 functions:

$$
\begin{aligned}
\text{A (r)} &= r \cdot G_0 \\
Z(sk, r, c) &= r - c \cdot sk \mod p \\
V(pk, c) &= c \cdot pk
\end{aligned}
$$

Here we abuse the symbol $\sum$ for consecutive points addition, respectively: $\sum_{i=1}^{n} G_i = G_1 + G_2 + \cdots + G_n$

Sign. Given a message m and a signer's private key sk_i, the signer will first choose a random number $r \in \mathbb{Z}_p$, and $n-1$ random numbers $c_j \in \mathbb{Z}_p, \forall i \neq j$. Then computes the intermediate commitment R as: $R = A(r) + \sum_{j \neq i} V(pk_j, c_j)$. After that, the signer forge the c_i as: $c_i = H(m, \mathbf{pks}, R) - \sum_{j \neq i} c_j$. Finally, the signer computes response $z = Z(sk_i, r, c_i)$, and return signature $sig = (c_0, \ldots, c_n, z)$.

Verify. To Verify, the verifier will reconstruct the intermediate commitment R based on the signature and publick keys: $R = A(z) + \sum_j V(pk_j, c_j)$. If the signature is generate by the valid secret key owner the following equation should hold: $H(m, \mathbf{pks}, R) = \sum_j c_j$

Algorithm 1 Dual-Ring Signature Scheme

procedure SETUP(n)
  Initialization
  for i in n **do**
    $pk_i = sk_i \cdot G_0$
  return $\mathbf{pks}, \mathbf{sks}$
procedure SIGN($m, \mathbf{pks}, sk_i$)
  $r \leftarrow \text{random}(\mathbb{Z}_p), c_j \leftarrow \text{random}(\mathbb{Z}_p), \forall j \neq i$
  $R = A(r) + \sum_{j \neq i} V(pk_j, c_j)$
  $c_i = H(m, \mathbf{pks}, R) - \sum_{j \neq i} c_j$
  $z = Z(sk_i, r, c_i)$
  return $\sigma = (c_0, \ldots, c_n, z)$
procedure VERIFY($m, \mathbf{pks}, \sigma$)
  $R = A(z) + \sum_j V(pk_j, c_j)$
  if $H(m, \mathbf{pks}, R) = \sum_j c_j$ **then**
    return 1s
  else
    return 0

2.2 Security Features

Definition .1 *R is a set of tupples(λ, x, w), where λ is public parameter x is called instance, w is called witness. V, P are refer to Verifier and Prover.*

2.2.1 Completeness

Definition .2 (Perfect Completeness) *:*
$\mathbf{Pr}[(\lambda, x, w) \in R < P(\lambda, x, w), V(\lambda, x)] = 1$

Proof: As $z = r - c_i \cdot sk_i$, for verifier V:

$$\begin{aligned} \mathrm{R}_v &= r \cdot G_0 - c_i \cdot sk_i \cdot G_0 + c_i \cdot pk_i + \textstyle\sum_{j \neq i} c_j \cdot pk_j \\ &= r \cdot G_0 + \textstyle\sum_{j \neq i} c_j \cdot pk_j \\ &= R_p \end{aligned}$$

Thus, $H(m, \mathbf{pks}, R_v) = \sum_j c_j = H(m, \mathbf{pks}, R_p)$ holds, which means Vrifier will always accept, as long as z and c_i are computed from P. ∎

2.2.2 Soundness

Definition .3 (Soundness) *:*
For every deterministic prover strategy P', if P' sends a value $\sigma! = SIGN(m, \mathbf{pks}, sk_i)$ at the start of the protocol, then $Pr[out(V, x, w, P') = 1] \leq negl(\lambda)$.

Proof: If P' does not know one of the secret key sk_i, the only way for P' pass the verification is find a z where it satisfies $\sum_j c_j = H(m, \mathbf{pks}, z \cdot G_0 + \sum c_j \cdot pk_j)$. But according to the Hash function's security property, it is impossible to find such z within polynomial time. ∎

3 Performance Analysis

3.1 Computational Complexity

The classical link-able ring-signature has n responses, but Dual-Ring signature contains only 1 response. the signature size is reduced from 2n to n+1.

References

[1] Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. Dualring: Generic construction of ring signatures with efficient instantiations. Cryptology ePrint Archive, Paper 2021/1213, 2021. `https://eprint.iacr.org/2021/1213`.