Ring Signature Variants

Anka Hu
MIT
ankaa@mit.edu

Brandon Luo
MIT LL
brandon.luo@ll.mit.edu

Vetri Vel
MIT
vetri@mit.edu

Abstract

We implement ring signature variants such as a post-quantum t-of-n ring signature scheme and a linkable/one-time within-ring signature scheme. We also show an improved method for non-interactive within-ring signatures in added $O(n)$ space which improves upon the previous non-interactive within-group signature scheme of Hu et. al [6] which has a space complexity of $O(n^2)$. We do this by using a black-box transformation to convert any minimal ring signature to a within-ring signature. We also measure signature sizes and computational requirements for the post-quantum threshold ring signature and linkable/one-time within-ring signature schemes.

1. Introduction

Within-group signatures are a cryptographic primitive in which any member of a group has the ability to anonymously produce a signature on behalf of the group which cannot be distinguished from forgery by external parties. Within-group signatures accomplish this by combining existing ideas from ring signatures [8], where members of a group anonymously create signatures on the behalf of the group, and designated verifier signatures, where only a fixed set of individuals are given the ability to correctly verify signatures. Such schemes can play important roles in supporting whistleblowing or authenticated but deniable communications between members of a secret group.

In within-ring signatures, the anonymity can not be revoked by the verifier and ring members can be added and removed for each signature. We introduce a transformation which converts any minimal ring signature to a within-ring signature in a black box way by encrypting the message with the public keys of the verifiers. The anonymity of the ring signature can not be revoked but the inability of non-verifiers to verify the proof depends on the honesty of the verifiers. In general, there is a within-witness transform which can be used to convert any minimal witness indistinguishable proof to a within-witness indistinguishable proof with added space and time complexity which is linear in the number of verifiers.

The advantage of within-ring signatures is the ability to leak a secret in an authenticated way to an arbitrary set of verifiers while maintaining a strong level of anonymity. This generalizes from ring signatures where the verifier set is the set of all verifiers and from designated verifier signatures where there is only one verifier [7]. In the context of whistleblowing, this allows for the secret to be privately leaked within a group without allowing the signature to be verifiable in a larger context.

2. Related Work

The current non-interactive scheme for within-group signatures by Hu et. al has a signature size which is quadratic in the size of the group. The interactive scheme has a communication complexity that is linear in the size of the group. It is not shown that their scheme is able to add or remove group members, and they list that as future work. Finally, the hardness assumption used for the cryptosystem is the discrete logarithm that is a special case of the hidden subgroup problem for normal subgroups and is efficiently computable by a quantum computer.

We solve the problems in the paper that are listed as future work. Our first solution modifies the non-interactive solution of Hu et. al to make the signature size linear in the size of the ring and adds support for adding and removing members. We apply this to a threshold ring signature scheme [5] and a linkable/one-time within-ring signature scheme which we implement.

The idea is to encrypt some value like the message which is used for verification for verification with the public keys of the verifier set to prevent non-verifiers from verifying the signature. The message is removed from the signature and encryptions of the message with the public key of each verifier $E(pk_i, m)$ are added to the original signature.

It may be possible to generalize the idea behind the within-ring transform to most witness indistinguishable proofs. The idea is to generalize to non-interactive within-witness indistinguishable proofs by encrypting some part v of the proof which is needed for verification with the public keys of all of the verifiers and removing v from the proof.

3. Within-Ring Signatures

We begin by introducing the building blocks of a within-ring signature scheme.

3.1. Algorithms

A within-ring signature scheme is define by three algorithms: $\mathsf{Setup}(\cdot), \mathsf{Sign}(m, \cdot)$ for some message $m \in \mathcal{M}$, and $\mathsf{Verify}(\cdot)$.

3.1.1 Key Generation

We assume that there exists a mapping from public keys to identities.

$$\underline{\mathsf{Setup}(1^n)} \rightarrow (s_i, pk_i)_{i=1}^n$$

3.1.2 Message Signing

One ring member is able to sign for the entire ring since the signature algorithm only requires one private key corresponding to one of the public keys in the ring.

$$\underline{\mathsf{Sign}(m, s_i, \{pk_i\}_{i=1}^n)} \rightarrow (\sigma, c)$$

1. $\sigma \leftarrow ring\_sign(m, s_i, \{pk_i\}_{i=1}^n, pk_r)$
2. $E_i = E(pk_i, m)$
3. Output $(\sigma, (pk_i; E_i))$

3.1.3 Verification

Only verifiers from within the ring should be able to efficiently decrypt the ciphertext to retrieve message and verify the validity of the signature. We describe the scheme as below:

$$\underline{\mathsf{Verify}(\sigma, s_i, (pk_i; E_i))} \rightarrow \{0, 1\}$$

1. $m \leftarrow D(s_i, E_i))$
2. Output $\mathsf{ring\_verify}(m, \{pk_i\}_{i=1}^n)$,

where ring_verify is the verification algorithm for the ring signature.

3.2. Security Properties

Within-ring signatures have the security goals of correctness, within-ring unforgeability, and anonymity. We give formal definitions for each of these below.

3.2.1 Correctness

For all messages and valid secret key and public key pairs, correctness of the scheme is achieved if verification of a valid signature by a member of the within-ring is successful. In other words,

$$\mathsf{Verify}(\mathsf{Sign}(m, s_i, \{pk_i\}_{i=1}^n)) = 1 \tag{1}$$

3.2.2 Within-Ring Unforgeability

In addition, we want to ensure that a member of the within-ring cannot produce an signature $\sigma*$ which will be successfully verified without knowing the secret key.

Intuitively, one can note that the unforgeability property of ring signatures is not affected by the encryption of the message with the public keys. There is a reduction from an efficient algorithm $\mathcal{A}$ which can break the EUF-CMA property of the within-ring signature to an efficient algorithm $\mathcal{A}'$ which can break the EUF-CMA property of the ring signature, which we will outline below:

$$\underline{\mathcal{A}'} \rightarrow \sigma*$$

1. An efficient algorithm $\mathcal{A} \rightarrow (m, \{\sigma, (pk_i; E_i)\})$
2. Output (m, σ)

3.2.3 Anonymity

One of the key security guarantees of ring signatures is the anonymity property. In other words, given a signature σ, signers are unable to determine from which member it was generated. We further categorize anonymity properties into statistical anonymity and computational anonymity.

3.2.4 Statistical Anonymity

A widely-used definition of anonymity for ring signature schemes describes how the signatures generated by different signers are indistinguishable and leak no information about the underlying signer. They are usually required to be statistically indistinguishable, which implies

$$\{\mathsf{Sign}(m, s_i, \{pk_i\}_{i=1}^n)\} \approx_s \{\mathsf{Sign}(m, s_i', \{pk_i\}_{i=1}^n)\},$$

The signatures distributions are generated over the message and public key spaces.

3.2.5 Computational Anonymity

A computational definition of anonymity requires signatures generated by different members of the ring to be computationally indistinguishable from one another. We give a security definition using the following anonymity

game instead of using statistical distance to allow definitions of computational security properties.

Anonymity Game:

1. The challenger randomly generates a signer index $i \leftarrow_R [n]$
2. The challenger randomly generates n secret-key, public-key pairs (s_j, pk_j) for all members of the ring, including the signer.
3. The adversary can query a ring signature oracle to generate ring signatures from message-signer pairs (m, i) and repeats this action a polynomial number of times
4. The adversary sends a new message m' to the challenger
5. The challenger sends back a signature $\sigma = \mathsf{Sign}(m', s_i, pk_i)$ using the keys of the signer
6. The adversary outputs a guess for the signer i

Computational anonymity is satisfied if the probability that the adversary correctly wins the game is at most $1/n + negl(\lambda)$.

3.2.6 Proof of Computational Anonymity

Intuitively, anonymity is not affected since the encrypted values of m can be computed by anyone with knowledge of the public keys. Assume there exists some algorithm $\mathcal{A}$ which can win the Anonymity Game described above with $1/n$ plus some non-negligible probability for a within-ring signature. We can use this to construct an efficient algorithm $\mathcal{A}'$ which can break the anonymity property for the ring signature as follows:

Efficient $\mathcal{A}'$:

1. Send m to the challenger and to get σ
2. Compute the encryption $E_i = E(pk_i, m)$ of m for each public key of the verifiers pk_i
3. Output $\mathcal{A}(\sigma, (E_i; pk_i))$

3.2.7 Outside Ring Obfuscation

In addition to within-ring anonymity properties, our within-ring signature also provides security guarantees towards obfuscating the validity of a signature from parties outside the set of verifiers.

3.2.8 Minimal Signatures

We define a signature σ to be *minimal* if no efficient algorithm is able to use only σ to distinguish between the messages used to create the signature.

3.2.9 Key Secrecy

Furthermore, the secret key used for the ring signature should be indistinguishable from uniformly random if the signature is known and signature can't be verified.

3.2.10 Proof

Intuitively, a non-verifier can't compute the message without the secret keys and is unable to verify the signature.

We construct a forgery algorithm without the secret key which is indistinguishable from a valid signature to prove outside ring obfuscation.

Steps:

1. Construct a ring signature with a uniformly random key and some message m.
2. Encrypt the message m with the public keys of the verifiers and remove the message from the output.

3.3. Adding and Removing Ring Members

The sign and verify algorithms have support for creating ring signatures for arbitrary sets of public keys.

3.4. Complexity Analysis

An additional n encrypted messages are added, while the message is removed. This requires an additional n public key operations.

4. Post Quantum Trapdoor Commitments

Trapdoor commitments can be used to implement threshold ring signature schemes in a black box way: we implement post-quantum trapdoor commitments using post-quantum sigma protocols and a post-quantum one-way function.

A *trapdoor commitment* is a commitment c to some value y such that the holder of some secret s corresponding to the committment is able to change the opened value of c to some other value y'. Trapdoor commitments are indistinguishable from a non trapdoor commitment.

4.1. Algorithms

4.1.1 Hard Instance Generation

$\mathsf{Setup}(1^\lambda) \rightarrow (s, w)$

4.1.2 Commitment

$\mathsf{Commit}(s, m) \rightarrow$ (commitment, opening)

1. Run the simulator with m as the challenge and s as the statement to prove.
2. Output the first message as the commitment.
3. The message is the challenge. In practice, a random oracle hash function is applied to the message to make it a fixed size.
4. The response is the opening.

4.1.3 Trapdoor Commitment

$\underline{\mathsf{Trapdoor}(s, m)} \rightarrow$ (commitment, state)

1. Run the prover with s as the statement to prove.
2. Output the first message as the commitment.
3. The state of the prover is also output to use as input for the trapdoor opening

4.1.4 Trapdoor Opening

$\underline{\mathsf{TrapOpen}(state, c, w, m', c)} \rightarrow$ response

1. Output the response generated by the prover as the opening. The prover can generate a response for any challenge (message) because the witness is known.

4.1.5 Verification of Opening

$\underline{\mathsf{VerifyOpen}(\mathsf{c}, \mathsf{m}, \mathsf{o})} \rightarrow \{0, 1\}$

1. Run the protocol and output the result of the verifier for commitment c, message m, and opening o. The commitment is the first message, the message is the second message, and the opening is the third message.

4.2. Security

The proof of security assumes that the sigma protocol has the special honest verifier zero knowledge property.

4.3. Correctness

The correctness of the trapdoor commitment scheme follows from the completeness of the sigma protocol.

4.4. Binding

The classical binding property of the commitment follows from the special soundness of the sigma protocol. This does not extend to the quantum setting, where a classically binding commitment scheme allows for a commitment to be opened to multiple messages [10]. This is not a problem in the protocol since there is at least one honest signer who performs a measurement on the message and opening for all of the commitments.

4.5. Hiding

The hiding of the commitment follows from trapdoor indistinguishability.

4.6. Trapdoor Indistinguishability

The indistinguishability of the trapdoor commitment and existence of the simulator for making commitments follows from the zero knowledge property of the sigma protocol.

5. Post Quantum (t, n) Ring Signatures

We build post quantum (t, n) ring signatures using post quantum trapdoor commitments, secret sharing, and a random oracle which has the same domain and range using the scheme by Haque et. al. We present a simplified version which uses a random oracle for the soundness proof, but the implemented version uses the method for post-quantum non-interactive zero knowledge proofs by Unruh [9].

5.1. Algorithms

We assume that the there is a mapping from public keys to identities and that the public keys are encoded in some way in the signature or agreed upon beforehand.

5.1.1 Hard Instance Generation

The instances are generated from a quantum-hard one way function f to define elements $(i, f(i))$ of a relation R. These are used for the trapdoor commitments.

$\underline{\mathsf{Setup}(1^n)} \rightarrow (s_i, w_i)_{i=1}^n$

5.1.2 Signing

For the input values for the points (i, o) we use the index i for ring member i starting from 1 and ending at n. A different set of input values (roots of unity) may be used in certain fields to speed up polynomial interpolation. In the actual implementation, the step marked simplified is replaced by the transform mentioned in section 7. The verification step is replaced by a similar transform.

A high-level overview of this threshold signature is described below:

$\underline{\mathsf{Sign}(m, \{s_i\}_{i=1}^t, \{pk_i\}_{i=1}^n)} \rightarrow (m, p_i, c_i)$

1. Generate random points $\{p_i\}_{i=1}^n \xleftarrow{R} F$
2. Generate the corresponding commitments:
   - $c_i \leftarrow \mathsf{Commit}(pk_i, p_i), i \in \{t+1, ..., n\}$
   - $c_i \leftarrow \mathsf{TrapCommit}(sk_i, pk_i, p_i), i \in \{1, ..., t\}$

3. Compute the challenge $p_0 \leftarrow H(m, \{c_i\}_{i=1}^{n})$ (simplified)
4. Generate a polynomial $\mathsf{poly}(\cdot) \leftarrow$ from a polynomial interpolation of $\left(p_0, \{p_i\}_{i=t+1}^{n}\right)$
5. All t threshold members now re-evaluate their point to be on the polynomial $p_i = \mathsf{eval\_poly}(i+1)$ for $i \in \{1, ..., t\}$.
6. Output (m, p_i, c_i)

5.1.3 Verification

$\underline{\mathsf{Verify}(m, p_i, c_i)} \rightarrow$

1. Verify openings (points) to commitments
2. Compute $p_0 = H(m, c_i)$ (simplified)
3. Compute $\mathsf{poly} = \mathsf{interpolate}(p_i)$
4. Output 1 if $\deg(\mathsf{poly}) \leq n - t$

5.2. Protocol

For simplicity and efficiency, a leader is used by Haque et. al to generate the commitments, points, and openings for non-signers. These values are sent to every signer: part of the reason is because classically binding commitments are not secure when the commitments are made by a quantum computer. Each signer then sends commitments for their points and openings (points) to all of the signers. The signers can then combine these commitments and points to construct a signature. We omit some details for simplicity.

5.3. Security

The proof of security is in the quantum random oracle model.

5.3.1 Correctness

The correctness of the threshold ring signature follows from the correctness of the polynomial interpolation algorithm and the correctness of the trapdoor commitment scheme.

5.3.2 Unforgeability

The unforgeabilility of the threshold ring signature scheme follows from the binding of the commitment scheme, the randomness of the random oracle, and the randomness of the points, and the properties of random polynomials.

5.3.3 Anonymity

The anonymity of the threshold ring signature scheme follows from trapdoor indistinguishability. The ordering of the ring members should be uniformly random and independent each time the signing algorithm is performed.

5.3.4 Within-Ring Transformation

The within-ring transformation applies since the signature scheme is minimal. This answers the problem from Hu et. al about constructing threshold within-ring signatures.

5.4. Complexity Analysis

In the presented scheme, there are n points and n commitments, which results in a space complexity which is linear in the ring size. In the actual implementation, the signature size also scales linearly with statistical security parameters for increasing soundness (repeating proofs/generating multiple openings).

The time complexity is dominated by the polynomial interpolation algorithm which has a general runtime of $O(n^2)$, although for certain fields it may be possible to do polynomial interpolation in $O(n \cdot log^2(n))$.

6. Post Quantum Non-Interactive Zero Knowledge Proofs

The standard soundness proofs for non-interactive zero knowledge proofs using rewinding or straight line extraction do not work in the quantum computational model for the general case. We use the transformation by Unruh to implement post-quantum trapdoor commitments and post-quantum threshold ring signatures by replacing the random oracle hash function with the transform.

The idea is to avoid measuring the inputs to the random oracle by using a random permutation which can be inverted in the soundness proof.

6.1. Transformation

s is a statement in the relation which is instantiated by a quantum-hard one way function (s, w) for which w is the witness. The random permutation π can be constructed by using a random oracle hash function with the same domain and range . The value k_i is a uniformly random value used for the random permutation. o is an index in the set of encrypted openings which determines which opening will be output in the signature. Our description of the transformation is specific to the threshold ring signature implementation.

We can repeat the following protocol for some t iterations to increase soundness as needed:

1. $\mathsf{Challenge}_i, \mathsf{Open}_i \leftarrow H(m, \langle \text{commits} \rangle, i), i \in [c]$
2. $k_i \leftarrow R, i \in [c]$
3. $e_i \leftarrow \pi(\mathsf{Open}_i, k_i, \langle \text{commits} \rangle)$
4. $o \leftarrow H(m, \langle \text{commits} \rangle, e_i)$
5. Output $\mathsf{Open}_o, k_o, \langle \text{commits} \rangle, \{e_i\}$

6.2. Security

Due to the 3-special soundness of the sigma protocol, we only make 4 openings. The number of openings is a power of two so that the index output by the hash function will not require rejection sampling. In the soundness proof, this is also enough openings to extract the witness.

6.3. Complexity Analysis

The space and time complexity are linear in the number of repetitions and number of openings made for the set of commitments.

7. Implementation

7.1. Linkable/One-Time Within-Ring Signatures

We extended pyring (a linkable/one-time ring signature scheme) by using our within-ring transform. We used El Gamal with elliptic curves points as the cyclic group elements to generate shared symmetric keys to encrypt verification values.

7.2. Performance

The estimated space requirements are a few hundred bytes for a linkable/one-time within-ring signature with a ring size of 3 and 128 bit security. The space used scales linearly with the ring size and the time used scales linearly with the ring size.

7.2.1 Signature Sizes

The signature sizes scale linearly with the number of ring members.

n	Size (b)
3	572
30	4448
300	41742

7.2.2 Signing Times

The signing times scale linearly with the number of ring members. Our results are nonlinear probably because of constant factors such as startup times. We were unable to test with many ring members due to limits on the number of open files.

n	Seconds
3	0.05
30	0.06
300	0.15

7.2.3 Verification Times

The verification times scale linearly with the number of ring members.

7.3. Threshold Ring Signatures

We build post-quantum trapdoor commitments from post-quantum sigma protocols that were proposed in Giacomelli et al. [3] and optimized by Chase et. al [2]. We then implement the transform used for the non-interactive zero knowledge proofs and use it to build the post-quantum threshold ring signature scheme by replacing the random oracle hash function with the transform.

7.4. Performance

The estimated space requirements are 2MB for a threshold ring signature with a ring size of 3 and 128 bit security. The space used scales linearly with the ring size and the time used scales quadratically with the ring size.

7.4.1 Signature Sizes

The signature sizes scale linearly with the number of ring members.

t	n	Size (Mb)
2	3	1.9
20	30	19
200	300	183

7.4.2 Signing Times

The signing times scale quadratically with the number of ring members and is reduced as the threshold increases.

t	n	Time (seconds)
2	3	0.2
20	30	2.4
200	300	25.4

7.4.3 Verification Times

The verification times scale quadratically with the number of ring members.

t	n	Time (s)
2	3	0.3
20	30	2.9
200	300	29.2

7.5. Security Parameters

For the linkable within-ring signatures, we use the same parameters as the original scheme. We use the parameters used by the Picnic signature scheme for the trapdoor commitments, a field size of 2^{256}, and 4 encrypted openings for the threshold ring signatures.

8. Conclusions

We were able to implement new ring signature variants and measure their performances for different ring sizes. We believe that more work needs to be done to make post-quantum threshold ring signatures usable, such as reducing the number of interactions, the amount of computation, and the sizes of signatures.

We also came up with an optimized implementation of within-ring signatures. We think that it may have applications to extendable threshold ring signatures to hide the signature until the signature is fully constructed, whistle-blowing, voting, and cryptocurrencies.

We open source our implementations at `https://github.com/max-p-log-p/pqthr` and `https://github.com/max-p-log-p/within-ring`.

Our contributions are as follows: all of us worked on the paper and presentation, the second author implemented the signature schemes.

9. Future Work

Extendable threshold ring signatures [1] allow for a reduction in the amount of interaction needed for threshold ring signatures and also allows for stronger anonymity guarantees.

We think it may be possible to implement post-quantum extendable threshold ring signatures with a space complexity which is sublinear in the number of ring members [4], since there are existing threshold ring signature constructions with a sublinear space complexity.

We also think that the current threshold ring signature could be made a lot more efficient by using lattice based trapdoor commitments or by using isogeny based trapdoor commitments.

Finally, we think it would be difficult but interesting to find new applications for within-ring signatures and threshold ring signatures.

References

[1] Gennaro Avitabile, Vincenzo Botta, and Dario Fiore. Extendable threshold ring signatures with enhanced anonymity. In *IACR International Conference on Public-Key Cryptography*, pages 281–311. Springer, 2023.

[2] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 acm sigsac conference on computer and communications security*, pages 1825–1842, 2017.

[3] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. {ZKBoo}: Faster {Zero-Knowledge} for boolean circuits. In *25th usenix security symposium (usenix security 16)*, pages 1069–1083, 2016.

[4] Abida Haque, Stephan Krenn, Daniel Slamanig, and Christoph Striecks. Logarithmic-size (linkable) threshold ring signatures in the plain model. In *IACR International Conference on Public-Key Cryptography*, pages 437–467. Springer, 2022.

[5] Abida Haque and Alessandra Scafuro. Threshold ring signatures: new definitions and post-quantum security. In *Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part II 23*, pages 423–452. Springer, 2020.

[6] Anson Hu, Leo Wang, and Sidharth Menon. Within-group signatures, 2023.

[7] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154. Springer, 1996.

[8] Ronald L Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*, pages 552–565. Springer, 2001.

[9] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II 34*, pages 755–784. Springer, 2015.

[10] Dominique Unruh. Collapse-binding quantum commitments without random oracles. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II 22*, pages 166–195. Springer, 2016.