



Norwegian University of
Science and Technology

A brown squirrel-like animal is perched on the top left of the title text, with its long tail extending downwards and curving to the left.

OLINGO: THRESHOLD LATTICE SIGNATURES WITH DKG AND IDENTIFIABLE ABORTS

NKS 31.10.2025

Kamil Doruk Gür, Patrick Hough, Jonathan Katz,
Caroline Sandsbråten, Tjerand Silde

Threshold Signatures

- ▶ A single private key, distributed among n parties.
- ▶ To create a signature, $t < n$ parties contribute their share of the signature.
- ▶ The signing shares are combined to form a valid signature.
- ▶ The signature can then be verified by the parties single public verification key.

Wanted properties

- ▶ Support many users
- ▶ Few rounds of communication
- ▶ **Distributed key generation** - missing from most schemes
- ▶ **Identifiable aborts** - missing from most schemes

Our Contributions

1. **Efficient sizes:** 2.6 KB public keys, 9.7 KB signatures
2. **3-round protocol:** Only last round depends on message
3. **Distributed Key Generation:** No trusted setup needed
4. **Identifiable Aborts:** Find misbehaving parties
5. **Scalability:** Works for up to 1024 users



High-Level Idea

Starting point: Raccoon signature scheme

- ▶ Signer has secret key s
- ▶ To sign: sample randomness r , create commitment, respond to challenge

Challenge: How to split s among parties?

Our solution:

1. Encrypt the secret key s
2. Share decryption key among n parties
3. Do signature computation in encrypted form
4. Use threshold decryption to get final signature

Verifiable Secret Sharing

Problem: How to ensure shares are distributed correctly?

Our approach:

- ▶ Use Shamir secret sharing to create shares
- ▶ Commit to each share
- ▶ **Novel combination of ZKP's:**
 - ▶ Shamir shares form a codeword in a Reed-Solomon code
 - ▶ Use efficient linear proofs to verify correctness
 - ▶ Prove that shares reconstruct to the original secret

Signing

Round 1: Each party commits to randomness

Round 2:

- ▶ Encrypt randomness
- ▶ Prove the randomness was committed to in Round 1 and correct encryption

Round 3:

- ▶ Validate proofs from other parties
- ▶ Compute challenge from message
- ▶ Compute the encryption of the signatures
- ▶ Decrypt the party's share of the signature
- ▶ Prove share is correct

Combine shares to get signature

State of the Art

| Scheme | PK | SIG | COM | Rounds | Users | DKG | ID-A |
|------------------------|------|------|-------------------|--------|-------|-----|------|
| [GKS24] | 13.6 | 46.6 | 3000 | 0 + 2 | 5 | ✓ | (✓) |
| [DKM ⁺ 24] | 3.9 | 12.7 | 28 | 0 + 3 | 1024 | ✗ | ✗ |
| [EKT24] | 5.5 | 10.8 | 538 | 1 + 1 | 1024 | ✗ | ✗ |
| [EKT24, ZT25] | 8.7 | 30.9 | 767 | 1 + 1 | 1024 | ✗ | ✗ |
| [BKL ⁺ 25] | 4.5 | 13.4 | 629 | 1 + 1 | 1024 | ✗ | ✗ |
| [PN25] | 2.6 | 2.7 | 5.6 | 2 + 1 | 8 | ✗ | ✗ |
| [dPKN ⁺ 25] | 3.9 | 12.7 | 88 + 6.4 <i>t</i> | 0 + 3 | 1024 | ✗ | ✓ |
| Our work | 2.6 | 9.7 | 953 | 2 + 1 | 1024 | ✓ | ✓ |

Timings: Sign

| Scheme | t | Sign $_{S_1}$ | Sign $_{S_2}$ | Sign $_{S_3}$ | Comb | Verify |
|----------|------|---------------|---------------|---------------|--------|--------|
| Olingo | 16 | 2.434 | 2590.972 | 2639.165 | 1.641 | 1.815 |
| Olingo | 64 | 2.140 | 2591.964 | 2647.786 | 2.715 | 1.712 |
| Olingo | 256 | 2.172 | 2592.281 | 2688.725 | 7.517 | 1.898 |
| Olingo | 1023 | 2.191 | 2593.413 | 2834.141 | 26.009 | 1.921 |
| TRaccoon | 16 | 20.076 | 2.102 | 5.559 | 1.209 | 1.093 |
| TRaccoon | 64 | 21.830 | 8.216 | 21.350 | 1.579 | 1.100 |
| TRaccoon | 256 | 33.549 | 32.788 | 84.333 | 3.186 | 1.095 |
| TRaccoon | 1024 | 67.213 | 131.887 | 338.614 | 11.571 | 1.106 |
| Ringtail | 16 | 29.084 | 164.646 | 13.904 | 0.179 | 1.331 |
| Ringtail | 64 | 73.684 | 313.938 | 52.036 | 0.339 | 1.627 |
| Ringtail | 256 | 254.253 | 829.033 | 201.272 | 0.821 | 1.515 |
| Ringtail | 1024 | 978.182 | 2949.853 | 794.288 | 2.706 | 1.434 |

Optimizations

Rounding

- ▶ Drop higher-order bits from public key and commitments
- ▶ Reduces communication at cost of small "hint"

Presigning

- ▶ First two rounds don't depend on message

Optimistic Approach

- ▶ Don't send proofs unless signature verification fails

Summary

Olingo has:

- ▶ ✓ 3 rounds (1 message dependent)
- ▶ ✓ Scales to 1024 users
- ▶ ✓ Distributed key generation
- ▶ ✓ Identifiable aborts

Additional contribution: Efficient verifiable secret sharing using Reed-Solomon codes

Paper: <https://eprint.iacr.org/2025/1789>

Questions?