FROST

draft-irtf-cfrg-frost

Connolly, Komlo, Goldberg, Wood - IETF 114 - CFRG
A Flexible Round-Optimized Schnorr Threshold Signature Scheme
A Flexible Round-Optimized Schnorr Threshold signature scheme
A Flexible Round-Optimized Schnorr Threshold signature scheme
A Flexible Round-Optimized Schnorr Threshold signature scheme

2 rounds (not including keygen)
A Flexible Round-Optimized Schnorr Threshold signature scheme
A Flexible Round-Optimized Schnorr Threshold signature scheme

Only Schnorr, no ECDSA here
A Flexible Round-Optimized Schnorr Threshold signature scheme
A Flexible Round-Optimized Schnorr Threshold signature scheme
A Flexible Round-Optimized Schnorr Threshold Signature Scheme
A Flexible Round-Optimized Schnorr Threshold signature scheme Indistinguishable from single signer
Two-Round FROST Signing Protocol

Keygen is done prior.

**Round 1:** generating nonces & commitments, publishing commitments

**Round 2:** signature share generation & publication

Coordinator aggregates signature shares into the final signature
FROST Overview

== Round 1 (Commitment) ==
   signer commitment
   <---------------------->
   ...                             
   signer commitment
   <---------------------->

== Round 2 (Signature Share Generation) ==
   signer input
   +------------------------>
   signature share
   <---------------------->
   ...                             
   signer input
   +------------------------>
   signature share
   <---------------------->

== Aggregation ==
signature
<---------------------->
0. Key generation and configuration
FROST Overview

0. Key generation and configuration

1. Round 1: nonce and commitment generation
FROST Overview

0. Key generation and configuration

1. Round 1: nonce and commitment generation

2. Round 2: signature share generation and verification
FROST Overview

0. Key generation and configuration

1. Round 1: nonce and commitment generation

2. Round 2: signature share generation and verification

3. Share aggregation and final signature publication
Status

Online signing protocol fully specified and stabilized

Four ciphersuites defined (Ristretto, P-256, Ed25519, Ed448)

- Ed25519 and Ed448 are compatible with RFC8032

5+ interoperable implementations in Rust, C, Python (Sage), multiple ciphersuites
Latest updates

Reverted group commitment optimizations [per analysis](#)

- Optimization led to inter-round signer malleability
- Non-optimized version requires $O(t)$ scalar operations instead of $O(1)$

Verification is a per-ciphersuite routine

- RFC8032-style verification stays in RFC8032
- Verification of signatures over prime-order groups is specified in FROST
Next Steps

Seeking Crypto Panel Review and wider CFRG review, specifically:

- Is the draft clear and unambiguous?
- Is there anything technically incorrect, non-secure, or unsafe in the specification?
- Is the specification written in a way that makes embedding FROST into higher-level application protocols straightforward?

More implementations welcome

Interest in one more ciphersuite (secp256k1)
Questions?

https://github.com/cfrg/draft-irtf-cfrg-frost
draft-irtf-cfrg-frost
We define trusted dealer in the document appendix.

We support distributed key generation (and implement it elsewhere) but do not define it in this document.

(The protocol requires signers to get public keys and private key shares that meet certain requirements, but is agnostic as to the algorithm/protocol that generates them.)
Reverting from FROST 2 to FROST 1

Optimization to make $O(t)$ scalar muls $O(1)$

Attack\(^1\) showed malleability of set of signers between rounds, doesn’t affect signature malleability

Decided to just back it out to $O(t)$ scalar muls to avoid it

\(^1\) https://eprint.iacr.org/2022/833.pdf