FROST: Flexible Round-Optimized Schnorr Threshold Signatures

CFRG Update for Threshold Signature Working Group, IETF 110

Chelsea Komlo^{1,2} Ian Goldberg¹

¹ University of Waterloo ² Zcash Foundation

March 2021

- Two-round Schnorr threshold signing protocol, or single-round with preprocessing.
- FROST specifically trades off robustness for round efficiency.
- Signing operations are secure when performed concurrently, against an adversary that controls up to t – 1 signers.

Key generation be performed by either a trusted dealer or via a Distributed Key Generation (DKG) Protocol

- Two-round Schnorr threshold signing protocol, or single-round with preprocessing.
- FROST specifically trades off robustness for round efficiency.
- Signing operations are secure when performed concurrently, against an adversary that controls up to t – 1 signers.
- Key generation be performed by either a trusted dealer or via a Distributed Key Generation (DKG) Protocol

- Two-round Schnorr threshold signing protocol, or single-round with preprocessing.
- FROST specifically trades off robustness for round efficiency.
- Signing operations are secure when performed concurrently, against an adversary that controls up to t – 1 signers.

Key generation be performed by either a trusted dealer or via a Distributed Key Generation (DKG) Protocol

- Two-round Schnorr threshold signing protocol, or single-round with preprocessing.
- FROST specifically trades off robustness for round efficiency.
- Signing operations are secure when performed concurrently, against an adversary that controls up to t – 1 signers.
- Key generation be performed by either a trusted dealer or via a Distributed Key Generation (DKG) Protocol

- FROST was adopted as a working group item at the end of January.
- We are working on the first draft, focusing on implementation details not specified in our paper.
- We are writing a second proof of security using standard assumptions.
- Several parallel implementations exist, but these need to converge.

- FROST was adopted as a working group item at the end of January.
- We are working on the first draft, focusing on implementation details not specified in our paper.
- We are writing a second proof of security using standard assumptions.
- Several parallel implementations exist, but these need to converge.

- FROST was adopted as a working group item at the end of January.
- We are working on the first draft, focusing on implementation details not specified in our paper.
- We are writing a second proof of security using standard assumptions.

Several parallel implementations exist, but these need to converge.

- FROST was adopted as a working group item at the end of January.
- We are working on the first draft, focusing on implementation details not specified in our paper.
- We are writing a second proof of security using standard assumptions.
- Several parallel implementations exist, but these need to converge.

Compatibility with EdDSA verification over Ed25519/Ed448.

- Using EdDSA-style deterministic nonces is insecure in a multi-signer setting.
- Our draft will specify signatures compatible with verification specified in RFC 8032.
- Preprocessing may be difficult to perform securely.

Compatibility with EdDSA verification over Ed25519/Ed448.

- Using EdDSA-style deterministic nonces is insecure in a multi-signer setting.
- Our draft will specify signatures compatible with verification specified in RFC 8032.
- Preprocessing may be difficult to perform securely.

Compatibility with EdDSA verification over Ed25519/Ed448.

- Using EdDSA-style deterministic nonces is insecure in a multi-signer setting.
- Our draft will specify signatures compatible with verification specified in RFC 8032.

Preprocessing may be difficult to perform securely.

Compatibility with EdDSA verification over Ed25519/Ed448.

- Using EdDSA-style deterministic nonces is insecure in a multi-signer setting.
- Our draft will specify signatures compatible with verification specified in RFC 8032.
- Preprocessing may be difficult to perform securely.

Compatibility with EdDSA verification over Ed25519/Ed448.

- Using EdDSA-style deterministic nonces is insecure in a multi-signer setting.
- Our draft will specify signatures compatible with verification specified in RFC 8032.
- Preprocessing may be difficult to perform securely.
- Option for either trusted dealer or DKG.



Full specification for v1.

Interoperable implementations.

Specify prime-order curves, then adapt to curves with cofactors, considering:

- Point validation during signing.
- Publishing verification-compatible signatures (RFC 8032).



- Full specification for v1.
- Interoperable implementations.
- Specify prime-order curves, then adapt to curves with cofactors, considering:
 - Point validation during signing.
 - Publishing verification-compatible signatures (RFC 8032).



- Full specification for v1.
- Interoperable implementations.

Specify prime-order curves, then adapt to curves with cofactors, considering:

- Point validation during signing.
- Publishing verification-compatible signatures (RFC 8032).



- Full specification for v1.
- Interoperable implementations.
- Specify prime-order curves, then adapt to curves with cofactors, considering:
 - Point validation during signing.
 - Publishing verification-compatible signatures (RFC 8032).



- Full specification for v1.
- Interoperable implementations.
- Specify prime-order curves, then adapt to curves with cofactors, considering:
 - Point validation during signing.
 - Publishing verification-compatible signatures (RFC 8032).

Within the core draft:

- Trusted dealer
- Two round
- Non-signing signature aggregator.

Possible future extensions:

- Preprocessing (single-round signing)
- DKG (possibly more generally useful)
- Share recovery/adding new participants

Within the core draft:

Trusted dealer

Two round

Non-signing signature aggregator.

Possible future extensions:

- Preprocessing (single-round signing)
- DKG (possibly more generally useful)
- Share recovery/adding new participants

Within the core draft:

- Trusted dealer
- Two round
- Non-signing signature aggregator.

Possible future extensions:

- Preprocessing (single-round signing)
- DKG (possibly more generally useful)
- Share recovery/adding new participants

- Within the core draft:
 - Trusted dealer
 - Two round
 - Non-signing signature aggregator.

Possible future extensions:

- Preprocessing (single-round signing)
- DKG (possibly more generally useful)
- Share recovery/adding new participants

- Within the core draft:
 - Trusted dealer
 - Two round
 - Non-signing signature aggregator.
- Possible future extensions:
 - Preprocessing (single-round signing)
 - DKG (possibly more generally useful)
 - Share recovery/adding new participants

- Within the core draft:
 - Trusted dealer
 - Two round
 - Non-signing signature aggregator.
- Possible future extensions:
 - Preprocessing (single-round signing)
 - DKG (possibly more generally useful)
 - Share recovery/adding new participants

- Within the core draft:
 - Trusted dealer
 - Two round
 - Non-signing signature aggregator.
- Possible future extensions:
 - Preprocessing (single-round signing)
 - DKG (possibly more generally useful)
 - Share recovery/adding new participants

- Within the core draft:
 - Trusted dealer
 - Two round
 - Non-signing signature aggregator.
- Possible future extensions:
 - Preprocessing (single-round signing)
 - DKG (possibly more generally useful)
 - Share recovery/adding new participants

- Within the core draft:
 - Trusted dealer
 - Two round
 - Non-signing signature aggregator.
- Possible future extensions:
 - Preprocessing (single-round signing)
 - DKG (possibly more generally useful)
 - Share recovery/adding new participants

Extras: Protocol Specifics

FROST Sign: Round One

Participant i

$$egin{aligned} (d_i, e_i) & \stackrel{\$}{\leftarrow} \mathbb{Z}_q^* imes \mathbb{Z}_q^* \ (D_i, E_i) &= (g^{d_i}, g^{e_i}) \end{aligned}$$

Signature Aggregator

FROST Sign: Round One

Participant i

$$egin{aligned} & (d_i, e_i) \stackrel{\$}{\leftarrow} \mathbb{Z}_q^* imes \mathbb{Z}_q^* \ & (D_i, E_i) = (g^{d_i}, g^{e_i}) \end{aligned}$$

Signature Aggregator

FROST Sign: Round One

Participant i

$$egin{aligned} (d_i, e_i) & \stackrel{\$}{\leftarrow} \mathbb{Z}_q^* imes \mathbb{Z}_q^* \ (D_i, E_i) &= (g^{d_i}, g^{e_i}) \end{aligned}$$

Signature Aggregator

 (D_i, E_i)

Signer i

Signature Aggregator $B = ((1, D_1, E_1), \dots, (t, D_t, E_t))$

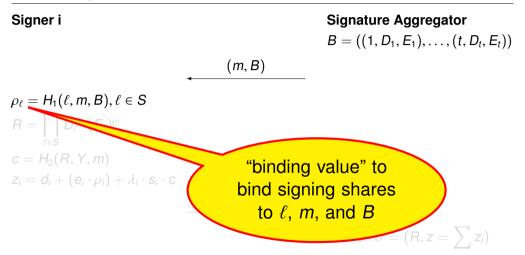


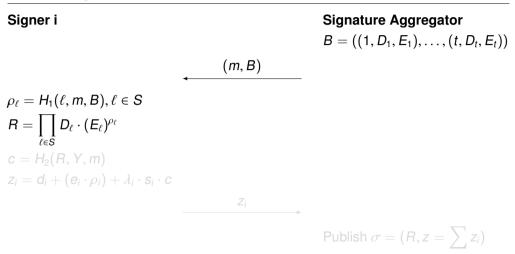


Signature Aggregator $B = ((1, D_1, E_1), \dots, (t, D_t, E_t))$



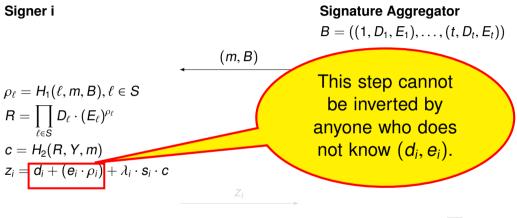
(m, B)





Signer i Signature Aggregator $B = ((1, D_1, E_1), \dots, (t, D_t, E_t))$ (m, B) $\rho_{\ell} = H_1(\ell, m, B), \ell \in S$ ${\pmb R} = \prod_\ell {\pmb D}_\ell \cdot ({\pmb E}_\ell)^{
ho_\ell}$ $c = H_2(R, Y, m)$

Signer i Signature Aggregator $B = ((1, D_1, E_1), \dots, (t, D_t, E_t))$ (m, B) $\rho_{\ell} = H_1(\ell, m, B), \ell \in S$ ${\pmb R} = \prod_\ell {\pmb D}_\ell \cdot ({\pmb E}_\ell)^{
ho_\ell}$ $c = H_2(R, Y, m)$ $z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$



Publish $\sigma = (R, z = \sum z_i)$

Signer i Signature Aggregator $B = ((1, D_1, E_1), \dots, (t, D_t, E_t))$ (m, B) $\rho_{\ell} = H_1(\ell, m, B), \ell \in S$ ${\pmb R} = \prod_\ell {\pmb D}_\ell \cdot ({\pmb E}_\ell)^{
ho_\ell}$ $c = H_2(R, Y, m)$ $z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$ Zi

Signer i Signature Aggregator $B = ((1, D_1, E_1), \dots, (t, D_t, E_t))$ (m, B) $\rho_{\ell} = H_1(\ell, m, B), \ell \in S$ $R = \prod_{\ell \in I} D_\ell \cdot (E_\ell)^{
ho_\ell}$ $c = H_2(R, Y, m)$ $z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$ Zi Publish $\sigma = (R, z = \sum z_i)$

