Schnorr Threshold Signatures

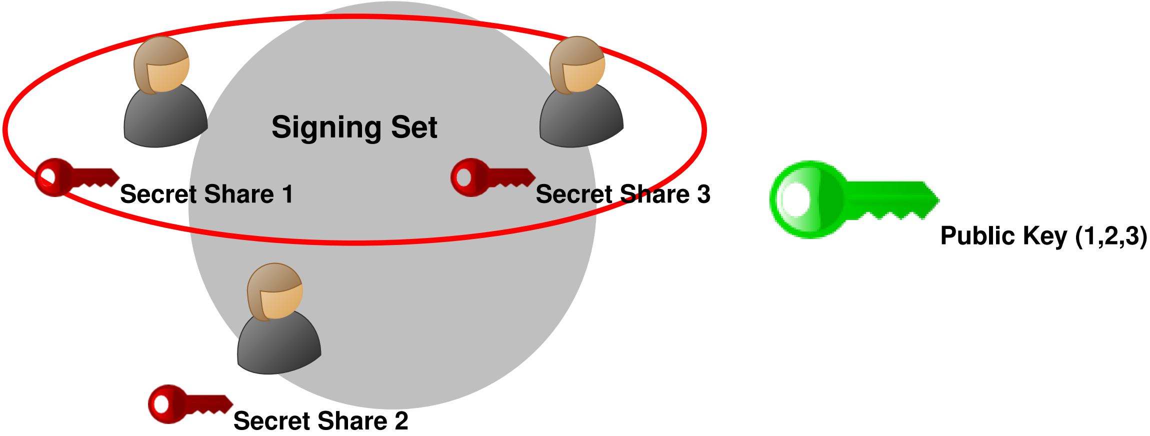
An Overview of the Current Landscape and Next Steps

Chelsea Komlo

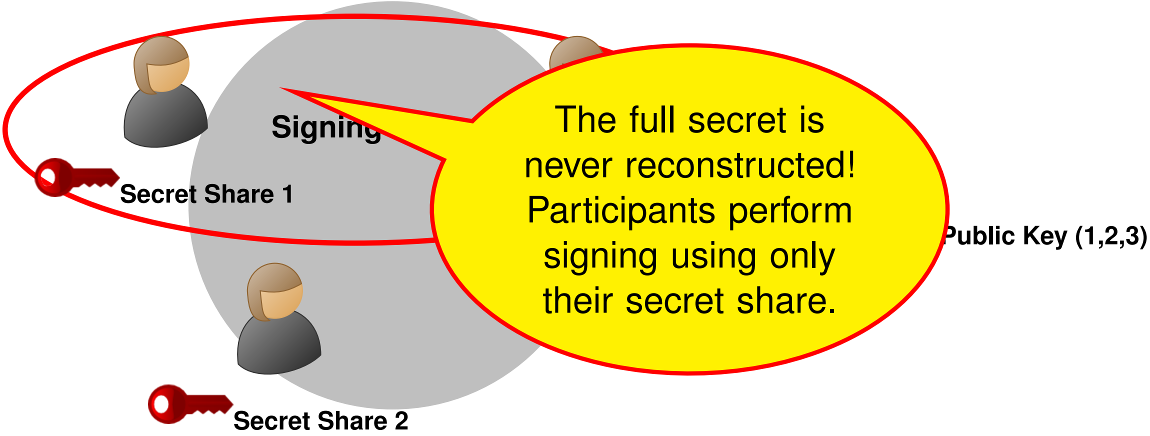
University of Waterloo

Microsoft Privacy and Cryptography Group, August 2021

Threshold Signatures: Joint Public Key, Secret-Shared Private Key



Threshold Signatures: Joint Public Key, Secret-Shared Private Key



Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

$$\xleftarrow{(m, Y)}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{(m, \sigma = (R, z))}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

(m, Y) ←

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$(m, \sigma = (R, z))$ →

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

$$\xleftarrow{(m, Y)}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{(m, \sigma = (R, z))}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer | **Verifier**

$(x, Y) \leftarrow KeyGen()$

$$\xleftarrow{\quad (m, Y) \quad}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{\quad (m, \sigma = (R, z)) \quad}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

$$\xleftarrow{(m, Y)}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{(m, \sigma = (R, z))}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer | **Verifier**

$(x, Y) \leftarrow KeyGen()$

$$\xleftarrow{\quad (m, Y) \quad}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{\quad (m, \sigma = (R, z)) \quad}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

$$\xleftarrow{(m, Y)}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{(m, \sigma = (R, z))}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

$$\xleftarrow{(m, Y)}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{(m, \sigma = (R, z))}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer

$(x, Y) \leftarrow KeyGen()$

Verifier

$$\xleftarrow{\quad (m, Y) \quad}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{\quad (m, \sigma = (R, z)) \quad}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Single-Party Schnorr Signing and Verification

Signer | **Verifier**

$(x, Y) \leftarrow KeyGen()$

$$\xleftarrow{\quad (m, Y) \quad}$$

$k \xleftarrow{\$} \mathbb{Z}_q$

$R = g^k \in \mathbb{G}$

$c = H(R, Y, m)$

$z = k + c \cdot x$

$$\xrightarrow{\quad (m, \sigma = (R, z)) \quad}$$

$c = H(R, Y, m)$

$R' = g^z \cdot Y^{-c}$

Output $R \stackrel{?}{=} R'$

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$$R = \prod_{\ell \in S} D_\ell$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

Warm up Example of Threshold Signatures

Signer i

Signature Aggregator

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$D_i = g^{d_i}$$

$$(m, B = ((1, D_1), \ldots, (t, D_t)))$$

$$R = \prod_{\ell \in S} D_\ell$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + \lambda_i \cdot s_i \cdot c$$

$$z_i$$

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

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$R = \prod_{\ell \in S} D_\ell$

$c = H_2(R, Y, m)$

$z_i = d_i + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$$R = \prod_{\ell \in S} D_\ell$$

$c = H_2(R, Y, m)$

$z_i = d_i + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$$R = \prod_{\ell \in S} D_\ell$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$$R = \prod_{\ell \in S} D_\ell$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$$R = \prod_{\ell \in S} D_\ell$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

Warm up Example of Threshold Signatures

Signer i | **Signature Aggregator**

$d_i \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{D_i = g^{d_i}}$$

$$\xleftarrow{(m, B = ((1, D_1), \ldots, (t, D_t)))}$$

$$R = \prod_{\ell \in S} D_\ell$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

Drijver's Attack

- Forgery attack on two-round multisignatures/threshold signatures.
- Requires an active attacker (a signer participating in the protocol).
- Relies on the Wagner-Fischer algorithm for finding a collision between hash function outputs.
- Difficult to find $H(x) = H(y)$.
- But finding an x such that $H(x) = H(w) + H(y) + H(z) + \ldots$ for some (w, y, z) is possible in polynomial time.

Drijver's Attack

- Forgery attack on two-round multisignatures/threshold signatures.
- Requires an active attacker (a signer participating in the protocol).
- Relies on the Wagner-Fischer algorithm for finding a collision between hash function outputs.
- Difficult to find $H(x) = H(y)$.
- But finding an x such that $H(x) = H(w) + H(y) + H(z) + \ldots$ for some (w, y, z) is possible in polynomial time.

Drijver's Attack

- Forgery attack on two-round multisignatures/threshold signatures.
- Requires an active attacker (a signer participating in the protocol).
- Relies on the Wagner-Fischer algorithm for finding a collision between hash function outputs.
- Difficult to find $H(x) = H(y)$.
- But finding an x such that $H(x) = H(w) + H(y) + H(z) + \ldots$ for some (w, y, z) is possible in polynomial time.

Drijver's Attack

- Forgery attack on two-round multisignatures/threshold signatures.
- Requires an active attacker (a signer participating in the protocol).
- Relies on the Wagner-Fischer algorithm for finding a collision between hash function outputs.
- Difficult to find $H(x) = H(y)$.
- But finding an x such that $H(x) = H(w) + H(y) + H(z) + \ldots$ for some (w, y, z) is possible in polynomial time.

Drijver's Attack

- Forgery attack on two-round multisignatures/threshold signatures.
- Requires an active attacker (a signer participating in the protocol).
- Relies on the Wagner-Fischer algorithm for finding a collision between hash function outputs.
- Difficult to find $H(x) = H(y)$.
- But finding an x such that $H(x) = H(w) + H(y) + H(z) + \ldots$ for some (w, y, z) is possible in polynomial time.

Drijver's Attack against the Warm Up Example

An adversary can query an individual signer, producing different D_A, m_A terms for themselves (therefore varying R, c) each time.

Eventually, an adversary could produce a c^* such that:

$$c^* = H(R^*, Y, m^*) = \sum_{i=1}^{k} H(R_i, Y, m_i) = \sum c_i \text{ for some } (R_i, m_i), \ldots$$

,

After sending receiving the victim's z_i for each (R_i, m_i), the adversary can produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + \lambda_t \cdot s_t \cdot c^*$$

Drijver's Attack against the Warm Up Example

An adversary can query an individual signer, producing different D_A, m_A terms for themselves (therefore varying R, c) each time.

Eventually, an adversary could produce a c^* such that:

$$c^* = H(R^*, Y, m^*) = \sum_{i=1}^{k} H(R_i, Y, m_i) = \sum c_i \text{ for some } (R_i, m_i), \ldots$$

,

After sending receiving the victim's z_i for each (R_i, m_i), the adversary can produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + \lambda_t \cdot s_t \cdot c^*$$

Drijver's Attack against the Warm Up Example

An adversary can query an individual signer, producing different D_A, m_A terms for themselves (therefore varying R, c) each time.

Eventually, an adversary could produce a c^* such that:

$$c^* = H(R^*, Y, m^*) = \sum_{i=1}^{k} H(R_i, Y, m_i) = \sum c_i \text{ for some } (R_i, m_i), \ldots$$

,

After sending receiving the victim's z_i for each (R_i, m_i), the adversary can produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + \lambda_t \cdot s_t \cdot c^*$$

Drijver's Attack against the Warm Up Example

An adversary can query an individual signer, producing different D_A, m_A terms for themselves (therefore varying R, c) each time.

Eventually, an adversary could produce a c^* such that:

$$c^* = H(R^*, Y, m^*) = \sum_{i=1}^{k} H(R_i, Y, m_i) = \sum c_i \text{ for some } (R_i, m_i), \ldots$$

,

After sending receiving the victim's z_i for each (R_i, m_i), the adversary can produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + \lambda_t \cdot s_t \cdot c^*$$

Drijver's Attack against the Warm Up Example

An adversary can query an individual signer, producing different D_A, m_A terms for themselves (therefore varying R, c) each time.

Eventually, an adversary could produce a c^* such that:

$$c^* = H(R^*, Y, m^*) = \sum_{i=1}^{k} H(R_i, Y, m_i) = \sum c_i \text{ for some } (R_i, m_i), \ldots$$

,

After sending receiving the victim's z_i for each (R_i, m_i), the adversary can produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + \lambda_t \cdot s_t \cdot c^*$$

Tradeoffs Among Constructions

- **Number of Signing Rounds:** Required network rounds to generate one signature.
- **Robust:** Can the protocol complete when participants misbehave?
- **Required Number of Signers:** Can a signature be created by just t participants, or are all n needed?
- **Parallel Secure:** Can signing operations be done in parallel without a reduction in security (Drijvers attack)?

Tradeoffs Among Constructions

- **Number of Signing Rounds:** Required network rounds to generate one signature.
- **Robust:** Can the protocol complete when participants misbehave?
- **Required Number of Signers:** Can a signature be created by just t participants, or are all n needed?
- **Parallel Secure:** Can signing operations be done in parallel without a reduction in security (Drijvers attack)?

Tradeoffs Among Constructions

- **Number of Signing Rounds:** Required network rounds to generate one signature.
- **Robust:** Can the protocol complete when participants misbehave?
- **Required Number of Signers:** Can a signature be created by just t participants, or are all n needed?
- **Parallel Secure:** Can signing operations be done in parallel without a reduction in security (Drijvers attack)?

Tradeoffs Among Constructions

- **Number of Signing Rounds:** Required network rounds to generate one signature.
- **Robust:** Can the protocol complete when participants misbehave?
- **Required Number of Signers:** Can a signature be created by just t participants, or are all n needed?
- **Parallel Secure:** Can signing operations be done in parallel without a reduction in security (Drijvers attack)?

Tradeoffs Among Constructions

	Num. Rounds	**Robust**	**Num. Signers**	**Parallel Secure**
Stinson Strobl	4	Yes	t	Yes
Gennaro et al.	1 w/ preprocessing	No	n	No
FROST	1 w/ preprocessing	No	t	Yes

Contributions of FROST

Flexible Round-Optimized Schnorr Threshold Signatures

- Two-round threshold signing protocol, or single-round protocol with preprocessing
- Secure against the Drijvers attack, for an adversary controlling up to $t - 1$ signers.
- Signing can be performed with a threshold t number of signers, where t can be less than the number of possible signers n.

Contributions of FROST

Flexible Round-Optimized Schnorr Threshold Signatures

- Two-round threshold signing protocol, or single-round protocol with preprocessing
- Secure against the Drijvers attack, for an adversary controlling up to $t - 1$ signers.
- Signing can be performed with a threshold t number of signers, where t can be less than the number of possible signers n.

Contributions of FROST

Flexible Round-Optimized Schnorr Threshold Signatures

- Two-round threshold signing protocol, or single-round protocol with preprocessing
- Secure against the Drijvers attack, for an adversary controlling up to $t - 1$ signers.
- Signing can be performed with a threshold t number of signers, where t can be less than the number of possible signers n.

FROST Sign

Signer i **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i | **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i | **Signature Aggregator**

$$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$$\rho_\ell = H_1(\ell, m, B), \ell \in S$$

$$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i | **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

“binding value” to bind signing shares to ℓ, m, and B

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

FROST Sign

Signer i | **Signature Aggregator**

$$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$$\rho_\ell = H_1(\ell, m, B), \ell \in S$$

$$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$$

$$c = H_2(R, Y, m)$$

$$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i

Signature Aggregator

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i | **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$

$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots}$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = \boxed{d_i + (e_i \cdot \rho_i)} + \lambda_i \cdot s_i \cdot c$

This step cannot be inverted by anyone who does not know (d_i, e_i).

$\xrightarrow{z_i}$

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

FROST Sign

Signer i **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$$

$$\xleftarrow{(m, B = ((1, D_1, E_1), \ldots, (t, D_t, E_t)))}$$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$$\xrightarrow{z_i}$$

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

FROST Sign

Signer i | **Signature Aggregator**

$(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$

$\xrightarrow{(D_i = g^{d_i}, E_i = g^{e_i})}$

$\xleftarrow{(m, B = ((1, D_1, E_1), .}$

$\rho_\ell = H_1(\ell, m, B), \ell \in S$

$R = \prod_{\ell \in S} D_\ell \cdot (E_\ell)^{\rho_\ell}$

$c = H_2(R, Y, m)$

$z_i = d_i + (e_i \cdot \rho_i) + \lambda_i \cdot s_i \cdot c$

$\xrightarrow{z_i}$

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

Signature format
and verification
are identical to
single-party Schnorr.

Security against Drijvers

Without $\rho_\ell = H_1(\ell, m, B)$, an adversary could produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + e_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + e_i + \lambda_t \cdot s_t \cdot c^*$$

$$R^* = g^{\sum d_i + e_i + s \cdot \sum c_i} \cdot g^{-sc^*}$$

The binding factor in FROST makes each z_i strongly tied to (m_i, R_i).

$$z = \sum d_i + (e_i * \rho_i) + \lambda_t \cdot s_t \cdot \sum c_i$$

Resulting in an invalid signature:

$$R^* \neq g^z \cdot Y^{-c}$$

$$R^* \neq g^{\sum d_i + (e_i * H(m, (D_i, E_i, i), \ldots)) + s \cdot \sum c_i} \cdot g^{-sc^*}$$

Security against Drijvers

Without $\rho_\ell = H_1(\ell, m, B)$, an adversary could produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + e_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + e_i + \lambda_t \cdot s_t \cdot c^*$$

$$R^* = g^{\sum d_i + e_i + s \cdot \sum c_i} \cdot g^{-sc^*}$$

The binding factor in FROST makes each z_i strongly tied to (m_i, R_i).

$$z = \sum d_i + (e_i * \rho_i) + \lambda_t \cdot s_t \cdot \sum c_i$$

Resulting in an invalid signature:

$$R^* \neq g^z \cdot Y^{-c}$$

$$R^* \neq g^{\sum d_i + (e_i * H(m,(D_i,E_i,i),\ldots)) + s \cdot \sum c_i} \cdot g^{-sc^*}$$

Security against Drijvers

Without $\rho_\ell = H_1(\ell, m, B)$, an adversary could produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + e_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + e_i + \lambda_t \cdot s_t \cdot c^*$$

$$R^* = g^{\sum d_i + e_i + s \cdot \sum c_i} \cdot g^{-sc^*}$$

The binding factor in FROST makes each z_i strongly tied to (m_i, R_i).

$$z = \sum d_i + (e_i * \rho_i) + \lambda_t \cdot s_t \cdot \sum c_i$$

Resulting in an invalid signature:

$$R^* \neq g^z \cdot Y^{-c}$$

$$R^* \neq g^{\sum d_i + (e_i * H(m, (D_i, E_i, i), \ldots)) + s \cdot \sum c_i} \cdot g^{-sc^*}$$

Security against Drijvers

Without $\rho_\ell = H_1(\ell, m, B)$, an adversary could produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + e_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + e_i + \lambda_t \cdot s_t \cdot c^*$$

$$R^* = g^{\sum d_i + e_i + s \cdot \sum c_i} \cdot g^{-sc^*}$$

The binding factor in FROST makes each z_i strongly tied to (m_i, R_i).

$$z = \sum d_i + (e_i * \rho_i) + \lambda_t \cdot s_t \cdot \sum c_i$$

Resulting in an invalid signature:

$$R^* \neq g^z \cdot Y^{-c}$$

$$R^* \neq g^{\sum d_i + (e_i * H(m, (D_i, E_i, i), \ldots)) + s \cdot \sum c_i} \cdot g^{-sc^*}$$

Security against Drijvers

Without $\rho_\ell = H_1(\ell, m, B)$, an adversary could produce a valid forgery $\sigma^* = (R^*, z)$, as

$$z = \sum d_i + e_i + \lambda_t \cdot s_t \cdot \sum c_i = \sum d_i + e_i + \lambda_t \cdot s_t \cdot c^*$$

$$R^* = g^{\sum d_i + e_i + s \cdot \sum c_i} \cdot g^{-sc^*}$$

The binding factor in FROST makes each z_i strongly tied to (m_i, R_i).

$$z = \sum d_i + (e_i * \rho_i) + \lambda_t \cdot s_t \cdot \sum c_i$$

Resulting in an invalid signature:

$$R^* \neq g^z \cdot Y^{-c}$$

$$R^* \neq g^{\sum d_i + (e_i * H(m, (D_i, E_i, i), \ldots)) + s \cdot \sum c_i} \cdot g^{-sc^*}$$

Implementation Requirements

- KeyGen requires a trusted, authenticated channel for distributing secret shares.
- Signing can be performed over a trustless public channel as all values exchanged during signing are public.
- Use of some underlying PKI is required for proving attribution of misbehaviour to a specific signer.

Implementation Requirements

- KeyGen requires a trusted, authenticated channel for distributing secret shares.
- Signing can be performed over a trustless public channel as all values exchanged during signing are public.
- Use of some underlying PKI is required for proving attribution of misbehaviour to a specific signer.

Implementation Requirements

- KeyGen requires a trusted, authenticated channel for distributing secret shares.
- Signing can be performed over a trustless public channel as all values exchanged during signing are public.
- Use of some underlying PKI is required for proving attribution of misbehaviour to a specific signer.

Takeaways

- FROST improves upon prior schemes by defining a single-round threshold signing protocol (with preprocessing) that is secure even when signing is performed concurrently.

- The simplicity and flexibility of FROST makes it attractive to real-world applications.

Find our paper and artifact at https://crysp.uwaterloo.ca/software/frost.

Takeaways

- FROST improves upon prior schemes by defining a single-round threshold signing protocol (with preprocessing) that is secure even when signing is performed concurrently.
- The simplicity and flexibility of FROST makes it attractive to real-world applications.

Find our paper and artifact at https://crysp.uwaterloo.ca/software/frost.