Dynamic-FROST: Schnorr Threshold Signatures with a Flexible Committee

A. Cimatti, F. Desclavis, G. Galano, S. Giammusso, M. Iezzi, A. Muci, M. Nardelli, M. Pedicini

May 3, 2025

ROMA TRE UNIVERSITÀ DEGLI STUDI

BANCA D'ITALIA

Threshold signatures

A **threshold signature scheme** allows any subgroup of t signers out of n participants to generate a signature which cannot be forged by any subgroup with fewer than t members.

Threshold signatures

Advantages:

- **scalability**: the length of the aggregated signature does not increase with the number of signers;
- **confidentiality**: the identity of actual signers remains confidential.

Problem

- In many threshold signatures, the threshold t and the number of participants n are fixed.
- It might be useful to increase or decrease these values:
  - *self-custodial cryptocurrency wallets* might require changes to the set of signers without moving funds to a new address, i.e., without modifying the group public key through a blockchain transaction.

Goal: change t and/or n without changing the secret s.

Contribution

- **Dynamic-FROST** (D-FROST) is the first Schnorr threshold signature scheme to support a flexible committee
- D-FROST is the result of merging FROST with CHURP
- D-FROST is EUF-CMA secure

Schnorr threshold signatures

- **FROST** is a Schnorr threshold signature scheme with many desirable properties:
  - decentralization;
  - efficiency;
  - number of actual signers hidden.

FROST

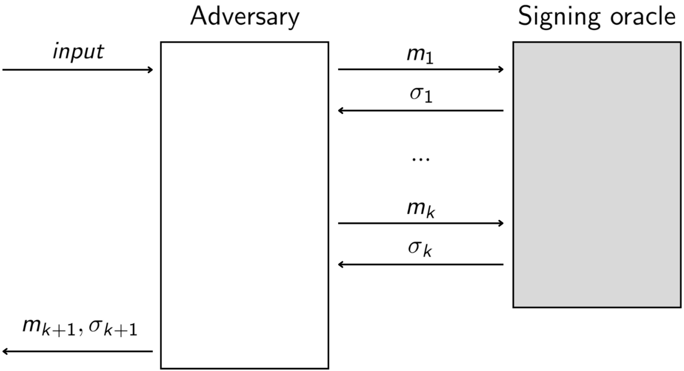
- In FROST, each participant has the same power, except for the **signature aggregator** (SA).
- SA is a semi-trusted node that has the ability to publish the group signature at the end of the protocol.

FROST

FROST is made of the following schemes:

- **KeyGen:** Distributed Key Generation (DKG) scheme based on Shamir's secret sharing;
- **Preprocess(π):** participants create a list of nonces that will be used during the signing phase;
- **Sign(m):** participants sign the message m with a Schnorr threshold signature.

EUF-CMA security



Security of FROST

The protocol is EUF-CMA secure against an adversary that corrupts at most $t-1$ nodes under the DL assumption in the random oracle model.

Proactive secret sharing

A **proactive secret sharing scheme** enables users to change the secret shares without changing the secret.

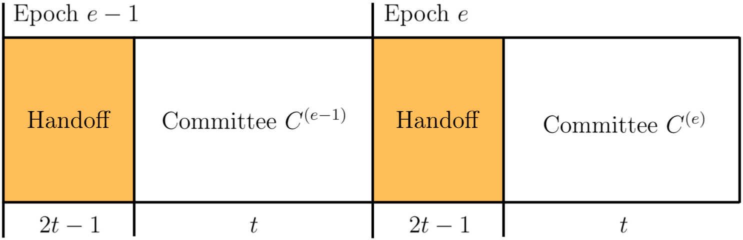
Dynamic proactive secret sharing

A **dynamic proactive secret sharing scheme** (**DPSS**) is a proactive secret sharing scheme that involves a dynamic committee.

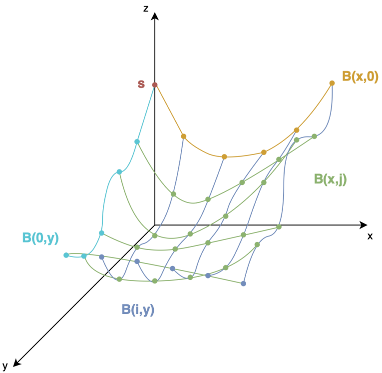
CHURP

- **CHURP** is a DPSS that uses a bivariate polynomial $B(x, y)$ to share the secret s.
- $B(x, y)$ has degree $\langle t-1, 2t-2\rangle$ and is such that $B(0, 0) = s$.

CHURP


Epoch $e-1$
Epoch e
Handoff
Committee $C^{(e-1)}$
Handoff
Committee $C^{(e)}$
$2t-1$
t
$2t-1$
t

CHURP



CHURP

It is composed by three subprotocols:

- Opt-CHURP (optimistic);
- Exp-CHURP-A (pessimistic);
- Exp-CHURP-B (pessimistic).

In the pessimistic paths, communication is on-chain only. To send messages peer-to-peer, participants encrypt the message using receiver's public key before publishing it on-chain.

CHURP

- We only consider Opt-CHURP and Exp-CHURP-A.
- Both these paths use the KZG scheme, which provides polynomial commitments and witnesses.

Steady state

To enter the handoff phase, the system must be in a **steady state**:

- the committee has all the necessary data to perform the KZG scheme;
- each P_i holds a $(2t-2)$-degree polynomial $B(i, y)$ such that $s_i = B(i, 0)$ is a (t, n)-share of $s = B(0, 0)$.

Handoff

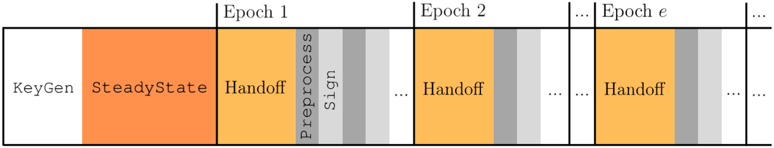
- During this phase, the shares are proactivized using a 0-hole random polynomial $Q(x, y)$, with $deg_Q = deg_B$.
- The new polynomial $B'(x, y) = B(x, y) + Q(x, y)$ is such that $B'(0, 0) = s$ and $deg_{B'} = \langle t - 1, 2t - 2 \rangle$.

Security of CHURP

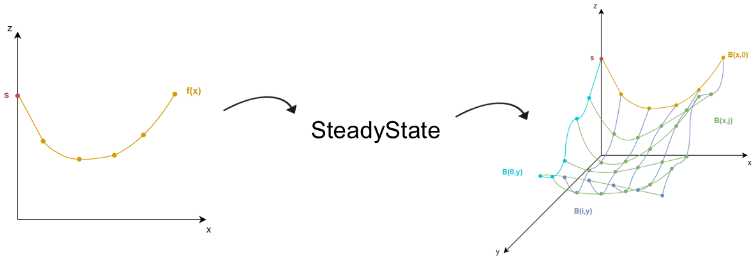
CHURP satisfies the following properties:

- **secrecy:** if an adversary A corrupts no more than $t-1$ nodes in a committee of any epoch, A learns no information about the secret s.
- **integrity:** if A corrupts no more than $t-1$ nodes in each of the committees $C^{(e-1)}$ and $C^{(e)}$, after the handoff, the shares for honest nodes can be correctly computed and the secret s remains intact.

Dynamic-FROST



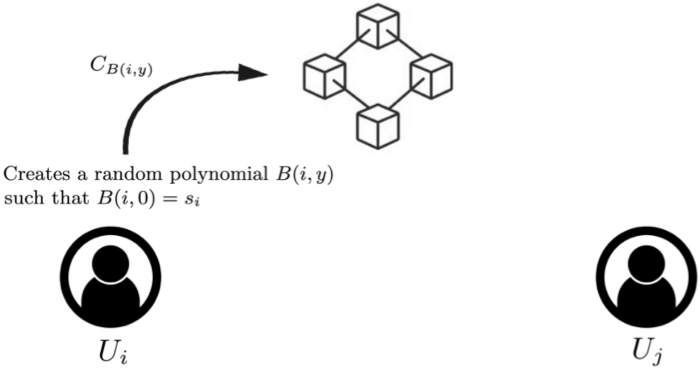
SteadyState



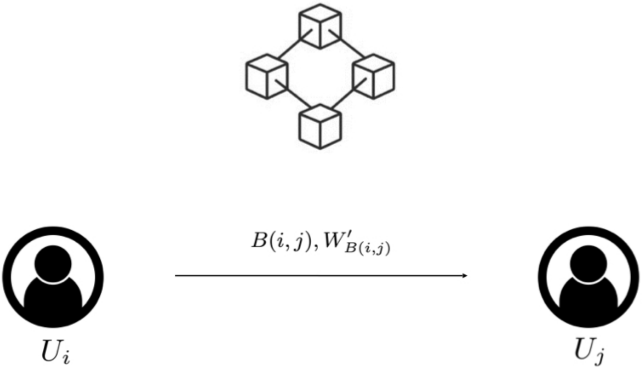
SteadyState

Choose $2t-1$ nodes in $C = \{P_i\}_{i=1}^{n}$, denoted as $\mathcal{U} = \{U_j\}_{j\in[2t-1]}$. Let U_i, $i \in [t]$, be the first t nodes in $\mathcal{U}$.

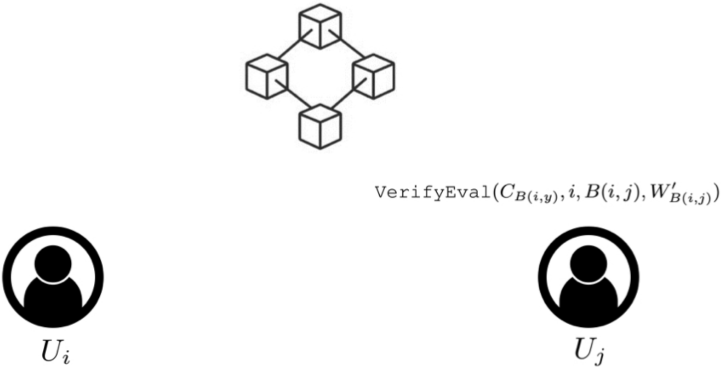
SteadyState



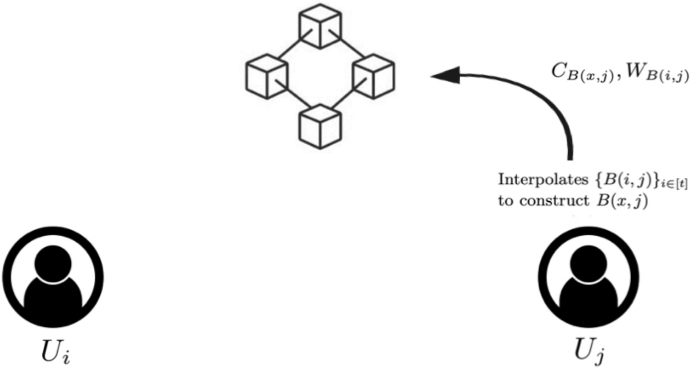
SteadyState



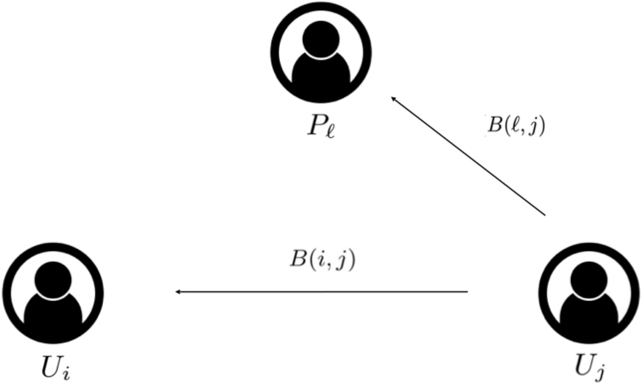
SteadyState



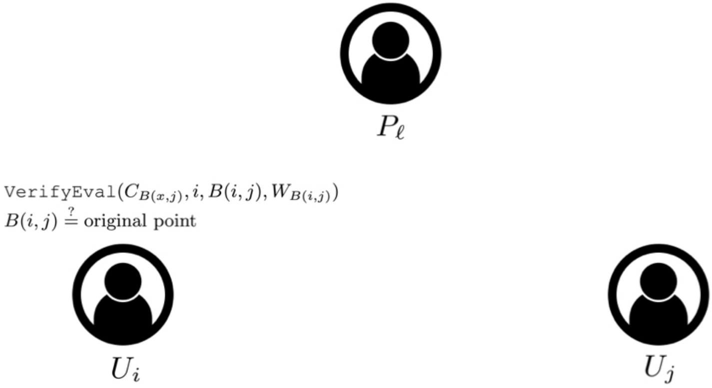
SteadyState



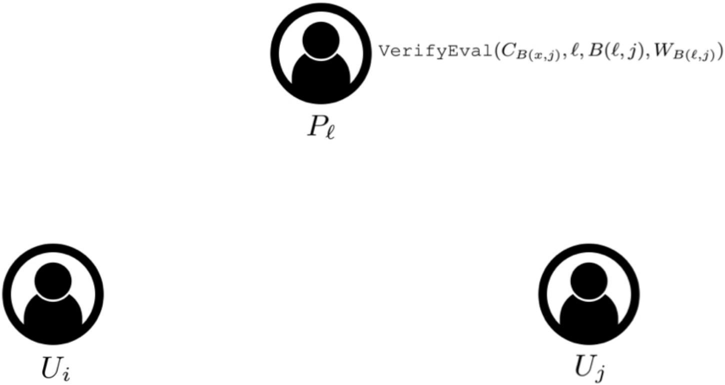
SteadyState



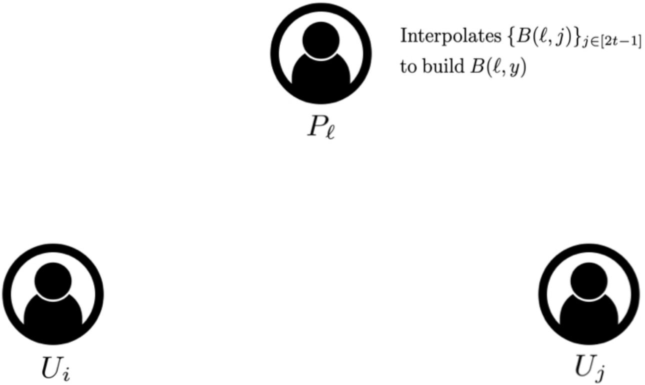
SteadyState



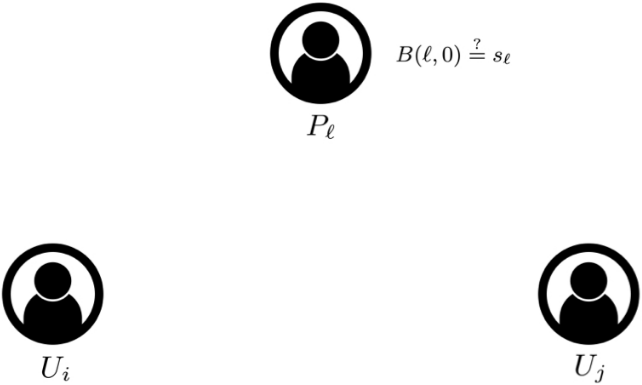
SteadyState



SteadyState



SteadyState



Security of D-FROST

Goal: prove that D-FROST is EUF-CMA secure in the random oracle model.

Security of SteadyState

We prove that the following properties are satisfied:

- **secrecy:** an adversary corrupting a set of at most $t-1$ parties cannot learn anything about the secret s;
- **integrity:** it must hold that $B(0,0)=s$.

Security in each epoch

Theorem

If the property of secrecy in CHURP holds, then D-FROST is EUF-CMA secure against an active adversary that corrupts no more than $t - 1$ nodes during an arbitrary epoch.

Security of D-FROST

- Secrecy and integrity hold throughout the protocol.
- The shares in one epoch are independent of the old ones, so the adversary does not obtain any additional data by putting together information learned during different epochs.
- D-FROST signatures are EUF-CMA secure.

Thank you for your attention!

The full paper can be found here: https://www.degruyterbrill.com/document/doi/10.1515/jmc-2024-0045/html.