FROST Federation Progress

Kulpreet Singh
kp@opdup.com

Contents

1. **What and why of FROST Federation**
2. Progress integrating FROST KeyGen and Signing
3. Challenges ahead
4. Next steps

A quick overview of FROST Federation

1. Networked, always online federation of nodes that need TSS

2. FROST usage is largely motivated for wallets

3. We also need a TSS scheme for Braidpool and Radpool

Membership Decided by PoW

1. Membership is decided out of band using Proof of Work

2. Each party is serving miners

3. Aggregate hashrate contributed by each party used to decide membership

PoW Helps Keep Members Honest

1. Misbehaving parties are left out
2. They can sybil attack - in two weeks time
3. Current DKG proceeds without misbehaving party

Contents

1. What and why of FROST Federation
2. **Progress integrating FROST KeyGen and Signing**
3. Challenges ahead
4. Next steps

Progress: Great Docs with No Surprises

1. Found the docs to be excellent. Thank you!

2. DKG example worked once the network was in place

3. Caveats are sufficiently well highlighted

Progress: KeyGen Using Bracha’s Broadcast

1. To prevent equivocation - we use Bracha’s broadcast

2. This can be optimised later, if we want to using schemes like AVID BFT Broadcast

Implementation Progress in Rust

1. Rust implementation progressing at https://github.com/pool2win/frost-federation
2. DKG working over Network Stack using Tokio and Tower Services

Next Steps (DKG): Identify Misbehaving Parties

1. Currently we only handle network failures

2. We need to handle identified culprits from round one and round two packages.

Progress: Signing

Not yet started, focussed entirely on DKG for the moment.

Signing “seems” much simpler as compared to DKG.

Contents

1. What and why of FROST Federation
2. Progress integrating FROST KeyGen and Signing
3. **Challenges ahead**
4. Next steps

Challenge: Signing

1. Robustness using one of ROAST optimisation schemes.

2. We have a decent network stack in place to handle this challenge.

3. Still, not to be underestimated

Aside: Lindell’s Schnorr Signature Scheme

1. Considered implementing it.

2. Decided to stick with FROST and make ROAST optimisation instead.

Next Steps

1. Make KeyGen robust for our use case of PoW backed membership.

2. Support signing using an optimisation of ROAST.

End

Thanks

kp@opdup.com