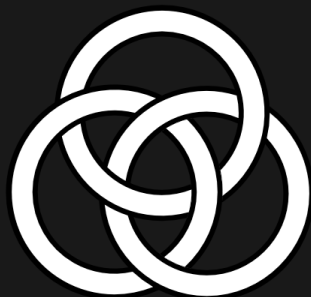




# Blockchain Commons

*Advocating for the Creation of Open, Interoperable,*



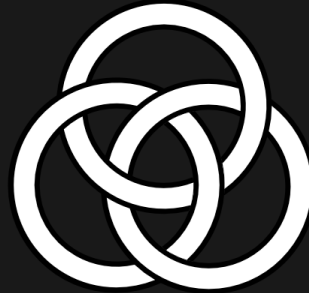
## What is Blockchain Commons?

- We are a community interested in self-sovereign control of digital assets.
- We bring together stakeholders to collaboratively develop interoperable infrastructure.
- We design decentralized solutions where everyone wins.
- We are a neutral "not-for-profit" that enables people to control their own digital destiny.

Thank you to our Sustaining Sponsors!



[github.com/sponsors/BlockchainCommons](https://github.com/sponsors/BlockchainCommons)



## Last Meeting

- Request & Response using Envelope
  - Implementation Guide (2024-04)
  - A Use Case: How It Improves Multisig
- Gordian Server 1.1.0



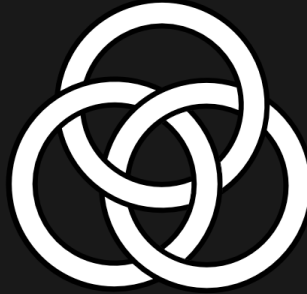
## Today's Topics

- Gordian Advancements
- Wyoming Legislation
- FROST!
- Next Time!



## Gordian Advances

- SSH Use Case
- New Python `ssh_envelope` CLI
  - Uses Rust `envelope` & `ssh-keygen` CLI tools
  - Import/Export SSH Keys & Signatures
  - Generate Private Ed25519 SSH Keys
  - Extract Public Keys from Private Keys
  - Sign Envelopes using SSH Private Keys
  - Verify Envelope signatures using SSH Public Keys
- Rust Stack updated
  - dCBOR now supports `no_std` environment
  - Reads version 1 and 2 tags, writes version 2 tags
  - Streamlined Envelope Rust API
- Swift Stack update in progress



## Gordian Server 1.1.0

- Gordian Server 1.1.0 Has Been Released
- Why Gordian Server?
  - It's a part of our TorGap ecosystem to support partitioning
  - It supports privacy and resists censorship
  - But like all of our apps, it's a reference: how we think things should work
- New 1.1.0 Version
  - RPCAuth Instead of plain text RPC credentials
  - Properly Supports M1/M2 binaries
  - Properly Supports Bitcoin 25/26
- Take a Look at @Fontaine's Fully Noded App for an Integrated Wallet



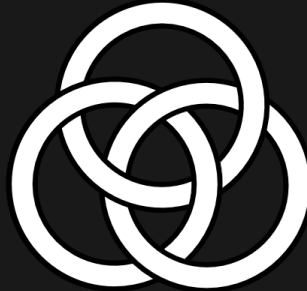


# Wyoming Legislation

- Passed recently:
  - Private Key Disclosure
  - Wyoming Registered Digital Asset
  - DAO LLC and Unincorporated DAO
    - Bitcoin friendly!

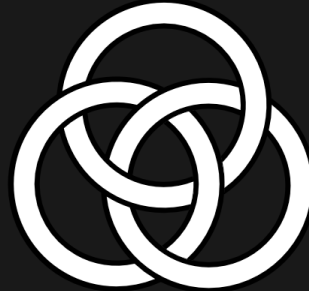
Coming up:

- Micro-DAO Series LLC (bitcoin descriptors?)
- Legalize data minimization through redaction & elision
- Much more, see <https://advocacy.blockchaincommons.com>



## Welcome to Jesse Posner

- Jesse Posner
- Senior Blockchain Engineer, Bitkey
- Working on FROST
  - secp256k1-zkp implementation
  - FROST BIPs
  - research projects



## What is FROST?

- Flexible Round-Optimized Schnorr Threshold Signatures (FROST)
- Uses Schnorr Signatures, added to Bitcoin with the Taproot soft fork
- Provides for distributed key generation and threshold signing, without scripts, using multi-party computation (MPC)
- Initial Paper by Chelsea Komlo & Ian Goldberg
  - <https://eprint.iacr.org/2020/852.pdf>



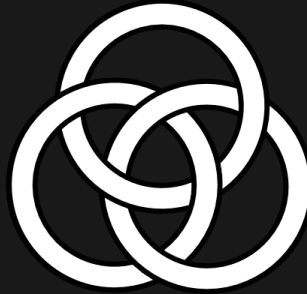
## What are the Major Elements of FROST?

- **Shamir Secret Sharing.** A secret is split into shares with a t-of-n configuration
- **Verifiable Secret Sharing (VSS).** Shares can be verified without reconstruction
- **Distributed Key Generation (DKG).** Shares can be generated without a trusted dealer
- **Schnorr Signatures.** Unlike ECDSA, Schnorr Signatures have a linear form
- **Signature Aggregation.** Multiple signers work together to construct a signature



## Why is FROST Important?

- **Advantages over Bitcoin Script.**
  - Better privacy: on-chain footprint is always a single key and a single signature, regardless of configuration
  - Lower fees: redeem scripts are much smaller than script-based multisig
  - Off-chain resharing: repair, refresh, enroll, disenroll, and modify the threshold without moving funds, incurring fees, and exposing private information
- **Advantages over Shamir Secret Sharing.**
  - No trusted dealer
  - No secret reconstruction



## FROST PRs

- FROST PR: [github.com/BlockstreamResearch/secp256k1-zkp/pull/138](https://github.com/BlockstreamResearch/secp256k1-zkp/pull/138)
- FROST Trusted Dealer PR: [github.com/BlockstreamResearch/secp256k1-zkp/pull/278](https://github.com/BlockstreamResearch/secp256k1-zkp/pull/278)
- FROST DKG BIP: [github.com/BlockstreamResearch/bip-frost-dkg](https://github.com/BlockstreamResearch/bip-frost-dkg)
  - batteries included
    - broadcast channel
    - pairwise secure channels
- FROST Signing BIP: [github.com/siv2r/bip-frost-signing](https://github.com/siv2r/bip-frost-signing)
- Zcash FROST taproot PR: <https://github.com/ZcashFoundation/frost/pull/584>



## New Papers

- Re-Randomized FROST: [eprint.iacr.org/2024/436](https://eprint.iacr.org/2024/436)
  - proves security for key tweaking (e.g. Taproot, BIP32)
- Arctic: Lightweight and Stateless Threshold Schnorr Signature: [eprint.iacr.org/2024/466](https://eprint.iacr.org/2024/466)
  - honest majority required:  $\mu \geq 2t-1$



## Proactive Secret Sharing (I)

- Refresh
  - planned in zcash: [github.com/ZcashFoundation/frost/issues/245](https://github.com/ZcashFoundation/frost/issues/245)
  - $n$  participants can update shares (or dis-enroll and re-enroll absent participants)
  - assumes at least  $t$  participants delete their old shares
  - can also be used for disenrollment of revoked participant
  - DKG with a 0 secret





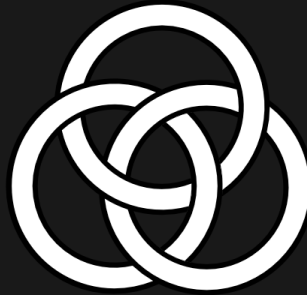
## Proactive Secret Sharing (II)

- Repair
  - implemented in zcash: [github.com/ZcashFoundation/frost/issues/41](https://github.com/ZcashFoundation/frost/issues/41)
  - $t$  participants can repair any lost shares
  - lost shares are not revealed to participants assisting in the repair
  - communication complexity of  $t(t + 1)/2$
  - can also be used for enrollment of new participant
  - additive secret sharing of polynomial shares interpolated at new ID



## Dynamic Secret Sharing

- Briefly discussed by zcash: [github.com/ZcashFoundation/frost/issues/519](https://github.com/ZcashFoundation/frost/issues/519)
- Threshold Increase by Zero Addition
  - $n$  participants can increase threshold
  - DKG with a 0 secret and higher degree polynomial
- Threshold Decrease by Public Evaluation
  - $n$  participants can decrease threshold
  - special subtraction of a publicly repaired share at new ID



## FROST Discussion

- Thoughts?
- Additions?
- Questions?



## Next Time (*May 1 - MayDay!*)

- Dan Gould on Serverless Payjoin v2
  - <https://github.com/bitcoin/bips/pull/1483>
- Improved UX with Gordian Request Reponse
  - <https://github.com/BlockchainCommons/SmartCustody/blob/master/Docs/Scenarios/Multisig-RR.md>

	Classic	R/R
Decision Points (💡)	5	2
Confirmation Points (👍)	0	6
Research Points (🧠)	11	1
Human Actions (👤)	31	14
Automated Actions (🤖)	5	33



[www.BlockchainCommons.com](http://www.BlockchainCommons.com)



Christopher Allen (@ChristopherA)