BLOCKCHAIN COMMONS

# INTRODUCTION TO FROST (FOR DEVELOPERS)

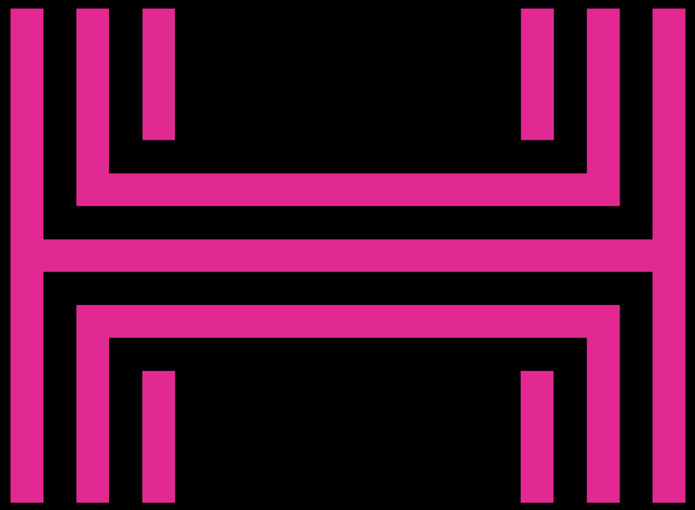
# WHAT IS BLOCKCHAIN COMMONS?

- We are a community that brings together stakeholders to collaboratively build open & interoperable, secure & compassionate infrastructure.
- We design decentralized solutions where everyone wins.
- We are a neutral “not-for-profit” that enables people to control their own digital destiny.

# CHRISTOPHER ALLEN

- The Past: Cryptographic Trust & Internet Privacy Pioneer
  - Co-editor of IETF TLS 1.0, world's broadest deployed security standard
  - ID 2020 Board Advisor, United Nations Summit on Digital Identity
  - Co-Author W3C Decentralized Identifiers 1.0
- The Present: Blockchain & Identity Architect
  - #RebootingWebOfTrust Design Workshops
  - W3C Invited Expert to DID 1.1 and Verifiable Credentials 1.1 Working Group
  - Co-author & Architect, IETF Drafts for dCBOR and Gordian Envelope
  - Producer,  Blockchain Commons FROST Meetings

PGP: **FDA6C78E**



Human Rights Foundation

Thanks to our 2024 FROST Sponsor

# WHAT IS FROST?

# Schnorr Threshold Signatures

- **FROST** Means
  - **F** lexible
  - **R** ound
  - **O** ptimized
  - **S** chnorr
  - **T** hreshold

# Schnorr Threshold Signatures

- **Schnorr**
  - Digital signature
  - Built on finite fields
  - Key is split
  - Efficient, short, scalable
- **Threshold**
  - M of N people can sign
  - where $M \leq N$

# HOW DOES FROST WORK?

# FROST Signing is Easy

- User has a secret share of a signing key
  - Generated with Shamir (VSS)
- A threshold of participants sign
  - Public verification share verifies a participant’s partial signature
  - Joint public key verifies aggregated signature

# FROST Keys

- Private Key
  - Split into Secret Shares
  - Does not exist intact (hopefully)
- Public Key
  - Available to verify aggregated signatures
  - “Group Verifying Key”
  - Does exist intact

# But How are Keys Generated?

- Two Methods
  - Trusted Dealer Generation
  - Distributed Key Generation (DKG)

# Trusted Dealer Generation

- Take a key, split a key
  - You must trust the entity (“dealer”) that does so
  - The key fully exists in memory when it’s split.
  - It’s a fairly traditional method

# Distributed Key Generation

- Multiparty protocol to create key
  - It’s never in memory!
- But there are a variety of ways to DKG
  - No official definition in RFC 9591
- PedPop DKG protocol is common
  - Pedersen with Proof of Possession
- ChillDKG Uses SimplPedPop+EncPedPop
- Luke Parker’s DKG-576 is 1 round!

# Signing Protocol

- R1: Commitment
  - “I made a secret” (nonce)
  - “It won’t change” (public commitment)
- R2: Signature Share Generation
  - “I’m proving my participation” (secret share)
  - “I’m signing” (signing share)
- Final: Signature Share Aggregation

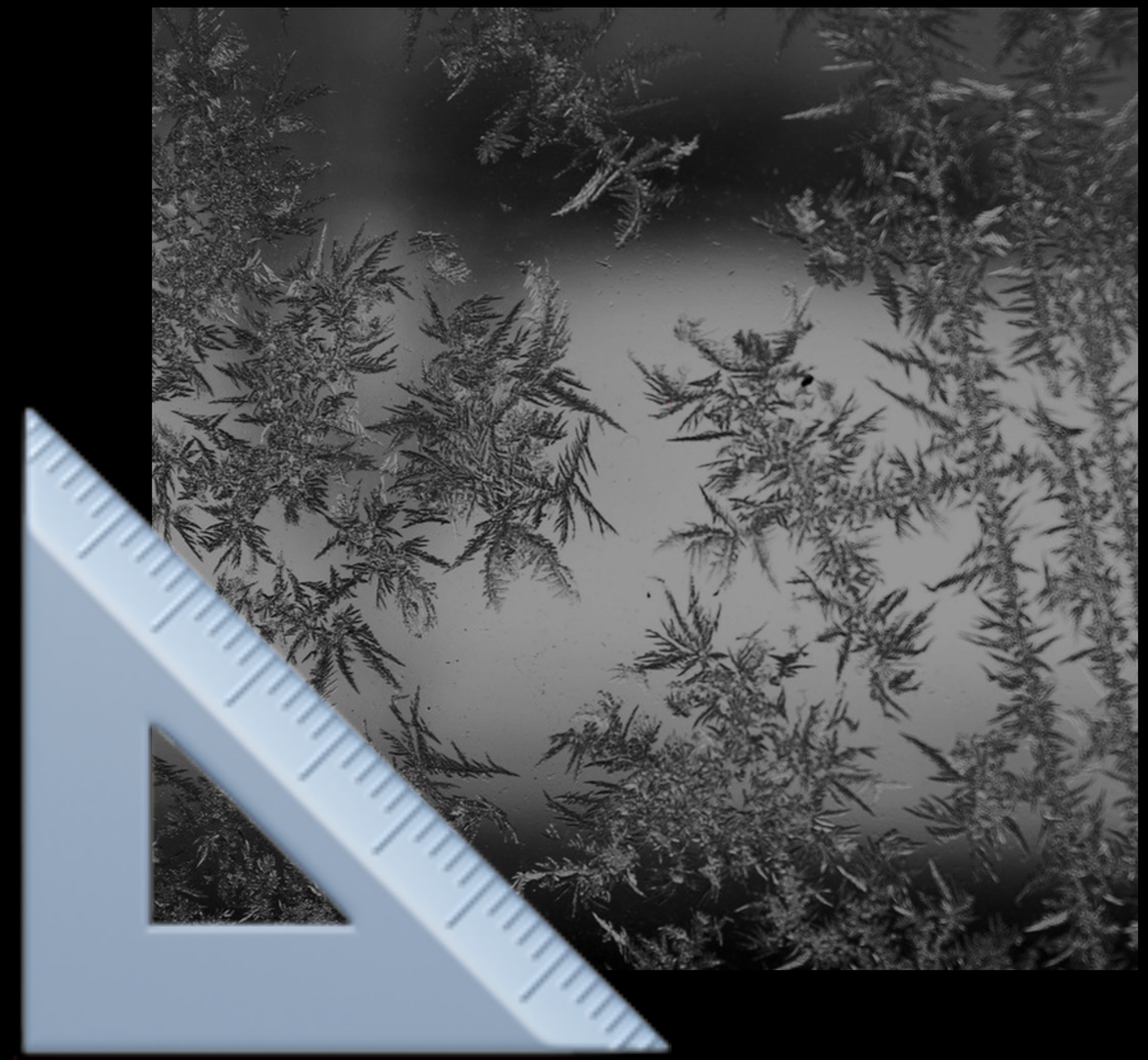
# WHY USE FROST?

# FROST is Secure

- Key is split
- With DKG, it never exists in one place.
- No Single Point of Failure (SPOF)

# FROST is Scalable

- FROST signatures are aggregatable
- All signatures are the same size
  - No matter how many people sign
- Allows for the creation of amazing sigs
  - Want a 66 of 100 threshold?
  - No problem!
- Even typical multisigs will be smaller
  - That means lower fees!



# FROST is Private

- All signatures look the same
  - No differences between 1 signature and 100 signatures
- Can’t even tell which members of a group signed a threshold
  - (unless they reveal secret info)

# FROST is Flexible

- Change groups & thresholds without changing public keys!
  - Repair: restore a lost share
  - Refresh: change shares
  - Enroll: add a member
  - Disenroll: remove a member
  - Modify: change threshold

# FROST is NOT Robust

- But beware: FROST is not robust!
- Misbehaving participant can “DoS” signature
- Coordinator has to identify misbehaving or non-participating members
- ROAST offers a wrapper to make FROST robust!

# HOW DO I USE FROST?

# Implementing FROST in 2025

- That’s the topic of today’s meeting!
- There are already FROST libraries that you can use.
- Stack is a production wallet that offers FROST.
- More to come (Space Wallet, Frostsnap, etc.)
- Be part of the coming FROST in 2025!

# Major Libraries

- ZF FROST (Rust)
  - frost-core, et. al.
- FROST UniFFI SDK (GoLang, Kotlin, Swift)
  - Translates ZF Frost
- SeraiDEX Crates (Rust)
  - dkg, modular-frost, bitcoin-serai
- BIP-340 FROST for secp256k1-zkp (C)
- frost-dalek (Rust), frost-ed25519 (Go), redjubjub (Rust), more!

## More on FROST

https://developer.blockchaincommons.com/frost/


CHRISTOPHER ALLEN

@ChristopherA

ChristopherA@BlockchainCommons.com

@BlockchainComns


"Advocating for the creation of open, interoperable, secure & compassionate digital infrastructure to enable people to control their own digital destiny and to maintain their human dignity online"