SkipChains: Offline and Peer-to-Peer Verifiable Blockchains

Prof. Bryan Ford
Decentralized/Distributed Systems (DEDIS)

IRTF CFRG – November 15, 2017

The Call of the Blockchain

(credit: Tony Arcieri)

But… Today's Blockchains Suck

Public/permissionless (e.g., Bitcoin, Ethereum)

- Weak probabilistic consistency
- Long transaction delays, low throughput
- **Clients must be online, well-connected**
- Mining is inefficient, insecure, re-centralizing

Private/permissioned (e.g., HyperLedger, R3, …)

- Weak security – single points of compromise

Problem: Efficient Verification

How does a “light” (low-power, mobile) client securely confirm a thing is **on the blockchain**?

- Especially after being offline for months, years?
- Without “just trusting” central party (exchange)?

Weak SPV approach: just verify block headers

- Still must gossip with many parties
- Still costs bandwidth, especially to “catch up”
- Vulnerable to (costly but feasible) fake views

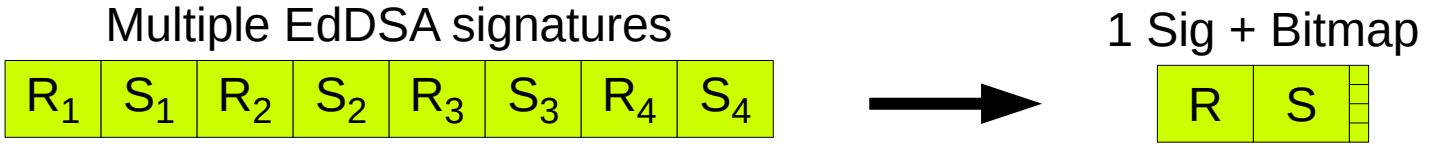
Cryptographic SkipChains

Offline- and peer-to-peer-verifiable blockchains

- DEDIS “Chainiac” paper [USENIX Security ‘17]
- Applied to secure key & software updates

Builds on Collective Signing (CoSi)

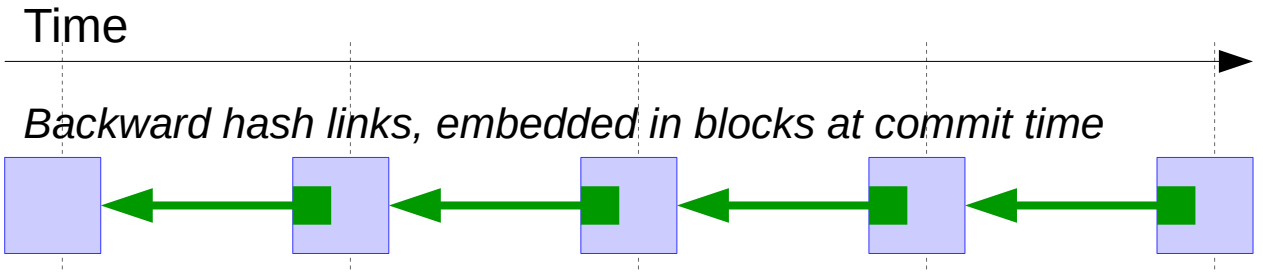
- DEDIS “Authorities” paper [IEEE S&P ‘16]
- Internet-Draft: draft-ford-cfrg-cosi-00



Backward and Forward Verifiability

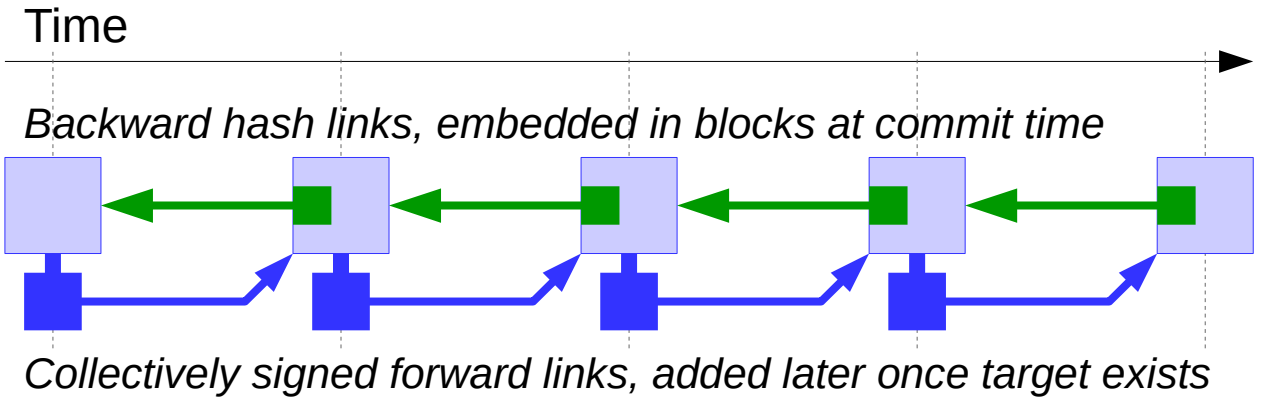
Standard blockchains traversable only **backward**

- Via hash back-links from current head



Chainiac adds traversability **forward in time**

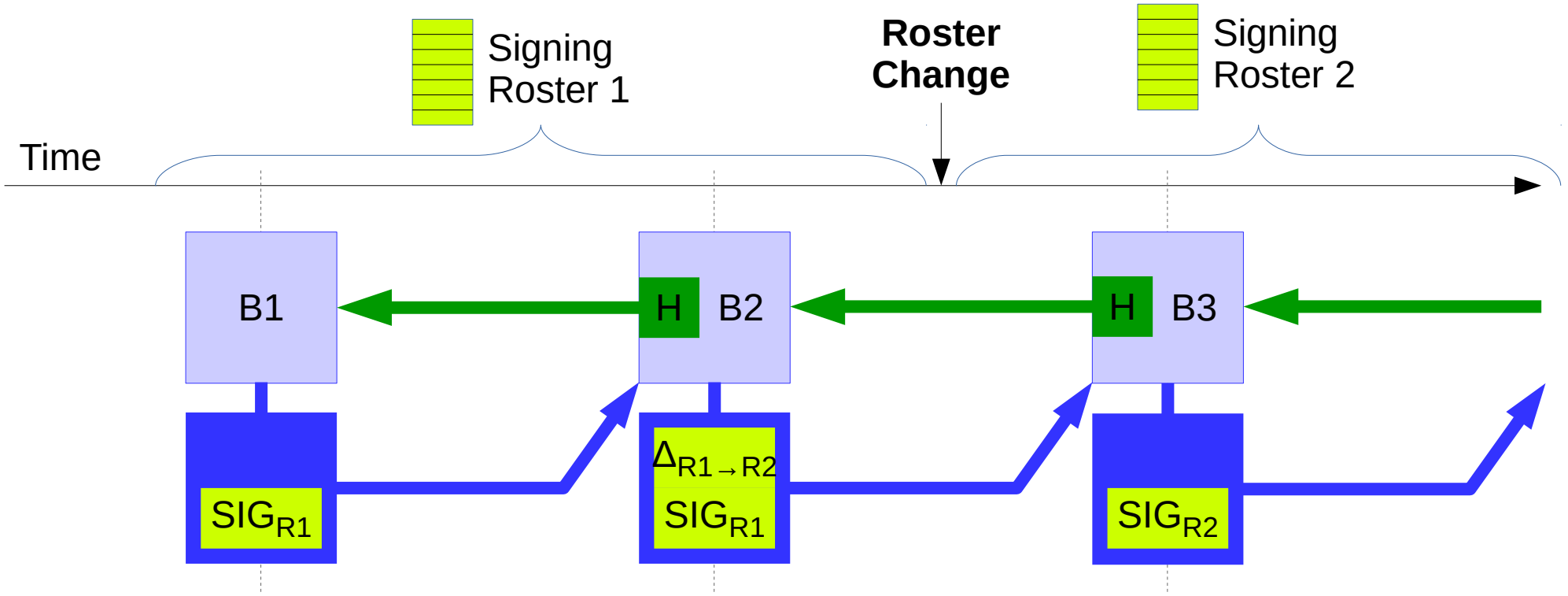
- Collective signature by prior consensus group



Signing Key Group Evolution

Forward pointers include signing-key-group deltas

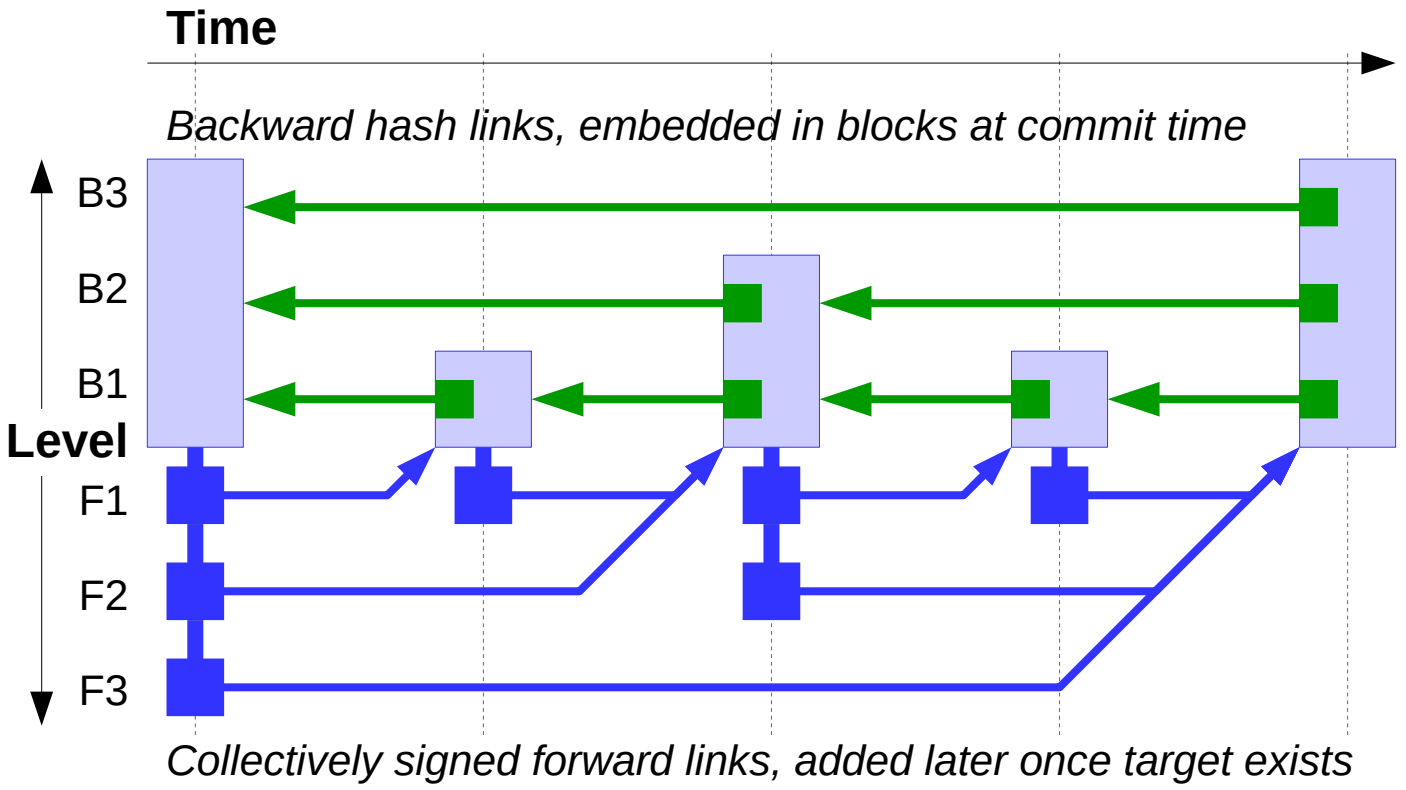
- Whenever public keys added, removed, rotated



Taking Leaps Through Time

Each block validates *prev* w/hash, *next* w/sig

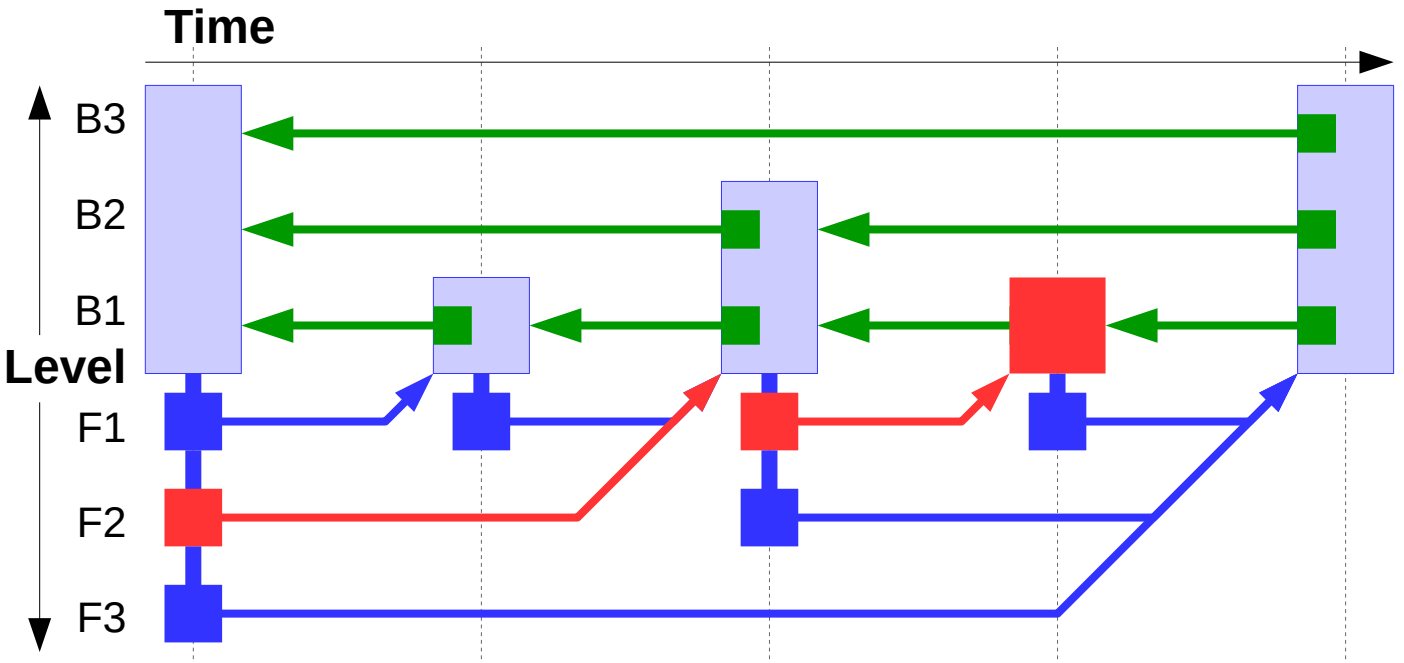
- Higher level hashes, sigs → longer hops
- O(log N) traversal arbitrarily forward, back



O(log N) On-SkipChain Proofs

Prove a thing is on-chain anywhere in time

- Securely help outdated peers “catch up”
- Already-up-to-date verifiers rely only on *recent* collective signatures for security



SkipChains: Summary

Cryptographically traversible blockchain

- Low-power clients can *follow* efficiently
  - Need not download/verify every block [header] *or* trust the word of any “full node”
- Verify transactions *forward* or *back* in time
  - Including disconnected, peer-to-peer clients
- Consensus group signing keys can change
  - Slowly: e.g., permissioned blockchain
  - Rapidly: e.g., proof-of-stake blockchain

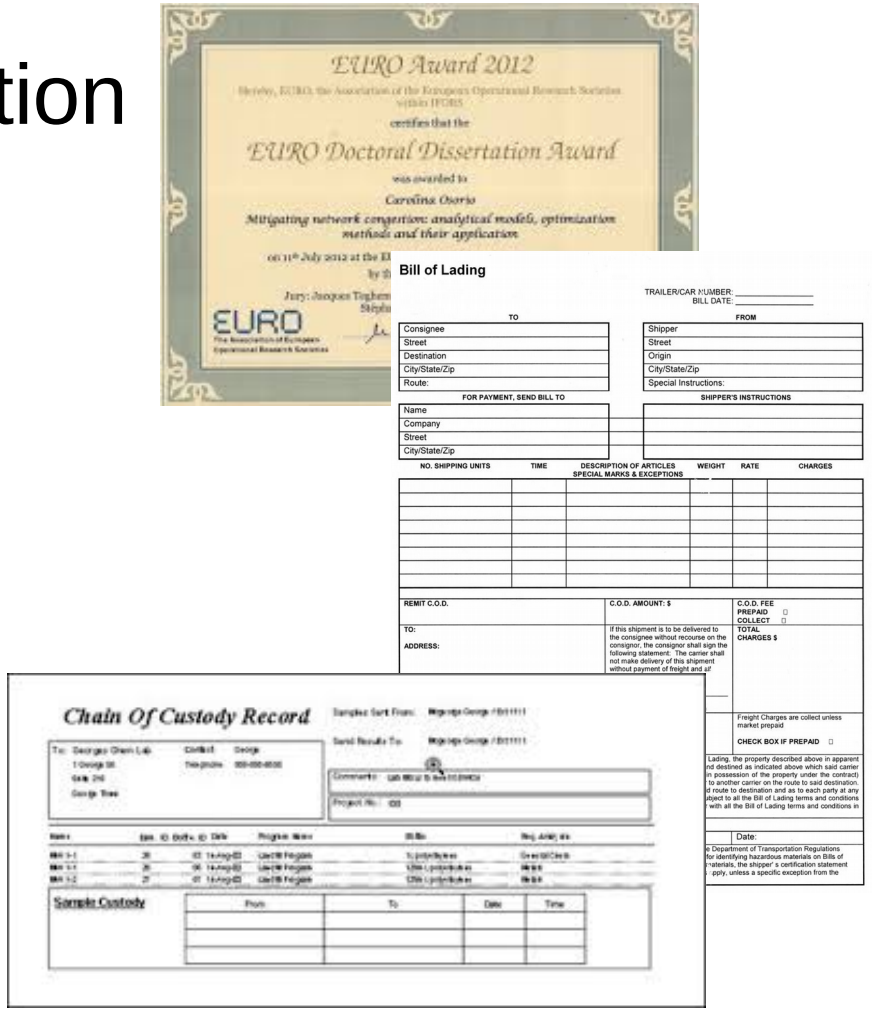
Applications of SkipChains

Enable Offline/P2P verification

- Works even if Internet is unavailable, slow, costly

Broad applications

- Software/key updates
- Blockchain-Attested Degrees, Awards, …
- Chain-of-Custody, Bills of Lading, …



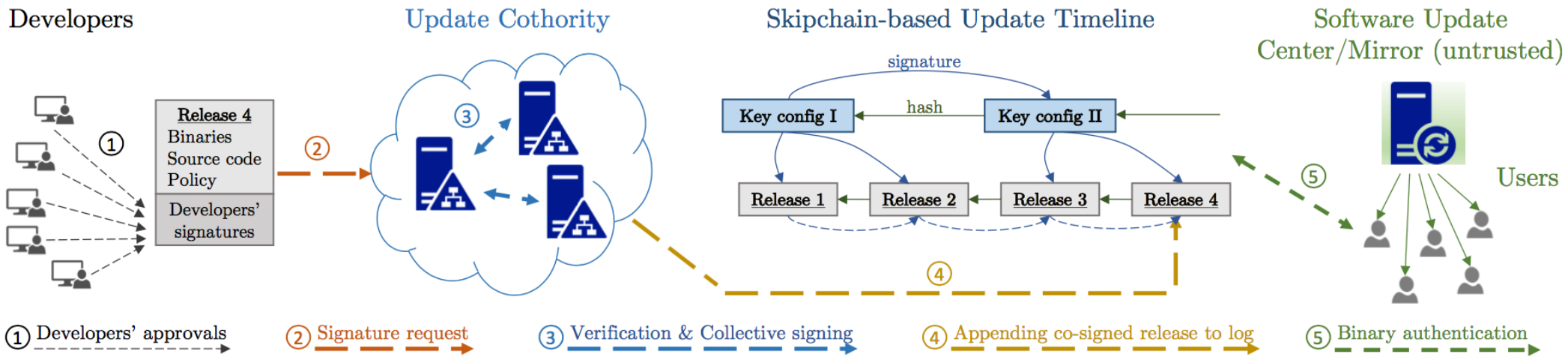
Blog: “How Do You Know It’s On the Blockchain?”

Chainiac: Secure, Transparent Software Development & Updates

End-to-end secure software supply chain

- Development: peer review, signoff workflow
- Build: independent verification of exact binaries
- Distribution: offline/P2P updates via SkipChains

Applicable to open source & proprietary software



Code available on GitHub…

All are welcome to use it and build on it...

Kyber: Advanced crypto library for Go

- https://github.com/dedis/kyber
- Public-key Encryption, Signatures, Shamir Secret Sharing, Zero-Knowledge Proofs, Verifiable Shuffles, Optimized Ed25519, …

Cothority: Collective authority software suite

- https://github.com/dedis/cothority
- CoSi, ByzCoin, Chainiac, …