Messaging Layer Security

Berlin Crypto Meetup, April 2nd 2019

Raphael Robert

Current secure messaging

- **mpOTR, (n+1) sec**: No PCS
- **OpenPGP, S/MIME**: No FS/PCS, linear scaling
- **Pairwise double ratchet**: (libsignal, Proteus, Olm, etc.) FS & partial PCS, asynchronous, but linear scaling, no crypto for group membership
- **Sender keys**: (WhatsApp, FBM, OMEMO, Megolm, etc.) Linear scaling, PCS possible but very expensive

Goals and security properties

- **Groups**: support large groups (up to 50k members) efficiently
- **Asynchronous**: No two participants are online at the same time
- **Security**: Manage group membership, modern properties like Forward Secrecy and Post-Compromise Security
- **Formal verification**: similar to TLS, verified specification & implementations
- **Standardized**: Interoperable implementations

History

2015 Tree-based encryption schemes
(FB, University of Oxford, Mozilla, Cisco, ...)

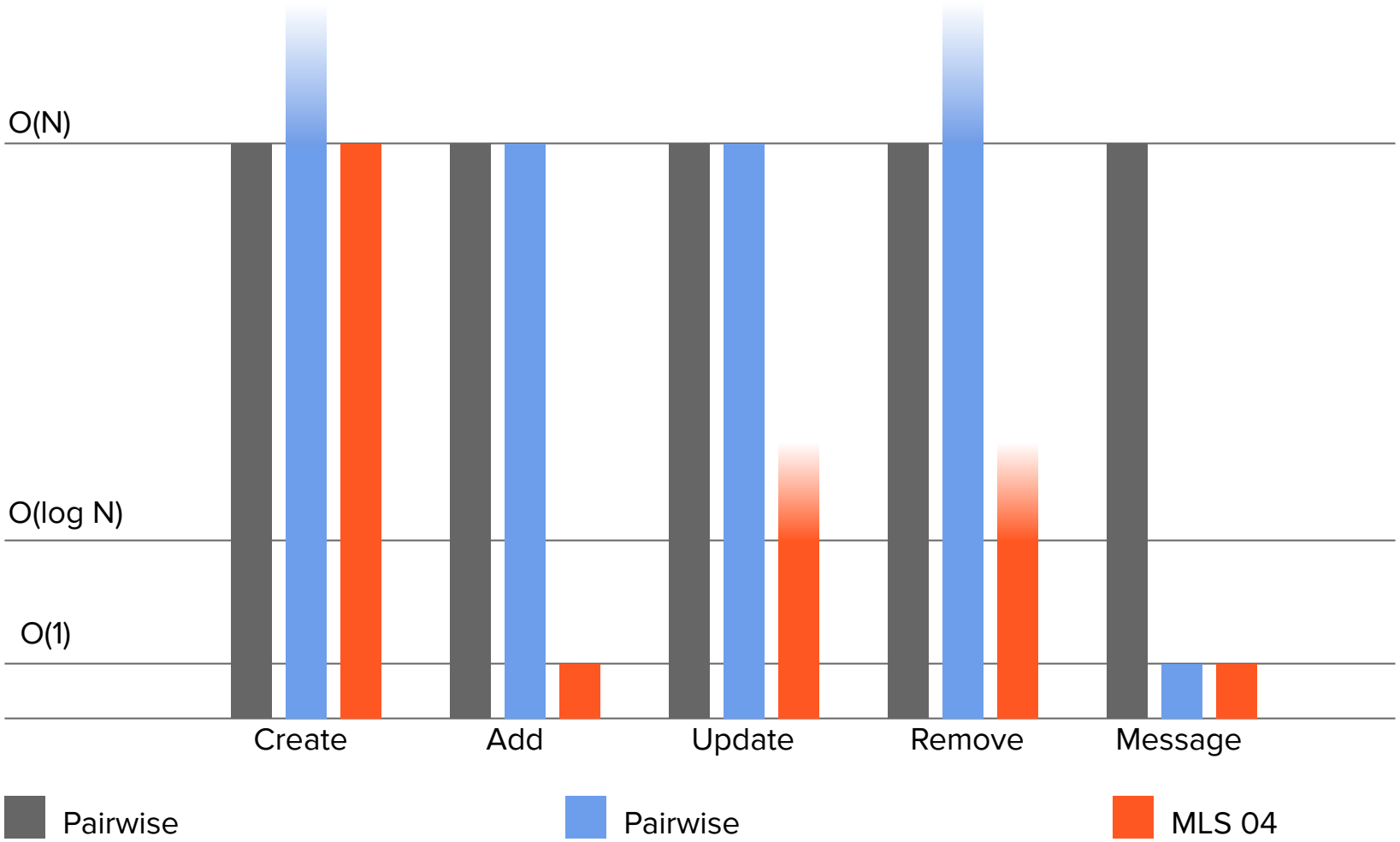
2016 Wire seeks partners for IETF standard

2017 Asynchronous Ratcheting Trees (paper)
(Millican, Cohn-Gordon, Cremers, et al)

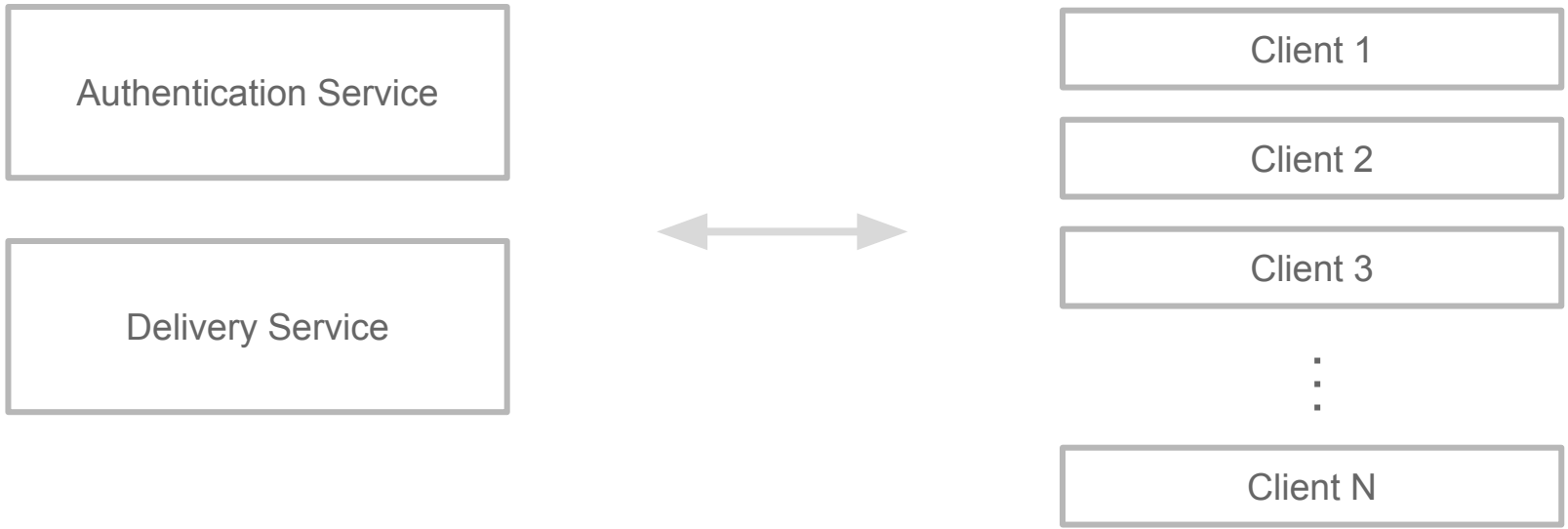
2018 MLS IETF BoF & WG

The IETF working group

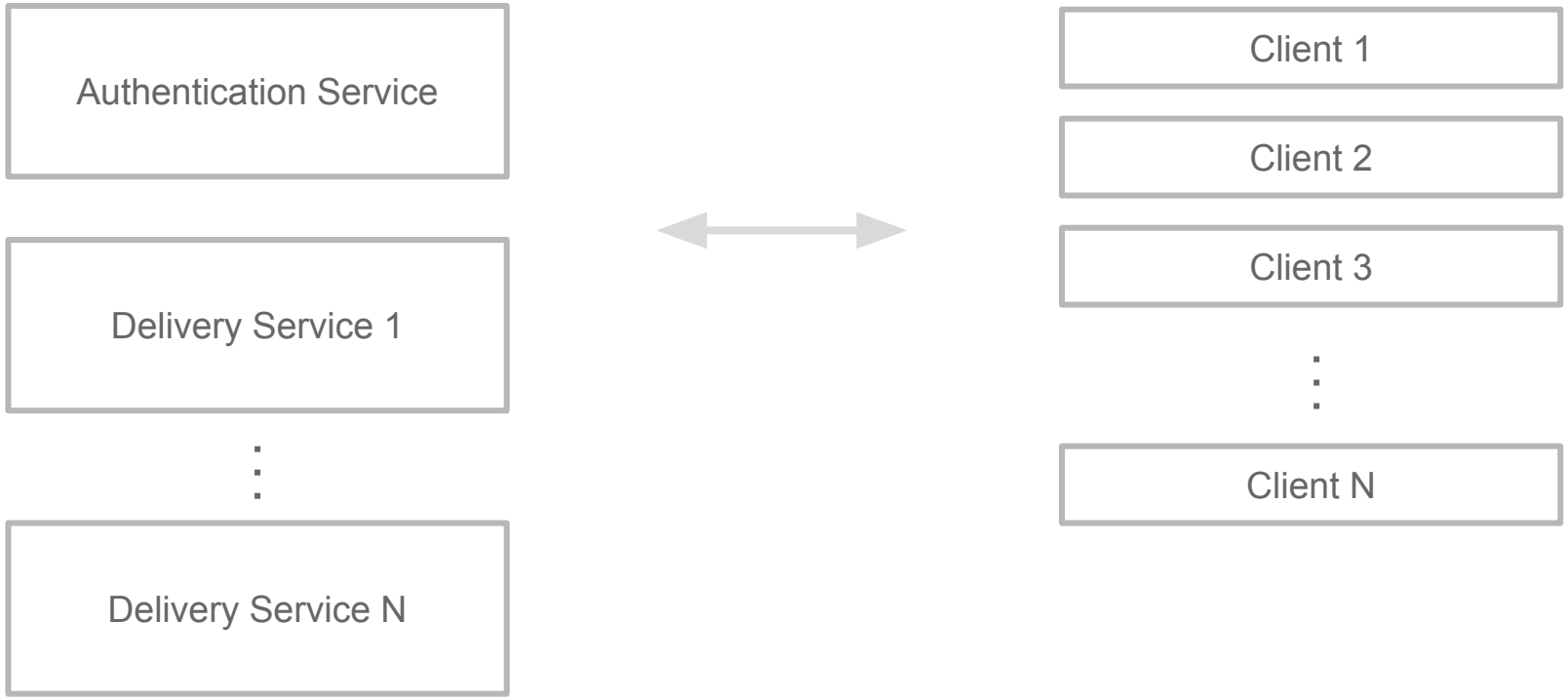
Forward Secrecy & Post-Compromise Security


O(N)
O(log N)
O(1)
Create
Add
Update
Remove
Message
Pairwise
Pairwise
MLS 04

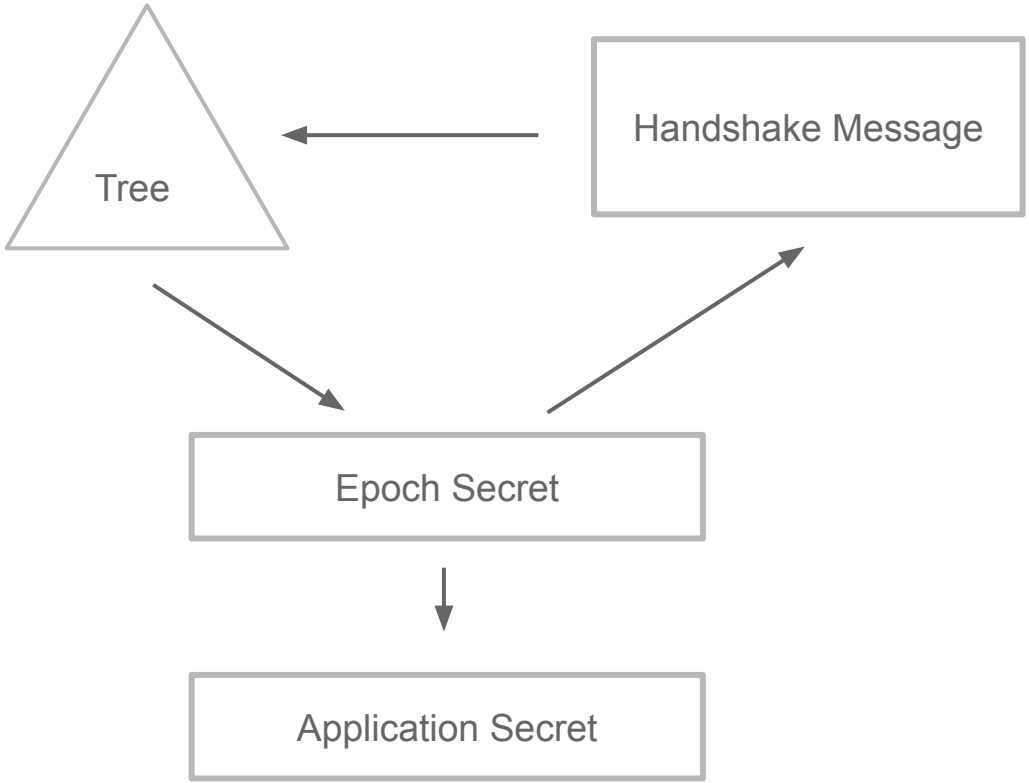
Architecture



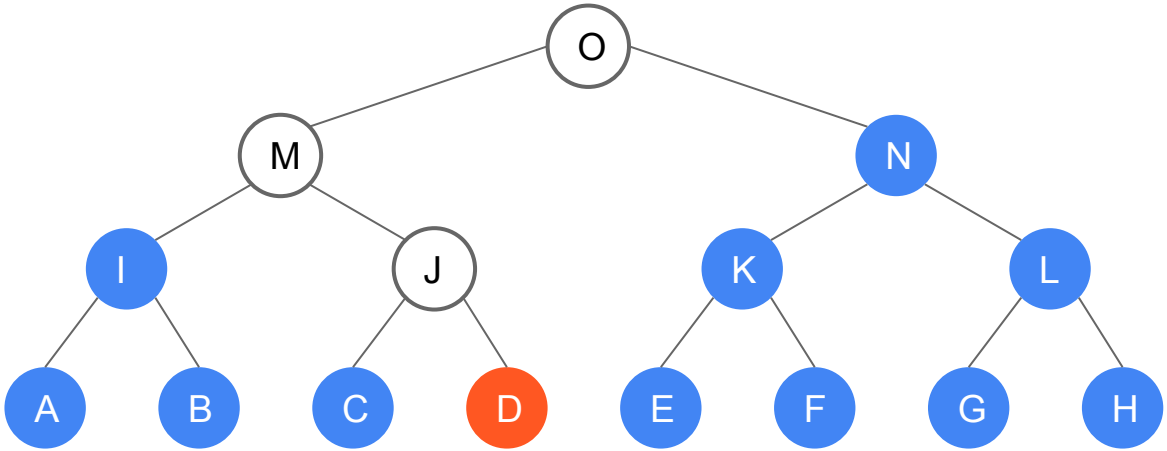
Federation


Authentication Service
Delivery Service 1
Delivery Service N
Client 1
Client 2
Client 3
Client N

Protocol

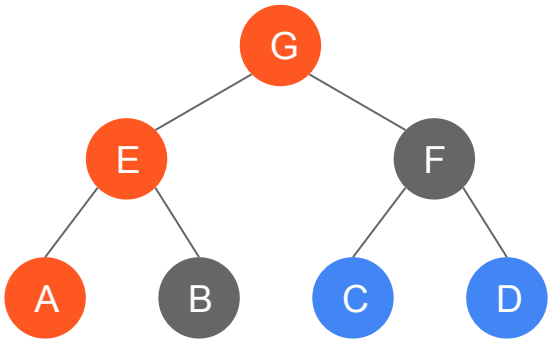


Tree invariant



The private key for an intermediate node is known to a member iff the node is an ancestor of the member's leaf

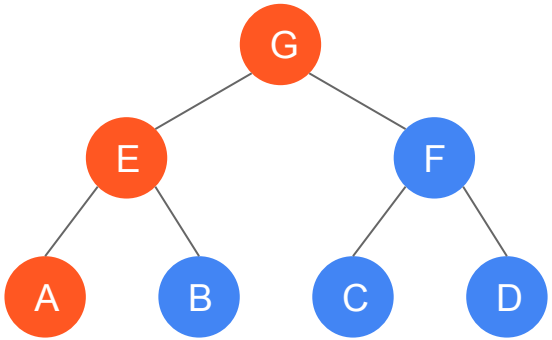
ART



E = DH(A, B)

G = DH(E, F)

TreeKEM



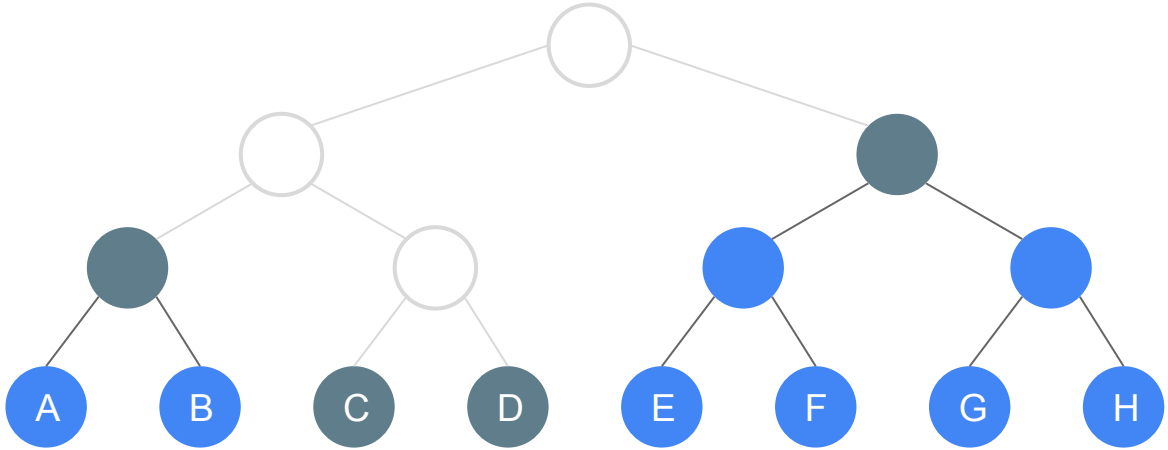
E = H(A)

G = H(E) = H(H(A))

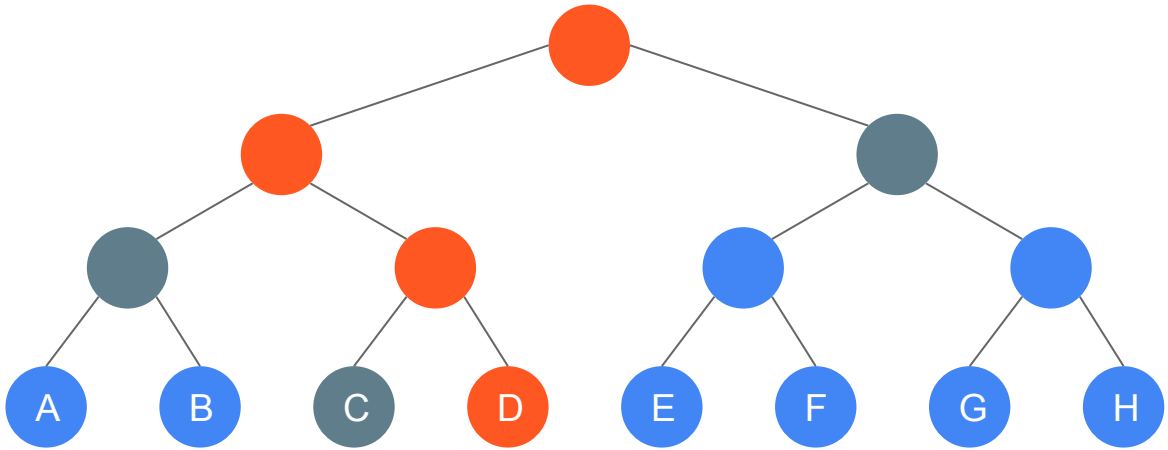
Advantages of TreeKEM

- Trees can contain blank nodes
- No “double joins”
- Updates can be processed in O(1) instead of O(log N)

Blank nodes



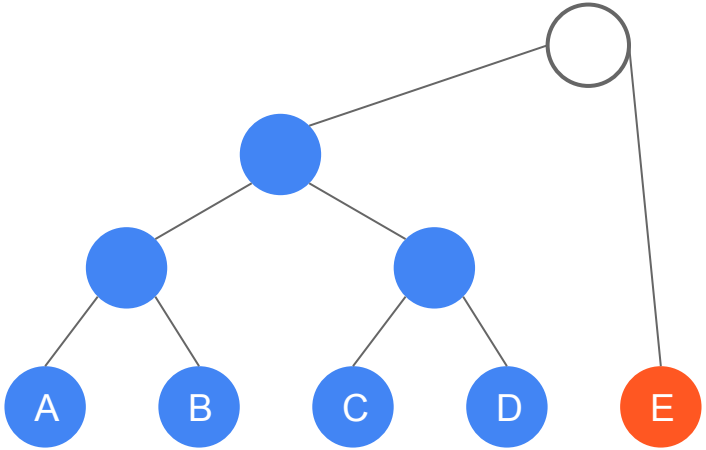
Update



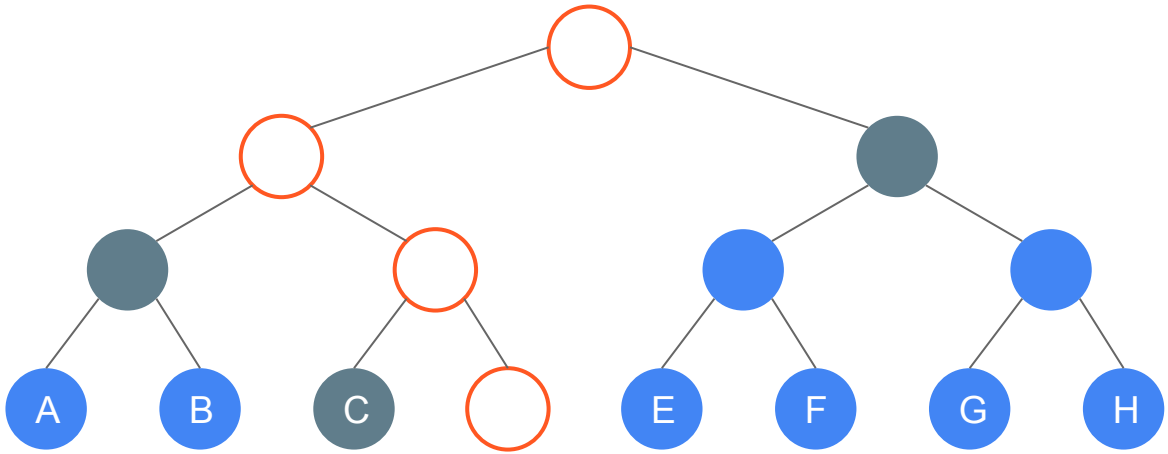
Direct path

Copath

Add



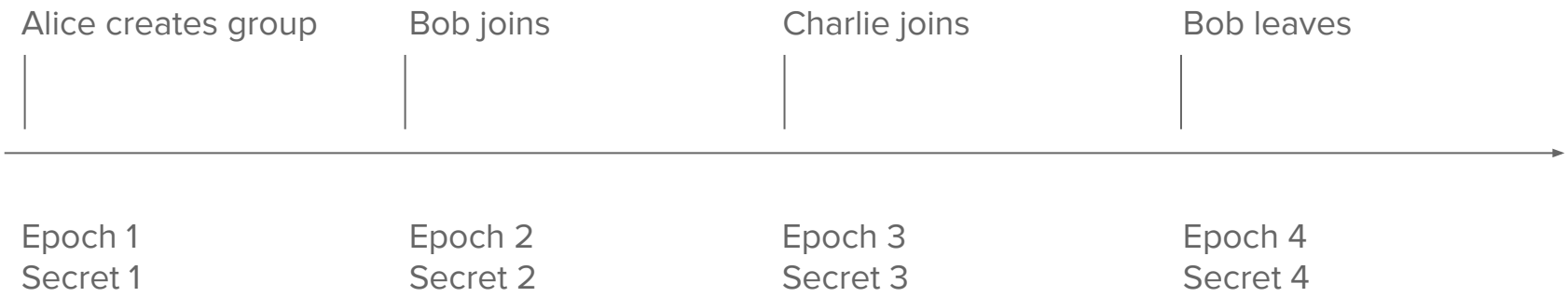
Remove



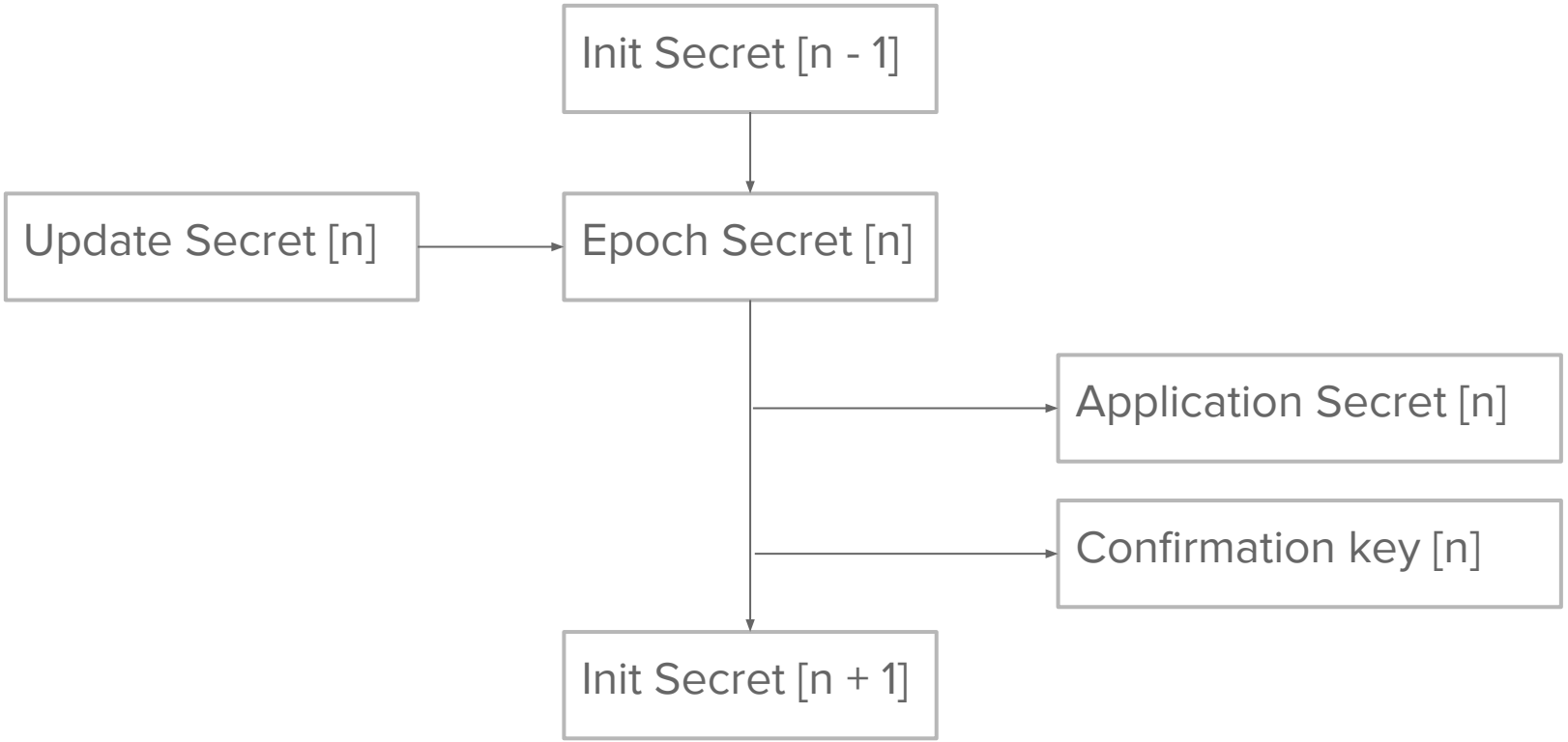
Direct path

Copath

Epochs


Alice creates group
Bob joins
Charlie joins
Bob leaves
Epoch 1
Secret 1
Epoch 2
Secret 2
Epoch 3
Secret 3
Epoch 4
Secret 4

Key Schedule



Authentication

Sign & MAC

- Handshake messages are signed with a signature key belonging to a client (keys are exchanged via Authentication Service)
- Handshake messages are authenticated with a MAC (confirmation key from the key schedule)

Working areas

- **Stabilizing** parts of the protocol for analysis
- **Security/Privacy**: HS message protection, general meta-data protection, malicious insider protection
- **Efficiency improvements**: Rebalancing trees, LazyUpdates, Tree warm-up, ...
- **Functionality improvements**: server initiated Add/Remove, recovery from state loss
- **Server**: Message ordering, ACK/NACK, federation, ...
- **Analysis**: TreeKEM & Authentication, general protocol, computational and symbolic verification, ...

Implementations

- **melissa** (Wire, Rust) https://github.com/wireapp/melissa
- **mlspp** (Cisco, C++) https://github.com/cisco/mlspp
- **MLS*** (Inria, F*)
- **molasses** (Trail of Bits, Rust) https://github.com/trailofbits/molasses
- **RefMLS** (NYU Paris, JS)
- [redacted] (Google, C++)

Messaging Layer Security

- Architecture: https://github.com/mlswg/mls-architecture
  https://architecture.messaginglayersecurity.rocks
- Protocol: https://github.com/mlswg/mls-protocol
  https://protocol.messaginglayersecurity.rocks
- Code + Interop: https://github.com/mlswg/mls-implementations
- Discussion: mls@ietf.org (archives)

https://app.wire.com/join/?key=qmrRRfaklMRm8UsYSqpA&code=KD8O6_Pvkli3pmzXbWtr

Thanks

Contact:

Raphael Robert

raphael@wire.com

@raphael on Wire

@raphaelrobert on Twitter