Invalid Commits and Access Control

Brendan McMillion

IETF 120 / July 22, 2024

Why is this PR needed?

- People have shared reservations about deploying MLS with encrypted handshake messages (primarily how to do access control)

- Other security issues in MLS deployments (in particular, MIMI) have come up, e.g. “Forced Rejoin” attacks

These are arch issues, not protocol issues => mls-arch is the right place to discuss them, and lack of current text is negatively affecting real-world use of MLS

What is a Forced Rejoin attack?

In many MLS deployments:

- Delivery Service chooses which Commit ends a given epoch, not Members
- Delivery Service may choose a Commit that Members are unable to process
- When members can’t process the Commit, they do an External Rejoin

What is a Forced Rejoin attack?

In many MLS deployments:

- Delivery Service chooses which Commit ends a given epoch, not Members
- Delivery Service may choose a Commit that Members are unable to process
- When members can’t process the Commit, they do an External Rejoin

Few issues here:

- Meant to prevent trivial DoS attack, but doesn’t
- Prevents group members from achieving PCS through Updates / Commits
- Allows arbitrary modifications to ratchet tree

Denial of Service

- Simple DoS attack: User sends a Commit that nobody can process
  - External Rejoin is meant to prevent this

- New DoS attack:
  1. User sends a Commit that nobody can process
  2. All group members are triggered to Externally Rejoin
  3. Malicious user sends a Commit with wrong GroupInfo
  4. Honest users that try to rejoin with wrong GroupInfo trigger all other users to rejoin again
  5. Cycle repeats

  DS has no real way to identify which user is malicious

Post-Compromise Security

Attack:

1. Alice's view of the key schedule is leaked in epoch i
2. Alice sends a PCS-achieving update which is processed by Bob
3. DS sends a junk commit to Bob, triggering him to do an External Rejoin
4. DS provides Bob with the GroupInfo corresponding to epoch i
5. Bob rejoins the compromised epoch, allowing DS to eavesdrop

In a pure RFC 9420 implementation, this is all that’s needed to prevent PCS!

What is in this PR? (Invalid Commits)

Brief descriptions of three different approaches:

1. Members choose Commit through consensus

2. Delivery Service chooses next Commit (possibly with External Rejoin)

3. Group Members silently ignore Commits they can’t process

Why these approaches? They cover all proposals I’ve seen

What do we say about these approaches?

1. Members choose Commit through consensus
   - DoS risk & can force users to stay in compromised state

2. Delivery Service chooses next Commit (possibly with External Rejoin)
   - DoS risk & can allow DS to revert PCS-achieving updates

3. Group Members silently ignore Commits they can’t process
   - Allows group state to fork, which complicates operation

What is in this PR? (Access Control)

Brief descriptions of two different approaches:

1. Public Handshake Messages
    - DS can inspect ratchet tree

2. Private Handshake Messages:
    - Bearer token can be derived from the MLS key schedule

Why these approaches? They cover all proposals I’ve seen

Recap

Motivation for PR:

- Implementors & other WGs **unintentionally** reducing MLS' security guarantees through specific architecture choices

PR addresses by discussing:

- Three approaches to handle Invalid Commits + discusses trade-offs between
- Two approaches for Access Control, for public or private handshake msgs
- Not prescriptive! Only descriptive