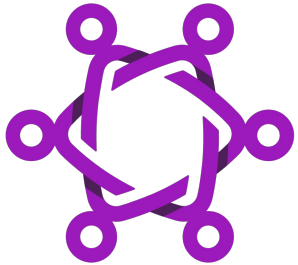


Light Clients

draft-kiefer-mls-light

Problem: Download and Memory

MLS requires that clients download, validate, and store the group's ratchet tree

Each participant only uses one MLSCiphertext in a Commit

… but has to download O(N) data to verify the Commit signature

In large groups, these objects can be L A R G E

In a 1,000-participant Webex meeting (empty tree):

Tree: 3.5MB in memory / 2.3MB gzip'ed

Commit: 391KB

Light Clients

A **light client** is a member of the group that **does not have the ratchet tree**

A light client **cannot commit**

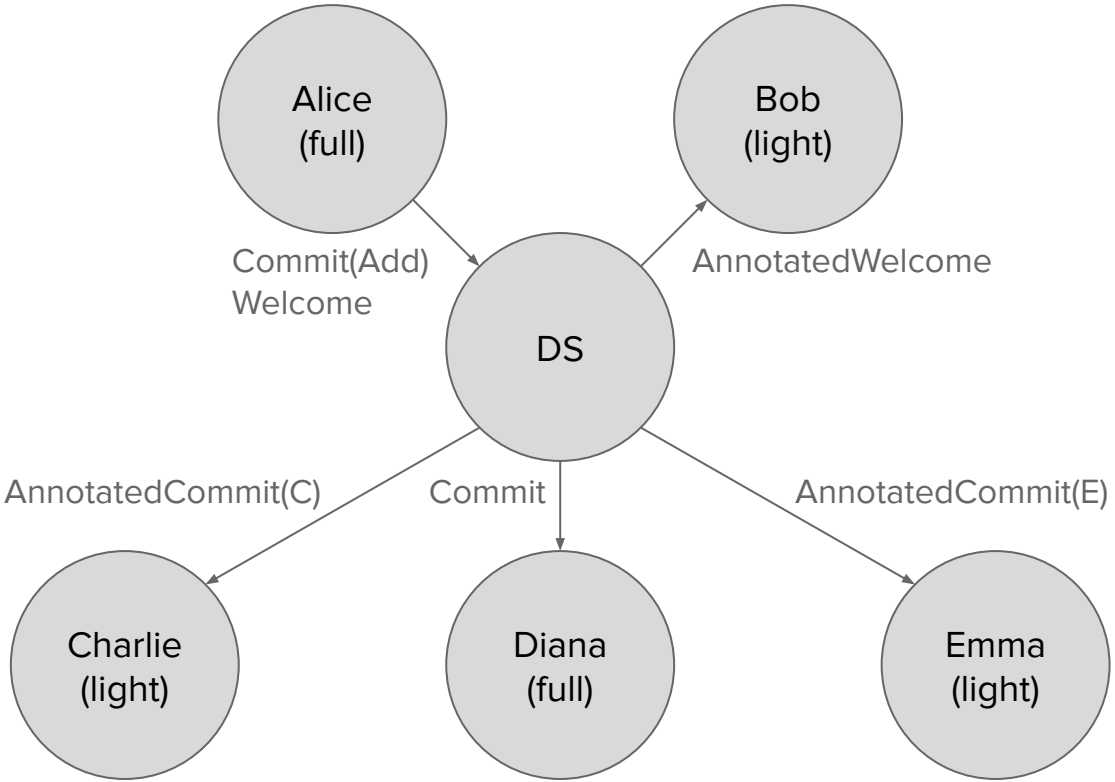
A light client **cannot process a normal Commit**

Instead:

- Light client joins with O(log N) information about the tree
- DS transforms Commit into per-light-client AnnotatedCommit

A light client can join the group with O(log N) download / memory

Operating a Group with Light Clients



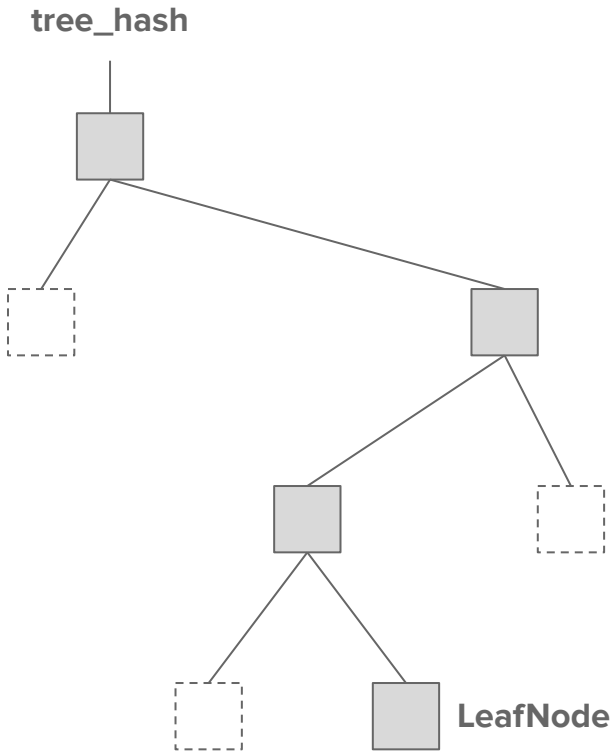
Tree Slices & Membership Proofs

Light clients don’t have the whole tree, but they do have the tree hash

Using Merkle-tree-like constructions, a light client can verify that a specific leaf node is in the tree.

These proofs are O(log N) in size and verification complexity

- Nodes along direct path
- Hashes along copath



AnnotatedWelcome & AnnotatedCommit

```
struct {
    T message;
    MembershipProof sender_membership_proof;
} SenderAuthenticatedMessage<T>;

struct {
    SenderAuthenticated<Welcome> welcome;
    MembershipProof joiner_membership_proof;
} AnnotatedWelcome;

struct {
    MLSMessage commit;
    optional<MembershipProof> sender_membership_proof;
    opaque tree_hash_after<V>;
    optional<uint32> resolution_index;

    MembershipProof sender_membership_proof_after;
    MembershipProof receiver_membership_proof_after;
} AnnotatedCommit;
```

SenderAuthenticatedMessage provides a membership proof so you can verify the signature on a message

AnnotatedWelcome provides a proof that the joiner is in the group

AnnotatedCommit provides:

- Coordinates for processing the path
- Proofs of membership after the commit

Light clients still need to see Proposals

Integrates well with Split Commits

draft-mularczyk-mls-splitcommit

Operation	RFC MLS	Light MLS	Split Commits	Light + Split
Join	O(N)	O(log N)	O(N)	O(log N)
Process Commit	O(N)	O(N)	O(log N)	O(log N)

Table 1: Download scaling under protocol variations

Split Commits change Commits from O(N) to O(log N)

If we do AnnotatedSplitCommit, we get everything O(log N) for light clients

The Authentication Trade-Off

Light MLS trades authenticity for confidentiality

It creates a state where a member can encrypt to/from the group, but doesn't know who is in the group (only a commitment = tree hash).

- SenderAuthenticatedMessage lets members authenticate senders
- … but they don't know who receives until they download the tree

This state is acceptable for several use cases, for example:

- Cases where the full member list isn't visible immediately anyway
- Broadcast cases, where receivers only care about source authentication

Status & Questions

This has been hanging around for a while now

- Formal analysis has already been done (Light MLS + SplitCommit)
- Implemented in MLSpp since Nov 2024
- Deployment interest from Webex, Discord, others

Do folks agree that this is a worthwhile problem to solve, and this approach works?

Should we combine this draft with draft-mularczyk-mls-splitcommit?