MLS Protocol

draft-ietf-mls-protocol-08

Richard Barnes, Raphael Robert,
Benjamin Beurdouche

You Are Here

draft-11 issued Dec 22, after issue resolutions at IETF 109

Since then, we have been in feature freeze, allowing for implementation

 Initial interop testing has begun!

Few issues/PRs since then with clarifications, bugfixes

Interop!

GRPC-BASED TEST HARNESS

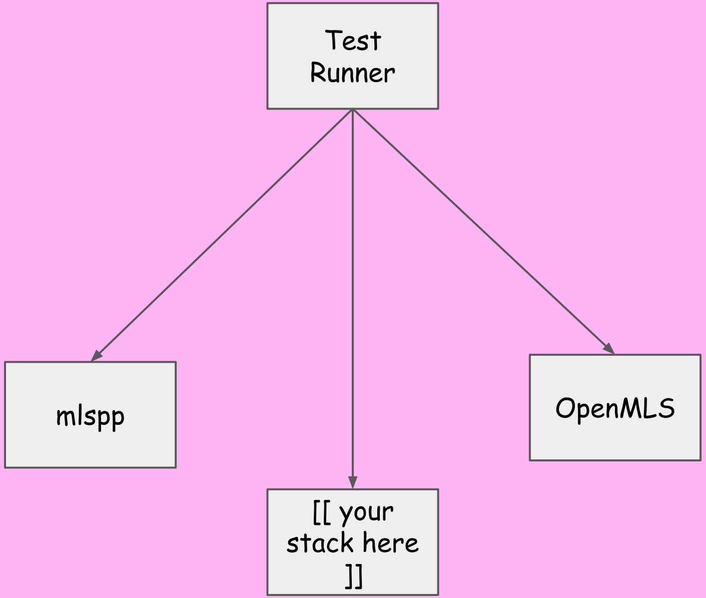
MLS client = gRPC server

Test runner = gRPC client, commands MLS implementations to do stuff

Two types of tests:

- Test vectors: Generate / Verify sample data from subsystems
- Scenarios: Actual protocol operations

Automatically generates permutations of ciphersuites / roles



Progress so far

Two implementations: mlspp and OpenMLS

Interop verified on test vectors:

- Tree math
- ~~Key schedule~~ joiner_secret
- Encryption
- TreeKEM
- Message encoding

First Interop!

```
{
  "test_vectors": {
    "tree_math": [{
        "generator": "mlspp",
        "verifier": "mlspp”
      }, {
        "generator": "mlspp",
        "verifier": "OpenMLS"
      }, {
        "generator": "OpenMLS",
        "verifier": "mlspp",
        "error": "rpc error: code = InvalidArgument desc = Error: parent  (nullopt) != 9"
      }, {
        "generator": "OpenMLS",
        "verifier": "OpenMLS"
      }]
  }
}
```

mlspp accepts its own test vector

OpenMLS accepts a test vector from mlspp

Mlspp is unhappy OpenMLS didn’t do all the cases

OpenMLS accepts its own test vector

Bugs found so far

Not generating full tree math test vectors

Swapped order of HKDF.Extract inputs

Wrong algorithms associated with a ciphersuite

No spec bugs … yet

More to come as we test more surface...

Spec Issues / PRs

Breaking Changes (editorial omitted)

#461 - Truncate tree size on remval / #459 - Trim tree after removal

#455 - Make PreSharedKeys non-optional in GroupSecrets

#453 - Use the GroupContext to derive the joiner_secret

#439 - Identities SHOULD be unique per group

#457 - Clarify ParentHash verification

#443 - External commit for resync used with PSK

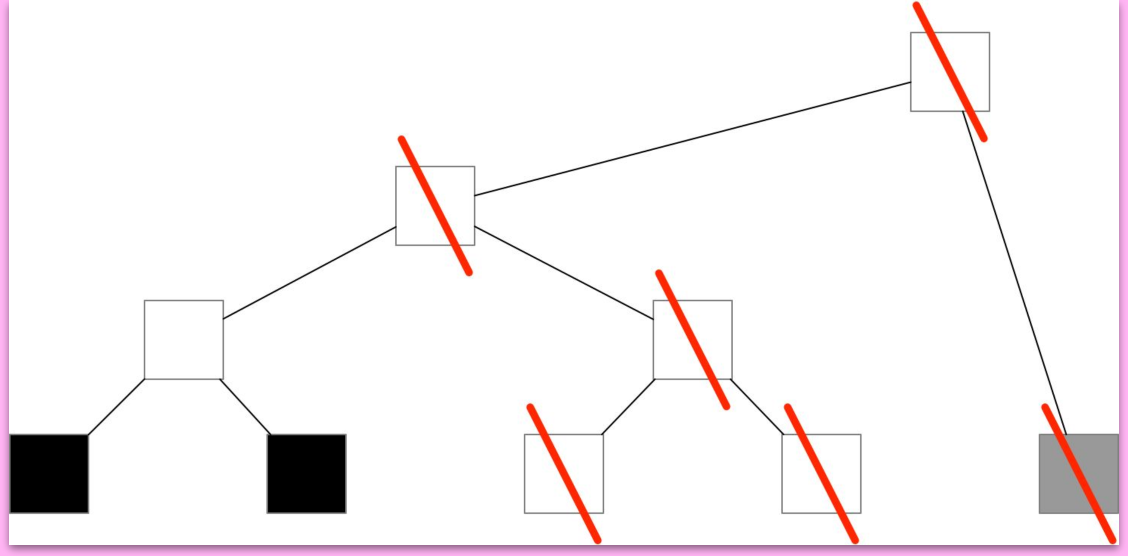
#XXX - Resolve ambiguity around which context is used when

#459 / #461 - Tree Trimming

In draft-07, we made the tree smaller on Remove:

- Truncate the tree such that the rightmost non-blank leaf is the last node of the tree, for the time of the computation

This got lost in draft-08 and later, just need to restore it



#455 - Make PreSharedKeys non-optional

Ambiguity in GroupSecrets:

PSK field is optional…

… but can also be zero length

What if it is present, but empty?

In other words, no need for the field to be optional

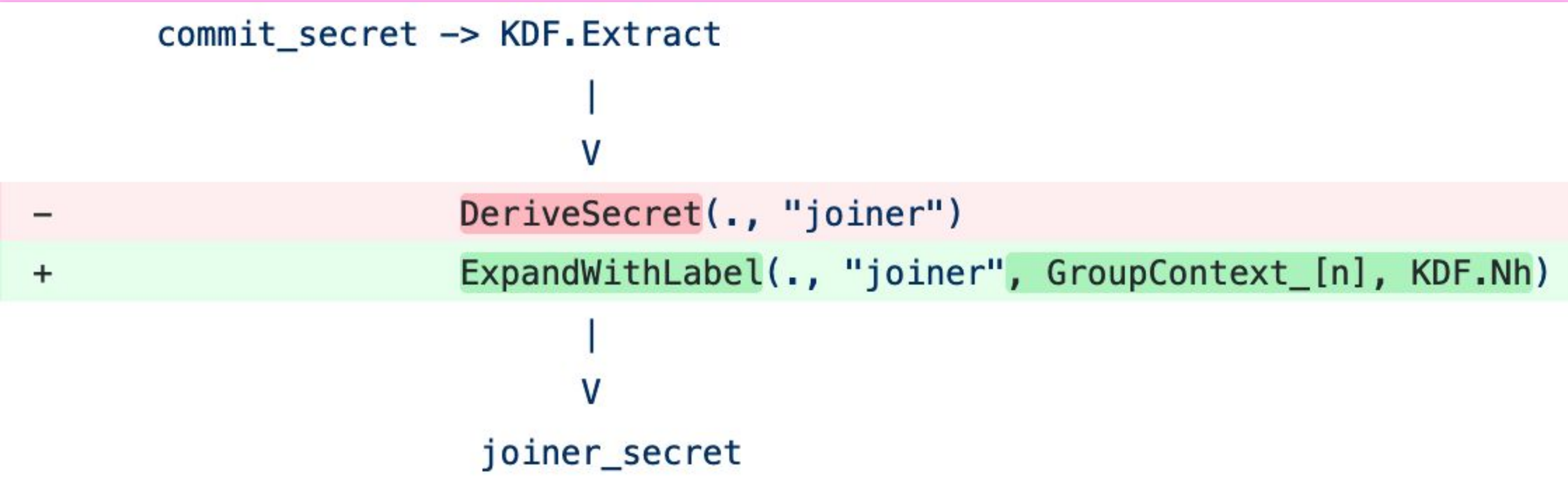
```
struct {
    PreSharedKeyID psks<0..2^16-1>;
} PreSharedKeys;

struct {
    opaque joiner_secret<1..255>;
    optional<PathSecret> path_secret;
    optional<PreSharedKeys> psks;
    PreSharedKeys psks;
} GroupSecrets;
```

#453 - GroupContext to derive joiner_secret

Goal: Prove that joiner knows GroupContext

Circular dependency:
joiner_secret <- GroupContext <- welcome_secret <- joiner_secret



```
      commit_secret -> KDF.Extract
                            |
                            V
-                    DeriveSecret(., "joiner")
+                    ExpandWithLabel(., "joiner", GroupContext_[n], KDF.Nh)
                            |
                            V
                      joiner_secret
```

#439 – Identities SHOULD be unique per group

… or rather, unique within the context of a group

Each leaf has a Credential => (identity, signature public key) pair

The current spec allows an identity or signature key to appear multiple times

Proposal is to require that both **identities** and **public keys** be unique

#457 – Clarify ParentHash verification

Obvious problem just a typo:

```
    If R is a blank leaf node, the check fails
```

More broadly: “I suggest to add a more formal description of the parent hash generation and verification (e.g. pseudocode) to reduce ambiguity”

#443 – External commit for resync with PSK

External commit introduces the possibility of a “resync” operation

Remove(old self) + Add(new self) within same external Commit

But “new self” doesn’t have to prove past membership

… notionally, with a PSK derived from an earlier epoch

Should we RECOMMEND / REQUIRE that this be done?

With identity uniqueness, this case is clearly recognizable

… but assumes that a client that has otherwise lost state still has PSK(s)

#XXX - Resolve GroupContext usage ambiguity

Generating and handling commits requires that committer/processor use a few different GroupContexts:

1. Encap/decap an UpdatePath - “provisional” GroupContext, proposals applied

2. Ratcheting forward the key schedule - GroupContext for next epoch

3. Signing the MLSPlaintext of the Commit - GroupContext for last epoch

Propose to clarify, align terminology around **old / provisional / new**

Way Forward

Final TODOs

1. Finish interop testing based on draft-11 (without further changes)
   a. ETA: April?
   b. **EVERYONE GET ANY LAST ISSUES / PRS IN WHILE THIS IS HAPPENING**
2. Issue draft-12 with the last round of changes
   a. ETA: As soon as we're done with interop testing
3. Update implementations and re-validate interop
   a. ETA: A couple of weeks after draft-12
4. Final WGLC and on to the IESG!
   a. ETA: May?