DRAFT-13
Change Log

- TLS syntax updates (including variable-header-length vectors) (*)
- Stop generating redundant PKE key pairs. (*)
- Move validation of identity change to the AS
- Add message/mls MIME type registration
- Split LeafNode from KeyPackage (*)
- Remove endpoint_id (*)
- Reorganize to make section layout more sane
- Forbid proposals by reference in external commits (*)
- Domain separation for KeyPackage and Proposal references (*)
- Downgrade MUST to SHOULD for commit senders including all valid commits
- Stronger parent hashes for authenticated identities (*)
- Move wire_format to a separate tagged-union structure MLSMessage
- Generalize tree extend/truncate algorithms
- Add algorithms for link-based trees
- Forbid self-Update entirely (*)
- Consolidate resumption PSK cases (*)
- 384 Ciphersuite Addition
- Remove explicit version pin on HPKE (*)
- Remove the requirement for Add in external commit (*)
- Use smaller, fixed-size hash-based identifiers (*)
- Be explicit that Credentials can attest to multiple identities (*)
Streamlined vector syntax

TLS specifies the size of a vector length\[\text{opaque vec}<0..2^{32}-1>\];

This leads to more code and interop hassle

Instead we use a varint length\[\text{opaque vec}<V>\];

*Full disclosure: idea stolen from [CTLS](#), which stole from [QUIC](#)*
**LeafNode / KeyPackage split**

struct {
    ProtocolVersion version;
    CipherSuite cipher_suite;
    HPKEPublicKey hpke_init_key;
    Credential credential;
    Extension extensions<8..2^32-1>;
    // Lifetime
    // Capabilities
    // Parent Hash
    opaque signature<0..2^16-1>;
} KeyPackage;

struct {
    HPKEPublicKey public_key;
    Credential credential;
    Capabilities capabilities;
    LeafNodeSource leaf_node_source;
    select (leaf_node_source) {
        case add: Lifetime lifetime;
        case update: struct {}
        case commit: opaque parent_hash<V>;
    }
    Extension extensions<V>;
    opaque signature<V>;
} LeafNode;

struct {
    ProtocolVersion version;
    CipherSuite cipher_suite;
    HPKEPublicKey init_key;
    LeafNode leaf_node;
    Extension extensions<V>;
    opaque signature<V>;
} KeyPackage;

- Split leaf node contents from KeyPackage
  - KeyPackage now only used for async join
- Mandatory extns -> Fields
- Dual signatures on LeafNode & KeyPackage
- Separate init & leaf HPKE public keys
- Update/Commit LeafNode sig covers group_id
No more redundant nodes
Tree-hash-based parent hash

Represent sibling in parent hash by tree hash, not resolution

Bind in tree structure, leaf node contents (e.g., credentials)
**Tree-hash-based parent hash**

Problem: The tree structure changes at the right edge

Need to re-compute tree hashes when verifying a tree on joining

$O(N \log N)$ hashes naïve, $O(N \log \log N)$ with memoization (vs $O(N)$ base case)
Tree-hash-based parent hash

Simulated using updated MLSpp on 2017-era MacBook Pro

- Full tree (no unmerged)
- 50% unmerged
- 75% unmerged
- One path populated (no unmerged)
Almost ready for WGLC...
Over to GitHub...
Onward!
To the IESG!