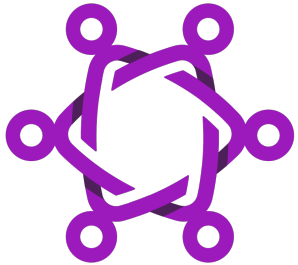


# Web Token Credentials in MLS

# SD-CWTs and SD-JWTs as MLS Credentials

- This is relatively straightforward.
- Currently uses the SD-KBT for SD-CWT. Might be able be to use the SD-CWT directly instead in MLS.
- Works great with pseudonyms.
- SD-CWT credential is implemented

```
struct {
    CredentialType credential_type;
    select (Credential.credential_type) {
        case basic:
            opaque identity<V>;
        case x509:
            Certificate certificates<V>;
        ...
        case sd_cwt:
            opaque sd_kbt<V>;
        case sd_jwt:
            SdJwt sd_jwt;
    };
} Credential;
```

# SD-JWT as Credential

- the canonical format of an SD-JWT looks like this:
- Can convey it "as is", or decomposed and unbase64ed to make it smaller
- Can convey a regular JWT

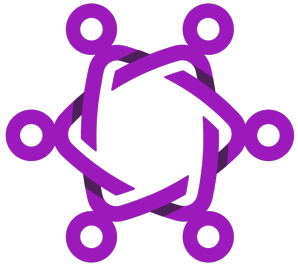

```
struct {
    Bool compacted;
    select (compacted) {
        case true:
            opaque protected<V>;
            opaque payload<V>;
            opaque signature<V>;
            SdJwtDisclosure disclosures<V>;
            opaque sd_jwt_key_binding<V>;
        case false:
            opaque sd_jwt_kb<V>;
    };
} SdJwt;
```

# Encrypted disclosures in SD-CWT

- Allows a disclosure to be ephemerally AEAD encrypted
- In the MLS context, allows a disclosure to be hidden from DS but shared with all (or a subset) of members.

```
<<[
    /salt/   h'bae611067bb823486797da1ebbb52f83',
    /value/  "ABCD-123456",
    /claim/  501    / inspector_license_number /
]>>,
```

```
/ sd_aead_encrypted_claims / 19 : [ / AEAD encrypted disclosures /
    [
        / nonce /      h'95d0040fe650e5baf51c907c31be15dc',
        / ciphertext / h'208cda279ca86444681503830469b705
                         89654084156c9e65ca02f9ac40cd62b5
                         a2470d',
        / tag /        h'1c6e732977453ab2cacbfd578bd238c0'
    ],
    ...
]
```

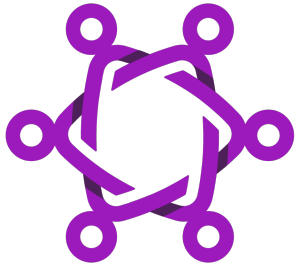


# Other individual drafts

draft-mahy-mls-ext-commit-pp
draft-mahy-mls-new-framed-content-tbs
draft-mahy-mls-semiprivatemessage
draft-mahy-mls-new-content-types-00 —presence / status

# Functionality/Robustness Holes in MLS

- Issues independent of handshake style
  - Member can replay messages within the same epoch
  - Malicious UpdatePath can fork a group ††
  - External Commits make most pending proposals invalid †
- PrivateMessage (DS only does ordering and fanout, GC/tree private to members) — XMTP, Germ, Mozilla
  - DS can't validate any signed message, since DS does not have the GroupContext. Solution: use new FramedContentTBS (signs message with *hash* of GroupContext); share GC hash in AAD of Commit. †
  - Privacy for External Commits and External Proposals †
  - Forward Welcome messages without leaking “user” membership in a specific group
- PublicMessage (smart DS, GroupContext/ratchet tree known by DS) — Cisco, Amazon, Apple, Google, Wire
  - Only your DS should see handshakes — † SemiPrivateMessage
  - DS cannot verify the GroupInfo `extensions` or `signature` for a Commit bundle
  - DS cannot check the validity of the `confirmation_tag` .



# Questions