

# SCITT Receipts as new COSE message type?

([draft-birkholz-scitt-receipts](#))

- Defined in SCITT WG as part of SCITT Transparency Services
- $\approx$  Signed tree head + inclusion proof for countersignature leaves
- Leaf binds `Countersign_structure` for some `COSE_Sign1` message
- Allows embedding of Receipt in unprotected header of `COSE_Sign1`

```
Receipt = [  
  protected: bstr .cbor { * label => value },  
  proof: any  
]
```

- `protected` is part of ledger leaf (countersigner protected header)
- `proof` type depends on ledger type

# SCITT Receipts as new COSE message type?

([draft-birkholz-scitt-receipts](#))

- In principle, receipts can be shoehorned into a signature algorithm
  - Not well received in community
- Better: Receipts are likely different enough to deserve their own type
- Being a new COSE message type would allow re-use of IANA registries
  - Would alg indicate the ledger type?
  - Or new parameters like ledger\_type, sig\_alg, hash\_alg?
- Does this make sense in principle?
- What should we do to fully evaluate this?

# COSE Header parameter for RFC 3161 Time-Stamp Tokens ([draft-birkholz-cose-tsa-tst-header-parameter](#))

[draft-ietf-cose-x509](#): *“The use of X.509 certificates allows for an existing trust infrastructure to be used with COSE.”*

RFC 3161 time-stamp tokens are sometimes used together with X.509 certificates (typically, signed documents or binaries). To avoid proprietary COSE header parameter labels, let's standardise one:

- **Name:** RFC 3161 time-stamp tokens
- **Label:** TBD
- **Value Type:** bstr / [+ bstr]
- **Description:** One or more RFC 3161 time-stamp tokens.