

COSE Key Thumbprint

[draft-isobe-cose-key-thumbprint](#)

Kohei Isobe
Hannes Tschofenig

Background

- Protocols and data formats require use of hashes of public keys.
- Need surfaced in SUIT and TEEP WGs.
 - Use in key derivation function
 - Identification of keys

Isn't there a standard already?

- JWK Thumbprint (RFC 7638)
- keyIdentifier in X.509 Certificate (RFC 5280)

- We need a solution for COSE!

How to calculate

- We used the approach specified for JWK Thumbprint.
 1. Make a COSE_Key Object
 - COSE_Key defined in RFC 9052
 - Use specific values with each algorithms
 2. Apply deterministic encoding
 - Ensures that elements appear in the same order.
 - RFC 8949 provides the deterministic encoding.
 3. Hash the COSE_Key Object

COSE Key Object for Thumbprint

- Asymmetric Key
 - kty is required to identify key type.
 - Using public key parameters only.
 - Contains the parameter lists for each key types.
- Symmetric Key
 - Not supporting in this draft

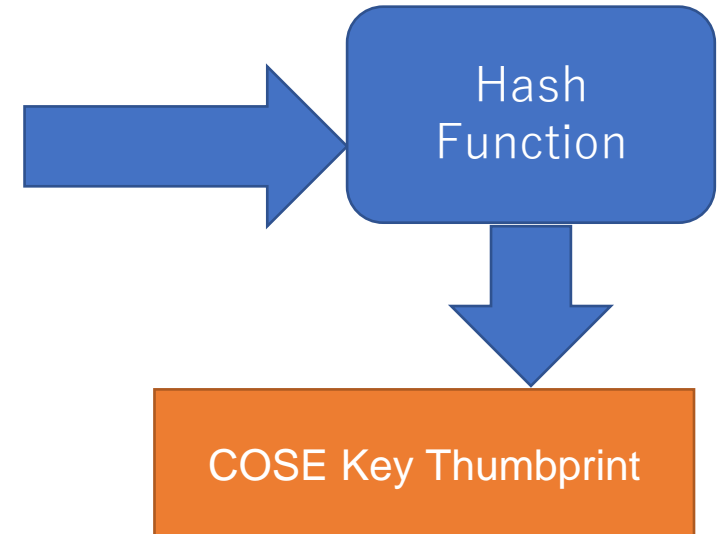
Example

```
{  
  / kty / 1: 2 / EC2 = Elliptic Curve Keys /,  
  / crv / -1: 1 / P-256 /,  
  / y-coordinate / -3:  
  h'1e52ed75701163f7f9e40ddf9f341b3dc9ba860af7e0ca7ca7e9eecd0084d19c',  
  / x-coordinate / -2:  
  h'65eda5a12577c2bae829437fe338701a10aaa375e1bb5b5de108de439c08551d',  
  / kid / 2: 'meriadoc.brandybuck@buckland.example'  
}
```

EC2 Key Materials (labels)

- kty (1)
- crv (-1)
- x (-2)
- y (-3)

```
{  
  1:2,  
  -1:1,  
  -2:h'65eda5a12577c2bae829437fe338701a10aaa375e1bb5b5de108de439c08551d',  
  -3:h'1e52ed75701163f7f9e40ddf9f341b3dc9ba860af7e0ca7ca7e9eecd0084d19c'  
}
```



Next Steps

- Call for adoption started on July 7th, see https://mailarchive.ietf.org/arch/msg/cose/8KehW_5s2icYIU_HZZE5PbtWe_G8/
- Submit WG document
- Issue WGLC since there are no open issues.