Public Key-derived secrets for MAC-based signatures

draft-bastian-jose-dvs

Stefan Santesson, Micha Kraus
Paul Bastian, Peter Lee Altmann (not present)

IETF 123

Previous Work

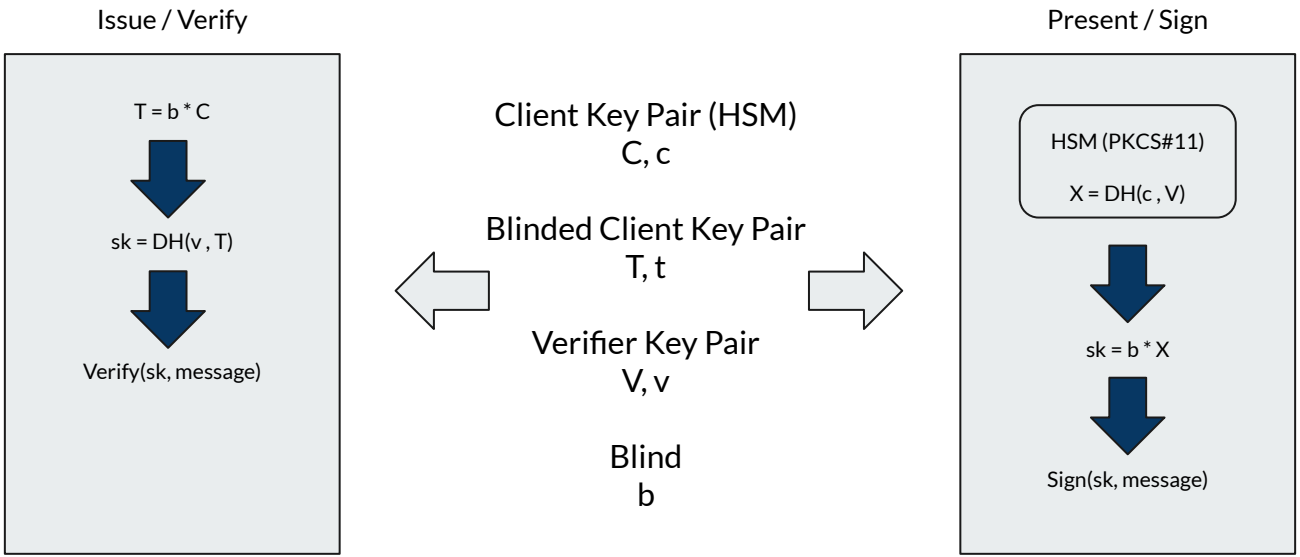
- Presentation of Designated Verifier Signatures at IETF 121
- Motivated by enabling repudiable signatures
- Removed HPKE option
- Added Stefan and Peter as co-authors

Focus

Define how to embed information in the JWS header to derive a secret key for Mac-based signatures

New Motivation

HSM supported key proof with blinded key



Key proof with blinded key

- Relevant for EU wallets (eIDAS) providing verifiable presentations bound to blinded PoP keys
- Relevant for use with SD-JWT as verifiable presentation format
- Can be done with ECDSA, but requires modified HSM (add then sign)
- HMAC signatures allows use of standard PKCS#11 HSM (Diffie-Hellman)
- HMAC signatures natively supported by mDoc / mDL (ISO 18013)
    - Fully specifies HMAC key derivation from public signer key
- HMAC signatures supported by SD-JWT
    - Key derivation from public signer key is NOT defined

Two Approaches

1. new fully-specified JOSE algorithm (in the current draft)
2. existing JOSE algorithms

Approach 1: fully-specified algorithm

Header:

```
"typ": "JWT"
"alg" : "DVS-P256-SHA-HS256",
"rpk": {
  "jwk": {
    "kty": "EC",
    "crv": "P-256",
    "x": "f83OJ3D2xF4YMzMvwPs6NLepWJPOcC5AxHkZkU8Gx1I",
    "y": "x_FEzRu9m_5JJMK0pYdIYt_UswsXz1fDNk0-uxs111c"
  }
},
"nonce": "efasad4w6123…"
"jwk": {..jwk of signing party…}
```

Signature:

Base64-encoded MAC

Approach 2: JOSE header + alg=HS256

- No need to specify additional signature algorithms. HS256, HS384 and HS512 works just fine
- Define new header parameter to specify key derivation from signer and verifier key pairs
  - Resulting derived key is used as input to HS256/HS484/HS512
- Compatible with existing JWS tools
  - Non critical extension with key derivation params
- Allows definition of multiple key derivation functions, but:
  - Default key derivation based on Diffie-Hellman + HDKF should suffice for a majority of cases
- Avoid KEM - Lacks fundamental property to bind the result to signer private key
  - Using HPKE is a valid use-case for JWE, but is outside the scope of this document.

Approach 2: JOSE header - Example

```
"alg" : HS256,
"pkds": {
  "suite": "ECDH-HKDF-SHA256",
  "rpk": {
    "jwk": {
      "kty": "EC",
      "crv": "P-256",
      "x": "f83OJ3D2xF4YMzMvwPs6NLepWJPOcC5AxHkZkU8Gx1I",
      "y": "x_FEzRu9m_5JJMK0pYdIYt_UswsXz1fDNk0-uxs1l1c"
    }
  },
  "spk": {
    "kid": "signer-key-id"
  },
  "params": {
    "info": "SW5mbw",
    "salt": "c2FsdA",
  },
  "length": 32
}
```

Two approaches - comparison

Fully specified algorithm

- More flexible
- Overlaps with defined algs like HS256
- Not compatible with current JWS tools

HS256 + key derivation header

- Less flexible
- Using standard defined algorithms
- Compatible with current JWS tools

Links

Datatracker -> https://datatracker.ietf.org/doc/draft-bastian-jose-dvs

Git Repository -> https://github.com/paulbastian/draft-bastian-jose-dvs

Current Editors Copy -> https://paulbastian.github.io/draft-bastian-jose-dvs/draft-bastian-jose-dvs.html