A Proposed Standard for Entity Attestation
draft-mandyam-eat-00

Laurence Lundblade

November 2018
EAT Overall System

**Entity Manufacturer** (e.g. chip or device vendor)

Manufacturing process to put seed, private and/or public key, cert or other on device (this is intentionally open-ended)

**Entity (e.g., Chip, Device...)**

- Immutable private key for signing. Stored securely on device
- Claims
- Token creation & signing

**EAT Token**

- Key ID or Cert
  - Nonce
  - Claim 1
  - Claim 2
  - ...
- Signature

**Relying Party (e.g., Server / Service)**

- Interactions to obtain public key and related data for token verification.
- Signature and public key verification process
- Claims

**EAT Target for standardization**

Device status & characteristics determination
FIDO Phone Use Case

Phone manufacturer

FIDO Alliance
Metadata service

ECDSA private key and X.509 certificate provisioned by phone maker

Obtain root and/or CA certificates to chain phone certificate to trusted root.

Phone

Private key for signing. Stored securely on device

Claims, new pub key

Token creation & signing

MakeCredential

Certs for verification
- Nonce
- New public key
- Claim 2
- ...

Signature

Relying Party (e.g., Server / Service)

Signature and public key verification process

Claims, new pub key

Enroll new phone as authenticator for user
Android Keystore

Phone manufacturer

Buckets of attestation key pairs

Google Service

Obtain root and/or CA certificates to chain phone certificate to trusted root.

ECDSA private key and X.509 certificate provisioned by phone maker

Phone

Private key for signing. Stored securely on device

Claims, new pub key

Token creation & signing

MakeCredential

Certs for verification
- Nonce
- New public key
- Claim 2
- ...
Signature

Relying Party (e.g., Server / Service)

Signature and public key verification process

Claims, new pub key

Enroll new phone as authenticator for user

Enroll new phone as authenticator for user
Payment Risk Engine

ECDSA private key and X.509 certificate provisioned by phone maker

Phone

Private key for signing. Stored securely on device

Token creation & signing

Lots of claims about device

MakeCredential

Cert for verification

• Nonce
• Claim #1
• ...
• Claim #41...

Signature

Phone manufacturer

Obtain root and/or CA certificates to chain phone certificate to trusted root.

Relying Party (e.g., Server / Service)

Signature and public key verification process

Lots of claims

Payment risk engine
Home Appliance & Web Service

ECDSA private key and X.509 certificate provisioned by appliance maker

Appliance (low security CPU)
- Private key for signing. Stored securely on device
- Device identity (serial number), appliance telemetry
- Token creation & signing

Request Service
- Certs for verification
  - Nonce
  - Device identity
  - Appliance telemetry
  - …
  - Signature

Appliance Manufacturer

Appliance Manufacturer Cloud Service
- Obtain root and/or CA certificates to chain phone certificate to trusted root.
- Signature and public key verification process
- telemetry
- Track performance of appliances
Enrollment of Low Cost IoT for Device Management

IoT Device Maker
- Deterministic ECDSA key gen

Data base key seeds

Nonce
- Obtain public key for signature verification

Request Service
- Key ID
  - Nonce
  - Device identity
  - Device key
- Signature

IoT Device
- Deterministic ECDSA key gen; make key ID
- Device identity (serial number), device key
- Token creation & signing

256-bit secret key seed

Data base of IoT devices

Device identity and key
- Signature and public key verification process
FIDO Security Key

Security Key with Embedded Secure Element
- Private key for signing. Stored securely on device
- Claims, new pub key
- Token creation & signing

Persona Computer
- Nonce

MakeCredential
- Cert for verification
  - Nonce
  - New public key
  - Claim 2
  - …
  - Signature

Relying Party (e.g., Server / Service)
- Signature and public key verification process
- Claims, new pub key
- Enroll new phone as authenticator for user

Security Key Vendor

FIDO Alliance Metadata service
- Obtain CA certificates to chain entity certificate to trusted root.

ECDSA private key and X.509 certificate provisioned by phone maker
Primary Standardization Goal is Semantic Interoperability of Claims

- Main types of claims to standardize:
  - Device Identity
  - Measurement
  - Device boot, debug and configuration state
  - Measurement and run time integrity checks
  - Geographic location
  - Device SW and HW versions
  - Public key created on the device - Keystore, IoT and FIDO use cases

- Claims should be generally applicable:
  - Not specific to TPM, TrustZone, SGX, Secure Element...
  - Not require any particular level of device security
    - Works with high-security device like Secure Elements and TPMs and low-security devices with nothing special at all.
EAT Format (basically CWT)
draft-mandyam-eat-00

Overall structure: COSE_Sign1

- **Algorithm** -- Examples: ECDSA 256, RSA 2048, ECDAA
- **Signing Scheme** -- Examples: IEEE IDevID, EPID, X.509 Hierarchy

**Key ID** -- identifies the key needed to verify signature

**Certs** (optional) -- to chain up to a root for some signing schemes

**Signed payload**

- CBOR formatted map of claims that describe device and its disposition
- Few and simple or many, complex, nested...
- All claims are optional -- no minimal set
- The format and meaning of a basic set of claims should be standardized for interoperability
- Should be adaptable to cover many different use cases from tiny IoT devices to complex mobile phones
- Privacy issues must be taken into account

**signature** -- Examples: 64 byte ECDSA signature, 256 byte RSA signature

- COSE format for signing
- Small message size for IoT
- Allows for varying signing algorithms, carries headers, sets overall format

- CBOR format for claims
- Small message size for IoT
- Labelling of claims
- Very flexible data types for all kinds of different claims.
- Translates to JSON

- Signature proves device and claims (critical)
- Accommodate different end-end signing schemes because of device manufacturing issues
- Privacy requirements also drive variance in signing schemes
Example Token

CBOR diagnostic representation of binary data of full signed token

```
/ protected / << {
    / alg / 1: -7 / ECDSA 256 /
} >>,
/ unprotected / {
    / kid / 4: h'4173796d6d65747269634543445341323536'
},
/ payload / << {
    / UEID / 8: h'5427c1ff28d23fbd1f29c4c7c6a55',
    / secure boot enabled / 13: true
    / debug disabled / 15: true
    / integrity / -81000: {
        / status / -81001: true
        / timestamp / 21: 1444064944,
    },
    / location / 18: {
        / lat / 19: 32.9024843386,
        / long / 20: -117.192956976
    }
},
} >>,
/ signature / h'5427c1ff28d23fbd1f29c4c7c6a555e601d6fa29f9179bc3d7438bacaca5ac08c8d4d4f96131680c429a01f85951ecee743a52b9b63632c57209120e1c9e30'
```

Payload Translated to JSON

- Integer labels mapped to strings
- Binary data base 64 encoded
- Floating point numbers turned into strings

```
{
    "UEID": "k8if9d98Mk979077L38Uw34kKFRHJgd18f==",
    "secureBoot": true,
    "debugDisable": true,

    "integrity": {
        "status": true,
        "timestamp": "2015-10-5T05:09:04Z",
    },

    "location": {
        "lat": "32.9024843386",
        "long": "-117.192956976",
    }
}
```
COSE Signing Scheme Flexibility

• Many standard algorithms already supported
  ◦ RSA, ECDSA and Edwards-Curve Signing (public key)
  ◦ HMAC and AES-based MACs (symmetric key)

• Extensible for future algorithms
  ◦ [IANA registry](https://www.iana.org) for algorithms exists today

• Extensible for special case schemes
  ◦ Proprietary simple HMACs schemes, perhaps HW based
  ◦ Possibly Intel EPID
  ◦ (non-standard algorithms will of course be less interoperable)
Privacy

• Entity Attestation Tokens are intended for many use cases with varying privacy requirements
  ◦ Some will be simple with only 2 or 3 claims, others may have 100 claims
  ◦ Simple, single-use IoT devices, have fewer privacy issues and may be able to include claims that complex devices like Android phones cannot

• Options for handling privacy
  ◦ Omit privacy-violating claims
  ◦ Redesign claims especially to work with privacy regulation
  ◦ Obtain user permission to include claims that would otherwise be privacy-violating

• Some signing schemes will be privacy-preserving (e.g. group key, ECDAA) and some will not