Signature Validation Token

Security Dispatch IETF 109

Stefan Santesson  
stefan@aaa-sec.com

Russ Housley  
housley@vigilsec.com
Signature Validation Token History

- Swedish government funded research project for Archivable electronic signatures
- Adopted by the Swedish Agency for Digital Government (DIGG)
- Developed as open source
- Refined and implemented for eduSign
- Approaching the IETF for standardization
Goal

Simple solution for validating signatures in a distant future
Important requirements

• Predictable outcome of future signature validation
• Avoid cascading evidence collection
• Avoid size explosion
• Avoid repeated storage of large common validation data
• Easy to implement
• Evidence renewal without significant increase of complexity
• Fast verification
• Off-line: Possibility to validate without access to external on-line services
• Compatible with current document parsers and signature validation software.
Running Code - In production

**eduSign**

**eduSign** - secure digital signature and validation

To sign

This service can be used to upload and sign PDF documents or XML documents. This is done easily by performing the following steps:

1. Upload documents to sign
2. Agree to sign
3. Identify yourself with the appropriate SWAMID electronic ID
4. Download signed document

For more information read further here.
Signature validation requires supporting evidence

- **While signature certificate is valid:**
  - Is certificate revoked? And if revoked:
    - When was certificate revoked?
    - Was signature created before revocation time?

- **After certificate expires:**
  - A time when the signature existed
  - The validity status at that time

- **When algorithms are no longer trusted:**
  - A time when the signature existed
  - The validity status at that time
  - The data that was signed (and the signature that signed it)
  - The certificates used to validate the signature
  - [Results from prior validations]
The R number for evidence reproduction

(When each supporting evidence requires more than one new supporting evidence)
Complexity of long-term signature validation is greatly reduced if we can limit the number of supporting evidence and the evidence R number.
IN FACT....
We can reduce it to one piece of external evidence
Signature Validation Token

Signed Document

Signature

Signed attributes
• Hash over signed data
• Algorithms
• Transforms etc

Signature value

Certificates

Signed JSON Web Token with SVT claims
• Issuer
• Time
• Algorithms
• Expiry?
• Signature claims
  • Signed data reference
    • Hash over signed content
  • Signature reference
    • Hash over signature context
    • Hash over signature value
  • Certificate reference
    • Verified certs or hash over verified certs
  • Verified times
    • List of verified evidence of time
• Validation results
  • Policy
    • Result
Simple compact format

SVT JWT Calims

```
{
    "aud": "https://example.com/audience1",
    "iss": "https://swedenconnect.se/validator",
    "iat": 1584703056,
    "jti": "45d4f765d1f9817f0c3046f5ad9491",
    "sig_val_claims": {
        "sig": [],
        "sig_val": [],
        "msg": "Passed basic signature validation",
        "res": "PASSED",
        "pol": "http://id.swedenconnect.se/svt/sigval-policy/chain/01"
    },
    "sig_ref": {
        "sig_hash": "mC0ReA...Vqdw==",
        "sb_hash": "DNn...aXg=="
    },
    "signer_cert_ref": {
        "ref": "fIdr...UnoA==",
        "type": "chain_hash"
    },
    "sig_data_ref": {
        "ref": "0 74697 79699 37821",
        "hash": "qmIjbB...5ihujvw=="
    },
    "time_val": [],
    "ver": "1.0",
    "profile": "PDF",
    "hash_algo": "http://www.w3.org/2001/04/xmlenc#sha512"
}
```
XML SVT example (ECDSA with SHA512)
Implementation profiles for PDF, XML, JWS, ...

- **PDF Document**
  - PDF Document data
  - Signature
  - Context
  - Signature
  - Certificates
  - Appended PDF Document data
  - Signature
  - Context
  - Signature
  - Certificates
  - Document timestamp
  - SVT

- **XML Document**
  - XML Document data
  - Signature
  - Context
  - Signature
  - Certificates
  - SVT

- **JWS Document**
  - Protected Headers
  - Certificates
  - context
  - Unprotected Headers
  - SVT
  - Payload
  - Signature
Resources

• Current drafts:
  • https://datatracker.ietf.org/doc/draft-santesson-svt/
  • https://datatracker.ietf.org/doc/draft-santesson-svt-pdf/
  • https://datatracker.ietf.org/doc/draft-santesson-svt-xml/

• IETF Draft development
  • https://github.com/swedenconnect/IETF-SVT

• Open Source
  • Basic SVT library:
    • https://github.com/idsec-solutions/sig-validation-svt
  • Basic library for issuing SVT and signature validation with SVT (Java):
    • https://github.com/idsec-solutions/sig-validation-base

• Test SVT issuing validation service
  • https://sandbox.swedenconnect.se/sigval/
Why IETF?

- Based on the IETF JWT format
- Can support IETF signature formats (CMS, JWS, ...)
- No other standards organization is doing this
- IETF has done similar work in the past
- It is a very important subject. Archival of signed electronic documents provide huge cost savings with greatly improved performance.
- We think LAMPS could be a suitable home for this work.
Questions