Signing HTTP Messages

draft-ietf-httpbis-message-signatures

HTTP Working Group Virtual Interim Meeting

October 19, 2020

Durable Signatures Over HTTP Message Parts

```
GET / HTTP/1.1
Host: httpwg.org
Accept: text/html
Date: Tue, 20 May 2020 20:51:35 GMT
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Authorization: Bearer 1234abcd5678efab
```

Create & Sign Signature Input

```
*created: 1590007895
*request-target: GET /
host: httpwg.org
authorization: Bearer 1234abcd5678efab
```

Attach Signature to Message

```
Signature-Input: sig1=(*created,
  *request-target, host, authorization);
  keyId="test-key-a"; created=1590007895
Signature: sig1=:1234567890abcdef...:
```

Status

- Draft lapsed… (oops, publishing update this week)
- Fixed minor issues called out in feedback on -00
- Debated creation/expiration time constraints in Git comments
- Adopted Structured Fields!

Structured Field Usage

```
Signature-Input: sig1=(*request-target,
  host, authorization); keyId="key-a";
  created=159000789

Signature: sig1=:AbCd1234...==:
```

Structured Fields: Two Header Fields

- Signature-Input
    - Dictionary of Lists of Tokens
    - Key: signature identifier
    - Value: covered content
    - Parameters: other metadata
- Signature
    - Dictionary of Byte Sequences
    - Key: signature identifier
    - Value: signature

```
Signature-Input: sig1=(
    *request-target, host,
    authorization);
  keyId="test-key-a";
  created=159000789

Signature: sig1=:AbCd1234...==:
```

Signing Individual Dictionary Members

*<lowercased field name>***:***<member name>*

```
# Given Header field:
X-Dictionary: a=1, b=2, c=3

# Example covered content:
x-dictionary: a=1, b=2, c=3
x-dictionary:a: a=1
x-dictionary:b: b=2
```

Signing List Prefixes

*<lowercased field name>***:***<member count>*

```
# Given Header field:
X-List: (a, b, c, d)

# Example covered content:
x-list: (a, b, c, d)
x-list:1: (a)
x-list:3: (a, b, c)
```

Sign Your Own Input

```
Signature-Input: sig1=(*request-target,
  signature-input:sig1);
  keyId="key-a"; created=159000789

Signature: sig1=:AbCd1234...==:
```

Multiple Signatures

```
Signature-Input: sig2=(signature:sig1,
  x-forwarded-for);
  keyId="key-b"; created=159000789

Signature: sig2=:AbCd1234...==:
```

Problems Solved

- No more confusing "headers" parameter name
- No more bespoke header field value format
- All signature parameters can now be signed
- Support multiple signatures
  - over different content
  - with different keys
- Signing parts of structured headers

Creation Time and Expiration Time

- Expiration Time is:
  - Signer's recommendation to the verifier
  - Limit of signer's accountability

- Verifier MAY enforce a higher or lower expiration time
  - Account for clock skew
  - Verification in async workflows
  - Verifier has tighter requirements (e.g., compliance regimes)

Next Steps/Open Items

- Clean up alg and keyId confusion
- Alignment with Web Packaging's Signed Exchanges
- Signature input format (bespoke or not?)
- Improve serialization rules (e.g., % encoding, collapsing whitespace)
- More content identifiers (*method, *path, *query, …)