HTTP Message Signing

OAuth Working Group
IETF 94, Yokohama, 2015

Justin Richer

What is it?

- Detached signature for HTTP requests
- To be used with PoP tokens

How does it work?

- Hash components of HTTP message
- List which components were hashed
- Cover hashes and lists in a JWS signature
  - Sign with PoP token key

Design Goals

- Re-use JOSE
- Completely deterministic
- Robust against HTTP request transformations

Why something new?

- OAuth 1.0 is fragile and awkward
- JWS would need to replicate the whole HTTP message
  - SOAP much?
- JWS detatched signatures can't carry appropriate metadata
  - Specifically, "What did you sign in what order?"
- AWS signature method is API-specific

Let’s take a closer look…

This presentation fills in some bits missing from the draft

Caveat loquitur:

It doesn’t actually work yet

PoP Token

access_token value:

98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y

PoP key value (EC):


```
{
 "kty": "EC",
 "d": "D8lA_TUGXddzksXKOtOrUNqWgw-5-jyFi6SyCKBF5SI",
 "crv": "P-256",
 "x": "pevswYm6-mAe2H4zOzyFe_rv3gL68XJy8_Zaj-OX9F4",
 "y": "YyzJiAuHNUk1zaTeRwfVcl8FaLT49EL350Aaipv22B4"
}
```

HTTP Message

```
GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Signed Message Content


```
{
  “at”:
“98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262

}
```

HTTP Message

```
GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Signed Message Content


```
{
  “at”:
“98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”
}
```

HTTP Message

GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

Signed Message Content


```
{
  “at”:
“98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”
}
```

HTTP Message

GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

Hash Generation

MD5(
  “foo=bar&baz=wat”
)

=>

92d42038006dba95d0c501951ac5b5eb

Signed Message Content

```
{
  “at”: “98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”,
  “q”: [[“foo”, “bar”],
        “92d42038006dba95d0c501951ac5b5eb”]
}
```

HTTP Message

GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

Hash Generation

MD5(
  “accept-language: en-us\nconnection: Keep-Alive”
)

=>

12b3b4240b9d9c98aab16baf1e30d6cb

Signed Message Content

```
{
  “at”: “98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”,
  “q”: [[“foo”, “bar”],
        “92d42038006dba95d0c501951ac5b5eb”],
  “h”: [[“accept-language”, “connection”],
        “12b3b4240b9d9c98aab16baf1e30d6cb”]
}
```

Put it into a JWS

(Details left to the reader)

HTTP Message

GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Authorization: OAuthPoP ejy….

Validating the message

Outgoing HTTP Message

GET /hello?foo=bar&baz=wat HTTP1.1
Host: api.example.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Authorization: OAuthPoP ejy….

Incoming HTTP Message

GET /hello? baz=wat&foo=bar&view=default HTTP1.1
Host: api.example.com
Connection: Keep-Alive
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Forwarded: for=192.0.2.60; proto=http;
by=203.0.113.43
Authorization: OAuthPoP ejy….

Stuff changed!

Incoming HTTP Message

GET /hello? baz=wat&foo=bar&view=default HTTP1.1
Host: api.example.com
Connection: Keep-Alive
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Forwarded: for=192.0.2.60; proto=http;
by=203.0.113.43
Authorization: OAuthPoP ejy….

Incoming HTTP Message

GET /hello? baz=wat&foo=bar&view=default HTTP1.1
Host: api.example.com
Connection: Keep-Alive
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Forwarded: for=192.0.2.60; proto=http;
by=203.0.113.43
Authorization: OAuthPoP ejy….

Pretty sure something's going to get screwed up?

Don't sign it in the first place!

(Part of API documentation?)

Unpack the JWS


```
{
  “at”: “98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”,
  “q”: [[“foo”, “bar”],
        “92d42038006dba95d0c501951ac5b5eb”],
  “h”: [[“accept-language”, “connection”],
        “12b3b4240b9d9c98aab16baf1e30d6cb”]
}
```

Check the signature and scope

```
{
  “at”: “98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”,
  “q”: [[“foo”, “bar”],
        “92d42038006dba95d0c501951ac5b5eb”],
  “h”: [[“accept-language”, “connection”],
        “12b3b4240b9d9c98aab16baf1e30d6cb”]
}
```

Compare Stuff

```
{
  “at”: “98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”,
  “q”: [[“foo”, “bar”],
        “92d42038006dba95d0c501951ac5b5eb”],
  “h”: [[“accept-language”, “connection”],
        “12b3b4240b9d9c98aab16baf1e30d6cb”]
}
```

HTTP Message

GET /hello? baz=wat&foo=bar&view=default HTTP1.1
Host: api.example.com
Connection: Keep-Alive
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Forwarded: for=192.0.2.60; proto=http;
by=203.0.113.43
Authorization: OAuthPoP ejy….

Compare Query

```
{
  “at”: “98yghgfr567uiko987ytrde45tyhjkoyre456yhji987y”,
  “ts”: 1446622262,
  “m”: “GET”,
  “u”: “api.example.com”,
  “p”: “/hello”,
  “q”: [[“foo”, “bar”],
        “92d42038006dba95d0c501951ac5b5eb”],
  “h”: [[“accept-language”, “connection”],
        “12b3b4240b9d9c98aab16baf1e30d6cb”]
}
```

HTTP Message

GET /hello? baz=wat&foo=bar&view=default HTTP1.1
Host: api.example.com
Connection: Keep-Alive
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Forwarded: for=192.0.2.60; proto=http;
by=203.0.113.43
Authorization: OAuthPoP ejy….

Hash Generation

```
MD5(
  “foo=bar&baz=wat”
)

=>

92d42038006dba95d0c501951ac5b5eb
```

Same thing for headers

Weak points

- Normalization scheme
  - Separation characters / boundary attacks
  - Easy to get wrong
- Encoding issues?
- Repeated query and header values
- Order-dependent APIs are not covered

We need implementations!

- Where's the running code?
  - We don't know what's missing yet
- IETF Hackathon topic?