Identity Changes

IETF 89, London, RTCWEB
Martin Thomson

Issue 1: Username

- API has three options to setIdentityProvider:
  - IdP name, IdP protocol, and user name (hint)
- Proposal: add username to the message, like so:

```
{
  "type": "SIGN", "id": "12", "origin": "https://example.org",
  "message": "...the binary blob...",
  "username": "user@example.com"
}
```

https://github.com/ekr/ietf-drafts/pull/16

Issue 2: User Login

- Draft currently requires the IdP to interact with the user to log them in if they can't authorise the "SIGN" message
- This doesn't work very well in practice
  - Breaks the sandbox
  - Makes the process more brittle
- Proposal: LOGINNEEDED message with a URL that the application can load to allow the user to log in

```
{
  "type": "LOGINNEEDED", "id": "12",
  "error": "Signature verification failed"
  "loginUrl": "https://login.example.com/?somecontextmaybe"
}
```

https://github.com/ekr/ietf-drafts/pull/13

Issue 3: Multiple Fingerprints

SDP!

```
v=3
o=no
s=
c=IN IP4 example.com
t=2 3
a=identity:identityassertiongoeshere
…
m=audio 9 blah…
a=fingerprint:md5 8ad287bf9a4b0c3a256d1f4f7cd0a8df
…
m=video 0 blah…
a=fingerprint:md5 8e532b772cb0e033d59c81801a2efa3e
…
```

Assertion:

```
{
  “fingerprint”: {
    “algorithm”: “md5”, “digest”: “8…oops”
  }
}
```

Issue 3: Multiple Fingerprints

- Option 1: Do nothing
  - Not important for browsers; keep things simple
- Option 2: Multiple identity assertions, same identity
  - Create a different identity assertion for each fingerprint
- Option 3: Include multiple fingerprints
  - Have the assertion cover all of the fingerprints in use
  - Maybe make `a=identity` an exclusively session-level attribute

https://github.com/ekr/ietf-drafts/pull/13

4: Fingerprint Algorithm Mismatch

- In theory—it's not specified—it is possible for the hash algorithm in `a=fingerprint` and the identity assertion to be different
  - Validating this blocks `setRemoteDescription()`, maybe
- Proposal: the algorithm in the assertion MUST match what is in SDP

also: https://github.com/ekr/ietf-drafts/pull/13

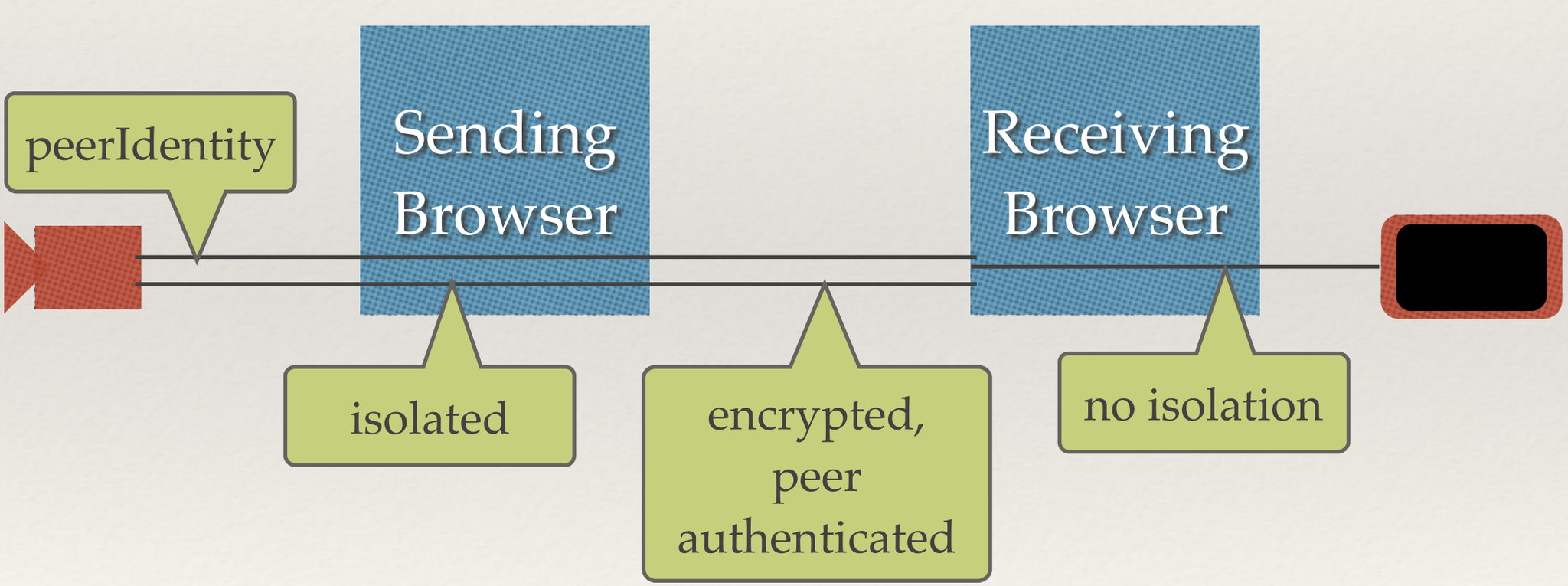
Issue 5: Validating on Servers

- The IdP stuff is geared toward browsers
  - Sure, you might be able to whip up a sandbox using gecko or chromium code, but it isn't that easy and it probably scales poorly
  - Requesting assertions might be tricky for a server, it would have to offer the IdP credentials
  - Validating on the other hand could be handy
- Proposal: Add a mapping whereby the protocol can be used with HTTP POST
  - The objects aren't JSON, but they can be
  - The exchanges are request/response

https://github.com/ekr/ietf-drafts/pull/14

Issue 6: Stream Isolation

- A receiver is unable to distinguish between streams that are isolated at the source and regular streams



6: Preserving Isolation

- Need to preserve the isolation property
  - …securely
- Option 1: propose an extension to DTLS to carry this
- Option 2: add extra signalling