PASSporT divert

IETF **100** (Singapore) STIR WG

Nov2017

draft-ietf-stir-passport-divert-01

- A feature many people have asked about
  - How do we handle **retargeting**?
  - To header field of SIP is signed by PASSporT
    - Original value may be lost with retargeting
- We define a special Identity header track it
  - With its own “ppt” – “**div**” for “divert”
- Different from History-Info and Diversion?
  - Yes, as it is signed by the original destination domain
  - Moreover, it only captures “major” changes
    - Thanks to our canonicalization procedures

Inverting the signer

- *A diverting auth service takes an existing PASSporT, moves the "dest" to "div," and populates "dest" with the new target*
- An Identity header with "div" always points to some prior Identity header
  - Though that header may in turn contain a div…
  - Chains back to an original assertion
- Instead of signing for the "orig" value, the auth service for "div" signs the "dest"
  - So relying parties get a direct cryptographic attestation that the original destination domain authorized the new target

Original vs. Divert Passport

Header:

```
{ "typ":”passport",
  "alg":”ES256“,
         "x5u":"https://www.example.com/cert.pkx" }
```

Claims:

```
{ ”orig":{“uri”:”alice@example.com”},
  ”dest":{“uri”:”firsttarget@example.com”}, <- original target
                  "iat": 1443208345 }
```

Original
PASSporT

Header:

```
{ "typ":”passport",
  "alg":”ES256“,
  "ppt":”div“,
  "x5u":"https://www.example.com/cert.pkx" }
```

Claims:

```
{ ”orig":{“uri”:”alice@example.com”},
  ”dest":{“uri”:”secondtarget@example.com”},  <- new target
                  "iat": 1443208345,
          “div”:{“uri”:”firsttarget@example.com”} }  <- original target
```

Added
when
retargeting

A wrinkle

- Out-of-band creates some new requirements
  - In OOB the called party asks the CPS for calls targeting its own credential (basically its own called party number)
  - How to correlate "divert" PASSporTs in the CPS with original PASSporTs?
    - In OOB both would be encrypted
    - A called party can't decrypt a PASSporT encrypted to a previous target
- How to handle this? A few options
  - Retargeting entity could encrypt a copy of the old PASSporT with the new target's key, maybe
    - Then in OOB there would be multiple PASSporTs encrypted to the same target that the called party could correlate
  - The current draft proposes a nested PASSporT
    - Optionally in the "opt" claim - full form only

Nested “divert” Passport

Header:

```
{ "typ":”passport",
  "alg":”ES256“,
  "ppt":”div“,
  "x5u":"https://www.example.com/cert.pkx" }
```

Retargeting entity
Will store this
In the CPS

Claims:

```
{ ”orig":{“uri”:”alice@example.com”},
  ”dest":{“uri”:”secondtarget@example.com”},  <- new target
                      "iat": 1443208345,
            “div”:{“uri”:”firsttarget@example.com”}   <- original target
  “opt”:"eyJhbGciOiJFUzI1NiIsInR5cCI6InBhc3Nwb3J0IiwieDV1I     \
      joiaHR0cHM6Ly9jZXJ0LmV4YW1wbGUub3JnL3Bhc3Nwb3J0LmNlciJ9.eyJ \
kZXN0Ijp7InVyaSI6WyJzaXA6YWxpY2VAZXhhbXBsZS5jb20iXX0sImlhdC \  <- original ppt
I6IjE0NDMyMDgzNDUiLCJvcmlnIjp7InRuIjoiMTIxNTU1NTEyMTIifX0.r \
q3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1VOgFWSjHBr8Qjpjlk-cpFYpFYs \
ojNCpTzO3QfPOlckGaS6hEck7w”}
  }
```

Which way to go?

- Could do the re-encryption of the original PASSporT by retargeting entity
- From a design perspective, do we want to allow both nested and unnested as options?
  - “opt” for some use cases and separate PPTs for others?
  - For ordinary in-band retargeting, nesting might make Identity headers bloated
- Might be useful for more than just OOB
  - If full form encrypted PASSporTs were ever carried in-band, we’d run into similar problems
  - Extensions like “rcd” might actually motivate that

Issues

- This is pretty close
- Need resolution on the nested/unnested issue

- But other than that, people seem to need this and we should move it along