Incremental Steps towards HTTP for ICN

Pasi Sarolahti
Jussi Kangasharju
Jörg Ott

82nd IETF – ICN Side Session

Starting Point

- An established infrastructure for content delivery
  - With all their tricks for improvement
- User front ends (web browsers)
- HTTP as a common denominator
  - and as the “API” for the front ends
- Familiar naming conventions
  - With all their pros and cons

An Initial Step with HTTP

- Address some of the security issues
- Keep distribution substrate & caching
- Maintain backwards compatibility

Looking at a GET Request

GET / HTTP/1.1
Host: www.ietf.org
User-Agent: Mozilla/5.0 …
Accept*: …
…

HTTP/1.1 200 OK
ETag: …
Server: Apache/2.2.10 …
Last-Modified: …
…
Content-Type: text/html

<HTML body goes here>

Basic idea:

- Add a Secure-Name: that points to the principal and a reference name for the resources and may be independent of the Request URI
- Used for identification in caches and content routing

Secure Name (DONA-style)

GET /index.html HTTP/1.1
Host: www.ietf.org
Secure-Name: a1b2c4…/index.ht
User-Agent: Mozilla/5.0 …
Accept*: …
…

HTTP/1.1 200 OK
Secure-Name: a1b2c4…/index.htm
ETag: …
Server: Apache/2.2.10 …
Last-Modified: …
…
Content-Type: text/html

<HTML body goes here>

Secure-Name: a1b2c4…/index.html

HASH (Public Key (ietf.org)) Resource id assigned by ietf.org

Securely Binding Content

GET /index.html HTTP/1.1
Host: www.ietf.org
Secure-Name: a1b2c4…/index.ht
User-Agent: Mozilla/5.0 …
Accept*: …
…

HTTP/1.1 200 OK
Secure-Name: a1b2c4…/index.htm
ETag: …
Server: Apache/2.2.10 …
Last-Modified: …
…
Public-Key: <algo>:jkdhks313202…
Signature: <algo>:c1ashkja32…
Content-Type: text/html

<HTML body goes here>

- May be validated in intermediaries
- Should be validated by endpoints

Obtaining a Secure-Name

- Origin server (non-secure request)
- Search engines
- Extended HREF attribute in <a> elements
- Referrals, e.g., in social networks
- Cached principal in the endpoints
  - Secure-Name: a1b2c4

Interactions

- Non-enhanced request — Req. URI, Host
  - Client can authenticate the response
  - Use external infrastructure to check public key/cert
- Request with principal only — Req. URI, Host, Sec.-Name
  - Client validates the origin of the resource
  - Request routing and caching based upon request URI
- Request with full Secure-Name — Req. URI, Host, Sec.-Name
  - Client can validate the origin of the resource and that the requested resource is returned
  - Request routing and caching based upon Secure-Name

Concluding remarks

- HTTP extension keeps present infrastructure
- Allows for diverse security models at the ends
- Internal operation of the net may evolve