Open Screen Protocol

TPAC 2022 - September 16, 2022

Mark A. Foltz
mfoltz@google.com / Google Inc.

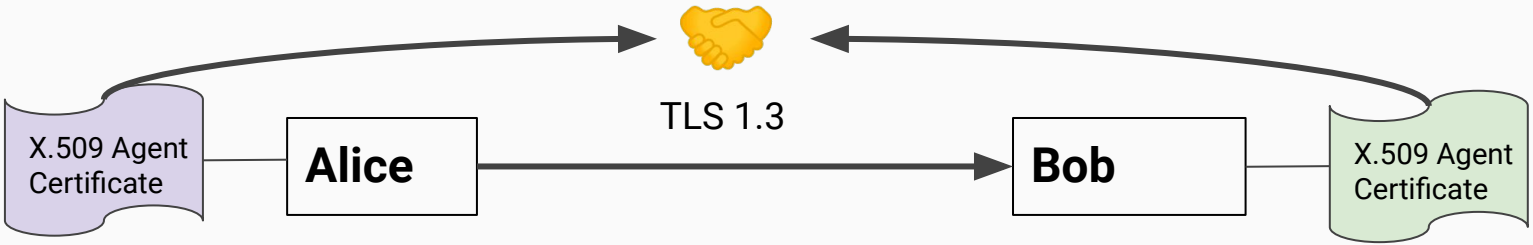
Security Updates

Background

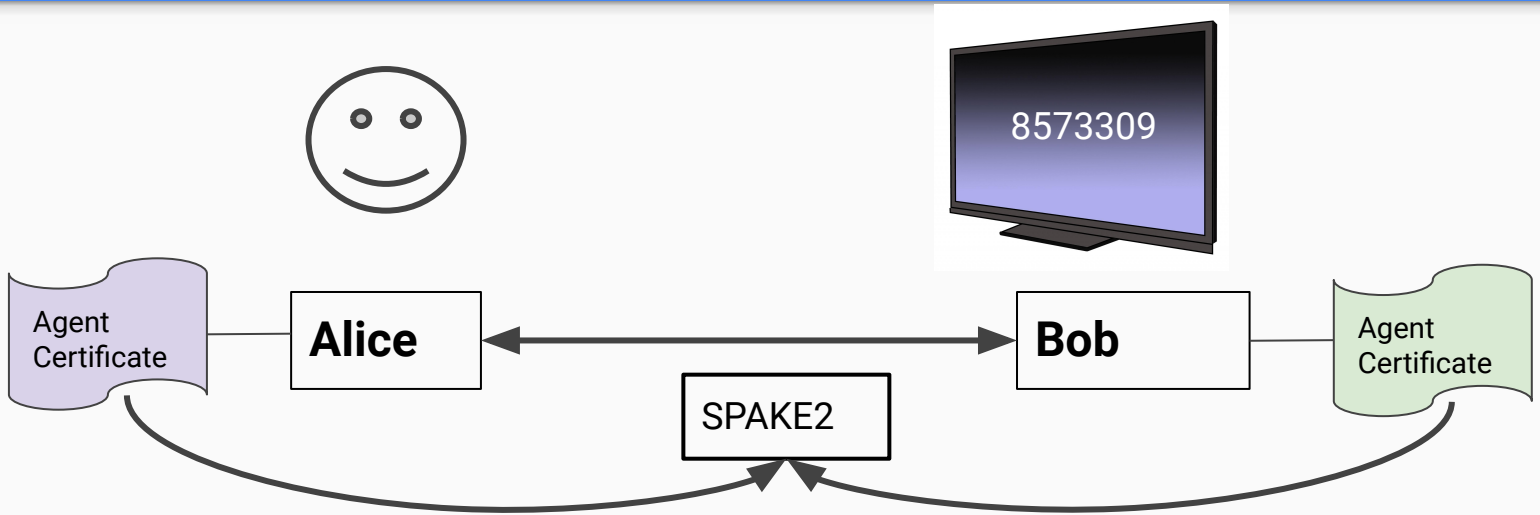
Open Screen Protocol includes an authentication sub-protocol for mutual authentication between agents.

This is important to prevent passive and active attackers from viewing/intercepting/modifying protocol messages.

Authentication



Authentication



Agent Certificates (Current)

1. 256-bit, 384-bit, or 521-bit ECDSA public key
2. Self-signed
3. Supporting certain signature algorithms
4. Valid for signing

The following X.509 v3 fields are to be set as follows:

Field	Value
Version Number	3
Serial Number	`<fp>`
Signature Algorithm ID	One of the values listed above.
Issuer Name	CN = The `model-name` from the `agent-info` message. O = See note. L = See note. ST = See note. C = See note.
Subject Name	CN = `<fp>`._openscreen._udp.local O = See note.
Subject Public Key Algorithm	Elliptic Curve Public Key
Certificate Key usage	Signing

Mandatory fields not mentioned above should be set according to [RFC5280].

The value `<fp>` above should be substituted with the agent fingerprint (as serialized in mDNS TXT).

Issue #276

“Agent Certificate has a circular dependency on itself”

The certificate serial number is its own fingerprint, making it impossible to compute the fingerprint value.

Proposal: Generate a serial number from a 32-bit random seed and a 32-bit counter.

PR #293: Add an algorithm for setting the agent certificate serial number

Issues #218/#277

#218: “Adjust cipher and signature algorithm preference list for hardware”

#277: “Consider removing support for P-521”

Mostly because of performance overheads on lower-end devices.

See this thread on mozilla.dev.security.policy

Issues #218/#277 (continued)

#218: “Adjust cipher and signature algorithm preference list for hardware”

#277: “Consider removing support for P-521”

Proposal: For ciphers, use TLS 1.3 list: AES-128, AES-256, ChaCha20. For signature schemes, require ecdsa_secp256r1_sha256 which is mandatory for TLS. Should we recommend ecdsa_secp384r1_sha384 for future compat?

PR #295: Remove P-521 curve from agent certificate requirements

PR #297 (in progress): Simplify TLS requirements…

Issue #278

“Do not use Distinguished Name <in the Subject and Issuer names> to convey protocol details”

1. RFC 6125 says how to set the commonName using a “SRV-ID”
2. Human readable text in the Distinguished Name is a problem. Instead use a random string.

Proposal: Set Issuer Name to a randomly generated string.

However, the SRV-ID includes the DNS-SD instance name, which violates #2…

Issues #279/#280

“Clarify the supported signature algorithms for certificates”

Signature algorithms and public key types in X.509 certs are represented by “Object IDs” like `1.2.840.10045.2.1` defined in RFC 4580 & RFC 5758

These IDs have a binary encoding (DER, from X.690) and we can include that format as well.

PR #288: Fixes algorithm and signing fields in agent certificate.

Agent Certificates 2.0

1. 256-bit ECDSA public key
2. Self-signed
3. Supporting ecdsaWithSha256
4. Valid for signing

Field	Old Value	New Value
Serial	Agent fingerprint	`<random>|<counter>`
Signature	???	`ecdsaWithSha256`
Issuer DN	Based on model name	`openscreen-<random>`
Subject DN	`<fp>._openscreen._udp._local`	To be determined
Key Algorithm	256, 384, or 521	`secp256r1`
Usage	`Signing`	`digitalSignature`

Issue #282

“Certificates should have a maximum lifetime” - WebTransport is 2 weeks

1. Switch agent fingerprint from cert fingerprint to SPKI from RFC 7469.
2. Rotate entire certificates without redoing SPAKE-2?

Proposals:

1. *PR #301: Define the agent fingerprint as the SPKI.*
2. *Key rotation is more complicated as agents will need to track multiple certificates per agent, verify certificate chains.*

Media Capabilities

Background

After one agent connects to another, it can request audio and video decoding and rendering capabilities.

```
; type key 122
streaming-capabilities-request = {
  request
}

; type key 123
streaming-capabilities-response = {
  response
  1: streaming-capabilities
       ;streaming-capabilities
}
```

```
streaming-capabilities = {
  0: [* receive-audio-capability] ;
receive-audio
  1: [* receive-video-capability] ;
receive-video
  2: [* receive-data-capability] ;
receive-data
}
```

receive-video-capability

```
receive-video-capability = {
  0: format ; codec
  <snip>
  ? 6: [* string] ; color-gamuts
  <snip>
}
```

```
format = {
  0: string ; name
  1: [* format-parameter] ; parameters
}
```

Issue #233: What codec name to use?

- Resolved at the Joint Media WG/SSWG meeting in Jan 2022:
  - “We can use references in the WebCodecs registry for codec string details”
- WebCodecs registry lists most common audio and video codecs
  - VP8, VP9, AV1; AVC, HEVC
  - Opus, FLAC, vorbis; mp3, mp4a
- For non-listed codecs, agents can use an RFC 6381 codec parameter
- *PR #299: Reference WebCodecs registry for codec names.*

Issue #194: HDR Capabilities

- Currently specify **`color-gamut`**
- Not sufficient to determine HDR metadata decoding capabilities
- Media Capabilities uses two additional attributes
  - `transfer-functions`
  - `hdr-metadata`
- Propose adding these fields to **`receive-video-capability`**
- *PR #300: Add transfer-functions and hdr-metadata to video-capabilities.*

SPAKE2 update

SPAKE2 RFC version 9 => version 26

- Terminology (variable names) have changed.
- The protocol described in the RFC is two round:
  - Alice and Bob exchange public values
  - Alice and Bob exchange confirmation values
- We can simplify the SPAKE2 protocol from 3 to 2 message types:
  - **`auth-spake2-handshake`** exchanges public values and coordinates PSK input
  - **`auth-spake2-confirmation`** exchanges confirmation values
- *PR #294: Update for SPAKE2*

Remaining v1-spec items

- Security
  - Issue #242: Decide if CPACE in scope for v1
  - Issue #282: Decide if key rotation is in scope for v1
  - Issue #275: Update or drop TLS SNI
  - Issue #278: Decide what subject name to use in the cert
  - Other security-tracker feedback
- Other
  - Remote playback refinements / feature requests (~ 6 issues)
  - Issue #132: Refine behavior around private browsing mode

FINIS

Issue #275

“TLS SNI requirement is incompatible with TLS SNI definition” (citing multiple RFCs)

TLS does not allow underscores in the SNI, so we can’t use `<fp>._openscreen._udp._local`

1. *Keep SNI, remove underscores*
2. *Come up with alternative SNI syntax*
3. *Remove SNI (may lose port sharing)*

Issue #279

“The keyUsage name is digitalSignature, not signing”

RFC 5280 has a specific token for this.

Proposal:

1. *Fix this.*

Issue #210

“Describe encoding/decoding of PSK into numeric and QR codes.”

We need a standardized way to turn a binary PSK into a characters or a QR code.

<TODO> Describe and/or screenshot examples

PR #296: Adds appendix with PSK specifications.

CPACE Update

- Recommended by the CFRG in March 2020.
- Computes a shared secret (intermediate key) with one round trip.
- The second version of the Internet-Draft for CPACE was published on 7/25/2021.
- There are two flavors (elliptic curves) that have different implementation properties.
- CPACE includes a shared “SID” parameter, whose properties may be specified by a Sep 2021 paper.
- There are several open source implementations, unsure how vetted they are.
- Can prepare a more detailed comparison with SPAKE2 for another meeting.