draft-ietf-lamps-csr-attestation

Mike Ounsworth, Hannes Tschofenig, Henk Birkholz,
Monty Wiseman, Ned Smith

(Michael Richardson also attending authors’ meetings)

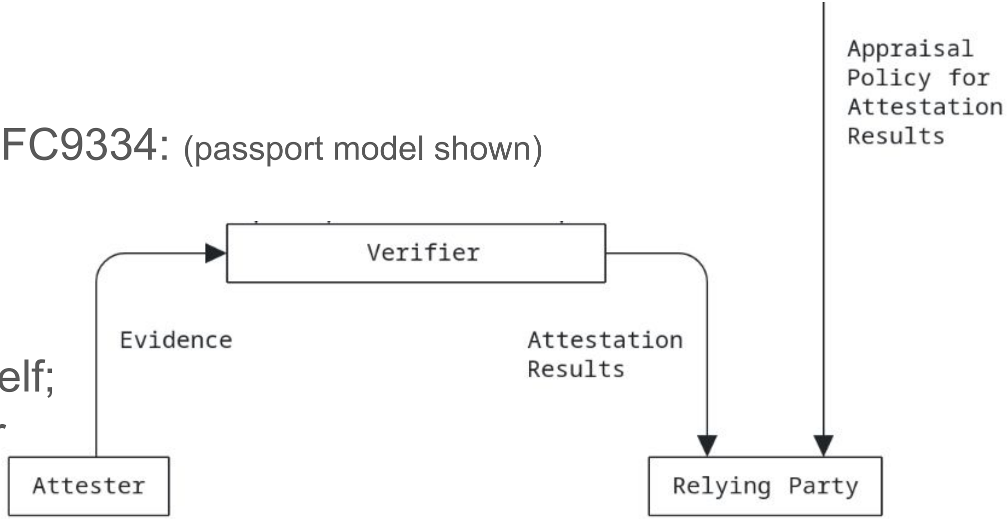
Document Status

- In WGLC.
- Mike StJohns (MSJ) provided detailed WGLC feedback – Thank You!
    - We’ll walk through these.

MSJ Feedback #1 – AttestationResults does not match Evidence

Background for the RATS-uninitiated.

RFC9334: (passport model shown)



It makes sense for CSRs to support both:

<u>Evidence</u>: Attester collects data about itself; the RA/CA will forward to a RATS Verifier (“background-check model”).

<u>Attestation Results (AR)</u>: when the Attester is expected to contact a Verifier before sending CSR to RA/CA (“passport model”).

Both Evidence and ARs are signed objects, signed by the Attester and Verifier, respectively.

MSJ Feedback #1 – AttestationResults does not match Evidence

In the current draft, our ASN.1 for carrying Evidence and ARs don't match:

```
EvidenceBundle ::= SEQUENCE {
   evidences SEQUENCE SIZE (1..MAX) OF EvidenceStatement,
   certs SEQUENCE SIZE (1..MAX) OF CertificateChoices OPTIONAL }
AttestationResult ::= SEQUENCE {
   type   ATTESTATION-RESULT.&id({AttestationResultSet}),
   stmt   ATTESTATION-RESULT.&Type({AttestationResultSet}{@type}) }
```

Specifically, AR is missing a spot to carry a cert chain.

Outcome: authors agree, thanks for pointing this out. In fact, we are missing an AttestationResultsBundle layer that has a `` `certs OPTIONAL` ``. We'll add that.

https://github.com/lamps-wg/csr-attestation/issues/198

MSJ Feedback #2 – No sample data of AttestationResults

That’s true, but a fair amount of work to write a reference implementation, especially since signed ARs are (currently) far less common in the wild, compared with Evidence.

<u>Author’s opinion:</u> The AR structure is (will be) identical up to the Evidence samples, but with a different OID and proprietary payload. If you can parse Evidence, then you can parse ARs.

<u>Question to WG:</u> Do this, or don’t do this??

https://github.com/lamps-wg/csr-attestation/issues/152

MSJ Feedback #3 – the new IANA registry should be more precise

The draft adds a new IANA registry of attestation payloads that you may encounter in CSRs.

MSJ suggests that this registry also list the ASN.1 structure that you expect to see so that you can uniquely construct an EVIDENECE-STATEMENT from this table.

8.3.2. Initial Registry Contents

The initial registry contents is shown in the table below. It lists entries for several evidence encoding OIDs including an entry for the Conceptual Message Wrapper (CMW) [I-D.ietf-rats-msg-wrap].

OID	Description	Reference(s)	Change Controller
2 23 133 5 4 1	tcg-dice-TcbInfo	[TCGRegistry]	TCG
2 23 133 5 4 3	tcg-dice-endorsement-manifest-uri	[TCGRegistry]	TCG
2 23 133 5 4 4	tcg-dice-Ueid	[TCGRegistry]	TCG

Authors have not yet discussed, but I personally believe this is a good suggestion.

https://github.com/lamps-wg/csr-attestation/issues/199

MSJ Feedback #4 – the Hint is not covering his usecase

This is the one that erupted on-list and Deb put both Mikes (O and MSJ) in timeout. Guilty 🤣

We’re talking about this:

```
EvidenceStatement ::= SEQUENCE {
   type   EVIDENCE-STATEMENT.&id({EvidenceStatementSet}),
   stmt   EVIDENCE-STATEMENT.&Type({EvidenceStatementSet}{@type}),
   hint   IA5String OPTIONAL }
```

Which is required by the draft to be a URI that SHOULD identify the software most likely to be able to parse this data; ex.: “nshield://warrantvalidatorv2.nshield.com”.

Defined vaguely on purpose.

It exists to provide a optional info to support cases where the `type` OID does not uniquely identify the data type of the payload, typically because the top-level payload is a CMW or WebAuthn wrapper object.

MSJ Feedback #4 – the Hint is not covering his usecase

MSJ: I don't need a hint; I need to cover 7 types of TPM attestations which have a single core attestation structure and different types of auxiliary data.

I don't need a Hint, I need AuxData, such as: a blob, a detached signature, and a cert chain.

He would like us to change (as far as I understand)

```
hint IA5String OPTIONAL
```

to

```
auxData &Type OPTIONAL
```

MSJ Feedback #4 – the Hint is not covering his usecase

Authors: We *could* do that; however, you can already achieve this usecase within the scope of the draft. Just define:

```
MSJ-Attest-Payload-Type ::= TYPE-IDENTIFIER {
  id OBJECT IDENTIFIER { id-msj-attest },
  type SEQUENCE{ payload MSJ-Payload, sig OCTET STRING }
}
```

`(This is very similar to the worked TPM example we have in A.2.3)`

Authors remark: Don't you anyway have to define 7 OIDS and 7 sets of parsing rules? Defining 7 data structures to go with it seems clean. So Authors believe the draft-as-currently-written is sufficient.

Authors' question: Does this satisfy, or are there other reasons for either expanding the `stmt` payload slot into multiple slots, or to have a Hint that is something other than a URI?

Summary

- Draft is in WGLC.
- We have some feedback that we need to address (but should not block WGLC).
- And we have some points that we need WG consensus on whether a change is required or not (thanks Russ for helping these along).