Digital Signatures

This page documents a series of proposals arising out of the digital signature implementation project

This topic is formally around coming to a resolution for Change Proposal tracker 3746

- 1 Specifically troubling
- 2 Signature Need
- 3 Some proposals
  - 3.1 Provenance Signature - clarification
  - 3.2 Proposal 1 : A page for digital signature implementation notes
  - 3.3 Proposal 2
  - 3.4 Proposal 3
  - 3.5 Signature Verifying Service
- 4 References

Specifically troubling

Many of the problems identified are exactly the kind of 'problem' that a Digital Signature is intended to protect against. That is, the purpose of a Digital Signature is to verify that the content has not changed. Determining which kinds of changes are allowed is 'policy', meaning a decision that element XYZ is allowed to change and not break the signature is declaring a policy that XYZ element is not important. However to enable broad interoperability some decisions need to be made.

- Advice is needed for how to create a Provenance.signature and how to validate one so that the signature survives various movements. Or do we just indicate that signatures are not intended to survive movements?
- What is the use-case for Bundle signature vs Provenance signature?
- How are signatures to be handled by a receiving system that imports the resource for use? That is it incorporates the resource content, and likely fixes up the References to local copies of the resource?
  - Should a counter-signature be used? That is the importing system will validate the incoming signature, and then counter sign the fixed-up resource, which should be a new version with a new Provenance.
- How to handle when References are included in the resource being signed?
  - Do they get turned into relative URLs?
  - If a system imports the resource and 'fixes up' the References to point at local copies of the referenced resource, then the signatures will fail to validate
  - Do they get turned into just the unique part, without the URL root?
  - Do they get turned into _id?
  - Do they get nulled?
- How are versioned Referenced resources identified?
  - Do you serialize the Referenced resource along with the resource? If so how deep?
- How to handle metadata?
  - Which metadata is included in the signature? Which is not?
  - How are signatures to be handled when the server 'improves' the resource content. Such as adding various unit conversions in an Observation.

Signature Need

The discussions to date in Security, CBCC, FM etc. can be boiled down to the need for 2 types of signatures:

integrity Signatures: Purpose to determine if content altered through a defined period (e.g. in transit). This signature is allowed to break if the resource is moved, or updated. Operation sign over all content in the resource. Should be able to apply to a bundle.

documentation Signature: Purpose to determine if content altered over time regardless of movement, should not break if moved. Operation: sign over immutable portions of current resource, and follow references in the order encountered and include either: if the referred to resource has a documentation signature include the documentation signature string if any (more efficient than harvesting the Signed-hash from the string) and do not follow references for this resource; if the referred to resource does not then include the immutable elements from the referred to resource and follow the references for this resource applying same logic as you go. This means there needs to be a specific signature transform canonicalization that isolates the portions needing signature while allowing others to change.

Document Signature: This is not 'needed' as it is already available and not a FHIR specific capability, let me explain. This is a signature across a whole Document that is managed as a whole document. This may be a FHIR document or other. Given that this document is managed as a whole document, then Document centric signatures already can be applied. See IHE-Document Digital Signature (DSG) profile for a profile of XAdES-X-L. The resulting signature is either Encapsulating the signed document, or is another document that must be managed in relationship (See DocumentReference).

Two types of digital signatures in FHIR include:

- Integrity Signatures
  - Applied to a bundle of resources
  - Includes immutable and mutable information
  - Only good for one transport session
- Document Signatures
  - Any digital signature on a resource needs to consist only of immutable information
Both the overarching and resource digital signatures need to remain intact through multiple transportation events unless content within the resource is changed. The integrity digital signature is only good for one transportation event and is used to ensure nothing was changed in transit.

Problem Statement:
The digital signature gets broken after transport when the recipient takes the resources and places them in the new system. This happens because the reference information associated with the resource is location specific. As a result, when those same resources are sent to a third party, the original signature is broken and the resources would need to be signed again; this is true whether or not the second recipient actually changed any of the other information within the resources. A broken signature results in a lack of trust and confidence in the accuracy of the information. There is no way to know whether or not the information going to the third party is the same as the information that originated with the first party. Even if the second party signs the resources, this only confirms that the information is consistent with the second party's system. There is no guarantee that the second party did or didn't make changes before sending the information on.

Proposed Solution (see diagram below):

- Do not create a new message and then re-attach the resources; if resources need to be sent to multiple parties, only forward the original message with the resources still embedded.
- Do not include reference information in the resource digital signatures: When a resource is signed, the digital signature hash is created from only the immutable information. Mutable information that is not critical to the integrity of the record, such as the resource information, is excluded. If there are multiple resources, each resource has its own unique signature minus the mutable information, such as reference information.
- In a case where there are multiple resources that go together, the base resource would have a second overarching digital signature associated with it, created by hashing the digital signatures from all of the resources included in the “bundle.” If this signature is broken, it means that either one of the resources changed, or that a resource is missing.
- Upon transport, all of the resources are bundled together and an integrity digital signature is applied to the entire bundle, including the resource information. The only purpose of this signature is to ensure integrity during transport. It is understood that this signature will be broken when the resources are incorporated into the recipient's system. A new integrity signature will be applied by the new sender anytime the resources are sent to another recipient.

Some proposals
Provenance Signature - clarification

The current text was carefully crafted, but clearly we need to clean up the signature definition in Provenance. This signature was only intended to bind the specific versioned content to the provenance that describes the 'provenance' of that versioned content. Therefore it should be a more simple flavor of a digital signature. However it does still suffer from many of the issues.

Proposal 1 : A page for digital signature implementation notes

We should add a digital signature page that:

- describes the use cases that we support
- details general implementation considerations & where to look for support
- describes a profile on xml digital signatures
- describes some FHIR specific transforms that when used with FHIR resources confer robustness against format changes, and meet the use cases
- deals with xml vs json

Proposal 2

We should add a note to both the xml and json pages describing signature ready xml and json. Implementers are recommended to conform to this

XML ready for Digital Signature

- default namespaces - only use default namespaces for fhir and xhtml namespaces
- outside the xhtml, no whitespace except in attributes
- no processing instructions, no xml comment

JSON ready for digital Signature

- all unimportant whitespace removed from the JSON representation

XML/JSON that conforms to this profile can be reproduced exactly from the data in the resource, without needing to store the actual XML (or can be recreated correctly from the JSON)

Proposal 3

That we define a standard transform

Signature Verifying Service

To address the problems that arise when the actor using a FHIR resource may be low-tech, or be bound to only one transform (XML, JSON, etc).

A signature creation must happen at a place where the private key of the signer is present. Thus the signature operation can't be easily encapsulated into a service.

- Note there are Timestamp Digital Signature services, that will apply a signature. This signature is by the service, and with the purpose only of binding an authoritative timestamp with the content. Although this is a type of signature it is not seen as the primary use-case.

A signature verifying service could be created. It would contain all the complexity necessary to verify any signature types defined. The additional benefit is that the client that needs proof of signature can call this service and not be bound to any specific policy, technology, encoding, etc.

The drawback is that the signature verifying service must be trusted completely, but this true of any software doing signature verifying.

References

- Email discussion in December 2014