Trust Model for Security Automation Data (TMSAD)

Harold Booth
NIST

Agenda

- Why a Trust Model
  - Goals
  - Use Cases
- Trust Model
  - W3C XML Signature
  - Algorithms and Parameters
- Example Signature
  - SCAP Source Stream (Input)
  - ARF (Results)
- Associating Identity

Goals

- Express signatures in a common format
- Begin signing content
- End-users able to validate signatures
  - Who is this from?
  - Has it changed?
  - Is this content authorized for my system?

Content Use Case (input)

- A content consumer needs to verify authenticity of a content stream
  - Content published by an author or authority
  - Validate that content has not been altered since publication by the author or authority
  - Consumers can establish trust with respect to content based upon identity of author or authority

Content Use Case (prior knowledge)

- Re-establish trust to content based upon prior knowledge
  - Assist with solving referential trust
  - Could be used in lieu of using identity of the author or authority

Content Quality Assurance Use Case

- An individual or organization signs content to assert confidence or trust in content
  - QA function – works in a defined environment
  - Organizational policy asserts only trusted content may be run
  - Need to maintain provenance information – who originally published
    - Traceability

Compositional Content Use Case

- A content consumer would like to know and verify that a content stream is composed of multiple source streams
  - An author may compose a data stream from multiple data streams and augment with own contribution
  - Allow reporting of results derived from a source stream to be performed independently of other source streams
  - Focus QA efforts only on augmented portion
  - Identify differences between source stream and composed stream

Results Use Case

- An organization needs results signed at the point of creation in order to verify authenticity of results
  - Results generated by a tool

Results Use Case (expanded)

- An organization needs results signed with source content identity and/or target identity at the point of creation in order to verify authenticity of produced results
  - Results created based on responses of a machine endpoint (e.g. OVAL) or individual (e.g. OCIL) – a target
  - Expanded to include identity of source content and/or target
  - Establishes identity of tool, target, and source content
  - Assumes targets have an identity capability

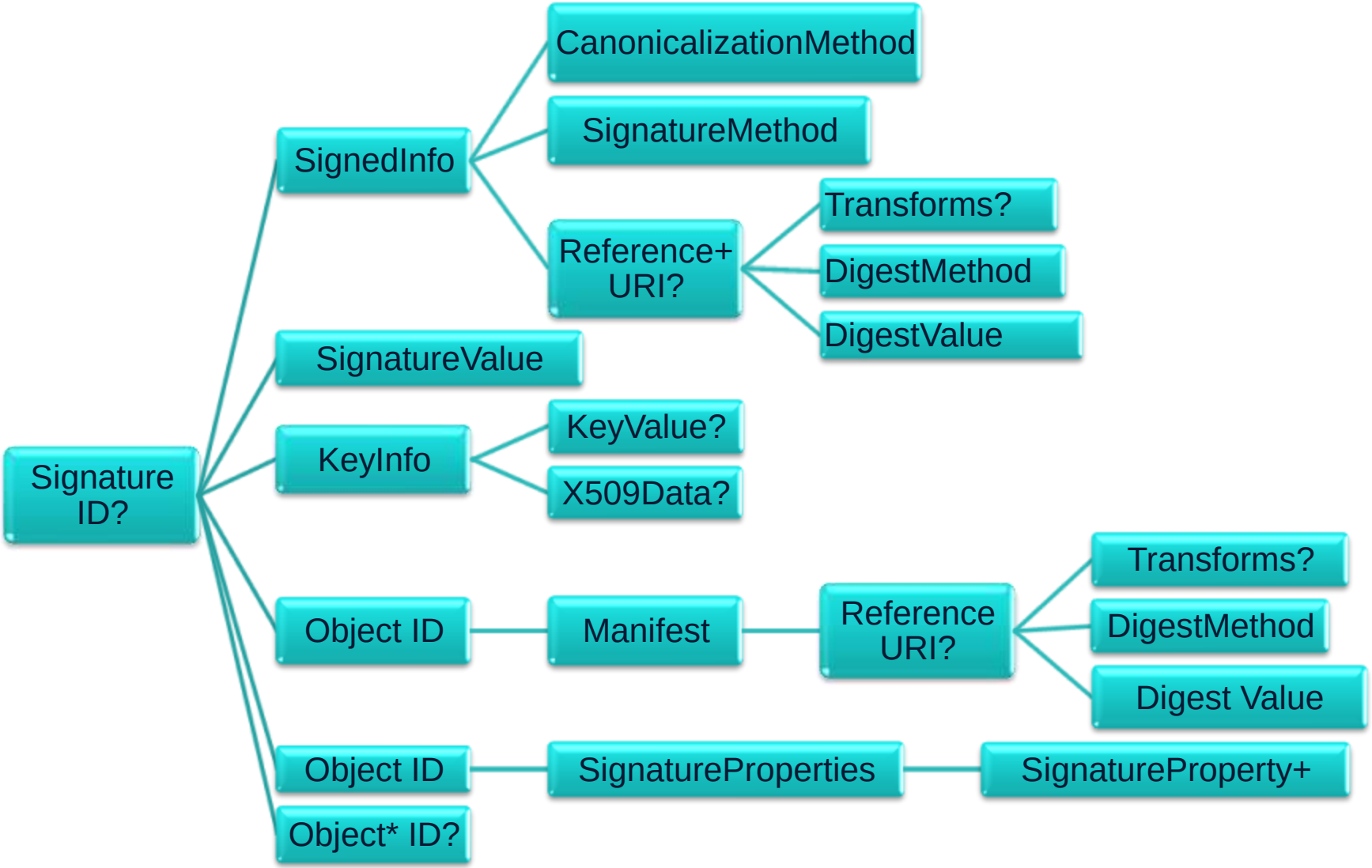
Aggregated Results Use Case

- Aggregation tools need to combine results and sign aggregated results
  - Maintain source data to allow consumers of aggregated data to validate findings at a later point
  - Provides traceability of aggregated results

XML Signature Syntax and Processing

- W3C Recommendation
- IETF RFC 3275 (initial release)
- Specialized to handle XML data
  - Canonicalization
  - Transform
- Defers to applications for verification logic
  - Public key is optional
- Hooks for X.509 Certificates and PGP Keys

XML Signature Overview



XML Signature W3C Example

```
[s01] <Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
[s02] <SignedInfo>
[s03]  <CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
[s04]  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
[s05]  <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
[s06]  <Transforms>
[s07]   <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
[s08]  </Transforms>
[s09]  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[s10]  <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK.../DigestValue>
[s11]  </Reference>
[s12] </SignedInfo>
[s13] <SignatureValue>...</SignatureValue>
[s14] <KeyInfo>
[s15a] <KeyValue>
[s15b]  <DSAKeyValue>
[s15c]   <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
[s15d]  </DSAKeyValue>
[s15e] </KeyValue>
[s16] </KeyInfo>
[s17] </Signature>
```

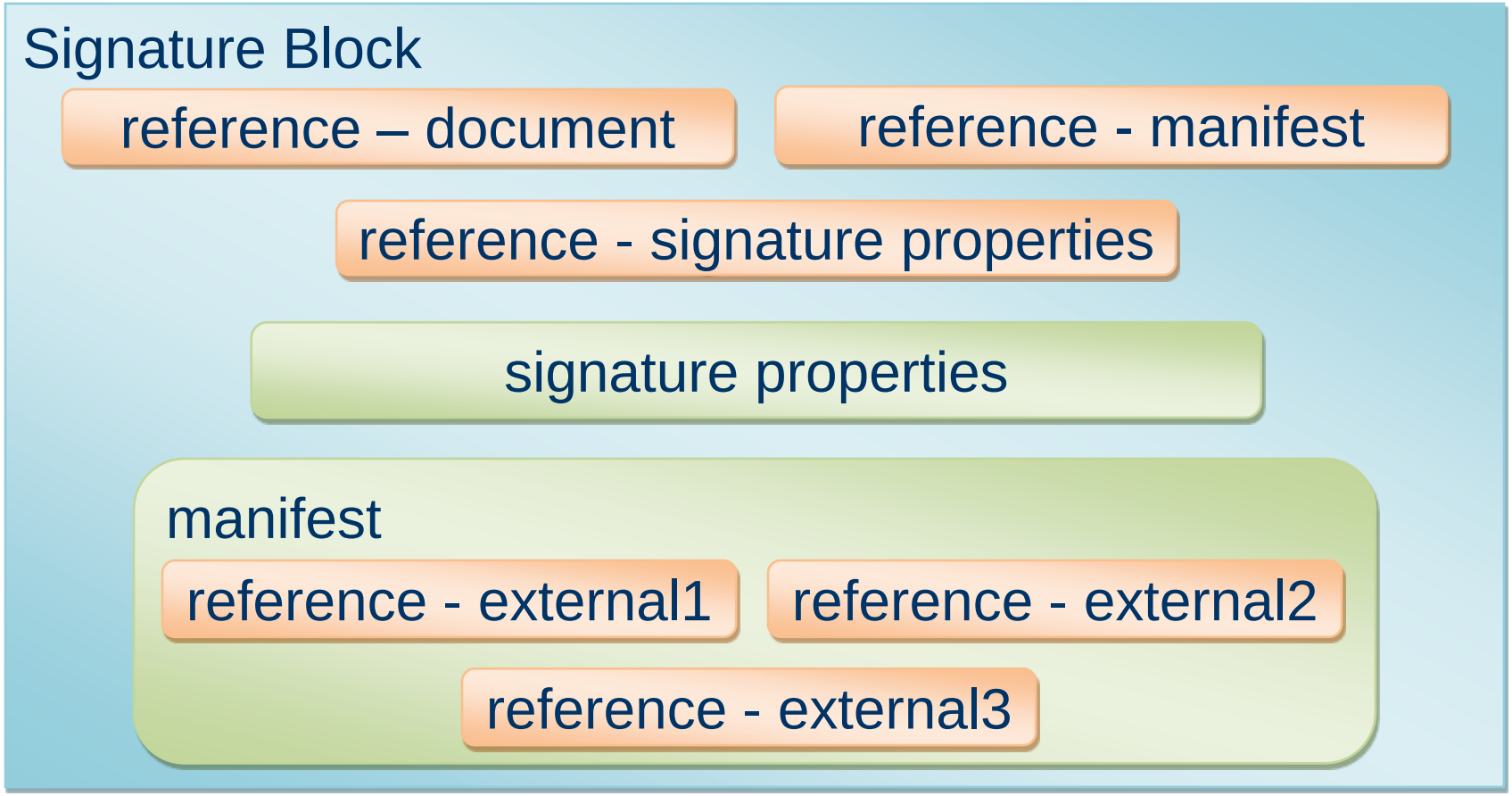
XML Signature Manifest Example

```
[ ]  ...
[m01] <Reference URI="#MyFirstManifest"
[m02] Type="http://www.w3.org/2000/09/xmldsig#Manifest">
[m03]   <Transforms>
[m04]     <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
[m05]   </Transforms>
[m06]   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[m07]   <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK...=</DigestValue>
[m08] </Reference>
[ ]   ...
[m09] <Object>
[m10]   <Manifest Id="MyFirstManifest">
[m11]     <Reference>
[m12]       ...
[m13]     </Reference>
[m14]     <Reference>
[m15]       ...
[m16]     </Reference>
[m17]   </Manifest>
[m18] </Object>
```

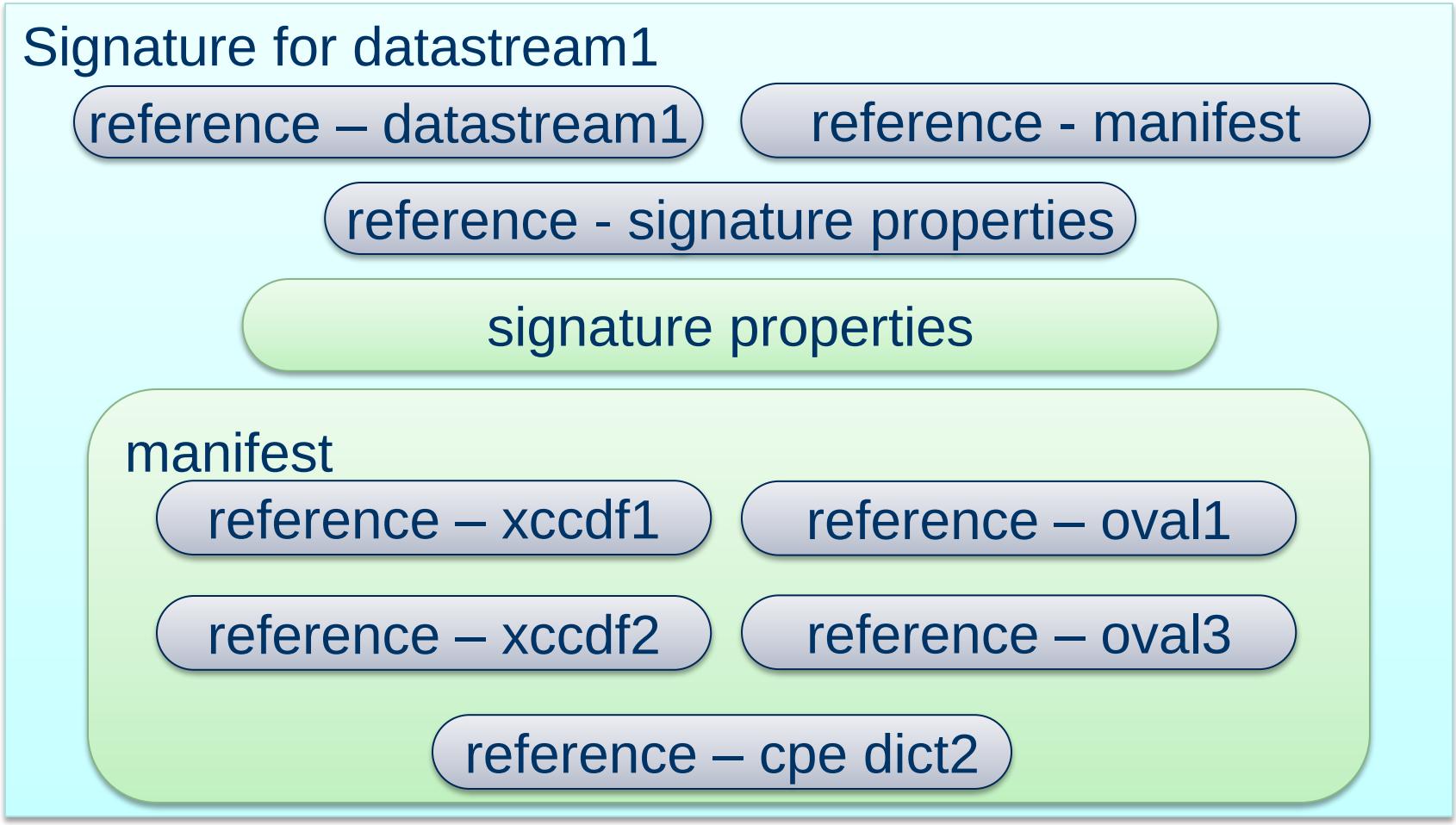
Algorithms and Parameters

- Based on recommendations in FIPS 186-3 and SP 800-57
- RSA
  - 2048-bit key
  - SHA-256 (SHA-384, SHA-512 optional)
  - PKCS #1.5 padding
- Elliptical Curve Digital Signature Algorithm
  - 256-bit Prime Curve
  - SHA-256

Signature Block



Signing SCAP 1.2 Datastream



Associating Content With an Identity

- X.509Data
  - X.509 Certificate Data Element within KeyInfo
  - The key can either be embedded with the signature or retrieved separately
- PGPData
  - A PGP Data Element within KeyInfo
  - Key material is included

References

- XML Signature Syntax and Processing
  - http://www.w3.org/TR/xmldsig-core/
- XML Signature Syntax and Processing Version 1.1, W3C Candidate Recommendation
  - http://www.w3.org/TR/2011/CR-xmldsig-core1-20110303/
- XML Signature Best Practices
  - http://www.w3.org/TR/xmldsig-bestpractices/
- Additional XML Security URIs
  - http://www.ietf.org/rfc/rfc4051.txt

Questions & Answers / Feedback

Harold Booth

Computer Scientist

Computer Security Division

Information Technology Laboratory

National Institute of Standards and Technology

scap-dev@nist.gov