Applying COSE Signatures for YANG Data Provenance

draft-lopez-opsawg-yang-provenance

D. López, A. Pastor, A. Méndez (*Telefónica)*
A. Huang Feng (*INSA-Lyon*)
H. Birkholz (*Fraunhofer SIT*)
S. García (*UC3M*)

IETF#122, Bangkok, March 2025

News on Prov·e·nance|no͞oz än ˈprävən(ə)ns|

- Annex complete with examples for JSON and XML
  - Using the reference implementation demonstrated at the Hackathon
- Added ”Implementation Status” section in -06
  - Noting that same reference implementation
- A few issues to follow up
  - On moving it from informational to the standards track
  - On canonicalization and deterministic YANG representation
- Working on other integration patterns
  - Accountability and auditability
  - Concurrency
  - Integration with pub/sub and time series DBs
- And additional enclosing methods
  - With special attention to YANG Push Lite
  - Facilitated by the core COSE approach

News on Prov·e·nance Code
| no͞oz än ˈprävən(ə)ns kōd|

- First reference implementation available
  - Java implementation
  - Addressing XML and JSON
  - And the four proposed enclosing methods
  - Integrable with different YANG modules and suites
- Demonstrated at the Hackathon
  - Four microservices: signer and verifier for, XML and JSON
  - REST interface, using serverless mechanisms
  - https://kafka.hexa.tid.es/function/<service>?enclosing=<method>
  - POST, with the YANG fragment to sign or verify
  - Service: `xml-signer`, `xml-verifier`, `json-signer`, `json-verifier`
  - Enclosing: `leaf`, `notification`, `instance`, `annotation`
- Code available in GitHub
- Containers available in DockerHub

The Next Intended Steps

- Enhance the design of the reference implementation, along
  - Core
  - Enclosing methods
  - Interfacing
- Complete the approach
  - Canonicalization
  - Integration patterns
  - Enclosing methods
  - CBOR
  - Aligning the draft and the reference implementation
  - Keeping a reference set of microservices
- And adoption
  - Once we have an implementation available
  - Within the standards track…