XML Signature Performance and One-Pass Processing Issues

Position Paper Presentation
Sean Mullan
Sun Microsystems
W3C Workshop on Next Steps for XML Signature and Encryption
Agenda

- Performance Issues
- One-Pass Processing Issues
- Solutions
- STaX Implementation
Performance Issues

- DOM provided good implementation solution for XML DSig
- But DOM can cause performance issues
  - Memory footprint increases as size increases
  - Building and navigating tree takes time
  - Not the best solution for certain applications where performance/scalability is very important, such as WSS
- Ok, then how about processing the signature in one-pass?
One-pass processing

• What do we mean by one-pass processing?
  – XML Signature can be generated or validated in a single pass (as a stream of data)
  – Minimal caching
  – Does not require document to be built as a tree in memory

• PKCS7 and PGP support one-pass processing
One-pass Implementation Issues

• Validation of backward references
  – Data objects located before Signature element
    
    ```xml
    <?xml version="1.0" encoding="UTF-8"?>
    <Data id="data"/>
    <Signature>
      ..<Reference URI="#data">
    </Signature>
    ```

• Potential solutions
  – Two-passes (or 1+)
  – Cache all elements with ID attributes
  – Use profile-specific knowledge
One-pass Implementation Issues

• KeyInfo located after SignedInfo
  – Cannot verify signature until you parse KeyInfo element and establish key
  – Cannot stream signature verification
  
    <Signature>
    <SignedInfo/>
    ..<KeyInfo>

• Potential Solutions
  – Cache SignedInfo element
  – Cache SignedInfo canonicalized bytes
One-pass Implementation Issues

- Cannot canonicalize/verify SignedInfo until CanonicalizationMethod and SignatureMethod are parsed

  `<SignedInfo>`
  `<CanonicalizationMethod Algorithm="..."/>`
  `<SignatureMethod Algorithm=".."/>`

- Minor issue, but must cache some data
One-pass Implementation Issues

• Canonicalization algorithms that depend on ancestor context (ex: inclusive C14N)
  – Namespaces, inheritable xml attributes
  – Already parsed, can't go back

• Potential solutions
  – Cache namespaces and xml attributes as parsed
  – Use parser that maintains namespace and xml attribute context
One-pass Implementation Issues

• Transform nodeset input/output model doesn't support streaming
One-pass Implementation Issues

• Signature generation issues
  – Data objects need to be hashed before SignedInfo is written
  – Forward references (data objects after Signature element) are problematic
  – Opposite problem of validation

• Potential Solutions
  – 2 passes
Ideal Solutions

• Signature header that identifies references, algs
• Signature(s) at end of document
• ... but this is at odds with verifying the signature first, then the references

```xml
<SignatureHeader>
  <Reference URI="#data">
    <DigestMethod Algorithm="..."/>
    <Transforms/>
    <!-- No DigestValue -->
  </Reference>
</SignatureHeader>

...

<Signature>
  <KeyInfo/>
  <SignedInfo/>
  <SignatureValue/>
</Signature>
```
XML DSig Streaming Impl.

- Apache project
  - Authors: Raul Benito Garcia (primary), Sean Mullan
- Based on STaX, JSR 105 API
- Supports exclusive C14N, forward references, enveloping signatures, Base64 Transform
- Does not support inclusive C14N, backward references, enveloped/XPath transform, signature generation