

Efficient Implementation of XML Security for Mobile Devices

International Conference on Web Services 2007

Jaakko Kangasharju

Helsinki Institute for Information Technology

July 11, 2007

Outline

- ① Introduction
- ② Compression with XML Encryption
- ③ Implementation Technique
- ④ Experimentation
- ⑤ Conclusions

Security in Mobile Web Services

- Some applications need XML security: fine-grained end-to-end encryption and signatures
- Overhead of XML security can be considerable compared to other security methods
- Mobile devices need energy-efficient implementations
- Efficiency in mobile computing mostly depends on amount of data transmitted over wireless network

Extending XML Encryption

- When compressing data, compression must be applied prior to encryption
- XML Encryption provides no way to indicate encrypted XML data is compressed
- Extend `EncryptedData` with attributes to provide MIME type and encoding of data
- MIME type allows alternate formats, encoding allows use of compression

Implementation Technique

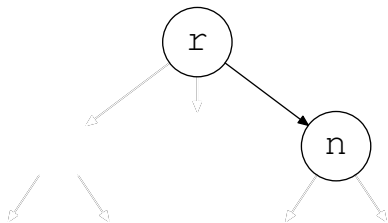
- Implementation built using XAS, a general-purpose XML API explicitly designed for innovative XML applications
- Only “special” feature used: Access to byte I/O streams during parsing and serialization
- XAS internal representation constructed to support efficient (inclusive) canonicalization
- Serialization based on in-memory node representation of XML
- Implementation uses capability for **application-specific** nodes in the API
- Parsing in a streaming manner to the extent possible

Signing Example

- Signing based on replacing signed nodes with their serialized form, **out-of-order** serialization
- XML document, wish to sign element n:

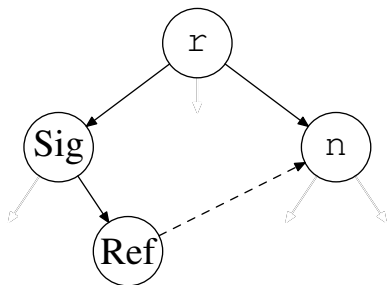
```
<r><n></n></r>
```

Signature Generation



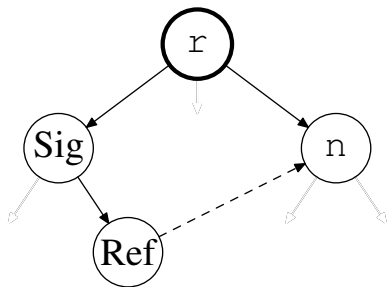
Initial representation.

Signature Generation



Add a special Signature node as a child of r , with a Reference pointing to n .

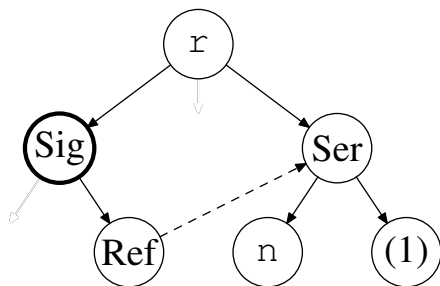
Signature Generation



`<r>`

Begin by serializing the start tag of the root node.

Signature Generation

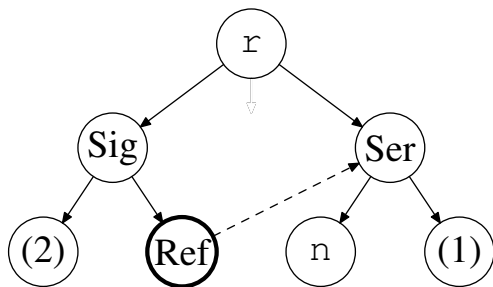


`<r><S><SI>`

Begin signature node processing by replacing all signed elements with serialized nodes.

Additional contents:
(1): `<n></n>`

Signature Generation

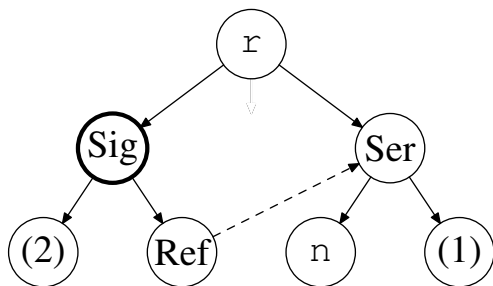


```
<r><S><SI><R><DM></DM>  
<DV>...</DV></R></SI>
```

Serialize SignedInfo using computed digests and also compute the digest of SignedInfo.

Additional contents:
(1): `<n></n>`
(2): Digest of SI

Signature Generation

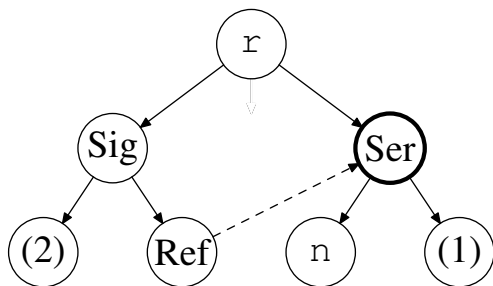


```
<r><S><SI><R><DM></DM>  
<DV>...</DV></R></SI>  
<SV>...</SV></S>
```

Compute the signature value based on the computed digest for SignedInfo.

Additional contents:
(1): `<n></n>`
(2): Digest of SI

Signature Generation

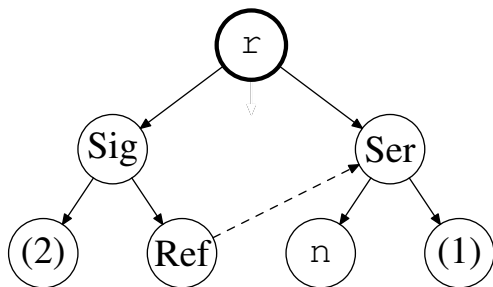


```
<r><S><SI><R><DM></DM>  
<DV>...</DV></R></SI>  
<SV>...</SV></S>  
<n></n>
```

Write the serialized bytes of element *n* directly into the output stream.

Additional contents:
(1): `<n></n>`
(2): Digest of SI

Signature Generation



All children of `r`
processed, output end
tag.

```
<r><S><SI><R><DM></DM>  
<DV>...</DV></R></SI>  
<SV>...</SV></S>  
<n></n></r>
```

Additional contents:

- (1): `<n></n>`
- (2): Digest of SI

Experimentation Setup

- Measurements ran on Nokia E61 using HTTP over UMTS
- Two formats: regular XML and binary format Xebu
- Three levels of compression: none, gzip at HTTP level (**z**), and gzip before encryption and at HTTP level (**zz**)
- SOAP messages, header single WS-Security header, body sequence of card elements representing credit cards (message size reported as number of cards), body both signed and encrypted
- Measured times for serialization, parsing, and communication
- Serialization and parsing times split into components

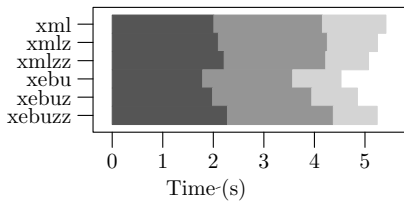
Total Sizes

Sizes in bytes

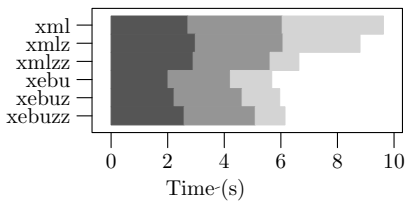
	none	HTTP	full		none	HTTP	full
XML	5141	3252	2168	XML	19925	14560	3484
Xebu	2949	2396	2232	Xebu	6229	5721	3734
	5 elements				50 elements		

Total Times

Times



5 elements



50 elements

- Message serialization
- Message parsing
- Communication

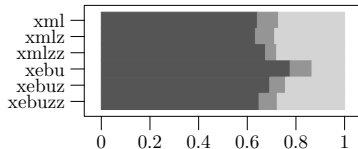
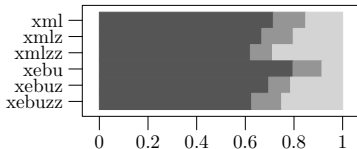
Serialization and Parsing Breakdown

Elem

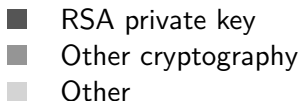
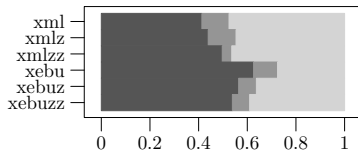
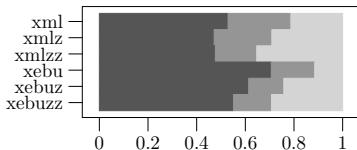
Serialize

Parse

5



50



- Reducing message size critical for secure mobile Web services

Conclusions

- Reducing message size critical for secure mobile Web services
- Generic compression not precluded due to inefficiency

- Reducing message size critical for secure mobile Web services
- Generic compression not precluded due to inefficiency
- XML Encryption must be extended to support compressed XML content

- Reducing message size critical for secure mobile Web services
- Generic compression not precluded due to inefficiency
- XML Encryption must be extended to support compressed XML content
- Efficiency of security operations, especially RSA, needs attention

Thank You

Questions?