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
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, \textit{out-of-order} serialization
- XML document, wish to sign element \texttt{n}:

\[
<r><n></n></r>
\]
Signature Generation

Initial representation.
Add a special Signature node as a child of \( r \), with a Reference pointing to \( n \).
Begin by serializing the start tag of the root node.

\(<r>\)
Signature Generation

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

Additional contents:
(1): <n></n>
Signature Generation

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

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

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

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

\[<r><S><SI><R><DM></DM><DV>...</DV><R><SI><SV>...</SV></S>\]

Additional contents:
(1): \(<n></n>\)
(2): Digest of SI
Signature Generation

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.

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

$$\text{Additional contents:}
(1): <n></n>
(2): Digest of SI$$
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

<table>
<thead>
<tr>
<th></th>
<th>none</th>
<th>HTTP</th>
<th>full</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>5141</td>
<td>3252</td>
<td>2168</td>
</tr>
<tr>
<td>Xebu</td>
<td>2949</td>
<td>2396</td>
<td>2232</td>
</tr>
</tbody>
</table>

5 elements

<table>
<thead>
<tr>
<th></th>
<th>none</th>
<th>HTTP</th>
<th>full</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>19925</td>
<td>14560</td>
<td>3484</td>
</tr>
<tr>
<td>Xebu</td>
<td>6229</td>
<td>5721</td>
<td>3734</td>
</tr>
</tbody>
</table>

50 elements
Total Times

Times

5 elements

- Message serialization
- Message parsing
- Communication

50 elements
Serialization and Parsing Breakdown

<table>
<thead>
<tr>
<th>Elem</th>
<th>Serialize</th>
<th>Parse</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><img src="image1" alt="Serialization Diagram" /></td>
<td><img src="image2" alt="Parsing Diagram" /></td>
</tr>
<tr>
<td>50</td>
<td><img src="image3" alt="Serialization Diagram" /></td>
<td><img src="image4" alt="Parsing Diagram" /></td>
</tr>
</tbody>
</table>

- **RSA private key**
- **Other cryptography**
- **Other**

Jaakko Kangasharju (HIIT)  XML Security Implementation  July 11, 2007
Conclusions

• 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
Conclusions

- 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
Conclusions

- 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?