Bewertung der Einsatzmöglichkeiten von XML Sicherheitslösungen in mobilen Kommunikationsumgebungen

Fabian Pretsch

Ziel

- Implementierung von XML Encryption/Signature in Java
- Testen der Implementierung auf einem PDA
- Bewertung der Ergebnisse
- Evaluierung von Alternativen

XML Signature/Encryption

- „Recommendations“ der W3C
- Standards herunterladbar unter www.w3c.org
- Ziel: Definition eines Standards zum sicheren Austausch von XML Dokumenten (insbesondere im wirtschaftlichen Umfeld)

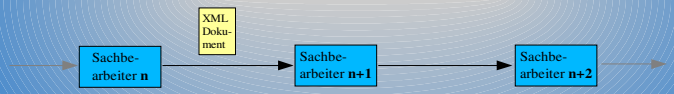
SSL/TLS

- Daten liegen unverschlüsselt auf den Servern, Verschlüsselung findet auf dem Übertragungsmedium statt
- + Einfache Anwendbarkeit
- - Angriffe möglich durch Sicherheitslücken in SSL/TLS
- - Server selbst können angegriffen werden: Hier lagern die Daten unverschlüsselt!

XML Encryption/Signature

- Nicht die Verbindung wird verschlüsselt, sondern die Daten, dadurch Reduzierung der Angriffsfläche
- Es können verschiedene Teilbäume von XML Dokumenten mit unterschiedlichen Schlüsseln und Methoden verschlüsselt werden
- Damit Unterstützung von Workflow Szenarien
- Jeder Sachbearbeiter kann nur „seinen“ Teilbereich des XML-Dokumentes lesen und verändern
- Es können beliebige Daten verschlüsselt/signiert werden, nicht nur XML-Dokumente

Workflow



Beispiel XML-Signature


```
<?xml version="1.0" encoding="ISO-8859-15"?>
<Signature>
<SignedInfo>
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
<Reference URI="file:///home/fab/Diplomarbeit/src/books.xml">
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>UuPdzaxg6k94nvlDtWaiGSUHqJQ=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>Pj7s2UzMB9K+0YyE7oVD5irXFbm0fXnzHjFBaG3i3C0CI6o2gOJEBg==</SignatureValue>
<KeyInfo><KeyValue><DSAKeyValue><P>MTA2NzcyMjM4MTkxNzIzNTg4MzMzMDU0MzkwNzkxMDU3MDc4ND
MxNTkwMDMyMjgzMzYyNTQ4NDg3NjQxMjYyNDk5MTI0MDQxMjY0MTA2NjE4ODczNDIwNDA2MjE5NDEyO
DE0MTQ2OTg3NTQ4MjA3NTQzNDg2OTcwMDQwNzg4NjE4NTUxNTUzNjM0MDQ2NzkyMjA0MDI3MDc5MTc=<
/P>
<Q>MTQwNjQzNzQyMzgzMDE1NDk0NDE0NjU3MjAxNjgwNDUyMzE3NzM0NDY0NTY0MjQxMw==</Q>
<G>Nzc2NDUzNzUwMTMxMTAxMzQwNDU3MjQxODIxMTY3MjUyMzczNjk2Mzc4NDQ3ODY1MzI2NTkwNzc1N
TMzNjE5ODM1Mjk1ODExNzI1ODk4NDc2NTU3MjkxNDEzNDU0NTMzNTIwMjUzNDAzODM4ODE2NjYzODM0O
TE2NDU2MTc0MzM1ODAwMDA5NTE2OTUxMzU3NTE2NDgyNjc0MA==</G><Y>ODQyMDMyNTYwMzU3MTM
2NzYwOTU2NzQxMjM2MzI4MDkyMzU0NTkyMTYxNzU5NDU4MjU0NTM3OTA4MjEyNDgwNTY4MzE5NzY1OT
YyOTYzNzg0MDYwNzUyMjYyNTkwNjEyOTgwOTM2Mzg1NzAyMjA0MDQ1NDgxMzU5NDM5ODM5NTI3MjcwNj
gwOTExOTE0Mzc1ODA2MTE2NDc2Mg==</Y></DSAKeyValue></KeyValue></KeyInfo>
</Signature>
```


Formen von XML-Signaturen

- Detached: Die Signatur ist von dem zu signierenden Objekt getrennt. Das Objekt wird durch eine URI identifiziert
- Enveloping: Das Objekt ist ein Teilbaum der Signatur
- Enveloped: Die Signatur ist ein Teilbaum des zu signierenden Objektes, die Signatur wird über den XML-Restbaum des Dokumentes berechnet

XML Canonicalization

- Die Signatur sollte vom Inhalt des zu signierenden Dokumentes abhängen, nicht von der Darstellung!
- Einheitliche Kodierung (UTF-8)
- Behandlung von Zeilendelimitern (DOS,Unix,Macintosh)
- Behandlung von Kommentaren
- Behandlung von Sonderzeichen
- Eliminierung von nicht relevanten Whitespaces

XML Canonicalization

- Vor der Signierung wird eine kanonische Form des XML-Dokumentes erzeugt
- Die Kanonische Form des Dokumentes wird signiert (die Originaldatei bleibt unverändert)
- Der Empfänger vergleicht die empfangene Signatur mit der Signatur der kanonischen Form des Originaldokumentes

Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<booklist  >
<!--Sinnloser Kommentar-->
<book     >

      <title>Einstieg in XML</title>
      <author>Helmut Vonhoegen</author>
      <publisher>Galileo Computing</publisher>

</book    >

</booklist>
```

Kanonische Version

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<booklist>
<book>

      <title>Einstieg in XML</title>
      <author>Helmut Vonhoegen</author>
      <publisher>Galileo Computing</publisher>

</book>

</booklist>
```

Beschränkung durch PDA-Umgebung

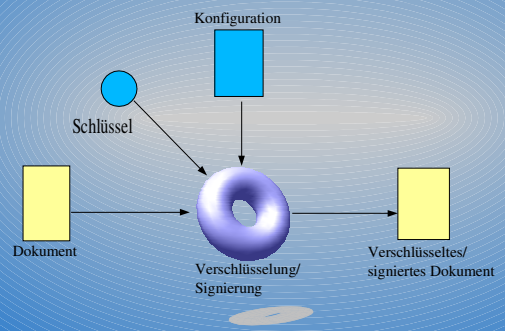
- Begrenzte CPU-Leistung
- Begrenzter RAM, insbesondere kein virtueller Speicher!
- Dadurch DOM-Darstellung des XML Baumes ungeeigent

Verwendete Bibliotheken

- Kxml2: Eventauslösender Pullparser (im Gegensatz zum Push-Ansatz von SAX)
- BouncyCastle: Crypto-API mit zahlreichen implementierten Verschlüsselungs-/Signieralgorithmen
- Beide Bibliotheken sind für den Einsatz mit der Java2 Micro Edition (J2ME) ausgelegt und daher sehr ressourcenschonend

Aufbau des Programms

- Initialisierung mittels einer XML-Konfigurationsdatei
- Im wesentlichen nur zwei öffentliche Methoden:
- Sign() bzw. encrypt()
- check(Reader input) bzw. decrypt(...)
- Was wie womit signiert/verschlüsselt wird ist in der Konfigurationsdatei festgelegt

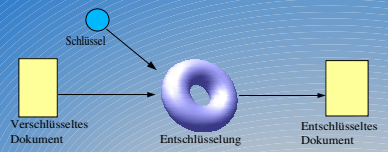


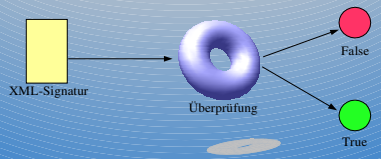
Beispiel Konfigurationsdatei

```
<SigConfig>
<Reference Method="detached">
file:///home/fab/Diplomarbeit/src/books.xml
</Reference>

<CanonicalizationMethod>
http://www.w3.org/TR/2001/REC-xml-c14n-20010315
</CanonicalizationMethod>

<SigMethod>
http://www.w3.org/2000/09/xmldsig#dsa-sha1
</SigMethod>

<Key>
file:///home/fab/Diplomarbeit/src/dsa_opts
</Key>
</SigConfig>
```

Beispiel Konfigurationsdatei 2

```
<SigConfig>
<Reference Method="embedded" Id="buecher">
file:///home/fab/Diplomarbeit/src/books.xml#title
</Reference>

<SigMethod>
http://www.w3.org/2000/09/xmldsig#dsa-sha1
</SigMethod>

<Key>
file:///home/fab/Diplomarbeit/src/dsa_opts
</Key>
</SigConfig>
```





Ende der Präsentation

- Fragen?