

XML Encryption and Signature

Hauptseminar Datenbanken und XML

Markus Hinkermann

08. Januar 2002

Motivation

1. Verschlüsselung *Encryption*

Unkenntlichmachung sensibler oder schützenswerter Daten

2. Signatur *Signature*

Authentifizierung von Daten

*

zurück

vor

Verschlüsselung in XML

Eine Rolle spielen bei einer Verschlüsselung

- der Sender oder Verfasser
- der Empfänger
- die Daten
- die verschlüsselten Daten
- die Schlüssel
- die Medien, auf denen die Daten liegen

*

[zurück](#)

[vor](#)

Verschlüsselungsalgorithmen

1. Verschlüsselung mit symmetrischem Schlüssel

Beispiel: *TripleDES*

2. Verschlüsselung mit asymmetrischem Schlüsselpaar

Beispiel: *RSA*

*

zurück

vor

Beispiel sensibler Daten

```
<Personaldaten>
  <Person ID="MaxMoritzen012000" >
    <Name> Max Moritzen </Name>
    <Gehalt> 2500 </Gehalt>
    <Beurteilung>
      <Beurteiler URI="#HansHansen091991" />
      <Text> Max macht sich hervorragend </Text>
      <Note> 1 </Note>
    </Beurteilung>
  </Person>

  <Person>
    :
  </Person>
  :
</Personaldaten>
```

*

zurück

vor

Granularität der Verschlüsselung

Verschlüsselt wird

1. ein XML-Element
2. Inhalt eines XML-Elements
3. Wert eines XML-Elements
4. beliebige Daten
5. Super-Encryption, d.h. Mehrfachverschlüsselung

*

zurück

vor

Verschlüsselung eines XML-Elements

```
<Personaldaten>  
  <Person ID="MaxMoritzen012000" >  
    <Name> Max Moritzen </Name>  
    <Gehalt> 2500 </Gehalt>  
    <Beurteilung>  
      <Beurteiler URI="#HansHansen091991" />  
      <Text> Max macht sich hervorragend </Text>  
      <Note> 1 </Note>  
    </Beurteilung>  
  </Person>  
</Personaldaten>
```

*

zurück

vor

Verschlüsselung eines XML-Elements

```
<Personaldaten>  
  <EncryptedData ID="Person1" Type="Element" >  
    <EncryptionMethod Algorithm="3des-cbc" />  
    <KeyInfo>  
      <KeyName> Total Security Key </KeyName>  
    </KeyInfo>  
    <CipherData>  
      <CipherValue>jzHj/jl8Zuhhje8j389</CipherValue>  
    </CipherData>  
  </EncryptedData>  
</Personaldaten>
```

*

zurück

vor

Verschlüsselung mehrerer XML-Elemente

```
<Personaldaten>
  <Person ID="MaxMoritzen012000" >
    <Name> Max Moritzen </Name>
    <Gehalt> 2500 </Gehalt>
    <Beurteilung>
      <Beurteiler URI="#HansHansen091991" />
      <Text> Max macht sich hervorragend </Text>
      <Note> 1 </Note>
    </Beurteilung>
  </Person>
</Personaldaten>
```

*

zurück

vor

Verschlüsselung mehrerer XML-Elemente

```
<Personaldaten>
  <Person ID="MaxMoritzen012000" >
    <Name> Max Moritzen </Name>
    <EncryptedData Type="Content" >
      <EncryptionMethod Algorithm="3des-cbc" />
      <KeyInfo>
        <RetrievalMethod URI="http://www.beispiel.de/key#key1" />
      </KeyInfo>
      <CipherData>
        <CipherValue>kgfui97Gj87Hju6640kHH</CipherValue>
      </CipherData>
    </EncryptedData>
  </Person>
</Personaldaten>
```

*

zurück

vor

Verschlüsselung eines Wertes

```
<Personaldaten>
  <Person ID="MaxMoritzen012000" >
    <Name> Max Moritzen </Name>
    <Gehalt> 2500 </Gehalt>
    <Beurteilung>
      <Beurteiler URI="#HansHansen091991" />
      <Text> Max macht sich hervorragend </Text>
      <Note> 1 </Note>
    </Beurteilung>
  </Person>
</Personaldaten>
```

*

zurück

vor

Verschlüsselung eines Wertes

```
<Personaldaten>
  <Person ID="MaxMoritzen012000" >
    <Name> Max Moritzen </Name>
    <Gehalt>
      <EncryptedData Type="Content" >
        <CipherData>
          <CipherValue>ui6hz7fg</CipherValue>
        </CipherData>
      </EncryptedData>
    </Gehalt>
    <Beurteilung>
      <Beurteiler URI="#HansHansen091991" />
      <Text> Max macht sich hervorragend </Text>
      <Note> 1 </Note>
    </Beurteilung>
  </Person>
</Personaldaten>
```

*

zurück

vor

Verschlüsselung beliebiger Daten

```
⋮  
<Beurteilung>  
  <Beurteiler URI="#HansHansen091991" />  
  <Text>  
    <TextExtern URI="http://www.beispiel.txt/beurt.txt" />  
  </Text>  
  <Note> 1 </Note>  
</Beurteilung>
```

*

zurück

vor

Verschlüsselung beliebiger Daten

```
⋮
<Beurteilung>
  <Beurteiler URI="#HansHansen091991" Type="Person" />
  <Text>
    <EncryptedData ID="Beurteilung1" Type="text/txt" >
      <KeyInfo>
        <EncryptedKey CarriedKeyName="BeurteilungsKey" >
          <KeyInfo>
            <KeyName> SchluesselKey </KeyName>
            <ReferenceList>
              <DataReference URI="#Beurteilung1" />
            </ReferenceList>
          </KeyInfo>
          <CipherData>
            <CipherValue>67GUo;667hhdik</CipherValue>
          </CipherData>
        </EncryptedKey>
      </KeyInfo>
      <CipherData>
        <CipherReference URI="http://www.beispiel.txt/beurt.txt" />
      </CipherData>
    </EncryptedData>
  </Text>
  <Note> 1 </Note>
</Beurteilung>
```

*

zurück

vor

Super-Encryption

```
<Personaldaten>  
  <Person ID="MaxMoritzen012000" >  
    <Name> Max Moritzen </Name>  
    <Gehalt> 2500 </Gehalt>  
    <Beurteilung>  
      <Beurteiler URI="#HansHansen091991" />  
      <Text> Max macht sich hervorragend </Text>  
      <Note> 1 </Note>  
    </Beurteilung>  
  </Person>  
</Personaldaten>
```

*

zurück

vor

Super-Encryption

```
<Personaldaten>  
  <Encrypted Data>  
    <KeyInfo>  
      <KeyName>PersonaldatenzugangsrechteKey</KeyName>  
    </KeyInfo>  
    <CipherData>  
      <CipherValue>  
        hier steht die Verschlüsselung der Daten,  
        im Besonderen die Verschlüsselung der Beurteilung  
      </CipherValue>  
    </CipherData>  
  </EncryptedData>  
</Personaldaten>
```

*

zurück

vor

Syntax mit XML-Schema

*

zurück

vor

Syntax – EncryptedData und EncryptedKey

EncryptedData und EncryptedKey leiten sich von EncryptedType ab:

```
<complexType name="EncryptedType" >  
  <sequence>  
    <element ref = "EncryptionMethod" minOccurs="0" />  
    <element ref = "KeyInfo" minOccurs="0" />  
    <element ref = "CipherData" />  
    <element ref = "EncryptionProperties" minOccurs="0" />  
  </sequence>  
  <attribute name="id" type="ID" />  
  <attribute name="Type" type="anyURI" />  
</complexType>
```

*

zurück

vor

Syntax – EncryptedData und EncryptionMethod

```
<element name="EncryptedData" />
  <complexType >
    <sequence>
      <element name = "EncryptionMethod" minOccurs="0" />
        <complexType>
          <sequence>
            <any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
          </sequence>
          <attribute name="Algorithm" type="uriReference" use="required" />
        </complexType>
      </element>
      <element ref = "KeyInfo" minOccurs="0" />
      <element ref = "CipherData" />
      <element ref = "EncryptionProperties" minOccurs="0" />
    </sequence>
    <attribute name="id" type="ID" />
    <attribute name="Type" type="anyURI" />
  </complexType>
</element>
```

*

[zurück](#)

[vor](#)

Syntax – EncryptedData und KeyInfo

```
<element name="EncryptedData" /> <complexType >
  <sequence>
    <element ref = "EncryptionMethod" minOccurs="0" />
    <element name = "KeyInfo" minOccurs="0" />
      <complexType>
        <choice maxOccurs="unbounded" >
          <element ref="KeyName" />
          <element ref="KeyValue" />
          <element ref="RetrievalMethod" />
            spezielle Schlüsseldaten
          <any namespace="##other" />
        </choice>
        <attribute name="id" type="ID" />
      </complexType>
    </element>
    <element ref = "CipherData" />
    <element ref = "EncryptionProperties" minOccurs="0" />
  </sequence>
</complexType> </element>
```

*

[zurück](#)

[vor](#)

Syntax – EncryptedData und CipherData

```
<element name="EncryptedData" /> <complexType >
  <sequence>
    <element ref = "EncryptionMethod" minOccurs="0" />
    <element ref = "KeyInfo" minOccurs="0" />
    <element name="CipherData" > <complexType>
      <choice>
        <element name="CipherValue" />
        <element name="CipherReference" >
          <complexType>
            <sequence>
              <element ref="Transforms" minOccurs="0" />
            </sequence>
            <attribute name="URI" type="uriReference" use="required" />
          </complexType>
        </choice>
      </complexType> </element>
    <element ref = "EncryptionProperties" minOccurs="0" />
  </sequence>
</complexType> </element>
```

*

zurück

vor

Syntax – EncryptedKey

```
<element name="EncryptedKey" >  
  <complexType>  
    <extension base="EncryptedType" >  
      <sequence>  
        <element ref="ReferenceList" minOccurs="0" />  
      </sequence>  
      <attribute name="CarriedKeyName" type="string" />  
      <attribute name="Recipient" type="string" />  
    </extension>  
  </complexType>  
</element>
```

*

zurück

vor

EncryptedType

Syntax – EncryptedData und EncryptedKey

EncryptedData und EncryptedKey leiten sich von EncryptedType ab:

```
<complexType name="EncryptedType" >  
  <sequence>  
    <element ref = "EncryptionMethod" minOccurs="0" />  
    <element ref = "KeyInfo" minOccurs="0" />  
    <element ref = "CipherData" />  
    <element ref = "EncryptionProperties" minOccurs="0" />  
  </sequence>  
  <attribute name="id" type="ID" />  
  <attribute name="Type" type="anyURI" />  
</complexType>
```

*

[zurück](#)

Syntax – EncryptedKey und ReferenceList

```
<element name="EncryptedKey" >
  <complexType>
    <extension base="EncryptedType" >
      <sequence>
        <element name="ReferenceList" minOccurs="0" />
          <complexType>
            <sequence>
              <element ref="DataReference" minOccurs="0" />
              <element ref="KeyReference" minOccurs="0" />
            </sequence>
          </complexType>
        </element>
      </sequence>
      <attribute name="CarriedKeyName" type="string" />
      <attribute name="Recipient" type="string" />
    </extension>
  </complexType>
</element>
```

*

[zurück](#)

[vor](#)

Ein weiterer Vorschlag für XML-Encryption

- ähnlich aufgebaut wie vorige Syntax
- aber mehr Elemente
- Schlüssel von Referenzen getrennt

*

zurück

vor

Signaturen in XML

Signatur beinhaltet

- verschlüsselte Kennzahl zur Authentifikation
- Referenzen der signierten Daten

Digest-Algorithmus

berechnet Hash-Wert einer Nachricht, der

- keinen Rückschluß auf die Nachricht zuläßt
- praktisch kollisionsfrei ist

*

zurück

vor

Beispiel einer XML-Signatur

```
⋮  
<Beurteilung>  
  <Beurteiler URI="#HansHansen091991" />  
  <Text>  
    <TextExtern URI="http://www.beispiel.txt/beurt.txt" />  
  </Text>  
  <Note> 1 </Note>  
</Beurteilung>
```

*

zurück

vor

Beispiel einer XML-Signatur

```
<Signature>
  <SignedInfo>
    <CanonicalizationMethod Algorithm="REC-xml-c14n-20010315" />
    <SignatureMethod Algorithm="rsa-sha1" />
    <Reference URI="http://www.beispiel.de/beurt.txt" >
      <DigestMethod Algoritm="sha1" />
      <DigestValue>j7H6hd8kh7I9jcz7hgtjIM6K0Pgb</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>78bla83ALBlA5=3jghblA9hBLaa8</SignatureValue>
  <KeyInfo>
    <KeyValue>77564992370893420571690641</KeyValue>
  </KeyInfo>
</Signature>
```

*

zurück

vor

Syntax – Signature

```
<element name="Signature" >
  <complexType>
    <sequence>
      <element ref="SignedInfo" />
      <element ref="SignatureValue" />
      <element ref="KeyInfo" minOccurs="0" />
      <element ref="Object" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>
```

*

[zurück](#)

[vor](#)

Syntax – SignedInfo und Reference

```
<element name="SignedInfo" >
  <complexType>
    <sequence>
      <element ref="CanonicalizationMethod" />
      <element ref="SignatureMethod" />
      <element name="Reference" >
        <complexType>
          <sequence>
            <element ref="Transforms" minOccurs="0" />
            <element ref="DigestMethod" />
            <element ref="DigestValue" />
          </sequence>
          <attribute name="URI" type="uriReference" use="optional" />
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
```

*

zurück

vor

Nutzbarkeit in (XML-)Datenbanken

Generell ist jeder Schutz von Daten wünschens- und empfehlenswert.

- Protokolle sind flexibel.
- Aufwand für den Sender/Verfasser ist gering, Zugriff auf teilweise verschlüsselte Datenbank ist einfach.
- abgestufte Zugriffsrechte
- Effizienz

*

zurück

vor

Software

XML Security Suite von IBM

- Application Programming Interface
- Code frei verfügbar
- Benutzung nur zu Evaluationszwecken

*

zurück

vor

namespaces

Encryption: xmlns:xenc='http://www.w3.org/2001/04/xmlenc#'

Signature: xmlns:ds='http://www.w3.org/2000/09/xmldsig#'

*

[zurück](#)

[vor](#)

Referenzen

<http://www.w3.org>

<http://csrc.nist.gov>

<http://www.alphaworks.ibm.com>

* [zurück](#)