Università degli studi di Milano
CLS "tecnologie dell'informazione e comunicazione"

Seminario di
Laboratorio di Base di Dati 2

Xml Encryption
e
Digital Signature

Stefania ACCORDINO 670909
Valeria D'ERRICO 670365

Sicurezza dei dati

Grazie alla sua caratteristiche di flessibilità ed espandibilità, XML è molto utilizzato come formato di scambi dati tra applicazioni.

↓ **SVANTAGGIO**

Carenza nella protezione dei dati

Sicurezza dei dati

In molti ambiti (commerciale, militare, etc.) risulta molto importante avere a disposizione uno **strumento in grado di assicurare la protezione dei dati** che vengono scambiati tra le applicazioni.

SSL (Secure Socket Layer)

- E' attualmente lo strumento maggiormente utilizzato per garantire la sicurezza dei dati;
- permette di creare un canale protetto per lo scambio di dati tra due applicazioni .

Svantaggio di SSL

- Non permette di proteggere singole parti del documento, ma per forza tutto il documento intero;
- Protegge il dato solo durante lo scambio; una volta che esso arriva a destinazione viene decifrato e non è quindi più protetto.

XML Encryption

- In XML il problema della sicurezza è affrontato dalla specifica di **XML Encryption**;
- sviluppato congiuntamente dall'IETF e dal W3C

XML Encryption

- E' una tecnologia in grado di proteggere un documento XML o parti di esso;
- Inoltre è possibile cifrare sia un valore arbitrario (file XML), sia un elemento XML e sia il contenuto di un elemento XML;

XML Encryption

- permette di cifrare gli elementi di un documento XML utilizzando i più diffusi algoritmi di crittografia, sia a chiave simmetrica che a chiave pubblica.
- Il risultato di un dato criptato è un elemento *EncryptedData*

Struttura

```
<EncryptedData Id? Type?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievaMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>?
    <CipherReference URI?>?
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>
```


Struttura

```
<EncryptedData Id? Type?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievaMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>?
    <CipherReference URI?>?
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>
```

Indica l'algoritmo di cifratura utilizzato

Struttura

```
<EncryptedData Id? Type?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievaMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>?
    <CipherReference URI?>?
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>
```

Indica la chiave usata per cifrare i dati

Struttura

```
<EncryptedData Id? Type?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievaMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>?
    <CipherReference URI?>?
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>
```

Contiene i dati cifrati o un riferimento alla loro posizione

Struttura

```
<EncryptedData Id? Type?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievaMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>?
    <CipherReference URI?>?
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>
```

Fornisce altre informazioni sulla generazione dei dati cifrati

Esempio 1/3

```
<?xml version="1.0"?>
<acquirente>
  <nome>Mario Rossi</nome>
  <carta_di_credito>
    <numero>1234 5678 9123
4567</numero>
    <scadenza>12/2003</scadenza>
  </carta_di_credito>
</acquirente>
```


Esempio 2/3

Contiene i dati protetti e le informazioni sull'algoritmo utilizzato

Tipo di algoritmo usato per decifrare

```
<?xml version="1.0"?>
<acquirente>
  <nome>Mario Rossi</nome>
  <carta_di_credito>
    <EncryptedData Id="Enc_esempio"
Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
      <EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-
1_5" />
```

L'attributo type permette di capire quali elementi decifrare

Esempio 3/3

Contiene le informazioni sulla chiave da utilizzare

```
<ds:KeyInfo
xmlns:ds="http://www.w3.org/2000/09/xmldisg#">
      <ds:KeyName>M_Rossi</ds:KeyName>
    </ds:KeyInfo>
    <CipherData>
      <CipherValue>A55BDBC11</CipherValue>
    </CipherData>
  </EncryptedData>
  <scadenza>12/2003</scadenza>
 </carta_di_credito>
</acquirente>
```

racchiude i dati codificati

Vantaggi

- Questo standard cripta anche le informazioni appartenenti al dato da proteggere (es. cripta "n_carta")
- Visto che è difficile inserire in un documento XML un dato binario, il sistema trasformerà il dato utilizzando un algoritmo Base64

Autenticazione del documento.

XML Digital Signature

introduzione

- La sicurezza dei dati è correlata al problema dell'**autenticazione** e dell'**integrità** dei documenti, oltre alla segretezza.
- Per l'autenticazione di un documento si ricorre alla tecnica della digital signature.
- La firma si ottiene usando un **algoritmo di cifratura a chiave pubblica.**

Hash di un documento

- Non si cifra il documento, ma il suo digest,detto anche hash.
- La funzione hash è molto sensibile ad ogni cambiamento del documento e quindi differenti versioni produrranno digest diversi.
- Il digest è usato per verificare l'integrità del documento.

Xml signature

- Tutti questi aspetti vengono considerati nello standard **XML DIGITAL SIGNATURE**. (IETF e W3C)
- Si tratta di una tecnologia che permette di cifrare i documenti xml,o parte di essi, garantendone autenticità e integrità.

Xml signature

- Per creare una xml signature per un contenuto digitale di un documento si deve
  1) Si valida la firma stessa, assicurandosi l'integrità del valore digest dell'elemento firmato
  2) Dopo si valida il valore digest di ciascun dato.

- Il processo di validazione prevede 2 passi:
  1) Si calcola un digest per ciascun dato-oggetto-che si vuole firmare.
  2) Esso viene inserito, insieme ad altri valori, in un elemento XML.
- Infine viene generato un digest dell'elemento che viene firmato con chiave privata.

Struttura generica dell'elemento <Signature>

<Signature ID?>
<SignedInfo>
<CanonicalizationMethod/>
<SignatureMethod/>
(<Reference URI?>
<DigestMethod>
<DigestValue>
</reference>)+
</SignedInfo>
<SignatureValue>
(<KeyInfo>)?
(Object ID?)*
</Signature>

? indica 0 o 1 occorrenza,
+ indica 1 o + occorrenze
* indica 0 o + occorrenze

Esempio/1 con Digital Signature

```
<?xml version="1.0" ?>
<signature xmlns=
  http://www.w3.org/2000/09/xmldisg# >
  (1)<SignedInfo Id="esempio_firma">
  <CanonicalizationMethod
     Algorithm=
  http://www.w3.org/TR/2001/REC-xml-c14n-
  20010315/>
```

Esempio/2 con Digital Signature

<**SignatureMethod Algorithm**=
http://www.w3.org/2000/097xmldisg#dsa-sha1/>
<**Reference**
URI="http://nome_dominio/esempio.xml#acq">
<**DigestMethod Algorithm**=
http://www.w3.org/2000/09/xmldsig#sha1"/>

Esempio/3 con Digital Signature

<**DigestValue**>
J4fdsu4325riwjerfow732ewjdp9
</DigestValue>
< **/Reference**>
< **/SignedInfo** >
(2)<SignatureValue>
ME4324W
</SignatureValue>

Esempio/4 con Digital Signature

(3)<KeyInfo>
<DSAkeyValue>
…..
< /DSAkeyValue>
</KeyInfo>
</Signature>

<Transform>

- In alcuni casi il verificatore della firma deve effettuare delle trasformazioni prima di poter correttamente controllare la firma.
- Le trasformazioni si definiscono con l'elemento <transform> e sono incluse all'interno dell'elemento <reference>.

<Transform>

- Indica il tipo di trasformazione da effettuare e l'elemento su cui effettuarlo, contenuto in <data reference>

<Transform>

<Signature ID?>
<SignedInfo>
<CanonicalizationMethod/>
<SignatureMethod/>
(<Reference URI?>
(<**Transform**>
<data reference>
</**transform**>)?
<DigestMethod>
<DigestValue>
</reference>)+
</SignedInfo>
<SignatureValue>
(<keyInfo>)?
(Object ID?)*
</Signature>

<Transform>

- Il verificatore della firma deve effettuare 2 trasformazioni prima di poterla controllare:

1)la prima per decifrare l'elemento contenuto in data reference;

2)la seconda per scrivere l'elemento in forma canonica.

Fine
xml encryption e digital signature