Composition de Crypto - 2011

La composition est composée de plusieurs exercices

Le support de cours est permis. La majeure partie des questions font appel au sens pratique.

1. Facturation en ligne

Un fournisseur a besoin d'envoyer une facture à son client. Pour cela il utilise les services d'un serveur de facturation en ligne, dans laquelle le fournisseur envoie la facture chiffrée et signée au serveur. Le serveur après déchiffrement et vérification de la signature du fournisseur, va transmettre la facture chiffrée au client après opposition d'une seconde signature sur la facture. L'algorithme de chiffrement est l'AES. Chaque chiffrement est réalisé par une clé de session, transmise chiffrée RSA 2048bits.

a) De quel type hiérarchique est la seconde signature et pour quelle raison ?
b) Que peut offrir le fournisseur de services comme mécanisme de non-répudiation ?
c) Après déchiffrement de la facture par le serveur, décrire la procédure du re-chiffrement de la facture.
d) La signature est de taille RSA 2048 bits. Chaque signataire a une carte à puce sur laquelle sont mises les clés privées de signature et de chiffrement sous la forme des restes chinois, ainsi que les certificats associés. Définir le mapping de la carte, sans préciser les droits d'accès.
e) Le fournisseur d'API de sécurité (de la signature) a la possibilité de fournir une interface de type CryptoAPI, PKCS11 ou accès direct à la carte aux fichiers clés et certificats. En une dizaine de lignes, justifier le choix d'une ou plusieurs interfaces.

L'implémentation des mécanismes de sécurité va utiliser la sécurité XML décrits dans les paragraphes suivants.
Pour toutes les valeurs binaires codifées en base64 mettre la valeur : vAlxyz….1234

2. Gestion de clés par XML (XKMS)

La spécification XKMS est composée de deux protocoles :

1. X-KISS (XML Key Information Service Specification) pour les requêtes de localisation et de validation des clés publiques ;
2. X-KRSS (XML Key Registration Service Specification) pour enregistrer, renouveler, révoquer et obtenir des clés.

Une requête XKMS est envoyée à un serveur tiers (webserices) spécialisé dans la gestion de certificats et de confiance. Le serveur va analyser la requête puis répondre à l'émetteur de la requête.

XKMS donne des exemples de requêtes, ainsi que les réponses associées :

Requête X-KISS (demande de clés)

```
<Locate>
  <Query>
    <ds:KeyInfo>
      <ds:RetrievalMethod
       URI="http://www.server.wou/webservice/123456"
Type="http://www.w3.org/2000/09/XML/DSIG#X509Data"/>
    </ds:KeyInfo>
  </Query>
  <Respond>
    <string>KeyName</string>
    <string>KeyValue</string>
  </Respond>
</Locate>
```

Réponse X-KISS

```
<LocateResult>
  <Result>Success</Result>
  <Answer>
    <ds:KeyInfo>
      <ds:KeyName>O=wouroudistan.org OU="CA"
                CN="Alice"</ds:KeyName>
      <ds:KeyValue>...</ds:KeyValue>
    </ds:KeyInfo>
  </Answer>
</LocateResult>
```

RetreivalMethod donne le lien de récupération d'une information concernant un certificat référencé par 123456 par le webservice http://www.server.wou/webservice.

Le schema défini par la norme XML/DSIG régissant KeyInfo donne les informations suivantes :

```
<element name="KeyInfo" type="ds:KeyInfoType"/>
  <complexType name="KeyInfoType" mixed="true">
   <choice maxOccurs="unbounded">
    <element ref="ds:KeyName"/>
    <element ref="ds:KeyValue"/>
    <element ref="ds:RetrievalMethod"/>
    <element ref="ds:X509Data"/>
    <element ref="ds:PGPData"/>
    <element ref="ds:SPKIData"/>
    <element ref="ds:MgmtData"/>
    <any processContents="lax" namespace="##other"/>
    <!-- (1,1) elements from (0,unbounded) namespaces -->
   </choice>
   <attribute name="Id" type="ID" use="optional"/>
  </complexType>
```

```
<element name="KeyValue" type="ds:KeyValueType"/>
  <complexType name="KeyValueType" mixed="true">
   <choice>
    <element ref="ds:DSAKeyValue"/>
    <element ref="ds:RSAKeyValue"/>
    <any namespace="##other" processContents="lax"/>
   </choice>
  </complexType>
```

```
<element name="RSAKeyValue" type="ds:RSAKeyValueType"/>
  <complexType name="RSAKeyValueType">
   <sequence>
    <element name="Modulus" type="ds:CryptoBinary"/>
    <element name="Exponent" type="ds:CryptoBinary"/>
   </sequence>
  </complexType>
```

```
<element name="X509Data" type="ds:X509DataType"/>
  <complexType name="X509DataType">
   <sequence maxOccurs="unbounded">
    <choice>
     <element name="X509IssuerSerial"
type="ds:X509IssuerSerialType"/>
     <element name="X509SKI" type="base64Binary"/>
     <element name="X509SubjectName" type="string"/>
     <element name="X509Certificate" type="base64Binary"/>
     <element name="X509CRL" type="base64Binary"/>
     <any namespace="##other" processContents="lax"/>
    </choice>
   </sequence>
  </complexType>
  <complexType name="X509IssuerSerialType">
   <sequence>
    <element name="X509IssuerName" type="string"/>
    <element name="X509SerialNumber" type="integer"/>
   </sequence>
  </complexType>
```

a) Les données binaires étant représentées en base64 (chaque 6 bits d'un flux d'octets binaires est représenté par un caractère imprimable de 8 bits) avec padding final. Compléter le ds :KeyValue de la réponse du serveur, en indiquant la taille des diverses zones. (Pour l'exposant prendre le nombre 0x010001L, et pour le modulo la valeur énoncée au paragraphe 1).

b) L'émetteur veut récupérer uniquement la valeur du certificat, modifier dans l'exemple, la requête XKMS et définir la réponse du serveur en indiquant la taille de la codification du certificat.

3. Signature XML

La signature utilisée se base sur la norme XML/DSIG dont la balise principale est l'élément Signature qui est décrit dans le schema comme suit :

Le schema défini par la norme XML/DSIG régissant KeyInfo donne les informations suivantes :

```
<element name="Signature" type="ds:SignatureType"/>
  <complexType name="SignatureType">
   <sequence>
    <element ref="ds:SignedInfo"/>
    <element ref="ds:SignatureValue"/>
    <element ref="ds:KeyInfo" minOccurs="0"/>
    <element ref="ds:Object" minOccurs="0"
maxOccurs="unbounded"/>
   </sequence>
   <attribute name="Id" type="ID" use="optional"/>
  </complexType>
```

```
<element name="SignedInfo" type="ds:SignedInfoType"/>
<complexType name="SignedInfoType">
 <sequence>
  <element ref="ds:CanonicalizationMethod"/>
  <element ref="ds:SignatureMethod"/>
  <element ref="ds:Reference" maxOccurs="unbounded"/>
 </sequence>
 <attribute name="Id" type="ID" use="optional"/>
</complexType>
```

```
<element name="SignatureValue" type="ds:SignatureValueType"/>
  <complexType name="SignatureValueType">
   <simpleContent>
    <extension base="base64Binary">
     <attribute name="Id" type="ID" use="optional"/>
    </extension>
   </simpleContent>
  </complexType>
```

```
<element name="Reference" type="ds:ReferenceType"/>
  <complexType name="ReferenceType">
   <sequence>
    <element ref="ds:Transforms" minOccurs="0"/>
    <element ref="ds:DigestMethod"/>
    <element ref="ds:DigestValue"/>
   </sequence>
   <attribute name="Id" type="ID" use="optional"/>
   <attribute name="URI" type="anyURI" use="optional"/>
   <attribute name="Type" type="anyURI" use="optional"/>
  </complexType>
```

Où SignatureValue va contenir la valeur base64 de la signature.

La structure de l'élément SignedInfo inclut l'algorithme de canonisation, un algorithme de signature, et une ou plusieurs références. L'élément SignedInfo peut contenir un attribut optionnel ID qui lui permet d'être référencé par d'autres signatures et objets.
CanonicalizationMethod définit la transformation canonique à appliquer sur le contenu des balises xml référencées par Reference.

Soit l'exemple de la signature suivante :

```
<Invoice>
<Number authenticate='true'>1234</Number>
<Date authenticate='true'>01-04-2011</Date>
<Customer authenticate='true' >
        <Name>WOUROUDISTANI</Name>
        <Surname>MIKE</Surname>
        <Ad line='1' >Flower Street</Ad>
        <Ad line='2' >Tulip Building</Ad>
        <City>SunflowerVille</City>
        <Code>1000</Code>
</Customer>
<Seller>
        <Name>WOUROUDISTANI</Name>
        <Surname>MIKE</Surname>
        <Ad mainad='true'>Flower Street</Ad>
        <City>SunflowerVille</City>
        <Code>1000</Code>
</Customer>
<Amount authenticate='true'>1234,56</Amount>
<Currency authenticate='true'>USD</Currency>
<Remarks>
        <Remark>Do not be late</Remark>
        <Remark>to pay us</Remark>
</Remarks>
<ds :Signature>
   <ds:SignedInfo>
     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/XML/DSIG-more#rsa-sha256"/>
        <ds:Reference URI="#xpointer(//*[@authenticate='true'])">
            <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            </ds:Transforms>
              <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
              <ds:DigestValue>8gHP8Ul11ZJqCswsbYFwuuJq2TApNvTOwVM1Okf9LWc=</ds:DigestValue>
         </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>LYO+hpYgpHYua……LjUEKdnJX5Hiw==</ds:SignatureValue>
 </ds :Signature>
</Invoice>
```

Application de la signature :
Les balises référencées par Reference sont enchaînées, canonisées, puis hashées par l'algorithme défini dans DigestMethod. La balise SignedInfo est créée avec son contenu, puis hashée et signée par la clé de signature telle que définie dans SignatureMethod. La balise Signature est créée en insérant les balises SignedInfo et SignatureValue.

a) Dans l'exemple précédent quelles sont les balises sur lesquelles s'appliquent la canonisation et le hash initial ?
b) Pour quelle raison la signature n'est pas appliquée directement sur le premier hash généré, mais sur le hash de la balise SignedInfo ?
c) La canonisation c14n donnée dans l'exemple consiste entre autre :
   - à mettre en forme les CR/LFs et tabulations
   - Mettre en ordre les attributs d'un élément
   - Faire apparaître les attributs par défaut.

   Donner deux raisons pour la nécessité de la canonisation.
d) En considérant que les éléments Seller et Customer respectent le même schema, canoniser l'élément Customer.
e) Côté serveur de facturation décrire la vérification de signature.
f) Dans l'exemple précédent compléter l'élément Signature pour inclure le certificat du signataire référencé par un nom de clé CN=Mike Wouroudistani
g) **Question Bonus:** Comment signer par XML/DSIG des données arbitraires non XML ?

4. Chiffrement XML

Le chiffrement des informations est réalisé par la norme XMLENC.

a) Soit un achat réalisé par internet et qui doit générer les données xml suivantes :

```
<?xml version='1.0'?>
 <PaymentInfo xmlns='http://example.org/paymentv2'>
   <Name>Jean Dupont</Name>
   <CreditCard Limit='5,000' Currency='USD'>
     <Number>4019 1234 5678 9999</Number>
     <Issuer>Wouroudistan Bank</Issuer>
     <Expiration>04/11</Expiration>
     <SecurityCode>1234</ SecurityCode >
   </CreditCard>
 </PaymentInfo>
```

Comparer entre les deux méthodes suivantes de chiffrement où EncryptedData contient les données chiffrées

Méthode 1

```
<?xml version='1.0'?>
  <PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name>Jean Dupont</Name>
    <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'
     xmlns='http://www.w3.org/2001/04/xmlenc#'>
      <CipherData>
        <CipherValue>A23B45C56</CipherValue>
      </CipherData>
    </EncryptedData>
</PaymentInfo>
```

Méthode 2

```
<?xml version='1.0'?>
  <PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name> Jean Dupont </Name>
    <CreditCard Limit='5,000' Currency='USD'>
      <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
Type='http://www.w3.org/2001/04/xmlenc#Content'>
        <CipherData>
          <CipherValue>A23B45C56</CipherValue>
        </CipherData>
      </EncryptedData>
    </CreditCard>
    </PaymentInfo>
```

b) Le chiffrement utilise une clé symétrique AES en mode CBC

```
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
      Type='http://www.w3.org/2001/04/xmlenc#Element'/>
    <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#aes128-cbc/>
    <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/XML/DSIG#'>
      <ds:KeyName>Jean Dupont</ds:KeyName>
    </ds:KeyInfo>
    <CipherData><CipherValue>DEAD……BEEF</CipherValue></CipherData>
  </EncryptedData>
```

EncryptedMethod désigne l'algorithme et le mode de chiffrement
KeyName est le nom de la clé AES utilisée entre les deux parties.
CipherData est la valeur chiffrée des informations.

Dans ce type de chiffrement comment la clé a été échangée ?
Décrire les opérations de chiffrement et déchiffrement des données et la création/lecture des diverses balises xml entre l'émetteur et le récepteur?

c) Au lieu d'utiliser la méthode en b pour l'utilisation de clé symétrique, une clé symétrique de session doit être générée et chiffrée par le certificat du destinataire :

```
      <EncryptedData Id='ED' xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#aes128-cbc'/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/XML/DSIG#'>
    <ds:RetrievalMethod URI='#EK' Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>
                  <ds:KeyName>Philippe Le Roi</ds:KeyName>
  </ds:KeyInfo>
  <CipherData><CipherValue>DEADBEEF</CipherValue></CipherData>
</EncryptedData>
```

CipherData indique la donnée chiffrée.
RetrievalMethod indique un lien local dans le document XML sur la clé AES chiffrée, définie par EncryptedKey.

```
<EncryptedKey Id='EK' xmlns='http://www.w3.org/2001/04/xmlenc#'>
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
  <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/XML/DSIG#'>
    <ds:KeyName>Jean Dupont</ds:KeyName>
  </ds:KeyInfo>
  <CipherData><CipherValue>xyz…abc</CipherValue></CipherData>
  <ReferenceList>
    <DataReference URI='#ED'/>
  </ReferenceList>
</EncryptedKey>
```

Que représentent :

- CipherData de EncryptedKey ?
- KeyName de EncryptedData ?
- KeyName de EncryptedKey ?
- DataReference de EncryptedKey ?

Dans ce type de chiffrement comment la clé a été échangée ? Décrire les opérations de chiffrement et déchiffrement des données et la création/lecture des diverses balises xml entre l'émetteur et le récepteur?

d) Quelle technique de sécurité XML proposez-vous pour récupérer le certificat du destinataire?
e) Les données à chiffrer ont une taille en clair de 753 octets, calculer la taille de la donnée chiffrée binaire puis en déduire la taille de la donnée chiffrée en base64.
f) **Question Bonus :** A partir des informations précédentes justifier l'application de XML/DSIG en premier sur un document XML, suivi du chiffrement XML/ENC.