Composition de Sécurité et Crypto - 2010

La composition est composée de plusieurs exercices

Il est recommandé aux élèves de bien choisir l'ordre des parties selon leurs compétences et rapidités.

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

1. Echange sécurisé

Alice envoie un fichier chiffré et signé à Bob suite une phase d'authentification mutuelle. Les divers services de sécurité Authentification, confidentialité et non répudiation utilisent les certificats, le RSA 2048 bits, le chiffrement AES 128 bits et le SHA256. Il y a usage d'un certificat différent par service de sécurité.
Alice utilise une carte à puce qui en plus de ses propres certificats, va contenir le certificat d'une autorité de certification (4096 bits) commune à Alice et Bob.

a) Le fichier signé et chiffré utilise l'enveloppe pkcs7 signed-enveloped dont la description est la suivante :

```
SignedAndEnvelopedData ::= SEQUENCE {
     version Version,
     recipientInfos RecipientInfos,
     digestAlgorithms DigestAlgorithmIdentifiers,
     encryptedContentInfo EncryptedContentInfo,
     certificates
        [0] IMPLICIT ExtendedCertificatesAndCertificates
          OPTIONAL,
     crls
       [1] IMPLICIT CertificateRevocationLists OPTIONAL,
     signerInfos SignerInfos }

RecipientInfos ::= SET OF RecipientInfo

   RecipientInfo ::= SEQUENCE {
     version Version,
     issuerAndSerialNumber IssuerAndSerialNumber,
     keyEncryptionAlgorithm

       KeyEncryptionAlgorithmIdentifier,
     encryptedKey EncryptedKey }

   EncryptedKey ::= OCTET STRING


   EncryptedContentInfo ::= SEQUENCE {
     contentType ContentType,
     contentEncryptionAlgorithm
       ContentEncryptionAlgorithmIdentifier,
     encryptedContent
       [0] IMPLICIT EncryptedContent OPTIONAL }

   EncryptedContent ::= OCTET STRING

SignerInfo ::= SEQUENCE {
     version Version,
     issuerAndSerialNumber IssuerAndSerialNumber,
     digestAlgorithm DigestAlgorithmIdentifier,
     authenticatedAttributes
       [0] IMPLICIT Attributes OPTIONAL,
     digestEncryptionAlgorithm
       DigestEncryptionAlgorithmIdentifier,
     encryptedDigest EncryptedDigest,
     unauthenticatedAttributes
       [1] IMPLICIT Attributes OPTIONAL }

   EncryptedDigest ::= OCTET STRING
```

Question : Commenter les séquences SignedAndEnvelopedData, EncryptedContentInfo, RecipientInfo, SignerInfo.

b) Décrire tous les échanges de sécurité entre Alice et Bob, en commentant d'une façon claire les échanges et les techniques de sécurité utilisées.
c) Définir le mapping de la carte d'Alice en se basant sur des clés privées sauvegardées sous la forme (S,N).
d) Choisir une carte entre 8K, 16k et 32K.
e) La signature utilise le mode PKCS1 v1.5 défini de la façon suivante :

$$EM = 0x00 \,||\, 0x01 \,||\, PS \,||\, 0x00 \,||\, T\,.$$

Où PS est constitué d'octets à 0xFF,
T est l'implémentation DER de :

```
DigestInfo ::= SEQUENCE {
    digestAlgorithm AlgorithmIdentifier,
    digest OCTET STRING
}
```

Où le digest est la valeur hash.

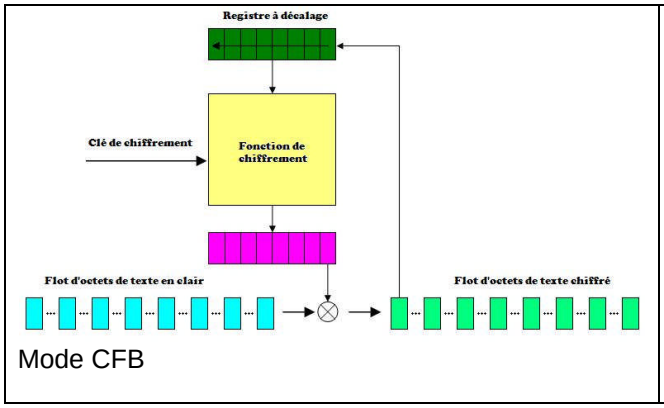
Lors de la vérification de la signature d'Alice, on peut obtenir les cas suivants après application de la clé publique d'Alice sur la signature EM' = Clé Publique Alice (Signature (EM))

- $EM'_1 = EM$
- EM'_2= suite de valeurs aléatoires,
- $EM'_3 = 0x00 \,||\, 0x01 \,||\, PS1 \,||\, 0x00 \,||\, T$. PS1 = suite de 0
- $EM'_4 = 0x00 \,||\, 0x01 \,||\, PS1 \,||\, 0x00 \,||\, T1$. T1 a les champs digestAlgorithm et digest erronés
- $EM'_5 = 0x00 \,||\, 0x01 \,||\, PS1 \,||\, 0x00 \,||\, T2$. T2 a le champ digest erroné.

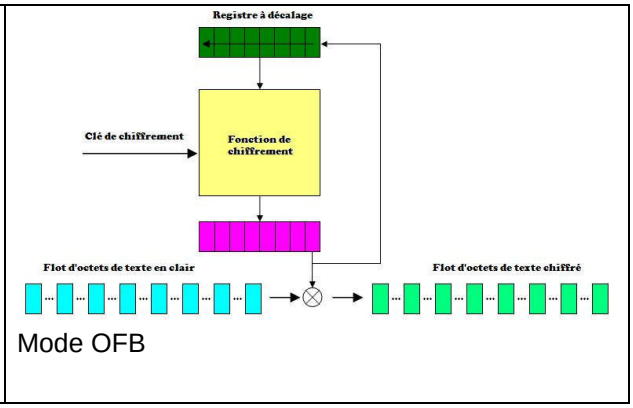
Question : Interpréter en une ligne chacune des valeurs EM'_i obtenue.

f) Bob veut envoyer à Alice un accusé de réception indiquant le résultat de la vérification. Quelles sont les techniques de sécurité que vous proposez pour l'accusé de réception ?

2. Modes de chiffrement CFB et OFB



Mode CFB



Mode OFB

La fonction de chiffrement utilisée est l'AES 128 bits.
Dans les deux modes les données ne sont pas directement chiffrées par l'AES mais chiffrées par une fonction XOR avec la sortie du chiffrement AES d'un registre à décalage.
Les deux modes étant utilisés pour chiffrer des flux de données, un octet ou un bit est chiffré chaque fois.
Dans cet exercice on va étudier l'effet d'une erreur sur les données chiffrées.

a) Pour les mode CFB et OFB, dessiner les diagrammes de déchiffrement, et écrire les formules de déchiffrement côté destinataire, liant Di (registre à décalage), AES, K (clé de chiffrement), C'i (octet chiffré), Ci(octet en clair), IV (valeur initiale du registre à décalage).
b) Pour le mode CFB si un octet chiffré C'i arrive altéré C''i chez le destinataire, quelle est la conséquence sur le déchiffrement des données chez le destinataire ?
c) Pour le mode OFB si un octet chiffré C'i arrive altéré C''i chez le destinataire, quelle est la conséquence sur le déchiffrement des données chez le destinataire ?
d) On va supposer maintenant que pour les deux modes, le décalage dans le registre à décalage se fait par un bloc de 16 octets, ce qui revient à faire du chiffrement des données par bloc de 16 octets. Pourquoi le registre à décalage ne sert plus à rien ? Ecrire les équations lors du chiffrement et déchiffrement pour les deux modes liant AES, K (clé de chiffrement), M'i (bloc chiffré), Mi(bloc en clair), , IV (valeur initiale du registre à décalage).
e) Pour chacun des deux modes CFB et OFB quelle est la conséquence d'un bloc chiffré B''i altéré qui arrive chez le destinataire sur le déchiffrement.
f) Conclure sur l'efficacité des deux modes sur la resynchronisation lors du déchiffrement.

3. Mots de Passe Base de données

Soit à protéger l'accès à une base de données selon plusieurs schémas.

a) L'application est une application client/serveur classique. L'administrateur définit pour chaque utilisateur un couple (username,password) d'accès à la base de données. Quelles sont les conséquences de point de la licence de la SGBD?
b) L'application est une application client/serveur classique. L'administrateur définit un couple (username,password) pour l'application d'accès à la base de données. Les utilisateurs sont définis par des couples $(username,password)_{applicatif}$ dans une table Users appartenant à l'utilisateur application. Quand un utilisateur de l'application veut l'utiliser, il présente son couple $(username,password)_{applicatif}$, l'application se connecte à la base de données en présentant son username et password puis va vérifier si le couple $(username,password)_{applicatif}$ de l'utilisateur appartient dans la table Users. Quels sont les risques (citer deux) dans ce type de schéma et comment se protéger contre ces risques ?
c) L'application est une application web et implémente un schéma d'accès à la base de données identique à b. Quels sont les risques (citer deux) dans ce type de schéma et comment se protéger contre ces risques ?