Chiffrement à clef publique, authentification et distribution des clefs

Plan

- Les principes de l'authentification de message
- Les fonctions de hachage sécurisées
  - SHA-1 et HMAC
- Principes de chiffrement à clef publique
- Algorithmes de chiffrement à clef publique
  - RSA
- Gestion des clefs
  - Diffie-Hellman
  - Certificats numériques

L'authentification

- Protection contre les attaques actives
  - corruption des données et des transactions
- Besoins – vérifier :
  1. L'identité de l'auteur/émetteur du message,
  2. Le contenu du message n'a pas été modifié (intégrité),
  3. Que le message a été transmis sans délai et dans le bon ordre.

Les principes de l'authentification de message

- L'authentification avec un chiffrement traditionnel
  - L'émetteur et le récepteur partage une clef
  - On ajoute au message :
    - un champ de contrôle de corruption (un code standard de détection d'erreur)
    - un numéro de séquence et/ou un horodatage
  - On chiffre totalement le message à l'émetteur et on déchiffre au récepteur, puis on vérifie s'il y a eu corruption
- L'authentification sans chiffrement total du message
  - Ne nécessite pas ou très peu de chiffrement :
    => efficacité
  - Un champ d'authentification (appelé signature) est ajouté à chaque message
  - Le message est transmis en clair
    - dissocie l'authentification de la confidentialité
  - Utilise de nombreux mécanismes de chiffrement

L'authentification sans chiffrement

Il existe 4 techniques :

- Avec "Message Authentication Code" (MAC)
  - Utilise généralement un chiffrement symétrique
  - Mais peut utiliser un chiffrement asymétrique
- Avec une fonction de hachage non-réversible (HMAC)
  - Avec chiffrement symétrique
  - Avec chiffrement asymétrique
  - Avec une valeur secrète

Authentification sans chiffrement avec MAC

- MAC = F(Key, Message)
- La taille du MAC est petit et fixe
- Le message n'a pas été modifié si MAC reçu et recalculé sont identiques
- L'émetteur est certifié : la clef est secrète
- Le message est bon : il contient un numéro de séquence et une date (=> il n'a pas été modifié)

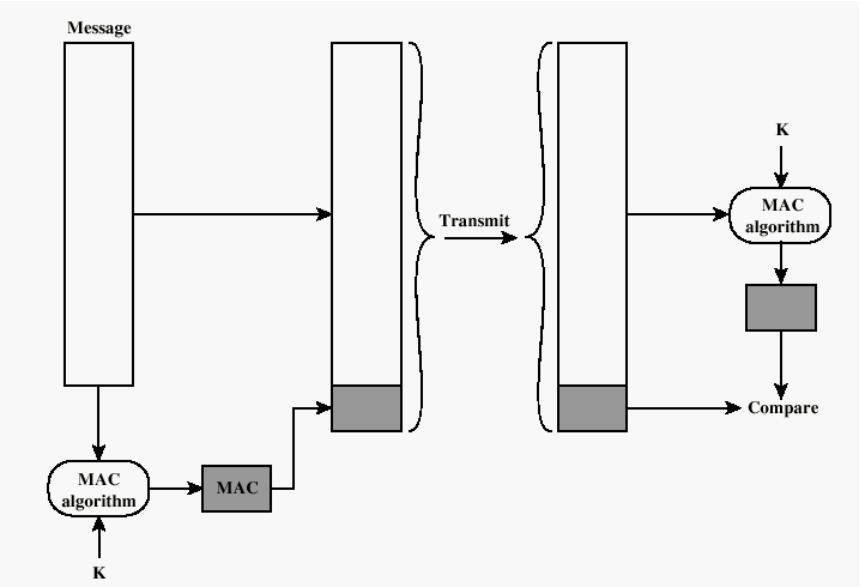


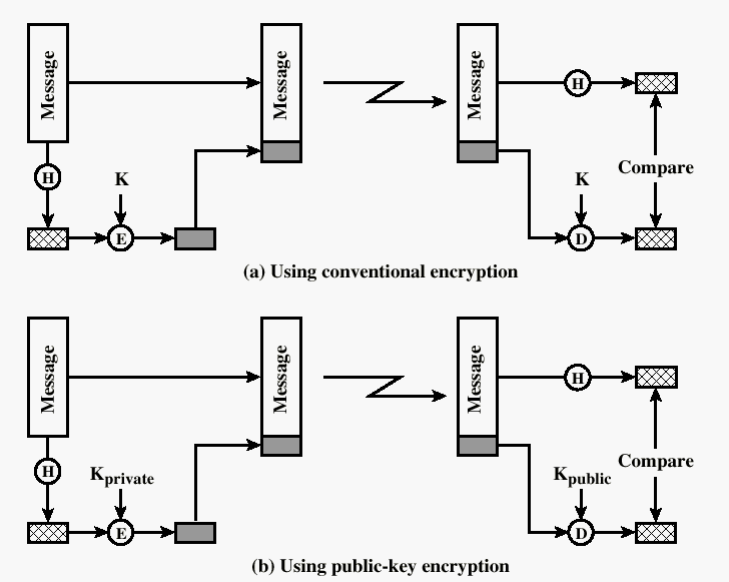
Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

7

Avec une fonction de hachage non-réversible

- Une fonction de hachage:
  - Ne nécessite pas de clef
  - Est plus rapide qu'un algorithme de chiffrement classique
  - N'est pas soumise aux restrictions concernant les exportations (des armes de guerre).

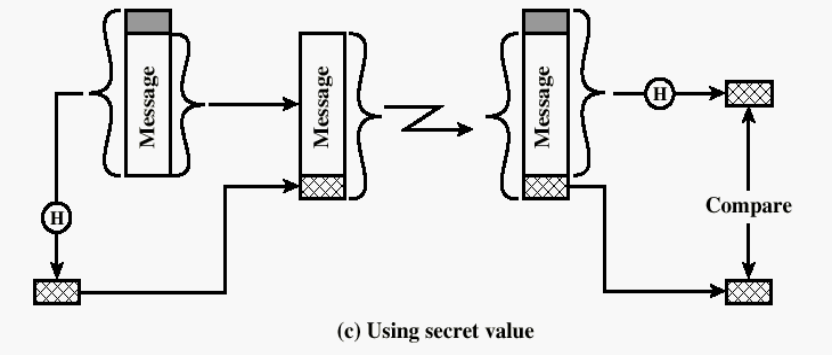
Avec fonction de hachage non-réversible



9

Avec fonction de hachage non-réversible

- Les partenaires partagent une valeur secrète qui est ajoutée au message avant le hachage mais n'est pas transmise.



Fonctions de hachage sécurisées

- Production d'une signature (”fingerprint").
- Propriétés de la fonction de hachage H :
  1. H utilise un bloc de données d'une taille quelconque
  2. H produit une signature de longueur fixe
  3. H(x) est facile à calculer.
  4. Il est calculatoirement impossible de trouver x tel que H(x)=h. (non réversibilité)
  5. Il est calculatoirement impossible de trouver x=/=y tels que H(x)=H(y). ("Weak collision resistance")
  6. Il est calculatoirement impossible de trouver (x, y) tels que H(x)= H(y). ("Strong collision resistance : against Birthday attack")

Fonction de hachage triviale

	bit 1	bit 2	• • •	bit *n*
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block *m*	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

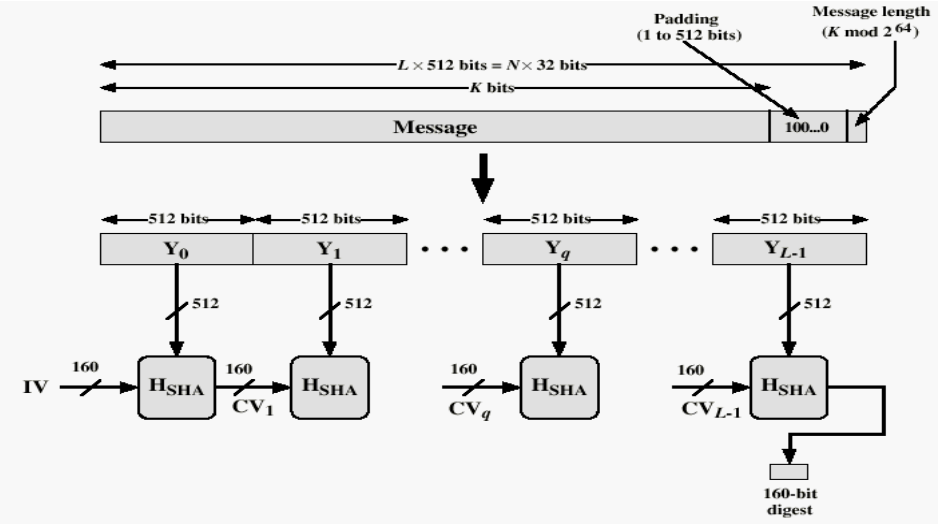
Figure 3.3 Simple Hash Function Using Bitwise XOR

- Un décalage circulaire d'un bit de la valeur de hachage après chaque bloc améliore la qualité du hachage.
- Faiblesse : la non-réversibilité n'est pas assurée, il est facile d'ajouter à n'importe quel (faux) message un bloc (de n bits) tel que le nouveau message produit la signature désirée (celle du vrai message).

SHA-1

- "Secure Hash Algorithm"
- Normalisé en 1995 FIPS PUB 180-1
- Signature de 160 bits, longueur maximum du message 2^{64} bits, blocs de 512 bits
- 4 tours de 20 étapes
- Utilise une constante K_t dont la valeur dépend du tour:
  - Si t= 1, 2, 3 et 4 alors K_t est la partie entière de respectivement : $2^{30} \times 2^{1/2}$, $2^{30} \times 3^{1/2}$, $2^{30} \times 5^{1/2}$, $2^{30} \times 10^{1/2}$
- Complexité :
  - Trouver 2 messages ayant même signature : 2^{80}
  - Trouver un message pour une signature donnée : 2^{160}

L'algorithme SHA-1



Traitement d'un bloc par SHA-1

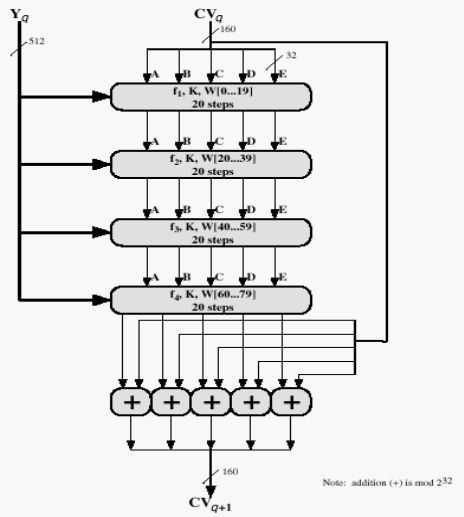


Figure 3.5 SHA-1 Processing of a Single 512-bit Block

15

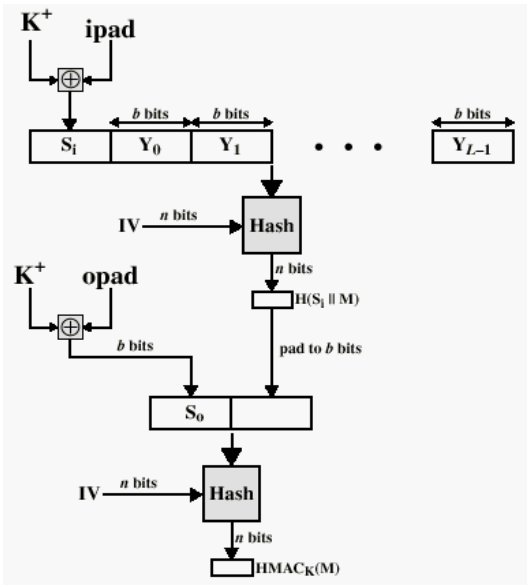
Fonctions de hachage sécurisées

	SHA-1	MD5 (Rfc 1321) "Ron Rivest"	RIPEMD-160
Digest length	160 bits	128 bits	160 bits
Basic unit of processing	512 bits	512 bits	512 bits
Number of steps	80 (4 rounds of 20)	64 (4 rounds of 16)	160 (5 paired rounds of 16)
Maximum message size	2^{64}-1 bits	∞	∞

HMAC

- Proposition d'utiliser sous Internet la technique HMAC pour n'importe quelle fonction de hachage sécurisée définie pour la technique MAC (par ex. SHA-1)
- Normalisé par rfc 2104, obligatoire pour IPsec, utilisé par TLS ou SET
- Clef secrète K : complétée à b bits si nécessaire par des 0 => K+
- Ipad/Opad : suite de 0x36 ou 0x5c

HMAC Structure



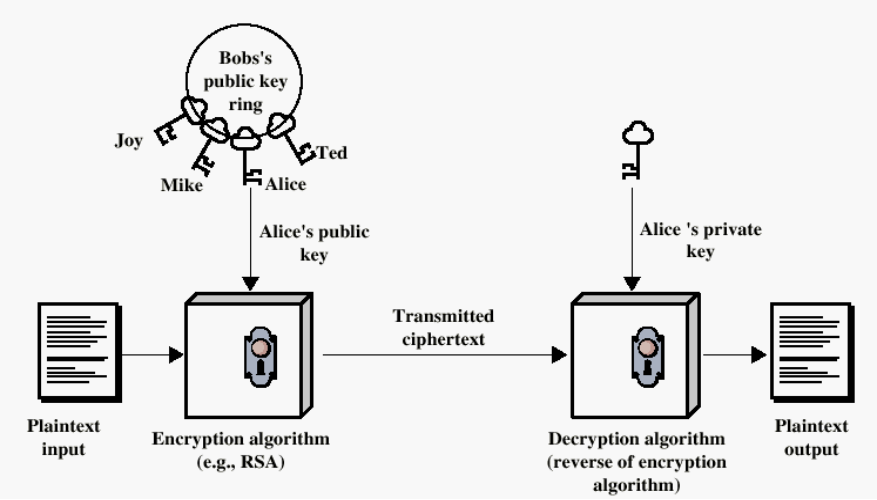
Formule HMAC

- $HMAC_K(M) = H[K^+ + opad) \,||\, H[K^+ + ipad) \,||\, M]]$

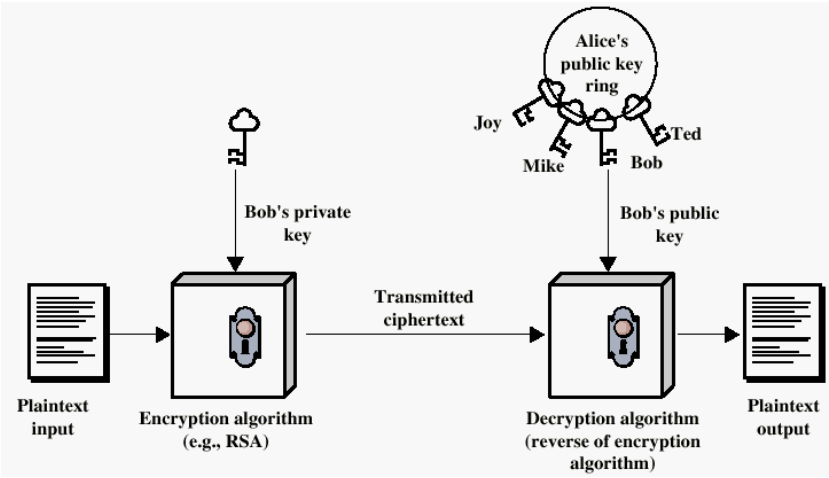
Chiffrement asymétrique Principes

- Utilise un couple de clefs : (clef privée, clef publique)
  - A révolutionné le chiffrement (1977)
  - À des conséquences sur : distribution des clefs, confidentialité et l'authentification.
- Manipule 6 objets (cf. transparent suivant) :
  - Message initial
  - Algorithme de chiffrement
  - Clef publique
  - Clef privée
  - Message chiffré
  - Algorithme de déchiffrement

Chiffrement utilisant un chiffrement asymétrique



Authentification utilisant un chiffrement asymétrique



Utilisations d'un chiffrement asymétrique

- 3 utilisations:
  - **Confidentialité :**
    - L'émetteur chiffre le message avec la clef public du récepteur.
  - **"Digital signature" :**
    - Authentification de l'émetteur et intégrité du message
    - L'émetteur signe le message avec sa clef privée.
  - **"Public Key Infrastructure" :**
    - L'utilisation d'un chiffrement asymétrique facilite le déploiement d'un système sûr de distribution de clefs.

Propriétés d'une fonction assurant un chiffrement asymétrique

1. Il est facile de générer une paire de clef ("public key KU_b", "private key KR_b")
2. Il est facile de chiffrer un message initial (M) connaissant la clef publique :

$$C = E_{KUb}(M)$$

3. Il est facile de déchiffrer le message chiffré (C) connaissant la clef privée :

$$M = D_{KRb}(C) = D_{KRb}[E_{KUb}(M)]$$

Propriétés d'une fonction assurant un chiffrement asymétrique

4. Il est quasi-impossible connaissant une clef de trouver l'autre clef.
5. Il est quasi-impossible connaissant une clef et le message chiffré de retrouver le message initial.
6. Si l'une des 2 clefs est utilisée pour le chiffrement alors l'autre permet le déchiffrement :

$$M = D_{KRb}[E_{KUb}(M)] = D_{KUb}[E_{KRb}(M)]$$

Algorithmes de chiffrement à clefs publiques

- **RSA** – [Ron Rives, Adi Shamir and Len Adleman at MIT, 1977].
  - RSA est un chiffrement par bloc
  - Le plus largement implémenté
  - La difficulté de la cryptanalyse est basée sur celle de la décomposition en nombres premiers
  - Avril 1994 à l'aide de 1600 ordinateurs un message chiffré avec une clef de 428 bits à été décodé après 8 mois d'analyse

Algorithmes de chiffrement à clefs publiques

- Digital Signature Standard (DSS)
  - Utilise SHA-1
  - Dernière version de la norme en 1996 – FIPS PUB 186
  - N'est pas prévu pour être utilisé ni pour le chiffrement ni l'échange de clefs
- Elliptic-Curve Cryptography (ECC)
  - Tente de répondre à l'accroissement de la longueur des clefs des autres méthodes (RSA)
  - Proposé à IEEE P1363
  - Peu sûr, comparé à l'expérience que l'on a sur RSA
  - Mathématiquement très complexe

RSA Algorithm – Key Generation

1.	Select *p,q*	*p* and *q* both prime
2.	Calculate $n = p \times q$	
3.	Calculate $\Phi(n) = (p-1)(q-1)$	
4.	Select integer *e*	$\gcd(\Phi(n), e) = 1; 1 < e < \Phi(n)$
5.	Calculate *d*	$d = e^{-1} \bmod \Phi(n)$
6.	Public Key	KU = {e,n}
7.	Private key	KR = {d,n}

The RSA Algorithm - Encryption

- Plaintext: $M<n$
- Ciphertext: $C = M^e \pmod n$

The RSA Algorithm - Decryption

- Ciphertext: C
- Plaintext: $M = C^d \pmod n$

Example of RSA Algorithm

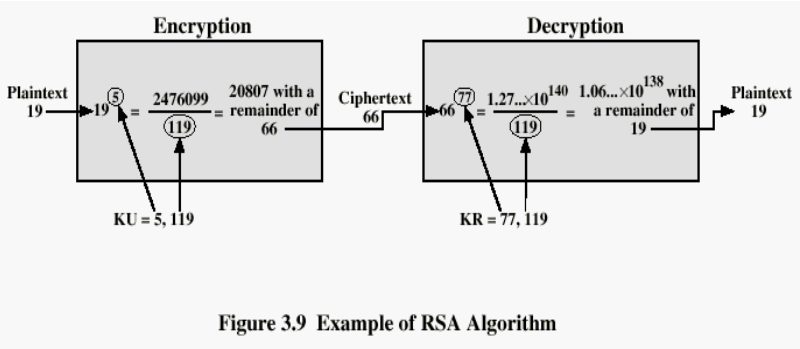


Figure 3.9 Example of RSA Algorithm

Gestion de clefs

- Génération de clefs
- Distribution de clefs

Génération de clefs

- Lorsque le chiffrement est symétrique
  - Génération de la clef partagée sur un partenaire
  - Distribution sûre de la clef vers l'autre partenaire
- Lorsque le chiffrement est asymétrique
  - Génération de la clef privée d'un partenaire sur le partenaire lui-même
  - Utilisation de la clef publique par l'autre partenaire
    - Distribution sûre de clefs publiques
  - Génération de la clef privée d'un partenaire par un système permettant de conserver la clef donc de la restituer au partenaire (s'il la perd par ex.)
    - Distribution sûre de clefs privées

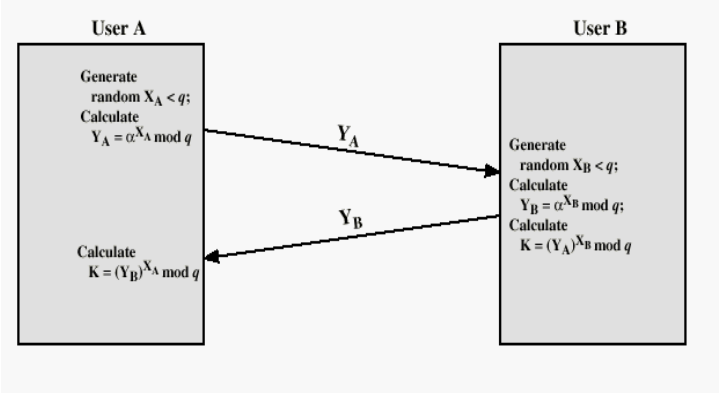
Distribution de clefs

- **Diffie-Hellman** [Diffie, Hellman, 1976]
  - Permet l'échange sécurisé d'une clef (partagée)
  - La difficulté de la cryptanalyse est basée sur celle du calcul de logarithmes discrets
  - Pas d'authentification des partenaires !
- Par un système de distribution des clefs ("Public key Infrastructure")

Diffie-Hellman

- Paramètres :
  - q : un nombre premier
  - <a> une racine primitive de q (qqs b<q, il existe i tq b = $<a>^i$ mod p)
- Produit un secret partagé : K

Diffie-Hellman Key Exchange



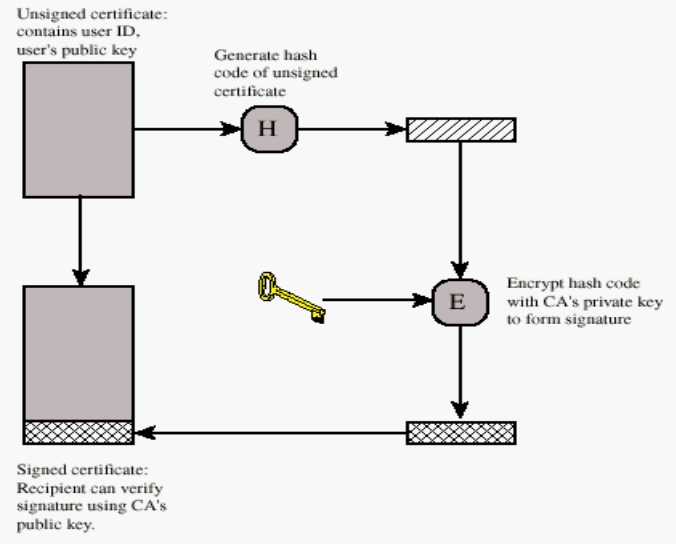
Distribution des clefs

- La distribution de clefs
  - La distribution de clefs publiques
  - La distribution des clefs secrètes (partagées ou privées)
- Utilisation du chiffrement asymétrique facilite la distribution de clefs
- La distribution de clefs publiques utilise la notion de certificats

Distribution des clefs publiques

- Les clefs publiques sont publiques … mais elles doivent être associées de manière sûre avec l'identité de leur propriétaire.
  - Grâce à des certificats… de clefs publiques
- Un tiers ("Certificate Authority" : CA), de confiance, distribue des certificats
  - Hypothèse on connait la clef publique du CA
- Certificat :
  - une clef publique + l'identité du propriétaire de la clef + un signature numérique produite par le CA et prouvant l'intégrité du certificat
  - Au format X.509,
    - utilisé par IPsec, SSL, S/MIME, etc.
- Efficace : la vérification du certificat est locale !

Distribution de clefs publiques



39

Distribution de clefs secrètes

- Diffie-Hellman n'est pas sûr:
  - n'authentifie pas les partenaires
- Utilisation de certificats de clefs publiques
- Utilisation normale du chiffrement asymétrique pour assurer une communication confidentielle
- Processus :
  - Préparation du message contenant la clef secrète
  - Chiffrement du message par la clef publique du destinataire, clef publique obtenue par un CA.
  - Transmission du message chiffré.
  - Déchiffrement du message avec la clef privée, récupération de la clef secrète

Distribution d'une clef de session

- Utilisation d'un chiffrement symétrique
  - Accélère le chiffrement et diminue la taille du message
  - Nécessite la connaissance (la transmission) d'une clef partagée
    - La clef de session
- Processus :
  - Préparation du message
  - Chiffrement symétrique du message avec la clef de session
  - Chiffrement asymétrique de la clef de session par la clef publique du destinataire (clef publique obtenue par un CA)
  - Transmission de la clef de session chiffrée et du message chiffré
- Seul le destinataire est capable de déchiffrer la clef de session et donc de déchiffrer le message