

Sécurité des systèmes informatiques  
**La gestion des clés**

Olivier Markowitch

# Gestion des clés cryptographiques

- Echange de clés symétriques/secrètes  
parfois = clés de session
- Gestion des clés publiques associées à une clé privée

# Les échanges de clés de session

Un **protocole d'établissement de clé** est un protocole au cours duquel une clé secrète devient disponible à deux (ou plus) entités

Un **protocole de transport de clé** est un protocole d'établissement de clé où une partie crée ou obtient la clé secrète et la transmet à l'autre (aux autres) partie(s)

Un **protocole d'accord sur la clé** est un protocole d'établissement de clé au cours duquel la clé secrète est dérivée sur base d'information de deux (ou plus, et idéalement : de chaque) parties, de manière à ce qu'aucune partie ne puisse prédéterminer la valeur de la clé secrète ainsi construite

# Les échanges de clés de session en pratique

- Protocoles de transport de clé basés sur le chiffrement asymétrique :
  - Needham-Schroeder
  - Needham-Schroeder modifié
- Protocoles d'accord sur la clé basés sur les techniques asymétriques :
  - Diffie-Hellman
  - Protocole de station à station

# Les échanges de clés symétriques

Needham-Schroeder :

$$A \rightarrow B : E_{K_B}(k_1, A)$$

$$B \rightarrow A : E_{K_A}(k_1, k_2)$$

$$A \rightarrow B : E_{K_B}(k_2)$$

# Les échanges de clés symétriques

Needham-Schroeder : problème

$$A \rightarrow B : E_{K_B}(k_1, A)$$

$$B \rightarrow C : E_{K_C}(k_1, A)$$

$$C \rightarrow B : E_{K_A}(k_1, k_2)$$

$$B \rightarrow A : E_{K_A}(k_1, k_2)$$

$$A \rightarrow B : E_{K_B}(k_2)$$

$$B \rightarrow C : E_{K_C}(k_2)$$

# Les échanges de clés symétriques

Needham-Schroeder corrigé :

$$A \rightarrow B : E_{K_B}(k_1, A)$$

$$B \rightarrow A : E_{K_A}(B, k_1, k_2)$$

$$A \rightarrow B : E_{K_B}(k_2)$$

# Les échanges de clés symétriques

Needham-Schroeder corrigé :

$$A \rightarrow B : E_{K_B}(k_1, A)$$

$$B \rightarrow C : E_{K_C}(k_1, A)$$

$$C \rightarrow B : E_{K_A}(C, k_1, k_2)$$

$$B \rightarrow A : E_{K_A}(C, k_1, k_2)$$

$$A \rightarrow B : \text{STOP}$$

# Les échanges de clés symétriques

Needham-Schroeder modifié :

$$A \rightarrow B : E_{K_B}(k_1, r_1, A)$$

$$B \rightarrow A : E_{K_A}(k_2, r_1, r_2)$$

$$A \rightarrow B : r_2$$

# Les échanges de clés symétriques

Diffie-Hellman :

Soient  $p$  premier et  $\alpha$  un générateur de  $\mathbb{Z}_p^*$

$A \rightarrow B : \alpha^x \bmod p$  (avec  $x$  valeur aléatoire secrète choisie par Alice)

$B \rightarrow A : \alpha^y \bmod p$  (avec  $y$  valeur aléatoire secrète choisie par Bob)

$$k = (\alpha^x)^y = (\alpha^y)^x = \alpha^{xy} \bmod p$$

# Les échanges de clés symétriques

Diffie-Hellman : homme au milieu

$$A \rightarrow O : \alpha^x \bmod p$$

$$O \rightarrow B : \alpha^{x'} \bmod p$$

$$B \rightarrow O : \alpha^y \bmod p$$

$$O \rightarrow A : \alpha^{y'} \bmod p$$

$$\text{Alice calcule } k_1 = (\alpha^{y'})^x \bmod p$$

$$\text{Bob calcule } k_2 = (\alpha^{x'})^y \bmod p$$

Oscar calcule :

$$k_1 = (\alpha^x)^{y'} \bmod p \text{ et}$$

$$k_2 = (\alpha^y)^{x'} \bmod p$$

# Les échanges de clés symétriques

Protocole de station à station :

Soient  $p$  premier et  $\alpha$  un générateur de  $\mathbb{Z}_p^*$

$$A \rightarrow B : \alpha^x \text{ mod } p$$

$$B \rightarrow A : \alpha^y \text{ mod } p, E_k(\text{Sig}_B(\alpha^x, \alpha^y))$$

$$A \rightarrow B : E_k(\text{Sig}_A(\alpha^x, \alpha^y))$$

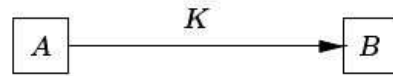
$$\text{Où } k = (\alpha^x)^y = (\alpha^y)^x = \alpha^{xy} \text{ mod } p$$

# Les autorités

Les clés secrètes peuvent être gérées et distribuées par une autorité qui sera :

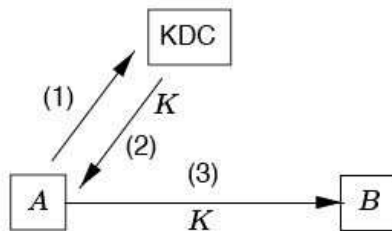
- un **centre de distribution de clés**, si les clés sont générées par l'autorité,
- un **centre de translation de clés**, si chaque clé est générée par un utilisateur et transmise à l'autorité

(a) Point-to-point key distribution

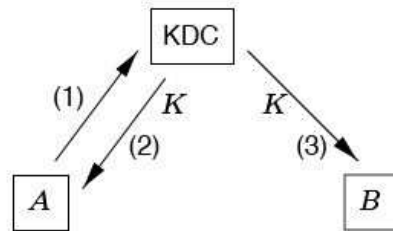


(b) Key distribution center (KDC)

(i)

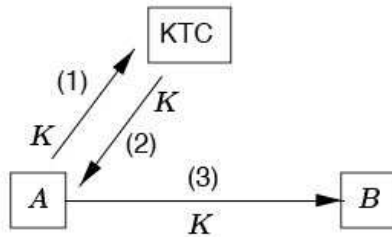


(ii)

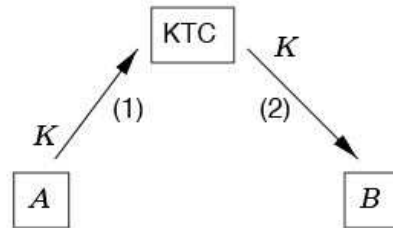


(c) Key translation center (KTC)

(i)



(ii)



**Figure 13.1:** Simple key distribution models (symmetric-key).

# La vie d'une clé (secrète ou publique)

La **crypto-période** d'une clé est la période au cours de laquelle une clé est valide

Cette crypto-période permet de limiter la durée de validité d'une information chiffrée ou encore de limiter l'usage d'une clé, sachant que sa durée de vie dépend des avancées technologiques

Une clé peut être à **court terme** ou à **long terme**

# Les certificats de clés publiques

Un **certificat d'une clé publique** consiste en des données et une signature digitale

Les données contiennent (au-moins) la clé publique et un string identifiant de manière unique, l'entité associée à cette clé publique

La signature digitale est réalisée par une autorité de certification sur les données du certificat

La clé publique de vérification de la signature de l'autorité de certification doit être publiquement connue

# La révocation

Une clé est **compromise** lorsqu'un adversaire possède de l'information sur des données secrètes

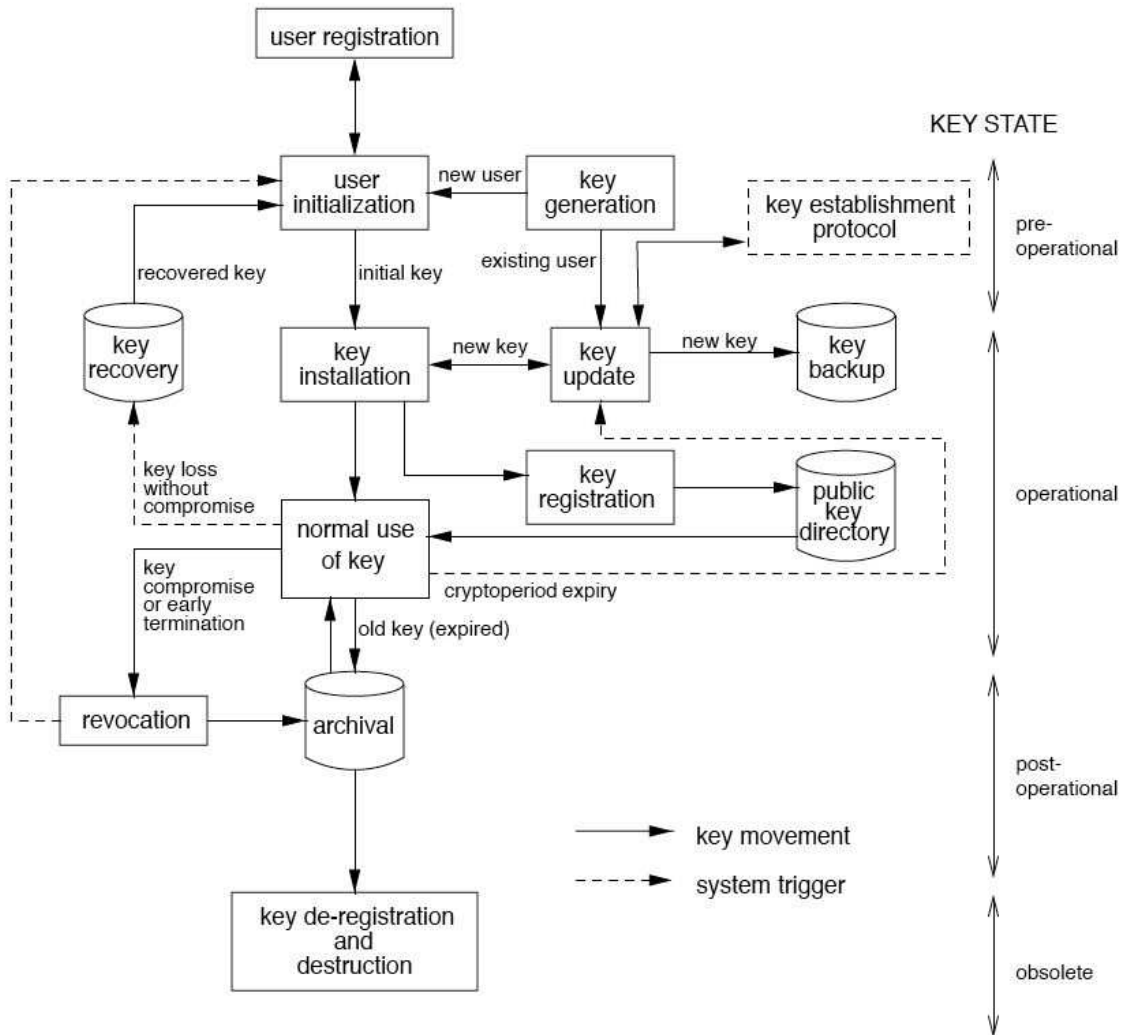
Lorsqu'une clé est compromise, elle doit être révoquée

Les certificats des clés révoquées doivent alors être mises dans une **liste des certificats révoqués** (CRL : *Certificate Revocation List*)

# La fin de vie d'une clé

Lorsqu'une clé arrive en fin de vie (crypto-période), il convient de créer et d'échanger une nouvelle clé pour remplacer l'ancienne

Il est tout à fait déconseillé d'utiliser l'ancienne clé pour transmettre la nouvelle clé confidentiellement



**Figure 13.10:** Key management life cycle.