info602 : Mathématiques pour l'informatique
TD 3 : cryptographie

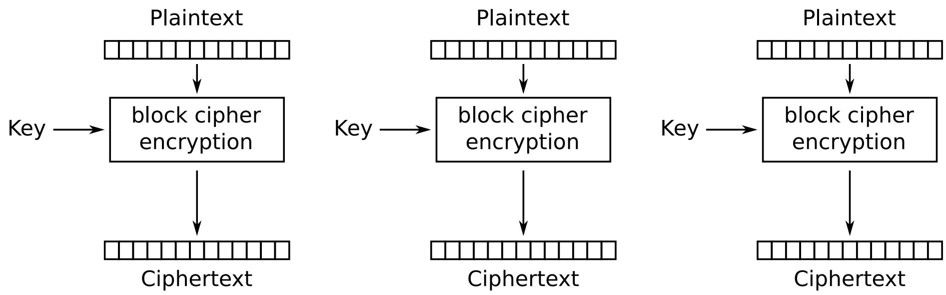
Pierre Hyvernat
Laboratoire de mathématiques de l'université Savoie Mont Blanc
bâtiment Chablais, bureau 17, poste : 94 22
email : `Pierre.Hyvernat@univ-smb.fr`
www : `http://www.lama.univ-smb.fr/~hyvernat/`

Partie 1 : modes de fonctionnement, chiffrement symétrique

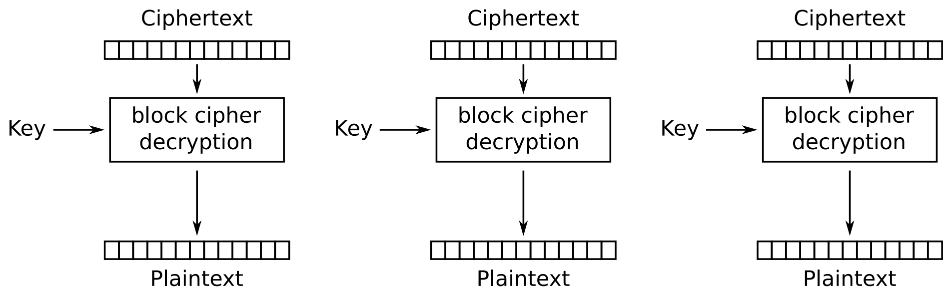
On commence par supposer que l'on utilise un algorithme de chiffrement symétrique comme AES (blocs de 16 octets) ou DES (blocs de 8 octets). La clé secrète est nécessaire pour le chiffrement et le déchiffrement.

Exercice 1 : Mode ECB

Le mode "ECB" (Electronic CodeBook) fonctionne en codant chaque bloc de manière indépendante. On peut représenter ce mode de fonctionnement par le schéma suivant (Wikipédia) :



Le déchiffrement est fait de la même manière :



Formellement, si on note F_k pour la fonction de chiffrement (avec la clé k) et F_k^{-1} pour la fonction de déchiffrement (avec la clé k), on a

- $C_i = F_k(B_i)$,
- $D_i = F_k^{-1}(C_i)$,

où B_i représente le i-ème bloc clair, C_i le i-ème bloc chiffré, et D_i le i-ème bloc déchiffré.

Les schémas montrent clairement que le chiffrement et le déchiffrement peuvent se faire en parallèle.

Question 1. Quel problème voyez vous avec ce mode de fonctionnement ?

Exercice 2 : Mode CBC

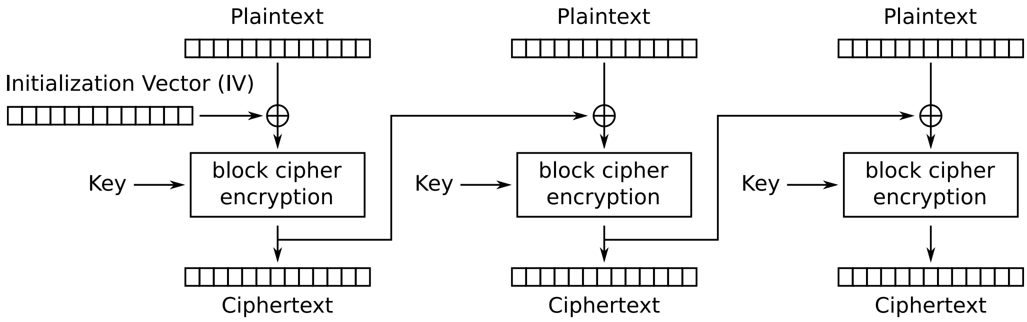
Le chiffrement du mode "CBC" (Cipher Block Chaining) est donné par
- $C_0 = IV$,
- $C_i = F_k(B_i \oplus C_{i-1})$,

et le déchiffrement par
- $D_i = F_k^{-1}(C_i) \oplus C_{i-1}$,

où IV est un bloc aléatoire qui doit être changé à chaque communication.

Question 1. Vérifiez que le déchiffrement fonctionne, c'est à dire que $D_i = B_i$.

Question 2. Le schéma du chiffrement est donné par



Donnez le schéma correspondant pour le déchiffrement.

Question 3. Est-ce que le chiffrement peut être fait en parallèle ?

Est-ce que le déchiffrement peut être fait en parallèle ?

Est-il possible de chiffrer un bloc partiel ? (Il faut pouvoir chiffrer la suite du bloc lorsqu'elle est connue.)

Question 4. Que se passe-t'il si un attaquant modifie un bit dans un bloc chiffré C_k ?

Exercice 3 : mode CTR

Le mode "CTR" (CounTeR) marche de la manière suivante :
- $C_i = F_k(IV \cdot i) \oplus B_i$,

où IV est un "morceau" de bloc aléatoire qui doit changer à chaque communication, et $\cdot$ représente la concaténation. "$IV \cdot i$" est donc un bloc qui se décompose en deux parties : IV (constant pour tous les blocs d'un message) suivi d'un compteur (différent pour tous les blocs d'un message).

Question 1. Pourquoi est-il important que IV soit différent à chaque communication ?

Question 2. Est-ce que cela pose problème si l'attaquant connait IV à l'avance ?

Question 3. Donnez la formule qui permet de calculer D_i à partir de C_i.

À quoi sert l'algorithme de déchiffrement F_k^{-1} ?

Question 4. Dessinez les schémas de chiffrement / déchiffrement correspondants.

Est-ce que le chiffrement peut être fait en parallèle ?

Est-ce que le déchiffrement peut être fait en parallèle ?

Est-il possible de chiffrer un bloc partiel ? (Il faut pouvoir chiffrer la suite du bloc lorsqu'elle est connue.)

Partie 2 : cryptographie sans clé

Exercice 1 : XOR

Question 1. Alice et Bob utilise le protocol suivant pour échanger un message m, sans avoir besoin de clé partagée :

- Alice tire une clé aléatoire K_a de même taille que le message, elle envoie $M_a = m \oplus K_a$ (XOR bit à bit) à Bob ;
- Bob reçoit M_a, il tire une clé K_b de même taille et renvoie $M_b = M_a \oplus K_b$ à Alice ;
- Alice "enlève" sa clé en renvoyant $M_c = M_b \oplus K_a$ à Bob ;
- Bob "enlève" sa clé en calculant $M = M_c \oplus K_b$. Il retrouve ainsi m !

Expliquez pourquoi ce protocol n'est pas fiable en montrant qu'un attaquant peut retrouver le message m à partir de M_a, M_b et M_c.

Exercice 2 : Fermat

Rappels :

- si a et b sont premier entre eux, alors il existe x et y vérifiant $ax + by = 1$. Dans ce cas, on dit que x est l'inverse de a modulo b. (Car $ax = 1 \pmod{b}$.)
- si p est un nombre premier, alors $\mathbf{Z}_p = \{0, 1, \ldots, p-1\}$ avec l'addition et la multiplication modulo p est un *corps*. Chaque nombre non nul a donc un inverse que l'on peut calculer avec l'algorithme d'Euclide pour le pgcd.
- si p est premier et $a \in \mathbf{Z}_p$, alors $a^{p-1} = 1 \pmod{p}$.

Alice et Bob utilisent le protocole suivant pour échanger un message :

- Alice et Bob choisissent un nombre premier p très grand ;
- pour envoyer $m < p-1$, Alice tire un nombre a premier avec $p-1$ et envoie $M_a = m^a \bmod p$ à Bob
- Bob tire un nombre b premier avec $p-1$ et renvoie $M_b = M_a^b \bmod p$ à Alice
- Alice "enlève" sa clé en renvoiyant $M_c = M_b^{a'} \bmod p$ à Bob, où a' est l'inverse de a modulo $p-1$
- Bob "enlève" sa clé en calculant $M = M_c^{b'} \bmod p$ où b' est l'inverse de b modulo $p-1$. Il retrouve ainsi m !

Question 1. En utilisant vos calculatrices / ordinateurs, utilisez les nombres $p = 1367$ $a = 129$ et $b = 1201$ pour transmettre $M = 666$ pour tester le protocole.

Commencez par vérifier que l'inverse de 129 modulo 1366 est 593, et que celui de 1201 est 505.

Question 2. Prouvez que le protocole est correct.

Question 3. Pourquoi ce protocole n'est il pas utilisé ?

Partie 3 : Échange de clés de Diffie-Hellman

Méthode :

- Alice et Bob choisissent un (grand) nombre premier p et un nombre g ;
- Alice choisit un nombre aléatoire a dans $\mathbf{Z}_p$ et envoie $g^a \bmod p$ à Bob ; Bob fait la même chose et envoie $g^b \bmod p$ à Alice ;
- Alice reçoit B et calcule $B^a \bmod p$ et Bob reçoit A et calcule $A^b \bmod p$;
- ils tombent sur le même résultat.

Question 1. Rappelez pourquoi Bob et Alice trouvent toujours le même résultat.

Question 2. Que se passe-t'il si le canal de communication est compromis et qu'un observateur malveillant (Eve) écoute les communications ?

Comment est-ce que Eve pourrait trouver le résultat partagé par Alice et Bob ?

Pourquoi n'est-ce pas raisonnable ?

Question 3. Faites tourner l'algorithme d'échange de clé de Diffie-Hellman avec les valeurs suivantes
- $p = 11$ comme nombre premier
- $g = 2$ comme générateur de $\mathbf{Z}/p\mathbf{Z}$
- $a = 4$ comme nombre secret choisi par Alice
- $b = 8$ comme nombre secret pour Bob

Question 4. Pour garantir qu'une recherche exhaustive est impossible, il faut que g^n prennent le plus de valeurs possible. On dit que g est *générateur* si $g^n \bmod p$ prend toutes les valeurs entre 0 et $p - 1$.

Vérifiez que 2 est générateur pour $p = 19$. Est-ce que 7 est générateur ? Quel problème cela peut-il poser ?

Question 5. Pouvez-vous généraliser le protocole d'échange pour partager une clé entre trois personnes ? Entre quatre ?

Partie 4 : chiffrement asymétrique

Exercice 1 : système Elgamal

Méthode : p est un nombre premier et g est un générateur du groupe $\mathbf{Z}_p$;
- Bob choisit un nombre b secret et publie sa clé $K_B = g^b \bmod p$
- pour envoyer M, Alice choisi un nombre k secret et envoie $(g^k, K_B^k * M \bmod p)$ à Bob
- à la réception de (C_1, C_2), Bob calcule C_2/C_1^b et obtient M.

Question 1. Justifiez le système en montrant que Bob récupère bien le message d'Alice.

Question 2. En prenant $p = 13$ et $g = 2$, faites les calculs et vérifications suivantes
- g est un élément générateur de $\mathbf{Z}_p$
- quelle est la clé publique de Bob si sa clé privée est $b = 9$?
- comment Alice code-t'elle le message 10 si elle choisit une clé temporaire $k = 6$?
- comment Bob décode-t'il le message ? Est-ce que ça a marché ?

Question 3. Que se passe-t'il si on utilise un nombre g qui n'est pas générateur ?

Question 4. Que se passe-t'il si on utilise un nombre p non premier ?

Question 5. Supposons qu'Alice utilise tout le temps la même clé k pour coder son message. Un observateur malveillant Eve peut alors obtenir des informations précieuses... Si Alice encode M_1 et M_2 avec k et Eve parvient à écouter les communications, elle pourra connaître la valeur de M_1/M_2. Comment ?

Comment est-ce que Eve peut mettre cette connaissance à profit ?