Cryptographie à clé publique : la méthode du sac-à-dos

Malika More
(malika.more@iut.u-clermont1.fr)
Jean Mailfert - Gisèle Provost

1A - IUT Info - Clermont 1

Arithmétique et Cryptographie

Année 2011-2012

Important

- Les transparents du cours et d'autres documents et informations sont disponibles sur la page du cours sur l'ENT
- Il est très fortement recommandé d'apporter une calculatrice en cours et en TD d'arithmétique

Plan du cours

Cryptographie moderne

- La cryptographie entre dans son ère moderne avec l'utilisation intensive des ordinateurs, c'est-à-dire à partir des années 1970.
- Dans la cryptographie moderne, on utilise des problèmes mathématiques que l'on ne sait pas résoudre rapidement, par exemple la factorisation de grands nombres (méthode RSA) ou le problème général du sac-à-dos (méthode du sac-à-dos).
- Plus anecdotique, on voit aussi apparaître deux personnages récurrents célèbres : Alice et Bob.
- Mais surtout, une famille de méthodes de chiffrement radicalement nouvelles, appelées *systèmes à clé publique*, a été inventée.

Le problème des clés

- Depuis les origines de la cryptographie, et jusqu'à récemment, tous les cryptosystèmes étaient basés sur une même notion fondamentale : chaque correspondant était en possession de deux clés secrètes, l'une utilisée pour chiffrer et l'autre pour déchiffrer, qui jouaient des rôles symétriques.
- Cela a un inconvénient majeur : comment communiquer ces clés au correspondant ? Il faut pouvoir utiliser un canal sûr, par exemple une valise diplomatique ou un émissaire de confiance.
- De toute manière, il faut un contact préalable avec la personne qui devra (dé)chiffrer nos messages.

Les systèmes à clés publiques

- C'est en 1976 que Diffie et Hellman ont imaginé un concept totalement inédit : une clé publique pour chiffrer et une clé privée pour déchiffrer, dont l'utilisation est totalement disymétrique.
- Bien entendu, il existe un lien mathématique entre ces deux clés, mais ce lien est constitué par ce que les deux inventeurs appellent une « fonction trappe à sens unique ».
- Cette fonction permet de calculer aisément la clé de chiffrement en connaissant la clé de déchiffrement.
- En revanche, l'opération inverse est pratiquement impossible.

Le problème des clés est réglé

- L'intérêt de ce système est considérable.
- En effet, toute personne ou toute entreprise disposant de moyens informatiques peut élaborer sa clé de déchiffrement - qu'elle garde secrète pour son usage exclusif - puis en déduire la clé de chiffrement correspondante.
- Tous les utilisateurs de ce système agissant de même, les clés de chiffrement peuvent ensuite être groupées dans une sorte d'annuaire mis à la disposition du public.
- Ainsi deux correspondants peuvent-ils communiquer secrètement sans aucun contact préalable.
- Les systémes à clés publiques les plus utilisés dans le monde sont RSA et PGP, mais il en existe d'autres.

La méthode du sac-à-dos

- Dans ce chapitre, on présente le premier chiffre à clé publique, qui a été proposé par Ralph Merkle et Martin Hellman en 1978.
- Il est basé sur le problème du sac-à-dos (Knapsack problem en anglais).
- Il n'est plus utilisé actuellement puisque ce chiffre, ainsi que de nombreuses variantes, a été cassé au début des années 1980 par Adi Shamir.

Un sac-à-dos numérique

- Étant donnés n entiers naturels $A_1, A_2, \ldots, A_n$ et un but C, il s'agit de trouver (si possible) des coefficients $m_1, m_2, \ldots, m_n$ valant 0 ou 1, de telle sorte que

$$C = m_1 A_1 + m_2 A_2 + \ldots + m_n A_n$$

- Le nom de *problème du sac-à-dos* vient du fait qu'on peut imaginer qu'il s'agit de mettre (ou pas) des objets de poids respectifs $A_1, \ldots, A_n$ dans un sac-à-dos de manière à obtenir un poids total valant exactement C.

Suite supercroissante

- On dit que la suite des poids $\mathcal{A} = (A_i)_{1 \leq i \leq n}$ est *supercroissante* lorsque chaque poids est strictement supérieur à la somme de tous les précédents, c.-à-d. :
  $A_1 < A_2$,
  $A_1 + A_2 < A_3$,
  $\ldots$,
  $A_1 + A_2 + \ldots + A_{n-1} < A_n$.

Algorithme glouton

- Lorsque la suite des poids est supercroissante, il existe une méthode simple permettant de trouver l'unique solution $M = (m_1, \ldots, m_n)$ si elle existe :

```
Pour i = n à 1
faire
        Si C ≥ A_i alors
                C = C − A_i
                m_i = 1
        sinon
                m_i = 0
finfaire
Si C = 0 alors
        M = (m_1, ..., m_n) est l'unique solution
sinon
        il n'y a pas de solution
```

Sac-à-dos facile ou difficile ?

- Au contraire, si la suite des poids n'est pas supercroissante, le seul algorithme connu consiste à essayer successivement toutes les solutions $M = (m_1, m_2, ..., m_n)$ possibles. Si la suite est suffisamment longue, c'est un algorithme impraticable.
- Par exemple, pour $n = 150$, on aurait 2^{150} solutions à essayer, ce qui est impraticable même avec les ordinateurs actuels. À titre de comparaison, on estime que le nombre d'atomes de la Terre est d'environ $10^{50} \approx 2^{150}$.
- Par abus de langage, on appelle une suite de poids supercroissante un *sac-à-dos facile* et une suite de poids non supercroissante un *sac-à-dos difficile*.

À vous de jouer

On considère le sac-à-dos facile
$A_1 = 2, A_2 = 3, A_3 = 6, A_4 = 12$.

1. En utilisant l'algorithme glouton, vérifiez que, pour le but $C = 15$, on obtient la solution
   $m_1 = 0, m_2 = 1, m_3 = 0, m_4 = 1$.
2. Montrer que le but $C = 10$ n'est pas accessible.
3. Donner la liste de tous les buts accessibles à l'aide de ce sac-à-dos.

À vous de jouer

On considère le sac-à-dos difficile
$A_1 = 111, A_2 = 124, A_3 = 175, A_4 = 184, A_5 = 198$.

1. Vérifier que le but $C = 497$ est obtenu par la suite de coefficients $m_1 = 0, m_2 = 1, m_3 = 1, m_4 = 0, m_5 = 1$.
2. Quel est le résultat de l'algorithme glouton pour le même but $C = 497$?
3. Le but $C = 471$ est-il accessible ? Et le but $C = 470$?

À vous de jouer

Les sacs-à-dos ci-dessous sont-ils faciles ? Justifier.

1. $\mathcal{A}_1 = (6, 14, 29, 50, 108)$
2. $\mathcal{A}_2 = (537, 923, 942, 1074, 1277, 1416, 3482, 3531)$
3. $\mathcal{A}_3 = (13, 32, 60, 112, 227, 454, 911, 1817)$
4. $\mathcal{A}_4 = (17, 27, 63, 113, 237, 451)$

Exercice 1

1. Le sac-à-dos $\mathcal{A} = (4, 5, 11, 27, 56, 111, 222, 443)$ est-il facile ?
2. Peut-on atteindre le but $C = 309$? Si oui, donner la solution correspondante.
3. Même question avec le but $C = 282$.

Exercice 2

1. Le sac-à-dos $\mathcal{A} = (381, 389, 463, 547, 594, 757, 778, 842)$ est-il facile ?
2. Peut-on atteindre le but $C = 2300$? Si oui, donner une solution correspondante (dans le cas d'un sac-à-dos difficile, il peut y avoir plusieurs solutions différentes permettant d'atteindre le même but).

Exercice 3

Justifier les noms de *sac-à-dos facile* pour une suite de poids supercroissante et de *sac-à-dos difficile* pour une suite de poids non supercroissante.

Principe

- C'est un chiffre à clé publique, basé sur la difficulté de résoudre le problème du sac-à-dos avec une suite de poids non supercroissante.
  - Les messages correspondent aux solutions $M = (m_1, \ldots, m_n)$ du problème du sac-à-dos, c'est pourquoi ils sont écrits en *binaire*.
  - On peut « cacher » un message binaire $M = (m_1, \ldots, m_n)$ à l'aide d'un sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ en calculant $C = m_1 B_1 + m_2 B_2 + \ldots + m_n B_n$. On sait en effet que si on connaît seulement C et $\mathcal{B}$, il est très difficile de retrouver M.
  - Inversement, lorsqu'on connaît $C' = m_1 A_1 + m_2 A_2 + \ldots + m_n A_n$ pour un sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$, on peut facilement trouver $M = (m_1, \ldots, m_n)$ à l'aide de l'algorithme glouton.
  - On dit que C (resp. C') est la *représentation* de M pour le sac-à-dos $\mathcal{B}$ (resp. $\mathcal{A}$).

Principe

- C'est un chiffre à clé publique, basé sur la difficulté de résoudre le problème du sac-à-dos avec une suite de poids non supercroissante.
  - Les messages correspondent aux solutions $M = (m_1, \ldots, m_n)$ du problème du sac-à-dos, c'est pourquoi ils sont écrits en *binaire*.
  - On peut « cacher » un message binaire $M = (m_1, \ldots, m_n)$ à l'aide d'un sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ en calculant $C = m_1 B_1 + m_2 B_2 + \ldots + m_n B_n$. On sait en effet que si on connaît seulement C et $\mathcal{B}$, il est très difficile de retrouver M.
  - Inversement, lorsqu'on connaît $C' = m_1 A_1 + m_2 A_2 + \ldots + m_n A_n$ pour un sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$, on peut facilement trouver $M = (m_1, \ldots, m_n)$ à l'aide de l'algorithme glouton.
  - On dit que C (resp. C') est la *représentation* de M pour le sac-à-dos $\mathcal{B}$ (resp. $\mathcal{A}$).

Principe

- C'est un chiffre à clé publique, basé sur la difficulté de résoudre le problème du sac-à-dos avec une suite de poids non supercroissante.
  - Les messages correspondent aux solutions $M = (m_1, \ldots, m_n)$ du problème du sac-à-dos, c'est pourquoi ils sont écrits en *binaire*.
  - On peut « cacher » un message binaire $M = (m_1, \ldots, m_n)$ à l'aide d'un sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ en calculant $C = m_1 B_1 + m_2 B_2 + \ldots + m_n B_n$. On sait en effet que si on connaît seulement C et $\mathcal{B}$, il est très difficile de retrouver M.
  - Inversement, lorsqu'on connaît $C' = m_1 A_1 + m_2 A_2 + \ldots + m_n A_n$ pour un sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$, on peut facilement trouver $M = (m_1, \ldots, m_n)$ à l'aide de l'algorithme glouton.
  - On dit que C (resp. C') est la *représentation* de M pour le sac-à-dos $\mathcal{B}$ (resp. $\mathcal{A}$).

Principe

- C'est un chiffre à clé publique, basé sur la difficulté de résoudre le problème du sac-à-dos avec une suite de poids non supercroissante.
  - Les messages correspondent aux solutions $M = (m_1, \ldots, m_n)$ du problème du sac-à-dos, c'est pourquoi ils sont écrits en *binaire*.
  - On peut « cacher » un message binaire $M = (m_1, \ldots, m_n)$ à l'aide d'un sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ en calculant $C = m_1 B_1 + m_2 B_2 + \ldots + m_n B_n$. On sait en effet que si on connaît seulement C et $\mathcal{B}$, il est très difficile de retrouver M.
  - Inversement, lorsqu'on connaît $C' = m_1 A_1 + m_2 A_2 + \ldots + m_n A_n$ pour un sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$, on peut facilement trouver $M = (m_1, \ldots, m_n)$ à l'aide de l'algorithme glouton.
  - On dit que C (resp. C') est la *représentation* de M pour le sac-à-dos $\mathcal{B}$ (resp. $\mathcal{A}$).

Principe

- C'est un chiffre à clé publique, basé sur la difficulté de résoudre le problème du sac-à-dos avec une suite de poids non supercroissante.
  - Les messages correspondent aux solutions $M = (m_1, \ldots, m_n)$ du problème du sac-à-dos, c'est pourquoi ils sont écrits en *binaire*.
  - On peut « cacher » un message binaire $M = (m_1, \ldots, m_n)$ à l'aide d'un sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ en calculant $C = m_1 B_1 + m_2 B_2 + \ldots + m_n B_n$. On sait en effet que si on connaît seulement C et $\mathcal{B}$, il est très difficile de retrouver M.
  - Inversement, lorsqu'on connaît $C' = m_1 A_1 + m_2 A_2 + \ldots + m_n A_n$ pour un sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$, on peut facilement trouver $M = (m_1, \ldots, m_n)$ à l'aide de l'algorithme glouton.
  - On dit que C (resp. C') est la *représentation* de M pour le sac-à-dos $\mathcal{B}$ (resp. $\mathcal{A}$).

Un couple de sacs-à-dos particuliers

- Pour chiffrer et déchiffrer des messages binaires, on pourrait donc utiliser un couple de sacs-à-dos $\mathcal{A}$ et $\mathcal{B}$, l'un facile et l'autre difficile, à condition de disposer d'une méthode permettant de passer de la représentation C d'un message binaire M pour le sac-à-dos difficile à la représentation C' du même message M pour le sac-à-dos facile.
- On explique ci-dessous comment fabriquer un couple de sacs-à-dos $\mathcal{A}$ et $\mathcal{B}$ possédant cette propriété.

Principe de fabrication

- On part de l'observation suivante : le problème du sac-à-dos est *homogène*, autrement dit, on ne change pas les solutions (c.-à-d. les messages) en multipliant (ou en divisant) tous les poids A_i et le but C par un même entier E.

Exemple

Pour $\mathcal{A} = (3, 5, 10)$, le but $C = 8$ correspond au message $M = (1, 1, 0)$. Si on multiplie par $E = 7$, on obtient $\mathcal{A}' = (21, 35, 70)$ et $C' = 56$, mais le message $M = (1, 1, 0)$ est inchangé.

Principe de fabrication

- De plus, les multiplications peuvent être effectuées modulo un entier N sans changer les messages.

Exemple

En calculant modulo $N = 13$, on obtient la suite des poids $\mathcal{B} = (8, 9, 5)$ et le but $C'' = 4$, mais le message $M = (1, 1, 0)$ est toujours inchangé, puisque $8 + 9 \equiv 4 \mod 13$.

Principe de fabrication

- Le point crucial est le fait que, même si le sac-à-dos initial $\mathcal{A} = (3, 5, 10)$ est facile, le nouveau sac-à-dos $\mathcal{B} = (8, 9, 5)$ ainsi obtenu sera en général difficile. Pour des raisons évidentes, on appellera *N* le *module* et *E* le *compliqueur*.

Principe de fonctionnement

- Inversement, le passage de la représentation C'' de M pour le sac-à-dos difficile $\mathcal{B}$ à la représentation C de M pour le sac-à-dos facile $\mathcal{A}$ se fait en « divisant » C'' par E modulo N, c'est-à-dire en *multipliant par l'inverse* D de E modulo N.

Exemple

On a $7^{-1} \equiv 2 \mod 13$, puisque $7 \times 2 - 13 \times 1 = 1$, donc l'inverse de $E = 7$ modulo $N = 13$ est $D = 2$. Par conséquent, on calcule $C'' \times D \equiv 4 \times 2 \equiv 8 \mod 13$, et on retrouve bien $C = 8$. On appelle D le *faciliteur*.

Principe de fonctionnement

- *Important* : En pratique, pour que tout marche bien, on choisit comme ici le module N supérieur à la somme des poids du sac-à-dos facile $\mathcal{A}$ et le compliqueur E inversible modulo N (évidemment !).

La méthode du sac-à-dos

- En résumé, tout est prêt maintenant pour utiliser la méthode du sac-à-dos - ou chiffre de Merkle-Hellman - pour échanger des messages secrets :
  - La clé (publique) de chiffrement est constituée du sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ et du module N.
  - La clé (privée) de déchiffrement est constituée du faciliteur D et du sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$.
  - Pour chiffrer un message de n bits $M = (m_1, \ldots, m_n)$, on calcule le cryptogramme $C \equiv m_1 B_1 + m_2 B_2 + \ldots + m_n B_n \mod N$.
  - Pour déchiffrer un cryptogramme C, on calcule $C' \equiv DC \mod N$, puis on retrouve les bits de M en utilisant le sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$ sur C' à l'aide de l'algorithme glouton.
  - Il est important de noter que les deux clés du chiffre de Merkle-Hellman ne peuvent pas être permutées.

La méthode du sac-à-dos

- En résumé, tout est prêt maintenant pour utiliser la méthode du sac-à-dos - ou chiffre de Merkle-Hellman - pour échanger des messages secrets :
  - La clé (publique) de chiffrement est constituée du sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ et du module N.
  - La clé (privée) de déchiffrement est constituée du faciliteur D et du sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$.
  - Pour chiffrer un message de n bits $M = (m_1, \ldots, m_n)$, on calcule le cryptogramme $C \equiv m_1 B_1 + m_2 B_2 + \ldots + m_n B_n \mod N$.
  - Pour déchiffrer un cryptogramme C, on calcule $C' \equiv DC \mod N$, puis on retrouve les bits de M en utilisant le sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$ sur C' à l'aide de l'algorithme glouton.
  - Il est important de noter que les deux clés du chiffre de Merkle-Hellman ne peuvent pas être permutées.

La méthode du sac-à-dos

- En résumé, tout est prêt maintenant pour utiliser la méthode du sac-à-dos - ou chiffre de Merkle-Hellman - pour échanger des messages secrets :
  - La clé (publique) de chiffrement est constituée du sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ et du module N.
  - La clé (privée) de déchiffrement est constituée du faciliteur D et du sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$.
  - Pour chiffrer un message de n bits $M = (m_1, \ldots, m_n)$, on calcule le cryptogramme $C \equiv m_1 B_1 + m_2 B_2 + \ldots + m_n B_n \mod N$.
  - Pour déchiffrer un cryptogramme C, on calcule $C' \equiv DC \mod N$, puis on retrouve les bits de M en utilisant le sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$ sur C' à l'aide de l'algorithme glouton.
  - Il est important de noter que les deux clés du chiffre de Merkle-Hellman ne peuvent pas être permutées.

La méthode du sac-à-dos

- En résumé, tout est prêt maintenant pour utiliser la méthode du sac-à-dos - ou chiffre de Merkle-Hellman - pour échanger des messages secrets :
  - La clé (publique) de chiffrement est constituée du sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ et du module N.
  - La clé (privée) de déchiffrement est constituée du faciliteur D et du sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$.
  - Pour chiffrer un message de n bits $M = (m_1, \ldots, m_n)$, on calcule le cryptogramme $C \equiv m_1 B_1 + m_2 B_2 + \ldots + m_n B_n \mod N$.
  - Pour déchiffrer un cryptogramme C, on calcule $C' \equiv DC \mod N$, puis on retrouve les bits de M en utilisant le sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$ sur C' à l'aide de l'algorithme glouton.
  - Il est important de noter que les deux clés du chiffre de Merkle-Hellman ne peuvent pas être permutées.

La méthode du sac-à-dos

- En résumé, tout est prêt maintenant pour utiliser la méthode du sac-à-dos - ou chiffre de Merkle-Hellman - pour échanger des messages secrets :
  - La clé (publique) de chiffrement est constituée du sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ et du module N.
  - La clé (privée) de déchiffrement est constituée du faciliteur D et du sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$.
  - Pour chiffrer un message de n bits $M = (m_1, \ldots, m_n)$, on calcule le cryptogramme $C \equiv m_1 B_1 + m_2 B_2 + \ldots + m_n B_n \mod N$.
  - Pour déchiffrer un cryptogramme C, on calcule $C' \equiv DC \mod N$, puis on retrouve les bits de M en utilisant le sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$ sur C' à l'aide de l'algorithme glouton.
  - Il est important de noter que les deux clés du chiffre de Merkle-Hellman ne peuvent pas être permutées.

La méthode du sac-à-dos

- En résumé, tout est prêt maintenant pour utiliser la méthode du sac-à-dos - ou chiffre de Merkle-Hellman - pour échanger des messages secrets :
  - La clé (publique) de chiffrement est constituée du sac-à-dos difficile $\mathcal{B} = (B_1, \ldots, B_n)$ et du module N.
  - La clé (privée) de déchiffrement est constituée du faciliteur D et du sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$.
  - Pour chiffrer un message de n bits $M = (m_1, \ldots, m_n)$, on calcule le cryptogramme $C \equiv m_1 B_1 + m_2 B_2 + \ldots + m_n B_n \mod N$.
  - Pour déchiffrer un cryptogramme C, on calcule $C' \equiv DC \mod N$, puis on retrouve les bits de M en utilisant le sac-à-dos facile $\mathcal{A} = (A_1, \ldots, A_n)$ sur C' à l'aide de l'algorithme glouton.
  - Il est important de noter que les deux clés du chiffre de Merkle-Hellman ne peuvent pas être permutées.

Exemple

1. La clé publique d'Alice est $\mathcal{B} = (8, 9, 5)$ et $N = 13$.
2. La clé privée d'Alice est $\mathcal{A} = (3, 5, 10)$ et $D = 2$.
3. Bob souhaite envoyer à Alice le message binaire $M = (1, 1, 0)$. Il chiffre son message à l'aide de la clé publique d'Alice en calculant $C = 1 \times 8 + 1 \times 9 + 0 \times 5 \equiv 17 \equiv 4 \mod 13$. Il envoie donc le cryptogramme $C = 4$ à Alice.
4. Pour déchiffrer le cryptogramme $C = 4$, Alice utilise sa clé privée pour calculer $C' \equiv 2 \times 4 \equiv 8 \mod N$. Puis elle utilise l'agorithme glouton :
   1. $C' = 8 < 10 = A_3$ donc $b_3 = 0$ et C' reste inchangé ;
   2. $C' = 8 \geq 5 = A_2$ donc $b_2 = 1$ et C' devient $C' = 8 - 5 = 3$;
   3. $C' = 3 \geq 3 = A_1$, donc $b_1 = 1$ et C' devient $C' = 3 - 3 = 0$
   4. On a terminé avec $C' = 0$, donc il y a une solution, et c'est $M' = (1, 1, 0)$.

   On peut vérifier que le message reçu par Alice est bien le même que le message envoyé par Bob.

À vous de jouer

1. Pour fabriquer son chiffre “sac-à-dos”, Alice choisit le sac-à-dos facile $SADF = (2, 4, 12, 21)$. Vérifiez qu’il s’agit bien d’un sac-à-dos facile.
2. Puis elle choisit le module $N = 67$. Pourquoi est-il acceptable ?
3. Ensuite, elle choisit le compliqueur $E = 19$. Pourquoi est-il acceptable ?
4. Calculez le sac-à-dos difficile $SADD$ d’Alice.
5. Calculez le faciliteur D correspondant au compliqueur E.
6. Alice va maintenant publier sa clé publique. Indiquez de quoi est constituée cette clé publique.

À vous de jouer

Le sac-à-dos facile d'Alice est $SADF = (2, 7, 16, 29, 59)$, son module est $N = 133$ et son faciliteur est $D = 53$. Alice reçoit le cryptogramme $C = 82$. Quel est le message binaire M qu'elle déchiffre ?

À vous de jouer

Le sac-à-dos difficile d'Alice est $SADD = (95, 45, 168, 85, 170)$ et son module est $N = 184$. Bob souhaite envoyer à Alice le message binaire $M = (0, 1, 0, 0, 1)$. Calculer le cryptogramme correspondant.

Exercice 4

Le chiffre “sac-à-dos” d’Alice est constitué du sac-à-dos facile $SADF_A = (7, 9, 24, 41, 83, 168)$, du sac-à-dos difficile $SADD_A = (310, 198, 60, 161, 266, 69)$, du module $N_A = 351$ et du faciliteur $D_A = 94$.
Le chiffre “sac-à-dos” de Bob est constitué du sac-à-dos facile $SADF_B = (8, 16, 27, 59, 113, 225)$, du sac-à-dos difficile $SADD_B = (326, 190, 3, 383, 389, 333)$, du module $N_B = 462$ et du faciliteur $D_B = 163$.

1. Alice veut envoyer à Bob le message binaire $M = (0, 1, 1, 0, 1, 0)$. Quels calculs Alice doit-elle effectuer pour chiffrer le message M ?
2. Bob reçoit d’Alice le cryptogramme $C = 118$. Quels calculs Bob doit-il effectuer pour déchiffrer le cryptogramme C ?

Exercice 5

Pour fabriquer un chiffre sac-à-dos, on a choisi le module $N = 168$ et le sac-à-dos facile $SADF = [4; 7; 15; 32; 71]$.

1. Vérifier que ces valeurs sont acceptables.
2. On choisit le compliqueur $E = 47$. Vérifier qu'il est acceptable, puis calculer le sac-à-dos difficile $SADD$ et le faciliteur D.
3. Indiquer quels éléments constituent la partie publique et quels éléments constituent la partie privée de ce chiffre sac-à-dos. Que faut-il faire des éléments restants ? Pourquoi ?

Exercice 6

Le code secret “sac-à-dos” d’Alice est constitué du sac-à-dos facile $SADF_A = [3, 5, 9, 20]$, du sac-à-dos difficile $SADD_A = [51, 29, 41, 4]$, du module $M_A = 56$ et du faciliteur $d_A = 33$.

Le code secret “sac-à-dos” de Bob est constitué du sac-à-dos facile $SADF_B = [2, 4, 8, 19]$, du sac-à-dos difficile $SADD_B = [26, 52, 49, 27]$, du module $M_B = 55$ et du faciliteur $d_B = 17$.

1. Bob reçoit le message suivant : $[44, 20, 23]$. Déchiffrez ce message, c’est-à-dire déterminez les trois quartets (mots de 4 bits) correspondants. Le bit de poids fort du message correspond au premier élément du sac-à-dos. Écrivez le résultat sous forme de trois chiffres hexadécimaux.
2. Bob veut envoyer à Alice le message binaire $mess = 10101110010011110100$.
   Calculer le message crypté *crypt*. Pour chaque bloc, le bit de poids fort du message correspond au premier élément du sac-à-dos.

Exercice 7

Bob a publié le sac-à-dos difficile
$SADD = (1531, 1481, 1261, 813, 1636)$ et le module $N = 1677$.
Alice a publié le cryptogramme $C = 878$ destiné à Bob. Mr X. a découvert que le compliqueur de Bob est $E = 1675$.
Pouvez-vous aider Mr X. à décrypter le cryptogramme C ?

Exercice 8

Bob choisit le sac-à-dos facile $SADF = (1, 3, 7, 13, 27)$ et le module $N = 103$.

- Bob choisit le compliqueur $E = 2$. Quelle est la clé publique de Bob ?
- En utilisant cette clé publique, Alice chiffre un message binaire M destiné à Bob et obtient le cryptogramme $C = 86$. Mr X. intercepte ce message. Montrer que Mr X. peut facilement décrypter C.
- Bob décide de changer de compliqueur et choisit cette fois $E' = 23$. Calculer la nouvelle clé publique de Bob.
- La faille de sécurité précédente est-elle réparée ?

Encodage binaire des caractères

- En utilisant 5 bits, on peut coder $2^5 = 32$ caractéres.
- Dans cette partie, on utilise la table des 32 caractères suivants :

A	B	C	D	E	F	G	H
(0,0,0,0,0)	(1,0,0,0,0)	(0,1,0,0,0)	(1,1,0,0,0)	(0,0,1,0,0)	(1,0,1,0,0)	(0,1,1,0,0)	(1,1,1,0,0)

I	J	K	L	M	N	O	P
(0,0,0,1,0)	(1,0,0,1,0)	(0,1,0,1,0)	(1,1,0,1,0)	(0,0,1,1,0)	(1,0,1,1,0)	(0,1,1,1,0)	(1,1,1,1,0)

Q	R	S	T	U	V	W	X
(0,0,0,0,1)	(1,0,0,0,1)	(0,1,0,0,1)	(1,1,0,0,1)	(0,0,1,0,1)	(1,0,1,0,1)	(0,1,1,0,1)	(1,1,1,0,1)

Y	Z		.	,	?	-	’
(0,0,0,1,1)	(1,0,0,1,1)	(0,1,0,1,1)	(1,1,0,1,1)	(0,0,1,1,1)	(1,0,1,1,1)	(0,1,1,1,1)	(1,1,1,1,1)

Exemple

- La clé publique de Bob est $SADD = (81, 108, 88, 48, 96, 64)$ et $N = 155$, sa clé privée est $D = 23$ et $SADF = (3, 4, 9, 19, 38, 77)$.
- Alice veut transmettre le mot "CODE" à Bob. Elle devra donc, en se référant à la table ci-dessus, chiffrer la chaîne $[(0, 1, 0, 0, 0), (0, 1, 1, 1, 0), (1, 1, 0, 0, 0), (0, 0, 1, 0, 0)]$. Comme la clé publique de Bob comporte 6 nombres, elle devra ensuite regrouper ces bits par paquets de 6, et au besoin ajouter des bits aléatoires (par exemple $(1, 1, 0, 1)$) pour obtenir un nombre de bits multiple le 6, ce qui donne : $[(0, 1, 0, 0, 0, 0), (1, 1, 1, 0, 1, 1), (0, 0, 0, 0, 0, 1), (0, 0, 1, 1, 0, 1)]$.

Exemple

- Le bloc $(0, 1, 0, 0, 0, 0)$ sera chiffré :
  $81 \times 0 + 108 \times 1 + 88 \times 0 + 48 \times 0 + 96 \times 0 + 64 \times 0 \equiv 108 \mod 155$.
- Le bloc $(1, 1, 1, 0, 1, 1)$ sera chiffré :
  $81 \times 1 + 108 \times 1 + 88 \times 1 + 48 \times 0 + 96 \times 1 + 64 \times 1 \equiv 437 \equiv 127 \mod 155$.
- Le bloc $(0, 0, 0, 0, 0, 1)$ sera chiffré :
  $81 \times 0 + 108 \times 0 + 88 \times 0 + 48 \times 0 + 96 \times 0 + 64 \times 1 \equiv 64 \mod 155$.
- Le bloc $(0, 0, 1, 1, 0, 1)$ sera chiffré :
  $81 \times 0 + 108 \times 0 + 88 \times 1 + 48 \times 1 + 96 \times 0 + 64 \times 1 \equiv 200 \equiv 45 \mod 155$.

Le message chiffré est donc : $[108, 127, 64, 45]$.

Exemple

- Pour déchiffrer, Bob utilise sa clé privée et applique l'algorithme glouton. Il obtient alors :
  - $108 \times 23 \mod 155 \equiv 2484 \mod 155 \equiv 4 \mod 155 \rightsquigarrow (0,1,0,0,0,0)$.
  - $127 \times 23 \mod 155 \equiv 2921 \mod 155 \equiv 131 \mod 155 \equiv 3+4+9+38+77 \mod 155 \rightsquigarrow (1,1,1,0,1,1)$.
  - $64 \times 23 \mod 155 \equiv 1472 \mod 155 \equiv 77 \mod 155 \rightsquigarrow (0,0,0,0,0,1)$.
  - $45 \times 23 \mod 155 \equiv 1035 \mod 155 \equiv 105 \mod 155 \equiv 9+19+77 \mod 155 \rightsquigarrow (0,0,1,1,0,1)$.

  Le message déchiffré est donc $[(0,1,0,0,0,0),(1,1,1,0,1,1),(0,0,0,0,0,1),(0,0,1,1,0,1)]$, ce qui est bien le message qu'avait envoyé Alice. Pour retrouver le mot, Bob n'a plus qu'à regrouper les bits par paquets de 5 et consulter le tableau de conversion.

Remarques

- Il est important que le nombre de bits par paquet soit différent de la longueur du sac-à-dos.
- En effet, si ce n'était pas le cas, on pourrait attaquer le texte chiffré par une analyse des fréquences, car chaque lettre serait toujours chiffrée par le même nombre.
- Il est clair que plus le sac-à-dos est long, plus le message sera difficile à décrypter.
- Dans la pratique, on utilise au moins 250 nombres dans le sac-à-dos et le module N est choisi pour avoir une longueur comprise entre 100 et 200 bits.

Exercice 9

1. Charlie souhaite transmettre à Bob le message secret "DIAMANTS". Chiffrez ce message en utilisant le même chiffre que dans l'exemple ci-dessus.
2. Par ailleurs, Bob a aussi reçu d'Alice le message secret suivant : $[81, 96, 34, 49, 129]$. Déchiffrez-le.

Exercice 10

1. Alice souhaite envoyer à Bob le message secret "POLICE ?". Chiffrez ce message.
2. Un peu plus tôt, Bob a aussi reçu de Charlie le message secret suivant : $[108, 64, 86, 22, 63, 88, 129]$. Déchiffrez-le.

Une fausse sécurité

- La sécurité supposée de la méthode du sac-à-dos reposait sur le fait que le seul algorithme connu qui permet de résoudre n'importe quel problème de sac-à-dos est celui qui consiste à essayer toutes les possibilités (voir paragraphe 2).
- Ceci est toujours vrai actuellement.
- Mais malheureusement, Adi Shamir a découvert au début des années 1980 que les sac-à-dos difficiles obtenus par multiplication modulaire à partir d'un sac-à-dos facile ne sont pas quelconques.
- Ils ont des propriétés particulières, trop compliquées pour être présentées ici.

L'algorithme LLL

- Des algorithmes sophistiqués exploitant ces propriétés ont été inventés. Ils permettent de retrouver le sac-à-dos facile qui a servi pour fabriquer un sac-à-dos difficile donné.
- Par la suite, chaque fois qu'on a essayé d'améliorer la méthode du sac-à-dos pour corriger ce défaut, un algorithme plus puissant permettant de retrouver le sac-à-dos facile a été trouvé.

FIN