Licence informatique & vidéoludisme
Semestre 5

Introduction à la sécurité

Chapitre 3
La cryptographie asymétrique

Pablo Rauzy <pr@up8.edu>
`pablo.rauzy.name/teaching/is`

La cryptographie asymétrique

Introduction

- La cryptographie *asymétrique* (aussi appelée *à clef publique et privée*) est une technique de chiffrement relativement récente.
- L'idée est de pouvoir communiquer de manière sûre sans dépendre de la communication d'une clef secrète.

La paire de clef

- Le principe est d'avoir deux clefs, de telle sorte que
  - un message chiffré avec la première clef ne peut être déchiffré qu'avec la seconde,
  - un message chiffré avec la seconde clef ne peut être déchiffré qu'avec la première.
- Par convention, l'une de ces clefs est appelée la *clef privée* et l'autre la *clef publique*.

Double fonctionnalité

- Un tel système permet de faire deux choses majeures.

1. Assurer la confidentialité du message :

2. Assurer l'authenticité de l'émetteur :

Double fonctionnalité

- Un tel système permet de faire deux choses majeures.

1. Assurer la confidentialité du message :
   - L'émetteur chiffre un message avec la clef publique du destinataire.
   - Seul le destinataire peut le déchiffrer avec sa clef privée.
2. Assurer l'authenticité de l'émetteur :
   - L'émetteur chiffre un message avec sa clef privée.
   - Le destinataire peut déchiffrer ce message avec la clef publique de l'émetteur.

Historique

- Le concept de cryptographie asymétrique a été présenté publiquement pour la première fois par Whitfield Diffie et Martin Hellman à la *National Computer Conference* en 1976.
- En 1974, Ralph Merkle a travaillé sur des puzzles qui constituent la première construction à clef asymétrique, mais ses travaux ne sont publié qu'en 1978.

- En fait, dans l'article de 1976, Diffie et Hellman n'ont pas donné d'exemple de cryptosystème asymétrique (ils n'en avaient pas trouvé).
- C'est en 1978, que Ronald Rivest, Adi Shamir, et Leonard Adleman présentent *RSA*.
- Le système Merkle-Hellman est généralement considéré comme la première réalisation pratique d'un système de chiffrement à clef publique, mais il a été cassé par Shamir en 1982.

- En parallèle des recherches publiques, le GCHQ aurait mené des recherches secrètes ayant abouties dès le début des années 1970 aux concepts et outils de la cryptographie asymétrique.
  - En 1970, James Ellis invente le concept.
  - En 1973, Clifford Cocks invente l'algorithme de RSA.
  - En 1974, Malcolm Williamson invente un protocole d'échange de clef très proche de celui de Diffie et Hellman.
- Ces découvertes n'ont été rendues publiques par le GCHQ qu'en 1997.

Fonctionnement général

- Le but est de pouvoir communiquer de manière sécurisée sur un canal non sécurisé.

Les puzzles de Merkle

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.
  2. L'émetteur chiffre tous ces messages avec des petites clefs de telle sorte qu'il soit possible de les déchiffrer en attaquant par force brute.

Les puzzles de Merkle

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.
  2. L'émetteur chiffre tous ces messages avec des petites clefs de telle sorte qu'il soit possible de les déchiffrer en attaquant par force brute.
  3. L'émetteur envoie tous les chiffrés au destinataire.

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.
  2. L'émetteur chiffre tous ces messages avec des petites clefs de telle sorte qu'il soit possible de les déchiffrer en attaquant par force brute.
  3. L'émetteur envoie tous les chiffrés au destinataire.
  4. Le destinataire en choisi un au hasard, et le casse par force brute.

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.
  2. L'émetteur chiffre tous ces messages avec des petites clefs de telle sorte qu'il soit possible de les déchiffrer en attaquant par force brute.
  3. L'émetteur envoie tous les chiffrés au destinataire.
  4. Le destinataire en choisi un au hasard, et le casse par force brute.
  5. Le destinataire envoie l'identifiant id qu'il a trouvé à l'émetteur, et garde la clef key.

Les puzzles de Merkle

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.
  2. L'émetteur chiffre tous ces messages avec des petites clefs de telle sorte qu'il soit possible de les déchiffrer en attaquant par force brute.
  3. L'émetteur envoie tous les chiffrés au destinataire.
  4. Le destinataire en choisi un au hasard, et le casse par force brute.
  5. Le destinataire envoie l'identifiant id qu'il a trouvé à l'émetteur, et garde la clef key.
  6. L'émetteur retrouve la clef key qui va avec l'identifiant id qu'il reçoit.

Les puzzles de Merkle

- La première idée de mise en œuvre d'un tel système sont les *puzzles de Merkle*.
- Le principe est le suivant :
  1. L'émetteur crée plein de messages de la forme (id, key) avec id un identifiant unique et key une clef aléatoire pour un cryptosystème symétrique.
  2. L'émetteur chiffre tous ces messages avec des petites clefs de telle sorte qu'il soit possible de les déchiffrer en attaquant par force brute.
  3. L'émetteur envoie tous les chiffrés au destinataire.
  4. Le destinataire en choisi un au hasard, et le casse par force brute.
  5. Le destinataire envoie l'identifiant id qu'il a trouvé à l'émetteur, et garde la clef key.
  6. L'émetteur retrouve la clef key qui va avec l'identifiant id qu'il reçoit.
  7. Les deux peuvent maintenant communiquer en chiffrant leur message de manière symétrique.

Formalisation

- On veut un système dans lequel on a une fonction F telle que :
  - on soit capable de générer des paires (pk, sk) telles que :
  - $c = F(pk, m)$ soit facile à calculer,
  - $m = F(sk, c)$ soit facile à calculer,
  - mais qu'il soit très difficile de retrouver m si on a pas sk.
  - Cela implique qu'il soit très difficile de retrouver sk à partir de pk.

- Une *fonction à sens unique* est une fonction qu'il est facile de calculer, mais qu'il est très difficile d'inverser.
- Une *fonction à brèche secrète* est une fonction à sens unique pour laquelle il existe un secret qui permet de facilité l'inversion de la fonction.
- En fait, la fonction F dont on souhaite disposer est une fonction à brèche secrète.

Problème ouvert

- On a aucune idée de si de telles fonctions existent effectivement.
- C'est une question ouverte de l'informatique et des mathématiques.
- Leur existence impliquerait de répondre non à la question bien connue $P \stackrel{?}{=} NP$.

Candidats

- Dans l'état actuel des connaissances, on a cependant plusieurs candidats qui pourraient être des fonctions à sens unique.
- C'est à dire qu'on connaît certaines fonctions qui sont calculables en temps polynomial dans un sens, et pour lesquels on ne connaît pas d'algorithme pour calculer la fonction inverse en temps polynomial.
- Exemples :
  - la multiplication (et la factorisation en nombre premier),
  - l'exponentiation modulaire (et le logarithme discret),
  - la multiplication scalaire sur courbe elliptique (et la division).
- Voyons un cryptosystème pour chacune de ces fonctions candidates à sens unique.

RSA

- RSA est un algorithme de cryptographie asymétrique qui est très utilisé.
- Il a été breveté mais ce brevet a expiré le 21 septembre 2000.
- Il peut être utilisé pour chiffrer ou pour signer des messages.

Fonctionnement

- RSA utilise la congruence sur les entiers et le petit théorème de Fermat pour construire une fonction à sens unique et à brèche secrète.
- Tous les calculs se font modulo n, où n est le produit de deux nombres premiers.
- La sécurité repose sur la difficulté de factoriser n.
- Le chiffrement comme le déchiffrement consistent à élever le message à une certaine puissance (qui dépendra de la paire de clefs) modulo n.

Génération des clefs

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.

Génération des clefs

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.
3. Calculer $\varphi(n) = (p-1)(q-1)$, la valeur de l'indicatrice d'Euler pour n.

Génération des clefs

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.
3. Calculer $\varphi(n) = (p-1)(q-1)$, la valeur de l'indicatrice d'Euler pour n.
4. Choisir un entier e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de chiffrement*.

Génération des clefs

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.
3. Calculer $\varphi(n) = (p-1)(q-1)$, la valeur de l'indicatrice d'Euler pour n.
4. Choisir un entier e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de chiffrement*.
5. Calculer l'entier d inverse de e modulo $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de déchiffrement*.

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.
3. Calculer $\varphi(n) = (p-1)(q-1)$, la valeur de l'indicatrice d'Euler pour n.
4. Choisir un entier e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de chiffrement*.
5. Calculer l'entier d inverse de e modulo $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de déchiffrement*.
   - On est sûr que d existe par le théorème de Bachet-Bézout, qui dit qu'il existe d et k tels que $ed + k\varphi(n) = 1 \iff ed \equiv 1 \mod \varphi(n)$.

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.
3. Calculer $\varphi(n) = (p-1)(q-1)$, la valeur de l'indicatrice d'Euler pour n.
4. Choisir un entier e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de chiffrement*.
5. Calculer l'entier d inverse de e modulo $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de déchiffrement*.
   - On est sûr que d existe par le théorème de Bachet-Bézout, qui dit qu'il existe d et k tels que $ed + k\varphi(n) = 1 \iff ed \equiv 1 \mod \varphi(n)$.
   - On sait calculer d efficacement avec l'algorithme d'Euclide étendu.

1. Choisir **aléatoirement** p et q deux grands nombres premiers distincts.
2. Calculer $n = pq$, qu'on appelle le *module de chiffrement*.
3. Calculer $\varphi(n) = (p-1)(q-1)$, la valeur de l'indicatrice d'Euler pour n.
4. Choisir un entier e premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de chiffrement*.
5. Calculer l'entier d inverse de e modulo $\varphi(n)$ et strictement inférieur à $\varphi(n)$, qu'on appelle l'*exposant de déchiffrement*.
   - On est sûr que d existe par le théorème de Bachet-Bézout, qui dit qu'il existe d et k tels que $ed + k\varphi(n) = 1 \iff ed \equiv 1 \mod \varphi(n)$.
   - On sait calculer d efficacement avec l'algorithme d'Euclide étendu.
6. La clef publique est le couple (e, n) et la clef privée est le couple (d, n).

- Le message m doit être strictement inférieur à n.
- On calcule le chiffré $c = m^e \bmod n$.

Déchiffrement

- Le chiffré c est strictement inférieur à n.
- On calcule le message $m = c^d \bmod n$.

- Pour assurer l'authenticité d'un message, son émetteur peut utiliser RSA pour le signer.
- Pour cela il lui suffit de chiffrer un condensé (un hash) du message avec sa clef privée : $s = H(m)^d \bmod n$.
- Le message signé est alors (m, s).

Vérification de signature

- Pour vérifier la signature, il suffit de comparer $H(m)$ avec le résultat du déchiffrement de la signature à l'aide de la clef publique :
  $v = s^e \bmod n$.
- Cela assure non seulement l'authenticité du message (on est assuré de l'identité de son émetteur), mais aussi son intégrité.

Vérification de signature

- Pour vérifier la signature, il suffit de comparer $H(m)$ avec le résultat du déchiffrement de la signature à l'aide de la clef publique :
  $v = s^e \bmod n$.
- Cela assure non seulement l'authenticité du message (on est assuré de l'identité de son émetteur), mais aussi son intégrité.
- En effet, si le message est corrompu, son condensé est différent.

- Comment faire pour communiquer en s'assurant à la fois de la confidentialité, de l'intégrité, et de l'authenticité ?

Justification (1/2)

- Rappel du petit théorème de Fermat : “si p est un nombre premier et si a est un entier non divisible par p, alors $a^{p-1} - 1$ est un multiple de p”.
  C'est à dire : $a^{p-1} \equiv 1 \mod p$.

Justification (1/2)

- Rappel du petit théorème de Fermat : “si p est un nombre premier et si a est un entier non divisible par p, alors $a^{p-1} - 1$ est un multiple de p”.
  C'est à dire : $a^{p-1} \equiv 1 \mod p$.
- Comme n est le produit de deux nombres premiers :
  - soit m est premier avec n,
  - soit il est multiple de p et pas de q,
  - soit il est multiple de q et pas de p,
  - soit il est multiple de n.

Justification (1/2)

- Rappel du petit théorème de Fermat : “si p est un nombre premier et si a est un entier non divisible par p, alors $a^{p-1} - 1$ est un multiple de p”.
  C’est à dire : $a^{p-1} \equiv 1 \mod p$.
- Comme n est le produit de deux nombres premiers :
  - soit m est premier avec n,
  - soit il est multiple de p et pas de q,
  - soit il est multiple de q et pas de p,
  - soit il est multiple de n.
- Le dernier cas est trivial car $m \equiv 0 \mod n$.

Justification (1/2)

- Rappel du petit théorème de Fermat : “si p est un nombre premier et si a est un entier non divisible par p, alors $a^{p-1} - 1$ est un multiple de p”.
  C'est à dire : $a^{p-1} \equiv 1 \mod p$.
- Comme n est le produit de deux nombres premiers :
  - soit m est premier avec n,
  - soit il est multiple de p et pas de q,
  - soit il est multiple de q et pas de p,
  - soit il est multiple de n.
- Le dernier cas est trivial car $m \equiv 0 \mod n$.
- Dans le premier cas, on peut simplement utiliser le théorème d'Euler, qui généralise le petit théorème de Fermat à tout entier n premier avec a : $a^{\varphi(n)} \equiv 1 \mod n$.
  Ce qui implique directement : $a^b \equiv a^{b \bmod \varphi(n)} \mod n$.
  On a donc $c^d \equiv m^{ed} \equiv m^{ed \bmod \varphi(n)} \equiv m \mod n$.

Justification (1/2)

- Rappel du petit théorème de Fermat : “si p est un nombre premier et si a est un entier non divisible par p, alors $a^{p-1} - 1$ est un multiple de p”.
  C’est à dire : $a^{p-1} \equiv 1 \mod p$.
- Comme n est le produit de deux nombres premiers :
  - soit m est premier avec n,
  - soit il est multiple de p et pas de q,
  - soit il est multiple de q et pas de p,
  - soit il est multiple de n.
- Le dernier cas est trivial car $m \equiv 0 \mod n$.
- Dans le premier cas, on peut simplement utiliser le théorème d’Euler, qui généralise le petit théorème de Fermat à tout entier n premier avec a : $a^{\varphi(n)} \equiv 1 \mod n$.
  Ce qui implique directement : $a^b \equiv a^{b \bmod \varphi(n)} \mod n$.
  On a donc $c^d \equiv m^{ed} \equiv m^{ed \bmod \varphi(n)} \equiv m \mod n$.
- Les deux cas restant sont symétriques.

Justification (2/2)

- Supposons que m soit multiple de q et pas de p.

Justification (2/2)

- Supposons que m soit multiple de q et pas de p.
- On a par le petit théorème de Fermat que $m^{p-1} \equiv 1 \mod p$.

Justification (2/2)

- Supposons que m soit multiple de q et pas de p.
- On a par le petit théorème de Fermat que $m^{p-1} \equiv 1 \mod p$.
- On a que $c^d \equiv m^{ed} \mod n$.

- Supposons que m soit multiple de q et pas de p.
- On a par le petit théorème de Fermat que $m^{p-1} \equiv 1 \mod p$.
- On a que $c^d \equiv m^{ed} \mod n$.
- Par construction de la paire de clef, on a que $ed \equiv 1 \mod \varphi(n)$, c'est à dire $\exists k, ed = 1 + k(p-1)(q-1)$.

- Supposons que m soit multiple de q et pas de p.
- On a par le petit théorème de Fermat que $m^{p-1} \equiv 1 \mod p$.
- On a que $c^d \equiv m^{ed} \mod n$.
- Par construction de la paire de clef, on a que $ed \equiv 1 \mod \varphi(n)$, c'est à dire $\exists k, ed = 1 + k(p-1)(q-1)$.
- On a donc $c^d \equiv m^{1+k(p-1)(q-1)} \equiv m \cdot (m^{p-1})^{k(q-1)} \equiv m \mod p$.

Justification (2/2)

- Supposons que m soit multiple de q et pas de p.
- On a par le petit théorème de Fermat que $m^{p-1} \equiv 1 \mod p$.
- On a que $c^d \equiv m^{ed} \mod n$.
- Par construction de la paire de clef, on a que $ed \equiv 1 \mod \varphi(n)$, c'est à dire $\exists k, ed = 1 + k(p-1)(q-1)$.
- On a donc $c^d \equiv m^{1+k(p-1)(q-1)} \equiv m \cdot (m^{p-1})^{k(q-1)} \equiv m \mod p$.
- Et comme on a $m \equiv 0 \mod q$, on a bien $c^d \equiv m \mod n$.

Asymétrie

- Pour chiffrer le message, il suffit de connaître e et n.
- Pour le déchiffrer, il faut d et n.
- Pour calculer d à partir de e et n (ce que voudrait faire un attaquant), il faut trouver l'*inverse modulaire* de e modulo $\varphi(n) = (p-1)(q-1)$.
- On ne sait pas faire cela sans connaître p et q.
- La sécurité de RSA repose sur la difficulté de la factorisation en nombres premiers.

- Factoriser est difficile, mais la structure du problème le rend quand même plus rapide à résoudre que de devoir essayer toutes les clefs d'une longueur donnée comme avec la cryptographie symétrique.
- Les tailles de clefs sont donc beaucoup plus importantes.
- De nos jours, le module de chiffrement doit faire au moins 2048 bits, et 4096 sont recommandés.

Cryptographie hybride

- Chiffrer tout un message avec RSA serait long en terme de calcul.
- Ce qui est fait le plus souvent est de se servir de la cryptographie asymétrique pour échanger une *clef de session* générée aléatoirement pour être utilisée symétriquement.

Échange de clef de Diffie-Hellman

- DH est une méthode de cryptographie asymétrique permettant à deux personnes de se mettre d'accord sur une clef secrète au travers d'un canal de communication non sécurisé.

Fonctionnement

- Les calculs se font dans un groupe $(\mathbb{Z}/p\mathbb{Z})^*$.
- DH utilise la commutativité et l'associativité de l'opération du groupe pour permettre aux deux participant·es de construire secret commun en échangeant seulement des éléments publics.
- La sécurité repose sur la difficulté d'inverser une exponentiation modulaire, c'est à dire de calculer un logarithme discret.

Protocole à 2 participant·es

- On choisit un groupe $(\mathbb{Z}/p\mathbb{Z})^*$, et g un générateur de ce groupe.
- Ces informations sont publiques.

1. Alice choisit aléatoirement un entier a, et calcul $A = g^a \mod p$.

Protocole à 2 participant·es

- On choisit un groupe $(\mathbb{Z}/p\mathbb{Z})^*$, et g un générateur de ce groupe.
- Ces informations sont publiques.

1. Alice choisit aléatoirement un entier a, et calcul $A = g^a \mod p$.
2. Alice envoie A à Bob.

Protocole à 2 participant·es

- On choisit un groupe $(\mathbb{Z}/p\mathbb{Z})^*$, et g un générateur de ce groupe.
- Ces informations sont publiques.

1. Alice choisit aléatoirement un entier a, et calcul $A = g^a \mod p$.
2. Alice envoie A à Bob.
3. Bob choisit aléatoirement un entier b, et calcul $B = g^b \mod p$.
   Il calcule aussi $K = A^b = (g^a)^b = g^{ab} \mod p$.

Protocole à 2 participant·es

- On choisit un groupe $(\mathbb{Z}/p\mathbb{Z})^*$, et g un générateur de ce groupe.
- Ces informations sont publiques.

1. Alice choisit aléatoirement un entier a, et calcul $A = g^a \mod p$.
2. Alice envoie A à Bob.
3. Bob choisit aléatoirement un entier b, et calcul $B = g^b \mod p$.
   Il calcule aussi $K = A^b = (g^a)^b = g^{ab} \mod p$.
4. Bob envoie B à Alice.

Protocole à 2 participant·es

- On choisit un groupe $(\mathbb{Z}/p\mathbb{Z})^*$, et g un générateur de ce groupe.
- Ces informations sont publiques.

1. Alice choisit aléatoirement un entier a, et calcul $A = g^a \mod p$.
2. Alice envoie A à Bob.
3. Bob choisit aléatoirement un entier b, et calcul $B = g^b \mod p$.
   Il calcule aussi $K = A^b = (g^a)^b = g^{ab} \mod p$.
4. Bob envoie B à Alice.
5. Alice calcule $K = B^a = (g^b)^a = g^{ab} \mod p$.

Exercice

- Jouer le protocole pour $p = 23$ et $g = 3$.

Asymétrie

- Les calculs d'exponentiations modulaires sont peu coûteux (linéaire en la taille de l'exposant).
- En revanche, un attaquant doit forcément retrouver a ou b pour calculer K, ce qui suppose a priori de calculer un logarithme discret.
- C'est sur la difficulté de ce problème que repose la sécurité de DH.

Sécurité

- Comme avec RSA, la taille des clefs est importante.
- Par rapport aux méthodes de calcul de logarithme discret connues, il est conseillé d'utiliser des nombres a et b du double de la taille de clef symétrique du niveau de sécurité voulu (par exemple, 256 bits pour une sécurité 128 bits symétrique).
- Le module p doit être grand (comparable aux recommandations pour le module de RSA).

Extension à n participant·es ?

- Proposez une version du protocole avec 3 participant·es, par exemple où Carole se joindrait à Bob et Alice.
- Est-il possible de l'étendre à n participant·es ?

Courbes elliptiques

- Les courbes elliptiques sont une famille d'objet mathématiques munis d'une structure de groupe additif, qui permet aussi de les utiliser en cryptographie en faisant reposer la sécurité sur la transposition à ces structures du problème du logarithme discret.
- L'échange de clef de Diffie-Hellman est possible sur les courbes elliptiques (ECDH).

Partage de secret

- Comment partager un secret (par exemple un mot de passe) de sorte à ce qu'il faille un minimum défini de personnes qui se mettent d'accord pour le révéler ?
- La solution est le *partage de secret*.
- Formellement, on a n dépositaires qui reçoivent chacun une *part* du secret, et on veut qu'il faille au moins que k dépositaires parmi les n mettent leur part en commun pour retrouver le secret.
- Si $k = n$, avez-vous une idée de comment faire ?

→ Dans le cas général, on peut utiliser SSSS (*Shamir's Secret Sharing Scheme*).

Chiffrement homomorphe

- Comment faire si on a besoin de déléguer un calcul (par exemple dans le cloud), mais qu'on veut garder confidentielles les données sur lequel il porte ?
- La solution est le *chiffrement homomorphe*.
- Un cryptosystème est dit homomorphe si il possède certaines caractéristiques algébriques qui permettent de réaliser des opérations sur les chiffrés.
- Soit $F : A \rightarrow B$ une fonction de chiffrement, et soient $\odot_A$ et $\odot_B$ des opérations sur A et B.
- On dit que le cryptosystème (F, F^{-1}) est homomorphe pour $\odot_A$ si on a $\odot_B$ telle que $F^{-1}(F(x) \odot_B F(y)) = x \odot_A y$.
- On parle de cryptosystème *partiellement homomorphe* quand il commute avec un ensemble restreint d'opérations.

→ Regardons quelques exemples (Paillier, HE1) ensemble.