

Cryptographie à clé publique

Cours 1

Julien Lavauzelle

Université Paris 8

Master 1 mathématiques et applications – parcours ACC

27/01/2023

Quelques informations personnelles :

- Julien Lavauzelle, maître de conférences à l'Université Paris 8
- email : julien.lavauzelle@univ-paris8.fr
- Ma recherche : codes correcteurs, applications en cryptographie
- Enseigne aussi : Algèbre linéaire 1 (L1), Calcul Formel (L2), Tremplin Master (L3), Théorie de l'information (M1), Algorithmes arithmétiques (M2)

La **page web** de ce cours :

www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/clef-publique.html

Contiendra :

- informations générales
- slides de cours
- TDs, solutions
- exercices à rendre

Modalités d'évaluation de l'UE :

1. **Première note : contrôle continu**

- 4 ou 5 **devoirs à la maison**, à rendre avant la fin de la séance suivante.
- Principalement des exercices de programmation.

2. **Seconde note : partiel sur table** en fin d'année

Note finale : 50 % note 1 + 50 % note 2




Planning :

12 séances, le vendredi de 09h00 à 11h30, en salle A043



Programme provisoire du cours :

1. Introduction, notion de problème difficile
2. Échange de clés de Diffie–Hellman
3. Chiffrement : RSA, El-Gamal, Rabin, ...
4. Signature : DSA, ECDSA, Schnorr, ...
5. Infrastructures, gestion de clés, normes et standards, ...
6. Cryptographie post-quantique
7. Si on a le temps :
 - autres primitives modernes (IBE, cryptographie « probabiliste »)
 - applications

Fondamentaux :

-  *Cryptographie : Théorie et Pratique*. D. Stinson. Vuibert. 2003.
-  *Exercices et problèmes de cryptographie*. D. Vergnaud. Dunod. 2018.
-  *Handbook of Applied Cryptography*. A. Menezes, P. van Oorschot, S. Vanstone. CRC. 1997.

Plus avancé :

-  *Mathematics of Public Key Cryptography*. S. Galbraith. en ligne. 2018.
<https://www.math.auckland.ac.nz/~sgal018/crypto-book/main.pdf>
-  *Introduction to modern cryptography*. J. Katz, Y. Lindell. CRC. 2007.

1. Introduction à la cryptographie à clé publique

2. Échange de clés

1. Introduction à la cryptographie à clé publique

2. Échange de clés

CRYPTOGRAPHIE : conception de systèmes pour assurer la **sécurité** de l'information en présence d'**adversaires**.

« **Sécurité** » ?

1. **confidentialité** : l'information n'est pas accessible aux tiers/adversaires
→ solution : **chiffrement**
2. **intégrité** : l'information n'est pas modifiée (sans le savoir)
→ solution : **signature, MAC** (*message authentication code*)
3. **authentification** : l'identité d'une entité, ou l'origine d'un fichier, est garantie
→ solution : **signature, MAC**
4. **non-répudiation** (pour la signature) : il est impossible de renier un fichier envoyé
5. ...

CRYPTOGRAPHIE : conception de systèmes pour assurer la **sécurité** de l'information en présence d'**adversaires**.

« **Adversaires** » ?

→ Différents modèles d'**adversaires**.

1. **Adversaires passifs**. Ces adversaires observent les communications qui transitent sur le canal public, mais ne les modifient pas.
2. **Adversaires actifs**. Ces adversaires peuvent modifier les communications qui transitent sur le canal public.

→ Différents modèles d'**attaques**. Ces modèles varient suivant la primitive cryptographique (chiffrement, signature, etc.).

Le **principe de Kerckhoffs** exprime que la sécurité du système ne doit reposer que sur le secret de la clé.

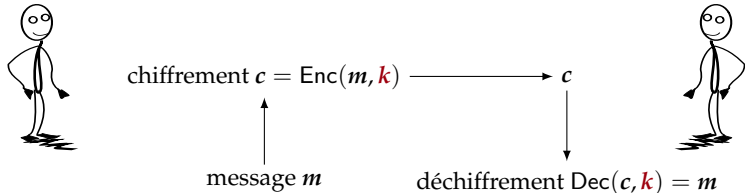
Conséquence. On suppose que tout adversaire connaît les algorithmes engagés dans le protocole.

La solution « historique » : la cryptographie symétrique.

Les protagonistes (Alice et Bob) possèdent un **secret commun**.

De manière **symétrique**, ils utilisent ce secret **commun** pour masquer l'information **et** pour la rendre de nouveau intelligible.

→ *Exemple pour le chiffrement* :



Typiquement : chiffrement AES, une clé secrète commune k , de taille 128 à 256 bits.

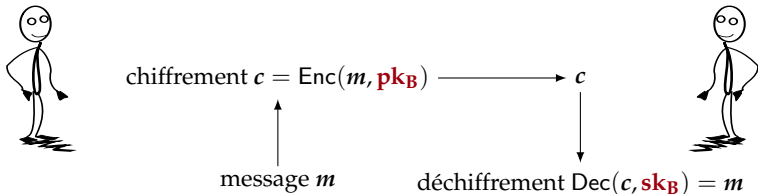
Quelques **inconvénients** liés à la cryptographie symétrique.

1. Si Alice et Bob ne se sont jamais rencontrés, comment peuvent-ils **mettre en place un secret commun** ?
2. Comment **engager** une conversation avec une entité inconnue ?
→ exemple : pour le paiement en ligne, vous envoyez vos données à une nouvelle autorité bancaire
3. Dans un réseau à n participants, il y a **une clé** à stocker **par paire** de participants, c'est-à-dire $\frac{n(n-1)}{2}$ clés
→ pour 1000 participants, $\simeq 500\,000$ clés à stocker...
4. Gestion de clés : ajout de nouveaux participants, mise à jour de clés
→ on a besoin de communiquer avec **tous** les autres participants
→ coût linéaire en n

Solution : la **cryptographie à clé publique**, ou **cryptographie asymétrique**.

- chaque protagoniste i engendre un couple de clés (pk_i, sk_i)
 - la clé publique pk_i est distribuée publiquement (donc, aussi connue des adversaires)
 - la clé privée sk_i est gardée secrètement par i
- n clés publiques, n clés privées (au lieu de $O(n^2)$ clés secrètes)


Exemple pour le chiffrement, avec Alice (A) et Bob (B) :



La clé pk_B est publique (tout le monde chiffre un message pour Bob).

La clé sk_B est privée (seul Bob peut déchiffrer ce qu'on lui envoie).

La cryptographie à clé publique est historiquement liée à **Walter Diffie et Martin Hellman** (prix Turing 2015). Article fondateur :

 *New directions in cryptography*. W. Diffie, M. Hellman. IEEE Trans. Inf. Theory. **1976**.

Auparavant, quelques mentions :

- Ellis en 1970, dans un article intitulé *The possibility of non-secret encryption*, initialement gardé secret.
- Cocks en 1973, dans une note intitulée *A note on non-secret encryption*.

Peu après, premières **mises en pratique** de l'idée. Parmi les plus célèbres :

- le système de chiffrement RSA (par Rivest, Shamir et Adleman) en 1977,
- le système de chiffrement ElGamal en 1984,
- le schéma de signature de Schnorr en 1989.

Domaine encore **en évolution** (recherche très active) :

- progrès sur les algorithmes d'analyse ?
- résistance à l'ordinateur quantique ?

1. Introduction à la cryptographie à clé publique

2. Échange de clés

Objectif. Alice et Bob souhaitent partager un **secret commun**, par exemple un nombre, ou une séquence de bits.

- On suppose qu'**avant** d'initier le protocole, ils ne détiennent aucune information commune.
- Un **attaquant observe** les échanges entre Alice et Bob, et souhaite obtenir des informations sur le secret.
- Des paramètres publics sont accessibles à Alice et Bob, mais également à l'attaquant.

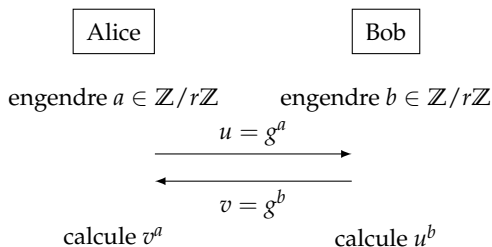
Motivation. Échanger une clé pour initier du chiffrement symétrique.

Pourquoi ?

- en pratique, le chiffrement symétrique est **plus rapide** que le chiffrement asymétrique
- cela permet d'instaurer des clés de session, **éphémères**

Soit G un groupe multiplicatif cyclique, d'ordre $r = |G|$. On rend public un générateur g de G .

Le **protocole de Diffie–Hellman** dans G se définit ainsi :



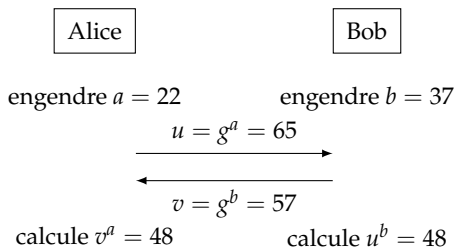
Alice et Bob obtiennent-ils une **valeur commune** ?

$$k = v^a = u^b = g^{ab}$$

Exemple. On prend le groupe multiplicatif $G = (\mathbb{Z}/p\mathbb{Z})^\times$, par exemple avec $p = 83$.

On a alors $r = |G| = p - 1 = 82$.

Un générateur de $(\mathbb{Z}/p\mathbb{Z})^\times$ est $g = 2$.



Le secret commun est 48.

Comment estimer la sécurité d'un système cryptographique ?

Deux approches :

1. sécurité **inconditionnelle** : peu importe la capacité de calcul de l'attaquant, il ne peut pas obtenir d'information sur le secret,
2. sécurité **calculatoire** : attaquer le système nécessite au moins une certaine capacité de calcul.

En pratique, une bonne sécurité calculatoire est suffisante. Actuellement,

- « très bonne sécurité » : $> 2^{128}$ opérations,
- « mauvaise sécurité » : $\leq 2^{80}$ opérations.

Pour la sécurité calculatoire, on essaie de **réduire** la résolution d'un problème que l'on suppose difficile, à la capacité à attaquer le système.

Informellement, cela signifie :

*« Si un attaquant sait attaquer le système cryptographique A ,
alors il peut résoudre le problème mathématique B »*

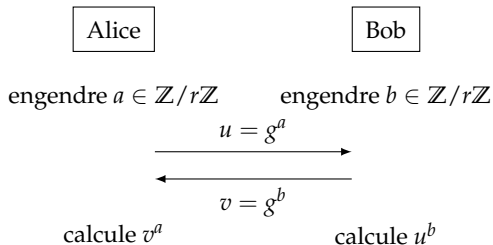
Remarques :

- pour démontrer qu'un système **n'est pas sûr**, il suffit donc de démontrer que l'on peut réduire l'attaque du système à la résolution d'un problème facile
- donc, si on démontre :

*« Si un attaquant sait résoudre le problème B,
alors il peut attaquer le système A »*,

alors :

- il est **nécessaire** que le problème B soit difficile pour que le système A puisse éventuellement être sûr,
- même si le problème B est difficile, cela ne **prouve** pas la sécurité du système A.



Question :

À quel problème calculatoire réduire la sécurité de ce protocole ?

Soit G un groupe fini noté multiplicativement, dont l'ordre est $r \geq 2$.

Définition. Le problème du **logarithme discret** dans G , noté DL (pour *discrete logarithm*), est le problème suivant :

Données : (g, g^x) où g est un générateur de G et $x \in \{0, \dots, r-1\}$

Objectif : calculer $x \in \{0, \dots, r-1\}$

Définition. Le problème de **Diffie–Hellman calculatoire** dans G , noté CDH (pour *computational Diffie–Hellman*), est le problème suivant :

Données : (g, g^a, g^b) où g est un générateur de G et $a, b \in \{0, \dots, r-1\}$

Objectif : calculer $g^{ab} \in G$

Remarque : résoudre le problème de **Diffie–Hellman calculatoire** correspond à obtenir le secret commun d'Alice et Bob.

Théorème. Si l'on sait résoudre DL pour un groupe G , alors on sait résoudre CDH pour ce même groupe.

Preuve.

On **suppose** qu'on a un algorithme A qui résout le problème DL.

$$\text{on a : } \quad \forall g, \forall x, \quad A(g, g^x) = x$$

On **veut** élaborer un algorithme B pour résoudre le problème CDH.

$$\text{on veut : } \quad \forall g, \forall a, b, \quad B(g, g^a, g^b) = g^{ab}$$

On définit l'algorithme $B(g, g^a, g^b)$:

1. Calculer $a = A(g, g^a)$
2. Calculer $b = A(g, g^b)$
3. Calculer ab puis g^{ab} .

Rappel du théorème. Si l'on sait résoudre DL pour un groupe G , alors on sait résoudre CDH pour ce même groupe.

Conséquence. Il est **nécessaire** de choisir des groupes pour lesquels DL est difficile. Lesquels ?

Première idée : le groupe $G = \mathbb{F}_q^\times$ où \mathbb{F}_q est le corps fini à q éléments.

Pour simplifier, on peut prendre $q = p$ un nombre premier ; alors

$$\mathbb{F}_p^\times = \mathbb{Z}/p\mathbb{Z} \setminus \{0\}.$$

Propriétés :

1. C'est un groupe cyclique d'ordre $p - 1$.
2. L'opération de base $(x, y) \mapsto xy$ est calculable efficacement.
3. L'exponentiation $(g, a) \mapsto g^a$ se réalise en $O(\log(a))$ opérations de base.

Que dire de la difficulté du calcul du logarithme discret $(g, g^a) \mapsto a$?

Que dire du calcul du logarithme discret $(g, g^a) \mapsto a$ dans \mathbb{F}_p^\times ?

- On ne connaît pas d'algorithme de résolution avec une complexité polynomiale en $\log a$.
- On pense généralement qu'il n'en existe pas.
- Le meilleur algorithme pour le résoudre est le **crible algébrique** (*number field sieve*, *NFS*), et a pour complexité :

$$2^e \quad \text{avec } e \simeq 1.92(\log p)^{1/3}(\log \log p)^{2/3}$$

Conséquences :

Pour p de taille 2048 bits (c'est-à-dire $p \simeq 2^{2048}$), on obtient une sécurité correcte.

Il faut prendre p de taille 3072 bits pour une sécurité à long terme.

Remarque :

Voici un nombre premier de 2048 bits :

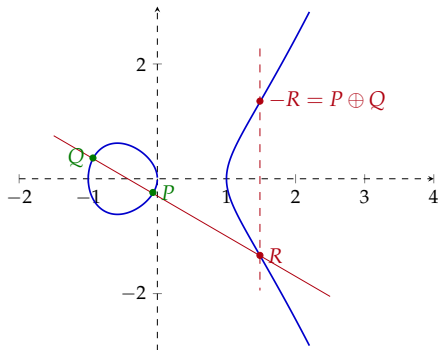
$$p = 4348576336955608842722082340646997911027892947861917701538539035578270$$
$$2708213090870636211985258713190858279602135141487887319571035622744728$$
$$5701299986303947454400689904754064697663494002738078569276393891444301$$
$$7557825716874851815509120573857111723596024028300232821342508911195161$$
$$9049238607489801492726673864401733583454738478465177578434288490173916$$
$$9865781616469518286339988579388655700220961757111910950132973379585844$$
$$7004148053782819526864042462011513619862843312200609650228054772177324$$
$$4393318566782660363542473442825968786189046789677679485116717004196018$$
$$773074102436677143438843334627292260350413426017177699411$$

Seconde idée : le groupe des points d'une courbe elliptique $\mathcal{E}(\mathbb{F}_q)$.

Dans de tels groupes, les meilleurs algorithmes résolvent le problème DL en temps

$$O(\sqrt{p})$$

où p est le plus grand nombre premier qui divise l'ordre du groupe.



Conséquence. Il est recommandé de choisir une courbe elliptique \mathcal{E} et un entier q tel que l'ordre du groupe $\mathcal{E}(\mathbb{F}_q)$ **admette un facteur premier de taille au moins 256 bits.**

Question. Et si l'on souhaite que l'attaquant n'apprenne **aucune information** sur le secret commun?

Définition. Le problème de **Diffie–Hellman décisionnel** dans \mathbb{G} , noté DDH (pour *decisional Diffie–Hellman*), est le problème suivant :

Données : (g, g^a, g^b) où $g \in \mathbb{G}$ et $a, b \in \{0, \dots, r-1\}$

Objectif : distinguer g^{ab} d'un élément aléatoire de \mathbb{G}

Fait. Si DDH est difficile dans un groupe \mathbb{G} , alors il **calculatoirement difficile** d'avoir une **quelconque information** sur le secret commun.

Question. Le problème DDH est-il difficile dans \mathbb{F}_p^\times pour p premier impair ?

Lemme (critère d'Euler). Soit p premier impair et g un générateur de \mathbb{F}_p^\times . Alors, pour $y = g^a \in \mathbb{F}_p^\times$ on a $y^{(p-1)/2} = (-1)^a$.

Conséquence. Étant donnée une instance (g, g^a, g^b) de DDH, on peut connaître la parité de a et b en calculant $(g^a)^{(p-1)/2} = (-1)^a$ et $(g^b)^{(p-1)/2} = (-1)^b$.

Comme $p - 1$ est pair, on obtient :

$$ab \bmod p - 1 \text{ est pair} \iff (a \bmod p - 1 \text{ est pair}) \text{ ou } (b \bmod p - 1 \text{ est pair})$$

Donc, on connaît la parité de ab et on peut donc distinguer g^{ab} d'un élément de G tiré aléatoirement.

Solution 1. On peut utiliser un sous-groupe de \mathbb{F}_p^\times pour lequel cette distinction n'existe plus.

Par exemple, on considère l'ensemble des carrés non-nuls dans \mathbb{F}_p^\times , aussi appelés **résidus quadratiques** modulo p

$$\text{QR}_p^\times := \{y \in \mathbb{F}_p^\times \mid \exists x \in \mathbb{F}_p^\times, y = x^2\}.$$

Remarques importantes.

- On peut caractériser $y \in \text{QR}_p^\times$ par $y^{(p-1)/2} = 1$.
- QR_p^\times est un groupe cyclique d'ordre $(p-1)/2$. (**exercice** : le montrer)

Si $p = 2p' + 1$ avec p' premier (p' est un nombre premier de Sophie Germain), alors l'ordre de QR_p^\times est p' et l'attaque précédente ne fonctionne plus.

Pour de tels nombres premiers, il est conjecturé que DDH est difficile.

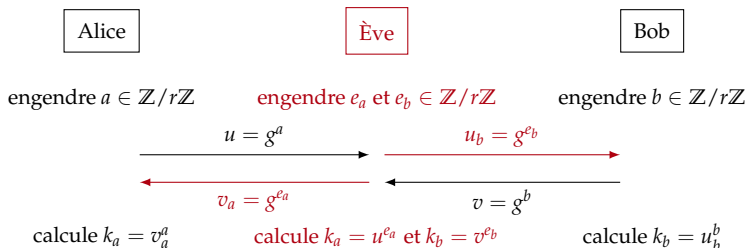
Solution 2. DDH est aussi conjecturé difficile dans les sous-groupes d'ordre premier d'une courbe elliptique.

Dans les versions récentes de **TLS** (*Transport Layer Security*, protocole de sécurité des communications sur Internet), c'est le protocole de Diffie–Hellman sur les courbes elliptiques qui est utilisé.

Attaque par le milieu (*man-in-the-middle*)

Question 2 : en s'introduisant dans le protocole, Ève peut-elle anéantir la confidentialité du secret ?

On suppose ici qu'Ève est un adversaire **actif**. Alors elle peut opérer une attaque « par le milieu » (*man-in-the-middle*).



Alors, Ève possède deux clés k_a et k_b qu'elle peut utiliser respectivement avec Alice et Bob, **sans qu'ils ne s'en aperçoivent**.

Contre-mesure. L'idée est de **signer** les valeurs échangées avec des clés asymétriques, **certifiées** par une autorité de confiance.

Questions ?