

Cryptographie à clé publique

Cours 4

Julien Lavauzelle

Université Paris 8

Master 1 mathématiques et applications – parcours ACC

17/02/2023

Vu à la **séance précédente** :

- Sécurité sémantique, OAEP, application à RSA
- Cryptosystème de Rabin
- Cryptosystème de Goldwasser-Micali
- Cryptosystème de Blum-Goldwasser

Questions ?

1. Notions élémentaires de courbes elliptiques
2. Cryptosystème ElGamal
3. Résoudre le problème du logarithme discret

1. Notions élémentaires de courbes elliptiques
2. Cryptosystème ElGamal
3. Résoudre le problème du logarithme discret

En cryptographie, on souhaite construire des groupes (au sens algébrique) tels que :

1. l'encodage des éléments soit efficace (peu de mémoire, peu de temps),
2. les opérations sur ce groupe sont efficaces,
3. la taille du groupe peut être grande et « diversifiée »,
4. **le problème du logarithme discret est difficile** dans ce groupe.

Pour le groupe \mathbb{F}_q^\times :

- Points 1, 2, 3 : ok.
- Point 4 : moyen.

On cherche donc une **nouvelle famille** de groupes.

→ l'ensemble des points de courbes elliptiques sur \mathbb{F}_q .

Courbes elliptiques réelles

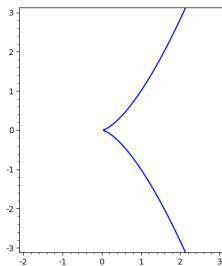
Pour se forger une intuition, on se place d'abord dans \mathbb{R} , l'ensemble des nombres réels.

Définition. Une **courbe elliptique réelle** est l'ensemble des solutions $(x, y) \in \mathbb{R} \times \mathbb{R}$ de l'équation

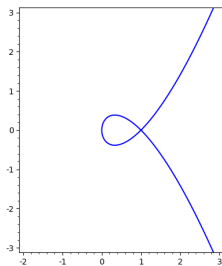
$$y^2 = x^3 + ax + b,$$

auquel on ajoute un point spécial, \mathcal{O} , le **point à l'infini**.

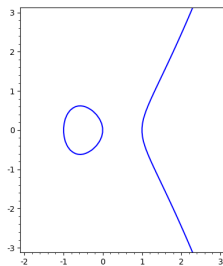
Si $4a^3 + 27b^2 = 0$, on dit que la courbe est **singulière**; c'est un cas particulier que l'on souhaite éviter (en général). Sinon, elle est **régulière**.



$y^2 = x^3$
(singulière)



$y^2 = x(x-1)^2$
(singulière)



$y^2 = x(x-1)(x+1)$
(régulière)

L'ensemble des points d'une courbe elliptique régulière réelle \mathcal{E} (en comptant le point à l'infini \mathcal{O}) forme un **groupe abélien**, noté $\mathcal{E}(\mathbb{R})$.

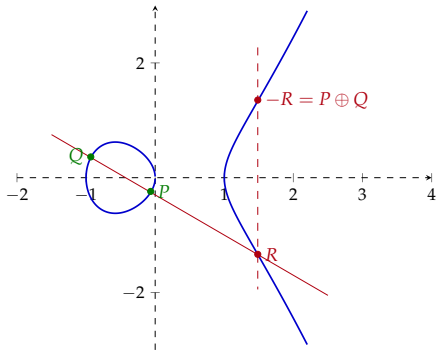
- Le neutre est le point à l'infini \mathcal{O} .
- L'opposé de $P = (x, y)$ est $-P := (x, -y)$.
- L'opération d'addition, notée pour l'instant \oplus , est définie selon la **méthode des cordes et tangentes**.

Si $P \neq Q$, alors on trace la droite passant par P et Q . Elle coupe la courbe \mathcal{E} en un troisième point R . On pose alors $P \oplus Q := -R$.

Remarque. Si $Q = -P$, alors on a bien

$$P \oplus Q = \mathcal{O}$$

car la droite verticale coupe \mathcal{E} en son point à l'infini.

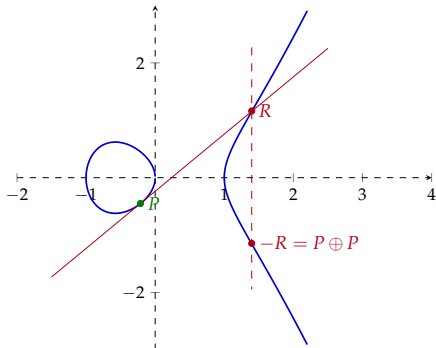


L'ensemble des points d'une courbe elliptique non-singulière \mathcal{E} (en comptant le point à l'infini \mathcal{O}) forme un **groupe abélien**, noté $\mathcal{E}(\mathbb{R})$.

- Le neutre est le point à l'infini \mathcal{O} .
- L'opposé de $P = (x, y)$ est $-P := (x, -y)$.
- L'opération d'addition, notée pour l'instant \oplus , est définie selon la **méthode des cordes et tangentes**.

Si $P = Q$, alors on trace la **tangente** à \mathcal{E} en P . Elle coupe la courbe \mathcal{E} en un second point R . On pose alors $P \oplus P = -R$.

Remarque. Il n'y a que deux points d'intersections entre \mathcal{E} et sa tangente en P (fini), car le polynôme définissant \mathcal{E} est de degré 3, et la multiplicité d'intersection en P est 3 (théorème de Bezout).



Rappel :

En cryptographie, on souhaite construire des groupes (au sens algébrique) tels que :

1. l'encodage des éléments soit efficace (peu de mémoire, peu de temps),
2. les **opérations sur ce groupe sont efficaces**,
3. la taille du groupe peut être grande et « diversifiée »,
4. le problème du logarithme discret est difficile dans ce groupe.

On a besoin de **formules algébriques** simples pour calculer la somme $P \oplus Q$.

1er cas. Si $P = \mathcal{O}$ ou $Q = \mathcal{O}$ ou $P = -Q$, on a le résultat sans calcul :

$$P \oplus \mathcal{O} = P \quad \mathcal{O} \oplus Q = Q \quad P \oplus -P = \mathcal{O}$$

Autres cas. Sinon, notons $P = (x_1, y_1)$ et $Q = (x_2, y_2)$, et $R = P \oplus Q = (x_3, y_3)$.

- ▶ Si $P \neq Q$: la droite passant par P et Q coupe la courbe en $-R$. Elle a donc pour équation $y = ux + v$, où

$$u = \frac{y_2 - y_1}{x_2 - x_1} = \frac{-y_3 - y_1}{x_3 - x_1} \quad \text{et} \quad v = y_1 - ux_1.$$

Les trois abscisses x_1, x_2 et x_3 sont donc racines de $(ux + v)^2 = x^3 + ax + b$. Par conséquent, $x_1 + x_2 + x_3 = u^2$ (regarder le terme de degré 2). On déduit

$$\begin{cases} x_3 &= u^2 - x_1 - x_2 \\ y_3 &= -y_1 + u(x_1 - x_3) \end{cases} \quad (1)$$

- ▶ Si $P = Q$: on calcule l'équation $y = ux + v$ de la tangente en différentiant. On obtient

$$u = \frac{3x_1^2 + a}{2y_1}.$$

Puis la formule est identique à (1).

Remarque. Bien que l'équation d'une courbe elliptique soit de degré 3, il n'est pas nécessaire d'extraire de racine carrée/cubique pour calculer l'opération de groupe.

Théorème. L'ensemble des points $\mathcal{E}(\mathbb{R})$ (en incluant le point à l'infini \mathcal{O}), muni de la loi définie plus haut, forme un groupe commutatif de neutre \mathcal{O} .

Important. On peut facilement observer que la loi de groupe ainsi définie est commutative. En revanche, il est bien plus difficile de démontrer qu'elle est associative.

Courbes elliptiques sur les corps finis

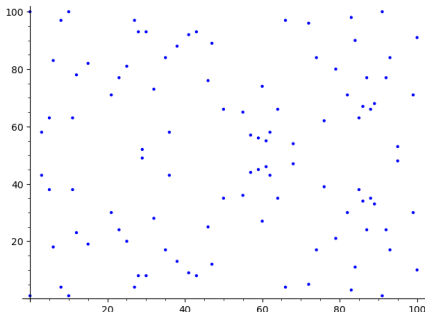
Sur le **corps fini** \mathbb{F}_q , avec $q = p^e$, on peut également définir une courbe elliptique

$$y^2 = x^3 + ax + b$$

avec $a, b \in \mathbb{F}_q$ tels que $4a^3 + 27b^2 \neq 0$ dans le corps \mathbb{F}_q . On évite donc $p = 2$ et $p = 3$.

Les formules d'additions sont identiques, et donnent un groupe abélien sur l'ensemble des **points rationnels** de la courbe, qui est noté $\mathcal{E}(\mathbb{F}_q)$.

Géométriquement, il est impossible d'observer la méthode des cordes et tangentes.



$$y = x^3 + x + 1 \text{ sur } \mathbb{F}_{101}$$

Soit $P \in \mathcal{E}(\mathbb{F}_q)$. On souhaite calculer $mP = \underbrace{P \oplus \cdots \oplus P}_{m \text{ fois}}$ pour $m \geq 1$.

On adapte la méthode *square-and-multiply* qui calcule une exponentiation rapide.

ALGORITHMHE « DOUBLE-AND-ADD »

Entrée : $P \in \mathcal{E}(\mathbb{F}_q)$ et $m \geq 1$

Sortie : mP

1. Calculer la décomposition de $m = \sum_{i=0}^k m_i 2^i$ en base 2.
2. Initialiser $Q \leftarrow P$ et $R \leftarrow \mathcal{O}$
3. **Pour** $i = 0, \dots, k$:
 - 3.1 **Si** $m_i = 1$, **alors** $R \leftarrow R \oplus Q$
Fin Si.
 - 3.2 $Q \leftarrow Q \oplus Q$
4. **Retourner** R .

ÉCHELLE DE MONTGOMERY

Entrée : $P \in \mathcal{E}(\mathbb{F}_q)$ et $m \geq 1$

Sortie : mP

1. Calculer la décomposition de $m = \sum_{i=0}^k m_i 2^i$ en base 2.
2. Initialiser $R_1 \leftarrow P$ et $R_0 \leftarrow \mathcal{O}$
3. **Pour** $i = k, \dots, 0$:
 - 3.1 **Si** $m_i = 0$, **alors**
 $R_1 \leftarrow R_0 \oplus R_1$
 $R_0 \leftarrow R_0 \oplus R_0$
 - 3.2 **Sinon,**
 $R_0 \leftarrow R_0 \oplus R_1$
 $R_1 \leftarrow R_0 \oplus R_0$
4. **Retourner** R_0 .

Remarque. Il existe une méthode *double-and-add-or-subtract* qui utilise le fait que $-P$ est très rapide à calculer. Par exemple, $1023P = 1024P - P = 2^{10}P - P$ nécessite 10 doubles et une soustraction, au lieu de 9 doubles et 9 additions par la méthode directe.

Quelques propriétés importantes de $\mathcal{E}(\mathbb{F}_q)$:


- ▶ Le groupe de points $\mathcal{E}(\mathbb{F}_q)$ est fini (car \mathbb{F}_q est fini) et son cardinal $\#\mathcal{E}(\mathbb{F}_q)$ vérifie

$$q + 1 - \sqrt{2q} \leq \#\mathcal{E}(\mathbb{F}_q) \leq q + 1 + \sqrt{2q}$$

C'est le **théorème de Hasse**.

- ▶ Il existe un algorithme polynomial en $\log(q)$ (dû à Schoof) pour calculer $\#\mathcal{E}(\mathbb{F}_q)$.
- ▶ Le groupe $\mathcal{E}(\mathbb{F}_q)$ est isomorphe à $\mathbb{Z}/n_1\mathbb{Z} \times \mathbb{Z}/n_2\mathbb{Z}$, avec $n_2 \mid n_1$. Il n'est donc pas toujours cyclique!

Pour **plus de détails**, voir

 *Courbes elliptiques : une présentation élémentaire pour la cryptographie*. Ph. Guillot. Lavoisier. **2010**.

1. Notions élémentaires de courbes elliptiques
2. Cryptosystème ElGamal
3. Résoudre le problème du logarithme discret

Soit G un groupe cyclique d'ordre n fini, et g un générateur de G .

Définition. Le problème du **logarithme discret** dans le groupe G est défini comme suit.

- **Données :** $y \in G$,
- **Objectif :** trouver $\ell \in \mathbb{Z}/n\mathbb{Z}$ tel que $g^\ell = y$.

Quels groupes ?

- groupe multiplicatif \mathbb{F}_q^\times ,
- groupe de points d'une courbe elliptique.

Sauf indication contraire, on va noter le groupe G multiplicativement.

Paramètres du système : un groupe G cyclique d'ordre fini n , et un générateur γ de G .

ELGAMAL : GÉNÉRATION DE CLÉS

1. Choisir aléatoirement $a \in \mathbb{Z}/n\mathbb{Z}$.
2. Calculer $\alpha = \gamma^a$
3. La **clé publique** est α , la **clé privée** est a .

ELGAMAL : CHIFFREMENT

L'espace des **clairs** est G . Si l'on souhaite chiffrer un message $\mu \in G$, alors

1. Tirer $k \in \mathbb{Z}/n\mathbb{Z}$ aléatoirement.
2. Calculer $\beta_1 = \gamma^k$ et $\beta_2 = \mu\alpha^k$.
3. Le chiffré est (β_1, β_2) .

ELGAMAL : DÉCHIFFREMENT

Pour **déchiffrer** (β_1, β_2) :

1. Avec la clé privée, on calcule $\lambda = \beta_1^a$.
2. On retourne β_2/λ .

CHIFFREMENT ELGAMAL : RÉSUMÉ

Paramètres du système : G un groupe cyclique d'ordre n et de générateur γ

Génération de clés :

- clé publique : $\alpha = \gamma^a \in G$
- clé privée : $a \in \mathbb{Z}/n\mathbb{Z}$.

Chiffré de $\mu \in G$: couple $(\beta_1 = \gamma^k, \beta_2 = \mu\alpha^k)$ où $k \in \mathbb{Z}/n\mathbb{Z}$ aléatoire.

Déchiffrement : on retourne β_2/β_1^a

Validité. Le chiffré est $(\beta_1 = \gamma^k, \beta_2 = \mu\alpha^k)$.

Si $\lambda = \beta_1^a$, alors on a bien

$$\beta_2\lambda^{-1} = \mu\alpha^k(\gamma^k)^{-a} = \mu\gamma^{ak}\gamma^{-ak} = \mu.$$

Exemple. Avec $G = \mathbb{F}_p^\times$ avec $p = 37$, de générateur $g = 2$, d'ordre $n = 36$.

Génération de clé. Alice choisit $a \in \mathbb{Z}/36\mathbb{Z}$, par exemple $a = 8$, et calcule $\alpha = g^a = 34$ (la clé publique).

Chiffrement. Pour chiffrer $\mu \in \mathbb{F}_p^\times$, Bob prend k aléatoire dans $\mathbb{Z}/36\mathbb{Z}$ et calcule

$$b_1 = g^k \quad \text{et} \quad b_2 = \mu \alpha^k.$$

Si $\mu = 21$ et $k = 11$, alors on obtient $b_1 = 13$ et $b_2 = 4$.

Déchiffrement. Pour déchiffrer (b_1, b_2) , Alice calcule $\lambda = b_1^a = 9$ et retrouve

$$\mu = b_2 / \lambda = 21.$$

Exemple. Avec des courbes elliptiques (notation **additive**).

Sur \mathbb{F}_{19} , avec la courbe $y^2 = x^3 + x + 1$. L'ensemble des points est :

$$\mathcal{E}(\mathbb{F}_{19}) = \{\mathcal{O}, (0, 1), (0, 18), (2, 7), (2, 12), (5, 6), (5, 13), (7, 3), (7, 16), (9, 6), (9, 13), \\ (10, 2), (10, 17), (13, 8), (13, 11), (14, 2), (14, 17), (15, 3), (15, 16), (16, 3), (16, 16)\}$$

On a $n = \#\mathcal{E}(\mathbb{F}_{19}) = 21$ qui est donc cyclique, de générateur $P = (16, 16)$ (par exemple).

Génération de clé. Alice choisit $a \in \mathbb{Z}/n\mathbb{Z}$, par exemple $a = 8$, et calcule $A = aP = (5, 6)$ (la clé publique).

Chiffrement. Pour chiffrer $M \in \mathcal{E}(\mathbb{F}_{19})$, Bob prend k aléatoire dans $\mathbb{Z}/21\mathbb{Z}$ et calcule

$$B_1 = kP \quad \text{et} \quad B_2 = M \oplus kA.$$

Si $M = (7, 3)$ et $k = 11$, alors on obtient $B_1 = (7, 16)$ et $B_2 = (2, 12)$.

Déchiffrement. Pour déchiffrer (B_1, B_2) , Alice calcule $L = aB_1 = (0, 1)$ et retrouve

$$M = B_2 \oplus (-L) = (2, 12) \oplus (0, 18) = (7, 3)$$

Remarque : il existe des méthodes pour compresser les points d'une courbe elliptique afin de diminuer l'expansion du chiffré par rapport au clair.

On réduit la sécurité d'ElGamal au **problème de Diffie–Hellman calculatoire** dans un groupe G cyclique d'ordre n .

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 $g \in G$ et $a, b \in \{0, \dots, n-1\}$

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

Définition. Le problème **ElGamal** dans G , noté EG, est le problème suivant :

Données : (g, g^a, g^k, mg^{ak}) où $g, m \in G$ et $a, k \in \mathbb{Z}/n\mathbb{Z}$

Objectif : calculer $m \in G$

Théorème. Les deux problèmes EG et CDH sont équivalents.

Preuve de l'équivalence entre CDH et EG.

1. Soit A_{CDH} un algorithme qui résout CDH, et (g, g^a, g^k, mg^{ak}) une instance du problème EG. On cherche à retrouver m .

Pour cela on exécute $A_{\text{CDH}}(g, g^a, g^k)$ et obtient g^{ak} .

Puis, grâce à mg^{ak} et g^{ak} , on obtient m par une simple division dans le groupe \mathbb{G} .

2. Soit A_{EG} un algorithme qui résout EG, et (g, g^a, g^b) une instance du problème CDH. On cherche à calculer g^{ab} .

Pour cela, on engendre un élément aléatoire $u \in \mathbb{G}$ et on exécute $A_{\text{EG}}(g, g^a, g^c, u)$. Par définition, cet algorithme nous retourne un élément $m \in \mathbb{G}$ tel que $u = mg^{ab}$. Il ne reste plus qu'à diviser u par m pour obtenir g^{ab} .

Théorème. Les deux problèmes EG et CDH sont équivalents.

Corollaire. Comme CDH se réduit au problème du logarithme discret, on en déduit que EG s'y réduit aussi.

Voyons maintenant quelques éléments attestant la difficulté du problème du logarithme discret.

1. Notions élémentaires de courbes elliptiques
2. Cryptosystème ElGamal
3. Résoudre le problème du logarithme discret

Rappelons que, pour que le problème CDH soit difficile, il faut que le problème du logarithme discret soit difficile.

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 $x \in \{0, \dots, n-1\}$

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

Objectif. Pour $y \in G$, on cherche $\ell \in \mathbb{Z}/n\mathbb{Z}$ tel que $g^\ell = y$.

Méthode exhaustive. Tester tous les g^i coûte $O(n)$.

Méthode « pas de bébé - pas de géant » (*baby-step-giant-step*), initiée par Shanks dans :

 *Class number, a theory of factorization and genera*. D. Shanks. Proc. Symp. Pure Math.. 1971.

Idée. Soit $m \geq \sqrt{n}$.

1. On calcule la liste

$$L = [g^0, \dots, g^{m-1}]$$

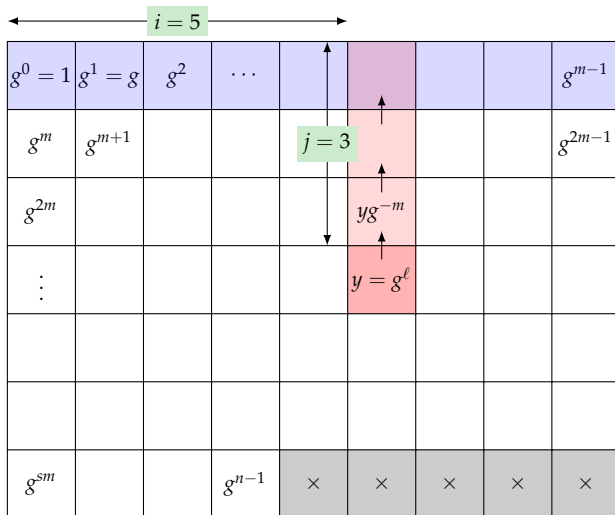
2. On cherche un j tel que yg^{-jm} est dans la liste L .

Si c'est le cas, on a alors :

$$yg^{-jm} = g^i \iff y = g^{i+jm}$$

Remarque. Un tel couple (i, j) existe toujours. On peut s'en apercevoir en effectuant la division euclidienne de ℓ par m .

Méthode « pas de bébé - pas de géant »



Sur l'exemple, on obtient $\ell = i + mj = 5 + 9 \times 3 = 32$

Dans l'algorithme qui suit, on va utiliser comme structure de données le **dictionnaire**.

$$D = \left\{ \begin{array}{ll} \text{key}_1 & \rightarrow \text{value}_1 \\ \vdots & \vdots \\ \text{key}_s & \rightarrow \text{value}_s \end{array} \right.$$

Grâce à une table de hachage : ajout d'un élément, et test d'appartenance en $O(1)$.

MÉTHODE « PAS DE BÉBÉ – PAS DE GÉANT » (SHANKS)

Entrée : un élément $y \in \mathbb{G}$, un générateur g de \mathbb{G} et l'ordre n de \mathbb{G}

Sortie : $\ell \in \mathbb{Z}/n\mathbb{Z}$ tel que $g^\ell = y$

1. Calculer $m = \lceil \sqrt{n} \rceil$.
2. Calculer un dictionnaire $D = \{(g^0 \rightarrow 0), \dots, (g^{m-1} \rightarrow m-1)\}$.
3. Initialiser $x \leftarrow y$ et $j \leftarrow 0$
4. **Tant que** x n'est pas une clé de D , **faire :**
 - $x \leftarrow x \times g^{-m}$
 - $j \leftarrow j + 1$
5. **Retourner** $jm + D[x]$.

Méthode « pas de bébé - pas de géant » : exemple 1

Dans le **groupe multiplicatif** \mathbb{F}_q^\times avec $q = 97$.

L'élément $g = 5$ engendre \mathbb{F}_{97}^\times .

1. On a $m = \lceil \sqrt{97} \rceil = 10$.
2. On construit le dictionnaire $D = \{g^i \rightarrow i\}_{i \leq m-1}$. Un dictionnaire n'est pas nécessairement « ordonné ». Ici, **sage** donne :

$$D = \{1 \rightarrow 0, 5 \rightarrow 1, 6 \rightarrow 8, 8 \rightarrow 6, 43 \rightarrow 4, 40 \rightarrow 7, 21 \rightarrow 5, 25 \rightarrow 2, 28 \rightarrow 3, 30 \rightarrow 9\}$$

3. Pour $y = 18$, on obtient ensuite

j	yg^{-jm}	Test($yg^{-jm} \in D$)
0	18	×
1	4	×
2	44	×
3	96	×
4	86	×
5	73	×
6	27	×
7	6	✓ ($D[6] = 8$)

4. Donc $\log_g(y) = i + jm = 8 + 7 \times 10 = 78$.

Méthode « pas de bébé - pas de géant » : exemple 2

Dans le **groupe de points de la courbe elliptique** (noté additivement)

$$\mathcal{E} : y^2 = x^3 + x + 1,$$

définie sur \mathbb{F}_q avec $q = 101$. Ce groupe est cyclique, isomorphe à $\mathbb{Z}/105\mathbb{Z}$, et admet comme générateur le point $P = (41, 92) \in \mathcal{E}(\mathbb{F}_q)$.

1. On a $m = \lceil \sqrt{105} \rceil = 11$.
2. On construit le dictionnaire $D = \{iP \rightarrow i\}_{i \leq m-1}$

$$D = \{ \mathcal{O} \rightarrow 0, \quad (41, 92) \rightarrow 1, \quad (64, 35) \rightarrow 2, \quad (38, 88) \rightarrow 8, \\ (43, 93) \rightarrow 3, \quad (35, 17) \rightarrow 9, \quad (93, 84) \rightarrow 4, \quad (55, 36) \rightarrow 10, \\ (74, 84) \rightarrow 5, \quad (29, 52) \rightarrow 6, \quad (87, 24) \rightarrow 7 \}$$

3. Pour un point $Q = (76, 39) \in \mathcal{E}(\mathbb{F}_q)$, on obtient ensuite

j	$Q - jmP$	Test($Q - jmP \in D$)
0	(76, 39)	×
1	(82, 71)	×
2	(99, 30)	×
3	(30, 93)	×
4	(46, 25)	×
5	(55, 36)	✓ ($D[(55, 36)] = 10$)

4. Donc $\log_P(Q) = i + jm = 10 + 5 \times 11 = 65$.

MÉTHODE « PAS DE BÉBÉ – PAS DE GÉANT » (SHANKS)

Entrée : un élément $y \in \mathbb{G}$, un générateur g de \mathbb{G} et l'ordre n de \mathbb{G}

Sortie : $\ell \in \mathbb{Z}/n\mathbb{Z}$ tel que $g^\ell = y$

1. Calculer $m = \lceil \sqrt{n} \rceil$.
2. Calculer un dictionnaire $D = \{(g^0 \rightarrow 0), \dots, (g^{m-1} \rightarrow m-1)\}$.
3. Initialiser $x \leftarrow y$ et $j \leftarrow 0$
4. **Tant que** x n'est pas une clé de D , **faire :**
 - $x \leftarrow x \times g^{-m}$
 - $j \leftarrow j + 1$
5. **Retourner** $jm + D[x]$.

Complexité pour les étapes :

- étapes 1, 3, et 5 : $O(1)$
- étape 2 : $O(\sqrt{n})$
- étape 4 : $O(\sqrt{n})$

\implies Complexité totale en $O(\sqrt{n})$

Lorsque $n = \prod p_i^{e_i}$ est composé, G admet des sous-groupes cycliques propres.

$$G \simeq \mathbb{Z}/p_1^{e_1}\mathbb{Z} \times \cdots \times \mathbb{Z}/p_k^{e_k}\mathbb{Z}$$

Algorithme de **Pohlig–Hellman** pour calculer x tel que $y = g^x$ dans G .

1. On obtient le logarithme discret dans les sous-groupes $\mathbb{Z}/p_1^{e_1}\mathbb{Z}$:
 - on calcule un logarithme dans \mathbb{F}_{p_i} (avec *pas de bébé - pas de géant*), puis dans $\mathbb{Z}/p_1^{e_1}\mathbb{Z}$ grâce à un **relèvement de Hensel**,
 - on obtient finalement $x_i = x \pmod{p_i^{e_i}}$.
2. On reconstruit x à partir des x_i par le théorème des restes chinois.

La complexité est en $O^{\sim}(\max\{\sqrt{p_i}\})$. Il faut donc choisir un groupe G qui admet un **grand sous-groupe cyclique d'ordre premier**.

Théorème (Shoup 1997, énoncé informel). La complexité du logarithme discret dans un groupe générique (= sans autre structure connue) d'ordre $n = p^t s$, où p est premier et p et s sont premiers entre eux, est en $\Omega(\sqrt{p})$.

On conjecture que le groupe des points d'une courbe elliptique est un groupe générique (mises à part des familles de courbes particulières).

Dans les groupes multiplicatifs de corps finis \mathbb{F}_q^\times , on peut faire mieux.

- ▶ La méthode de **calcul d'indice** exploite les relations linéaires entre logarithmes de petits nombres premiers. Sa complexité est en

$$\exp((\log q)^{1/2+o(1)}).$$



Fast, rigorous factorization and discrete logarithm algorithms. Pomerance. Discrete algorithms and complexity. **1987**.

- ▶ Grâce aux cribles par corps de nombres et corps de fonctions, on peut réduire la complexité à

$$\exp((\log q)^{1/3+o(1)})$$



Using number fields to compute logarithms in finite fields. Schirokauer. Math. Comp. **2000**.

- ▶ Pour des valeurs de q particulières, il existe des algorithmes plus efficaces. Notamment, si $q = p^\ell$ avec p très petit (typiquement $p \in \{2, 3\}$), alors il existe un **algorithme quasi-polynomial**, *i.e.* en temps

$$\exp((\log q)^{o(1)})$$



A Heuristic Quasi-Polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic. Barbuslecu, Gaudry, Joux, Thomé. EUROCRYPT. **2014**.


Évolution des résolutions du logarithme discret pour p petit :

année	q	auteurs	heures CPU
1992	2^{401}	Gordon, McCurley	114 000
2005	2^{613}	Joux, Lercier	26 000
2013	2^{1778}	Joux	220
2013	2^{6168}	Joux	550
2014	2^{9234}	Granger, Kleinjung, Zumbragel	398 000

En **grande** caractéristique :

- \mathbb{F}_{p^2} avec p premier de 160 chiffres (532 bits)
- auteurs : Barbulescu, Gaudry, Guillevic, Morain (2014) : 68 jours CPU + 30 heures GPU

Un **état de l'art** assez récent :

 *The Past, evolving Present and Future of Discrete Logarithm*. Joux, Odlyzko, Pierrot.

<https://members.loria.fr/CPierrot/papers/DlogSurvey.pdf>. 2014.

Conclusion.

- Pour le **groupe multiplicatif d'un corps fini** \mathbb{F}_q de grande caractéristique :
 - Il faut que $q - 1$ admette un grand facteur premier, de taille comparable à $q - 1$
 - En pratique, $\log_2(q) \geq 2048$, voir 3072, est recommandé.
 - Taille de clé publique et privée ≥ 2048 bits.
 - Taille de chiffrés deux fois plus gros.
- Pour le groupe des points d'une **courbe elliptique** avec un grand sous-groupe cyclique :
 - On peut s'autoriser des courbes sur \mathbb{F}_q avec $q \simeq 2^{256}$ de grande caractéristique.
 - Taille de clé publique et privée ≥ 256 bits.
 - Calculs \simeq aussi efficaces que sur \mathbb{F}_q^\times avec $\log(q) = 2048$.
 - Il est recommandé d'utiliser par exemple **Curve25519** (standardisée), offrant une sécurité proche de 128 bits, définie sur $\mathbb{F}_{2^{255}-19}$ par $y^2 = x^3 + 486662x^2 + x$.
 - Par ailleurs, sur cette courbe les calculs peuvent être accélérés (forme de Montgomery).

Questions ?