

Algorithmes pour l'arithmétique II

Cours 9

Julien Lavauzelle

Université Paris 8

Master 2 ACC – Algorithmes pour l'arithmétique

28/11/2023

Questions?

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

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

Soit G un groupe cyclique (noté multiplicativement) d'ordre fini n , 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 $g^\ell = y$.

Soit G un groupe cyclique (noté multiplicativement) d'ordre fini n , 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 $g^\ell = y$.

Application principale : en cryptographie. La sécurité de nombreux systèmes cryptographiques repose sur la **difficulté** de ce problème : ElGamal, DSA.

Soit G un groupe cyclique (noté multiplicativement) d'ordre fini n , 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 $g^\ell = y$.

Application principale : en cryptographie. La sécurité de nombreux systèmes cryptographiques repose sur la **difficulté** de ce problème : ElGamal, DSA.

Quels groupes ?

Quelle mesure de complexité ? Nombre d'opérations élémentaires dans G : multiplications, inversions, carrés.

Soit G un groupe cyclique (noté multiplicativement) d'ordre fini n , 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 $g^\ell = y$.

Application principale : en cryptographie. La sécurité de nombreux systèmes cryptographiques repose sur la **difficulté** de ce problème : ElGamal, DSA.

Quels groupes ?

- groupe multiplicatif \mathbb{F}_q^\times ,

Quelle mesure de complexité ? Nombre d'opérations élémentaires dans G : multiplications, inversions, carrés.

Soit G un groupe cyclique (noté multiplicativement) d'ordre fini n , 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 $g^\ell = y$.

Application principale : en cryptographie. La sécurité de nombreux systèmes cryptographiques repose sur la **difficulté** de ce problème : ElGamal, DSA.

Quels groupes ?

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

Quelle mesure de complexité ? Nombre d'opérations élémentaires dans G : multiplications, inversions, carrés.

Soit G un groupe cyclique (noté multiplicativement) d'ordre fini n , 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 $g^\ell = y$.

Application principale : en cryptographie. La sécurité de nombreux systèmes cryptographiques repose sur la **difficulté** de ce problème : ElGamal, DSA.

Quels groupes ?

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

Quelle mesure de complexité ? Nombre d'opérations élémentaires dans G : multiplications, inversions, carrés.

1. Logarithme discret dans un groupe générique

Pas de bébé – pas de géant

Algorithme de Pohlig-Hellman

Autres algorithmes

2. Borne inférieure de complexité

1. Logarithme discret dans un groupe générique

Pas de bébé – pas de géant

Algorithme de Pohlig-Hellman

Autres algorithmes

2. Borne inférieure de complexité

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

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

Méthode exhaustive. Tester tous les g^i coûte


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

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

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

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


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.

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

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

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 $m \geq \sqrt{n}$.

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

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

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

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

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

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

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 .

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

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

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 trouvé $i \leq m - 1$ tel que

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

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

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

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 trouvé $i \leq m - 1$ tel que

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

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 »

$g^0 = 1$	$g^1 = g$	g^2	\dots					g^{m-1}
g^m	g^{m+1}							g^{2m-1}
g^{2m}								
\vdots								
g^{sm}			g^{n-1}	×	×	×	×	×

Méthode « pas de bébé - pas de géant »

$g^0 = 1$	$g^1 = g$	g^2	\dots					g^{m-1}
g^m	g^{m+1}							g^{2m-1}
g^{2m}								
\vdots					$y = g^\ell$			
g^{sm}			g^{n-1}	×	×	×	×	×

Méthode « pas de bébé - pas de géant »

$g^0 = 1$	$g^1 = g$	g^2	\dots					g^{m-1}
g^m	g^{m+1}							g^{2m-1}
g^{2m}								
\vdots					$y = g^\ell$			
g^{sm}			g^{n-1}	\times	\times	\times	\times	\times

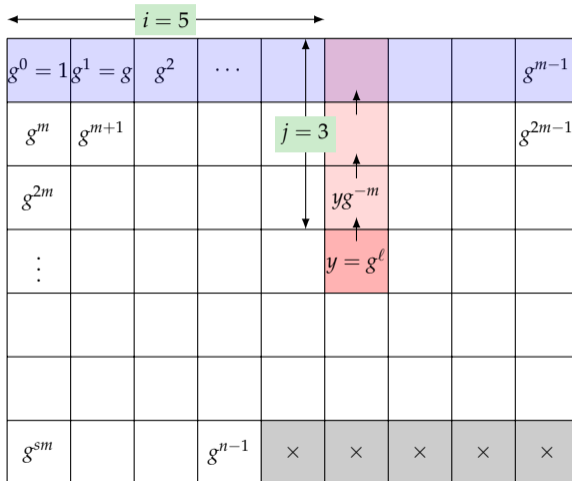
Méthode « pas de bébé - pas de géant »

$g^0 = 1$	$g^1 = g$	g^2	\dots					g^{m-1}
g^m	g^{m+1}							g^{2m-1}
g^{2m}					yg^{-m}			
\vdots					\uparrow $y = g^\ell$			
g^{sm}			g^{n-1}	\times	\times	\times	\times	\times

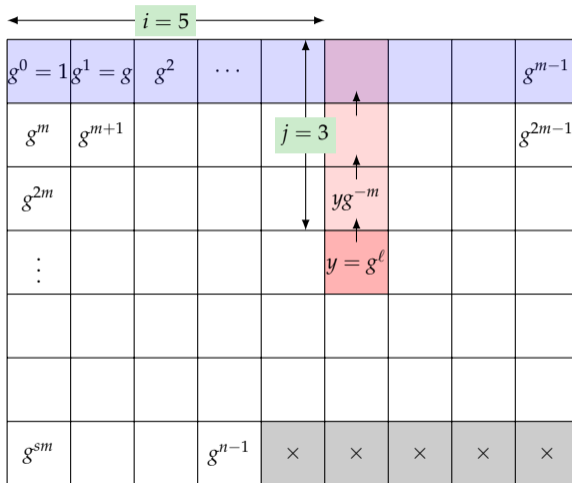
Méthode « pas de bébé - pas de géant »

$g^0 = 1$	$g^1 = g$	g^2	\dots					g^{m-1}
g^m	g^{m+1}							g^{2m-1}
g^{2m}								
\vdots								
g^{sm}			g^{n-1}	×	×	×	×	×

Méthode « pas de bébé - pas de géant »



Méthode « pas de bébé - pas de géant »



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

Méthode « pas de bébé - pas de géant » : algorithme

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** : l'ajout d'un élément et le test d'appartenance sont en $O(1)$.

Méthode « pas de bébé - pas de géant » : algorithme

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

$$D = \begin{cases} \text{key}_1 & \rightarrow & \text{value}_1 \\ \vdots & & \vdots \\ \text{key}_s & \rightarrow & \text{value}_s \end{cases}$$

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

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

Entrée : un élément $y \in G$, un générateur g de G et l'ordre n de 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** $jm + D[x]$.

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

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

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

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

1. On a $m = \lceil \sqrt{97} \rceil = 10$.

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 ».

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 :

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

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($yg^{-jm} \in D$)

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	×

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	×

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	×

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	×

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	✓ ($D[6] = 8$)

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)

$$E : y^2 = x^3 + x + 1,$$

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 $P = [41 : 92 : 1] \in E(\mathbb{F}_q)$.

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

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

$$E : y^2 = x^3 + x + 1,$$

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 $P = [41 : 92 : 1] \in E(\mathbb{F}_q)$.

1. On a $m = \lceil \sqrt{105} \rceil = 11$.

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

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

$$E: y^2 = x^3 + x + 1,$$

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 $P = [41 : 92 : 1] \in E(\mathbb{F}_q)$.

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

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

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

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

$$E: y^2 = x^3 + x + 1,$$

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 $P = [41 : 92 : 1] \in E(\mathbb{F}_q)$.

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

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

3. Pour un point $Q = [76 : 39 : 1] \in E(\mathbb{F}_q)$ dont on cherche le logarithme en base P , on obtient ensuite

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

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

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

$$E: y^2 = x^3 + x + 1,$$

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 $P = [41 : 92 : 1] \in E(\mathbb{F}_q)$.

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

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

3. Pour un point $Q = [76 : 39 : 1] \in E(\mathbb{F}_q)$ dont on cherche le logarithme en base P , on obtient ensuite

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

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

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

Entrée : un élément $y \in \mathbf{G}$, un générateur g de \mathbf{G} et l'ordre n de \mathbf{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 » (SHANKS)

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

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

Entrée : un élément $y \in G$, un générateur g de G et l'ordre n de 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 : $O(1)$

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

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

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

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

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

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

1. Logarithme discret dans un groupe générique

Pas de bébé – pas de géant

Algorithme de Pohlig-Hellman

Autres algorithmes

2. Borne inférieure de complexité

Fait. L'algorithme *baby-step-giant-step* permet de calculer le logarithme discret d'un élément $y \in \mathbb{G}$ en temps $O(\sqrt{n})$ où $n = |\mathbb{G}|$.

Fait. L'algorithme *baby-step-giant-step* permet de calculer le logarithme discret d'un élément $y \in \mathbf{G}$ en temps $O(\sqrt{n})$ où $n = |\mathbf{G}|$.

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

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

Fait. L'algorithme *baby-step-giant-step* permet de calculer le logarithme discret d'un élément $y \in \mathbf{G}$ en temps $O(\sqrt{n})$ où $n = |\mathbf{G}|$.

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

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

Question. Peut-on mettre à profit cette structure pour améliorer la complexité ?

Fait. L'algorithme *baby-step-giant-step* permet de calculer le logarithme discret d'un élément $y \in \mathbf{G}$ en temps $O(\sqrt{n})$ où $n = |\mathbf{G}|$.

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

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

Question. Peut-on mettre à profit cette structure pour améliorer la complexité ?

Deux cas à étudier :

1. $n = st$, où s et t sont premiers entre eux,
2. $n = p^e$ où p est premier et $e \geq 2$.

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in \mathbb{G}$ dont on cherche $\ell = \log_g(y)$.

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in \mathbb{G}$ dont on cherche $\ell = \log_g(y)$.

On va utiliser

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in \mathbf{G}$ dont on cherche $\ell = \log_g(y)$.

On va utiliser le théorème des restes chinois. L'application

$$\begin{aligned} \phi : \mathbb{Z}/n\mathbb{Z} &\rightarrow \mathbb{Z}/s\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z} \\ x &\mapsto (x \bmod s, x \bmod t) \end{aligned}$$

est un isomorphisme d'anneaux que l'on sait évaluer et inverser efficacement.

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in \mathbf{G}$ dont on cherche $\ell = \log_g(y)$.

On va utiliser le théorème des restes chinois. L'application

$$\begin{aligned} \phi : \mathbb{Z}/n\mathbb{Z} &\rightarrow \mathbb{Z}/s\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z} \\ x &\mapsto (x \bmod s, x \bmod t) \end{aligned}$$

est un isomorphisme d'anneaux que l'on sait évaluer et inverser efficacement.

Pour obtenir le logarithme $\ell \in \mathbb{Z}/n\mathbb{Z}$, on se ramène donc à calculer $(\ell \bmod s)$ et $(\ell \bmod t)$.

Question intermédiaire. À partir de l'instance (y, g) , construire un couple (y', g') dans un groupe \mathbf{G}' tel que $\log_{g'}(y') = \ell \bmod s$.

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in G$ dont on cherche $\ell = \log_g(y)$.

On va utiliser le théorème des restes chinois. L'application

$$\begin{aligned} \phi : \mathbb{Z}/n\mathbb{Z} &\rightarrow \mathbb{Z}/s\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z} \\ x &\mapsto (x \bmod s, x \bmod t) \end{aligned}$$

est un isomorphisme d'anneaux que l'on sait évaluer et inverser efficacement.

Pour obtenir le logarithme $\ell \in \mathbb{Z}/n\mathbb{Z}$, on se ramène donc à calculer $(\ell \bmod s)$ et $(\ell \bmod t)$.

Question intermédiaire. À partir de l'instance (y, g) , construire un couple (y', g') dans un groupe G' tel que $\log_{g'}(y') = \ell \bmod s$.

Idée. On construit

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in G$ dont on cherche $\ell = \log_g(y)$.

On va utiliser le théorème des restes chinois. L'application

$$\begin{aligned} \phi : \mathbb{Z}/n\mathbb{Z} &\rightarrow \mathbb{Z}/s\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z} \\ x &\mapsto (x \bmod s, x \bmod t) \end{aligned}$$

est un isomorphisme d'anneaux que l'on sait évaluer et inverser efficacement.

Pour obtenir le logarithme $\ell \in \mathbb{Z}/n\mathbb{Z}$, on se ramène donc à calculer $(\ell \bmod s)$ et $(\ell \bmod t)$.

Question intermédiaire. À partir de l'instance (y, g) , construire un couple (y', g') dans un groupe G' tel que $\log_{g'}(y') = \ell \bmod s$.

Idée. On construit $g' = g^t$ et $y' = y^t$. Alors, g' engendre un sous-groupe G' de G d'ordre s , et on a

$$y' = y^t = g^{\ell t} = (g')^\ell.$$

Premier cas. Si $n = st$ avec s et t premiers entre eux.

Soit $y \in G$ dont on cherche $\ell = \log_g(y)$.

On va utiliser le théorème des restes chinois. L'application

$$\begin{aligned} \phi : \mathbb{Z}/n\mathbb{Z} &\rightarrow \mathbb{Z}/s\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z} \\ x &\mapsto (x \bmod s, x \bmod t) \end{aligned}$$

est un isomorphisme d'anneaux que l'on sait évaluer et inverser efficacement.

Pour obtenir le logarithme $\ell \in \mathbb{Z}/n\mathbb{Z}$, on se ramène donc à calculer $(\ell \bmod s)$ et $(\ell \bmod t)$.

Question intermédiaire. À partir de l'instance (y, g) , construire un couple (y', g') dans un groupe G' tel que $\log_{g'}(y') = \ell \bmod s$.

Idée. On construit $g' = g^t$ et $y' = y^t$. Alors, g' engendre un sous-groupe G' de G d'ordre s , et on a

$$y' = y^t = g^{\ell t} = (g')^\ell.$$

Donc, y' appartient à ce sous-groupe G' , et $\log_{g'}(y') = \ell \bmod s$.

On suppose que l'on dispose d'un algorithme L qui calcule le logarithme discret dans un groupe fini quelconque. L'algorithme L peut être :

- l'algorithme de Pohlig-Hellman lui-même (appel récursif),
- un autre algorithme, comme *baby-step-giant-step* par exemple.

On suppose que l'on dispose d'un algorithme L qui calcule le logarithme discret dans un groupe fini quelconque. L'algorithme L peut être :

- l'algorithme de Pohlig-Hellman lui-même (appel récursif),
- un autre algorithme, comme *baby-step-giant-step* par exemple.

ALGORITHME DE POHLIG-HELLMAN (ÉTAPE « PREMIERS ENTRE EUX »)

Entrée : un élément $y \in G$ où $|G| = n = st$ et $\text{pgcd}(s, t) = 1$, et un générateur g de G

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

1. Calculer y^s, y^t, g^s et g^t .
2. À l'aide de l'algorithme L , calculer $\ell_s = \log_{g^t}(y^t)$ et $\ell_t = \log_{g^s}(y^s)$.
3. Calculer les coefficients de Bezout a, b tels que $as + bt = 1$
4. Effectuer la reconstruction par restes chinois :

$$\ell = as\ell_t + bt\ell_s$$

5. **Retourner** $\ell \bmod n$

On suppose que l'on dispose d'un algorithme L qui calcule le logarithme discret dans un groupe fini quelconque. L'algorithme L peut être :

- l'algorithme de Pohlig-Hellman lui-même (appel récursif),
- un autre algorithme, comme *baby-step-giant-step* par exemple.

ALGORITHME DE POHLIG-HELLMAN (ÉTAPE « PREMIERS ENTRE EUX »)

Entrée : un élément $y \in G$ où $|G| = n = st$ et $\text{pgcd}(s, t) = 1$, et un générateur g de G

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

1. Calculer y^s, y^t, g^s et g^t .
2. À l'aide de l'algorithme L , calculer $\ell_s = \log_{g^t}(y^t)$ et $\ell_t = \log_{g^s}(y^s)$.
3. Calculer les coefficients de Bezout a, b tels que $as + bt = 1$
4. Effectuer la reconstruction par restes chinois :

$$\ell = as\ell_t + bt\ell_s$$

5. **Retourner** $\ell \bmod n$

Remarque : il faut connaître la factorisation de n ...

Exemple. On prend la courbe elliptique E définie sur \mathbb{F}_q , $q = 97$, par l'équation

$$y^2 = x^3 - 9x + 33$$

Le groupe $E(\mathbb{F}_q)$ est cyclique, de cardinal $n = 115$, et $P = [91 : 29 : 1]$ en est un générateur.

Exemple. On prend la courbe elliptique E définie sur \mathbb{F}_q , $q = 97$, par l'équation

$$y^2 = x^3 - 9x + 33$$

Le groupe $E(\mathbb{F}_q)$ est cyclique, de cardinal $n = 115$, et $P = [91 : 29 : 1]$ en est un générateur.

1. On a choisi $Q = [9 : 14 : 1] \in E(\mathbb{F}_q)$

Exemple. On prend la courbe elliptique E définie sur \mathbb{F}_q , $q = 97$, par l'équation

$$y^2 = x^3 - 9x + 33$$

Le groupe $E(\mathbb{F}_q)$ est cyclique, de cardinal $n = 115$, et $P = [91 : 29 : 1]$ en est un générateur.

1. On a choisi $Q = [9 : 14 : 1] \in E(\mathbb{F}_q)$
2. On a $n = s \times t$ où $s = 5$ et $t = 23$.

Exemple. On prend la courbe elliptique E définie sur \mathbb{F}_q , $q = 97$, par l'équation

$$y^2 = x^3 - 9x + 33$$

Le groupe $E(\mathbb{F}_q)$ est cyclique, de cardinal $n = 115$, et $P = [91 : 29 : 1]$ en est un générateur.

1. On a choisi $Q = [9 : 14 : 1] \in E(\mathbb{F}_q)$
2. On a $n = s \times t$ où $s = 5$ et $t = 23$.
3. Puis, on obtient

sQ	sP	ℓ_t		tQ	tP	ℓ_s
$(54 : 68 : 1)$	$(71 : 32 : 1)$	13		$(48 : 96 : 1)$	$(48 : 96 : 1)$	1

Exemple. On prend la courbe elliptique E définie sur \mathbb{F}_q , $q = 97$, par l'équation

$$y^2 = x^3 - 9x + 33$$

Le groupe $E(\mathbb{F}_q)$ est cyclique, de cardinal $n = 115$, et $P = [91 : 29 : 1]$ en est un générateur.

1. On a choisi $Q = [9 : 14 : 1] \in E(\mathbb{F}_q)$
2. On a $n = s \times t$ où $s = 5$ et $t = 23$.
3. Puis, on obtient

$$\begin{array}{ccc|ccc} sQ & sP & \ell_t & tQ & tP & \ell_s \\ \hline (54 : 68 : 1) & (71 : 32 : 1) & 13 & (48 : 96 : 1) & (48 : 96 : 1) & 1 \end{array}$$

4. Les coefficients de Bezout de s et t sont $-9s + 2t = 1$

Exemple. On prend la courbe elliptique E définie sur \mathbb{F}_q , $q = 97$, par l'équation

$$y^2 = x^3 - 9x + 33$$

Le groupe $E(\mathbb{F}_q)$ est cyclique, de cardinal $n = 115$, et $P = [91 : 29 : 1]$ en est un générateur.

1. On a choisi $Q = [9 : 14 : 1] \in E(\mathbb{F}_q)$
2. On a $n = s \times t$ où $s = 5$ et $t = 23$.
3. Puis, on obtient

$$\begin{array}{ccc|ccc} sQ & sP & \ell_t & tQ & tP & \ell_s \\ \hline (54 : 68 : 1) & (71 : 32 : 1) & 13 & (48 : 96 : 1) & (48 : 96 : 1) & 1 \end{array}$$

4. Les coefficients de Bezout de s et t sont $-9s + 2t = 1$
5. Donc on peut reconstruire (théorème des restes chinois) :

$$\ell = -9 \times 5 \times 13 + 2 \times 23 \times 1 \equiv 36 \pmod{115}$$

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois.

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e - 1$.

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e - 1$.

Si g engendre G , le sous-groupe G_i est engendré par

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e - 1$.

Si g engendre G , le sous-groupe G_i est engendré par $g^{p^{e-i}}$.

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e - 1$.

Si g engendre G , le sous-groupe G_i est engendré par $g^{p^{e-i}}$.

Question. Comment utiliser ces sous-groupes ?

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e - 1$.

Si g engendre G , le sous-groupe G_i est engendré par $g^{p^{e-i}}$.

Question. Comment utiliser ces sous-groupes ?

Écrivons $\ell = \log_g(y)$ en base p :

$$\ell = \ell_0 + \ell_1 p + \cdots + \ell_{e-1} p^{e-1}$$

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e-1$.

Si g engendre G , le sous-groupe G_i est engendré par $g^{p^{e-i}}$.

Question. Comment utiliser ces sous-groupes ?

Écrivons $\ell = \log_g(y)$ en base p :

$$\ell = \ell_0 + \ell_1 p + \dots + \ell_{e-1} p^{e-1}$$

On a

$$y^{p^{e-1}} = g^{\ell_0 p^{e-1} + \ell_1 p^e + \dots + \ell_{e-1} p^{2e-2}} = g^{\ell_0 p^{e-1} + p^e(\ell_1 + \dots + \ell_{e-1} p^{e-2})} = g^{\ell_0 p^{e-1}}$$

Donc, l'élément ℓ_0 est le logarithme de $y^{p^{e-1}}$ en base $g^{p^{e-1}}$ dans G_1 .

Second cas. Si $n = p^e$ avec p premier et $e \geq 2$.

On n'a plus l'isomorphisme des restes chinois. Néanmoins, $G \simeq \mathbb{Z}/p^e\mathbb{Z}$ admet des sous-groupes cycliques propres G_i de la forme $\mathbb{Z}/p^i\mathbb{Z}$ où $1 \leq i \leq e-1$.

Si g engendre G , le sous-groupe G_i est engendré par $g^{p^{e-i}}$.

Question. Comment utiliser ces sous-groupes ?

Écrivons $\ell = \log_g(y)$ en base p :

$$\ell = \ell_0 + \ell_1 p + \dots + \ell_{e-1} p^{e-1}$$

On a

$$y^{p^{e-1}} = g^{\ell_0 p^{e-1} + \ell_1 p^e + \dots + \ell_{e-1} p^{2e-2}} = g^{\ell_0 p^{e-1} + p^e(\ell_1 + \dots + \ell_{e-1} p^{e-2})} = g^{\ell_0 p^{e-1}}$$

Donc, l'élément ℓ_0 est le logarithme de $y^{p^{e-1}}$ en base $g^{p^{e-1}}$ dans G_1 .

Question. Comment poursuivre pour obtenir ℓ_1 ?

Question. Comment poursuivre pour obtenir ℓ_1 ?

Question. Comment poursuivre pour obtenir ℓ_1 ?

On itère le procédé avec :

$$\frac{y}{g^{\ell_0}} = (g^p)^{\ell_1 + \ell_2 p + \dots + \ell_{p-1} p^{e-2}} .$$

Question. Comment poursuivre pour obtenir ℓ_1 ?

On itère le procédé avec :

$$\frac{y}{g^{\ell_0}} = (g^p)^{\ell_1 + \ell_2 p + \dots + \ell_{p-1} p^{e-2}}.$$

Puis

$$\left(\frac{y}{g^{\ell_0}} \right)^{p^{e-2}} = g^{\ell_1 p^{e-1}}$$

Question. Comment poursuivre pour obtenir ℓ_1 ?

On itère le procédé avec :

$$\frac{y}{g^{\ell_0}} = (g^p)^{\ell_1 + \ell_2 p + \dots + \ell_{p-1} p^{e-2}}.$$

Puis

$$\left(\frac{y}{g^{\ell_0}} \right)^{p^{e-2}} = g^{\ell_1 p^{e-1}}$$

Donc, ℓ_1 est le logarithme de $y^{p^{e-2}} g^{-\ell_0 p^{e-2}}$ en base $g^{p^{e-1}}$ dans \mathbb{G}_1 .

Question. Comment poursuivre pour obtenir ℓ_1 ?

On itère le procédé avec :

$$\frac{y}{g^{\ell_0}} = (g^p)^{\ell_1 + \ell_2 p + \dots + \ell_{p-1} p^{e-2}}.$$

Puis

$$\left(\frac{y}{g^{\ell_0}} \right)^{p^{e-2}} = g^{\ell_1 p^{e-1}}$$

Donc, ℓ_1 est le logarithme de $y^{p^{e-2}} g^{-\ell_0 p^{e-2}}$ en base $g^{p^{e-1}}$ dans \mathbb{G}_1 .

Par **induction**, on va calculer tous les ℓ_i grâce à :

$$\frac{y}{g^{\ell_0 + \ell_1 p + \dots + \ell_{i-1} p^{i-1}}} = (g^{p^{i-1}})^{\ell_i + \ell_{i+1} p + \dots + \ell_{p-1} p^{e-i-1}}$$

Question. Comment poursuivre pour obtenir ℓ_1 ?

On itère le procédé avec :

$$\frac{y}{g^{\ell_0}} = (g^p)^{\ell_1 + \ell_2 p + \dots + \ell_{p-1} p^{e-2}}.$$

Puis

$$\left(\frac{y}{g^{\ell_0}}\right)^{p^{e-2}} = g^{\ell_1 p^{e-1}}$$

Donc, ℓ_1 est le logarithme de $y^{p^{e-2}} g^{-\ell_0 p^{e-2}}$ en base $g^{p^{e-1}}$ dans \mathbb{G}_1 .

Par **induction**, on va calculer tous les ℓ_i grâce à :

$$\frac{y}{g^{\ell_0 + \ell_1 p + \dots + \ell_{i-1} p^{i-1}}} = (g^{p^{i-1}})^{\ell_i + \ell_{i+1} p + \dots + \ell_{p-1} p^{e-i-1}}$$

Au passage, on a démontré

Lemme. Soit $\ell = \log_g(y)$, $y_j = y^{p^{e-j}}$ et $g_j = g^{p^{e-j}}$. Alors, le logarithme de y_j en base g_j est $\ell_j = \ell \bmod p^j$.

Une nouvelle fois, on suppose que l'on dispose d'un algorithme L qui calcule le logarithme discret dans un groupe fini quelconque.

ALGORITHME DE POHLIG-HELLMAN (ÉTAPE « p^e »)

Entrée : un élément $y \in G$ où $|G| = n = p^e$, et un générateur g de G

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

1. Calculer $g_1 \leftarrow g^{p^{e-1}}$.
2. Initialiser $\ell \leftarrow 0$
3. **Pour** i allant de 1 à e , **faire** :
 - 3.1 Calculer $z \leftarrow (y/g^\ell)^{p^{e-i}}$
 - 3.2 À l'aide de l'algorithme L , calculer le logarithme ℓ_{i-1} de z en base g_1
 - 3.3 Mettre à jour $\ell \leftarrow \ell + \ell_{i-1}p^{i-1}$
4. **Retourner** ℓ

Une nouvelle fois, on suppose que l'on dispose d'un algorithme L qui calcule le logarithme discret dans un groupe fini quelconque.

ALGORITHME DE POHLIG-HELLMAN (ÉTAPE « p^e »)

Entrée : un élément $y \in G$ où $|G| = n = p^e$, et un générateur g de G

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

1. Calculer $g_1 \leftarrow g^{p^{e-1}}$.
2. Initialiser $\ell \leftarrow 0$
3. **Pour** i allant de 1 à e , **faire** :
 - 3.1 Calculer $z \leftarrow (y/g^\ell)^{p^{e-i}}$
 - 3.2 À l'aide de l'algorithme L , calculer le logarithme ℓ_{i-1} de z en base g_1
 - 3.3 Mettre à jour $\ell \leftarrow \ell + \ell_{i-1}p^{i-1}$
4. **Retourner** ℓ

Remarque : on peut s'épargner quelques exponentiations en calculant $(y/g^\ell)^{p^{e-i}}$ intelligemment.

Exemple. On se place dans le groupe \mathbb{F}_q^\times où $q = 257 = 2^8 + 1$ (troisième nombre de Fermat).
Un générateur de \mathbb{F}_q^\times est $g = 3$.

Exemple. On se place dans le groupe \mathbb{F}_q^\times où $q = 257 = 2^8 + 1$ (troisième nombre de Fermat).
 Un générateur de \mathbb{F}_q^\times est $g = 3$.

1. Pour $y = 101$, on a

i	y/g^i	ℓ_{i-1}
1	101	1

Exemple. On se place dans le groupe \mathbb{F}_q^\times où $q = 257 = 2^8 + 1$ (troisième nombre de Fermat).
Un générateur de \mathbb{F}_q^\times est $g = 3$.

1. Pour $y = 101$, on a

i	y/g^{ℓ}	ℓ_{i-1}
1	101	1
2	205	1

Exemple. On se place dans le groupe \mathbb{F}_q^\times où $q = 257 = 2^8 + 1$ (troisième nombre de Fermat).
Un générateur de \mathbb{F}_q^\times est $g = 3$.

1. Pour $y = 101$, on a

i	y/g^i	ℓ_{i-1}
1	101	1
2	205	1
3	137	0
4	137	1
5	241	0
6	241	0
7	241	1

Exemple. On se place dans le groupe \mathbb{F}_q^\times où $q = 257 = 2^8 + 1$ (troisième nombre de Fermat).
Un générateur de \mathbb{F}_q^\times est $g = 3$.

1. Pour $y = 101$, on a

i	y/g^i	ℓ_{i-1}
1	101	1
2	205	1
3	137	0
4	137	1
5	241	0
6	241	0
7	241	1
8	1	0

Exemple. On se place dans le groupe \mathbb{F}_q^\times où $q = 257 = 2^8 + 1$ (troisième nombre de Fermat).
Un générateur de \mathbb{F}_q^\times est $g = 3$.

1. Pour $y = 101$, on a

i	y/g^i	ℓ_{i-1}
1	101	1
2	205	1
3	137	0
4	137	1
5	241	0
6	241	0
7	241	1
8	1	0

2. On en déduit que $\ell = \sum_{i=0} \ell_i 2^i = 75$

Question. Comment regrouper ces étapes ?

Idée : si $G \simeq \mathbb{Z}/p_1^{e_1}\mathbb{Z} \times \cdots \times \mathbb{Z}/p_k^{e_k}\mathbb{Z}$, alors on calcule itérativement les logarithmes dans les sous-groupes $G_i = \mathbb{Z}/p_1^{e_1}\mathbb{Z} \times \cdots \times \mathbb{Z}/p_i^{e_i}\mathbb{Z}$, pour i allant de 1 à k .

Question. Comment regrouper ces étapes ?

Idée : si $G \simeq \mathbb{Z}/p_1^{e_1}\mathbb{Z} \times \cdots \times \mathbb{Z}/p_k^{e_k}\mathbb{Z}$, alors on calcule itérativement les logarithmes dans les sous-groupes $G_i = \mathbb{Z}/p_1^{e_1}\mathbb{Z} \times \cdots \times \mathbb{Z}/p_i^{e_i}\mathbb{Z}$, pour i allant de 1 à k .

ALGORITHME DE POHLIG-HELLMAN

Entrée : un élément $y \in G$ où $|G| = n$, et un générateur g de G et une factorisation $F = [(p_i, e_i)]_i$ de $n = \prod_i p_i^{e_i}$

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

1. Initialiser $\ell \leftarrow 0$ et $m \leftarrow 1$.
2. **Pour tout** (p_i, e_i) dans F , **faire** :
 - 2.1 Calculer $\ell_i \leftarrow \text{pohlig_hellman_pow}(y, g, p_i, e_i)$
 - 2.2 **Si** $m \neq 1$, **alors** calculer $\ell \leftarrow \text{pohlig_hellman_coprime}(y, g, s = p_i^{e_i}, t = m)$
 - 2.3 **Sinon**, assigner $\ell \leftarrow \ell_i$
 - 2.4 $m \leftarrow m \times p_i^{e_i}$
3. **Retourner** ℓ

Complexité. On note $C(p_i)$ le coût d'un l'algorithme externe L de résolution de logarithme discret, utilisé avec comme entrée un sous-groupe d'ordre p_i .

Complexité. On note $C(p_i)$ le coût d'un l'algorithme externe L de résolution de logarithme discret, utilisé avec comme entrée un sous-groupe d'ordre p_i .

Pour chaque facteur $p_i^{\ell_i}$ de n :

Complexité. On note $C(p_i)$ le coût d'un l'algorithme externe L de résolution de logarithme discret, utilisé avec comme entrée un sous-groupe d'ordre p_i .

Pour chaque facteur $p_i^{e_i}$ de n :

- `pohlig_hellman_pow`(y, g, p_i, e_i) effectue deux appels à L , un calcul de pgcd, quelques calculs modulaires et exponentiations dans le groupe

$$\implies \text{complexité en } O\left(C(p_i) \text{polylog}(p_i^{e_i})\right)$$

Complexité. On note $C(p_i)$ le coût d'un l'algorithme externe L de résolution de logarithme discret, utilisé avec comme entrée un sous-groupe d'ordre p_i .

Pour chaque facteur $p_i^{e_i}$ de n :

- `pohlig_hellman_pow`(y, g, p_i, e_i) effectue deux appels à L , un calcul de pgcd, quelques calculs modulaires et exponentiations dans le groupe

$$\implies \text{complexité en } O\left(C(p_i) \text{ polylog}(p_i^{e_i})\right)$$

- `pohlig_hellman_coprime`($y, g, s = p_i^{e_i}, t = m$) effectue e_i appels à L et quelques exponentiations dans le groupe

$$\implies \text{complexité en } O\left(e_i C(p_i) \text{ polylog}(p_i^{e_i})\right)$$

Complexité. On note $C(p_i)$ le coût d'un l'algorithme externe L de résolution de logarithme discret, utilisé avec comme entrée un sous-groupe d'ordre p_i .

Pour chaque facteur $p_i^{e_i}$ de n :

- `pohlig_hellman_pow`(y, g, p_i, e_i) effectue deux appels à L , un calcul de pgcd, quelques calculs modulaires et exponentiations dans le groupe

$$\implies \text{complexité en } O\left(C(p_i) \text{polylog}(p_i^{e_i})\right)$$

- `pohlig_hellman_coprime`($y, g, s = p_i^{e_i}, t = m$) effectue e_i appels à L et quelques exponentiations dans le groupe

$$\implies \text{complexité en } O\left(e_i C(p_i) \text{polylog}(p_i^{e_i})\right)$$

- les autres opérations sont un nombre constant d'exponentiations

Complexité. On note $C(p_i)$ le coût d'un l'algorithme externe L de résolution de logarithme discret, utilisé avec comme entrée un sous-groupe d'ordre p_i .

Pour chaque facteur $p_i^{e_i}$ de n :

- `pohlig_hellman_pow`(y, g, p_i, e_i) effectue deux appels à L , un calcul de pgcd, quelques calculs modulaires et exponentiations dans le groupe

$$\implies \text{complexité en } O\left(C(p_i) \text{ polylog}(p_i^{e_i})\right)$$

- `pohlig_hellman_coprime`($y, g, s = p_i^{e_i}, t = m$) effectue e_i appels à L et quelques exponentiations dans le groupe

$$\implies \text{complexité en } O\left(e_i C(p_i) \text{ polylog}(p_i^{e_i})\right)$$

- les autres opérations sont un nombre constant d'exponentiations

Complexité. Si la factorisation de n est connue et si L a complexité en $O(\sqrt{|\mathbb{G}_i|})$, alors l'algorithme de Pohlig-Hellman fonctionne en temps

$$O\left(\sqrt{p} \text{ polylog}(n)\right)$$

où p est le plus gros facteur premier de n .

1. Logarithme discret dans un groupe générique

Pas de bébé – pas de géant

Algorithme de Pohlig-Hellman

Autres algorithmes

2. Borne inférieure de complexité

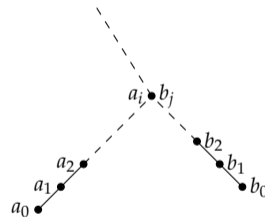
Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

Idée. On définit une suite récurrente de la forme

$$x_{i+1} = x_i g^{f(x_i)}$$

où $f : G \rightarrow S \subset \mathbb{Z}$ est surjective (S pas trop grand).



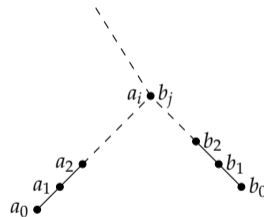
Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

Idée. On définit une suite récurrente de la forme

$$x_{i+1} = x_i g^{f(x_i)}$$

où $f : G \rightarrow S \subset \mathbb{Z}$ est surjective (S pas trop grand).

Puis, on considère deux instances (a_i) et (b_i) de cette suite, avec deux conditions initiales $a_0 = y^\alpha$ et $b_0 = y^\beta$ telles que $\text{pgcd}(\alpha - \beta, n) = 1$.



Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

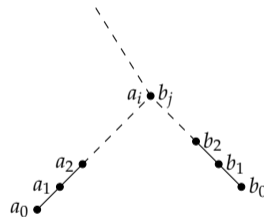
Idée. On définit une suite récurrente de la forme

$$x_{i+1} = x_i g^{f(x_i)}$$

où $f : G \rightarrow S \subset \mathbb{Z}$ est surjective (S pas trop grand).

Puis, on considère deux instances (a_i) et (b_i) de cette suite, avec deux conditions initiales $a_0 = y^\alpha$ et $b_0 = y^\beta$ telles que $\text{pgcd}(\alpha - \beta, n) = 1$. **S'il advient que $a_i = b_j$** , alors on aura :

$$y^\alpha g^{f(a_0)+\dots+f(a_{i-1})} = y^\beta g^{f(b_0)+\dots+f(b_{j-1})}$$



Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

Idée. On définit une suite récurrente de la forme

$$x_{i+1} = x_i g^{f(x_i)}$$

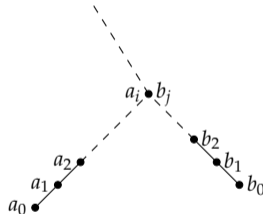
où $f : G \rightarrow S \subset \mathbb{Z}$ est surjective (S pas trop grand).

Puis, on considère deux instances (a_i) et (b_i) de cette suite, avec deux conditions initiales $a_0 = y^\alpha$ et $b_0 = y^\beta$ telles que $\text{pgcd}(\alpha - \beta, n) = 1$. **S'il advient que $a_i = b_j$** , alors on aura :

$$y^\alpha g^{f(a_0) + \dots + f(a_{i-1})} = y^\beta g^{f(b_0) + \dots + f(b_{j-1})}$$

Puis, ℓ est solution de :

$$(\beta - \alpha)\ell = \left(f(a_0) + \dots + f(a_{i-1}) \right) - \left(f(b_0) + \dots + f(b_{j-1}) \right)$$



Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

Idée. On définit une suite récurrente de la forme

$$x_{i+1} = x_i g^{f(x_i)}$$

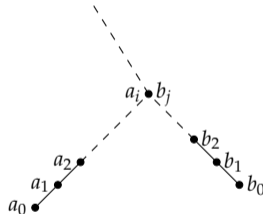
où $f : G \rightarrow S \subset \mathbb{Z}$ est surjective (S pas trop grand).

Puis, on considère deux instances (a_i) et (b_i) de cette suite, avec deux conditions initiales $a_0 = y^\alpha$ et $b_0 = y^\beta$ telles que $\text{pgcd}(\alpha - \beta, n) = 1$. **S'il advient que $a_i = b_j$** , alors on aura :

$$y^\alpha g^{f(a_0) + \dots + f(a_{i-1})} = y^\beta g^{f(b_0) + \dots + f(b_{j-1})}$$

Puis, ℓ est solution de :

$$(\beta - \alpha)\ell = \left(f(a_0) + \dots + f(a_{i-1})\right) - \left(f(b_0) + \dots + f(b_{j-1})\right)$$



Complexité.

Présentation rapide de l'**algorithme** λ de Pollard, aussi appelé « méthode des kangourous ».

Idée. On définit une suite récurrente de la forme

$$x_{i+1} = x_i g^{f(x_i)}$$

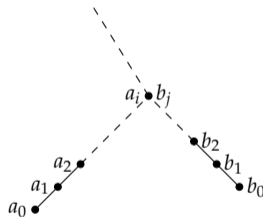
où $f : G \rightarrow S \subset \mathbb{Z}$ est surjective (S pas trop grand).

Puis, on considère deux instances (a_i) et (b_i) de cette suite, avec deux conditions initiales $a_0 = y^\alpha$ et $b_0 = y^\beta$ telles que $\text{pgcd}(\alpha - \beta, n) = 1$. **S'il advient que $a_i = b_j$** , alors on aura :

$$y^\alpha g^{f(a_0) + \dots + f(a_{i-1})} = y^\beta g^{f(b_0) + \dots + f(b_{j-1})}$$

Puis, ℓ est solution de :

$$(\beta - \alpha)\ell = \left(f(a_0) + \dots + f(a_{i-1}) \right) - \left(f(b_0) + \dots + f(b_{j-1}) \right)$$



Complexité. Avec le paradoxe des anniversaires, on obtient une collision après avoir calculé $O(\sqrt{n})$ termes au total.

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.
4. Alors, les suites (a_i) et (b_i) sont :

i	a_i	b_i

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.
4. Alors, les suites (a_i) et (b_i) sont :

i	a_i	b_i
0	$3^2 = 9$	$3^3 = 5$

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.
4. Alors, les suites (a_i) et (b_i) sont :

i	a_i	b_i
0	$3^2 = 9$	$3^3 = 5$
1	$9 \times 2^9 = 10$	$5 \times 2^5 = 6$

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.
4. Alors, les suites (a_i) et (b_i) sont :

i	a_i	b_i
0	$3^2 = 9$	$3^3 = 5$
1	$9 \times 2^9 = 10$	$5 \times 2^5 = 6$
2	$10 \times 2^{10} = 10$	$6 \times 2^6 = 10$

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.
4. Alors, les suites (a_i) et (b_i) sont :

i	a_i	b_i
0	$3^2 = 9$	$3^3 = 5$
1	$9 \times 2^9 = 10$	$5 \times 2^5 = 6$
2	$10 \times 2^{10} = 10$	$6 \times 2^6 = 10$

5. On a $a_1 = b_2$. Donc ℓ est solution de

$$(\beta - \alpha)\ell = f(a_0) - f(b_0) - f(b_1) \pmod{10}$$

Exemple. On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

1. On cherche le logarithme en base g de $y = 3$.
2. On prend la fonction $f : G \rightarrow \mathbb{Z}$ définie par $f(x) = x$ (on ne pourra pas toujours).
3. On choisit $\alpha = 2$ et $\beta = 3$.
4. Alors, les suites (a_i) et (b_i) sont :

i	a_i	b_i
0	$3^2 = 9$	$3^3 = 5$
1	$9 \times 2^9 = 10$	$5 \times 2^5 = 6$
2	$10 \times 2^{10} = 10$	$6 \times 2^6 = 10$

5. On a $a_1 = b_2$. Donc ℓ est solution de

$$(\beta - \alpha)\ell = f(a_0) - f(b_0) - f(b_1) \pmod{10}$$

6. On obtient $\ell = 9 - 5 - 6 \equiv 8 \pmod{10}$.

Une variation de l'idée de la méthode ρ de Pollard

Idée : on réécrit $yg^{-\ell} = 1$ et on cherche encore une collision sur les $\{y^a g^b \mid (a, b) \in \mathbb{Z}\}$.

Une variation de l'idée de la méthode ρ de Pollard

Idée : on réécrit $yg^{-\ell} = 1$ et on cherche encore une collision sur les $\{y^a g^b \mid (a, b) \in \mathbb{Z}\}$.

S'il advient que $y^a g^b = y^c g^d$, alors on a $\ell \equiv (d - b) \times (a - c)^{-1} \pmod n$, pourvu que $a - c$ soit inversible dans $\mathbb{Z}/n\mathbb{Z}$.

Une variation de l'idée de la méthode ρ de Pollard

Idée : on réécrit $yg^{-\ell} = 1$ et on cherche encore une collision sur les $\{y^a g^b \mid (a, b) \in \mathbb{Z}\}$.

S'il advient que $y^a g^b = y^c g^d$, alors on a $\ell \equiv (d - b) \times (a - c)^{-1} \pmod n$, pourvu que $a - c$ soit inversible dans $\mathbb{Z}/n\mathbb{Z}$.

Il faut maintenant construire une suite récurrente aléatoire $(x_i)_{i \in \mathbb{N}}$ où

$$x_{i+1} = f(x_i) \quad \text{et} \quad x_i = y^{a_i} g^{b_i}.$$

Une variation de l'idée de la méthode ρ de Pollard

Idée : on réécrit $yg^{-\ell} = 1$ et on cherche encore une collision sur les $\{y^a g^b \mid (a, b) \in \mathbb{Z}\}$.

S'il advient que $y^a g^b = y^c g^d$, alors on a $\ell \equiv (d - b) \times (a - c)^{-1} \pmod n$, pourvu que $a - c$ soit inversible dans $\mathbb{Z}/n\mathbb{Z}$.

Il faut maintenant construire une suite récurrente aléatoire $(x_i)_{i \in \mathbb{N}}$ où

$$x_{i+1} = f(x_i) \quad \text{et} \quad x_i = y^{a_i} g^{b_i}.$$

En **pratique**, on obtient de bons résultats en choisissant f de la sorte. On se fixe trois sous-ensembles **quelconques** S_0, S_1, S_2 de \mathbb{G} .

$$f(x) = \begin{cases} yx & \text{si } x \in S_0 & \text{(on incrémente } a) \\ x^2 & \text{si } x \in S_1 & \text{(on double } a \text{ et } b) \\ gx & \text{si } x \in S_2 & \text{(on incrémente } b) \end{cases}$$

Une variation de l'idée de la méthode ρ de Pollard

Idée : on réécrit $yg^{-\ell} = 1$ et on cherche encore une collision sur les $\{y^a g^b \mid (a, b) \in \mathbb{Z}\}$.

S'il advient que $y^a g^b = y^c g^d$, alors on a $\ell \equiv (d - b) \times (a - c)^{-1} \pmod n$, pourvu que $a - c$ soit inversible dans $\mathbb{Z}/n\mathbb{Z}$.

Il faut maintenant construire une suite récurrente aléatoire $(x_i)_{i \in \mathbb{N}}$ où

$$x_{i+1} = f(x_i) \quad \text{et} \quad x_i = y^{a_i} g^{b_i}.$$

En **pratique**, on obtient de bons résultats en choisissant f de la sorte. On se fixe trois sous-ensembles **quelconques** S_0, S_1, S_2 de \mathbb{G} .

$$f(x) = \begin{cases} yx & \text{si } x \in S_0 & \text{(on incrémente } a) \\ x^2 & \text{si } x \in S_1 & \text{(on double } a \text{ et } b) \\ gx & \text{si } x \in S_2 & \text{(on incrémente } b) \end{cases}$$

Autrement dit, les séquences (a_n) et (b_n) sont régies par :

$$a_{n+1} = \begin{cases} n+1 & \text{si } a_n \in S_0 \\ 2n & \text{si } a_n \in S_1 \\ n & \text{si } a_n \in S_2 \end{cases} \quad \text{et} \quad b_{n+1} = \begin{cases} n & \text{si } b_n \in S_0 \\ 2n & \text{si } b_n \in S_1 \\ n+1 & \text{si } b_n \in S_2 \end{cases}$$

Une variation de l'idée de la méthode ρ de Pollard

Idée : on réécrit $yg^{-\ell} = 1$ et on cherche encore une collision sur les $\{y^a g^b \mid (a, b) \in \mathbb{Z}\}$.

S'il advient que $y^a g^b = y^c g^d$, alors on a $\ell \equiv (d - b) \times (a - c)^{-1} \pmod n$, pourvu que $a - c$ soit inversible dans $\mathbb{Z}/n\mathbb{Z}$.

Il faut maintenant construire une suite récurrente aléatoire $(x_i)_{i \in \mathbb{N}}$ où

$$x_{i+1} = f(x_i) \quad \text{et} \quad x_i = y^{a_i} g^{b_i}.$$

En **pratique**, on obtient de bons résultats en choisissant f de la sorte. On se fixe trois sous-ensembles **quelconques** S_0, S_1, S_2 de \mathbb{G} .

$$f(x) = \begin{cases} yx & \text{si } x \in S_0 & \text{(on incrémente } a) \\ x^2 & \text{si } x \in S_1 & \text{(on double } a \text{ et } b) \\ gx & \text{si } x \in S_2 & \text{(on incrémente } b) \end{cases}$$

Autrement dit, les séquences (a_n) et (b_n) sont régies par :

$$a_{n+1} = \begin{cases} n+1 & \text{si } a_n \in S_0 \\ 2n & \text{si } a_n \in S_1 \\ n & \text{si } a_n \in S_2 \end{cases} \quad \text{et} \quad b_{n+1} = \begin{cases} n & \text{si } b_n \in S_0 \\ 2n & \text{si } b_n \in S_1 \\ n+1 & \text{si } b_n \in S_2 \end{cases}$$

Ensuite, on peut utiliser la méthode de recherche de collision de Floyd ($x_i \stackrel{?}{=} x_{2i}$) pour chercher les collisions.

Complexité : $O(\sqrt{n})$ par le paradoxe des anniversaires.

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.
On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$.
On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0
1	$yx_0 = 3$	$a_0 + 1 = 1$	$b_0 = 0$

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0
1	$yx_0 = 3$	$a_0 + 1 = 1$	$b_0 = 0$
2	$yx_1 = 9$	$a_1 + 1 = 2$	$b_1 = 0$

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0
1	$yx_0 = 3$	$a_0 + 1 = 1$	$b_0 = 0$
2	$yx_1 = 9$	$a_1 + 1 = 2$	$b_1 = 0$
3	$gx_2 = 18 \equiv 7$	$a_2 = 2$	$b_2 + 1 = 1$

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0
1	$yx_0 = 3$	$a_0 + 1 = 1$	$b_0 = 0$
2	$yx_1 = 9$	$a_1 + 1 = 2$	$b_1 = 0$
3	$gx_2 = 18 \equiv 7$	$a_2 = 2$	$b_2 + 1 = 1$
4	$gx_3 = 14 \equiv 3$	$a_3 = 2$	$b_3 + 1 = 2$

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0
1	$yx_0 = 3$	$a_0 + 1 = 1$	$b_0 = 0$
2	$yx_1 = 9$	$a_1 + 1 = 2$	$b_1 = 0$
3	$gx_2 = 18 \equiv 7$	$a_2 = 2$	$b_2 + 1 = 1$
4	$gx_3 = 14 \equiv 3$	$a_3 = 2$	$b_3 + 1 = 2$

3. On obtient alors

$$(a_4 - a_1)\ell = b_1 - b_4 \pmod{10} \equiv \ell = -2 \equiv 8 \pmod{10}.$$

Exemple sans la recherche de collision de Floyd (pour clarifier). On prend $G = \mathbb{F}_{11}^\times$, engendré par $g = 2$. On cherche le logarithme en base g de $y = 3$ (même exemple que pour λ).

1. On partitionne $G = \{1, \dots, 10\}$ en trois ensembles de taille semblable :

$$S_0 = \{1, 2, 3\}, \quad S_1 = \{4, 5, 6\}, \quad S_2 = \{7, 8, 9, 10\}.$$

2. On initialise avec $a_0 = 0, b_0 = 0$ et $x_0 = g^{b_0} = 1$. On obtient alors

i	x_i	a_i	b_i
0	1	0	0
1	$yx_0 = 3$	$a_0 + 1 = 1$	$b_0 = 0$
2	$yx_1 = 9$	$a_1 + 1 = 2$	$b_1 = 0$
3	$gx_2 = 18 \equiv 7$	$a_2 = 2$	$b_2 + 1 = 1$
4	$gx_3 = 14 \equiv 3$	$a_3 = 2$	$b_3 + 1 = 2$

3. On obtient alors

$$(a_4 - a_1)\ell = b_1 - b_4 \pmod{10} \equiv \ell = -2 \equiv 8 \pmod{10}.$$

En pratique, avec la recherche de collision de Floyd on aurait plutôt calculé les $(x_i, a_i, b_i, x_{2i}, a_{2i}, b_{2i})$ et simplement testé si $x_i = x_{2i}$.

1. Logarithme discret dans un groupe générique

Pas de bébé – pas de géant

Algorithme de Pohlig-Hellman

Autres algorithmes

2. Borne inférieure de complexité

L'**algorithme de Pohlig-Hellman** couplé à la méthode « pas de bébé – pas de géant » (par exemple) permet de résoudre le problème du logarithme discret dans un groupe G d'ordre n .

L'**algorithme de Pohlig-Hellman** couplé à la méthode « pas de bébé – pas de géant » (par exemple) permet de résoudre le problème du logarithme discret dans un groupe G d'ordre n .

Ces algorithmes utilisent **uniquement la structure du groupe** G . On dit qu'on résout le logarithme discret dans un groupe générique.

L'**algorithme de Pohlig-Hellman** couplé à la méthode « pas de bébé – pas de géant » (par exemple) permet de résoudre le problème du logarithme discret dans un groupe G d'ordre n .

Ces algorithmes utilisent **uniquement la structure du groupe** G . On dit qu'on résout le logarithme discret dans un groupe générique.

La **complexité** de Pohlig-Hellman est approximativement de $O(\sqrt{p})$ où p est le plus grand facteur premier de n .

L'**algorithme de Pohlig-Hellman** couplé à la méthode « pas de bébé – pas de géant » (par exemple) permet de résoudre le problème du logarithme discret dans un groupe G d'ordre n .

Ces algorithmes utilisent **uniquement la structure du groupe** G . On dit qu'on résout le logarithme discret dans un groupe générique.

La **complexité** de Pohlig-Hellman est approximativement de $O(\sqrt{p})$ où p est le plus grand facteur premier de n .

Question. Est-il possible de faire sensiblement mieux ?

Définition. Un groupe générique d'ordre n est la donnée :

- d'un groupe G d'ordre n ,
- d'un isomorphisme de groupe aléatoire $\sigma : \mathbb{Z}/n\mathbb{Z} \rightarrow G$.

Définition. Un groupe générique d'ordre n est la donnée :

- d'un groupe G d'ordre n ,
- d'un isomorphisme de groupe aléatoire $\sigma : \mathbb{Z}/n\mathbb{Z} \rightarrow G$.

Idée : on fait un codage quelconque (aléatoire) de G pour masquer toute éventuelle structure supplémentaire.

Définition. Un groupe générique d'ordre n est la donnée :

- d'un groupe \mathbb{G} d'ordre n ,
- d'un isomorphisme de groupe aléatoire $\sigma : \mathbb{Z}/n\mathbb{Z} \rightarrow \mathbb{G}$.

Idée : on fait un codage quelconque (aléatoire) de \mathbb{G} pour masquer toute éventuelle structure supplémentaire.

On a alors le résultat suivant :

Théorème. (Shoup 1997) Soit (\mathbb{G}, σ) un groupe générique d'ordre $n = p^t s$ où p est premier, et p et s sont premiers entre eux. Soit g un générateur de \mathbb{G} et $y \in \mathbb{G}$. Alors, pour tout algorithme $L(y, g)$ faisant au plus m requêtes à σ , on a

$$\mathbb{P} \left(g^{L(y,g)} = y \right) \leq \frac{m^2}{p}.$$

Définition. Un groupe générique d'ordre n est la donnée :

- d'un groupe G d'ordre n ,
- d'un isomorphisme de groupe aléatoire $\sigma : \mathbb{Z}/n\mathbb{Z} \rightarrow G$.

Idée : on fait un codage quelconque (aléatoire) de G pour masquer toute éventuelle structure supplémentaire.

On a alors le résultat suivant :

Théorème. (Shoup 1997) Soit (G, σ) un groupe générique d'ordre $n = p^t s$ où p est premier, et p et s sont premiers entre eux. Soit g un générateur de G et $y \in G$. Alors, pour tout algorithme $L(y, g)$ faisant au plus m requêtes à σ , on a

$$\mathbb{P} \left(g^{L(y,g)} = y \right) \leq \frac{m^2}{p}.$$

Autrement dit, pour obtenir une probabilité constante de réussite, il faut que m soit au moins de l'ordre de \sqrt{p} .

Définition. Un groupe générique d'ordre n est la donnée :

- d'un groupe G d'ordre n ,
- d'un isomorphisme de groupe aléatoire $\sigma : \mathbb{Z}/n\mathbb{Z} \rightarrow G$.

Idée : on fait un codage quelconque (aléatoire) de G pour masquer toute éventuelle structure supplémentaire.

On a alors le résultat suivant :

Théorème. (Shoup 1997) Soit (G, σ) un groupe générique d'ordre $n = p^t s$ où p est premier, et p et s sont premiers entre eux. Soit g un générateur de G et $y \in G$. Alors, pour tout algorithme $L(y, g)$ faisant au plus m requêtes à σ , on a

$$\mathbb{P} \left(g^{L(y,g)} = y \right) \leq \frac{m^2}{p}.$$

Autrement dit, pour obtenir une probabilité constante de réussite, il faut que m soit au moins de l'ordre de \sqrt{p} .

« La complexité du log discret dans un groupe générique est $\Theta(\sqrt{p})$. »

Questions?