ELGamal

Andreea Dragut (dragut@univmed.fr) Cours de cryptographie Chapitre IV

4.0.1 Fonctions conjecturées à sens unique : Le problème du logarithme discret

Definition. *Un groupe cyclique G, est un groupe dans lequel il existe un élément g tel que tout élément du groupe puisse s'exprimer sous forme d'un multiple/puissance de g, cet élément g est appelé générateur du groupe et on note $G =< g >$.*

En notation additive : $(G, +)$ $[n]g$
En notation multiplicative : $(G, \cdot)$ g^n

Definition. *Soit G un groupe et $g \in G$, alors le groupe sous-groupe H généré par g, noté $H =< g >$, est le plus petit sous-groupe de G contenant g.*

Definition. *L'ordre d'un élément g d'un groupe G est l'ordre du sous-groupe $H =< g >$ généré par cet élément. L'ordre de g est noté $ordre(g)$ ou $o(g)$. Si l'ordre est fini, il est le plus petit entier $m > 0$*

- *En notation additive : $mg = 0$*
- *En notation multiplicative : $g^m = 1$*

***On peut dire que si l'ordre de** g **est fini** $ordre(g) = |H| = | < g > |$ **la cardinalité sous-groupe** $H =< g >$ **généré par** g.*

Exemple. *Trouver l'ordre de l'élément 2 dans*

- $F_{11}^* = (\mathbb{Z}/11\mathbb{Z})^* = \{1, 2, ..., 10\}$ *le groupe multiplicatif de restes modulo* 11
- $F_{17}^* = (\mathbb{Z}/17\mathbb{Z})^* = \{1, 2, ..., 16\}$ *le groupe multiplicatif de restes modulo* 17

Dans $\mathbb{F}_{11}^*$ nous avons $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 \equiv 5$, $2^5 \equiv 10$, $2^6 \equiv 9$, $2^7 \equiv 7$, $2^8 \equiv 3$, $2^9 \equiv 6$. Mais $2^{10} \equiv 1$ et on commence à répéter les éléments.

Donc $ordre(2) = 10$ et il génère tout le groupe G.

Dans $\mathbb{F}_{17}^*$ nous avons $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$,$2^4 = 16$, $2^5 \equiv 15$, $2^6 \equiv 13$, $2^7 \equiv 9$. Mais $2^8 \equiv 1$ et on commence à répéter les éléments. Donc 2 n'est pas un générateur pour

$\mathbb{F}_{17}^*$. Il génére que le sous-groupe $H =< 2 >= \{1, 2, 4, 8, 9, 13, 15, 16\}$ et son ordre est la cardinalité de ce sous-groupe, donc 8.

Le problème DLP peut être formulé pour un groupe générique H et $< g >= G \subset H$. On va étudier après le DLP sur des groupes particuliers : le groupe multiplicatif $(\mathbb{Z}/p\mathbb{Z})^*$ et le groupe additif de courbes elliptiques. Pour certains groupes le problème du logarithme discret est "difficile" : soit on ne connaît aucun algorithme sous-exponentiel qui résout le problème DLP, soit on ne connaît aucun algorithme qui résout le problème DLP pour des tailles grandes (disons de 1024 bits et plus).

Definition. [Le problème du logarithme discret DLP] *Soit H un groupe et g un générateur d'ordre p d'un sous-groupe $G \subseteq H$. Étant donnés $A \in G$, trouver x tel que $A = g^x \Leftrightarrow \log_g(A) = x$.*

Le système de chiffrement El Gamal est basé sur le problème du logarithme discret, c'est à dire d'inverser l'exponentielle modulaire. Il a été créé par Taher Elgamal. Cet algorithme est utilisé par GNU Privacy Guard et de récentes versions de PGP.

La clé privée de de ElGamal est a. La clé publique de ElGamal est $A = g^a \pmod p$, où $g \in H$ avec un ordre p grand. Décrypter clé publique de ElGamal en partant de sa clé publique revient à résoudre un problème du logarithme discret pour $y = A$.

Création d'un paramètre publique

Un tiers parti de confiance "Trent" :
- H le groupe choisi
- $g \in H$ avec un ordre p grand

Création de clé

- on choisit $1 \leq a \leq p - 1$.
- on calcule $A = g^a \pmod p$
- on publie la clé publique (H, p, g, A)

Cryptage

- Input :
  - la clé publique (H, p, g, A)
  - le message m
  - une clé éphémère k
- calcul :
  - $y_1 = g^k \pmod p$
  - $y_2 = m \cdot A^k \pmod p$
- Output : (y_1, y_2)

Décryptage

- Input :
  - la clé privée a
  - le message chiffré (y_1, y_2)
- calcul : $(y_1^a)^{-1} y_2 \pmod p$

Théorème. [El Gamal fonctionne correctement] *Soit H un groupe et g un générateur d'ordre p d'un sous-groupe $G \subseteq H$ (donc G a p éléments). Soit la clé privée a tel que $1 \leq a \leq p = ord(G)$. Soit le message x et une clé éphémère k. En calculant $A = g^a \pmod p$, $y_1 = g^k \pmod p$, $y_2 = m \cdot A^k \pmod p$ selon l'algorithme du ElGamal nous avons que $(y_1^a)^{-1} y_2 \equiv x \pmod p$*

Preuve. *On peut calculer $(y_1^a)^{-1}$ parce que avec la clé privée on peut calculer d'abord $z = y_1^a \pmod p$ et après $z^{-1} \pmod p$.*

$$(y_1^a)^{-1} y_2 \equiv (g^{ak})^{-1} (m \cdot A^k) \equiv (g^{ak})^{-1} \left(m (g^a)^k\right)$$

parce que $A = g^a \pmod p$ donc $(y_1^a)^{-1} y_2 \equiv m \pmod p$.

Algorithmes génériques pour résoudre le problème DLP

Proposition. *Soit G un groupe et $g \in G$ un élément d'ordre $N \leq p = ord(G)$. Le problème DLP :trouver x tel que $g^x = A$ peut être résolu en $O(N) = O(S_x)$ pas.*

Preuve. *On génère par multiplication successive la liste des valeurs S_x pour $x = 0, 1, \ldots, N-1$. S'il existe une solution $g^x = A$ on va la retrouver dans la liste.*

Le problème DLP et l'algorithme de Shanks peuvent être formulés pour un groupe générique H et $< g >= G \subset H$.

Proposition. [Shanks – pas de bébé, pas de géant] *Soit $(G, \cdot)$ un groupe et $g \in G$ un élément d'ordre $N > 2$ (pour $H = G = \mathbb{Z}_p$, $N = p - 1$. L'algorithme suivant résout le problème DLP en $O(\sqrt{N} \log N)$ pas.*

1. *soit $s = 1 + \lfloor \sqrt{N} \rfloor > \sqrt{N}$*
2. *calculer g^{-s}*
3. *créer deux listes*
   L_1 $e, g, g^2, g^3, \ldots, g^s$
   L_2 $A, A \cdot g^{-s}, A \cdot g^{-2s}, A \cdot g^{-3s}, \ldots, A \cdot g^{-s^2}$
4. *trouver une occurence commune $g_{r_0} = A \cdot g^{-k_0 \cdot s}$*
5. *$x = r_0 + k_0 \cdot s$ est une solution pour $g^x = A$.*

Complexité d'attaque de Shanks : $O(s \log s) \simeq O(\sqrt{N} \log N)$
- créer les deux listes : $2s$ multiplications

– si une occurrence commune existe, on peut la trouver en $O(\log s)$

Preuve. *On utilise le théorème d'Euclide.*

Soit x_0 la solution du DLP $g^x = A$.

Alors, ils existent k_0, r_0 entiers tels que $x_0 = sk_0 + r_0$, $0 \leq r_0 < s$, donc $1 \leq x_0 < N$.

$q_0 = x_0 - r_0 s < \frac{N}{s} < s$, parce que nous avons choisi $s < \sqrt{N}$.

On peut réécrire $g^{x_0} = A$ comme $L_1 \ni g^{r_0} = A \cdot g^{-k_0 \cdot s}, \in L_2$, $0 \leq r < s$, $0 \leq k < s$. Donc chaque liste a au plus s éléments.

Réduction à un problème connu

Supposons que dans le groupe G on peut résoudre (DLP) $g^x = A \pmod{ord(g) = p}$ "facilement" (c.à-d. en complexité sous-exponentielle). Alors, dans un groupe plus genéral on essaye de réduire "facilement" le (DLP) dans le groupe H à plusieurs problèmes (DLP) $\overline{g}^x = A \pmod{ord(\overline{g}) = p}$ dans des sous-groupes de type $\overline{G} =< \overline{g} >$, avec $ord(\overline{g}) = p$ un entier premier. Ce dernier problème étant similaire au (DLP) $g^x = A \pmod{ord(g) = p}$.

1. $G =< g >$ avec $\text{ord}(g) = p^k$: en construisant $g^{p^{k-1}}$ qui est un élément d'ordre p.
2. $G =< g >$ avec $\text{ord}(g) = p_1^{k_1}, \ldots, p_n^{k_n}$: en construisant $g_i = g^{N/p_i^{k_i}}$ qui est un élément d'ordre $\text{ord}(g_i) = p_i^{k_i}$ et en utilisant le théorème des restes chinois pour assembler les solutions.

Proposition. *Soit G un groupe, p un nombre premier, $g \in G$. Il faut S_p pas pour résoudre le problème*
DLP_p : $g^x = h$ et $\text{ord}(g) = p$
Alors si $\text{ord}(g) = p^k$ il faut $O(kS_p)$ pour résoudre DLP $g^x = h$.

Preuve (ébauche)**.** *Soit $\overline{x}$ la solution du DLP.*

$$\overline{x} = x_0 + x_1 p + \cdots + x_{k-1}^{p^{k-1}}$$

$\text{ord}(g^{p^{k-1}}) = p$ – *élever à la puissance p^{k-1}, donc*

$$\begin{aligned}
h^{p^{k-1}} &= (g^{\overline{x}})^{p^{k-1}} \\
&= (g^{x_0 + x_1 p + \cdots + x_{k-1} p^{k-1}})^{p^{k-1}} \\
&= g_0^x p^{k-1} \cdot \underbrace{(g^{p^k})}_{=1}^{x_1 + x_2 p + \cdots + x_{k-1} p^{k-1}} \\
&= \left(g^{p^{k-1}}\right)^{x_0} \\
\textit{(DLP_p facile) :}(g^{p^{k-1}})^{x_0} &= h^{p^{k-1}}, g^{p^{k-1}} \in G \textit{ et son ordre est} \\
\text{ord}(g^{p^{k-1}}) &= p
\end{aligned}$$

Pour déterminer x_1, il faut élever à la puissance p^{k-2}. Donc

$h^{p^{k-2}} = g^{x_0 p^{k-2}} \cdot g^{x_1^{p^{k-1}}} \iff$
$(g^{p^{k-1}})^{x_1} = (h \cdot g^{-x_0})^{p^{k-2}}$

Proposition. *Soit G un groupe et $g \in G$, $\mathrm{ord}(g) = N = p_1^{k_1} \cdot p_t^{k_t}$. Soit un algorithme qui résout $DLP_{p_i^{k_i}}$ en $O(S_{p_i} k_i)$ pas, alors DLP $g^x = h$ peut être résolu dans $O(\sum_{i=1}^{t} S_{p_i} k_i + \log N)$ pas.*

Algorithme Pohling Hellmann

1. Pour chaque $1 \leq i \leq t$, $g_i = g^{N/p_i^{k_i}} \pmod N$ et $h_i = h^{N/p_i^{k_i}} \pmod N$, $\mathrm{ord}(g_i) = p_i^{k_i}$ est soit z_i la solution du DLP_i ${g_i}^{z_i} = h_i$.
2. on résout le système

$$\begin{cases} x & \equiv & z_1 & \pmod{p_1^{k_1}} \\ & \vdots & & \\ x & \equiv & z_t & \pmod{p_t^{k_t}} \end{cases}$$

Preuve. *Le premier pas prend $O(\sum k_i S_{p_i})$ et le système du théorème des restes chinois prend $O(\log N)$ pas.*

Si x est la solution du système de congruences pour chaque i il existe un w_i entier et $x = z_i + p_i^{k_i} w_i$.

$$\begin{aligned} (g^x)^{N/p_i^{k_i}} &= g^{(z_i + p_i^{k_i} w_i)\frac{N}{p_i^{k_i}}} \\ &= g^{\frac{N z_i}{p_i^{k_i}}} \cdot g^{N w_i} = g^{\frac{N z_i}{p_i^{k_i}}} (g^N)^{w_i} \\ g^N &\equiv 1 \pmod N \text{ parce que } \mathrm{ord}(g) = N \\ &= \left(g^{\frac{N}{p_i^{k_i}}}\right)^{z_i} = g_i^{z_i} = h_i = h^{\frac{N}{p_i^{k_i}}} \end{aligned}$$

Donc $\frac{N}{p_i^{k_i}} x \equiv \frac{N}{p_i^{k_i}} \log_g h \pmod N$.

Parce que $\mathrm{pgcd}\left(\frac{N}{p_i^{k_i}}, \frac{N}{p_j^{k_j}}\right) = 1$ *pour tout $i \neq j$, nous avons qu'il existe les entiers* $\alpha_1, \ldots, \alpha_t$

$$\alpha_1 \frac{N}{p_1^{k_1}} + \cdots + \alpha_t \frac{N}{p_t^{k_t}} = 1$$

donc

$$\sum_{i=1}^{t} \alpha_i \frac{N}{p_i^{k_i}} \cdot x \equiv \sum_{i=1}^{t} \alpha_i \frac{N}{p_i^{k_i}} \log_g(h) \pmod N$$

donc $x = \log_g h \pmod N$.

El Gamal sur $((\mathbb{Z}/n\mathbb{Z})^* = \mathbb{Z}/n\mathbb{Z}, +)$**,** $(\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z}, \cdot)$

Le problème du logarithme discret en $(\mathbb{Z}/n\mathbb{Z})^* = \mathbb{Z}/n\mathbb{Z}, \cdot)$ est difficile, mais pas autant que dans un groupe générique. En $(\mathbb{Z}/n\mathbb{Z})^*, \cdot)$ on connaît des algorithmes sous-exponentiels pour le résoudre, comme l'algorithme de calcul d'index. Dans un groupe générique on doit se contanter des algorithmes de Shanks et Pohling et d'autres, mais qui sont tous exponentiels. Ceci a des conséquences sur la taille du groupe à utiliser pour que le problème DLP reste "difficile". Le nombre premier p doit avoir au minimum 1024 bits pour le $(\mathbb{Z}/n\mathbb{Z})^*$, pour assurer la même sécurité qu'un groupe générique d'ordre ayant 160 bits.

Rappel sur le groupe $(\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z}, \cdot)$

Proposition. *L'anneau* $\mathbb{Z}/n\mathbb{Z}$ *est un corps si, et seulement si,* n *est un nombre premier. Plus, pour tout nombre premier* p*,* $\mathbb{F}_p$ *est le seul corps de cardinal* p*.*

La structure du groupe multiplicatif de $\mathbb{Z}/p\mathbb{Z}$ est celle d'un groupe abelién fini cyclique d'ordre $\phi(p) = p - 1$.

Théorème. [racine primitive/générateur] *Soit* p *premier. Alors il existe* $g \in \mathbb{F}_p^*$ *tel que* $(\mathbb{F}_p^*, \cdot) = \{1, g, \ldots, g^{p-2}\}$*. L'ordre de* g *est* $ord(g) = |\mathbb{F}_p^*| = p - 1$*.*

Corollaire. *(du Th. Lagrange) Soit* p *un entier premier. Soit* $G =< g >\subseteq (\mathbb{F}_p^*, \cdot)$ *et* $ord(g) = |G| = q$*. Alors* $q/p - 1$*.*

Exemple. $\mathbb{F}_{11}^*$ *:* $2^0 = 1$*,* $2^1 = 2$*,* $2^2 = 4$*,* $2^3 = 8$*,* $2^4 \equiv 5$*,* $2^5 \equiv 10$*,* $2^6 \equiv 9$*,* $2^7 \equiv 7$*,* $2^8 \equiv 3$*,* $2^9 \equiv 6$*,* $2^{10} \equiv 1$ *, mais* 2 *n'est pas un générateur pour* $\mathbb{Z}_{17}$*.*

Exemple. *Le groupe multiplicatif* $\mathbb{F}_{11}^*$ *a la cardinalité* 10*, donc les ordres des elements de* $\mathbb{Z}_{11}$ *sont :* $1, 2, 5, 10$*.*

Definition (DLP – Problème du logarithme discret en $(\mathbb{F}_p^* = \mathbb{Z}/p\mathbb{Z}, \cdot)$)**.** *Soit* p *premier,* g *un générateur (racine primitive) de* $\mathbb{F}_p^*$*, et* $h \in \mathbb{F}_p^*$*.*

DLPp : trouver $1 \leq x \leq p - 1$ *tel que* $g^x = h \pmod{p}$*.*

Le nombre x *s'appelle le logarithme discret de* h *en base* g*.*

Remarque. $\log_g \mathbb{F}_p^* \to \mathbb{Z}_{p-1}$ *est une fonction bien définie, et* $\log_g(ab) = \log_g(a) + \log_g(b)$*,* $\forall a, b \in \mathbb{F}_p^*$*.*

Remarque. *Si DLPp a une solution, il a a une infinité.*

Si x_0 est une solution, on a $g^{x_0} = h$. Mais selon le petit théorème de Fermat, $g^{p-1} \equiv 1 \pmod{p}$. Donc $x_0 + k(p-1)$, avec k entier, est aussi une solution.

Exercice. *Coder $A = 01$,$B = 02$, ...,$Z = 26$. Un mot devienne une sequence de nombres ($AB = 0102 = 102$, $OB = 1502$). Soit $p = 150001$, $g = 7$,$a = 113$ tel que $a^k = 7^{113} \equiv 66436 \pmod{p}$. La clé publique est$(p, g, A) = (150001, 7, 66436)$. La clé privée est $a = 113$.*

Soit le message "Hi Mom" : $x_1 = 0809 = 809$ et $x_2 = 131513$. Cryptage : Soit la clé éphémère $k = 1000$. On calcule $y_{11} = g^k \equiv 7^{1000} \equiv 90429 \pmod{150001}$. Donc $y_1 = 90429$. On calcule $y_{21} = x_1 A^k = 809 \cdot (7^{113})^{1000}$, donc $y_{21} \equiv 809 \cdot 66436^{1000} \equiv 809 \cdot 4654 \equiv 15061 \pmod{150001}$ et $y_2 2 = x_2 A^k = 131513 \cdot (7^{113})^{1000} \equiv 57422 \pmod{150001}$. On transmet les deux paires $(90429, 15061)$ et $(90429, 57422)$.

Décryptage : On calcule $(y_{11}^a)^{-1} y_{21} \equiv (90429^{113})^{-1} 15061 \pmod{150001}$, Donc $(y_{11}^a)^{-1} y_{21} \equiv 4654^{-1} \cdot 15061 \equiv 80802 \cdot 15061 \equiv 809 \pmod{150001}$.

Exercice. *A résoudre en utilisant le logiciel de calcul modulaire de*

http ://ptrow.com/perl/calculator.pl

(a) Soit $p = 541$, $g = 2$, $a = 113$ et $k = 101$. Crypter $x = 200$ and $x = 201$ en utilisant ElGamal.

(b) Soit $p = 541$, $g = 2$, $a = 101$. Décrypter les chiffrés ElGamal $y = (54, 300)$ and $y = (54, 301)$.

Exercice. *Soit $\mathbb{F}_{13}^*$ Soit $p = 13$, $g = 2$,$a = 7$. On calcule la clé publique $A = a^k = 2^7 \equiv 11 \pmod{13}$. La clé publique est$(p, g, A) = (13, 7, 11)$. La clé privée est $a = 113$.*

Soit le message : $x = 3$.

Cryptage :

Soit la clé éphémère $k = 5$. On calcule $y_1 = g^k \equiv 2^5 \equiv 6 \pmod{13}$. On calcule $y_2 = x \cdot A^k = 3 \cdot (11)^5$, donc $y_2 \equiv 3 \cdot 7 \equiv 8 \pmod{13}$. On transmet la paire $(6, 8)$.

Décryptage : On calcule $(y_1^a)^{-1} y_2 \equiv (6^7)^{-1} \cdot 8 \equiv 7^{-1} \cdot 8 \equiv 2 \cdot 8 \equiv 3$

Exercice. *Calculer $log_3 525$ en $H = \mathbb{Z}/(p-1)\mathbb{Z}$, $p = 809$.*

Donc $n\lceil\sqrt{p} + 1\rceil = 29$. Pas de bébé : $(j, 525 \cdot (3^j)^{-1} \pmod{p}$ pour $j = 0, \cdots, 28$:

L_2 : (0 , 525) (1 , 175) (2 , 328) (3 , 379) (4 , 396) (5 , 132) (6 , 44) (7 , 554) (8 , 724) (9 , 511) (10 , 440) (11 , 686) (12 , 768) (13 , 256) (14 , 355) (15 , 388) (16 , 399) (17 , 133) (18 , 314) (19 , 644) (20 , 754) (21 , 521) (22 , 713) (23 , 777) (24 , 259) (25 , 356) (26 , 658) (27 , 489) (28 , 163)

Pas de géant : $(j, 3^{j \cdot s} \pmod{p})$ pour $j = 0, \cdots, 28$:

L_1 : (0 , 1) (1 , 99) (2 , 93) (3 , 308) (4 , 559) (5 , 329) (6 , 211) (7 , 664) (8 , 207) (9 , 268) (10 , 644) (11 , 654) (12 , 26) (13 , 147) (14 , 800) (15 , 727) (16 , 781) (17 , 464) (18 , 632) (19 , 275) (20 , 528) (21 , 496) (22 , 564) (23 , 15) (24 , 676) (25 , 586) (26 , 575) (27 , 295) (28 , 81)

Donc$t = 10\lfloor\sqrt{p} + 1\rfloor + 19 = 309$

Attaques : El Gamal en $(\mathbb{F}_p^*)$

Le problème DLP est "facile" à résoudre pour $\mathbb{F}_p^*$: un algorithme du type calcul d'index résout le problème en temps sous-exponentiel.

Remarque. *Parce que* $|\mathbb{F}_p^*| = p - 1$ *pour* p *premier il est donc toujours paire. Donc c'est mieux de choisir* $p = 2q + 1$ *avec* q *un nombre premier large.*

Exemple. *On résout*

$$5448^x = 6909 \text{ dans } \mathbb{Z}_{11251}^*$$

Le nombre premier $p = 11251$ a la propriété que $p - 1$ est divisile par 5^4, et il est facile de vérifier que 5448 a l'ordre exactement 5^4 dans $\mathbb{Z}_{11251}^*$. Le premier pas est la résolution de

$$\left(5448^{5^3}\right)^{x_0} = 6909^{5^3},$$

ce qui se réduit à $11089^{x_0} = 11089$. Cette équation est facile – la solution est $x_0 = 1$, donc notre valeur initiale pour x est $x = 1$.

Le pas suivant est la résolution de

$$\left(5448^{5^3}\right)^{x_1} = (6909 \cdot 5448^{-x_0})^{5^2} = (6909 \cdot 5448^{-1})^{5^2},$$

ce qui se réduit à $11089^{x_1} = 3742$. Remarquons qu'il suffit de vérifier seulement des valeurs de x_1 entre 1 et 4, même si dans le cas où q serait grand, il serait plus convenable d'utiliser un algorithme plus rapide comme celui de Shanks avec les pas de bébé-pas de géant pour résoudre ce problème de logarithme discret. De toutes manières, la solution est $x_1 = 2$, donc la valeur de x est maintenant $x = 11 = 1 + 2 \cdot 5$.

Continuant, nous résolvons maintenant

$$\left(5448^{5^3}\right)^{x_2} = (6909 \cdot 5448^{-x_0 - x_1 \cdot 5})^5 = (6909 \cdot 5448^{-11})^5,$$

ce qui se réduit à $11089^{x_2} = 1$. Donc $x_2 = 0$, ce qui veut dire que la valeur de x reste à $x = 11$.

Le pas final est la résolution de

$$\left(5448^{5^3}\right)^{x_3} = 6909 \cdot 5448^{-x_0 - x_1 \cdot 5 - x_2 \cdot 5^2} = 6909 \cdot 5448^{-11}.$$

Ceci se réduit à résoudre $11089^{x_3} = 6320$, ce qui a la solution $x_3 = 4$. Donc notre réponse finale est

$$x = 511 = 1 + 2 \cdot 5 + 4 \cdot 5^3.$$

Pour vérifier le résultat, on peut calculer

$$5448^5 11 = 6909 \text{ dans } \mathbb{Z}_{11251}.$$

Exemple. *Considérons le problème de logarithme discret*

$$23^x = 9689 \text{ dans } \mathbb{Z}_{11251}.$$

q	e	$g^{(p-1/q^e)}$	$h^{(p-1/q^e)}$	Résoltn $\left(g^{(p-1/q^e)}\right)^x = h^{(p-1/q^e)}$ pour x
2	1	11250	11250	1
3	2	5029	10724	4
5	4	5448	6909	511

TABLE 4.1 – Trois sous-problèmes du logarithme discret

La base 23 est une racine primitive dans $\mathbb{Z}_{11251}$, c'est-à-dire elle a l'ordre 11250. Comme $11250 = 2 \cdot 3^2 \cdot 5^4$ est un produit de nombres premiers petits, l'algorithme de Pohlig-Hellman devrait fonctionner bien. Dans la notation de sa description, on pose

$$p = 11251, \qquad g = 23, \qquad h = 9689, \qquad N = p - 1 = 2 \cdot 3^2 \cdot 5^4.$$

Le premier pas est la résolution de trois sous-problèmes de logarithme discret, comme indiqué dans la table 4.1.

Remarquons que le premier problème est trivial, tandis que le troisième problème est le problème qu'on vient de résoudre dans l'exemple 4.0.1. De toutes manières, les problèmes individuels dans ce pas de l'algorithme peuvent être résolus avec Shanks.

Le second pas est l'utilisation du théorème des restes chinois pour résoudre les congruences simultanées

$$x \equiv 1 \pmod 2, \qquad x \equiv 4 \pmod{3^2}, \qquad x = 511 \pmod{5^4}$$

La plus petite solution est $x = 4261$. On vérifie notre réponse en calculant

$$23^{4261} = 9689 \text{ dans } \mathbb{Z}_{11251}.$$

Exemple. *Illustrer la méthode pas-de-bébé-pas-de-géant (babystep-giantstep) en l'utilisant pour résoudre le problème du logarithme discret*

$$g^x = h \text{ dans } \mathbb{Z}_p^* \text{ avec } g = 9704, h = 13896, \text{ et } p = 17389.$$

Le nombre 9704 a l'ordre 1242 dans $\mathbb{Z}_{17389}^*$. Soit $n = \lceil\sqrt{1242}\rceil + 1 = 36$ et $u = g^{-n} = 9704^{-36} = 2494$. La table 4.2 donne les valeurs de g^k et $h \cdot u^k$ pour $k = 1, 2, \ldots,$. Depuis la table on trouve la collision

$$9704^7 = 14567 = 13896 \cdot 2494^{32} \text{ dans } \mathbb{Z}_{17389}^*.$$

Utilisant le fait que $2494 = 9704^{-36}$, on calcule

$$13896 = 9704^7 \cdot 2494^{-32} = 9704^7 \cdot (9704^{36})^{32} = 9704^{1159} \text{ dans } \mathbb{Z}_{17389}^*.$$

Par conséquent, $x = 1159$ résout le problème $9704^x = 13896$ dans $\mathbb{Z}_{17389}^*$.

k	g^k	$h \cdot u^k$	k	g^k	$h \cdot u^k$	k	g^k	$h \cdot u^k$	k	g^k	$h \cdot u^k$
1	9704	347	9	15774	16564	17	10137	10230	25	4970	12260
2	6181	13357	10	12918	11741	18	17264	3957	26	9183	6578
3	5763	12423	11	16360	16367	19	4230	9195	27	10596	7705
4	1128	13153	12	13259	7315	20	9880	13628	28	2427	1425
5	8431	7928	13	4125	2549	21	9963	10126	29	6902	6594
6	16568	1139	14	16911	10221	22	15501	5416	30	11969	12831
7	14567	6259	15	4351	16289	23	6854	13640	31	6045	4754
8	2987	12013	16	1612	4062	24	15680	5276	32	7583	14567

TABLE 4.2 – Pas-de-bébé-pas-de-géant pour résoudre $9704^x \equiv 13896 \pmod{17389}$

El Gamal Courbes elliptiques

On s'interesse aux courbes elliptiques élémentaires en forme d'équation de Weierstrass.

Definition. *Soit $a, b \in \mathbb{R}$. Une courbe elliptique E(a,b) est une courbe algébrique du troisième degré (cubique) (c.à-d l'ensemble des points $(x, y) \in \mathbb{R} \times \mathbb{R}$ et un point à l'infini) vérifiant l'équation : $y^2 = x^3 + ax + b \pmod{p}$ où $a, b \in \mathbb{R}$ tels que $\delta := 16(4a^3 + 27b^2) \neq 0$, c.à-d où le second membre, polynôme du 3ème degré en x, n'a pas de solution double.*

La quantité δ s'appelle le discriminant de la courbe. Le graphe d'une courbe elliptique depend du signe du discriminant :

- positif : il présente deux composantes (Ex : $y^2 = x^3 - x$, $\delta := 64$, le polynôme cubique $x^3 + ax + b$ a exactement trois racines réelles distinctes)
- négatif :il présente deux composantes (Ex : $y^2 = x^3 - x + 1$, $\delta := -368$, le polynôme cubique $x^3 + ax + b$ a exactement une racine réelle)

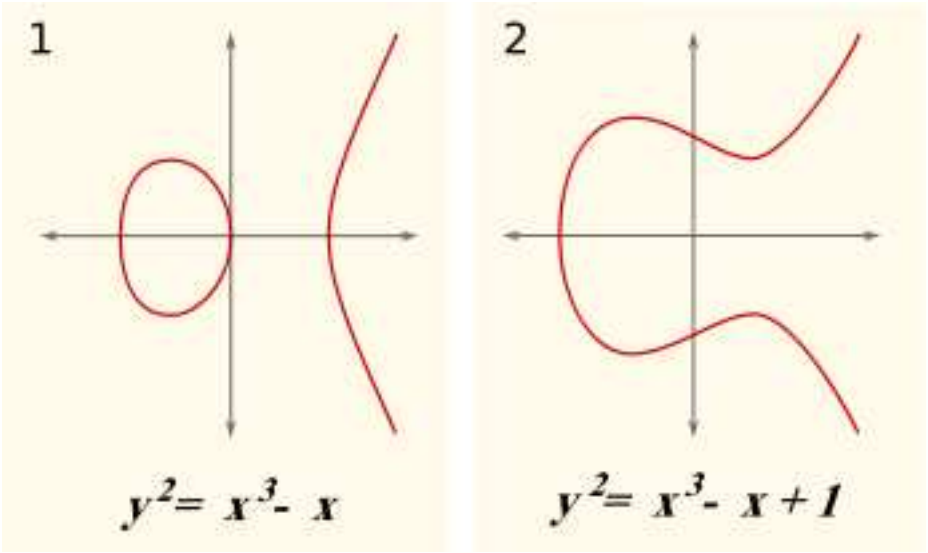

FIGURE 4.1 – Courbes elliptiques

"Addition" sur les courbes elliptiques :

Théorème (Bézout)**.** *Deux courbes algébriques projectives planes C,D de degrés m et n, définies sur un corps algébriquement clos et sans composante irréductible commune, ont exactement mn points d'intersections, comptés avec multiplicités.*

Par deux points distincts P, Q situés sur une courbe elliptique passe une droite bien définie. Comme consequence une droite sécante passant par deux points d'une courbe elliptique recoupe la courbe en un troisième point (distinct ou non). Ce point est consideré la "somme P+Q".(voir figure 4.2).

Lemme 1. *Soit $a, b \in \mathbb{R}$. Les points rationnels d'une courbe elliptique E(a,b) forment un groupe abélien $E(a, b)$ avec l'operation d'addition suivante : Soit $P = (x_1, y_1)$, $Q = (x_2, y_2) \in E(a, b)$.*

- *Si $P = O$ alors $P + Q = P$*
- *Si $Q = O$ alors $P + Q = Q$*
- *Si $x_2 = x_1$ et $y_2 = -y_1$ alors $P + Q = O$*
- *Autrement $P+Q = (x_3, y_3)$ avec $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$, où $\lambda = \frac{3x_2 - 1 + a}{2y_1}$ si $P = Q$ et $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ sinon.*

Donc

- Le point à l'infini O est l'élément neutre : $P + O = O + P = P$
- L'inverse de $P = (x_1, y_1)$ est $Q(x_1, -y_1)$, $P + Q = Q + P = O$

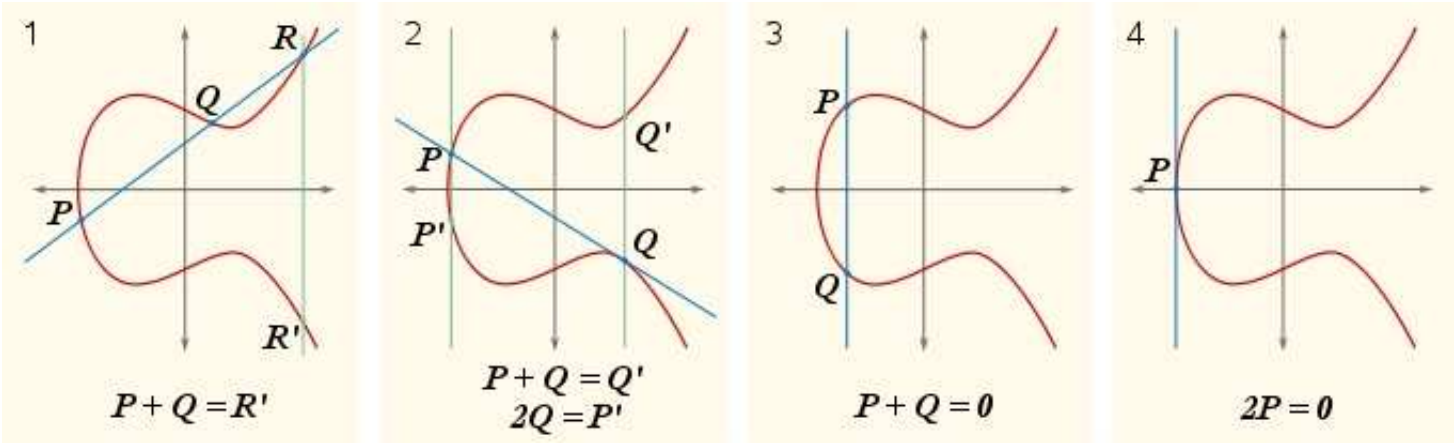

FIGURE 4.2 – Addition courbes elliptiques

- $P + Q = (x_3, y_3)$ avec $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$, où $\lambda = \frac{3x_1^2+a}{2y_1}$ si $P = Q$ et $\lambda = \frac{y_2-y_1}{x_2-x_1}$ sinon.
- La propriété d'associativité peut être directement vérifie.

Pour la multiplication scalaire sur $E(a, b)$ on utilise la notation suivante : P, $[2]P = P + P$, $[3]P = [2]P + P$,... , $[n]P = [n-1]P + P$. Les résultats précédents sont encore valables lorsque le corps de définition de la courbe elliptique est $\mathbb{F}_p^k$ avec $p > 3$. On denote le groupe correspondant par $E(a, b, p)$. Pour les applications en cryptographie lies asymetrique il nous faut : une manière efficace de calculer $[n]P$, un sous-groupe cyclique de $E(a, b, p)$, connaître/approximer l'ordre du groupe $E(a, b, p)$ et les points rationnels, évaluer la difficulté du problème du logarithme discret/factorisation.

Pour calculer nP il y a l'algorithme "Double and Add" qui est très similaire à celui de l'exponentiation pour RSA avec additions à la place des multiplications. Pour calculer On decompose $[k]P$ d'abord en binaire $k = \sum_{i \leq t} k_i 2^i$, où $k_i \in 0, 1$ et $t = [log_2 k], k_t = 1$. Après

$$[2](....([2]([2]([2]([2]P + [k_{(l-1)}]P) + [k_{(l-2)}]P) + [k_{(l-3)}]P) + \dots) \cdots + [k_1]P) + [k_0]P$$

$$= [2^l]P + [k_{(l-1)}2^{l-1}]P + \cdots + [k_1 2]P + [k_0]P$$

La ordre du groupe pour $K = \mathbb{F}_{p^k}^* = \frac{\mathbb{Z}}{p^k\mathbb{Z}}$ avec $p > 3$ est donnée par le résulta suivant :

Théorème (Hasse). *Soit $q = p^k$ alors $q + 1 - 2\sqrt{q} \leq |E(a, b, p)| \leq q + 1 + 2\sqrt{q}$*

et l'algorithme de Schoof permet de déterminer le nombre de points sur une courbe elliptique.

Théorème (Tsfasman-Voloch-Ruck Théorème de structure). *Soit $q = p^k$. Le groupe $E(a, b, p)$ est soit*

- *un groupe cyclique soit*
- *isomorphe avec un produit de deux groupes cycliques $\mathbb{Z}/u\mathbb{Z}/ \times \mathbb{Z}/v\mathbb{Z}$ où $u|pgcd(v, q-1)$*

Système de cryptage elliptique à clé publique ElGamal sur $E(a, b, p)$

Création de paramètre publique

Une entité "de confiance" choisit et publie un nombre premier p (large), une courbe elliptique E sur $\mathbb{Z}_{p^k}$ et un point P dans $E(a, b, p)$

Création de clé

Choisir une clé privée n_A.
Calculer $Q_A = n_A P$ dans $E(a, b, p)$
Publier la clé publique Q_A

Cryptage

Choisir le texte en clair $x \in E(a, b, p)$.
Choisir une clé éphémère k
Utiliser la clé publique d'Alice Q_A pour calculer $y_1 = kP \in E(a, b, p)$ et $y_2 = x + kQ_A \in E(a, b, p)$.
Envoyer le texte chiffré (y_1, y_2).

Décryptage

Calculer $y_2 - n_A y_1 \in E(a, b, p)$.

Cette valeur est égale à x

Le système de cryptage elliptique à clé publique El Gamal fonctionne très bien, mais il présente quelques difficultés d'ordre pratique :

1. Il n'ya pas de manière évidente pour attacher des messages en clair à des points de $E(\mathbb{F}_p)$.
2. Le système de cryptage elliptique El Gamal a un taux d'expansion de message de 4-pour-1, tandis que le système de cryptage sur $\mathbb{F}_p$ El Gamal a un taux d'expansion de message de 2-pour-1.

Les avantages de l'utilisation du problème du logarithme discret sur des courbes elliptiques sont gue DLP est "difficile" sur le groupe associé à une courbe elliptique (c.à.d. qu'il n'y a pas d'attaque connue de complexité sous-exponentielle) et que les clés peuvent être plus petites pour assurer un niveau de sécurité équivalent : RSA 1024bits versus ECC 160bits. Pour le problème de factorisation (c.à-d. la décomposition en produit de facteurs entiers premiers) sur les courbes elliptiques il y a parmi d'autres l'algorithme probabiliste de Lenstra qui a une complexité sous-exponentielle.

Class	Taille typique de clé	Exemples
Clé publique	1024-2048 bits (non courbe-elliptique) 163-233 bits (courbe elliptique)	Diffie-Hellman, ElGamal, DSA
Clé symétrique	128-256 bits	DES, AES
Hash	N/A	SHA, MD5

TABLE 4.3 – Types d'algorithmes et leur tailles de clé actuelles