Chiffrement ElGamal et attaques sur le logarithme discret
Option agrégation C

Christophe Ritzenthaler

December 12, 2007

Abstract

Résumé : on décrit ici le chiffrement d'ElGamal et différentes attaques sur le problème du logarithme discret.

Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury demande que la discussion soit accompagnée d'exemples traités par ordinateur. Il est souhaitable que vous organisiez votre présentation comme si le jury n'avait pas connaissance du texte. Le jury aura néanmoins le texte sous les yeux pendant votre exposé.

1 Introduction

Il y a à peine quelques années, le problème de la sécurité des transmissions de données semblait être l'apanage des seuls militaires. Ce n'est plus le cas, avec l'essor des techniques numériques dans le commerce (Internet, les cartes de crédit, les télécommunications, les décodeurs de télévision, etc.) Les méthodes empiriques traditionnelles se sont révélées trop fragiles ; elles reposaient souvent sur le schéma suivant : on choisit un livre, une fois pour toutes, et on considère la permutation des vingt-six lettres de l'alphabet définie par les vingt-six premières lettres de ce livre. Le codage d'un texte consiste alors à appliquer cette permutation σ au texte à coder, et le décodage à appliquer la permutation réciproque σ^{-1} au texte à décoder. En numérique, si par exemple le message à transmettre est codé sur 64 bits, on peut employer cette technique en considérant une permutation $\sigma \in \Sigma_{64}$. C'est ce genre d'idées qui est sousjacent au procédé DES, dû à IBM, et qui est encore le plus utilisé en informatique. Le talon d'Achille de ce genre de cryptosytème est le suivant : si quelqu'un sait coder, il sait aussi décoder, car il est très facile de trouver la permutation réciproque d'une permutation sur $26, 64, 128$ ou même 256 lettres. C'est pourquoi l'apparition de cryptosystèmes dits à clé publique, à la fin du siècle dernier, a fait sensation. Ces cryptosystèmes, comme le nom l'indique, sont tels que le codage est public : tout le monde connaît l'algorithme pour coder le message. Mais on ne peut pas en déduire le décodage.
En fait, cela revient à construire une permutation σ d'un ensemble à N éléments, mais ici N est gigantesque (de l'ordre de 10^{500}, par exemple.) On ne peut même pas écrire la liste de ces éléments, et la méthode habituelle pour trouver la permutation réciproque d'une permutation donnée ne peut plus s'appliquer.

Nous présentons ici un concurrent de RSA appelé ElGamal et qui repose sur le problème du logarithme discret dans $(\mathbb{Z}/p\mathbb{Z})^*$.

2 Chiffrement ElGammal

Soit p un grand nombre premier. On sait que le groupe multiplicatif $H = (\mathbb{Z}/p\mathbb{Z})^*$ est cyclique. Soit α un générateur de ce groupe. On définit le système de chiffrement suivant :

Ensemble des messages $\mathcal{P} = (\mathbb{Z}/p\mathbb{Z})^*$;
Ensemble des chiffrés $\mathcal{C} = H \times (\mathbb{Z}/p\mathbb{Z})^*$;
Clé secrète : $a \in [0, \#H - 1]$;
Clé publique : p, α et $\beta \equiv \alpha^a \pmod p$;
Chiffrement : $\mathcal{E}_e(x) \equiv (y_1, y_2) \equiv (\alpha^k \pmod p, x\beta^k \pmod p)$ pour un $k \in [0, \#H - 1]$ aléatoire, différent pour chaque x;
Déchiffrement : $\mathcal{D}_d((y_1, y_2)) \equiv y_2 y_1^{-a} \pmod p$.

La sécurité du chiffrement repose sur la difficulté (présumée) du calcul de a sachant β. Ce problème est appelé *problème du logarithme discret* (DLP) dans H. On ne connaît pas d'algorithme rapide pour résoudre ce problème mais nous allons présenter ci-dessous certaines attaques qui améliore l'algorithme naïf de recherche exhaustive.

3 Attaques sur le logarithme discret

3.1 Meilleure attaque générique

Nous présentons ici l'algorithme des pas de bébé, pas de géant. Soit un groupe cyclique G d'ordre n et le DLP dans G : $\alpha^x = y$. On pose $m = \lceil \sqrt{n} \rceil$ et on écrit $x = qm + r$ avec $0 \leq r, q < m$. On a

$$\alpha^{qm+r} = y \Rightarrow (\alpha^m)^q = y\alpha^{-r}.$$

On calcule d'abord les *pas de bébé*

$$B = \{(y\alpha^{-r}, r),\ 0 \leq r < m\}.$$

Si on trouve $(1, r)$ alors $y = \alpha^r$. Sinon on calcule $\delta = \alpha^m$. On teste alors pour $q = 1, 2, \ldots, m$ si l'élément δ^q est la première composante d'un élément de B. Dès que c'est le cas on a une solution du DLP. δ^q est appelé *pas de géant*.
Il est facile de voir que cet algorithme est en $\mathcal{O}(\sqrt{\#G})$.

Remarque 1. *Cette méthode doit stocker $\mathcal{O}(\sqrt{\#G})$ éléments. Il existe un algorithme probabiliste ρ Pollard qui nécessite peu de mémoire. La fonction 'pseudo-aléatoire' utilisée est $f : G \to G$ définie par*

$$f(\beta) = \begin{cases} \alpha\beta & \text{if } \beta \in G_1, \\ \beta^2 & \text{if } \beta \in G_2, \\ y\beta & \text{if } \beta \in G_3 \end{cases}$$

où G_1, G_2, G_3 forment une partition de G. On choisit $x_0 \in \{1, \ldots, n\}$ puis on calcule $\beta_0 = \alpha^{x_0}$ et

$$\beta_{i+1} = f(\beta_i).$$

3.2 L'attaque de Pohlig-Hellman

Nous montrons ici que le DLP dans un groupe cyclique G d'ordre n peut être réduit à des DLP dans des sous-groupes d'ordre premiers si on connaît la factorisation de n

$$n = \#G = \prod_p p^{e(p)}.$$

Soit donc le DLP dans G : $\alpha^x = y$.

1. Réduction aux puissances de nombres premiers. Pour chaque diviseur premier p de n, on pose
$$n_p = n/p^{e(p)}, \quad \alpha_p = \alpha^{n_p}, \quad y_p = y^{n_p}.$$
Si on sait alors résoudre les DLP
$$\alpha_p^x = y_p$$
dans les sous-groupes d'ordre une puissance de p, grâce au théorème des restes chinois on déduit la solution x.

2. Réduction au facteur premier. On suppose maintenant que $\#G = p^e$ pour un premier p et on souhaite résoudre un DLP dans G. On a $x < p^e$ et on peut donc écrire
$$x = x_0 + x_1 p + \ldots x_{e-1} p^{e-1}, \quad 0 \leq x_i < p, \quad 0 \leq i \leq e-1.$$
On va montrer que les x_i sont les solutions de DLP dans un groupe d'ordre p. En effet
$$p^{e-1} x = x_0 p^{e-1} + p^e (x_1 + x_2 p + \ldots + x_{e-1} p^{e-2}).$$
Soit
$$(\alpha^{p^{e-1}})^{x_0} = y^{p^{e-1}}.$$
L'équation montre que x_0 est la solution du DLP dans un groupe d'ordre p. Les autres coefficients sont déterminés récursivement de la même manière.

Cet algorithme montre que la complexité du calcul du DLP dans G est dominée par la complexité du calcul du DLP dans le sous-groupe d'ordre le plus grand diviseur premier de $\#G$.

3.3 Méthode du calcul de l'index

Quand $G = (\mathbb{Z}/p\mathbb{Z})^*$ il existe un algorithme spécifique appelé *méthode de l'index*. Nous en décrivons une version simplifiée.

L'idée. Soit p un nombre premier, α un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$ et $y \in \{1, \ldots, p-1\}$. Nous voulons résoudre $\alpha^x \equiv y \pmod p$. Soit B un nombre et

$$F(B) = \{q \in \mathbb{P},\ q \leq B\}.$$

Cet ensemble est appelé *base des facteurs* et un entier b est dit *B-lisse* s'il n'a que des facteurs premiers dans $F(B)$. On procède en deux étapes. Tout d'abord on calcule les DLP

$$\alpha^{x(q)} \equiv q \pmod p$$

pour tout $q \in F(B)$. Puis on détermine $\delta \in \{1, \ldots, p-1\}$ tel que $y\alpha^\delta \pmod p$ est B-lisse

$$y\alpha^\delta \equiv \prod_{q \in F(B)} q^{e(q)} \pmod p.$$

Ceci permet alors de retrouver x.

DLP des éléments de la base. Pour calculer le DLP des éléments de la base, on choisit aléatoirement $z \in \{1, \ldots, p-1\}$ et on calcule $\alpha^z \pmod p$. On vérifie si ces nombres sont B-lisses. Si c'est le cas, on calcule leur décomposition

$$\alpha^z \pmod p = \prod_{q \in F(B)} q^{f(q,z)}.$$

Chaque vecteur $(f(q,z))_{q \in F(B)}$ est appelé une *relation*. Si on obtient assez de relations, on peut trouver les DLP des éléments de la base en résolvant un système linéaire.

Remarque 2. *On peut montrer qu'une telle attaque est en* $L_p(1/2, C) = \exp(C \log(p)^{1/2} (\log\log p)^{1/2})$ *pour une certaine constante* C. *Ainsi si* $G = (\mathbb{Z}/p\mathbb{Z})^*$, *on peut résoudre le logarithme discret lorsque la taille de* p *est plus petite que* 1024 *bits.*

4 Suggestions

Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.

1. On pourra commenter la construction des éléments p et α.
2. On pourra bien sûr implenter le chiffrement ElGammal.
3. Pourquoi l'attaque par énumération n'est pas possible ?
4. Commenter la méthode des pas de bébé, pas de géant. Expliquer l'algorithme ρ-Pollard.
5. Compléter les trous dans la méthode de Polhig-Hellman.
6. Déduire de l'attaque de Polhig-Hellman qu'il n'est pas utile de prendre $H = (\mathbb{Z}/p\mathbb{Z})^*$ dans ElGamal mais qu'on peut prendre un sous-groupe cyclique d'ordre q où q est le plus grand facteur premier de $p-1$. Comment effectue-t-on pratiquement la construction de (p, q) ?
7. Compléter les trous dans l'attaque du calcul de l'index.
8. En regardant toutes ces attaques, comment doivent-être choisis les paramêtres d'ElGammal ?