Logaritmo Discreto

Este documento no pretende ser un curso exhaustivo, en el mejor de los casos, únicamente puede considerarse como un conjunto de notas complementarias en alguna asignatura. Por supuesto, todo documento es susceptible de mejora, cualquier sugerencia o comunicación de error u omisión será bienvenida.

La seguridad de muchos métodos criptográficos está basada en la complejidad de computar el logaritmo discreto, esto es, dado un grupo $\langle G, \oplus \rangle$, un generador α y un valor β obtenido como composición del generador un número determinado de veces, el logaritmo discreto de β en base α es el valor k tal que $\beta = \alpha^k$. Entre los métodos criptográficos que consideran este problema para conseguir seguridad computacional están el esquema Diffie-Hellman de intercambio público de clave, o los esquemas de ElGamal de cifrado o firma. Los esquemas criptográficos basados en curvas elípticas consideran este mismo problema pero considerando un grupo de valores discretos obtenidos a partir de una curva elíptica.

Pollard-rho

El algoritmo *bay-step giant-step* de Shank para el cálculo del logaritmo discreto tiene complejidad temporal y espacial de $\mathcal{O}(\sqrt{p})$. Un algoritmo heurístico con la misma complejidad temporal es el de Pollard-rho. Este algoritmo es una versión del método con el mismo nombre para la factorización de enteros y no tiene las necesidades de memoria del algoritmo propuesto por Shank.

Para describir el método, consideraremos un primo p, el grupo multiplicador $\mathbb{Z}_p^*$, un generador α de $\mathbb{Z}_n$ y un elemento β del grupo, del cual se busca obtener el valor t tal que $\alpha^t \bmod p = \beta$.

El algoritmo considera la división del grupo $\mathbb{Z}_p^*$ en tres conjuntos disjuntos S_1, S_2 y S_3 del mismo tamaño aproximado y donde el elemento neutro no esté en S_2. La elección de los conjuntos debe hacerse teniendo en cuenta que, para cualquier elemento del grupo, la decisión de en qué conjunto se encuentra debe de estar basada en una regla fácil de comprobar (de coste temporal constante).

A partir de $x_0 = 1$, se construye una secuencia de elementos del grupo $x_1, x_2, \ldots x_k$ de acuerdo con:

$$x_{i+1} = \begin{cases} \beta x_i \bmod p & \text{si} x_i \in S_1 \\ x_i^2 \bmod p & \text{si} x_i \in S_2 \\ \alpha x_i \bmod p & \text{si} x_i \in S_3 \end{cases} \quad (1)$$

En paralelo, esta secuencia está acompañada de otras dos $a_1, a_2 \dots a_k$ y $b_1, b_2 \dots b_k$ definidas a partir de $a_0 = b_0 = 0$ como:

$$a_{i+1} = \begin{cases} a_i & \text{si} x_i \in S_1 \\ 2a_i \text{ mód } p-1 & \text{si} x_i \in S_2 \\ a_i + 1 \text{ mód } p-1 & \text{si} x_i \in S_3 \end{cases} \quad (2)$$

y

$$b_{i+1} = \begin{cases} b_i + 1 \text{ mód } p-1 & \text{si } x_i \in S_1 \\ 2b_i \text{ mód } p-1 & \text{si } x_i \in S_2 \\ b_i & \text{si } x_i \in S_3 \end{cases} \quad (3)$$

que cumplen que, para todo valor x_i:

$$x_i = \alpha^{a_i} \beta^{b_i} \text{ mód } p$$

El algoritmo se basa en la búsqueda de dos elementos x_i y x_{2i} del grupo que sean iguales, en ese caso:

$$\alpha^{a_i} \beta^{b_i} \equiv \alpha^{a_{2i}} \beta^{b_{2i}} \pmod{p}$$

por lo que

$$\beta^{b_i} \beta^{-b_{2i}} \equiv \alpha^{a_{2i}} \alpha^{-a_i} \pmod{p}$$

teniendo en cuenta que $\beta = \alpha^t$ mód p y que t es justo el logaritmo de β en base α módulo p, el valor que estamos intentando calcular:

$$\begin{array}{ll} \beta^{b_i} \beta^{-b_{2i}} \equiv \alpha^{a_{2i}} \alpha^{-a_i} & \pmod{p} \\ \beta^{(b_i - b_{2i})} \equiv \alpha^{(a_{2i} - a_i)} & \pmod{p} \\ \alpha^{t(b_i - b_{2i})} \equiv \alpha^{(a_{2i} - a_i)} & \pmod{p} \end{array}$$

y teniendo en cuenta que p es primo, α es generador de $\mathbb{Z}_p^*$ y que el orden del grupo multiplicativo es $p-1$, tenemos que:

$$t(b_i - b_{2i}) \equiv (a_{2i} - a_i) \pmod{p-1}$$

por lo que puede calcularse el logaritmo discreto si es posible calcular el inverso de $(b_i - b_{2i})$ módulo $p-1$:

$$t \equiv (a_{2i} - a_i)(b_i - b_{2i})^{-1} \pmod{p-1}$$

Existe una probabilidad de que $b_i \equiv b_{2i} \pmod{p-1}$ y que por lo tanto no exista el inverso, en ese caso, distintos textos recomiendan repetir una ejecución considerando valores aleatorios de a_0 y b_0 e inicializando $x_0 = \alpha^{a_0}\beta^{b_0} \bmod p$. Obviamente, en caso de trabajar con un subgrupo de $\mathbb{Z}_p^*$ es necesario operar módulo el orden del subgrupo. El Algoritmo 1 describe este método.

Algoritmo 1 Algoritmo de Pollard-rho para el cálculo del logaritmo discreto.

Entrada: Un entero p
Entrada: Un generador α de un subgrupo de $\mathbb{Z}_p^*$ de orden o
Entrada: Un valor β de $< \alpha >$
Salida: k tal que $\beta \equiv \alpha^k \pmod{p}$ //Logaritmo discreto de β en base α módulo p

1: **Método**
2: $a = b = aa = bb = 0$ // puede sustituirse por valores aleatorios
3: $x = xx = 1$ // en cuyo caso $x = \alpha^a\beta^b \bmod p$ y $xx = \alpha^{aa}\beta^{bb} \bmod p$
4: $i = 1$
5: **Mientras** $i < p$ **hacer**
6: $x = f(x, a, b, p, o)$
7: $xx = f(xx, aa, bb, p, o)$
8: $xx = f(xx, aa, bb, p, o)$
9: **Si** $x == xx$ **entonces**
10: **Si** $mcd(b - bb, o) \neq 1$ **entonces**
11: **Devolver** $False$
12: **FinSi**
13: **Devolver** $(aa - a)(b - bb)^{-1} \bmod o$
14: **FinSi**
15: **FinMientras**
16: **Devolver** $False$
17: **FinMétodo.**

La función pseudoaleatoria implementa las funciones descritas en las ecuaciones 1, 2 y 3, pero puede modificarse para considerar distintas opciones, siendo importante ser consistente y modificar convenientemente los valores de a y b para que en todo momento se cumpla que:

$$x = \alpha^a\beta^b \bmod p$$

Ejemplo 1. *Utilizaremos el algoritmo de Pollard-rho para calcular el logaritmo discreto de* 804 *en base* 9 *módulo* 853*, donde el subgrupo generado por* 9 *es de tamaño* 71*. La*

siguiente tabla resume la ejecución del algoritmo cuando se considera la función pseudo-aleatoria descrita por las ecuaciones 1, 2 u 3, :

i	x_i	a_i	b_i	x_{2i}	a_{2i}	b_{2i}
1	804	0	1	412	1	1
2	412	1	1	826	3	2
3	850	2	2	192	8	4
4	826	3	2	628	9	5
5	610	4	2	92	36	20
6	192	8	4	192	1	42

Para calcular el logaritmo discreto, obtenemos $(42-4)^{-1} \bmod 71 = 43$ *y calculamos* $(8-1)43 \bmod 71 = 17$. *En efecto,* $9^{17} \bmod 853 = 804$.

Ejercicios

Ejercicio 1.

Utilice el algoritmo de Pollard-rho para calcular el logaritmo discreto de 649 *en base* 7 *módulo* 809, *sabiendo que el subgrupo generado por* 7 *es de tamaño* 101.

Ejercicio 2.

Utilice el algoritmo de Pollard-rho para calcular el logaritmo discreto de 827 *en base* 16 *módulo* 857, *sabiendo que el subgrupo generado por* 16 *es de tamaño* 107.
