ALGORITHME POUR FACTORISATION DES ENTIERS : POLLARD P-1

Niveau :
BUT 1 SCIENCE DE DONNEES

Réalisé par :
Papa Mar FALL
Ibrahima BODIAN
AWA Sokhna BOUSSA SYLLA
Viny Presty NAKAVOUA NSALOUMOUNA

Enseignant référent
Axel DURBET

TABLES DES MATIERES

1 Introduction

« La cryptographie est la discipline de la cryptologie qui s'attache à l'étude de la sécurité, c'est-à-dire de la confidentialité, l'authenticité et l'intégrité, de messages échangés entre deux ou plusieurs entités. La cryptographie est divisée en deux branches : la cryptographie à clé privée (ou symétrique) et la cryptographie à clé publique (ou asymétrique). Dans la cryptographie à clé privée, la sécurité des messages est assurée par un secret partagé par toutes les entités impliquées. Dans la cryptographie à clé publique, il existe deux clés (une privée et une publique) et la sécurité des messages repose sur la difficulté de trouver la clé privée à partir de la clé publique. Pour cela, des fonctions à sens unique sont utilisées, c'est-à-dire des fonctions facilement calculables mais dont l'inverse est très difficile à calculer. La sécurité de la majorité des cryptosystèmes à clé publique actuels est basée sur la difficulté de deux problèmes : la factorisation d'entiers et le calcul de logarithme discret. » (Bouvier, 2015)
Dans ce projet, nous allons développer l'un des outils de factorisation d'entiers développé par le mathématicien britannique John Michael Pollard. Il s'agit de l'algorithme de factorisation des entiers POLLARD P-1.

2 Principe de la méthode de factorisation pollard p-1

« Soit n un entier divisible par un nombre premier p, avec n ≠ p.
D'après le petit théorème de Fermat, on a :
$a^{p-1} \equiv 1[p]$

pour *a* premier avec *p*. Ici *(≡)* désigne la congruence sur les entiers.
Cela implique que pour tout multiple *M* de *p* – 1, on a :
$a^M - 1 \equiv 0[p]$ ce qui implique $a^M \equiv 1[p]$

Si *p* – 1 est B-superlisse pour un certain seuil *B*, alors *p* – 1 divise le plus petit commun multiple des entiers de 1 à *B*. Donc, si l'on pose *M* = ppcm(1, … , *B*), on a

Autrement dit, *p* divise $a^M - 1$ et donc le pgcd de *n* et $a^M - 1$ est supérieur ou égal à *p*. En revanche, il est possible que ce pgcd soit égal à *n* lui-même auquel cas, on n'obtient pas de facteur non trivial. » (WIKIPEDIA, s.d.)

3 Présentation de l’algorithme

« L'algorithme de Pollard p-1 fonctionne comme suit :

1. *Obtenir n, un entier impair à factoriser.*
2. *Laisser a = 2 et i = 2.*
3. *Calculer a = ai modulo n.*
4. *Calculer d = pgcd(a - 1, n).*
5. *Si 1 < d < n, alors produire d comme un facteur de n.*
6. *Si d = 1,*
7. *Sinon i = i + 1, et retourner à l'étape 3. »* (Maya , Mohammad , & Rachmawat, 2020)

Par exemple, permettez-nous de factoriser n = 209. Soit a = 2 et i = 2. Calculons a = 2^2 mod 209 = 4. Calculons d = pgcd(4 - 1, 209) = 1. Étant donné que d = 1, calculons i = 2 + 1 = 3, et passons à l'étape 3. Calculons a = 4^3 mod 209 = 64. Calculons d = pgcd(64 - 1, 209) = 1. Étant donné que d = 1, calculons i = 3 + 1 = 4, et passons à l'étape 3. Calculons a = 64^4 mod 209 = 159. Calculons d = pgcd(159 - 1, 209) = 1. Étant donné que d = 1, calculons i = 4 + 1 = 5, et passons à l'étape 3. Calculons a = 159^5 mod 159 = 144. Calculons d = pgcd(144 - 1, 209) = 11. Étant donné que 1 < d < 209, d = 11 est un facteur de 209. L'autre facteur de 209 est 209/11 = 19.

4 Modification de l’algorithme de factorisation pollard p-1

L'algorithme de Pollard p-1 d'origine est conçu pour factoriser des entiers en deux facteurs premiers distincts. Cependant, dans le contexte de la cryptographie RSA, il est souvent nécessaire de factoriser un nombre entier en plusieurs facteurs premiers, plutôt que seulement deux. De plus, il est souvent nécessaire de s'assurer que tous ces facteurs sont premiers.

Par conséquent, l'algorithme de Pollard p-1 a été modifié pour répondre à ces exigences spécifiques de la factorisation du module RSA. Cette modification permet à l'algorithme de gérer la factorisation d'un nombre entier en deux ou plusieurs facteurs premiers et de garantir que tous ces facteurs sont premiers en utilisant l'algorithme de test de primalité de Fermat. En adaptant l'algorithme de cette manière, il devient plus adapté à la factorisation des modules RSA et peut être utilisé efficacement dans le contexte de la cryptographie.

5 Explication du code

Ce code Python met en œuvre une version modifiée de l'algorithme de Pollard p-1 pour factoriser un nombre entier n. Voici une explication ligne par ligne :

5.1 Définition de la fonction `is_prime(n)` :

Cette fonction vérifie si un nombre est premier. Si le nombre est inférieur à 2 ou s'il est pair (excepté 2), la fonction retourne False. Sinon, elle vérifie si le nombre est divisible par des entiers impairs jusqu'à sa racine carrée. Si c'est le cas, la fonction retourne False. Sinon, elle retourne True.

5.2 Définition de la fonction `pgcd(a, b)` :

Cette fonction calcule le PGCD (plus grand commun diviseur) de deux nombres `a` et `b`. Elle utilise l'algorithme d'Euclide étendu pour trouver le PGCD ainsi que les coefficients `x` et `y` tels que `a*x + b*y = PGCD(a, b)`.

5.3 Définition de la fonction `modexp(a, i, n)` :

Cette fonction calcule `a^i mod n`, c'est-à-dire la puissance `i` de `a` modulo `n`, où `n` est le modulus. Elle utilise l'exponentiation modulaire pour optimiser le calcul de grandes puissances.

5.4 Définition de la fonction `pollard(n, iterations=1)` :

Cette fonction prend en entrée un nombre entier `n` à factoriser ainsi qu'un paramètre optionnel `iterations` (par défaut à 1) qui spécifie le nombre d'itérations à effectuer. Elle implémente l'algorithme de Pollard p-1 modifié pour factoriser le nombre `n`. L'algorithme utilise des itérations pour tenter de factoriser `n`. Il effectue des calculs de `modexp` et de `pgcd` pour trouver des diviseurs de `n`. Il accumule les facteurs trouvés dans une liste `factor`. Une fois `n` complètement factorisé, la liste des facteurs est retournée.

Enfin, le test de la fonction est effectué en utilisant un nombre spécifique `number` et les facteurs obtenus sont affichés.

6 Implémentation du code et résulats : cas du nombre 123469

Factoring 123469

a = 2 ^ 2 mod 123469 = 4

d = gcd(4 - 1, 123469) = 1

Factoring 123469

a = 4 ^ 3 mod 123469 = 64

d = gcd(64 - 1, 123469) = 1

Factoring 123469

a = 64 ^ 4 mod 123469 = 108901

d = gcd(108901 - 1, 123469) = 1

Factoring 123469

a = 108901 ^ 5 mod 123469 = 32697

d = gcd(32697 - 1, 123469) = 1

Factoring 123469

a = 32697 ^ 6 mod 123469 = 41441

d = gcd(41441 - 1, 123469) = 37 is a factor

Now, factoring 123469 / 37 = 3337.0

Factoring 3337

a = 41441 ^ 2 mod 3337 = 2801

d = gcd(2801 - 1, 3337) = 1

Factoring 3337

a = 2801 ^ 3 mod 3337 = 1883

d = gcd(1883 - 1, 3337) = 1

Factoring 3337

a = 1883 ^ 4 mod 3337 = 1820

d = gcd(1820 - 1, 3337) = 1

Factoring 3337

a = 1820 ^ 5 mod 3337 = 900

d = gcd(900 - 1, 3337) = 1

Factoring 3337

a = 900 ^ 6 mod 3337 = 2593

d = gcd(2593 - 1, 3337) = 1

Factoring 3337

a = 2593 ^ 7 mod 3337 = 1563

d = gcd(1563 - 1, 3337) = 71 is a factor

Now, factoring 3337 / 71 = 47.0

Factoring 47

47 is already a prime, thus it is a factor

Factors of 123469 : [37, 47, 71]

7 Conséquences cryptologiques

L'efficacité de cet algorithme est liée à la forme des nombres premiers composant l'entier à factoriser, plus précisément à l'existence d'un facteur premier p tel que $p - 1$ soit B-superlisse. En conséquence, les systèmes de chiffrement à clé publique fondés sur la difficulté de la factorisation, comme RSA, imposent d'utiliser des nombres premiers n'ayant pas cette propriété pour un seuil B trop petit.

8 Limites de l'algorithme

L'algorithme de Pollard p-1, bien qu'utile dans de nombreux cas, présente également certaines limites. Voici quelques-unes des limitations courantes de cet algorithme :

Limitation du domaine d'application : L'algorithme de Pollard p-1 est efficace pour factoriser les entiers qui ont un facteur premier relativement petit par rapport à la racine carrée de l'entier. Cependant, il peut être inefficace pour factoriser les entiers qui ont des facteurs premiers relativement grands ou pour les entiers avec une structure particulière.

9 Diffusion du rapport et du code

Pour permettre aux lecteurs de découvrir la richesse de ce rapport et aussi le code python de l'algorithme modifié de Pollard p-1, nous mettons à leur disposition le lien suivant pour le téléchargement : Projet Pollard P-1

10 Bibliographie

Bouvier, C. (2015). *Algorithmes pour la factorisation d'entiers et le calcul.* Lorraine: Université de Lorraine.

Maya , S. L., Mohammad , A. B., & Rachmawat, D. (2020). *On using Pollard's p-1 Algorithm to Factor RPrime RSA Modulus.* Indonesia.

WIKIPEDIA. (s.d.). Récupéré sur WIKIPEDIA: https://fr.wikipedia.org/wiki/Algorithme_p-1_de_Pollard