УДК 511:003.26.09

В. М.Місько, м. Київ

ПРИСКОРЕННЯ МЕТОДУ КВАДРАТИЧНОГО РЕШЕТА НА ОСНОВІ ВИКОРИСТАННЯ ДОДАТКОВОГО ПОШУКУ B-ГЛАДКИХ ЧИСЕЛ

Abstract. The quadratic sieve method is the fastest for integers under 100 decimal digits or so. To get the best speed and less amount of memory we need to successfully choose the size of factor base and sieving interval. To small size will lead to small amount of B-smooth numbers, to big size will lead to lack of memory. Good size of factor base and sieve interval will lead to minimum amount of B-smooth for solution. This paper describes method which will allow us to reduce size of factor base and sieving interval (memory size) without reducing the size of B-smooth numbers.

Вступ

В основі криптостійкості найбільш популярного сьогодні асиметричного криптоалгоритму RSA є складність факторизації великих цілих чисел. Відкритий ключ містить велике складене ціле число – криптомодуль N, що є добутком двох великих простих чисел. На даний час нема відомого простішого універсального шляху зламати шифрування як факторизація N. Тоді ми зможемо отримати два простих числа з добутку та розшифрувати повідомлення [7,8].

В 1977 році, коли був винайдений алгоритм RSA, факторизація цілих чисел з 80 десятковими знаками здавалась неможливою; 256-бітові ключі були надійними. Першим серйозним проривом було квадратичне решето (Quadratic Sieve) [1], метод винайдений Карлом Померансом в 1981 році, який може факторизувати числа розміром порядку 100 десяткових символів та більше. На сьогодні це найкращий відомий метод факторизації чисел, розміром менше 110 десяткових знаків. Поява ідей, які дозволять знизити обчислювальну складність методу Квадратичного решета, може розширити множину великих чисел, де цей метод буде найкращим. Це дасть змогу покращити процес криптоаналізу, хоча може призвести до збільшення числа розрядів N для криптостійких шифрів RSA. Тому розробка нових способів прискорення методу Квадратичного решета та їх дослідження є актуальним. В даній статті для зменшення розміру факторної бази та інтервалу просіювання пропонується використати поняття умовно B-гладких чисел.

Постановка задачі

Припустимо, що N - число яке ми повинні факторизувати, алгоритм квадратичного решета намагається знайти два числа x та y, таких щоб $x \neq \pm y (\mathrm{mod}\, n)$ та $x^2 = y^2 (\mathrm{mod}\, n)$. Це буде означати, що $(x-y)(x+y)=0 (mod\ n)$, і ми просто вирахуємо множники N як $НОД(x-y,n)$ та $НОД(x+y,n)$, використовуючи алгоритм Евкліда. Є принаймні ½ шансу, що цей додаток

буде не тривіальним дільником N.

Алгоритм квадратичного решета генерує послідовність квадратів використовуючи многочлен $x^2 - N$, змінюючи x від $\sqrt{N}$ до $\sqrt{N} + M$ [2]. Величина M збільшується до границі $|M| \le L^b, L^b$ - **інтервал просіювання**. Це місце, де метод стає евристичним, тому що абсолютно точного способу обчислення інтервалу просіювання немає.

У квадратичному решеті ми вираховуємо остачі $x^2 \bmod N$ для деяких x, та потім знаходимо таку множину, добуток елементів якої є квадратом. Це приводить до порівняння квадратів. Однак, піднесення до квадрату множини випадкових чисел за модулем N приводить до великої кількості різних простих множників, великим векторам та до великого розміру матриці спеціальної системи лінійних рівнянь. Тому, для спрощення, ми спеціально шукаємо пари цілих чисел x та $y(x)$, які відповідають значно простішим умовам ніж $x^2 = y^2 (\bmod N)$. Алгоритм вибирає набір простих чисел, **який називається факторною базою**, та намагається знайти x таке щоб залишок $y(x) = x^2 \bmod N$ був добутком простих чисел, що входять до факторної бази. Такі x називаються гладкими по відношенню до факторної бази, або B-гладкими.

У якості факторної бази B береться множина простих чисел, яка складається з p, які не перевищують задану границю L^a (яка вибирається із врахувань оптимальності). Границя L^a - це ще одне евристичне місце алгоритму.

Алгоритм працює в два етапи: етап збору даних, де він збирає інформацію, яка може привести до рівності квадратів; та етап обробки даних, де він розміщує всю зібрану інформацію у матрицю та оброблює її для отримання рішення. Другий етап потребує великі об'єми пам'яті та його важко розпаралелити.

Швидкість та результати роботи алгоритму залежить від таких факторів:

1. Розмір факторної бази.
2. Розмір інтервалу просіювання.

Якщо кількість простих чисел у факторній базі (розмір факторної бази) дуже малий, то розмір вектора степенів буде малим, це значно зменшує кількість операцій. Проблема в тому, щоб знайти такі B-гладкі числа, які б входили в цю факторну базу. Чим менше факторна база, тим суттєво меншою є кількість B-гладких чисел, тобто необхідно значно збільшувати інтервал просіювання. Якщо створити велику за розміром факторну базу, то перед нами б постала проблема вирішення СЛАУ спеціального виду з матрицею великої розмірності, що потребує великої кількості пам'яті та ресурсів. Оптимальне значення розміру факторної бази пропонується в роботі [3], яке обчислюється за формулою:

$$A = L^a = \left(e^{\sqrt{\ln(n)\ln\ln(n)}}\right)^{\sqrt{2}/4} = L(n)^{\sqrt{2}/4} = L^{\sqrt{2}/4} \qquad (1)$$

Ця формула не дає остаточної відповіді. Для кожного випадку найкращий розмір факторної бази є індивідуальним і може відрізнятися від значення отриманого за форму лою.

Наприклад, коли факторизували RSA-129 в 1994 році, використовували факторну базу простих чисел розміром 534339.

Інтервал просіювання повинен бути таким щоб *B*-гладких була більше, за кількість елементів у кожному векторі. Але цієї умови не достатньо. Ми можемо скласти матрицю де кількість векторів більше за кількість елементів у кожному векторі, та отримати хибне рішення. В такому випадку нам знадобиться розширити інтервал просіювання, для отримання додаткових векторів. Для загального випадку (*згідно з* [3]), отримати розмір інтервалу просіювання можна за формулою:

$$M_{\max} = L^b = \left(e^{\sqrt{\ln(n)\ln\ln(n)}}\right)^{3\sqrt{2}/4} = L(n)^{3\sqrt{2}/4} = L^{3\sqrt{2}/4} \qquad (2)$$

Якщо, після ділення числа *M* на всі прості числа з факторної бази *B*, залишок не дорівнює одиниці, ми відкидаємо таке число. Додатковий аналіз цих чисел може надати більшу кількість векторів, для побудови матриці. Схожа ідея описується у літературі [2, 5, 6] , але розглядалися тільки прості залишки.

Основною проблемою для методу квадратичного решета - є пошук достатньої кількості *B*-гладких чисел. Тому пошук способів отримання додаткових варіантів остач, що можуть розглядатися як *B*-гладкі числа, є актуальним завданням, що розглядається в даній статті.

Ідея ж полягає в тому, щоб розглянути залишки, які є квадратами простих чисел, які не ввійшли у факторну базу. Вектори таких чисел можна добавляти до матриці не враховуючи ці залишки, як квадрати вони ні як не впливають на рішення. Якщо $y(a) = 7*11^2*23*137^2$ та $y(b) = 7*23$, тоді $y(a)*y(b) = 7^2*11^2*23^2*137^2$. При обраному максимальному числі для факторної бази 23, вектор *y(a)* увійде до матриці. Ми можемо не враховувати 137^2 при розв'язанні матриці, тому що 137 має парну степінь. Такі залишки, в подальшому, будемо називати **умовно *B*-гладкими.** Покажемо, що застосування умовно *B*-гладких чисел дозволяє знизити розміри факторної бази, матриці та отримати рішення без розширення інтервалу просіювання.

Застосування аналізу умовно B-гладких чисел.

Розглянемо на прикладі ефективність запропонованої модифікації. Оберемо *p=401* та *q=103*, ці прості числа створюють нам число для факторизації *p*q=N=41303*. Обчислимо за формулою (1) розмір факторної бази *A=6*. За допомогою формули (2) отримаємо інтервал просіювання *M=203*.

Після просіювання варіантів *y(x)* через факторну базу, отримуємо *B*-гладкі числа. Ці числа зображені в таблиці 1.

Таблиця 1.

B-гладкі числа

Знак числа	2	11	19	23	29	37	*B*-гладкі
1	1	1	0	0	2	0	-18502
1	0	1	0	0	0	2	-15059
1	1	0	0	1	0	1	-1702
0	1	0	2	0	0	0	722
0	0	0	0	2	1	0	15341
0	1	1	0	1	0	1	18722

Цих чисел не достатньо для факторизації обраного *N*. Знайдемо умовно *B*-гладкі числа, вони зображені в таблиці 2.

Таблиця 2.

Умовно B-гладкі числа

-	2	11	19	23	29	37	Дільники які не входять до факторної бази	Умовно *B*-гладкі
0	0	0	0	0	0	0	149^2	22201
0	1	0	0	0	0	0	131^2	34322
0	1	0	0	0	0	0	157^2	49298

Число 22201 не увійшло до матриці тому що воно має прості дільники які не потрапили до факторної бази. Число 22201 є квадратом, завдяки йому ми отримуємо рішення. Числа 32322 та 49298 не є квадратами, але разом дають нам ще одне рішення.

Розглянемо інший приклад. Оберемо *p=11* та *q=601*, отримаємо *p*q=N=6611*. Обчислимо розмір факторної бази та інтервал просіювання *A=5, M=102*.

Після просіювання варіантів *y(x)* через факторну базу, отримуємо *B*-гладкі числа. Ці числа зображені в таблиці 3.

Таблиця 3.

B-гладкі числа

Знак числа	2	5	17	29	31	*B*-гладкі
1	1	1	1	1	0	-4930
1	0	1	0	1	1	-4495
1	1	1	2	0	0	-2890
1	1	0	1	1	0	-986
1	0	0	1	0	1	-527
1	1	2	0	0	0	-50
0	0	1	0	2	0	4205
0	1	1	1	0	1	5270

Обчислюючи матрицю створену з векторів з таблиці 3 ми отримаємо тільки хибні рішення. Знайдемо умовно *B*-гладкі числа, вони зображені в таблиці 4.

Таблиця 4.

Умовно *B*-гладкі числа

-	2	5	17	29	31	Дільники які не входять до факторної бази	Умовно *B*-гладкі
1	1	0	0	0	0	41^2	-3362
0	0	1	0	0	0	37^2	6845

Число -3362 дозволило сформувати рішення з чисел: -4930, -4495, -3362 та -527.

Приклади випадків де умовно *B*-гладкі входять до рішення наведені в таблиці 5.

Таблиця 5.

Приклади факторизації з умовно *B*-гладкими числами

p	q	N	*B*-гладкі, які утворюють квадрат	Умовно *B*-гладкі	Множники Умовно *B*-гладких
27743	41203	1143094829	45292900	45292900	$5^2*7^2*673^2$
89	46411	4130579	-1496450, -5618	-1496450	$-1*2*5^2*173^2$, $-1*2*53^2$
5647	40577	229138319	-29848630, -2996875, 11514850	-29848630,	$-1*2*5*7653^2$, $-1*5^2*7*137$, $2*5^2*41^2*137$
29741	40087	1192227467	26759929	26759929	7^2*739^2
30271	48533	1469142443	83375161	83375161	23^2*739^2
30707	32089	985356923	477481	477481	691^2
31729	32423	1028749367	120409	120409	347^2
32443	45137	1464379691	40284409	40284409	11^2*577^2
32887	39371	1294794077	10510564	10510564	2^2*1621^2
6163	44777	275960651	-22386875, -2107, 23952473	23952473	$-1*5^4*7^2*17*43$, $-1*7^2*43$, $17*1187^2$
36353	39511	1436343383	2493241	2493241	1579^2
37561	43067	1617639587	7579009	7579009	2753^2
38239	45413	1736547707	12866569	12866569	17^2*211^2
39157	45119	1766724683	8886361	8886361	11^2*271^2
40577	46811	1899449947	9715689	9715689	3^2*1039^2
41719	45137	1883070503	2920681	2920681	1709^2

6359	43051	273761309	-38568413 -23710340 -6177145 -685684	-38568413 -23710340	$-1*13*41*269^2$, $-1*2^2*5*37*179^2$, $-1*5*13*29^2*113$, $-1*2^2*37*41*113$
44867	47911	2149622837	2316484	2316484	2^2*761^2
45403	46589	2115280367	351649	351649	593^2
48193	48539	2339240027	29929	29929	173^2

Порівнювальна оцінка аналізу умовно B-гладких чисел

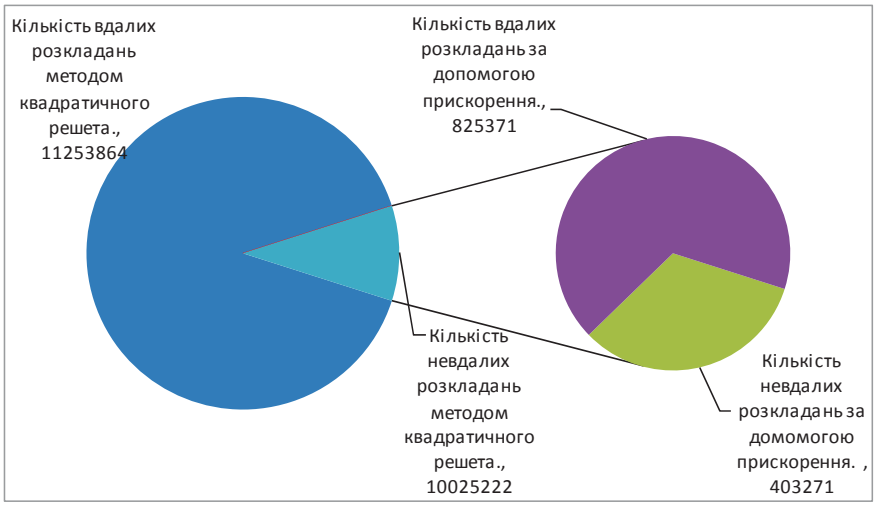
Додаткові вектори у сформованій матриці дозволили отримати рішення без розширення факторної бази або інтервалу просіювання.

Отримати числа у яких залишок є квадратом можна доволі часто. Взявши перших 5000 простих чисел та сформувавши за них $12.5*10^6$ можливих варіантів *N*, ми знайшли додаткові вектори у 99% випадках.

Корисну дію цього методу можна побачити, якщо обрати випадки в яких базовий алгоритм квадратичного решета при рекомендованих[2, 4, 6] розмірах факторної бази та інтервалу просіювання обчислених за формулами (1) та (2) не зміг знайти рішення, та застосувати аналіз *y(x)*, у яких залишок після просіювання є просте число у парній степені.

У 7% випадках модифікований алгоритм зміг факторизувати число.

Діаграма 1.



Варто зазначити, що якщо для порівнювального аналізу взяти меншу кількість простих чисел, починаючи не з першого простого числа, ми отримаємо кращі результати. Наприклад якщо взяти тисячу простих чисел починаючи з простого числа з порядковим номером 4000 або 5000 ми знайдемо, що модифікований алгоритм зміг факторизувати всі числа.

Оцінка складності та часу виконання.

Складання матриці для стандартного алгоритму квадратичного решета потребує L^{2a} місця [3]. При застосуванні додаткового аналізу варіантів *y(x)* нам необхідно для кожного варіанта *y(x)* запам'ятовувати залишок (якщо він є квадратом), тому цей об'єм пам'яті збільшується і стає рівним L^{2a+1}.

Кількість варіантів *y(x)* які нам підійдуть для стандартного квадратичного решета можливо розрахувати за формулою $L^{b-(4a)^{-1}}$. При застосуванні додаткового аналізу варіантів *y(x)* ця кількість збільшується на деяке γ і становить $L^{b-(4a)^{-1}+y}$. Значення b обирається таким щоб кількість варіантів *y(x)* які нам підходять становила L^a, тому $b = a + (4a)^{-1} - \gamma$. Як ми бачимо інтервал просіювання зменшився.

Оцінити швидкість модифікованого алгоритму можна за формулою:

$$L^{\max\left\{2a+1,\, a+(4q)^{-1}-y,\, 3a\right\}} \qquad (3)$$

Швидкість просіювання зменшилась на γ, де γ кількість елементів *y(x)* доданих умовно B-гладких залишків.

Висновки

Ефективність методу квадратичного решета залежить від вдалого вибору розміру факторної бази та інтервалу просіювання. На основі проведених чисельних експериментів показано, що використання умовно B-гладких чисел дозволяє збільшити кількість векторів для побудови матриці, що означає можливість використання факторної бази та інтервалу просіювання меншого розміру.

1. The quadratic sieve factoring algorithm, C. Pomerance, Advances in Cryptology, Proceedings of Eurocrypt 84, Paris, 1984, T. Beth. N. Cot, and I. Ingemarsson, eds., Lecture Notes in Computer Sci. 209 (1985), 169–182.
2. The Quadratic Sieve Factoring Algorithm Eric Landquist MATH 488: Cryptographic Algorithms December 14, 2001
3. Analysis and comparison of some integer factoring algorithms, C. Pomerance, Computational Methods in Number Theory, Part I, H.W. Lenstra, Jr. and R. Tijdeman, eds., Math. Centre Tract 154, Amsterdam, 1982, 89–139.
4. Carl Pomerance. Smooth numbers and the quadratic sieve. Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography, MSRI Publications, 44:69–81, 2008
5. Song Y. Yan. Cryptanalytic attacks on RSA / Song Y. Yan – Springer Science and Business Media, Inc. 2008. – P.255
6. Prime Numbers: a computational perspective, second edition, R. E. Crandall and C. Pomerance, Springer, New York, 2005.
7. *Горбенко И.Д.* Анализ каналов уязвимости системы RSA / И.Д. Горбенко, В.И. Долгов, А.В. Потий, В.Н. Федорченко // Безопасность информации. – 1995. – № 2. – С.22-26.
8. *Daniel R. L. Brown.* Breaking RSA May Be As Difficult As Factoring. – [Электронный ресурс]. Режим доступа: http://www.pgpru.com/novosti /2005/1026vzlomrsabezfaktorizaciirealennoneeffektiven – Название с экрана.

Поступила 13.04.2017р.