УДК 511:003.26.09
DOI: 10.15587/2313-8416.2017.118298

ПРИСКОРЕННЯ МЕТОДУ КВАДРАТИЧНОГО РЕШЕТА НА ОСНОВІ ПОШУКУ ДОДАТКОВИХ B-ГЛАДКИХ ЧИСЕЛ

© В. М. Місько

Метод квадратичного решета є найкращим відомим методом факторизації чисел, розміром менше 100 десяткових знаків. Швидкість методу та розмір необхідної пам'яті у багатьох випадках залежить від вдало обраного розміру факторної бази та інтервалу просіювання. У даному дослідженні зображено метод, який дозволяє зменшити розмір факторної бази та інтервалу просіювання без зменшення кількості B-гладких чисел

Ключові слова: *метод квадратичного решета, додаткові B-гладкі числа, факторна база, інтервал просіювання*

1. Вступ

В інформаційно-телекомунікаційних системах для рішення задачі захисту інформації часто використовують RSA алгоритм, який став стандартом дефакто для багатьох криптографічних додатків. Широке розповсюдження цього алгоритму робить актуальним його криптоаналіз.

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 можна отримати два простих числа з добутку та розшифрувати повідомлення [1, 2].

Тому вдосконалення алгоритмів факторизації є актуальною задачею.

2. Літературний огляд

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) [3], 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 100 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.

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.

Додатковий аналіз B-гладких чисел згадується в літературі [4, 5]. Пропонується запам'ятовувати B-гладкі з неодиничним простим залишком. При знаходженні B-гладких з однаковими залишками використовувати їх разом.

В даному дослідженні пропонується додатково використовувати B-гладкі з не одиничними залишками, які являються квадратами простих чисел. Такі B-гладкі будемо називати – додаткові B-гладкі.

3. Мета та задачі дослідження

Мета дослідження – аналіз степені прискорення методу квадратичного решета на основі пошуку додаткових B-гладких чисел.

Для досягнення мети були поставлені наступні задачі:

1. Аналіз впливу додаткових умовно B-гладких на швидкість та результат факторизації.

2. Аналіз кількості випадків появи додаткових B-гладких чисел.

3. Оцінка складності та часу виконання.

4. Постановка задачі

Припустимо, що N – число яке автор повинен факторизувати, алгоритм квадратичного решета намагається знайти два числа x та y, таких щоб $x \neq \pm y \left(mod\, n\right)$ та $x^2 = y^2 \left(mod\, n\right)$. 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 [6, 7].

Алгоритм квадратичного решета генерує послідовність квадратів використовуючи многочлен $x^2 - N$, змінюючи x від $\sqrt{N}$ до $\sqrt{N} + M$ [6]. Величина M збільшується до границі $\left|M\right| \le L^b$, L^b – інтервал просіювання. Це місце, де метод стає евристичним, тому що абсолютно точного способу обчислення інтервалу просіювання немає.

У квадратичному решеті вираховуємо остачі $x^2 \, mod\, N$ 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, спеціально шукаємо пари цілих чисел x та $y(x)$, які відповідають значно простішим умовам ніж $x^2 \equiv y^2 \left(mod\, n\right)$. Алгоритм вибирає набір

простих чисел, який називається факторною базою, та намагається знайти *x* таке щоб залишок $y(x) = x^2 mod\, n$ 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 обробляє 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 автором б постала проблема вирішення системи лінійних алгебраїчних рівнянь (СЛАУ) спеціального виду з матрицею великої розмірності, що потребує великої кількості пам'яті та ресурсів. Оптимальне значення розміру факторної бази пропонується в дослідженні [8], яке обчислюється за формулою:

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

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

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. Можна скласти матрицю де кількість векторів більше за кількість елементів у кожному векторі, та отримати хибне рішення. В такому випадку необхідно розширити інтервал просіювання, для отримання додаткових векторів. Для загального випадку (*згідно з* [8]), отримати розмір інтервалу просіювання можна за формулою:

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

Якщо, після ділення числа *M* на всі прості числа з факторної бази *B*, залишок не дорівнює одиниці, відкидаємо таке число. Додатковий аналіз цих чисел може надати більшу кількість векторів, для побудови матриці.

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

$$y(a) = 7*11^2*23*137^2 \text{ та } y(b) = 7*23,$$

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. Можна не враховувати 137^2 при розв'язанні матриці, тому що 137 має парну степінь. Такі залишки, в подальшому, будемо називати додатковими *B*-гладкими. Покажемо, що застосування додаткових *B*-гладких чисел дозволяє знизити розміри факторної бази, матриці та отримати рішення без розширення інтервалу просіювання.

5. Застосування аналізу додаткових B-гладких чисел.

Розглянемо на прикладі ефективність запропонованої модифікації. Оберемо *p=401* та *q=103*, ці прості числа створюють 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 табл. 1.

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

Число 22201 не увійшло до матриці тому що воно має прості дільники які не потрапили до факторної бази. Число 22201 є квадратом, завдяки йому отримуємо рішення. Числа 32322 та 49298 не є квадратами, але разом дають 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 табл. 3.

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

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

Приклади випадків де додаткові *B*-гладкі входять до рішення наведені в табл. 5.

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

Таблиця 2

Додаткові B-гладкі числа

Знак числа	2	11	19	23	29	37	Дільники які не входять до факторної бази	Додаткові *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

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

Таблиця 4

Додаткові *B*-гладкі числа

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

Таблиця 5

Приклади факторизації з додатковими *B*-гладкими числами

p	q	N	*B*-гладкі, які утворюють квадрат	Додаткові *B*-гладкі	Множники додаткових *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*397^2
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	$–1*5^4*7^2*17*43$, $–1*7^243$, $17*1187^2$
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

6. Порівнювальна оцінка аналізу додаткових 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, було знайдено додаткові вектори у 99 % випадках.

Корисну дію цього методу можна побачити, якщо обрати випадки в яких базовий алгоритм квадратичного решета при рекомендованих [9, 10] 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. Результати такого експерименту зображені на рис. 1.

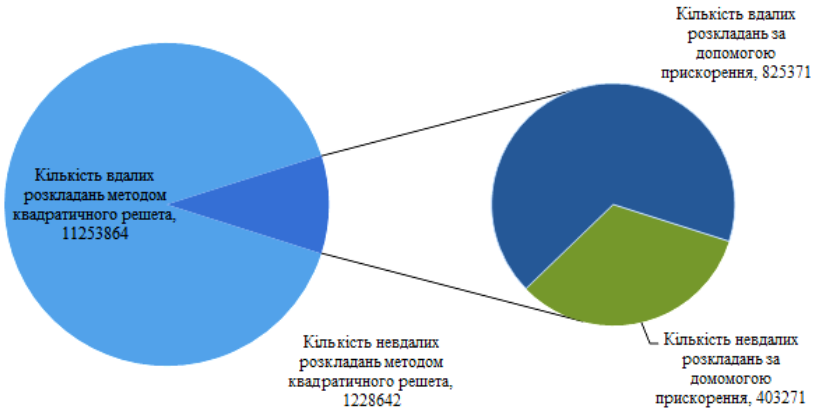

Рис. 1. Порівнювальна характеристика базового та модифікованого методів квадратичного решета

У 7 % випадках модифікований алгоритм зміг факторизувати число.

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

7. Оцінка складності та часу виконання

Складання матриці для стандартного алгоритму квадратичного решета потребує L^{2a} місця [7]. Кількість варіантів y(x) які потрібні (B-гладкі) для стандартного квадратичного решета можливо розраховувати за формулою $L^{b-(4a)^{-1}}$.

Всі залишки y(x) вже отримані, їх пошук не потребує додаткової роботи. При застосуванні аналізу додаткових B-гладких чисел необхідно для кожного варіанта y(x) запам'ятовуватиалишок (якщо він є квадратом), тому необхідний об'єм пам'яті збільшується і стає рівним L^{2a+1}.

На перший погляд додатковий аналіз варіантів y(x) робить алгоритм складнішим, та збільшує час його роботи – додаткове обчислення квадратного кореню з усіх залишків y(x) більших за одиницю. Однак, при застосуванні додаткового аналізу варіантів y(x), кількість y(x) які підходять збільшується (за рахунок додаткових B-гладких) на деяке γ, і становить $L^{b-(4a)^{-1}+\gamma}$. Значення b – кількість ітерацій в алгоритмі, обирається таким, щоб кількість варіантів y(x) які підходять становила L^a, тому $b=a+(4a)^{-1}-\gamma$. Як бачимо ця кількість зменшилась.

Беручи до уваги те що b – показник степені інтервалу просіювання L^b, необхідно відмітити що кожне знайдене додаткове B-гладке значення значно зменшує інтервал просіювання.

Оцінити швидкість модифікованого алгоритму можна за формулою:

$$L^{max\left\{2a+1,a+(4a)^{-1}-\gamma,3a\right\}}.$$

Швидкість просіювання зменшилась на γ, де γ кількість елементів y(x) доданих завдяки аналізу додаткових B-гладких залишків.

8. Висновки

Швидкість методу квадратичного решета залежить від таких евристичних значень як розмір факторної бази та інтервал просіювання. На основі проведених чисельних експериментів показано:

1. Пошук додаткових B-гладких чисел дозволяє факторизувати число у тих випадках, коли базовий алгоритм квадратичного решета (при стандартному інтервалі просіюванні та розміру факторної бази) не зміг сформувати матрицю для отримання рішення.

2. Модифікований алгоритм зміг зменшити кількість невдалих факторизацій з 11 % до 3 %, відносно базового алгоритму квадратичного решета.

3. Швидкість просіювання модифікованого алгоритму зменшилась на γ. Для кожного випадку значення γ є різним і дорівнює кількості елементів y(x) доданих завдяки використанню додаткових B-гладких чисел.

Слід додати що, були знайдені випадки, коли додаткові B-гладкі були квадратами простих чисел, що значно зменшило час факторизації у конкретних випадках.

Література

1. Горбенко, И. Д. Анализ каналов уязвимости системы RSA [Текст] / И. Д. Горбенко, В. И. Долгов, А. В. Потий, В. Н. Федорченко // Безопасность информации. – 1995. – № 2. – С. 22–26.

2. Brown, D. R. L. Breaking RSA May Be As Difficult As Factoring [Electronic resource] / D. R. L. Brown // Cryptology ePrint Archive. – 2005. – Available at: https://eprint.iacr.org/2005/380

3. Pomerance, C. The quadratic sieve factoring algorithm [Text] / C. Pomerance // Lecture Notes in Computer Science. – 1985. – P. 169–182. doi: 10.1007/3-540-39757-4_17

4. Lindquist, E. The Quadratic Sieve Factoring Algorithm [Text] / E. Lindquist // Math 488: Cryptographic Algorithms, Diciembre. – 2001.
5. Song, Y. Quadratic Sieve [Text] / Y. Song // Primality Testing and Integer Factorization in Public-Key Cryptography. – New York: Springer, 2009. – P. 234–239.
6. Diffie, W. New directions in cryptography [Text] / W. Diffie, M. Hellman // IEEE Transactions on Information Theory. – 1976. – Vol. 22, Issue 6. – P. 644–654. doi: 10.1109/tit.1976.1055638
7. Шенхаге, А. Быстрое умножение больших чисел [Текст] / А. Шенхаге, В. Штрассен // Кибернетический сборник. – 1973. – № 2. – С. 87–98.
8. Pomerance, C. Analysis and comparison of some integer factoring algorithms [Text] / C. Pomerance // Mathematisch Centrum Computational Methods in Number Theory. P. 1. – 1982. – P. 89–139.
9. Pomerance, C. Smooth numbers and the quadratic sieve [Text] / C. Pomerance // Proc. of an MSRI workshop. – 2008. – P. 69–81.
10. Crandall, R. Smooth numbers and the quadratic sieve [Text] / R. Crandall, C. Pomerance // Prime Numbers A Computational Perspective. – New York: Springer, 2005. – P. 261–315.

Рекомендовано до публікації д-р техн. наук Винничук С. Д.
Дата надходження рукопису 20.10.2017

Місько Віталій Миколайович, аспірант, Відділ автоматизації проектування енергетичних установок, Інститут проблем моделювання в енергетиці ім. Г. Є. Пухова НАН України, вул. Генерала Наумова, 15, м. Київ, Україна, 03164
E-mail: vitalii.misko@gmail.com

УДК 621.316:629.4.048.7
DOI: 10.15587/2313-8416.2017.118277

РАЗРАБОТКА ЭНЕРГОСБЕРЕГАЮЩИХ СИСТЕМ ЭЛЕКТРИЧЕСКОГО ОТОПЛЕНИЯ С НОЧНЫМ АККУМУЛИРОВАНИЕМ ТЕПЛОТЫ

© А. М. Андрющенко, В. Р. Никульшин, А. Е. Денисова

Предложен метод и разработана программа для расчета систем электроотопления с ночной аккумуляцией теплоты, а также для определения ежемесячного теплопотребления здания с расчетом стоимости отопления при использовании других энергетических ресурсов (централизованное теплоснабжение, газовое отопление, пеллетное отопление). Показано, что стоимость электрического отопления с ночной аккумуляцией теплоты в 2 раза ниже стоимости централизованного отопления
***Ключевые слова**: энергосбережение, электроотопление, ночная аккумуляция теплоты, затраты на разные виды энергетических ресурсов*

1. Введение

Резкое и неравномерное удорожание стоимости различных видов энергоресурсов, используемых для отопления, побуждает к поиску альтернативных экономичных решений. Снизить затраты на отопление можно за счет снижения потерь теплоты ограждающими конструкциями зданий в результате их термомодернизации [1], а также за счет перехода на использование альтернативных энергетических ресурсов. Среди альтернативных вариантов, в первую очередь для крупных городов, в последние годы рассматриваются электрическое отопление с ночным аккумулированием теплоты и электрический подогрев теплоносителя в ночное время в системах централизованного теплоснабжения. В обоих случаях будет достигаться эффект выравнивания суточного графика электрической нагрузки [2], объединённой энергосистемы Украины [3], что благоприятно скажется на её работе за счёт более полной загрузки украинских АЭС, вырабатывающих электроэнергию по самой низкой стоимости по сравнению с другими видами генерации. Ещё одним положительным аспектом применения электроотопления при условии его достаточного распространения, является экономия природного газа.

Был рассмотрен наиболее простой вариант электроотопления – использование ТЭНов, а не тепловых насосов. Тепловые насосы, как известно, позволили бы примерно в три раза уменьшить затраты электроэнергии на отопление, но при этом потребовали бы многотысячных долларовых вложений со сроком окупаемости от 3 лет и более, что в нынешней экономической ситуации для Украины мало приемлемо.

2. Анализ литературных данных и постановка проблемы

Проблема поиска эффективных решений для различных систем отопления является актуальной на протяжении последних десятилетий и будет оставаться таковой в ближайшем обозримом будущем. Число публикаций этой тематике исчисляется десятками тысяч. Так, поисковый запрос «Google» «электрокотлы с аккумуляторами теплоты» дает около 50000 тысяч ссылок. Поэтому кратко остановимся только на некоторых из них.