УДК 681.322

Р.Б. Попович
Національний університет “Львівська політехніка”
кафедра “Електронні обчислювальні машини”

КРИПТОАНАЛІЗ СИСТЕМИ RSA НА ОСНОВІ ПОШУКУ ЗНАЧЕННЯ ФУНКЦІЇ ЕЙЛЕРА

Запропоновано підхід до криптоаналізу системи RSA, при якому немає потреби працювати з матрицями великого розміру й використовувати для цього суперкомп’ютер.

An approach of cryptoanalisys of RSA system is offered, for which it is not necessary to work with large dimension matrices and to use for this a supercomputer.

1. Окреслення проблеми

Криптографія – це захист інформації шляхом її перетворення, що виключає прочитання цієї інформації сторонньою особою. Ще кілька десятиліть тому такий підхід стосувався в основному військових операцій або був пов’язаний зі шпигунськими історіями, а не був предметом широкого використання. Причиною бурхливого розвитку криптографії є широке використання комп'ютерних мереж, зокрема глобальної мережі Internet, якими передаються великі обсяги інформації державного, військового, комерційного й приватного характеру, що не допускає можливості доступу до неї сторонніх осіб. Широковживаним є алгоритм RSA шифрування з відкритим ключем. Багато провідних світових IT-компаній вклали в його розвиток значні кошти, на його основі функціонують Internet-платежі eMoney. Алгоритм RSA використовується в системі електронних платежів НБУ – загальнодержавній платіжній системі, що забезпечує здійснення розрахунків між банківськими установами, органами державного казначейства на території України із застосуванням електронних засобів приймання, опрацювання, передавання та захисту інформації.

Тому аналіз підходів до зламування таких систем, а отже, і їх стійкості є актуальним питанням.

2. Аналіз останніх досліджень та публікацій

Система шифрування з відкритим ключем RSA [1,2] запропонована у 1977 р. Використовується як для шифрування повідомлень, так і для утворення цифрових підписів. Стійкість її ґрунтується на складності розв’язування задачі розкладу числа на прості множники.

Вибір ключа. Випадково вибирають два великі прості числа p і q та обчислюють їх добуток $n=p\cdot q$. Випадково вибирають число e таке, що $e<\varphi(n)=(p-1)(q-1)$, де $\varphi(n)$ – значення функції Ейлера для n. Число e повинно бути того самого порядку, що й число n і, крім того, числа e та $\varphi(n)$ повинні бути взаємно простими. За допомогою алгоритму Евкліда обчислюють число d, що задовольняє умову $e\cdot d \equiv 1 \pmod{\varphi(n)}$. Пара e, n є утвореним публічним ключем, а число d – приватним ключем.

Шифрування. Шифрувати можна числа з проміжку $0\le m\le n$. Шифрування полягає в обчисленні значень: $E(m) \equiv m^e \bmod n$.

Дешифрування полягає в обчисленні значень $D(c) \equiv c^d \bmod n$.

У 1991 р. фірма RSA Security почала проводити конкурси із розкладу чисел.

Найвагомішим нині результатом є факторизація числа RSA-155 з 155 десятковими розрядами (512 біт) [4]. Факторизацію завершено 1999 р. після семи місяців роботи. Обчислення виконували на 300 робочих станціях і персональних комп'ютерах.

Сьогодні на конкурсі виставлені числа, які мають від 576 до 2048 біт; суми винагород за їх факторизацію – від 10000 до 200000 дол. [6].

Перший алгоритм факторизації (і одночасно тест простоти) – сито Ератосфена – запропонований понад дві тисячі років тому Ератосфеном з Олександрії. Це процедура, яка створює список усіх простих чисел у межах від одиниці до заданого числа n. Коли початковий відрізок простих чисел знайдено, наступне просте число визначається шляхом перевірення його подільності на кожне із попередніх. Описаний алгоритм та його модифікації є експоненційними за часом виконання й практично незастосовними.

Ефективні сучасні тести простоти чисел [1,2] хоча й дають змогу швидко розпізнати, складеним чи простим є задане число, проте не дають жодної інформації про дільники складеного числа.

Ферма запропонував [2] записати число, що розкладається на множники у вигляді різниці квадратів двох натуральних чисел $n=x^2-y^2$, а потім, обчислюючи найбільший спільний дільник чисел n та $x–y$, пробувати знайти нетривіальний дільник числа n.

Лежандр звернув увагу [2] на те, що за такого підходу можна замість рівності $n=x^2–y^2$ розглядати порівняння $x^2 \equiv y^2 \bmod n$. Зрозуміло, що коли числа x,y задовольняють рівність $n=x^2-y^2$, то вони також задовольняють і наведене порівняння. Проте не кожна пара чисел, що задовольняє наведене порівняння, задовольняє цю рівність. А тому не кожна пара чисел x,y, що задовольняє порівняння, дає змогу розкласти n на прості множники.

Залежно від того, у який спосіб отримано числа, пов'язані наведеним порівнянням, розрізняють кілька різних методів факторизації.

Метод квадратичного решета. Цей метод запропонований Померанце 1982 р. [2,3]. Час його роботи оцінюють як $O(\exp(\sqrt{9/8 \cdot k \cdot \log_2 k})$, де k – кількість бітів у числі n.

Нехай $A(x)=(x+[\sqrt{n}])^2-n$. Для реалізації алгоритму вибирають набір обмежених за значенням простих чисел $\{t_1,t_2,\ldots,t_s\}$ (так звана база множників).

Задавши деяку границю U, для кожного простого числа t_j з бази множників і кожного показника степеня k_j ($k_j \le U$) знаходимо ті числа x, за яких $A(x)$ ділиться на $t_j^{k_j}$. Числа x, за яких значення $A(x)$ виявляються повністю розкладеними на степені простих чисел з бази множників, дають вираз $A(x) = \prod_{j=1}^{s} t_j^{k_j}$.

Згідно з цим розкладом числу x зіставляють вектор показників $(k_1,k_2,\ldots,k_s)$. Обчислення виконують доти, доки не буде збудовано $s+2$ таких векторів.

В отриманій матриці показників можна так підібрати вектори-рядки, що їхня сума дасть вектор з парними значеннями компонент: $2(l_1,l_2,\ldots,l_s)$. Тоді отримуємо порівняння

$$\prod_x (x+[\sqrt{n}])^2 \equiv (\prod_{j=1}^{s} t_j^{l_j})^2 (\bmod n).$$

Метод решета числового поля. Найліпшим відомим сьогодні методом факторизації великих чисел є метод решета числового поля, розроблений групою авторів (Буглер, А.Ленстра, Г.Ленстра, Манассе, Поллард, Померанце) на початку 90-х років XX ст. [4,5,7].

Для складності алгоритму є евристична оцінка – $O(\exp(\sqrt{1,9\cdot(\ln k)^{1/3}\cdot(\ln\ln k)^{2/3}})$, де k – кількість бітів у числі n.

У цьому методі використовують деякі найпростіші результати з теорії алгебраїчних чисел. Розглядають розширення $Z[\alpha]$ кільця цілих чисел Z, де α – корінь деякого багаточлена $f(x)=x^d-c$, $c\in Z$. У цьому разі існує таке ціле m, що $m^d \equiv c \bmod n$. З кожним алгебраїчним числом $a+\alpha b$, $a,b\in Z$ пов'язують його норму $N(a+\alpha b)=a^d-c(-b)^d$. Є відповідність між розкладами на прості множники алгебраїчного числа та його норми.

Нехай ϕ – гомоморфізм, що переводить елемент α кільця $Z[\alpha]$ в елемент $m \bmod n$ кільця Z_n. Ідея методу решета числового поля полягає у використанні порівняння $\phi(a+\alpha b)\equiv(a+mb) \bmod n$ та відшуканні таких пар взаємно простих цілих чисел a та b, для яких алгебраїчне ціле $a+\alpha b$ та ціле $a+mb$ задовольняють умови гладкості: $\left|N(a+\alpha b)\right| = \prod_{p-просте,\, p\le B} p^{v_p}$; $\left|a+mb)\right| = \prod_{p-просте,\, p\le B} p^{w_p}$, де v_p,w_p – невід'ємні цілі числа.

Використовуючи розклад норми $N(a+\alpha b)$ на прості числа, можна отримати розклад алгебраїчного цілого $a+\alpha b$ на оборотні та прості елементи в $Z[\alpha]$, а саме: $a+\alpha b = (\prod_{u\in U} u^{t_u})\ (\prod_{g\in G} g^{v_g})$, де t_u – цілі, а v_g – невід'ємні цілі числа. Тоді U – певна наперед визначена множина твірних групи оборотних елементів, а G – множина твірних простих ідеалів у $Z[\alpha]$ з нормами, що є простими числами, які не перевищують B. Звідси випливає $(\prod_{u\in U} \phi(u)^{t_u})\ (\prod_{g\in G} \phi(g)^{v_g}) \equiv \prod_{p-просте,\, p\le B} p^{w_p} \bmod n$. Використовуючи низку таких співвідношень, можна отримати такі цілі числа x, y, що $x^2\equiv y^2 \bmod n$.

Практична реалізація обох методів складається з двох фаз. На початковій фазі відбувається формування елементів, що задовольняють відповідні порівняння. Після неї виконується матрична фаза, результатом якої є матриця великого розміру з елементами, отриманими на першій фазі. З цієї матриці отримують шуканий розклад числа на прості множники.

Фаза відбирання чисел та порівнянь виконується на великій кількості процесорів, що працюють одночасно. Матрична фаза потребує великої пам'яті і зазвичай її виконують на суперкомп'ютері.

3. Завдання роботи

Суттєвим недоліком відомих сьогодні методів криптоаналізу системи RSA є необхідність працювати на останньому етапі з матрицями великого розміру. Так, при факторизації числа RSA-155 [4] виникла розріджена матриця з 6699191 рядками та 6711336 стовпцями (у середньому 62 ненульових елементи в рядку). Для опрацювання цієї матриці використовували суперкомп'ютер Cray C916 з комп'ютерного центру SARA в Амстердамі (Голландія). Виконання відповідного алгоритму потребувало 224 CPU год. та 2 Гбайт оперативної пам'яті. Календарний час роботи – 9,5 днів. Як бачимо, потрібна потужна обчислювальна техніка, яка не завжди є доступною.

Іншим можливимим підходом до криптоаналізу системи RSA є знаходження значення функції Ейлера. При такому підході немає потреби використовувати суперкомп'ютер. Метою роботи є дослідження саме такого підходу.

4. Криптоаналіз системи RSA

Алгоритм відшукання розкладу числа n на прості множники еквівалентний алгоритму відшукання значення функції Ейлера. Справді, оскільки $n=p\cdot q,$, то $\varphi(n)=\varphi(p)\varphi(q)=(p-1)(q-1)=pq-(p+q)+1$ і $p+q=n-\varphi(n)+1$. Звідси, множники p,q є коренями квадратного рівняння $X^2-\frac{n-\varphi(n)+1}{2}X+n=0$.

Отже, знаючи $\varphi(n)$, можна, розв'язуючи це рівняння, знайти числа p і q, а знаючи p і q, легко обчислити $\varphi(n)$.

Зрозуміло, що $\varphi(n)=n-(p+q)+1$. Виходячи із розкладів чисел RSA-140, RSA-155 [6], можна взяти таку евристичну оцінку $1,9\sqrt{n} < p+q < 2,1\sqrt{n}$. Отже, пошук величини $\varphi(n)$ треба виконувати на відрізку $[n-2,1\sqrt{n}, n-1,9\sqrt{n}]$.

Нижче описано схему пошуку величини $\varphi(n)$. Хоча це зроблено стосовно одного конкретного числа, ця схема може бути використана у загальному випадку.

Через n надалі позначатимемо число RSA-576, яке має 576 біт (174 десяткових розряди) і виставлене на конкурс із факторизації чисел фірми RSA Security [6].

Залишок від ділення n на просте число 3 дорівнює 2, тобто $n \equiv 2 \bmod 3$. Множення ненульових за модулем 3 чисел описується так.

·	1	2
1	1	2
2	2	1

Оскільки $p\cdot q \equiv n \bmod 3$, то можливі два варіанти:

1) $p \equiv 1 \bmod 3$, $q \equiv 2 \bmod 3$. Тоді p-1 ділиться на 3;

2) $p \equiv 2 \bmod 3$, $q \equiv 1 \bmod 3$. Тоді q-1 ділиться на 3.

Отже, p-1 або q-1 ділиться на 3, і завжди $\varphi(n)=(p-1)(q-1)$ ділиться на 3.

Тепер розглядаємо рівність $p\cdot q \equiv n \bmod 8$. Залишок від ділення n на число 8 дорівнює 3. Таблиця множення ненульових за модулем 8 чисел має такий вигляд.

·	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	0	2	4	6
3	3	6	1	4	7	2	5
4	4	0	4	0	4	0	4
5	5	2	7	4	1	6	3
6	6	4	2	0	6	4	2
7	7	6	5	4	3	2	1

З неї видно, що добуток двох чисел дорівнює 3 за модулем 8, якщо

1) одне з чисел, скажімо p, дорівнює $p \equiv 1 \bmod 8$, а інше – $q \equiv 4 \bmod 8$. Тоді p-1 ділиться на 8;

2) одне з чисел, скажімо p, дорівнює $p \equiv 5 \bmod 8$, а інше – $q \equiv 7 \bmod 8$. Тоді $(p-1)(q-1) \equiv 4\cdot 6 \equiv 0 \bmod 8$.

Як бачимо, в обох випадках $\varphi(n)$ ділиться на 8.

Далі розглядаємо рівність $p \cdot q \equiv n \bmod 5$. Залишок від ділення n на число 5 дорівнює 4. Таблиця множення ненульових за модулем 5 чисел має такий вигляд.

·	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Згідно з цією таблицею добуток двох чисел дорівнює 4 за модулем 5, якщо

1) одне з чисел, скажімо p, дорівнює $p \equiv 1 \bmod 5$, а інше – $q \equiv 4 \bmod 5$. Тоді $p-1$ ділиться на 5;

2) обидва числа дорівнюють 2 mod 5. Тоді $\varphi(n)=1 \cdot 1 \equiv 1 \bmod 5$;

3) обидва числа дорівнюють 3 mod 5. Тоді $\varphi(n)=2 \cdot 2 \equiv 4 \bmod 5$.

Отже, $\varphi(n)$ при діленні на 5 дає залишок 0, 1 або 4.

Подібний аналіз рівності $p \cdot q \equiv n \bmod a$ за різними модулями a здійснено за допомогою програми на мові PASCAL. Деякі з отриманих результатів наведені нижче.

Модуль a	Залишок від ділення n на a	Можливі залишки від ділення $\varphi(n)$ на a
5	4	0,1,4
7	6	0,2,5
11	5	0,1,3,5,7,9
13	2	0,1,5,6,8,11
17	8	0,1,2,3,5,9,13,15,16
19	2	0,3,6,8,9,12,13,16,17
23	5	0,3,5,6,7,9,13,17,18,22
29	20	0,1,2,4,6,7,9,11,12,13,17,18,21,24,25
31	21	0,1,5,8,12,13,16,18,19,21,22,23,25,26,28

Отримані результати дозволяють просіювати кандидатів на $\varphi(n)$ у вибраному інтервалі натуральних чисел.

Зрозуміло, що $\varphi(n)$ може бути лише серед чисел, кратних 24: 0, 24, 48, 72, 96, ... На інтервалі від 0 до 119 ($120=24 \cdot 5$) є лише три такі кандидати: 0, 24, 96. На інтервалі від 120 до 239 такими кандидатами є 120, 144, 216. Тобто кандидати на інтервалі [120,239] отримані додаванням числа 120 до чисел-кандидатів на інтервалі [0,119]. І на будь-якому інтервалі , який починається з числа, кратного 120 та має довжину 120, є три таких кандидати.

Кількість можливих кандидатів на інтервалах різної довжини наведена нижче.

Довжина інтервалу	Кількість кандидатів
$120=24 \cdot 5$	3
$840=24 \cdot 5 \cdot 7$	9
$9240=24 \cdot 5 \cdot 7 \cdot 11$	54
$120120=24 \cdot 5 \cdot 7 \cdot 11 \cdot 13$	324
$2042040=24 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17$	2916

Зазначимо, що після описаного просіювання можна використати таке додаткове просіювання. Оскільки $\varphi(n)$ є значенням функції Ейлера числа n, то $u^{\varphi(n)} \equiv 1 \bmod n$ для будь-якого натурального u. Зокрема, візьмемо $u=2$. Елементи 2^0, 2^1, 2^2,..., 2^{575} мають лише один відмінний від нуля біт. Звідси можна зробити такий висновок. Якщо 2^v та 2^{v+w} ($w \le 575$) за модулем n мають хоча би два відмінні від нуля біти, то на проміжку $[v, v+w]$ немає числа $\varphi(n)$.

Схема просіювання усього інтервалу $[n-2,1\sqrt{n}, n-1,9\sqrt{n}]$ полягає в такому:

1) увесь інтервал розбивається на підінтервали фіксованої довжини, скажімо, по 2042040 чисел. Просіюємо ці підінтервали у такому порядку: один підінтервал ліворуч від середини всього інтервалу, один підінтервал праворуч – і так чергуємо підінтервали доти, доки не знайдемо значення $\varphi(n)$;

2) кандидатів на підінтервалах визначаємо додаванням до наперед обчислених кандидатів на початковому підінтервалі числа, кратного довжині підінтервалу (у нашому випадку 2042040);

3) отриманих на попередньому кроці кандидатів піддаємо описаному вище додатковому просіюванню.

Результатом є число, яке витримує всі просіювання.

5. Висновки

Запропоновано підхід до криптоаналізу системи RSA, при якому немає потреби працювати з матрицями великого розміру, зводячи докупи результати з різних комп'ютерів. При такому підході криптоаналіз системи RSA можна розподілити між багатьма повністю незалежними комп'ютерами, до швидкодії та об'єму оперативної пам'яті яких не ставлять якихось особливих умов. Результат може бути отриманий на одному з цих комп'ютерів.

У цій роботі лише накреслено шлях, ідучи яким можна було б зламати сучасну складну систему RSA.Отримання точніших оцінок, як працює запропонований підхід, є завданням подальших досліджень.

1. Вербіцький О. В. Вступ до криптології. – Львів; 1998. 2. Введение в криптографию/ Под общ. ред. В.В. Ященко – СПб., 2001. 3. S.P.Contini. Factoring Integers with the Self-Initializing Quadratic Sieve, Ms. of Arts Thesis, Georgia University, Athens, Greece, 1997. – 78p. 4. Factorization of a 512-bit RSA modulus./ S.Cavallar, W.M.Lioen, H.J. te Riele, B.Dodson, A.K.Lenstra, P.L.Montgomery, B.Murphy et al. Modelling, Analysis and Simulation (MAS)–R0007, February 29, 2000. 5. Factorization of RSA-140 using the number field sieve./ S.Cavallar, B.Dodson, A.K.Lenstra, P.C.Leyland, W.M.Lioen, P.L.Montgomery, B.Murphy, H.J. te Riele, P.Zimmermann. MAS-R9925, September 30, 1999. 6. www.rsasecurity.com 7. www.crypto-world.com