УДК 511:003.26.09

С.Д. Винничук, д-р техн. наук,
Ин-т проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины,
(Украина, 03164, Киев, ул. Генерала Наумова, 15,
тел. (044) 4249171, e-mail: vynnychuk@i.ua),
А.В. Жилин, канд. техн. наук, **В.Н. Мисько**,
Ин-т специальной связи и защиты информации НТУУ «КПИ»,
(Украина, 01011, Киев, ул. Московская, 45/1,
тел. (044) 2543479, (093) 9922936, e-mail: jhartem@i.ua, vitalik560@yandex.ru)

Факторизация числа $N = pq$ при простых p и q методом дискретного логарифмирования

Предложен метод разложения на множители числа $N = pq$, где p и q — простые, в виде решения задачи определения показателя степени в уравнении $a^x \bmod n = b$. Показано, что предложенный метод и метод Ферма эквивалентны по вычислительной сложности, но число итераций для метода дискретного логарифмирования в $[0{,}5 \log_2 N]$ раз меньше, чем для метода Ферма.

Запропоновано метод розкладання на множники числа $N = pq$, де p і q — прості, у вигляді розв'язку задачі визначення показника ступеня в рівнянні $a^x \bmod n = b$. Показано, що запропонований метод і метод Ферма є еквівалентними за обчислювальною складністю, але кількість ітерацій для методу дискретного логарифмування в $[0{,}5 \log_2 N]$ разів менше, ніж для методу Ферма.

Ключевые слова: факторизация, метод Ферма, RSA алгоритм, вычислительная сложность.

В настоящее время определенное место в общем множестве криптоалгоритмов (КА), используемых для защиты информации, получили асимметричные криптоалгоритмы (АКА), принципы построения которых впервые сформулированы У. Диффи и М. Хеллманом [1]. Эти принципы основаны на понятиях «односторонняя функция» и «односторонняя функция с потайным ходом (лазейкой)». Разработано большое число АКА, криптографическая стойкость которых основана на трудоемкости вычислительной задачи обращения односторонних функций, лежащих в основе математической модели построения данного АКА [2—5].

Асимметричные КА используются в средствах криптографической защиты информации (КЗИ) информационно-телекоммуникационных сис-

тем (ИТС), как правило, для распределения ключей, шифрования передаваемых ключей и формирования общего секретного ключа; обеспечения аутентификации абонентов и электронной цифровой подписи; генерации криптографически сильных псевдослучайных последовательностей; генерации параметров ключевых массивов, используемых для создания ключей средств КЗИ и др. [6—8].

Из всех АКА алгоритм RSA, стойкость которого основана на трудоемкости вычислительной задачи факторизации больших чисел, — наиболее распространен в зарубежных криптосистемах и стал стандартом де-факто для многих криптографических приложений [9]. Использование RSA алгоритма для решения задач КЗИ в ИТС рекомендовано рядом международных и национальных стандартов, например ISO/IEC 11166-2:1994, 18033-2:2006 и 9796-2:2010, IEEE Std 1363-2000 и 1363a-2004, PKCS # 1, RFC 2437, ANSI X9.44, FIPS 186-3:2009, ITU-T X.509, PEM и др. Кроме того, алгоритм RSA рекомендован некоторыми стандартами банковских систем электронных платежей, S.W.I.F.T и ANSI X9.31, белорусским стандартом СТБ 34.101.22-2009 и австралийским стандартом управления ключами AS2805.6.5.3. В связи с этим изучение криптостойкости алгоритма RSA является актуальной задачей.

Результаты исследования методов криптографического анализа RSA алгоритма приведены в работах [10—12]. Однако в [13] показано, что известные способы компрометации алгоритма RSA работают только применительно к определенным практическим его реализациям. Как правило, эти способы, в общем случае, не являются более эффективными, чем решение задачи факторизации.

В настоящее время разработаны методы факторизации [8], имеющие как экспоненциальную, так и субэкспоненциальную вычислительные сложности. При этом многие современные методы факторизации основаны на ряде фундаментальных соотношений из классического алгоритма Ферма, используемого при факторизации чисел, для которых отношение их множителей близко к единице.

Рассмотрим метод факторизации чисел $N = pq$, где p и q — простые, с помощью дискретного логарифмирования, которое в $[0,5 \log_2 N]$ раз эффективнее метода Ферма.

Алгоритм факторизации, основанный но дискретном логарифмировании. Согласно теореме Эйлера для произвольного числа m и взаимно простого с ним основания a справедливо равенство $a^{\varphi(m)} \bmod m = 1$, где φ — функция Эйлера. Для чисел вида $N = pq$ при простых p и q значение функции Эйлера определяется из соотношения $\varphi(N) = (p-1)(q-1)$. Тогда можно записать

$$a^N \bmod N = a^{(p-1+1)(q-1+1)} \bmod N = a^{\varphi(N)+(p+q-1)} \bmod N = a^{p+q-1} \bmod N. \quad (1)$$

Если c — величина, обратная $a^{p+q-1} \bmod N$, то последнюю величину ищут из условия выполнения равенства $(a^{p+q-1} c) \bmod N = 1$. При известном значении $c = a^{-(p+q-1)} \bmod N$ величину $s = p + q - 1$ находим из уравнения

$$a^s c \bmod N = 1. \tag{2}$$

Уравнение (2) имеет множество решений, среди которых будем искать $s = p + q - 1$ из условия, что при $x = (s+1)/2$ в соотношении Ферма

$$x^2 = N + y^2 \tag{3}$$

y — целое положительное число. Тогда $p = x + y$ и $q = x - y$. В уравнении (2) основание a может быть произвольным числом в диапазоне от двух до $N - 1$. Пусть $a = 2$. Тогда с учетом нечетности N получим

$$2^{-1} \bmod N = \frac{N+1}{2} \bmod N = \frac{N+1}{2}, \tag{4}$$

так как $\left(2\frac{N+1}{2}\right) \bmod N = (N+1) \bmod N = 1$. Из соотношений (1) и (4) следует

$$2^{-(p+q-1)} \bmod N = \left(\left(\frac{N+1}{2}\right)^{p+q-1}\right) \bmod N = \left(\frac{N+1}{2}\right)^N \bmod N,$$

т.е. задача разложения числа $N = pq$ на простые множители p и q сводится к решению уравнения (2) при $a = 2$, в котором величина $c = \left(\frac{N+1}{2}\right)^N \bmod N$ может быть определена заранее, так как значение N известно.

Простейший алгоритм решения уравнения (2) следующий

А л г о р и т м А1.

1. Определяем величину $z = (2^{t_0} \bmod N\, c) \bmod N$, где $t_0 < p + q$. Счетчик итераций $i = 0$.

2. Проверяем условие $z = 1$. Если оно выполнено, то переходим к п. 4, иначе — к п. 3.

3. При $z = (2^1 z) \bmod N$, $i = i + 1$, переходим к п. 2.

4. При $x = (t_0 + i)/2$ из соотношения (3) определяем значение y. Если y — целое положительное число, то $p = x + y$ и $q = x - y$, а иначе — переходим к п. 3.

При выборе начального приближения t_0 исходим из того, что оно должно быть близким к решению и для него должно выполняться условие $t_0 < p + q$. С учетом того, что

$$N = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 < \left(\frac{p+q}{2}\right)^2,$$

начальное значение t_0 принимаем равным $\lfloor 2\sqrt{N} \rfloor$. В алгоритме А1 число итераций составляет

$$T_{A1}(N) = i = p + q - [2\sqrt{N}] \approx p + q - 2\sqrt{N} = p + q - 2\sqrt{pq} = (\sqrt{p} - \sqrt{q})^2.$$

Следовательно, вычислительная сложность метода оценивается величиной порядка $O(N)$.

Число итераций в алгоритме А1 можно уменьшить с помощью дополнительного учета свойств модуля, а именно: модуль произведения чисел равен модулю произведения модулей этих чисел. Пусть для некоторого a выполнено условие $a \bmod N = 1$. Тогда $(2a) \bmod N = 2$ при $N > 2$, $(4a) \bmod N = 4$ при $N > 4$ и так далее. В общем случае при условии $1 \le k \le [\log_2 N]$ справедливо равенство

$$(2^k a) \bmod N = 2^k, \qquad (5)$$

так как $(2^k a) \bmod N = ((2^k \bmod N)(a \bmod N)) \bmod N = (2^k 1) \bmod N = 2^k$.

Пусть $K = [\log_2 N]$. Покажем, что используя соотношения (5), число итераций в алгоритме А1 можно уменьшить в K раз.

А л г о р и т м А2.

1. Определяем величину $z = (2^{t_0} \bmod N\, c) \bmod N$, где $t_0 = \lfloor 2\sqrt{N} \rfloor$. Счетчик итераций $i = 0$.

2. Проверяем условие $z = 2^m$, где $0 \le m \le K$. Если оно выполнено, то переходим к п. 4, а иначе — к п. 3.

3. При $z = (2^K z) \bmod N$, $i = i + 1$, переходим к п. 2.

4. Определяем $p + q$ из соотношения $p + q = x_0 + 1 + Ki - m$. При $p + q = 2x$ из соотношения (3) определяем значение y. Если y — целое число, то $p = x + y$ и $q = x - y$, иначе — переходим к п. 3.

В алгоритме А2 число итераций уменьшается в K раз (вследствие изменения в п. 3) по сравнению с алгоритмом А1, но увеличивается сложность одной итерации (умножение z на большое число и сравнение не только с единицей). Но такое усложнение можно нивелировать выбором разрядной базы для работы с большими числами. Если в качестве базы выбрать основание два, то умножение на 2^K сводится к сдвигу регистров на K единиц. Сравнение с 2^m эквивалентно тому, что в остатке от деления z на N при разложении по базе два будет только одно значение разряда, равное единице. Поэтому вычислительная сложность алгоритма А2 в K раз меньше, чем алгоритма А1, хотя также составит величину порядка $O(N)$:

$$T_{A2}(N) = i = (p + q - [2\sqrt{N}])/K \approx (p + q - 2\sqrt{N})/K = (\sqrt{p} - \sqrt{q})^2/K. \qquad (6)$$

Алгоритм А2 реализован программно с использованием библиотеки больших чисел GMP [14].

Примеры работы алгоритма А2. П р и м е р 1. Пусть $p = 7$, $q = 5$. Тогда $n = p\,q = 7 \cdot 5 = 35$. Следуя алгоритму А2, определяем:

$$c = \left(\frac{N+1}{2}\right)^N \bmod N = 18^{35} \bmod 35 = 2;$$

$$K = [\log_2 N] = [\log_2 35] = 5;$$

$$t_0 = \left\lfloor 2\sqrt{N} \right\rfloor = \left\lfloor 2\sqrt{35} \right\rfloor = [2 \cdot 5{,}916] = 11;$$

$$z = (2^{t_0} \bmod N\, c) \bmod N = (2^{11} \bmod 35 \cdot 2) \bmod 35 = 1.$$

Устанавливаем значение счетчика итераций $i = 0$.

Проверяем условие $z = 2^m$, где $0 \le m < K$. В данном случае $z = 1 = 2^0$. Следовательно, условие п. 2 выполнено, переходим к п. 4.

Определяем $p + q = t_0 + 1 + K\,i - m = 11 + 1 + 0 \cdot 5 - 0 = 12$.

Определяем $p - q = \sqrt{(p+q)^2 - 4N} = \sqrt{(12)^2 - 4 \cdot 35} = \sqrt{144 - 140} = \sqrt{4} = 2$.

Следовательно, большее из чисел

$$p = ((p+q) + (p-q))/2 = (12+2)/2 = 7;\ q = 5.$$

П р и м е р 2. Пусть $p = 1031$, $q = 31$. Тогда $n = pq = 1031 \cdot 31 = 31961$. Следуя алгоритму А2, определяем:

$$c = \left(\frac{N+1}{2}\right)^N \bmod N = 15981^{31961} \bmod 31961 = 6960;$$

$$K = [\log_2 N] = [\log_2 31961] = 14;$$

$$t_0 = \left\lfloor 2\sqrt{N} \right\rfloor = \left\lfloor 2\sqrt{31961} \right\rfloor = [2 \cdot 178{,}7764] = 357;$$

$$z = (2^{t_0} \bmod N\, c) \bmod N = (2^{357} \bmod 31961 \cdot 6960) \bmod 35 = 8682.$$

Устанавливаем значение счетчика итераций $i = 0$.

Проверяем условие $z = 2^m$, где $0 \le m < K$. В данном случае $z \ne 2^m$, где $0 \le m < K$, поэтому выполняем операции п. 2 и 3 до тех пор, пока не будет выполнено условие $z = 2^m$.

Первый раз условие $z = 2^m$ выполняется при $i = 14$. Тогда $z = 128 = 2^7$.

Следовательно, переходим к п. 4.

Определяем $p + q = t_0 + 1 + Ki - m = 357 + 1 + 14 \cdot 14 - 7 = 547$, что не является решением. Поэтому переходим к п. 3 и продолжаем поиск $p + q$. Сле-

дующее выполнение условия $z = 2^m$, где $0 \le m < K$, достигается при $i = 51$. Тогда $z = 1024 = 2^{10}$, откуда находим $p + q = t_0 + 1 + Ki - m = 357 + 1 + 14 \cdot 51 - 10 = 1062$, что является решением.

Определяем

$$p - q = \sqrt{(p+q)^2 - 4N} = \sqrt{(1061)^2 - 4 \cdot 31961} = \sqrt{1000000} = 1000;$$

$$p = ((p+q) + (p-q))/2 = (1062 + 1000)/2 = 1031;\ q = 31.$$

Сравним вычислительную сложность разработанного метода и метода Ферма. Для метода Ферма вычислительная сложность оценивается по числу вариантов выбора пробных значений и описывается величиной $T_{\Phi}(N) = \frac{(\sqrt{p} - \sqrt{q})^2}{2}$. Аналогичная оценка для предлагаемого метода описывается зависимостью (6). Следовательно, при $K = [\log_2 N]$ предложенный метод факторизации в $[0{,}5 \log_2 N]$ раз эффективнее, чем метод Ферма. И эффективность его увеличивается с возрастанием значения N.

Для качественной оценки эффективности алгоритма А2 по сравнению с алгоритмом квадратичного решета получены сравнительные характеристики его трудоемкости, оцениваемые согласно (6) величиной, характеризующей вычислительную сложность алгоритмов квадратичного решета T_{QS}: $T_{QS}(N) = \exp((\ln N)^{1/2} (\ln \ln N)^{1/2})$. Результаты получены с использованием системы компьютерной алгебры Maple и приведены на рис. 1 (см. вклейку), из которого видно, что разработанный метод факторизации имеет меньшую вычислительную сложность при условии близости сомножителей p и q, т.е. когда их отношение близко к единице. Область более высокой эффективности разработанного метода по сравнению с методом квадратичного решета лежит внутри кривой пересечения графиков вычислительной сложности метода квадратичного решета (зеленый цвет) и алгоритма А2 (синий цвет).

Сравним также вычислительную сложность разработанного метода и метода решета числового поля $T_{GFNS}(N) = \exp(1{,}92 (\ln N)^{1/3} (\ln \ln N)^{2/3})$. Результаты сравнений получены с использованием системы компьютерной алгебры Maple и приведены на рис. 2 (см. вклейку), из которого видно, что разработанный метод факторизации при условии близости сомножителей p и q эффективнее метода квадратичного решета.

Выводы

При решении задачи разложения большого числа $N = pq$, где p и q — простые, на множители, когда отношение множителей p и q близко к единице, предложенный метод факторизации во всем диапазоне чисел $N = pq$ эффективнее метода Ферма, что теоретически обосновано.

Вычислительная сложность алгоритма факторизации чисел вида $N = pq$, где p и q — простые, в методе дискретного логарифмирования оценивается величиной $O(N)$. Поэтому при больших значениях N его эффективность ниже, чем методов квадратичного решета и решета числового поля для случая произвольных простых множителей p и q. В то же время, его эффективность очевидна в случае, когда отношение множителей p и q близко к единице.

The authors have proposed a method for factoring the number of the form $N = pq$, where p and q are simple, as the solution to the problem of determining the exponent in the equation $a^x \mathrm{mod} n = b$. It is shown that the proposed method and the method of Fermat are equivalent in terms of computational complexity, but the number of iterations of the discrete logarithm is $[0{,}5\log_2 N]$ times less than for the method of Fermat.

СПИСОК ЛИТЕРАТУРЫ

1. *Diffie W., Hellman M.* New Directions in Cryptography // IEEE Trans. Inf. Theory. — 1976. — IT-22, № 6. — P. 644—654.
2. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. — М. : Мир, 1982. — 416 с.
3. *Шенхаге А., Штрассен В.* Быстрое умножение больших чисел // Кибернетический сборник. — 1973. — Вып. 2. — С. 87—98.
4. *Pomerance C., Smith W., Tuler R.* A pipe-line architecture for factoring large integers with the quadratic sieve algorithm // SIAM Journal of Computing. — 1988. — Vol. 17. — P. 387—403.
5. *Мао Венбо.* Современная криптография: теория и практика: Пер. с англ — М. : Изд. дом «Вильямс», 2005. — 768 с .
6. *Саломаа А.* Криптография с открытым ключом: Пер. с англ. — М. : Мир, 1996. — 318 с.
7. *Ростовцев А.Г., Маховенко Е.Б.* Теоретическая криптография. — СПб : АНО НПО «Профессионал», 2004. — 480 с.
8. *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии. — М. : МЦНМО, 2003. — 328 с.
9. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. — М. : Триумф, 2002. — 816 с.
10. *Song Y.Yan.* Cryptanalytic attacks on RSA. — Springer Science and Business Media, Inc., 2008. — 255 p.
11. *Авдошин С.М., Савельева А.А.* Криптоанализ: современное состояние и перспективы развития // Новые технологии. Приложение к журналу «Информационные технологии». № 3. — М. : Машиностроение, 2007. — 24 с.
12. *Горбенко И.Д., Долгов В.И., Потий А.В., Федорченко В.Н.* Анализ каналов уязвимости системы RSA // Безопасность информации. — 1995. — № 2. — С. 22—26.
13. *Brown D.R.L.* Breaking RSA May Be As Difficult As Factoring// Электронный ресурс. — Режим доступа: http://www.pgpru.com/novosti/2005/1026vzlomrsabezfaktoriza ciirealen noneeffektiven.
14. *The GNU* Multiple Precision Arithmetic Library. Edition 5.1.1.11 February 2013. [Электронный ресурс]. — Режим доступа: http://gmplib.org/gmp-man-5.1.1.pdf.

Поступила 30.07.13
после доработки 02.09.13

ВИННИЧУК Степан Дмитриевич, д-р техн. наук, ст. науч. сотр., и.о. зав. отделом ИПМЭ им. Г.Е. Пухова НАН Украины. В 1977 г. окончил Черновицкий государственный университет. Область научных исследований — моделирование тепловых и гидравлических процессов в системах кондиционирования воздуха и процессов динамического изменения частоты в элект роэнергетических системах, теория алгоритмов.

ЖИЛИН Артем Викторович, канд. техн. наук, ведущий научный сотрудник Ин-та специальной связи и защиты информации Национального технического университета Украины «КПИ». В 2005 г. окончил Житомирский военный институт радиоэлектроники им. С.П. Королева. Область научных исследований — асимметричная криптография, численные методы и алгоритмы факторизации, защита информации.

МИСЬКО Виталий Николаевич, курсант Ин-та специальной связи и защиты информации Национального технического университета Украины «КПИ». Область научных исследований — численные методы и алгоритмы факторизации.