COMPUTATIONAL METHODS

УДК 003.26.09:511.1

СТЕПАН ВИННИЧУК,
ЕВГЕНИЙ МАКСИМЕНКО

ФОРМИРОВАНИЕ НЕРАВНОМЕРНЫХ ПРИРАЩЕНИЙ ДЛЯ БАЗОВОГО ОСНОВАНИЯ МОДУЛЯ В ЗАДАЧЕ ФАКТОРИЗАЦИИ МЕТОДОМ ФЕРМА

Одним из способов обеспечения заданного уровня защиты информации является применение алгоритмов асимметричного шифрования, криптографическая стойкость которых основана на трудности выполнения задачи факторизации. Современные методы разложения больших чисел на множители базируются на фундаментальных представлениях классического алгоритма факторизации Ферма. К числу возможных направлений ускорения метода Ферма можно отнести уменьшение количества арифметически сложных операций извлечения квадратного корня за счет реализации процедуры модульного деления с использованием нескольких оснований модуля (метод решета). Данная модификация метода позволяет исключать из рассмотрения варианты допустимых значений X, которые не удовлетворяют условию $X^2 = N + Y^2$. В процессе проведенных исследований модифицированного метода факторизации Ферма было определено, что при поиске решения уравнения $Y^2 = X^2 - N$ возможно использование не предварительно просеянных больших значений X, а приращений к ним, являющихся малыми числами. Предложен способ формирования данных о неравномерных приращениях для допустимых значений X относительно базового основания модуля при факторизации чисел методом Ферма, позволяющий существенно уменьшить количество проверяемых значений. Кроме того, применение данного алгоритма обеспечивает не более одной операции сложения многоразрядных чисел. Все же другие операции проводятся с малыми числами, как правило, не превышающими 2^{15}.

Ключевые слова: факторизация, разложение на множители, метод Ферма, прореживание, метод решета, ускорение.

Вступление. В информационно-телекоммуникационных системах для обеспечения криптографической защиты информации применяется RSA алгоритм, что вызывает интерес к его криптоанализу. В работе [1] показано, что известные примеры компрометации RSA алгоритма не являются эффективней задачи факторизации. Основные используемые методы решения задачи факторизации представлены в [2-5]. Известно, что идеи классического алгоритма Ферма положены в основу таких современных методов факторизации как метод квадратичного решета и решета числового поля. Данный факт позволяет утверждать, что усовершенствования метода Ферма могут послужить основой для ускорения и других методов факторизации.

Постановка задачи. Пусть задано $N = p \cdot q$ – составное нечетное число, которое следует разложить на множители, где p и q некоторые нечетные числа, не обязательно являющиеся простыми.

Согласно исходному варианту метода Ферма для определения p и q решают уравнение

$$X^2 = N + Y^2, \qquad (1)$$

где X и Y – целые положительные числа. Поскольку $X > \sqrt{N}$, то ее можно представить в виде $X = \left(\left\lfloor \sqrt{N} \right\rfloor + 1\right) + x = x_0 + x$, где x = 0, 1, 2, …. Тогда решение уравнения (1) получают перебором значений x до тех пор, пока остаток $X^2 - N$ не окажется полным квадратом целого

числа. Следовательно, исходный алгоритм метода Ферма можно представить как цикл, на каждом шаге которого увеличивается на единицу число x, определяется разность $r=(x_0+x)^2-N$ до тех пор, пока r не окажется квадратом целого числа.

Уже в работах Ферма и при дальнейших исследованиях метода Ферма было показано, что часть значений x можно исключить из рассмотрения за счет процедуры просеивания [4, 6]. В работах [7, 8], показано, что при этом важное значение имеет первичное просеивание по основанию большого основания модуля bb, но способ формирования данных, используемых при просеивании, требует дополнительных исследований. В частности, в настоящей работе рассматриваются случаи использования больших bb, значения которые могут превышать 2^{31}.

Непосредственное определение данных о неравномерных приращениях неизвестной для основания модуля первичного просеивания. В работе [7] представлен алгоритм формирования данных для первичного основания bb, используемых при просеивании (далее для упрощения будем обозначать его алгоритм А), требующий при реализации выполнения следующих операций:

А1. Определение множества $MN(N) = \{N \bmod b_i\}_{i=1}^{m}$ – возможных неотрицательных остатков от деления N на bb.

А2. Для основания bb определение максимального возможного числа остатков от деления допустимых X на основание модуля bb – числа $r_{\max}(bb)$.

А3. Определение квадратичных остатков для основания bb (формируется массив $Mbb[bb]$).

А4. Для элемента $N \bmod bb$ множества $MN(N)$ определение множества $Xbb(N \bmod bb)$ допустимых $X \bmod bb$, т.е. тех, для которых $((X \bmod bb)-N \bmod bb) \bmod bb$ является квадратичным остатком по модулю bb. В массиве $Xbb(N)$ число элементов при разных значениях $N \bmod bb$ может быть разным, равным $r(N,bb)$, но максимальное их количесиво не превышает $r_{\max}(bb)$. Если в ячейке $Xbb(N)[0]$ указать число $r(N,bb)$, то первоначально в массиве $Xbb(N)$ записываются допустимые значения $X \bmod bb$ в порядке их возростания, а после, на их основании, формируется последовательность шагов переменной длины между допустимыми X по правилам:

$$\begin{cases} a = Xbb(N)[1]; \\ Xbb[k] = Xbb(N)[k+1] - Xbb(N)[k]; \qquad (k = 1 \div (Xbb[0]-1)). \\ Xbb(N)[Xbb(N)[0]] = a - Xbb(N)[Xbb(N)[0]] + bb \end{cases} \quad (2)$$

В дальнейшем отношение $z(N,bb) = bb / r(N,bb)$ будем называть коэффициентом ускорения при прореживании пробных x для числа N с использованием основания модуля bb.

Размерности всех используемых в алгоритме А массивов значений непосредственно связаны с основанием модуля bb, поэтому будем говорить, что алгоритм А – это алгоритм непосредственного определение данных о неравномерных приращениях неизвестной для основания модуля первичного просеивания.

Следует отметить, что реализация алгоритма А уже при значениях bb порядка 10^9 и больше требует решения проблем с выделением необходимого объема памяти и временем его выполнения. В первую очередь достаточно трудоемким оказывается определение множества $Mbb[bb]$, поскольку требуется определять значения $t^2 \bmod bb$ для большого количества значений t=0÷(bb/2). Так в случае $t > 2^{16}$ число t^2 превышает ограничение для беззнакового целого типа long, что еще увеличивает время вычислений. Кроме того, полученные значения квадратичных остатков $t^2 \bmod bb$ необходимо отсортировать в порядке их роста, после чего, для каждого t от 0 до bb-1 проверять является ли значение $(t^2 - N \bmod bb) \bmod bb$ квадратичным остатком. Поэтому при bb больших требуется разработка более эффективного алгоритма поиска допустимых $X \bmod bb$.

В работе [7] приведены экспериментальные данные о том, что при увеличении *bb*, сопровождающегося увеличением коэффициентом ускорения, существенно меняется время вычислений. Так при прочих равных условиях время расчета при *bb*=277200 в 1.74 ÷ 1.78 раза (в среднем в 1.77 раза) меньше, чем в случае *bb* = 25200 (25200 = 277200/11). А при *bb*=3600 (3600=277200/11/7) время расчета увеличилось в 3.06 ÷ 3.17 раза (в среднем в 3.11 раза). Поэтому увеличение первичного основания модуля при решении задачи факторизации представляется актуальным, следовательно, актуальной является и задача разработки эффективного алгоритма определения данных о неравномерных приращениях неизвестной для основания модуля первичного просеивания. Для его построения проведем анализ структуры *bb* и ее связи с коэффициентом ускорения $z(N,bb)=bb/|r|(N,bb)$.

Структура эффективного модуля *bb* и ее влияние на величину коэффициента ускорения. В работе [8] показано, что значение эффективного модуля *bb* следует искать среди чисел вида

$$bb=\prod_{k=1}^{m(bb)} p_k^{s_k}, \qquad (3)$$

где $m(bb)$ – количество простых чисел – множителей *bb*, p_k (k=1÷ $m(bb)$) – простые числа – множители *bb*, s_k (k=1÷$m(bb)$) – показатели степеней p_k причем для минимально возможных значений ускорений при одинаковых суммах числа показателей степеней большее значение будет для случая, когда при росте p_k (k=1÷ $m(bb)$) не растут s_k (k=1÷$m(bb)$). Однако следует отметить, что для коэффициентов ускорения, превышающих минимальные, при определенных значениях $N \bmod bb$ данное правило может нарушаться.

Рассмотрим ряд примеров оснований модуля *bb* и определим коэффициенты ускорения для возможных $N \bmod bb$, не имеющих общих делителей с *bb*. Зависимости коэффициента ускорения $z(N,bb)$ от значений *bb* представлены в табл. 1.

Таблица 1 – Значения коэффициентов ускорения при изменениях *bb* для взамно простых с 2, 3 и 5 *N*mod*bb* меньших 60 в сравнении *b*0=60.

bb	60	180	240	300	540	720	900	8640
	*b*0	*b*0*3	*b*0*4	*b*0*5	*b*0*9	*b*0*12	*b*0*15	*b*0*144
Степени простых множителей N	2^2, 3^1, 5^1	2^2, 3^2, 5^1	2^4, 3^1, 5^1	2^2, 3^1, 5^2	2^2, 3^3, 5^1	2^4, 3^2, 5^1	2^2, 3^2, 5^2	2^6, 3^3, 5^1
*N*mod*bb*	$z(N,bb)$							
1	5	15	10	10,71	22,5	30	32,14	90
7	7,5	22,5	15	7,5	33,75	45	22,50	67,5
11	10	10	20	21,43	10	20	21,43	20
13	7,5	22,5	15	7,5	33,75	45	22,5	135
17	15	15	30	15	15	30	15	60
19	5	15	10	10,71	22,5	30	32,14	45
23	15	15	30	15	15	30	15	30
29	10	10	20	21,43	10	20	21,43	40
31	5	15	10	10,71	22,5	30	32,14	45
37	7,5	22,5	15	7,5	33,75	45	22,5	135
41	10	10	20	21,43	10	20	21,43	40
43	7,5	22,5	15	7,5	33,75	45	22,5	67,5
47	15	15	30	15	15	30	15	30
49	5	15	10	10,71	22,5	30	32,14	90
53	15	15	30	15	15	30	15	60
59	10	10	20	21,43	10	20	21,43	20

В табл. 1 представлены все варианты коэффициентов ускорения, которые встречаются при различных остатках *N*mod*bb*, взаимно простых с *bb*.

Из анализа данных табл. 1 следует, что в ряде случаев рост коэффициентов ускорения при увеличение *bb* имеет определенную закономерность. Так при умножении *b*0 на 3 часть коэффициентов ускорения увеличивается в три раза, либо их значения не меняются. При умножении *b*0 на 4 коэффициенты ускорения всегда увеличиваются в два раза. При умножении *b*0 на 5 часть коэффициентов ускорения увеличивается на одну и ту же величину, примерно равную 2,14 раза, либо их значения не меняются.

На основании численных экспериментов было установлено, что для коэффициентов ускорения $z(bb,k)$, где $bb = \prod_{i=1}^{m(bb)} p_i^{s_i}$, а k – взаимно простое с bb и $kc<bb$, соблюдается равенство

$$z(bb,k) = \prod_{i=1}^{m(bb)} z(p_i^{s_i}, k \bmod p_i^{s_i}). \quad (4)$$

Второе важное свойство для коэффициентов ускорения состоит в том, что при $m>0$ и $p > 2$ неравенство $z(p^{m+2},k) > z(p^{m+1},k)$ имеет место только для части (или всех) тех k, для которых $z(p^{m+1},k) > z(p^m,k)$. При этом $z(p^2,k) > z(p,k)$ только для тех k, для которых $z(p,k) = z_{\min}(p)$. В случае $p = 2$ второе свойство имеет место для m > 4.

Для иллюстрации описанных свойств определим значения коэффициентов ускорения для $z(p^m,k)$ при p=2 для m=1÷6, p=3 и m=1÷3, p=5 и m=1÷5, p=7 и m=1÷2. Значения коэффициентов ускорения приведены в табл. 2-5.

Таблица 2 – Значения коэффициентов ускорения для $bb=2^m$ при m=3÷7

bb	$z_{\min}$	$z_{\max}$								
8	2	4	*k*	1	3	5	7				
			z(*bb*,*k*)	2	4	2	4				
16	4	4	*k*	1	3	5	7	9	11	13	15
			z(*bb*,*k*)	4	4	4	4	4	4	4	4
32*)	4	8	*k*	1	3	5	7	9	11	13	15
			z(*bb*,*k*)	4	4	**8****)	4	4	4	**8**	4
64*)	4	8	*k*	1	3	5	7	9	11	13	15
			z(*bb*,*k*)	8	4	**8**	4	8	4	**8**	4
128*)	4	10.67	*k*	1	3	5	7	9	11	13	15
			z(*bb*,*k*)	10.67	4	**8**	4	10.67	4	**8**	4

Примечания: *) – значения ускорений повторяются с шагом 8 по *k*, **) – значения коэффициентов ускорений равны $z_{\max}(2^5)$ и не меняются при увеличении *bb* для всех *k* mod 8 = 5.

Таблица 3 – Значения коэффициентов ускорения для $bb=3^m$ при m=1÷3

bb	$z_{\min}$	$z_{\max}$							
3	1.5	3	*k*	1	2				
			z(*bb*,*k*)	1.5	**3***)				
9	3	4.5	*k*	1	2	4	5	7	8
			z(*bb*,*k*)	4.5	**3**	4.5	**3**	4.5	**3**
27	3	6.75	*k*	1	2	4	5	7	8
			z(*bb*,*k*)	6.75	**3**	6.75	**3**	6.75	**3**
			k	10	11	13	14	16	17
			z(*bb*,*k*)	6.75	**3**	6.75	**3**	6.75	**3**
			k	19	20	22	23	25	26
			z(*bb*,*k*)	6.75	**3**	6.75	**3**	6.75	**3**

Примечания: *) – значения коэффициентов ускорений не меняются при увеличении *bb* для всех *k* mod3=2.

Таблица 4 – Значения коэффициентов ускорения для $bb=5^m$ при $m=1\div3$

bb	z_{min}	z_{max}								
5	1.67	2.5	*k*	1	2	3	4				
			z(*bb*,*k*)	1.67	**2.5**[**)]	**2.5**	1.67				
25[*)]	2.5	3.57	*k*	1	2	3	4	6	7	8	9
			z(*bb*,*k*)	3.57	**2.5**	**2.5**	3.57	3.57	**2.5**	**2.5**	3.57
125[*)]	2.5	4.03	*k*	1	2	3	4	6	7	8	9
			z(*bb*,*k*)	4.03	**2.5**	**2.5**	4.03	4.03	**2.5**	**2.5**	4.03

Примечания: [*)] – значения ускорений повторяются с шагом 5 по *k*, [**)] – значения коэффициентов ускорений не меняются при увеличении *bb* для всех *k* mod 5 =2 и *k* mod 5 =3.

Таблица 5 – Значения коэффициентов ускорения для $bb=7^m$ при $m=1\div3$

bb	z_{min}	z_{max}							
7	1.75	2.33	*k*	1	2	3	4	5	6
			z(*bb*,*k*)	1.75	1.75	**2.33**[**)]	1.75	**2.33**	**2.33**
49[*)]	2.33	3.57	*k*	1	2	3	4	5	6
			z(*bb*,*k*)	3.06	3.06	**2.33**	3.06	**2.33**	**2.33**
343[*)]	2.33	3.57	*k*	1	2	3	4	5	6
			z(*bb*,*k*)	3.236	3.236	**2.33**	3.236	**2.33**	**2.33**

Примечания: [*)] – значения ускорений повторяются с шагом 7 по *k*, [**)] – значения коэффициентов ускорений не меняются при увеличении *bb* *k* для всех *k*mod 7=3, *k*mod7=5 и *k*mod7=6.

Следует отметить, что при увеличении показателя степени $p=2$ до значений $m=8, 9, 10, 11$ и далее меняются коэффициенты ускорений только для *k*mod8=1. Они принимают одинаковое значение равное $z_{max}(2^m)$ для каждого из *m*, где $z_{max}(2^8)=16$, $z_{max}(2^9)=18.29$, $z_{max}(2^{10})=21.23$, $z_{max}(2^{11})=22.26$. Коэффициенты ускорений не меняются для всех *k* таких, что $z_{max}(2^5)=8$, т.е. при *k*mod8=5. При этом, в отличие от других вариантов *p*, при $m \geq 4$ не меняются также значения коэффициентов ускорения для всех *k* mod4=3, т.е. $z(2^m, k)=4$, если *k*mod4=3 и $m \geq 4$.

В случае простого $p=3$ при увеличении показателя степени до значений $m=4, 5, 6, 7$ и далее меняются коэффициенты ускорений только для *k*mod3=1. Они принимают одинаковое значение равное $z_{max}(3^m)$ для каждого из *m*, где $z_{max}(3^4)=10.13$, $z_{max}(3^5)=11.05$, $z_{max}(3^6)=11.76$, $z_{max}(3^7)=11.89$.

При $p=5$ с увеличением показателя степени до значений $m=4, 5, 6$ и далее меняются коэффициенты ускорений только для *k*mod5=2 и *k*mod5=3. Они принимают одинаковое значение равное $z_{max}(5^m)$ для каждого из *m*, где $z_{max}(5^4)=4.25$, $z_{max}(5^5)=4.27$, $z_{max}(5^6)=4.28$.

При $p=7$ с увеличением показателя степени до значений $m = 4, 5$ и далее меняются коэффициенты ускорений только для *k*mod7=1, *k*mod7=2 и *k*mod7=3. Они принимают одинаковое значение равное $z_{max}(7^m)$ для каждого из *m*, где $z_{max}(7^4)=3.289$, $z_{max}(7^5)=3.293$.

Для дальнейшего анализа будет важно знать точные значения максимального и минимального коэффициентов ускорения, представленные виде отношения двух целых. Такая информация представлена в табл. 6.

Таблица 6 – Точные значения коэффициентов ускорения для *bb* вида p^m при $p = 2, 3, 5$ и 7, для $bb < 17000$

p	*m*	p^m	$z_{min}(p^m)$	$z_{max}(p^m)$	$zz = z_{max}(p^{m+1}) / z_{max}(p^m)$	$p / (zz\text{-}1)$
1	2	3	4	5	6	7
2	1	2	1	2	-	
	2	4	2	2	-	
	3	8	2	4	-	
	4	16	4	4	-	
	5	32	4	8	-	

Продолжение таблицы 6

1	2	3	4	5	6	7
	6	64	4	8	-	
	7	128	4	32 / 3	4 / 3	6
	8	256	4	16	3 / 2	4
	9	512	4	128 / 7	8 / 7	14
	10	1024	4	64 / 3	7 / 6	12
	11	2048	4	512 / 23	24 / 23	46
	12	4096	4	256 / 11	23 / 22	44
	13	8192	4	2048 / 87	88 / 87	176
	14	16384	4	1024 / 43	87 / 86	172
3	1	3	3/2	3	-	
	2	9	3	9 / 2	-	
	3	27	3	27 / 4	3 / 2	6
	4	81	3	81 / 8	3 / 2	6
	5	243	3	243 / 22	12 / 11	33
	6	729	3	729 / 62	33 / 31	93 / 2 > 46
	7	2187	3	2187 / 184	93 / 92	276
	8	6561	3	6561 / 548	138 / 137	411
5	1	5	5/3	5 / 2	-	
	2	25	5/2	25 / 7	-	
	3	125	5/2	125 / 31	35 / 31	155 / 4 > 38
	4	625	5/2	625 / 147	155 / 147	735 / 8 > 91
	5	3125	5/2	3125 / 731	735 / 731	3655 / 4 > 913
	6	15625	5/2	15625 / 3647	3655 / 3647	
7	1	7	7 /4	7 /3	-	
	2	49	7/3	49 / 16	-	
	3	343	7/3	343 / 106	56/53	371 / 3 > 123
	4	2401	7/3	2401 / 730	371 / 365	2555 / 6 > 425
	5	16807	7/3	16807 / 5104	2555 / 2552	17864 / 3 > 5954

По данным табл. 2, 3, 4 и 6 проверим выполнение равенств (4) для коэффициентов ускорения при *bb*=900 и 8640, представленных в табл. 1 для взаимно простых с *bb* чисел *k*<60. Результаты проверки, подтверждающие данные табл. 1, представлены в табл. 7.

Таблица 7 – Результаты проверки выполнения равенства (4) для bb = 900 и 8640

$bb = 900 = 2^2 * 3^2 * 5^2$					$bb = 8640 = 2^6 * 3^3 * 5^1$				
k	$z(2^2,k)^{*)}$	$z(3^2,k)^{*)}$	$z(5^2,k)^{*)}$	$z(bb,k)$	k	$z(2^6,k)^{*)}$	$z(3^3,k)^{*)}$	$z(5^1,k)^{*)}$	$z(bb,k)$
1	2	9/2	25/7	225/7 =32.14	1	8	27/4	5/3	90
7	2	9/2	5/2	45/2 = 22.5	7	4	27/4	5/2	135/2 = 67.5
11	2	3	25/7	150/7=21.43	11	4	3	5/3	20
13	2	9/2	5/2	45/2 = 22.5	13	8	27/4	5/2	135
17	2	3	5/2	15	17	8	3	5/2	60
19	2	9/2	25/7	225/7 =32.14	19	4	27/4	5/3	45
23	2	3	5/2	15	23	4	3	5/2	30
29	2	3	25/7	150/7=21.43	29	8	3	5/3	40
31	2	9/2	25/7	225/7 =32.14	31	4	27/4	5/3	45
37	2	9/2	5/2	45/2 = 22.5	37	8	27/4	5/2	135
41	2	3	25/7	150/7=21.43	41	8	3	5/3	40
43	2	9/2	5/2	45/2 = 22.5	43	4	27/4	5/2	67.5
47	2	3	5/2	15	47	4	3	5/2	30

Продолжение таблицы 7

49	2	9/2	25/7	225/7 =32.14	49	8	27/4	5/3	90
53	2	3	5/2	15	53	8	3	5/2	60
59	2	3	25/7	150/7=21.43	59	4	3	5/3	20

В табл. 6 представлены значения коэффициентов $zz=z_{\max}(p^{m+1}) / z_{\max}(p^m)$, по величине которых можно оценивать относительное увеличение объема памяти при умножении *bb* на *p*, равное *p/zz*. А если исходить из того, что коэффициент ускорения для первой степени простого числа *q* $z_{cp}(q)=2$, то отношение *p*/(*zz*-1) показывает, при каких простых числах, не превышающих *p* / (*zz*-1), отношение объема требуемой памяти для хранения неравномерных приращений к коэффициенту ускорения будет меньшей, чем в случае умножения *bb* на *p*. На основании таких данных уже возможно определять эффективное значение основания первичного модуля *bb*. При этом в случае *p*>2 при равенстве $z(p^{m+1}, N\mathrm{mod}p) = z(p^m, N\mathrm{mod}p)$ следует выбирать минимальное значение *m*=1. В остальных случаях для *p*>2 определять степень *m*, используя значения отношения *p*/(*zz*-1). В случае *p*=2 всегда следует использовать показатель степени $m \geq 4$, поскольку при $m = 4$ всегда коэффициент ускорения равно 4. Далее необходимо определять $z(2^5, N\mathrm{mod}2^5)$ и $z(2^6, N\mathrm{mod}2^6)$. Если имеет место рост коэффициента ускорения, то показатель степени следует увеличивать, т.е. $m \geq 6$. Если $z(2^5, N\mathrm{mod}2^5) = 8$, то *m*=5. Если $z(2^5, N\mathrm{mod}2^5) = z(2^6, N\mathrm{mod}2^6)=4$, то *m*=4.

Алгоритм определения неравномерных приращений для больших значениях первичного модуля *bb*. Правило определения коэффициента ускорения (6) с учетом данных табл. 2-7 и рекомендаций по выбору эффективного первичного модуля при разложении на множитель числа *N*, позволяет найти значение *bb*. Однако при больших значениях *bb* нецелесообразно использовать алгоритмом А для определения неравномерных приращений, соответствующих *bb* и *N*mod*bb*, поскольку формирование массива $Xbb(N)$ и расчет последовательности шагов переменной длины между допустимыми *X* по правилам (2) является достаточно трудоемкой операцией. В связи с чем предлагается новый модифицированный алгоритм А1, позволяющий работать с малыми числами и значительно уменьшить объем вычислений.

Алгоритм А1 описывается следующей последовательностью шагов.

Шаг С1. Число *bb* представить в виде (3): $bb = \prod_{k=1}^{m(bb)} \mathrm{p}_k^{s_k}$, где $p_1=2$, $p_2=3$, $p_3=5$, $p_4=7$, $p_5=11$ и т.д. Определить $n_0 = N\mathrm{mod}bb$, $x_0 = \lfloor\sqrt{N}\rfloor + 1$ и $x_1 = x_0\mathrm{mod}bb$.

Шаг С2. Определить минимальное допустимое значение *X*mod*b* для основания модуля $b = 2^{m_1} = p_1^{m_1}$ при $m_1 \geq 4$ для *N*mod*b*, которое обозначим через x_2, а также все другие допустимые значения *X*mod*b*, и по ним определить приращения для допустимых *X*mod*b* согласно соотношений (2). Пусть множество *M*2 таких приращений содержит *k*2 элементов.

Шаг С3. Для простых $p = 3, 5, 7, 11$ и т.д. в случаях, когда $m_k > 0$, определить все допустимые значения *X*modb_k для оснований модуля $b_k = p_k^{m_k}$ при $m_k > 0$. Сформировать массивы *M*(*k*) каждый разной размерностью b_k, в которых допустимому *X* по модулю b_k соответствует единица, а остальным – нуль.

Шаг С4. Определить приращения для допустимых *X*mod*bb* и начальное (стартовое) минимальное допустимое для основания модуля *bb* значение $Xst > x_0$ согласно следующих правил.

Шаг С4.1. Присвоить переменной *t* начальное значение $t = x_2$, параметру цикла *i* – значение 0, *Xst* =-1, приращению *r* = -1, счетчику *j* номера допустимого *X*mod*bb* значение -1.

Шаг С4.2. Если *i* = *k*2, присвоить *i* = 0.

Шаг С4.3. *t* = *t* +*M*2[*i*]; При $r \geq 0$ присвоить *r* = *r* +*M*2[*i*].

Шаг С4.3. В цикле по k от 2 до $m(bb)$ определять значения: $v = M(k)[t \bmod b_k]$ при $r<0$, или $v = M(k)[r \bmod b_k]$ при $r > 0$. При первом же случае равенства v нулю перейти к шагу С.4.7. Если же $v = 1$ для всех k, то определено допустимое $t \bmod bb$.

Шаг С4.4. Если $r < 0$, присвоить $r = 0$ и $j = 0$, а иначе: $Xbb[j]=r$; $j = j + 1$; $r = 0$.

Шаг С4.5. Если $Xst < 0$ и $t \geq x_1$, присвоить $Xst = t$.

Шаг С4.6. Если $t > bb$, присвоить $Xbb[0]= j$ и перейти к шагу С.5, а иначе к С4.7.

Шаг С4.7. $i = i + 1$. Перейти к шагу С4.2.

Шаг С5. Определить итоговое значение Xst по формуле: $Xst = Xst + \left[x_0 / bb\right] * bb$. Завершить работу алгоритма.

Работу алгоритма А1 проиллюстрируем на примере числа $N = 7*113 =791$ и $bb = 48$.

Шаг С1. $bb = 2^4*3^2$. $n_0 = N \bmod bb=23$, $x_0 = 29$.

Шаг С2. $b = 16$. $N \bmod b=7$. Допустимые $X \bmod b$, т.е. те, что $\left((X \bmod b)^2 - N \bmod b\right) \bmod b$ является квадратичным остатком по модулю b – числа 0, 4, 8, 12. $x_2 =0$. Приращения для допустимых $X \bmod b$ согласно соотношений (2) – $M2[4] = \{4, 4, 4, 4\}$.

Шаг С3. $b_2 = 3$. $N \bmod b_2 = 2$. Допустимые $X \bmod b$, т.е. те, что $\left((X \bmod b)^2 - N \bmod b\right) \bmod b$ является квадратичным остатком по модулю b – одно число 0. $M(k)[3] = \{1, 0, 0\}$.

Шаг С4. Определить приращения для допустимых $X \bmod bb$ и начальное (стартовое) минимальное допустимое для основания модуля bb значение $Xst > x_0$ согласно следующих правил.

Изменение данных при выполнении шагов С4.1 – С4.7 представлены в табл. 8.

Таблица 8. Последовательность определения значений приращений для допустимых $X \bmod bb$ *и* Xst

i	0	1	2	3	4	5	6
t	0	4	8	12	16	20	24
r	0	4	8	12	4	8	12
$t \bmod 3$	0	1	2	0	1	2	0
$v = M(k)[t \bmod b_k]$	1	0	0	1	0	0	1
допустимое $t \bmod bb$	0			12			24
j	-1			0			1
$Xbb[j]$				12			12
Xst	-1(t<29)			-1(t<29)			-1(t<29)
i	7	8	9	10	11	12	
t	28	32	36	40	44	44	
r	4	8	12	4	8	12	
$t \bmod 3$	1	2	0	1	2	0	
$v = M(k)[t \bmod b_k]$	0	0	1	0	0	1	
допустимое $t \bmod bb$			36			48	
j			2			3	
$Xbb[j]$			12			12	
Xst			36>29			36	

Шаг С5. $Xst = Xst + \left[x_0 / bb\right] * bb = 36 + \left[29/48\right] * 48 = 36 + 0 = 36$. Работа алгоритма завершена.

Дальнейший поиск корня уравнения (1) начинается с X=36. Если дополнительных оснований модуля при поиске корня не предусмотрено, то при X=36, X=36+12=48, X=48+12=60 определяется $\sqrt{X^2 - N}$. Решение найдено, если $\sqrt{X^2 - N}$ является целым числом. В случае рассмотренного примера числа N значение корня X=60.

При анализе алгоритма А1 легко заметить, что и при $bb>10^{10}$ только операции сложения допускают значения $t > 2^{31}$ (ограничение для типа long). Все же другие операции проводятся с малыми числами, как правило не превышающие 2^{15}.

Выводы. Использование многократного просеивание пробных значений X относительно множества оснований модуля в процедурах разложения на множители чисел N методом Ферма позволяет существенно уменьшить число сложных операций извлечения корня $\sqrt{X^2 - N}$. Одна из ключевых функций здесь принадлежит первичному модулю – базовому основанию *bb*, определяющему наибольшее значение ускорения процесса поиска корня. А поскольку корень уравнения (1) ищется с помощью приращений к допустимым X, являющихся малыми числами, то важно иметь алгоритм поиска таких приращений. Предложенный в работе алгоритм *A1* обладает тем свойством, что и при $bb > 10^{10}$ только для одной операции сложения допускаются значения $t > 2^{31}$ (ограничение для типа long). Все же другие операции проводятся с малыми числами, как правило, не превышающими 2^{15}. Кроме того, при поиске допустимых X относительно базового основания модуля *bb* анализируется не более чем *1/4* чисел из диапазона $0 \div (bb-1)$.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

[1] D. Brown, “Breaking RSA may be as difficult as factoring”, *Journal of Cryptology*, vol. 29, iss. 1, pp. 220-241, January 2016. doi: 10.1007/s00145-014-9192-y.

[2] О.Н. Василенко. *Теоретико-числовые алгоритмы в криптографии*. Москва, Россия: МЦНМО, 2003.

[3] R.S. Lehman, “Factoring Large Integers”, *Mathematics of Computation*, vol. 28. iss.126, pp. 637-646, 1974. doi: 10.1090/S0025-5718-1974-0340163-2.

[4] Д. Кнут. *Искусство программирования. Том 2.* Москва, Россия: Вильямс, 2007.

[5] С.Д. Винничук, А.В. Жилин, и В.Н. Мисько, “Алгоритм Ферма факторизации чисел вида $N=pq$ методом прореживания”, *Электронное моделирование*, т. 36, № 2, с. 3-14, 2014.

[6] Е.В. Максименко, “Способ эффективного использования приращений при многократном прореживании пробных значений для метода факторизации Ферма”, *Information Technology and Security*, vol. 4, iss. 1, pp. 13-24, 2016.

[7] С.Д. Винничук, и Е.В. Максименко, “Многократное прореживание для ускорения метода факторизации Ферма при неравномерных шагах для неизвестной”, *Вісник НТУУ “КПІ”. Інформатика, управління та обчислювальна техніка*, № 64, с. 13-24, 2016.

[8] Е.В. Максименко, “Выбор эффективного базового основания модуля при многократном прореживании пробных значений в методе факторизации Ферма с неравномерным шагом”, *Інформатика та математичні методи в моделюванні*, том 6, № 3, с. 270-279, 2016.

Статья поступила в редакцию 24.09.2016.

REFERENCES

[1] D. Brown, “Breaking RSA may be as difficult as factoring”, *Journal of Cryptology*, vol. 29, iss. 1, pp. 220-241, January 2016. doi: 10.1007/s00145-014-9192-y.

[2] О.Н. Василенко. *Теоретико-числовые алгоритмы в криптографии*. Москва, Россия: МЦНМО, 2003.

[3] R.S. Lehman, “Factoring Large Integers”, *Mathematics of Computation*, vol. 28. iss.126, pp. 637-646, 1974. doi: 10.1090/S0025-5718-1974-0340163-2.

[4] Д. Кнут. *Искусство программирования. Том 2.* Москва, Россия: Viliams, 2007.

[5] S.D. Vynnychuk, A.V. Zhylyn, and V.N. Mysko, “Algoritm Ferma faktorizatsii chisel vida N=p*q metodom prorezhivaniia”, *Electronic Modeling*, vol. 36, iss. 2, pp. 3-14, 2014.

[6] Y.V. Maksymenko, “The way of effective use of incremental with multiple thinning of test values for Fermat’s factoring method”, *Information Technology and Security*, vol. 4, iss. 1, pp. 13-24, 2016.
[7] S.D. Vynnychuk, and Ye.V. Maksymenko, “Multiple thinning to accelerate Fermat's method of factorization with uneven steps for the unknown”, *Proceedings of the National technical university of Ukraine “KPI”: Informatyka, management and computing*, iss. 64, pp. 13-24, 2016.
[8] Y.V. Maksymenko, “Selection of effective basic basis of module with multiple thinning trial value in the factorization Fermat's method with irregular pitch”, *Informatics & Mathematical Methods in Simulation*, vol. 6, iss. 3, pp. 270-279, 2016.

СТЕПАН ВИННИЧУК,
ЄВГЕН МАКСИМЕНКО

ФОРМУВАННЯ НЕРІВНОМІРНИХ ПРИРОСТІВ ДЛЯ БАЗОВОЇ ОСНОВИ МОДУЛЯ В ЗАДАЧІ ФАКТОРИЗАЦІЇ МЕТОДОМ ФЕРМА

Одним із способів забезпечення заданого рівня захисту інформації є застосування алгоритмів асиметричного шифрування, криптографічна стійкість яких ґрунтуэться на трудності виконання завдання факторизації. Сучасні методи розкладання великих чисел на множники базуються на фундаментальних засадах класичного алгоритму факторизації Ферма. До числа можливих напрямів прискорення методу Ферма можна віднести зменшення кількості арифметично складних операцій обчислення квадратного кореня за рахунок реалізації процедури модульного ділення з використанням декількох основ модуля (метод решета). Ця модифікація методу дозволяє виключати з розгляду варіанти допустимих значень X, які не задовольняють умові $X^2 = N + Y^2$. В процесі проведених досліджень модифікованого методу факторизації Ферма було визначено, що при пошуку рішення рівняння $Y^2 = X^2 - N$ можливе використання не самих великих значень X, що отримано на етапі попереднього просіювання, а приростів до них, що є малими числами. Запропонований спосіб формування даних про нерівномірні прирости для допустимих значень X відносно базової основи модуля при факторизації чисел методом Ферма, що дозволяє істотно зменшити кількість значень, що перевіряються. Крім того, застосування цього алгоритму забезпечує не більше однієї операції додавання багаторозрядних чисел. Всі інші операції проводяться з малими числами, які як правило, не перевищують 2^{15}.

Ключові слова: факторизація, розкладання на множники, метод Ферма, проріджування, метод решета, прискорення.

STEPAN VYNNYCHUK,
YEVHEN MAKSYMENKO

FORMATION OF NON-UNIFORMITY INCREMENT FOR THE BASIC MODULE BASE IN THE PROBLEM OF FERMAT’S FACTORIZATION METHOD

One way to ensure the specified level of information security is the use of asymmetric encryption algorithms. The cryptographic durability of such algorithms is based on the difficulty of execution the factorization problem. Modern methods of decomposition of multiple-bit sequences at factors are based on the fundamental concepts of classical Fermat's factoring algorithm. Some of the possible directions of the acceleration Fermat’s method include reducing the number of arithmetically complex operations extracting the square root. The modular division procedure using several module bases (the sieve method) allows you to exclude from consideration the options of acceptable values of *X*, which do not satisfy the condition $X^2 = N + Y^2$. In the process of research the modification Fermat's factorization method it has been determined that it is possible to use not

the pre-sieved high values of X during the search for a solution of equation $Y^2 = X^2 - N$, but increments to them that are the small numbers. A method of forming a non-uniform incremental data for the valid values of X relation to the basic module base during the factorization of numbers with Fermat's method is offered. It can significantly reduce the number of scanned values. Furthermore, the use of this algorithm provide not more than one addition operation of multi-bit numbers. All other operations are carried out with small numbers usually not exceeding 2^{15}.

Keywords: factorization, Fermat's factorization method, thinning, sieve method, acceleration.

Степан Дмитриевич Винничук, доктор технических наук, старший научный сотрудник, исполняющий обязаности заведующего отделом моделирования энергетических процесов и систем, Институт проблем моделирования в энергетике им. Г.Е. Пухова Национальной академии наук Украины, Киев, Украина.

E-mail: vynnychuk@i.ua.

Евгений Васильевич Максименко, заместитель заведующего кафедрой кибербезопасности и применения автоматизировных информационных систем и технологий, Институт специальной связи и защиты информации Национального технического университета Украины "Киевский политехнический институт имени Игоря Сикорского", Киев, Украина.

E-mail: iszzi@i.ua.

Степан Дмитрович Винничук, доктор технічних наук, старший науковий співробітник, виконуючий обовязки завідувача відділом моделювання енергетичних процесів і систем, Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова Національної академії наук України, Київ, Україна.

Євген Васильович Максименко, заступник завідувача кафедри кібербезпеки та застосування автоматизованих інформаційних систем та технологій, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна.

Stepan Vynnychuk, doctor of technical sciences, senior researcher, acting head of the department of modeling of energy processes and systems, Pukhov institute for modeling in energy engineering of National academy of sciences of Ukraine, Kyiv, Ukraine.

Yevhen Maksymenko, deputy head of academic department cybersecurity and application of information systems and technologies, Institute of special communications and information protection National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute", Kyiv, Ukraine.