COMPUTATIONAL METHODS

DOI: 10.20535/2411-1031.2018.6.2.153492
УДК 004.056.53::003.26.09

СТЕПАН ВИННИЧУК,
ЄВГЕН МАКСИМЕНКО

МОДИФІКОВАНИЙ АЛГОРИТМ ФАКТОРИЗАЦІЇ ВЕЛИКИХ ЧИСЕЛ МЕТОДОМ ФЕРМА З ВИКОРИСТАННЯМ БАЗОВОЇ ОСНОВИ МОДУЛЯ

Метод факторизації Ферма вважається кращим при факторизації чисел виду $N = p \cdot q$ у випадках коли множники p і q близькі за значенням. Обчислювальна складність базового алгоритму методу факторизації Ферма визначається кількістю пробних значень X при вирішенні рівняння $Y^2 = X^2 - N$, а також складністю виконання арифметичних операцій з великими числами: зведення в квадрат, складання, обчислення квадратного кореня. Зниження обчислювальної складності зазначеного методу забезпечується за рахунок використання безлічі основ модулів в рівнянні $Y^2 modb = (X^2 - N) modb$, що в свою чергу дозволяє знехтувати складністю операцій обчислення квадратного кореня. Для зменшення числа пробних значень X розглядалася можливість використання базової (первинної) основи модуля bb. Оптимальний вибір базової основи модуля bb дозволяє зменшити число пробних X на величину, близьку за значенням до величини коефіцієнта прискорення $z(N, bb) = bb/bb^*$, де bb^* – число елементів множини коренів рівняння $(Ymodb)^2 modb = ((Xmodb)^2 modb - Nmodb) modb$. Показано, що в загальному випадку базова основа модуля bb є добутком ступенів простих чисел p, коефіцієнт прискорення для якого $z(N, bb)$ є функцією $Nmodp$ і показників ступенів p. Визначено ступінь впливу значення залишків $Nmodp$ (в разі $p=2$ використовуються залишки $Nmod8$) і показників ступенів простих p на величину коефіцієнта прискорення $z(N, bb)$). Запропоновано постановку задачі пошуку оптимального значення базової основи модуля bb при обмеженнях на обсяг пам'яті ЕОМ і спосіб її вирішення. Наведено оцінку ефективності запропонованого модифікованого алгоритму методу факторизації Ферма. Встановлено, що запропонований алгоритм для чисел 2^{1024} забезпечує зниження обчислювальної складності в порівнянні з базовим алгоритмом в середньому в 10^7 разів. Використання запропонованого методу дозволить проектувати більш ефективні, з точки зору швидкодії, апаратно-програмні засоби проведення криптоаналізу асиметричних криптографічних алгоритмів та, як наслідок, підвищити якість оцінки криптостійкості алгоритму RSA.

Ключові слова: факторизація, метод Ферма, проріджування, базова основа, обчислювальна складність.

Вступ. На даний час питання захисту інформації відносяться до одних з найбільш актуальних. Для їх вирішення використовується в тому числі зашифрування інформації. Серед способів шифрування широке застосування набув асиметричний криптоалгоритм (АКА) RSA. Його криптографічна стійкість зумовлена складністю розкладання на множники великих чисел $N = p \cdot q$, де p і q – прості. В роботах [1, 2] показано, що відомі приклади компрометації RSA алгоритму працюють тільки для окремих його реалізацій, та, як правило, у загальному випадку не є ефективнішими за вирішення задачі факторизації.

На даний час розроблено багато методів факторизації, серед яких до найбільш вживаних відносять методи решета числового поля (GNFS), квадратичного решета (QS), ρ-метод Поларда та метод Ферма [3-6]. При цьому вважається, що кожен з цих методів є найкращим (найбільш ефективним в смислі обчислювальної складності) для своєї області

використання. Так метод Ферма є найбільш ефективним при достатньо близьких значеннях простих множників p і q. ρ-метод Поларда найбільш ефективний при достатньо малому значенні одного з множників. За виключенням областей значень N, де найбільш ефективними є методи Ферма чи ρ-метод Поларда, метод квадратичного решета найбільш ефективний для $N < 10^{110}$ [4], а при $N > 10^{110}$ найбільш ефективним є метод решета числового поля. Тому розробка модифікацій таких методів, що дозволяють знизити обчислювальну складність довільного з таких методів, на етапі криптографічного аналізу АКА RSA забезпечить більшу надійність ключа шифрування. Слід також відмітити, що і QS і GNFS є різновидностями методу факторних баз, що є узагальненням методу факторизації Ферма. Отже, метод Ферма займає особливе положення серед відомих методів факторизації, а дослідження, пов'язані зі зниженням обчислювальної складності алгоритму методу Ферма, можуть бути корисними і для інших методів, що визначає їх актуальність.

Аналіз останніх досліджень і публікацій. Згідно класичного варіанту методу Ферма [7], для визначення значень p и q вирішується рівняння

$$X^2 - N = Y^2, \tag{1}$$

де X і Y – цілі додатні числа.

Невідома X представляється у вигляді

$$X_i = \left(\left\lfloor \sqrt{N} \right\rfloor + 1\right) + x_i = x_0 + i. \tag{2}$$

Рішення рівняння (1) отримують перебором значень i = 0, 1, 2, …, до тих пір, поки залишок $X^2 - N$ не виявиться повним квадратом цілого числа. Якщо рішення (1) отримано при $X^* = x_0 + x^*$, де $Y^* = \sqrt{\left(X^*\right)^2 - N}$, p і q визначаються згідно співвідношень:

$$\begin{cases} p = X - Y \\ q = X + Y \end{cases}. \tag{3}$$

Основним недоліком методу факторизації Ферма є необхідність багатократного виконання арифметично-складних операцій піднесення до квадрату, віднімання та обчислення квадратного кореня для великих чисел, що визначає його обчислювальну складність.

Ідея "перевірочного" обчислення кореня тільки для тих X, які відповідають необхідним умовам, була покладено в основу перших модифікацій методу самим П. Ферма [7]. Математик "розглядав величину і, виходячи з значень двох молодших розрядів, робив висновок про те, чи може ця величина бути повним квадратом (останні 2 розряди повного квадрата повинні бути 00, е1, е4, 25, о6 або е9, де е – парна, а о – непарна цифри)" [7]. Наприклад, 5^2=25; 10^2=100; 25^2=625; 354^2=125316; 1782^2=3175524 і т. д.

Іншими словами, Ферма помітив, що немає необхідності перевіряти на можливість отримання цілого кореня з $X^2 - N$ для всіх підряд значень X, а слід перевірити лише ті з них, молодші розряди яких у виразі $Y^2 = X^2 - N$ при заданому N відповідали б заздалегідь визначеним значенням. Такий результат легко отримати, якщо від рівності (1) перейти до рівності

$$Y^2 \bmod b = \left(X^2 - N\right) \bmod b, \tag{4}$$

що еквівалентно виконанню співвідношення

$$\left(Y \bmod b\right)^2 \bmod b = \left(\left(X \bmod b\right)^2 \bmod b - N \bmod b\right) \bmod b. \tag{5}$$

Слід зазначити, що якщо виконується співвідношення (1), то для довільного b має місце рівність (5). Зворотне невірно, тобто з виконання (5) не випливає виконання (1). Однак якщо не виконується співвідношення (5), то не буде виконуватися і (1). Тому для тих X, для яких не виконується (5), можна не визначати квадратний корінь, оскільки він не може бути точним квадратом цілого числа, за рахунок чого знижується обчислювальна складність методу Ферма.

Суттєво зменшити кількість обчислень квадратного кореня можна за рахунок використання в (5) багатьох основ модуля. В [7] відмічається, що використання одного з простих модулів в (5) практично вдвічі зменшується число *X*, для яких різниця $X^2 - N$ може бути повним квадратом. Такі *X* в подальшому будемо називати допустимими, а інше недопустимими. Ідея використання множини $MB = \{b_k\}_{k=1}^{m}$ основ модулів для перевірки виконання співвідношення (5), де кожен з них є простим числом, описана в роботі [7]. Вона була покладена в основу розробки ряду машин, що згідно алгоритму D реалізують попереднє просіювання в методі Ферма. Побудована в 1995 році машина [7] могла аналізувати $6.144 \cdot 10^9$ чисел в секунду.

Аналіз виконання співвідношень (5) при *b*=4 показує, що в залежності від *N* значення *X* можуть бути або парними, або непарними. Для допустимих парних *X* це дозволяє виключити з розгляду непарні *X* і навпаки, тобто в (2) можна використати крок рівний 2, де кожне з пробних *X* буде допустимим для *b*=4. При цьому не тільки вдвічі зменшиться кількість перевірок виконання (5) для наступних значень основ модулів *b*, але й не буде потреби в перевірці виконання (5) для *b*=4. Це суттєвий момент, оскільки при умові, що використання одного з простих модулів в (5) практично вдвічі зменшується число *X*, для яких різниця $X^2 - N$ може бути повним квадратом [7], число перевірок виконання співвідношення (5) не перевищить $x^*/2$.

Для основи *b*=3 при *Nmod3*=2 співвідношення (5) виконується тільки при *X* кратних 3. Тоді при використанні модулів *b*=4 та *b*=3 крок в (2) можна збільшити до значення 6. Отже кількість перевірок виконання (5) буде близькою до $x^*/6$, тобто можна говорити про прискорення процесу аналізу пробних *X* в 6 разів.

Легко перевірити, що використовуючи початкову основу модуля *b*=12 за умови *Nmod3*=2 при *Nmod12*=5 рівняння (5) матиме розв'язки *Xmod12*=3 та *Xmod12*=9, а при *Nmod12*=11 – значення *Xmod12*=0 та *Xmod12*=6. Тобто кожне наступне допустиме значення *X* відрізняється від попереднього на одну і ту ж величину 6, яка дорівнює відношенню

$$z(N,b) = b / b^*, \tag{6}$$

де b^* - число коренів рівняння (5) при відомих *N* та *b*.

При виборі різних значень початкової основи модуля в (5), яку надалі будемо називати базовою та позначати через *bb*, забезпечується зменшення числа перевірок в (5) на величину, близьку до *z(N,bb)*, яку в подальшому будемо називати прискоренням. Такий висновок підтверджений багатьма чисельними експериментами. Крім того, експериментально підтверджено, що за рахунок вибору множини основ модулів $MB = \{b_k\}_{k=1}^{m}$ число перевірок виконання співвідношень (5) може не перевищувати *x*/(4*z(N,bb))*. Це визначає особливу роль базової основи модуля та породжує задачу пошуку оптимального її значення для відомого *N*. Результати вирішення даної задачі представлені далі.

Метою статті є зменшення обчислювальної складності алгоритму факторизації великих чисел методом Ферма при використанні базової основи модуля.

Виклад основного матеріалу досліджень.

Структура допустимих *X* для базової основи модуля. Вже для *bb*=12 при *Nmod12*=1 (тобто *Nmod3*≠2) рівняння (5) матиме розв'язки *Xmod12*=1, *Xmod12*=5, *Xmod12*=7 та *Xmod12*=11. Тут кожне наступне допустиме значення *X* відрізняється від попереднього не на одну і ту ж величину, а коефіцієнт прискорення $z(N,b) = b/b^* = 12/4 = 3$. Тобто для обчислення наступних допустимих значень *X* не можна скористатися рівномірним кроком. Тому, на відміну від (2), нове допустиме для базової основи *bb* значення *X* в загальному випадку будемо визначати за правилом

$$X_{i+1} = X_i + \Delta x_i, \tag{7}$$

де Δx_i є різницею між наступним та попереднім значенням допустимих *X* для основи *bb*, а X_0 – найменше допустиме X, не менше за x_0. У співвідношенні (7) значення Δx_i, як правило

є різними величинами, тобто для отримання допустимих для *bb* пробних *X* використовуються нерівномірні кроки. Для довільного *bb* величини різниць періодично повторюються, що випливає з рівності

$$X_{i+1} \bmod bb = (X_i \bmod bb + \Delta x_i) \bmod bb, \quad (8)$$

а кількість елементів періодичної послідовності пропорційне кількості допустимих *X* (коренів рівняння (5) при *b*=*bb*) та може дорівнювати йому. Для наведеного вище прикладу *bb*=12 при *Nmod12*=1 , коли рівняння (5) матиме розв'язки *Xmod12*=1, *Xmod12*=5 *Xmod12*=7 та *Xmod12*=11, періодичну частина послідовності значень Δx_i можна визначити як 4, 2, 4, 2, або тільки 4, 2.

На простому прикладі *N*=145 та при *bb*=12 покажемо як формуються допустимі значення *X*. Згідно означення $x_0 = \left(\left\lfloor \sqrt{N} \right\rfloor + 1\right) = 13$ та $X_0 = 13$ (оскільки $x_0 \bmod 12 = 13 \bmod 12 = 1$, а 1 є коренем (5) при *bb*=12). При *Nmod12*=1 значення приростів – це періодична послідовність 4, 2, 4, 2, ... Тому допустимими *X* будуть: 13, 17, 19, 23, 25, 29, 31, …, де корінь рівняння (1) отримаємо при *X*=17.

В загальному випадку алгоритм методу багаторазового проріджування МР для базової основи модуля *bb* і множини додаткових основ модулів представлено у [8].

Вплив показників степенів простих чисел в структурі базової основи. На основі численних експериментів було виявлено, що коефіцієнт прискорення змінюється по різному як для різних випадків *N* при фіксованому *bb*, так і при фіксованому *N* і зміні показників ступеня простих множників в формованому *bb* [9, 10]. Однак питання впливу значення числа *N*, що факторизується, на вибір показників ступенів простих множників *bb* для досягнення максимального прискорення не досліджувалося.

Загальне уявлення про зміну коефіцієнта прискорення при змінах *bb* і фіксованому значенні *N* можна отримати на основі даних, представлених в табл. 1 і 2, де представлена інформація для всіх *Nmodbb* < 60, взаємно простих з 2, 3 і 5.

Таблиця 1 – Значення коефіцієнтів прискорення *z(bb, Nmodbb)* для різних *bb* як добутків числа 60 на степені 2, 3 і 5 для *Nmodbb* < 60, взаємно простих з 2, 3 і 5

Nmodbb	Варіанти значень *bb*											
	60	240	180	300	720	900	1200	960	540	1500	8640	24000
	*1	*4	*3	*5	*12	*15	*20	*16	*9	*25	*144	*400
1	5	10	15	10,71	30	32,14	21,43	20	22,5	12,1	90	48,39
7	7,5	15	22,5	7,5	45	22,50	15,00	15	33,75	7,5	67,5	15,00
11	10	20	10	21,43	20	21,43	42,86	20	10	24,19	20	48,39
13	7,5	15	22,5	7,5	45	22,5	15	30	33,75	7,5	135	30
17	15	30	15	15	30	15	30	60	15	15	60	60
19	5	10	15	10,71	30	32,14	21,43	10	22,5	12,1	45	24,19
23	15	30	15	15	30	15	30	30	15	15	30	30
29	10	20	10	21,43	20	21,43	42,86	40	10	24,19	40	96,77
31	5	10	15	10,71	30	32,14	21,43	10	22,5	12,1	45	24,19
37	7,5	15	22,5	7,5	45	22,5	15	30	33,75	7,5	135	30
41	10	20	10	21,43	20	21,43	42,86	40	10	24,19	40	96,77
43	7,5	15	22,5	7,5	45	22,5	15	15	33,75	7,5	67,5	15
47	15	30	15	15	30	15	30	30	15	15	30	30
49	5	10	15	10,71	30	32,14	21,43	20	22,5	12,1	90	48,39
53	15	30	15	15	30	15	30	60	15	15	60	60
59	10	20	10	21,43	20	21,43	42,86	20	10	24,19	20	48,39

Таблиця 2 – Зміни коефіцієнтів прискорення *z(bb, Nmodbb)* при змінах *bb* в порівнянні з *bb*=60 для *Nmodbb* < 60, взаємно простих з 2, 3 и 5

Nmodbb	Варіанти значень *bb*											
	60	240	180	300	720	900	1200	960	540	1500	8640	24000
	*1	*4	*3	*5	*12	*15	*20	*16	*9	*25	*144	*400
1	1	2	3	2,14	6	6,43	4,29	4	4,5	2,42	18	9,68
7	1	2	3	1	6	3	2	2	4,5	1	9	2
11	1	2	1	2,14	2	2,14	4,29	2	1	2,42	2	4,84
13	1	2	3	1	6	3	2	4	4,5	1	18	4
17	1	2	1	1	2	1	2	4	1	1	4	4
19	1	2	3	2,14	6	6,43	4,29	2	4,5	2,42	9	4,84
23	1	2	1	1	2	1	2	2	1	1	2	2
29	1	2	1	2,14	2	2,14	4,29	4	1	2,42	4	9,68
31	1	2	3	2,14	6	6,43	4,29	2	4,5	2,42	9	4,84
37	1	2	3	1	6	3	2	4	4,5	1	18	4
41	1	2	1	2,14	2	2,14	4,29	4	1	2,42	4	9,68
43	1	2	3	1	6	3	2	2	4,5	1	9	2
47	1	2	1	1	2	1	2	2	1	1	2	2
49	1	2	3	2,14	6	6,43	4,29	4	4,5	2,42	18	9,68
53	1	2	1	1	2	1	2	4	1	1	4	4
59	1	2	1	2,14	2	2,14	4,29	2	1	2,42	2	4,84

На основі аналізу даних табл. 1, 2 можна зробити два основних висновки:

- коефіцієнти прискорення приймають одинакові значення для множини величин *Nmodbb*, які є різними для різних основ модулів;
- при зміні *bb* коефіцієнт прискорення змінюється в залежності від значення додаткового множника в ньому.

Так при збільшенні *bb* в 3 рази (*bb*=180) коефіцієнт прискорення збільшується або в три рази, або залишається незмінним. А при збільшенні *bb* в 9 разів (*bb*=540) коефіцієнт прискорення збільшується або в 4.5 рази, або залишається незмінним. Причому збільшення має місце для тих же *Nmod3* що і при *bb*=180. У разі збільшення *bb* в 5 або 25 разів збільшення коефіцієнта прискорення також має місце для тих же *Nmod5*. Збільшення ж *bb* в $2^{c1}*3^{c2}*5^{c3}$ раз призводить до збільшення коефіцієнта прискорення, рівного добутку прискорень, пов'язаних зі збільшенням в *bb* показника ступеня числа 2 на c_1, показника ступеня числа 3 на c_2 і показника ступеня числа 5 на c_3.

Отже, можна припустити, що для множників *bb*, рівних p^t, можна визначити множину значень $Nmodp^t$, для яких при зміні показника ступеня t значення коефіцієнтів прискорення не змінюються, а також тих, для яких вони змінюються. Перевірка такого припущення проводилася з використанням чисельних експериментів для множників *bb* - простих *p* від *p*=2 до *p*=31. Нижче наведено результати таких досліджень.

а) Множник *bb* *p*=2. На основі чисельних експериментів було встановлено, що для 2^t доцільно використовувати значення показника ступеня $t \geq 2$, оскільки при *t*=1, *Nmod2* прискорення дорівнює 1. У табл. 3 представлено значення коефіцієнтів прискорення для всіх взаємно простих з *bb* непарних значень $Nmod2^t$ при *t*=3÷7.

Таблиця 3 – Коефіцієнти прискорення *z(bb, Nmodbb)* для непарних $Nmod2^t$ при *t*=3÷7

bb	Прискорення *z = z(bb, Nmodbb)* для всіх можливих значень *Nmodbb*																
8	*Nmodbb*	1	3	5	7	-	-	-	-	-	-	-	-	-	-	-	-
	z	2	4	2	4	-	-	-	-	-	-	-	-	-	-	-	-
16	*Nmodbb*	1	3	5	7	9	11	13	15	-	-	-	-	-	-	-	-

Продовження таблиці 3

	z	4	4	4	4	4	4	4	4	-	-	-	-	-	-	-	-
32	*Nmodbb*	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
	z	4	4	8	4	4	4	8	4	4	4	8	4	4	4	8	4
64	*Nmodbb*	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
	z	8	4	8	4	8	4	8	4	8	4	8	4	8	4	8	4
	Nmodbb	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63
	z	8	4	8	4	8	4	8	4	8	4	8	4	8	4	8	4
128	*Nmodbb*	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
	z	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4
	Nmodbb	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63
	z	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4
	Nmodbb	65	67	69	71	9	75	77	79	81	83	85	87	89	91	93	95
	z	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4
	Nmodbb	97	99	101	103	105	107	109	111	113	115	117	119	121	123	125	127
	z	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4	10.7	4	8	4

На основі даних табл. 3 можна зробити висновок, що характер зміни значень коефіцієнтів прискорення для $bb=2^t$ при $t > 3$ визначається величиною залишку *Nmod8*, що було підтверджено при додаткових чисельних експериментів з $bb=2^t$ при $t \leq 14$ Для таких значень *bb* при $t=1\div14$ в табл. 4 наведені значення коефіцієнтів прискорення в залежності від *Nmod8*.

Таблиця 4 – Коефіцієнти прискорення *z(bb, Nmodbb)* для непарних *Nmod8* при $bb=2^t$ $t=1\div14$

Nmod8	1	3	5	7	*Nmod8*	1	3	5	7
t=1	1	-	-	-	*t*=8	16	4	8	4
t=2	2	2	-	-	*t*=9	18.2857	4	8	4
t=3	2	4	2	4	*t*=10	21.3333	4	8	4
t=4	4	4	4	4	*t*=11	22.2609	4	8	4
t=5	4	4	8	4	*t*=12	23.2727	4	8	4
t=6	8	4	8	4	*t*=13	23.5402	4	8	4
t=7	10.6667	4	8	4	*t*=14	23.8140	4	8	4

Згідно з даними табл. 4 величина коефіцієнта прискорення для $bb=2^t$ при $t>2$ визначається показником степеня *t* та значенням *Nmod8*. Це підтверджується даними табл. 3 при $t=3\div7$, де, наприклад, при $t=7$ і *Nmod8*=1 однаковими (рівними 32/3) будуть коефіцієнти прискорень для значень $Nmod2^7$, рівних 1, 9, 17, 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105 і 113, тобто для тих, що $(Nmod2^7)mod8=1$. Тому дані табл. 4 дозволяють оцінити можливості побудови ефективної первинної основи *bb* для випадків, коли серед простих множників *bb* є число 2. Так при *Nmod8*=3 і *Nmod8*=7 при $bb=2^t$ і $t\geq3$ коефіцієнт прискорення завжди дорівнює 4 і в *bb* не має сенсу використовувати ступінь 2 з показником вище 3. Якщо *Nmod8*=5, то в *bb* оптимальним буде використовувати ступінь 2 з показником 5. Але якщо *Nmod8*=1, то в *bb* можна використовувати ступінь 2 з показником 8 і більше.

б) Множник *bb* $p=3$. У разі, коли до складу *bb* входить множник 3, було встановлено, що при *Nmod3*=2 співпадають значення коефіцієнтів прискорення для $bb=3^t$ при $t\geq1$, що підтверджено чисельними експериментами з $bb = 3^t$ при $t=1\div8$. Для таких значень *bb* в табл. 5 наведені значення коефіцієнтів прискорення в залежності від *Nmod3* при $t=1\div8$.

Згідно даних табл. 5 можна оцінити можливості побудови ефективної первинної основи *bb* для випадків, коли серед простих множників *bb* є число 3. Так при *Nmod3*=2 при $bb=3^t$ і $t>0$ коефіцієнт прискорення завжди дорівнює 3. Тому в *bb* не має сенсу використовувати

ступінь 3 з показником вище 1. Якщо ж *Nmod3*=1, то в *bb* можна використовувати ступінь 3 з показником 4 і більше.

Таблиця 5 – $z(3^t, N \bmod 3)$ для взаємно простих з 3 *Nmod3* при *bb*=3^t і *t*=1÷8

Nmod3	1	2	*Nmod3*	1	2
t=1	1.5	3	*t*=5	11.0455	3
t=2	4.5	3	*t*=6	11.7581	3
t=3	6.75	3	*t*=7	11.8859	3
t=4	10.125	3	*t*=8	11.9726	3

в) Множник *bb* *p*=5. Якщо до складу *bb* входить множник 5, було встановлено, що значення коефіцієнтів прискорення для *bb*=5^t при *t*≥1 співпадають для всіх *Nmod5*=2 і *Nmod5*=3, що було підтверджено чисельними експериментами з *bb*=5^t при *t*=1÷6. Але при *t*>1 та *Nmod5*=1 і *Nmod5*=4 значення коефіцієнта прискорення зростає. Таке значення при кожному t є однаковим при *Nmod5*=1 і *Nmod5*=4. Для таких значень *bb* в табл. 6 наведені значення коефіцієнтів прискорення в залежності від *Nmod5*.

Таблиця 6 – $z(5^t, N \bmod 5)$ для взаємно простих з 5 *Nmod5* при *bb*=5^t і *t*=1÷6

Nmod5	1	2	3	4	*Nmod5*	1	2	3	4
t=1	1.6667	2.5	2.5	1.6667	*t*=4	4.2517	2.5	2.5	4.2517
t=2	3.5714	2.5	2.5	3.5714	*t*=5	4.2750	2.5	2.5	4.2750
t=3	4.0323	2.5	2.5	4.0323	*t*=6	4.2843	2.5	2.5	4.2843

На основі даних табл. 6 можна більш точно оцінити можливості побудови ефективної первинної основи *bb* для випадків, коли серед простих множників *bb* є число 5. Так при *Nmod5*=2 або *Nmod5*=3 при *bb*=5^t и *t*>0 $Z(5^t, N \bmod 5)$=2.5. Тому доцільно використати показник степеня *t*=1. Якщо *Nmod5*=1 або *Nmod5*=4, то в *bb* можна використовувати ступінь 5 з показником 2 і вище.

г) Множник *bb* *p*=7. Якщо до складу *bb* входить множник 7, було встановлено, що значення коефіцієнтів прискорення для *bb*=7^t при *t*≥1 співпадають для всіх *Nmod7*=3, *Nmod7*=5 і *Nmod7*=6, що було підтверджено чисельними експериментами з *bb*=7^t при *t*=1÷3. Але при *t* > 1 та *Nmod7*=1 *Nmod7*=2 і *Nmod7*=4 значення коефіцієнта прискорення зростає та приймає одинакове значення. Для таких значень *bb* в табл. 7 наведені значення коефіцієнтів прискорення в залежності від *Nmod7* при *t*=1÷4.

На основі даних табл. 7 можна більш точно оцінити можливості побудови ефективної первинної основи *bb* для випадків, коли серед простих множників *bb* є число 7. Так при *Nmod7*=3, *Nmod7*=5 або *Nmod7*=6 при *bb*=7^t і *t*>0 коефіцієнт прискорення завжди дорівнює 2.3333. Тому в *bb* не має сенсу використовувати ступінь 7 з показником вище 1. Якщо *Nmod7*=1, *Nmod7*=2 або *Nmod7*=4, то в *bb* можна використовувати ступінь 7 з показником 2 і вище.

Таблиця 7 – Коефіцієнти прискорення для *Nmod7*, взаємно простих з 7

Nmod7	1	2	3	4	5	6
t=1	1.75	1.75	2.3333	1.75	2.3333	2.3333
t=2	3.0625	3.0625	2.3333	3.0625	2.3333	2.3333
t=3	3.2358	3.2890	2.3333	3.2890	2.3333	2.3333

д) Множники *bb* *p*>7 і *p*≤23. Для простих *p* – множників *bb*, рівних 11, 13, 17, 19, 23 було встановлено, що характер зміни значень коефіцієнтів прискорення для *bb*=p^t при *t*≥1 визначається значенням *Nmodp*, що було підтверджено чисельними експериментами з *bb*=p^t

при t=1÷4. Для таких значень *bb* в табл. 8 наведені значення коефіцієнтів прискорення в залежності від *Nmodp* при t=1, 2.

Таблиця 8 – Значення i=*Nmodp*, для яких $z(p,1)=z(p^2,i)$ і $z(p,i)< z(p^2,i)$

p	Значення i=*Nmodp*, для яких $z(p,i) = z(p^2,i)$	t	$z(p^t,i)$	Значення i=Nmodp, для яких $z(p,i) < z(p^2,i)$	t	$z(p^t,i)$
11	2, 6, 7, 8, 10	1	2.2000	1, 3, 4, 5, 9	1	1.8333
		2	2.2000		2	2.6300
13	2, 5, 6, 7, 8, 11	1	2.1667	1, 3, 4, 9, 10, 12	1	1.8571
		2	2.1667		2	2.5224
17	3, 5, 6, 7, 10, 11, 12, 14	1	2.1250	1, 2, 4, 8, 9, 13, 15, 16	1	1.9000
		2	2.1250		2	2.3884
19	2, 3. 8, 10, 12, 13, 14, 15, 18	1	2.1111	1, 4, 5, 6, 7, 9, 11, 16, 17	1	1.8889
		2	2.1111		2	2.3442
23	5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 22	1	2.0909	1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18	1	1.9167
		2	2.0909		2	2.2902

Як і у випадку простих p, рівних 2, 3, 5 і 7, дані табл. 8 доцільно використати для побудови ефективних *bb*. На прикладах чисел N, представлених в табл. 9 покажемо, що врахування специфіки чисел N, а саме залишків від ділення N на 8 і на прості р, рівні 3 і 5, дозволяє побудувати нове bb^*, при якому множина $D(bb^*, N)$ міститиме число допустимих X, що не перевищує його значення для *bb*=277200, яке дорівнює 2880 комірок пам'яті типу *int*. Первинна основа *bb*=277200 є добутком ступенів простих чисел 2, 3, 5, 7 і 11: $bb=2^4 * 3^2 * 5^2 * 7 * 11$. Згідно даних, наведених в табл. 4 - 8, сформуємо нові, більш ефективні *bb*. Для цього для кожного з j=1÷6 чисел N_j визначимо залишки від ділення на 8 і на прості p від 3 до 23. Дані представлено в табл. 9.

Таблиця 9 – Значення N_j *modp* для $p=2^3=8$ і простих p=3÷23

j	N	p								
		8	3	5	7	11	13	17	19	23
1	2 190 107 742 436 404 740 487 152 427 983	7	2	3	2	5	1	5	15	22
2	115 103 357 258 699 743 681 239 319 283	3	2	3	1	5	1	13	11	17
3	24 197 500 008 691 435 623 032 029 847	7	2	2	2	4	3	13	16	12
4	7 193 959 711 947 061 718 333 522 687	7	2	2	1	9	2	1	10	2
5	3 024 687 551 113 421 119 054 532 273	1	2	3	2	1	12	4	13	21
6	1 798 489 957 219 681 011 882 800 933	5	2	3	1	3	1	13	13	1

З урахуванням наведених вище рекомендацій щодо вибору показників ступенів простих р – множників *bb* в табл. 10 наведено уточнені значення bb^*, які враховують специфіку числа N, що факторизується.

Таблиця 10 – Уточнені первинні основи bb^*, сформовані для чисел N з урахуванням даних табл. 4-6

j	Показники ступенів для простих чисел, що складають нові первинні основи									bb^*	z	Обсяг пам'яті bb^*/z
	2	3	5	7	11	13	17	19	23			
1	3	1	1	2	1	1	-	-	-	840840	312.812	2688
2	3	1	1	2	1	1	-	-	-	840840	312.812	2688
3	3	1	1	2	1	1	-	-	-	840840	312.812	2688
4	3	1	1	2	1	1	-	-	-	840840	364.948	2304
5	10	1	1	2	-	-	-	-	-	752640	490	1536
6	5	1	1	2	-	-	-	1	-	446880	387.917	1152

Як випливає з даних табл. 10, для уточнених *bb** за рахунок врахування специфіки числа *N* знижено об'єм необхідної пам'яті ЕОМ та одночасно збільшено значення коефіцієнта прискорення в 3.25 – 5.091, що приблизно у таке ж число разів зменшує час розкладання чисел на множники. Тому доцільно розглянути задачу пошуку оптимального *bb*, що враховує специфіку довільних чисел *N*, що факторизуються, та додаткові можливості економії пам'яті ЕОМ.

Визначення оптимальної первинної основи *bb* з урахуванням специфіки числа, що факторизується. При постановці завдання пошуку оптимальної первинної основи модуля *bb* будемо використовувати інформацію про структуру *bb*, про властивості коефіцієнтів прискорення і про кількість елементів множини *D(bb, Nmodbb)*:

$$bb = \prod_{i=1}^{h} p_i^{k_i}; \tag{9}$$

$$z(bb, N) = \prod_{i=1}^{h} z\left(p_i^{k_i}, N\right); \tag{10}$$

кількість елементів масиву D(bb, Nmodbb) дорівнює:

$$bb / z(bb, N) = \prod_{i=1}^{h} p_i^{k_i} / z(p_i^{k_i}, N). \tag{11}$$

Згідно (9) - (11) для визначення *bb* досить визначити показники ступенів простих чисел – множників *bb*, де необхідно враховувати співвідношення між значенням простого числа і зростанням прискорення при збільшенні показника його ступеня. Для простих *p* від 2 до 23 відповідні їм значення коефіцієнтів прискорення визначаються за даними табл. 4 - 8. При постановці завдання пошуку оптимального *bb* з урахуванням *N*, будемо розглядати можливі типи варіантів значень показників ступенів залежно від *p*, серед яких буде і варіант коли множник *p* не використовується в *bb* та $z(p^0, N)=1$.

Для $p=2$ можливі три типи варіантів:

1) при $Nmod8=3$ або $Nmod8=7$ показник ступеня *t* завжди дорівнює 3;

2) при $Nmod8=5$ показник ступеня *t* завжди дорівнює 5;

3) при $Nmod8=1$ значення показника ступеня *t* необхідно визначати.

У разі простих $p > 2$ також необхідно розглядати три типи варіантів:

1) *Nmodp* приймає значення таке, що $z(p, Nmodp)=z(p^2, Nmodp)$ і $t=1$;

2) $t=0$ і $z(p^0, N)=1$ (множник p не використовується в *bb*);

3) *Nmodp* приймає значення таке, що $z(p, Nmodp < z(p^2, Nmodp)$ і $t \geq 1$.

Отже, при виборі показника ступеня простого p - множника *bb* тільки для третього з варіантів показник ступеня не визначений. Для оцінки можливого діапазону показників ступенів у варіанті 3 використаємо функцію відносного приросту коефіцієнта прискорення, приведеного до одиниці пам'яті:

$$s(p,t) = (z(p^{t+1},1) / z(p^t,1) - 1) / p, \tag{12}$$

що дозволяє дати наближену оцінку ефективності первинної основи модуля, пов'язану з домножуванням *bb* на простий множник *p*. Значення функції $s(p,t)$ для простих $p \geq 2$ і $p \leq 31$ для ряду варіантів показників ступенів наведені в табл. 11.

Таблиця 11 – Значення функції $s(p, t)$ для простих $p \geq 2$ і $p \leq 31$

p	*t*	$z(p^t,1)$	$z(p^{t+1},1)$	$s(p,t)$	*p*	*t*	$z(p^t,1)$	$z(p^{t+1},1)$	$s(p,t)$
2	5	4	8	0.5	7	0	1	1.75	0.10714
	6	8	10.66667	0.16667		1	1.75	3.0625	0.10714
	7	10.66667	16	0.25		2	3.0625	3.2358	0.00808
	8	16	18.2857	0.07143		3	3.2358	3.2890	0.002349
	9	18.2857	21.3333	0.08333	11	0	1	1.8333	0.07576
	10	21.3333	22.2609	0.02174		1	1.8333	2.6300	0.03953
	11	22.2609	23.2727	0.02273	13	0	1	1.8571	0.06593

Продовження таблиці 11

	12	23.2727	23.5402	0.00575		1	1.8571	2.5224	0.02755
3	0	1	1.5	0.16667	17	0	1	1.8889	0.05229
	1	1.5	4.5	0.66667		1	1.8889	2.3884	0.01556
	2	4.5	6.75	0.16667	19	0	1	1.9000	0.04737
	3	6.75	10.125	0.16667		1	1.9000	2.3442	0.01230
	4	10.125	11.0455	0.03031	23	0	1	1.9167	0.03986
	5	11.0455	11.7581	0.02151		1	1.9167	2.2902	0.00847
5	0	1	1.6667	0.13333	29	0	1	1.9333	0.03218
	1	1.6667	3.5714	0.22857		1	1.9333	2.2190	0.00510
	2	3.5714	4.0323	0.02581	31	0	1	1.9375	0.03024
	3	4.0323	4.2517	0.01088		1	1.9375	2.2041	0.00444

Значення *s(p,t)* при їх сортуванні в порядку спадання дають можливість оцінити наскільки ефективним буде додавання множника *p* в *bb*. Чим більше *s(p,t)*, тим ефективність вище. Якщо ж *s(p,t)* близьке до нуля, то при збільшенні *bb* коефіцієнт прискорення зростає незначно, але істотно зростає обсяг пам'яті ЕОМ, що використовується для зберігання приростів для допустимих *X*. Для пошуку оптимального *bb* з урахуванням специфіки *N* і способів скорочення пам'яті ЕОМ використовуються: співвідношення (9) - (11) та обмеження на обсяг пам'яті ЕОМ. Пошук максимального коефіцієнта прискорення за рахунок перебору допустимих варіантів показників ступенів простих р – множників *N*.

При чисельних розрахунках з визначення оптимального *bb* з урахуванням допустимого обсягу пам'яті, необхідного для зберігання приростів допустимих *X* було прийнято, що первинна основа модуля *bb* є добутком ступенів простих чисел *p*, рівних 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31 де множина варіантів показників ступенів простих чисел р - множників *bb*, вибиралася на підставі даних табл. 5 - 8 і 11 з умови $s(p, t) > 0.03$ (*t* - показник ступеня для *p*). При цьому враховувалися такі варіанти впливу на коефіцієнт прискорення:

– для $p=2$ по *Nmod8* вибирався один з можливих типів варіантів, де в разі $Nmod8 = 1$ розглядалися показники степеня $t=3\div12$;

– для $p\geq3$ і $p\leq31$ вибиралися два типи варіантів: тип 2, а також один з типів 1 або 3 в залежності від значення Nmodp.

Крім того, при визначенні необхідного обсягу пам’яті ЕОМ враховувалося те, що *bb* завжди ділиться на 4, тобто циклічна послідовність приростів для *bb* повторюється як мінімум двічі.

У чисельних експериментах для обсягів пам'яті (величина Q_{max}) 10^2, 10^3, 10^4, 10^5, 10^6, 10^7 для кожного з варіантів впливу на коефіцієнт прискорення визначалося максимальне значення коефіцієнта прискорення. Оскільки число таких варіантів виявляється досить великим (рівним $3 * 2^8=768$), то в табл. 12 далі наведені дані тільки про z_{min}, z_{max}, та середнє z_{cp}, яке дорівнює середньому значенню для всіх варіантів, де для досягнення їх рівнозначності випадків при $p=2$ і типі варіанту 1, присвоювався коефіцієнт 2, а для типів варіантів 2 і 3 - коефіцієнт 1. Тоді зважена сума всіх отриманих максимальних коефіцієнтів прискорення ділилася на 1024. Отримані значення z_{min}, z_{max} і z_{cp} представлені в табл. 12 та у вигляді діаграми на рис. 1.

Таблиця 12 – Значення: z_{min}, z_{max}, $_{cp}$ і необхідного обсягу пам’яті ЕОМ $z_q(z)$ для різних граничних значень Q_{max}

Q_{max}	z_{min}	$z_q(z_{min})$	$bb(z_{min})$	z_{max}	$z_q(z_{max})$	$bb(z_{max})$	z_{cp}
10^2	89.610	77	13800	472.5	96	90720	214.520
10^3	213.571	924	394680	1386	960	2661120	579.731
10^4	405.786	9240	7498920	3003	8640	51891840	1365.952
10^5	822.833	92736	152612460	9572.063	92160	1764322560	2976.528

Продовження таблиці 12

10^6	1563.382	927360	2899636740	20207.687	829440	33522128640	6007.030
10^7	2407.279	8814960	42440137740	42252.438	9123840	771008958720	11088.624

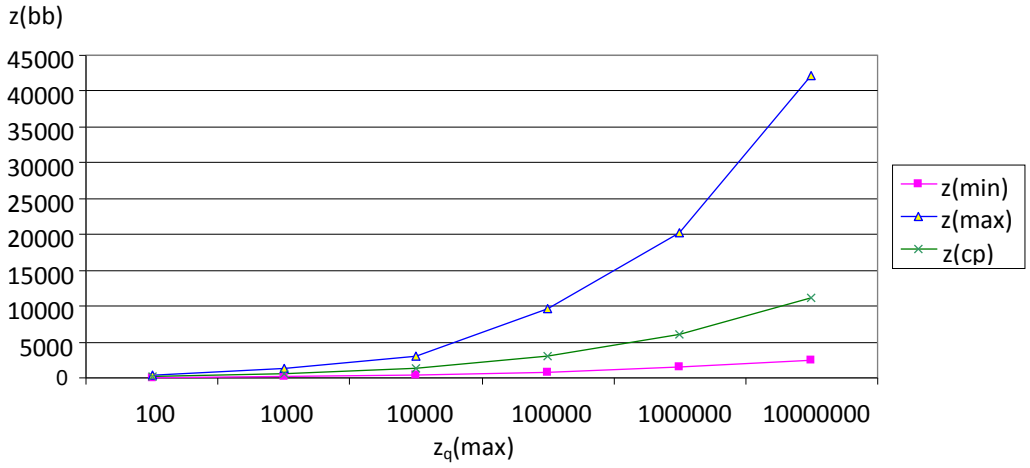

Рисунок 1 – Значення коефіцієнтів прискорення *z(bb)*
при обмеженнях на об'єм $Q_{(max)}$ доступної пам'яті

В модифікованому алгоритмі методу методу Ферма при використанні співвідношення (7) з приростами, що є числами типу *long*, отриманими для сформованного оптимального значення первинної основи *bb*, арифметичні операції з великими числами виконуються лише у двох випадках:

– залишки *Xmodbk (k=1÷m)* для поточного X є такими, що X буде допустимим для всіх основ модулів з множини $MB = \{b_k\}_{k=1}^{m}$, а також для *b=bb*. Тоді обчислюється значення $X^2 - N$ та корінь з нього;

– при представленні поточного X у вигляді суми $X = X^* + \sum_i \Delta x_i = X^* + \Delta X$, де X^* – деяке проміжне фіксоване значення, рівне спочатку X_0, величина ΔX близька до граничого для даних типу *long*. Тоді виконуються операції $X^* = X^* + \Delta X$, $\Delta X = 0$ та обчислюються поточні залишки *X*modbk (k=1÷m)*, де число модулів m вибирається таким, що при оцінках обчислювальної складності затратами на обчислення квадратного кореня можна знехтувати.

Тоді для чисел N порядку 2^{1024} обчислювальна складність модифікованого алгоритму по відношенню до базового алгоритму Фермі знижується не менше ніж в $2.4 \cdot 10^6 \div 4.2 \cdot 10^7$ разів в залежності від факторизованого N, оскільки число пробних X в залежності від N зменшується в $2.4 \cdot 10^3 \div 4.2 \cdot 10^4$ разів, а при кожному пробному X замість складних операцій з багаторозрядними числами (піднесення до квадрату, віднімання та обчислення квадратного кореня) виконуються операції з числами типу *long*:

– збільшення ΔX на величину приросту для отримання наступного допустимого для *bb* значення;

– визначення остач *ΔXmodbk* починаючи з *k*=1 до першого з випадків, коли *ΔXmodbk* не буде коренем рівняння (5), де число таких перевірок при кожному з пробних X в середньому не перевищує 2.

Висновки. При використанні співвідношення (5) для оцінки можливості отримання рішення рівняння (1) для множини основ модулів їх число можна вибрати таким, що обчислювальною складністю операції визначення квадратного кореня з великого числа можна знехтувати при загальній оцінці алгоритму Ферма. При цьому обчислювальна складність алгоритму визначатиметься числом перевірок виконання співвідношень (5), яке за

рахунок вибору множини основ модулів $MB = \{b_k\}_{k=1}^{m}$ може не перевищувати *x*/(4*z(N,bb)*). Це визначає особливу роль первинної основи модуля *bb* та актуальність завдання пошуку такого *bb*, для якого максимальним буде прискорення *z(N,bb)*).

В загальному випадку первинна основа модуля bb є добутком степенів простих чисел *p*, коефіцієнт прискорення якої *z(N,bb)*) є функцією *Nmodp* (для *p*=2 - *Nmod8*) та показників степенів *p*. На основі визначення степені їх впливу запропоновано спосіб формування первинної основи модуля *bb* з максимальним коефіцієнтом прискорення для числа *N*, що факторизується, на основі наступних положень:

– величина коефіцієнта прискорення *z(bb, Nmodbb)* визначається значеннями показників ступенів *t* простих чисел *p* – множників *bb* для кожного з простих p незалежно;

– для $p > 2$ для половини можливих значень *Nmodp* значення *z(p, Nmodp)* залишається незмінним незалежно від показника ступеня $t > 0$, в зв'язку з чим для таких *Nmodp* показник ступеня слід вибрати рівним 1. Для інших варіантів значень Nmodp зі збільшенням показника ступеня *t* зростає як $z(p^t, Nmodp)$, так і обсяг пам'яті для зберігання приростів допустимих *X*;

– для *p*=2 величина ефективного коефіцієнта прискорення визначається відповідно до значення *Nmodp3=Nmod8*, де при *Nmod8*=3 і *Nmod8*=7 оптимальне значення коефіцієнта прискорення рівне 4 досягається для p^3. У разі *Nmod8*=5 – для p^5, а при *Nmod8*=1 зростає і коефіцієнт прискорення, і обсяг необхідної пам'яті для зберігання приростів допустимих *X*;

– якщо *p*=2 і *Nmod8*=1 або $p > 2$ і *Nmodp* таке, що показник ступеня простого *p* – множника *N* впливає на величину коефіцієнта прискорення, то множина варіантів значень показників ступенів простих p, що можуть бути використані в *bb*, визначається з умови $s(p, t) > s_{min}$, де *s(p, t)* – відносна зміна коефіцієнта прискорення для простого *p* при збільшенні показника ступеня t на одиницю, приведене до одиниці пам'яті.

Максимальне значення коефіцієнта прискорення *z(bb, Nmodbb)* при врахуванні обмеження на максимально допустиму величину наявного обсягу пам'яті для зберігання приростів допустимих *X* визначається методом повного перебору показників ступенів множини простих *p* – множників *bb*, де для кожного варіанту значень використовуваних в *bb* простих *p* і показників їх ступенів визначається *z(bb, Nmodbb)* і необхідний обсяг пам'яті з урахуванням циклічності послідовності приростів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Brown, Breaking "RSA May Be As Difficult As Factoring". Cryptology ePrint Archive. 2005. [Online]. Available: https://eprint.iacr.org/2005/380. Accessed on: Apr. 24, 2018.

[2] D. Aggarwal, U. Maurer, "Breaking RSA Generically is Equivalent to Factoring". Advances in Cryptology - EUROCRYPT 2009. Germany. [Online]. Available: https://eprint.iacr.org/2008/260. Accessed on: May 05, 2018.

[3] C. Pomerance, H. Lenstra, R. Tijdeman, "Analysis and comparison of some integer factoring algorithms", *Computational methods in number theory*, no. 1, pp. 89-139, 1982.

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

[5] Ш. Ишмухаметов, *Методы факторизации натуральных чисел*: учебное пособие. Казань, Россия: Казанский университет, 2011.

[6] А. Корнейко, А. Жилин, "Анализ известных вычислительных методов факторизации многоразрядных чисел", *Моделювання та і формаційні технології: Збірник наукових праць*, № 61, С. 3-13, 2011.

[7] Д. Кнут, *Искусство программирования. Получисленные алгоритмы*. Москва, Россия: Вильямс, 2001.

[8] С. Винничук, Е. 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.

[9] Є. Максименко, "Вибір ефективної базової основи модуля при багаторазовому проріджуванні пробних значень в методі факторизації Ферма з нерівномірним кроком", *Інформатика та математичні методи в моделюванні*, том 6, № 3, С. 270-279, 2016.
[10] С. Винничук, Е. Максименко, "Формирование неравномерных приращений для базового основания модуля в задаче факторизации методом Ферма", *Information Technology and Security*, том. 4, № 2, С. 245-254, 2016.

Стаття надійшла до редакції 18 вересня 2018 року.

REFERENCE

[1] D. Brown, "Breaking RSA May Be As Difficult As Factoring". [Online]. Available at: https://eprint.iacr.org/2005/380.
[2] D. Aggarwal, U. Maurer, "Breaking RSA Generically is Equivalent to Factoring", in *Proc. "Advances in Cryptology – EUROCRYPT"*, 2009 [Online]. Available at: https://eprint.iacr.org/2008/260.
[3] C. Pomerance, H. Lenstra, R. Tijdeman, "Analysis and comparison of some integer factoring algorithms", *Computational methods in number theory*, no. 1, pp. 89-139, 1982.
[4] O. Vasylenko, Teoretyko-chislovye algoritmy v kryptohrafyy. Moskva, Russia: MCNMO, 2003.
[5] Sh. Ishmukhametov, Methods of factoring integers. Kazan, Russia: Kazan University, 2011.
[6] A. Korneiko, and A. Zhilin, "Analysis of the known methods for computing the factorization of large numbers", *Collection of scientific works Institute of Modelling Problems inPower Engineering*, vol. 61, pp. 3-13, 2011.
[7] D. Knuth, The Art of Computer Programming. Moscow, Russia: Vilyams, 2001.
[8] S. Vinnichuk, and Y. Maksymenko, "Mnogokratnoe prorezhivanie dlya uskoreniya metoda faktorizacii Ferma pri neravnomernyh shagah dlya neizvestnoj", *Informatics, operation and computer scienc*, no. 64, pp. 13-24, 2016.
[9] Y. Maksymenko, "Vybir efektyvnoi bazovoy osnovy modulya pry bagatorazovomu proridzhuvanni probnykh znachen v metodi faktoryzaciyi Ferma z nerivnomirnym krokom", *Informatyka ta matematychni metody v modeliuvanni*, vol. 6, no. 3, pp. 270-279, 2016.
[10] S. Vynnychuk, and Y. Maksymenko, "Formirovanie neravnomernyh prirashchenij dlya bazovogo osnovaniya modulya v zadache faktorizacii metodom Ferma", *Information technology and security*, vol. 4, no. 2, pp. 245-254, 2016.

СТЕПАН ВИННИЧУК,
ЕВГЕНИЙ МАКСИМЕНКО

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ МЕТОДА ФАКТОРИЗАЦИИ ФЕРМА С ПРИМЕНЕНИЕМ БАЗОВОГО ОСНОВАНИЯ МОДУЛЯ

Метод факторизации Ферма считается лучшим при факторизации чисел вида $N = p \cdot q$ в случаях когда множители p и q близки по значению. Вычислительная сложность базового алгоритма метода факторизации Ферма определяется количеством пробных значений X при решении уравнения $Y^2 = X^2 – N$, а также сложностью выполнения арифметических операций с большими числами: возведения в квадрат, сложения, вычисления квадратного корня. Снижение вычислительной сложности указанного метода обеспечивается за счет использования множества оснований модулей в уравнении $Y^2 modb = (X^2-N) modb$, что в свою очередь позволяет пренебрегать сложностью операций вычисления квадратного корня. Для уменьшения числа пробных значений X рассматривается возможность использования базового (первичного) основания модуля bb. Оптимальный выбор базового основания bb позволяет уменьшить число пробных X на величину, близкую по значению к величине коэффициента ускорения $z(N, bb) = bb/bb^*$, где bb^* - число элементов множества корней уравнения $(Ymodb)^2 modb = ((Xmodb)^2 modb - Nmodb) modb$. Показано, что в общем случае

первичное основание модуля *bb* является произведением степеней простых чисел *p*, коэффициент ускорения для которого *z(N,bb)* является функцией *Nmodp* и показателей степеней *p*. Определено степень влияния значения остатков *Nmodp* (в случае *p*=2 используются остатки *Nmod8*) и показателей степеней простых *p* на величину коэффициента ускорения *z(N,bb)*). Предложено постановку задачи поиска оптимального значения первичного основания модуля bb при ограничениях на объем памяти ЕВМ и способ ее решения. Приведена оценка эффективности предложенного модифицированного алгоритма метода факторизации Ферма. Установлено, что предложенный алгоритм для чисел 2^{1024} обеспечивает снижение вычислительной сложности по сравнению с базовым алгоритмом в среднем в 10^7 раз. Использование предложенного метода позволит проектировать более эффективные, с точки зрения быстродействия, аппаратно-программные средства проведения криптоанализа асимметричных криптографических алгоритмов и, как следствие, повысить качество оценки криптостойкости алгоритма RSA.

Ключевые слова: факторизация, метод Ферма, прореживание, базовое основание, вычислительная сложность.

STEPAN VYNNYCHUK,
YEVHEN MAKSYMENKO

THE MODIFIED ALGORITHM OF FERMAT'S FACTORIZATION METHOD WITH BASE FOUNDATION OF MODULE

Fermat's factorization method is considered the best at factorization the numbers $N = p \cdot q$, when *p* and *q* are close in value. The computational complexity of Fermat's base factorization method is defined by the number of trial *X's* when solving the equation $Y^2=X^2-N$, and the complexity of arithmetic operations with large numbers: quadrating, addition, square root calculation. The decrease in computing complexity of this method is provided due to use of a set of bases of modules in $Y^2 modb = (X^2-N) modbb$ that in turn allows to neglect complexity of calculation square root. Allocation of primary basis of the *bb* module allows to reduce number of the trial *X* in number times close to $z(N, bb) = bb/bb^*$, where *bb** - number of roots of an equation $(Ymodb)^2 modb = ((Xmodb)^2 \ modb - Nmodb) modb$. It is shown that in the general case the primary base of the module *bb* is the product of the degrees of prime numbers *p*, the acceleration factor for which *z(N, bb)* is the *Nmodp* and exponents of *p*. It is determined that the values of the *Nmodp* residues influence the value of *z(N, bb)* (for *p*=2, the *Nmod8* residues are used) and exponents of simple p by the value of the acceleration coefficient *z(N, bb)*). It is offered problem formulation of search optimal *bb* with restrictions for the memory size of PC and way of it decision. An estimate of the effectiveness proposed a modified algorithm of the Fermat's factorization method is given. It is shown that offered algorithm in the case of numbers 2^{1024} provides a decrease in computational complexity in comparing with the basic algorithm of the Fermat's method on average of 10^7 times. Use of the proposed method will allow developing more efficient, in terms of speed, hardware and software cryptanalysis tools for asymmetric cryptographic algorithms and, as a result, improve quality the evaluation of strong cryptography of asymmetric RSA cryptographic algorithms.

Keywords: factorization, Fermat method, thinning, base foundation, computational complexity.

Виничук Степан Дмитрович, доктор технічних наук, старший науковий співробітник, керівник відділу моделювання енергетичних процесів і систем, Інститут проблем моделювання в енергетиці ім. Г. Є. Пухова НАН України, Київ, Україна.

ORCID: https://orcid.org/0000-0002-0605-1576

E-mail: vynnychuk@i.ua.

Максименко Євген Васильович, кандидат технічних наук, 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.

ORCID: https://orcid.org/0000-0003-4947-2247

E-mail: maksimenco@gmail.com.

Виничук Степан Дмитриевич, доктор технических наук, старший научный сотрудник, руководитель отдела моделирования энергетических процессов и систем, Институт проблем моделирования в энергетике им. Г. Е. Пухова НАН Украины. Киев, Украина.

Максименко Евгений Васильович, кандидат технических наук, заместитель заведующего кафедры кибербезопасности и применения информационных систем и технологий, Институт специальной связи и защиты информации Национального технического университету Украины "Киевский политехнический институт имени Игоря Сикорского", Киев, Украина.

Stepan Vynnychuk, doctor of technical sciences, senior researcher, head of department of modeling of energy processes and systems, Pukhov Institute for Modelling in Energy Engineering of National academy of sciences of Ukraine, Kyiv, Ukraine.

Yevhen Maksymenko, candidate of technical sciences, deputy head at the cybersecurity and application of information systems academic department, Institute of Special Communication and Information Protection of National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kiev, Ukraine.