УДК 004.056.05

О. Марковський, Аль-Мріят Гассан Абдель Жаліль

МЕТОД ПРИСКОРЕНОГО МОДУЛЯРНОГО МНОЖЕННЯ ДЛЯ МЕХАНІЗМІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ З ВІДКРИТИМ КЛЮЧЕМ

Анотація: В статті запропоновано метод прискорення важливої для криптографічних застосувань операції модулярного множення довгих чисел за рахунок суміщенням множень секцій чисел з симетричними індексами, а також чередування циклів додавання секційних добутків з однаковою вагою і груповою редукцією Монтгомері.

Розроблений метод в теоретичному плані базується на технологій швидкого множення і являє собою їх розвиток для задачі попарного множення секцій довгих чисел. Наведені теоретичні обґрунтування та формалізований виклад запропонованого методу. Виклад проілюстровано числовим прикладом.

Теоретично показано і експериментально підтверджено, що запропонований метод дозволяє за рахунок скорочення кількості операцій процесорного множення та групової редукції Монтгомері прискорити обчислювальну реалізацію важливу для криптографічних застосувань операцію модулярного множення довгих чисел в 4-6 раз.

Ключові слова: модулярне множення, технології швидкого множення, модулярна редукція Монтгомері, криптографія з відкритии ключем

Вступ

Криптографія з відкритим ключем відіграє ключову роль в арсеналі сучасних засобів захисту інформації. Вона лежить в основі таких важливих механізмів як несиметричне шифрування, обмін ключами, цифровий підпис, криптографічно строга ідентифікації учасників віддаленої інформаційної взаємодії [1]. Найбільш суттєвий недолік існуючих алгоритмів на основі криптографії з відкритим ключем полягає в тому, що їх реалізація потребує обчислення модулярної експоненти над числами, довжина яких значно перевищує розрядність процесора. Зокрема, в більшості протоколів ця операція здійснюється над числами, довжина яких становить 2048 або 4096 біт [2]. Відповідно, реалізація таких громіздких обчислень потребує значного часу. Для великої частини обчислювальних платформ, включаючи ноутбуки, ця проблема вирішується за рахунок комплектування їх криптопроцесорами – спеціалізованими пристроями, які на 2-3 порядки прискорюють обчислення модулярної експоненти. Проте існує широкий клас обчислювальних платформ, для яких таке рішення не прийнятне. Зокрема, сьогодні динамічно розвиваються системи комп'ютерного моніторингу стану та управління різноманітними віддаленими

© **О. Марковський, Аль-Мріят Гассан Абдель Жаліль**

 ISSN 1560-8956

об’єктами реального світу на базі технології IoT, тобто з використанням в якості середовища обміну даними Інтернету [3]. Безпосереднє управління в таких систем здійснюється обладнаними радіо модемами малопотужними термінальними мікроконтролерами. Разом з тим, в таких системах віддаленого управління важливо забезпечити надійний захист від стороннього втручання [4], що потребує використання всього арсеналу сучасних криптографічних механізмів, в тому числі з відкритим ключем. При цьому важливо забезпечити реалізацію криптографічних алгоритмів з відкритим ключем в реальному часі.

Таким чином, наукова задача прискорення мультиплікативних операцій модулярної арифметики для механізмів криптографічного захисту є актуальною та практично важливою для сучасного етапу розвиту комп’ютерних технологій.

Огляд методів прискорення модулярного множення

Як зазначалося вище, базовою обчислювальною операцією криптографії з відкритим ключем є модулярне експоненціювання. При цьому рівень захищеності повною мірою визначається розрядністю n чисел, над якими здійснюється ця операція [4].

Обчислення модулярної експоненти $A^E \bmod M$ реалізується за одним із двох алгоритмів, які передбачають послідовний аналіз двійкових розрядів коду експоненти E. Алгоритми відрізняються порядком, в якому аналізуються розряди коду експоненти. При експоненціюванні зі старших розрядів коду експоненти E задіяна одна змінна R, яка на початку встановлюється в одиницю. В кожному із n циклів змінна R спочатку підноситься до квадрату $R=R^2 \bmod M$, а потім, в залежності від значення поточного біту коду E, множиться на A: $R=R\cdot A \bmod M$. Алгоритм модулярного експоненціювання з молодших розрядів коду експоненти E передбачає використання двох змінних: D і R, стартові значення яких: D=A та $R=1$. В кожному із n циклів змінна R в залежності від значення поточного біту коду E, множиться на D: $R=R\cdot D \bmod M$, а змінна D підноситься до квадрату $D=D^2 \bmod M$. Цілком очевидно, що обидва алгоритми модулярного експоненціювання мають строго послідовний характер і потребують $1.5\cdot n$ операцій модулярного множення n-розрядних чисел.

Таким чином, основний резерв прискорення базової операції криптографії з відкритим ключем – модулярного експоненціювання полягає в зменшенні часу виконання модулярного множення багаторозрядних чисел.

Операція модулярного множення $A\cdot B \bmod M$ складається з власне множення $A\cdot B$ та модулярної редукції, тобто віднаходження залишку від ділення добутку $A\cdot B$ на модуль M. Ці дві складові можуть виконувати роздільно або суміщуватися.

На практиці операція множення n-розрядних може побітово або фрагментами, довжина яких дорівнює розрядності r процесора. Останній варіант отримав більше

розповсюдження на практиці в силу того, що він дозволяє повною мірою використовувати можливості процесора по виконанні операції процесорного множення. В теоретичному плані, в більшості процесорів операція множення здійснюється за однією з 4-х класичних схем множень, теоретична складність яких $O(r^2)$.

Теоретично відомі більш швидкі схеми множення, історично першою з яких є метод А. Карацуби [5] з асимптотичною складністю $O(n^{\log 3/ \log 2})$. Згідно з методом А. Карацуби числа A і B розрядністю n розділяються на два фрагменти рівної довжини: $A= a_1 \cdot 2^{n/2} + a_0$ та $B= b_1 \cdot 2^{n/2} + b_0$. За класичною схемою множення добуток $A \cdot B$ обчислюється у вигляді $A \cdot B = a_1 \cdot b_2 \cdot 2^n + (a_1 \cdot b_0 + a_0 \cdot b_1) \cdot 2^{n/2} + a_0 \cdot b_0$, що потребує 4 множення $n/2$-розрядних чисел. За схемою А.Карацуби добуток $A \cdot B$ обчислюється у вигляді: $A \cdot B = a_1 \cdot b_1 \cdot 2^n + 2^{n/2} \cdot ((a_1 + a_0) \cdot (b_1 + b_0) - a_1 \cdot b_1 - a_0 \cdot b_0) + a_0 \cdot b_0$, тобто потребує три операції множення $n/2$-розрядних чисел. Зазначена вище складність досягається асимптотично, тобто при достатньою великих значеннях n і рекурсивному розкладенні множників.

Пізніше були опубліковані методи з меншою асимптотичною складністю. Зокрема алгоритм Шенхаге-Штрасера використовує для швидкого множення цілих чисел швидке перетворення Фур'є забезпечуючи при цьому складність $O(n \cdot \log n \cdot \log \log n)$ [6]. Широку відомість отримав метод Фюрера [7], асимптотична складність якого ще менша. Ще пізніше був запропонований ще більш швидкий метод множення [8]. Ці методи мають більше теоретичне, ніж практичне значення, оскільки різниця в швидкодії наведених методів стає помітною тільки при великих значеннях $n>40000$ [8]. Незважаючи на значну складність практичної реалізації схем швидкого множення в їх класичному вигляді, окремі їх елементи можуть бути ефективно використані для зменшення операцій секційного множення.

Для вирішення задачі модулярної редукції застосовуються технології заміни операції ділення на операції множення та зсуву. Перша з цих технологій запропонована Барреттом [9] і полягає в обчисленні $A \cdot B \bmod M$ у вигляді різниці: $A \cdot B \bmod M = A \cdot B - q \cdot M$. Значення q обчислюється як результат добутку: $\lfloor \gamma \cdot \lfloor A \cdot B \cdot 2^{2 \cdot n} \rfloor \cdot 2^{-n} \rfloor$, де γ - константа, яка залежить тільки від модуля M: $\gamma = \lfloor 2^{2 \cdot n} \cdot M^{-1} \rfloor$ і, відповідно обчислюється один раз при зміні модуля. З викладеного ясно, що в технології Барретта модулярна редукція здійснюється двома операціями множення. Відповідно, для прискорення редукції потрібно зменшувати час множення. При цьому, якщо час множення довгих чисел зменшити в α раз, то відповідним чином, тобто в α раз прискориться виконання операції модулярного множення.

Друга з відомих технологій модулярної редукції запропонована П. Монтгомері [10] і передбачає віднаходження найменшого u такого, що сума $A \cdot B + u \cdot M$ націло ділиться на 2^n. Технологічно редукція здійснюється за n циклів, в кожному із яких

©

значення поточного результату Q (початкове значення $Q=A\cdot B$) зсувається праворуч. Якщо Q непарне, то до нього попередньо додається непарне значення модуля M : $Q=Q+M$. Результатом цієї процедури є число $Q=A\cdot B\cdot Y^{-1} \bmod M$, де Y^{-1} – мультиплікативна інверсія коду $Y= 2^n$ по модулю M. Вагома перевага розглянутої технології модулярної редукції Монггомері полягає в можливості суміщення в часі множення та редукції.

Проведений огляд існуючих технологій комп’ютерної реалізації модулярного множення чисел, довжина яких значно перевищує розрядність процесора показав, що існують певні ресурси для прискорення цієї операції за рахунок застосування елементів технологій швидкого множення А. Карацуби, а також зменшення часу редукції при використанні схеми П. Монтгомері за рахунок використання передобчислень, що залежать лише від модуля.

Мета досліджень полягає в прискоренні обчислювальної реалізації модулярного множення чисел, довжина яких значно перевищує розрядність процесора – базової операції сучасних криптографічних алгоритмів захисту даних з відкритим ключем.

Обчислення модулярного добутку з суміщенням секційних множеньта групової редукції Монтгомері

Для досягнення поставленої мети пропонується нове застосування технологій прискореного множення А. Карацуби [5] для множення чисел, довжина яких значно перевищує розрядність процесора, в поєднанні з використанням передобчислень для швидкої реалізації групової редукції Монтгомері.

При реалізації модулярного множення $A\cdot B \bmod M$ n-розрядних чисел $A<M$ та $B<M$ на r-розрядному процесорі, $n>>r$, вони поділяються на $s=n/r$ секцій, довжина яких дорівнює розрядності процесора: $A=a_s\cdot 2^{(s-1)\cdot r} + a_{s-1}\cdot 2^{(s-2)\cdot r}+\ldots+a_2\cdot 2^r + a_1$, $B=b_s\cdot 2^{(s-1)\cdot r} + b_{s-1}\cdot 2^{(s-2)\cdot r}+\ldots+b_2\cdot 2^r + b_1$, $\forall j\in\{1,2,\ldots,s\}$: $a_j,b_j\in\{0,1,\ldots,2^r-1\}$.

Обчислення модулярного добутку $A\cdot B \bmod M$ організується шляхом попарних множень секцій чисел A і B з урахуванням їх вагових коефіцієнтів, відповідно до наступної формули:

$$A\cdot B \bmod M = \left(\sum_{j=1}^{s}\sum_{i=1}^{s} a_i\cdot b_j\cdot 2^{(i+j-2)\cdot r}\right) \bmod M \ . \qquad (1)$$

Сума двох добутків $a_i\cdot b_j + a_j\cdot b_i$, що входять в формулу (1) з однаковими ваговими коефіцієнтами може бути обчислена у вигляді:

$$a_i\cdot b_j + a_j\cdot b_i = (a_i + a_j)\cdot(b_i + b_j) - a_1\cdot b_i + a_j\cdot b_j \ . \qquad (2)$$

З урахуванням отриманого виразу (2), формула для обчислення модулярного добутку (1) може бути перетворена до наступного виду:

$$A \cdot B \bmod M = (\sum_{j=1}^{s-1} \sum_{i=j+1}^{s} ((a_i + a_j) \cdot (b_i + b_j) - a_i \cdot b_i - a_j \cdot b_j) \cdot 2^{(i+j-2) \cdot r} + \sum_{l=1}^{s} a_l \cdot b_l \cdot 2^{2 \cdot l \cdot r}) \bmod M \ . \quad (3)$$

Обчислення модулярного добутку за формулою (3) дозволяє практично вдвічі скоротити кількість операцій процесорного множення в порівнянні з формулою (1). Суттєвий недолік обчислення модулярного добутку за формулою (3) полягає в тому, що фактично операції множення та модулярної редукції рознесені. Це має наслідком накопичення розрядності добутку до $2 \cdot n$ ($2 \cdot s$ секцій), що ускладнює організацію і збільшує час обчислень.

Для поєднання в часі процесів множення та модулярної редукції формула (3) має бути модифікована таким чином, щоб згрупувати всі мультиплікативні доданки з однаковими ваговими коефіцієнтами. Формулу (1) можна розглядати як суму елементів квадратної матриці, що містить по s рядків та стовпців. Кожен елемент такої матриці, локалізований в її j–тому рядку та i-тому стовпці являє собою добуток $a_i \cdot b_j$ помножений на ваговий коефіцієнт $2^{(i+j-2) \cdot r}$. Відповідно, під вказаним кутом зору, формулу (3) можна трактувати як суму елементів діагоналі цієї матриці і суму модифікованих формулою (2) елементів матриці, що знаходяться вище побічної діагоналі. Цілком очевидно, що елементи, що знаходяться на побічних діагоналях матриці мають однаковий ваговий коефіцієнт. Тому, для отримання формули ефективного обчислення модулярного множення потрібно організувати послідовне сканування фрагментів побічних діагоналей, що знаходяться на головній діагоналі і вище її . Оскільки загальна кількість побічних діагоналей матриці розміром s × s становить $2 \cdot s-1$, то зовнішній індекс j їх сканування може змінюватися від 2-х до $2 \cdot s$. Внутрішній індекс і сканування побічної діагоналі до її перетину з головною діагоналлю змінюється від одиниці при $j \le s$ та від $j-s+1$ при значеннях j більших s. Відповідна формула для обрахування модулярного добутку має наступний вигляд:

$$A \cdot B \bmod M = (\sum_{j=2}^{2 \cdot s} (((j-1) \bmod 2) \cdot a_{j/2} \cdot b_{j/2} + \sum_{\substack{j \le s : i=1 \\ j > s : i=j-s+1}}^{j/2-1} ((a_i + a_{j-i}) \cdot (b_i + b_{j-1}) - a_i \cdot b_i - a_{j-i} \cdot b_{j-i})) \cdot 2^{j \cdot r}) \bmod M) \bmod M \ . \quad (4)$$

Формула (4) дозволяє сумістити операції множення та модулярної редукції Монтгомері. Часткову модулярну редукцію Монтгомері пропонується реалізувати поетапно після обрахунку суми компонентів формули (4), що мають однаковий ваговий коефіцієнт. При цьому часткова редукція Монтгомері здійснюється на r розрядів. Таким чином.

Для прискорення обчислювальної реалізації модулярної редукції Монтгомері пропонується організувати одночасну редукцію не на один розряд, а відразу на групу із

η розрядів. Для цього до поточного коду суми часткових добутків пропонується додавати добуток модуля M на ціле e: $e \cdot M$, таке, щоб η молодших розрядів суми стали нульовими. Відповідно, код такої суми може бути зсунутий праворуч відразу на η розрядів. В силу того, що в практичних застосуваннях модуль M завжди непарне число, очевидно, що $0 \le e \le 2^{\eta}-1$.

Для реалізації групової редукції Монтгомері за запропонованою схемою доцільно використати попередньо створену таблицю передобчислень. В таблиці для кожного із 2^{η} можливих значень η молодших розрядів поточного коду суми часткових добутків зберігається відповідне значення добутку $e \cdot M$. Очевидно, що об’єм такої таблиці передобчислень становить $2^{\eta} \cdot (n+\eta)$. Чисельне значення η обирається виходячи з обмежень на об’єм пам'яті для зберігання таблиць передобчислень і таким чином, щоб r було кратним η. Оскільки заповнення таблиці передобчислень залежить тільки від модуля M, вона створюється один раз при зміні модуля. В силу того, що модуль M в реальних системах криптографічного захисту є частиною відкритого ключа, він може вважатися практично незмінним.

В формалізованому вигляді пропонований метод прискореного обчислення модулярного добутку довгих чисел зводиться до виконання наступної послідовності дій:

1. Обчислюються значення s добутків однойменних секцій операндів A та B: $a_1 \cdot b_1$, $a_2 \cdot b_2$,…, $a_s \cdot b_s$.

2. Початкове значення індексу j визначається рівним 2-м: j=2. Початкове значення поточної суми часткових добутків дорівнює нулю: Q=0.

3. Якщо значення j парне, тобто $j \bmod 2 = 0$, до Q додається добуток $a_{j/2} \cdot b_{j/2}$: $Q = Q + a_{j/2} \cdot b_{j/2}$.

4. Якщо значення зовнішнього індексу j не перевищує s, тобто $j \le s$, то початкове значення внутрішнього індексу i визначається рівним одиниці: i=1, інакше старше значення і визначається у вигляді: $i = j - s + 1$.

5. Якщо $i < j/2$, то до Q додається $(a_i + a_{j-i}) \cdot (b_{j-i} + b_i) - a_i \cdot b_i - a_{j-i} \cdot b_{j-i}$: інакше перехід на п. 7.

6. Значення внутрішнього індексу i збільшується на одиницю: $i = i + 1$ з переходом на повторне виконання п.5.

7. Якщо $j = 2 \cdot s$ виконується перехід на п.10, інакше здійснюється r/η циклів групової редукції Монтгомері з використанням таблиць передобчислень. Індекс k циклу редукції встановлюється в одиницю: k=1.

8. Обчислюється код числа u, утвореного η молодшими розрядами числа Q: $u = q_{\eta} \cdot 2^{\eta-1} + q_{\eta-1} \cdot 2^{\eta-2} + \ldots + q_2 \cdot 2 + q$. До числа Q додається u-те значення таблиці передобчислень G: $Q = Q + G[u]$. Код Q зсувається праворуч на η бітів: $Q = Q >> \eta$.

Збільшується на одиницю індекс k циклу редукції: $k=k+1$. Якщо $k \le r/\eta$, виконання п.8 повторюється.

9. Збільшується на одиницю значення j: $j=j+1$ та повергнення на повторне виконання п. 3.

10. Кінець процедури модулярного множення. В змінній Q знаходиться результат у вигляді: $Q=A\cdot B\cdot Y^{-1} \bmod M$, де Y^{-1} – мультиплікативна інверсія коду $Y= 2^{2\cdot(n-r)}$.

Робота викладеного методу прискореного обчислення модулярного добутку може бути ілюстрована наступним прикладом. Нехай модуль M = 67·59 = 3953 = 111 101 110 001_2. Відповідно, n=12. Якщо прийняти, що розрядність r процесора дорівнює 3-м, тобто r = 3, то кожен з операндів складається з 4-х 3-розрядних секцій, тобто кількість s секцій дорівнює 4-м: s = 4. Якщо кількість η розрядів, що оброблюються одночасно в рамках групової редукції Монтгомері прийняти рівним 3-м, тобто η=3, то таблиця передобчислень G включає в себе $2^{\eta}=2^3$ =8 значень і має вигляд показаний в табл. 1.

Таблиця 1. **Таблиця передобчислень *G*[*u*]**

u	$p_3\,p_2\,p_1$	$G[u]$	u	$p_3\,p_2\,p_1$	$G[u]$
0	0 0 0	0	4	1 0 0	4·M = 15812
1	0 0 1	7·M = 27671	5	1 0 1	3·M = 11859
2	0 1 0	6·M = 23718	6	1 1 0	2·M = 7906
3	0 1 1	5·M = 19765	7	1 1 1	M = 3953

Припустимо, що виконується модулярне множення A·B mod M = 2937·2460 mod 3953 = 2889. Відповідно, операнд A =2937=101 101 111 001_2, а операнд B = 2460_{10} = = 100 110 011 100_2. Відповідно, операнд A складається з секцій: $a_1 = 001_2$ =1; $a_2 = 111_2$= =7; $a_3 = 101_2$ =5; $a_4 = 101_2$=5, а секції операнду B мають такий вигляд: $b_1=100_2$ =4; $b_2=011_2$ =3; $b_3=110_2$ =6; $b_4=100_2$ =4.

У відповідності з п.1 обчислюються значення $a_1\cdot b_1$ = 1·4 =4; $a_2\cdot b_2$ = 7·3 = 21; $a_3\cdot b_3$ = 5·6 =30; $a_4\cdot b_4$ = 5·4 =20. З значення індексу j визначається рівним 2-м: j=2, а початкове значення Q встановлюється рівним нулю: Q = 0. Оскільки, поточне значення індексу j=2 парне, то згідно п.3 здійснюється додавання до Q добутку $a_{j/2}\cdot b_{j/2} = a_1\cdot b_1$: Q = Q+ $a_1\cdot b_1$ = 0 + 4=4.

В силу того, що при j=2 і значенні внутрішнього індексу i=1 умова $i < j/2$ не виконується ($i= j/2$ =1), то здійснюється перехід на п.7, в рамках якого реалізується редукція проміжного результату Q. Для поточного прикладу кількість розрядів, на які виконується редукція дорівнює розрядності процесора: η= r =3, тому редукція здійснюється за одну операцію. Ця операція включає додавання до Q табличного значення, що індексується трьома молодшими двійковими розрядами Q: Q = = $Q + G[q_3\cdot 4+q_2\cdot 2+q_1]$ = 4 + G[4] = 4+15812 = 15816. Після додавання здійснюється зсув

 ISSN 1560-8956

Q на 3 розряди праворуч: Q = Q>>3 = 15816>>3 = 1977.

При j=3 та значенні внутрішнього індексу i=1 умова $i < j/2$ виконується. Тому, до поточного значення Q=1977 додається $(a_i + a_{j-i})\cdot(b_{j-i} + b_i) - a_i\cdot b_i - a_{j-i}\cdot b_{j-i} = (a_1+ a_2)\cdot(b_2+ b_1) - a_1\cdot b_1 - a_2\cdot b_2$: Q=1977+ (1+7)·(3+4) - 4 – 21 = 1977+31 = 2008 = =111 110 110 000_2. Групова редукція Монтгомері реалізується додаванням до Q табличного значення, що індексується трьома молодшими двійковими розрядами Q, які для Q=2008 дорівнюють нулю. Відповідно, до Q додається нуль, після чого Q зсувається на три розряди праворуч: Q = Q>>3 = 2008>>3 = 251.

При j=4, у відповідності з п.3, до поточного результату Q додається значення добутку $a_{j/2}\cdot b_{j/2} = a_2\cdot b_2$: Q = Q+ $a_2\cdot b_2$ = 251 + 21= 272. При j=4 внутрішній індекс приймає тільки одне значення i=1. Відповідно, до поточного значення Q=272 додається $(a_i + a_{j-i})\cdot(b_{j-i} + b_i) - a_i\cdot b_i - a_{j-i}\cdot b_{j-i} = (a_1+ a_3)\cdot(b_3+ b_1) - a_1\cdot b_1 - a_3\cdot b_3$: Q=272+ (1+5)·(6+4) - – 4 – 30 = 272+26 = 298 = 100 101 010_2. Групова редукція Монтгомері реалізується додаванням до Q табличного значення, що індексується трьома молодшими двійковими розрядами Q, які для Q=298 утворюють номер 010_2=2. Відповідно, до Q додається табличне значення G[2]: Q = 298 + 23718 = 24016 = 101 110 111 010 000_2. Отриманий код Q зсувається на три розряди праворуч: Q = Q>>3 = 24016>>3 = 3002.

При j=5 внутрішній індекс i приймає два значення: i=1 та i=2. При i=1 до Q додається $(a_1+ a_4)\cdot(b_4+ b_1) - a_1\cdot b_1 - a_4\cdot b_4$: Q=3002+ (1+5)·(4+4) - 4 – 20 = 3002+24 = = 3026_2. При i=2 до Q додається $(a_2+ a_3)\cdot(b_3+ b_2) - a_2\cdot b_2 - a_3\cdot b_3$: Q=3026+ (7+5)·(6+3) - –21 – 30 = 3026+57 = 3083_2= 110 000 001 011_2. Групова редукція Монтгомері здійснюється в рамках п.7 додаванням до Q табличного значення, що індексується трьома молодшими двійковими розрядами Q, які для Q=3083 утворюють номер 011_2=3. Відповідно, до Q додається табличне значення G[3]: Q = 3083 + 19765 = 22848 = =101 100 101 000 000_2. Отриманий код Q зсувається на три розряди праворуч: Q = Q>>3 = 22848>>3 = 2856.

При j=6, у відповідності з п.3, до поточного результату Q додається значення добутку $a_{j/2}\cdot b_{j/2} = a_3\cdot b_3$: Q = Q+ $a_3\cdot b_3$ = 2856 + 30= 2886. При j=6 внутрішній індекс приймає тільки одне значення i=2. Відповідно, до поточного значення Q=2886 додається $(a_i + a_{j-i})\cdot(b_{j-i} + b_i) - a_i\cdot b_i - a_{j-i}\cdot b_{j-i} = (a_2+ a_4)\cdot(b_4+ b_2) - a_2\cdot b_2 - a_4\cdot b_4$: Q=2886+ (7+5)·(4+3) - 21 – 20 = 2886+43 = 2929 = = 101 101 110 001_2. Групова редукція Монтгомері реалізується додаванням до Q табличного значення, що індексується трьома молодшими двійковими розрядами Q, які для Q=2929 утворюють номер 001_2=1. Відповідно, до Q додається табличне значення G[1]: Q = 2929 + 27671 = 30600 = 111 011 110 001 000_2. Отриманий код Q зсувається на три розряди праворуч: Q = Q>>3 = 30600>>3 = 3825.

При j=7 внутрішній індекс i приймає тільки одне значення: i=3. До Q додається $(a_3+ a_4)\cdot(b_4+ b_3) - a_3\cdot b_3 - a_4\cdot b_4$: Q=3825+ (5+5)·(4+6) - 30 – 20 = 3825+50 = 3875_2= = 111 100 100 011_2. Групова редукція Монтгомері здійснюється в рамках п.7 додаванням до Q табличного значення, що індексується трьома молодшими двійковими

розрядами Q, які для Q=3875 утворюють номер 011_2=3. Відповідно, до Q додається табличне значення G[3]: Q = 3875 + 19765= 23640 = 101 110 001 100 000_2. Отриманий код Q зсувається на три розряди праворуч: Q = Q>>3 = 23640>>3 = 2955.

При j=8 до поточного результату Q додається значення добутку $a_{j/2}\cdot b_{j/2} = a_4\cdot b_4$: Q = Q+ $a_4\cdot b_4$ = 2955 + 20= 2975. Редукція при j=8 не виконується. Отриманий результат Q в силу застосування редукції Монтгомері відрізняється від справжнього. Для отримання останнього потрібно виконати модулярне множення одержаного значення Q на два в ступені кількості виконаних правих зсувів. В рамках розглянутого прикладу було здійснено 6 циклів, в кожному з яких реалізовано 3 зсувів. Таким чином, для одержання істинного результату потрібно здійснити модулярне множення $Q\cdot 2^{18} \bmod M = 2975\cdot 2^{18} \bmod 3953 = 2889$.

Оцінка ефективності

Виходячи з цілі дослідження, в якості основного показника ефективності запропонованого методу може розглядатися коефіцієнт β прискорення обчислювальної реалізації операції модулярного множення цілих чисел, довжина яких значно перевищує розрядність процесора. Чисельне значення коефіцієнту прискорення β визначається співвідношенням часу T_b виконання цієї операції за базовим методом до часу T реалізації за запропонованим:

$$\beta = \frac{T_b}{T}. \tag{5}$$

В якості базового методу доцільно розглядати реалізацію з розділенням множення та редукції отриманого добутку. В базовій реалізації використовується попарне секційне множення та редукція отриманого результату по технології Монтгомері, тобто в базовій реалізації час модулярного множення складається з суми двох складових: часу T_m секційного множення та часу T_r редукції $2\cdot n$-го бітового результату множення: $T_b = T_m + T_r$. Теоретично, час T_r модулярної редукції вдвоє перевищує час виконання множення [11], тобто $T_r \approx 2\cdot T_m$. В базовій реалізації здійснюється n циклів редукції, в кожному з яких виконуються опрерації над числами, розрядність яких зменшується від $2\cdot n$ до n. Тобто, в середньому, можна вважати, що операції редукції виконуються над $1.5\cdot n$ – розрядними числами, тобто для редукції необхідно виконати $1.5\cdot n^2$ бітових операцій. За запропонованим методом здійснюється $(2\cdot s\text{-}2)\cdot r/\eta = 2\cdot(n\text{-}r)/\eta \approx 2\cdot n/\eta$ циклів редукції, в кожному з яких виконуються операції над n-розрядними числами. Тобто, в середньому, можна вважати, що для редукції необхідно виконати $2\cdot n^2/\eta$ бітових операцій. Це означає, що в оціночному плані, операція редукції в запропонованому методі виконується в $0.75\cdot\eta$ раз швидше в порівнянні з базовою реалізацією. Вище було показано, що за рахунок використання

 ISSN 1560-8956

технології А. Карацуби [5] час операції секційних множень скорочується практично вдвічі. Виходячи з наведено, значення коефіцієнту β прискорення модулярного множення при використанні запропонованого методу визначається формулою:

$$\beta = \frac{3 \cdot T_m}{0.5 \cdot T_m + \frac{2 \cdot T_m}{0.75 \cdot \eta}} = \frac{3 \cdot \eta}{0.5 \cdot \eta + 1.33}. \qquad (6)$$

Наприклад, при η=8, значення коефіцієнта β прискорення становить 4.5. Експериментальні дослідження показали близькі до теоретичних оцінок значення коефіцієнту прискорення β. Очевидно, що теоретично максимальне прискорення обчислення модулярного добутку при застосуванні розробленого методу дорівнює 6-ти.

Висновки

В результаті проведених досліджень запропоновано та досліджено метод прискореного модулярного множення довгих чисел, який відрізняється суміщенням множень секцій чисел з симетричними індексами, за рахунок чого скорочено практично вдвічі кількість операцій процесорного множення, а також чередування циклів додавання секційних добутків з однаковою вагою і груповою редукцією Монтгомері, що забезпечило скорочення часу модулярної редукції.

Розроблений метод в теоретичному плані базується на технологій швидкого множення А. Карацуби і являє собою їх розвиток для задачі попарного множення секцій довгих чисел. Метод також реалізує суміщення циклів множення та модулярної редукції Монтгомері шляхом їх чередування на рівні обробки добутків секцій з однакового вагою з одночасною редукцією групи бітів проміжної суми добутків. Це дозволило при використанні передобчислень значно прискорити обчислювальну реалізацію модулярної редукції.

Теоретично показано і експериментально підтверджено, що запропонований метод дозволяє за рахунок скорочення кількості операцій процесорного множення та групової редукції Монтгомері прискорити обчислювальну реалізацію важливу для криптографічних застосувань операцію модулярного множення довгих чисел в 4-6 раз.

Запропонований метод орієнтовано для реалізації протоколів захисту інформації в реальному часі для комп’ютерних систем моніторингу стану та управління віддаленими об’єктами реального світу, що використовують мережу Інтернет в якості середовища обміну даними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Menezes Alfred*, Handbook of Applied Cryptography. / Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone // CRC Press. – 2001. – 780 p.

2. *Listyowati D.* Public Key Crytography / Dwi Listyowati // Journal of Physics Conference Series. Ser. 1477.-2020.- p.1-7. DOI;10.1088/1742-6596/1477/5/052062.

3. *Jurcut A.D., Xu R.R.* Introduction to IoT Security/ In IoT Security: Advances in Authentication. – 2020.- pp. 1-64. DOI:10.1002.9781119527978.ch2.

4. *Unal D., Ali-Ali A., Catak F.O.* A secure and efficient Internet of Things cloud encryption scheme with forensics investigation compatibility based on identity-based encryption. Future Generation Computer Systems. Vol. 125. 2021.- pp. 433-445. DOI: 10.1016 /J.FUTURE.2021.06.050.

5. *Карацуба А.О.* Множення багатоцифрових чисел на автоматах / А.О. Карацуба, Ю.П.Офман / Доповіді Академії наук СРСР/ 1962.- № 2.- С.293-294.

6. *Schőnhage A.* Schnelle Multiplikation grosser Zahlen / A. Schőnhage and V. Strassen. //, Computing.- Vol.7.- 1971.- p.281–292.

7. *Fürrer M.* Faste Integer Multiplication / M. Fürrer //SIAM Journal on Computing. Vol. 39.- № 3.-2009. p.979-1005. DOI.ORG/10.1137/070711716.

8. *Harvey D.* Even faster integer multiplication / D.Harver, J. Van Der Hoeven, G. Lecerf // Journal of Complexity. Vol. 36.- № 2.-2014. p.979-1005. DOI.ORG/10.1016/j.jco.2016.03.001

9. *Barrett P.* Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor / P. Barrett // Proceedings CRYPTO'86. 1986.– p. 311-323.

10. *Montgomery P.* Modular multiplication without trial division / P. Montgomery // Mathematics of Computation. – 44(170). – 1985. – p. 519–521.

11. *Markovskyi O.* An Accelerate Approach for Public Key Cryptography Implementation on IoT Terminal Platforms / Oleksandr Markovskyi, Al-Mrayat Ghassan Abdel Jalil Halil, Nikolaos Doukas, Nikos Bardis // 13-th International Conference on Dependable system, Service and Technologies DESSERT-2023, 13-15 October, Greece, Athens. DOI 10.1109/DESSERT61349.2023.10416516.