УДК 004.052.42 DOI: 10.18372/2073-4751.86.21281

Русанова О.В., к.т.н.,
orcid.org/0000-0003-0145-3012,
e-mail: olga.rusanova@gmail.com,

Аль-Мраят Нізар Гассан Абдель Жаліль,
orcid.org/0009-0003-6296-485X,
e-mail: nezarukr@gmail.com,

Джура Анастасія Андріївна,
orcid.org/0009-0005-8346-1252,
e-mail: nastyafffokina@gmail.com,

Аль-Мраят Гассан Абдель Жаліль,
orcid.org/0000-0002-1610-1119,
e-mail: gmrayatjo@gmail.com

ПІДХІД ДО ПРИСКОРЕННЯ КОМП'ЮТЕРНОЇ РЕАЛІЗАЦІЇ МОДУЛЯРНОГО ЕКСПОНЕНЦІЮВАННЯ ДОВГИХ ЧИСЕЛ

Національний технічний університет України

"Київський політехнічний інститут ім. Ігоря Сікорського"

Вступ

Досягнутий протягом останніх двох десятиліть розвиток глобальних мереж суттєвим чином змінив підхід до використання технологій Інтернету речей (Internet of Things – IoT). В останні роки використання IoT, як технології контролю віддаленими об'єктами реального світу, яка була створена більше 20 років тому як дешевий спосіб управління побутовими приладами з використанням застосуванням Інтернету для обміну даними, зазнало докорінних змін. Нині технології IoT широко використовуються для віддаленого управління у багатьох сферах людської діяльності: у медицині - для дистанційного моніторингу стану хворих, у військовій - для управління безпілотними засобами ураження та розвідки, у промисловості - для управління віддаленими технологічними процесами, а також для керування безпілотними транспортними засобами [1]. У всіх зазначених сферах, використання існуючої інфраструктури Інтернету забезпечує низьку вартість проєктів та їх прискорене впровадження, але, разом з тим, передача даних через потенційно відкрите середовища Інтернету створює потенційну небезпеку зовнішнього втручання в роботу систем управління на базі IoT [2].

Таке втручання в роботу систем дистанційного контролю та управління може здійснюватися шляхом внесення змін в дані, якими обмінюються компоненти систем управління або надсилання їм фальшивих даних чи команд.

Класичним та ефективним криптографічним механізмом протидії таким видам атак та цілісність та автентичність даних виступає цифровий підпис. Всі існуючі механізми цифрового підпису, зокрема ISO-970 та DSS, базуються на використанні операцій модулярного експоненціювання, що виконуються над числами великої розрядності [3]. Саме розрядність n чисел визначає рівень криптостійкості механізмів цифрового підпису. На сьогодні NIST стандартизована розрядність n чисел, з якими працює цифровий підпис ISO-970, рівною 4096 біт, з тенденцією до подальшого збільшення до 8192 біт. Обчислення модулярної експоненти над такими довгими числами потребує мільйонів процесорних команд. На малопотужних термінальних мікроконтролерах систем

дистанційного управління виконання таких операцій потребує значного часу, що суттєво впливає на можливість роботи цих систем у реальному часі. Тому важливою та актуальною задачею є пошук шляхів прискорення виконання операцій модулярного експоненціювання на термінальних мікроконтролерах.

Таким чином, наукова задача прискорення комп'ютерної реалізації операцій модулярного множення, як частини модулярного експоненціювання, є актуальною та важливою з огляду на особливості сучасного етапу розвитку систем управління віддаленими об'єктами на базі технологій IoT.

Огляд сучасних технологій прискорення модулярного множення

Базова обчислювальна операція практично всіх сучасних механізмів захисту інформації на основі криптографії з відкритим ключем – модулярне експоненціювання $A^E \bmod M$ реалізується з використання одного із двох різновидів класичного алгоритму [3]. Цей алгоритм передбачає обчислення модулярної експоненти у вигляді n циклів, в кожному з яких виконуються операції, що залежать від поточного біту коду експоненти E=.

Зазначені вище різновиди класичного алгоритму експоненціювання вирізняються порядком, в якому послідовно аналізуються біти експоненти $E=e_0+ e_1\cdot 2+ e_2\cdot 2^2+\ldots+e_{n-1}\cdot 2^{n-1}$, $\forall i=0,1,\ldots,n-1$: $e_i\in\{0,1\}$.

В різновиді, в якому ці біти аналізуються починаючи від старшого, використовується одна змінна – R, в якій формується результат. Перед початком виконання n циклів, її значення встановлюється рівним одиниці: .R=1. В кожному із n циклів виконується дві операції: модулярне піднесення до квадрату поточного результату: R: $R=R^2 \bmod M$, а також модулярне множення на А поточного результату R: $R=R\cdot A \bmod M$; остання операція реалізується лише за умови коли поточний розряд e_i коду експоненти рівний одиниці.

В різновиді класичного алгоритму модулярного експоненціювання з аналізом бітів експоненти Е в напрямку від молодших розрядів до старших, задіюються дві змінні R та Q, початкові значення яких встановлюються рівними, відповідно, одиниці та числу A: R=1, $Q=A$. В кожному із n циклів також послідовно виконуються дві операції: спочатку, за умови, що поточний розряд e_i експоненти E рівний одиниці, значення результату R модулярно множиться на поточне значення Q: $R=R\cdot Q \bmod M$. Потім значення Q підноситься до квадрату по модулю M: $Q=Q^2 \bmod M$.

Операція модулярного множення $A\cdot B \bmod M$ складається з двох фаз: власне множення $A\cdot B$ та редукції, тобто обчислення залишку від ділення добутку $A\cdot B$ на модуль M. Ці дві фази можуть виконуватися як послідовно, так і одночасно.

Для прискорення виконання фази множення $A\cdot B$ можуть бути застосовані відомі методи, такі, як:

- одночасне множення на q розрядів множника B з використанням передобчислень всіх 2^q-1 можливих значень $C=c_1\cdot A+c_2\cdot 2\cdot A+\ldots+c_q\cdot 2^q\cdot A$; $\forall l=1,2,\ldots,q$: $c_l\in\{0,1\}$. Множення на декілька розрядів множника особливо ефективно при експоненціюванні зі старших розрядів коду експоненти, коли здійснюється n раз множення на однакове число;
- Застосування ітераційних методів прискореного множення довгих чисел, що мають за основу відомі схеми А.Карацуби та М. Фюрера [4];
- Організація множення секціями, довжина яких дорівнює розрядності r процесора [5]. Значима перевага цього підходу полягає в використанні повною мірою апаратних можливостей сучасних процесорів при виконанні операцій множення.

Редукція добутку $A\cdot B$ по модулю М може виконуватися за технологіями П. Барреттта або П.Монтгомері. Перша з цих технологій зводить віднаходження залишку від ділення добутку $A\cdot B$ на модуль М до двох операцій модулярного множення [6]. В технології П.Монтгомері [7] модулярна редукція зводиться до додавання модуля і зсуву. Найбільш вагомою перевагою редукції П.Монтгомері є можність

ефективного суміщення з фазою множення. Такий варіант суміщення отримав назву множення П.Монтгомері. Воно передбачає виконання n циклів, в кожному із яких виконуються такі операції: до суми S часткових добутків, при одиничному значенні поточного біту множника B додається множиме A: $S = S + A$; якщо сума S після цієї операції непарна, то неї додається модуль M: $S=S+M$. Виконання циклу завершується зсувом суми S ліворуч. Після виконання n циклів, в S сформовано значення $A \cdot B \cdot 2^{-n} \bmod M$. Щоб отримати правильне значення $A \cdot B \bmod M$, значення S потрібно домножити на 2^n: $A \cdot B \bmod M = S \cdot 2^n \bmod M$. Цілком очевидно, що реалізація множення П.Монтгомері потребує, в середньому, $2 \cdot n$ операцій над довгими числами, або $2 \cdot n^2 \cdot r^{-1}$ процесорних операцій.

Для прискорення модулярного множення П. Монтгомері було запропоновано низку методів. Один із них пропонує для прискорення множення Монтгомері можливості сучасних векторних процесорів [8]. Метод дозволяє суттєво прискорити реалізацію множення, проте не може бути застосований на дешевих малопотужних термінальних мікроконтролерах систем віддаленого управління на базі технологій IoT. Для прискорення множення Монтгомері пропонується також метод комбінованого використання множення на q розрядів множника B і групової редукції Монтгомері [9]. Метод забезпечує прискорення реалізації модулярного множення в q раз. Ще більш продуктивний метод модулярного множення Монтгомері запропоновано в роботі [10], який суміщає в часі додавання множника та модулярну корекцію.

Ще один відомий підхід до прискорення модулярного експоненціювання зі старших розрядів коду експоненти E полягає в суміщенні обробки групи з h розрядів коду експоненти [11]. Це дозволяє кількість операцій модулярного множення при реалізації експоненціювання приблизно в $h/2$ раз.

Проте, в силу того, що питома вага операцій множення на постійне число в алгоритмі експоненціювання складає близько третини, вказані методи, які прискорюють тільки операцію модулярного множення на постійне число, теоретично не здатні прискорити обчислення модулярної експоненти більше ніж в 1.5 рази. Перехід через цю межу можливий лише за умови прискорення модулярного піднесення до квадрату. Підхід, до вирішення цієї задачі запропоновано в методі [10], сутність якого полягає в прискоренні обчислення модулярного квадрату за рахунок динамічної зміни довжини операндів. Це надає змогу практично вдвоє прискорити виконання цієї операційної компоненти модулярного експоненціювання. Джерелом прискорення обчислення модулярного квадрату в згаданій роботи виступає притаманна піднесенню до квадрату операційна надлишковість. Інший підхід, який комбінує секційну обробку співмножників, а також схеми скороченого множення запропоновано в роботі [5]. Доведено, що час модулярного піднесення до квадрату може бути скорочений в 2.67 раз.

Таким чином, проведений огляд існуючих методів прискорення мультиплікативних операцій модулярної арифметики .дозволяє зробити висновок про те, що для суттєвого прискорення важливої для криптографічних застосувань операції модулярного експоненціювання необхідно комплексно пришвидшувати реалізацію обох базових її компонентів – модулярних операцій піднесення до квадрату та множення.

Мета досліджень

Мета досліджень полягає в прискоренні основної операції криптографії з відкритим ключем – модулярного експоненціювання при її реалізації на терміналах мікроконтролерах систем дистанційного управління на базі технологій IoT, за рахунок скорочення часу реалізації обох її базових компонентів – модулярних операцій піднесення до квадрату та множення, шляхом застосування передобчислень.

Метод прискорення обчислення модулярного добутку

Як зазначалося вище, операція модулярного множення $A \cdot B \bmod M$, як і модулярне піднесення до квадрату, складається з двох фаз: власне множення, тобто обчислення $A \cdot B$ та модулярної

редукції, тобто віднаходження залишку від ділення $A \cdot B$ на модуль M. При суміщеній реалізації цих двох фаз по методу Монтгомері, в кожному із n циклів здійснюється, з ймовірністю 0,5, одна операція додавання множимого A та одна операція додавання модулярної корекції (додавання модулю M), а також завжди виконується зсув праворуч коду проміжного результату. Один з підходів до прискорення модулярного множення полягає в групуванні вказаних операцій з використанням передобчислень. При цьому, необхідною умовою для застосування передобчислень при групуванні операцій додавання множимого A виступає сталість коду A при виконанні всіх циклів множення. Аналогічно, для групування операцій модулярної корекції виступає сталість модуля M, що на практиці завжди виконується.

Об'єм передобчислень для групування k операцій додавання пропорційний 2^k. Це обмежує кількість k таких операцій, що об'єднуються в одну. При обчислені модулярної експоненти $A^E \bmod M$ передобчислення виконуються безпосередньо перед початком експоненціювання і, при цьому, його результати використовуються $0{,}5 \cdot n^2/k$ разів. Це дозволяє прискорити обчислення модулярного множення $A \cdot R \bmod M$ в k раз. Оскільки передобчислення виконуються лише один раз перед початком експоненціювання, то час його виконання мало впливає на швидкість реалізації цієї операцій. Це значною мірою знімає обмеження на величину k і, тим самим, відкриває значні можливості для прискорення обчислення модулярного добутку $A \cdot B \bmod M$. Разом з тим, така організація передобчислень дозволяє прискорити лише операцію модулярного множення $A \cdot R \bmod M$, які складають лише третину від загальної кількості мультиплікативних операцій. Це означає, що така організація передобчислень теоретично не здатна забезпечити прискорення обчислення модулярної експоненти більше ніж у 1,5 рази.

Очевидно, що для того, щоб передобчислення забезпечували більшу швидкодію обчислення модулярної експоненти, вони мають прискорювати не тільки множення, але й піднесення до квадрату: $R^2 \bmod M$. В силу того, що код R змінюється в кожному циклі експоненціювання, очевидним є те, що передобчислення потрібно робити на початку кожного циклу обчисленя модулярної експоненти.

В алгоритмі експоненціювання зі старших розрядів коду експоненти в кожному циклі виконуються такі дві мультиплікативні операції над довгими числами, що не мають однакових операндів. Тому виконання передобчислень в межах циклу для цього різновиду алгоритму модулярного експоненціювання менш ефективне.

В різновиді класичного алгоритму обчислення модулярної експоненти з молодших розрядів, в межах одного циклу виконується множення поточного значення R на поточний код D: $R=R \cdot A \bmod M$, а також обчислення $D=D^2 \bmod M$. Тобто в обох операціях в якості множимого виступає одне і те ж саме число - поточне значення D. Це створює передумови для ефективного використання передобчислень, пов'язаних саме з цим числом. Відповідно, результати цих обчислень можуть бути ефективно використані як для прискорення виконання модулярного множення $R=R \cdot D \bmod M$, і модулярного піднесення до квадрату $D=D \cdot D \bmod M$.

Розроблений метод прискореного модулярного множення передбачає етап формування таблиці передобчислень. Ця таблиця будується на основі чисел A, модуля M та складається з k зон. Перший елемент кожної j-тої ,$j \in \{0,1,\ldots,k\text{-}1\}$ зони містить значення $A \cdot 2^{k-1}$, другий елемент - $M \cdot 2^{k-1}$ та третій - $(A + M) \cdot 2^{k-1}$. Загальний об'єм таблиці складає $k \cdot 3$ n-розрядних чисел.

Процедура побудови таблиці передбачає формування значень таблиці по трійкам, тобто організовану у вигляді $k/3$ циклів. При цьому другі рядки кожної із трійок, які містять значення M, $2 \cdot M$,

$4\cdot M,\ldots,2^{k-1}\cdot M$ фактично не змінюються в силу того, що залежать від постійного значення модулю M, який є частиною відкритого ключа криптосистеми. Це означає, що побудова таблиці передобчислень зводиться до рекурсивної зміни лише першого та третього в кожній із трійок рядків. При цьому перший рядок кожної наступної трійки, починаючи з другої, являє собою зсунуте значення першого рядка попередньої трійки (значення першого рядка першої трійки встановлюється рівним коду поточного результату *R,* отриманому на попередньому циклі експоненціювання). Третій рядок трійки в кожному циклі формування таблиці формується як результат додавання другого та першого рядків цієї трійки або зсуву третього рядка попередньої трійки. Таким чином, в кожному з $k/3$ циклів виконується дві операції над довгими числами. Виходячи з цього, час t_{pc} передобчислень визначається за наступною формулою:

$$t_{pc} = \frac{2}{3}\cdot k \cdot t_a, \qquad (1)$$

де t_a – час виконання операції зсуву чи додавання n-розрядних довгих чисел.

Створення таблиці передобчислення може бути продемонстровано наступним прикладом.

Нехай значення k = 3, значення A = 2678, значення модулю M = 3011, відповідно значення $A + M$ = 5689. При цих значеннях таблиця передобчислень має вигляд представлений в таблиці 1.

Таблиця 1. Приклад таблиці передобчислень для k=3, значення A=2678 та значення модуля M=3011.

j	$T[j]$	
	Формула	Значення
0	A	2678
1	M	3011
2	$A+M$	5689
3	$2\cdot A$	5356
4	$2\cdot M$	6022
5	$2\cdot(A+M)$	11378
6	$4\cdot A$	10712
7	$4\cdot M$	12044
8	$4\cdot(A+M)$	22756
9	$8\cdot A$	21424
10	$8\cdot M$	24088
11	$8\cdot(A+M)$	45512

Запропонований метод передбачає виконання процедури модулярного множення $A \cdot B \bmod M$ з використанням створеної таблиці T передобчислень у вигляді наступної послідовності дій:

1. Початкове значення змінної R поточного результату R встановлюється в нуль: $R = 0$, так само, як значення номеру j поточного біту множника j: $j = 0$, а також індекс f номера біту в групі f: $f = 0$.
2. Якщо значення r_f поточного біту коду $R = r_0 + r_1\cdot 2+\ldots+r_{n-1}\cdot 2^{n-1}, \forall i\in(0,1,\ldots,n\text{-}1\}$: $r_i\in\{0,1\}$ дорівнює нулю, тобто $r_f = 0$, то до поточного значення результату R додається добуток модуля M на 2^f, тобто $R = R + T[f \cdot 3+1]$. Здійснюється перехід на виконання п. 7 процедури.
3. Якщо поточний біт b_j множника $B = b_0 + b_1 \cdot 2 + b_1 \cdot 4 + \ldots + b^{n-1} \cdot 2^{n-1}$ дорівнює одиниці, тобто при b_j=1, і при цьому, молодший біт a_0 множимого A дорівнює поточному біту r_f результату R: $r_f = a_0$, до результату R додається добуток $A\cdot 2^f$: $R = R + T[f \cdot 3]$. Здійснюється перехід на виконання п. 7.
4. Якщо поточний біт b_j множника B дорівнює одиниці, але при цьому молодший біт a_0 множника A має значення одиниця, а поточний біт r_f результату R дорівнює нулю, до R додається сума множника A та модуля M, помножена на 2^f: $R=R+T[f\cdot 3+2]$. Здійснюється перехід на виконання п. 7 процедури.
5. Якщо значення поточного біту b_j множника B дорівнює нулю: b_j=0, значення поточного біту r_f результату R рівне одиниці: r_f=1, до результату R додається добуток модуля M на 2^f: $R=R + T[f \cdot 3+1]$. Здійснюється перехід на виконання п. 7.
6. Якщо поточний біт b_j множника B дорівнює одиниці, молодший біт a_0 множника A дорівнює нулю, а поточне значення біту r_f результату R становить одиницю, також до результату R додається зсунута на f розрядів ліворуч $A+M$: $R = R+T[f \cdot 3+2]$.
7. Індекс біта f в групі f збільшується на одиницю: $f=f+1$; якщо $f=k$, то здійснюється зсув поточного результату R на k розрядів праворуч: $R=R>>k$. Після цього значення індексу біту в групі f встановлюється рівним нулю: f=0.

8. Значення j номеру ітерації збільшується на одиницю: $j=j+1$, якщо $j<n$, то здійснюється повернення на повторне виконання п.2 процедури.

9. В змінній R сформовано код: $R=A\cdot B\cdot Y \bmod M$, де Y – мультиплікативна інверсія $2^n \bmod M$.

Робота описаної процедури модулярного множення з використанням таблиці передобчислень може бути проілюстровано наступним прикладом. Нехай здійснюється обчислення модулярного добутку $A\cdot B \bmod M$, для якого значення множимого A=2678, значення множника B=2845 та модуль M=3011. Правильне чисельне значення результату 2678·2845 mod 3011 дорівнює 1080. Виконання обчислення запропонованої процедури здійснюється в наступному порядку:

В рамках виконання п.1 процедури початкове значення змінної R встановлюється в нуль: R=0, так само, як значення номеру j поточного біту множника j: j=0, а також індекс f номера біту в групі f: f=0.

З умов 2-6 виконується лише пункт 3: оскільки поточний біт b_j множника B дорівнює один: b_j=1, а поточний біт r_0 результату R дорівнює молодшому біту a_0 множимого A: $r_0=a_0$=0. Відповідно до пункту 3, до поточного результату R додається значення з таблиці T[0]: $R=R+T[0]$=0+2678=2678.

Виконується перехід на п.7, в якому значення f збільшується на одиницю: $f=f+1$, і стає рівним 1: f=1. Наступним п.8 значення j збільшується на одиницю: $j=j+1$, і стає рівним 1: j=1. Оскільки $j<M$, то здійснюється повернення на повторну перевірку умов 2-6.

При виконанні п. 5, в якому значення поточного біту r_f результату R рівне одиниці: r_f=1, а значення поточного біту b_j множника B – нулю: b_j=0, до результату R додається значення з таблиці T[4]: $R=R+T[4]$=2678+6022=8700. Після цього виконується п.7 та п.8: $f=f+1$, $j=j+1$. Подальший хід обчислення запропонованої процедури продемонстровано за допомогою наступної таблиці:

Таблиця 2. Динаміка покрокової трансформації змінних запропонованої процедури прискореного модулярного множення при виконанні 2678·2845 mod 3011.

j	f	b_j	a_0	r_f	R
0	1	1	0	0	$R=R+T[0]$=0+2678=2678
1	2	0	0	1	$R=R+T[4]$=2678+6022=8700
2	3	1	0	1	$R=R+T[8]$=8700+22756=31456
3	4	1	0	0	$R=R+T[9]$=31456+21424=52880
4	1	1	0	1	$R=R+T[2]$=3305+5689=8994
5	2	0	0	1	$R=R+T[4]$=8994+6022=15016
6	3	0	0	0	-
7	4	0	0	1	$R=R+T[10]$=15016+24088=39104
8	1	1	0	0	$R=R+T[0]$=2444+2678=5122
9	2	1	0	1	$R=R+T[5]$=5122+11378=16500
10	3	0	0	1	$R=R+T[7]$=16500+12044=28544
11	4	1	0	0	$R=R+T[9]$=28544+21424=49968

Отриманий результат R=3123 являє модулярний добуток трьох співмножників: $A\cdot B\cdot 2^{-n} \bmod M$, де 2^{-n} - модулярна інверсія числа $2^n \bmod M$. Для отримання правильного результату R' потрібно обчислене в результаті виконання процедури значення R=3123 домножити на 2^n: $R'=R\cdot 2^n \bmod M$=3123·212 mod 3123 =1080.

Таким чином, розроблена процедура прискореного множення забезпечує отримання правильного результату модулярного множення. На практиці корекція після кожного модулярного множення чи піднесення до квадрату не виконується : процедура експоненціювання Монтгомері [7] потребує здійснення такої корекції лише один раз, після здійснення всіх n циклів експоненціювання.

Оцінка ефективності

Виходячи з поставленої мети – прискорення базової операції криптографії з відкритим ключем - модулярного експоненціювання, основним критерієм ефективності обрано коефіцієнт β прискорення, який визначається як відношення часу T_0 виконання операції експоненціювання за класичним алгоритмом до часу T цієї операції за запропонованим методом:

$$\beta = \frac{T_0}{T}. \qquad (2)$$

Час T_0 обчислення модулярної експоненти класичним алгоритмом визначається часом виконання $1{,}5 \cdot n$ мультиплікативних операцій над n-розрядними числами. В свою чергу, час виконання кожної мультиплікативної операції, при використанні редукції Монтгомері, визначається часом виконання $2 \cdot n$ адитивних операцій над довгими числами. Відповідно, час виконання модулярного експоненціювання T_0 визначається наступною формулою:

$$T_0 = 3 \cdot n^2 \cdot t_a, \qquad (3)$$

Час T_m виконання операції модулярного множення запропонованим методом визначається наступним чином. Метод передбачає здійснення модулярного множення у вигляді n циклів, в кожному з яких з ймовірністю 0.75 виконується операція додавання до коду поточного результату табличного значення. Крім того, в кожному k-тому циклі, тобто всього n/k разів, виконується операція зсуву коду поточного результату. Тобто, чисельне значення T_m визначається за формулою:

$$T_m = 0.75 \cdot n \cdot t_a + \frac{n}{k} \cdot t_{sh}, \qquad (4)$$

де t_{sh} – час виконання зсуву довгого числа. Для сучасних мікроконтролерів час t_a виконання операцій арифметичного додавання та зсуву t_{sh} приблизно однакові, відповідно формула (4) може бути трансформована до наступного виду:

$$T_m = n \cdot t_a \cdot (0.75 + \frac{1}{k}). \qquad (5)$$

При виконанні модулярного експоненціювання з застосуванням запропонованого методу здійснюється n циклів, в кожному з яких реалізується спочатку передобчислення для вхідного для циклу значення поточного результату Q. Потім, в циклі з застосуванням результатів передобчислень виконується зі ймовірністю 0.5 (в залежності від значення поточного біту коду експоненти) модулярне множення $R=R \cdot Q \bmod M$, операція піднесення $Q=Q^2 \bmod M$. Тобто, в середньому, в рамках одного циклу здійснюється, крім передобчислень, півтори мультиплікативні операції з використанням передобчислень. Відповідно, час T модулярного експоненціювання з використанням запропонованого методу визначається за наступною формулою:

$$T = n \cdot (t_{pc} + 1.5 \cdot T_m). \qquad (6)$$

Підстановкою значення t_{pc}, що визначається формулою (1), а також, виразу (5) для T_m в формулу (6) отримується наступний вираз для часу T експоненціювання за запропонованим методом:

$$T = n \cdot t_a \cdot (\frac{2 \cdot k}{3} + n \cdot \frac{3}{2}(\frac{3}{4} + \frac{1}{k})) = \\ = n^2 \cdot t_a \cdot (\frac{2 \cdot k}{3 \cdot n} + \frac{9}{8} + \frac{3}{2 \cdot k}) \qquad . (7)$$

На практиці значення k становить величину в діапазоні від 8 до 32-х, при цьому $k<<n$. Виходячи з цього, значення часу T модулярного експоненціювання за запропонованим методом може бути представлено (для значення k=8) у наступному вигляді: $T = 1.31 \cdot n^2 \cdot t_a$. При більших значеннях k коефіцієнт при $n^2 \cdot t_a$ незначно знижується, але не нижче 1.125 (наприклад, для $k = 16$, $T = 1.22 \cdot n^2 \cdot t_a$).

Відповідно, чисельне значення коефіцієнту β прискорення комп'ютерної реалізації модулярного експоненціювання за рахунок використання запропонованого методу можу бути обчислене за наступною формулою:

$$\beta = \frac{T_0}{T} = \frac{3 \cdot n^2 \cdot t_a}{1.31 \cdot n^2 \cdot t_a} = 2.3. \qquad (8)$$

Таким чином, наведені теоретичні обрахунки показали, що запропонований метод обчислення базової операції криптографії з відкритим ключем – модулярного експоненціювання дозволяє прискорити її комп'ютерну реалізацію більш ніж в два рази. Проведені експериментальні дослідження робити програмного коду, який реалізує запропонований метод прискореного обчислення модулярної експоненти

підтвердили зроблені теоретичні обрахунки. Суттєва перевага запропонованого методу в порівнянні з іншими відомими методами прискорення модулярного множення за рахунок застосування передобчислень полягає в тому, що його реалізація потребує відносно невеликий об’єм пам'яті для зберігання таблиці передобчислень: $3 \cdot k \cdot n$ бітів, який лінійно залежить як від значення n так і від k. Так, при типових для практики значеннях k=8 та n=4096 розроблений метод потребує всього 12 Кбайт пам'яті для зберігання таблиці передобчислень.

Подальший розвиток запропонованого методу вбачається в суміщенні обробки g розрядів коду множника. Це матиме наслідком зростання потрібного об’єму пам'яті в $2g$ раз, але забезпечує прискорення обчислень в g раз. Зокрема, при суміщен0ні обробки двох розрядів множника (g=2), об’єм таблиць передобчислень зростає в чотири рази, проте загальне значення коефіцієнту прискорення підвищиться вдвоє, тобто буде близьким до п'яти.

Висновки

Проведені дослідження, направлені на прискорення комп’ютерної реалізації базової операції механізмів захисту даних на основі криптографії з відкритим ключем – модулярного експоненціювання дозволили отримати наступні результати.

Теоретично обґрунтовано, розроблено та досліджено метод прискорення обчислення модулярної експоненти, який відрізняється тим, що при виконанні модулярного множення суміщується додавання до суми часткових добутків зсунутого множимого та коду модулярної корекції за рахунок передобчислень, які здійснюються на кожному циклі модулярного експоненціювання, що реалізується скануванням коду показника в напрямку від молодших його розрядів до старших зі зсувом суми відразу на k розрядів, за рахунок чого досягається прискорення обчислення модулярної експоненти в чотири рази в порівнянні з реалізацією за класичним алгоритмом.

Проведені експериментальні дослідження з застосуванням спеціально розроблених програмних засобів, в цілому, підтвердили чотирьохкратне прискорення роботи механізмів інформаційної безпеки на основі криптографії з відкритим ключем, реалізованих з використанням запропонованого методу.

Розроблений метод орієнтовано для застосування при реалізації цифрового підпису даних в системах дистанційного контролю та управління, побудованих на основі технології IoT, для протидіям спробам зовнішнього втручання в їх функціонування.

Література

1. Nimodiya A. A Review on Internet of Things / A. Nimodiya, S.S. Ajankar // International Journal of Advnced Research in Science Communication and Technology.- Vol.2.-Issue 1.-2022.-P. 135-144. DOI: 10.48175/IJARSCT-2251.

2. Elgazzar Khalid. Revisiting the internet of things: New trends, opportunities and grand challenges./ Elgazzar Khalid, Haytham Khalil, Taghreed Alghamdi, Ahmed Badr, Ghadeer Abdelkader Abdelrahman Elewah, Rajkumar Buyya // Frontiers in Internet of Things.-2022.-Vol.1.- P.1-18. DOI:10.3389 /friot2022. 1073780.

3. Schneier B. Appllied Cryptography. Protocols, Algorithms and Source Code in C . Wiley.-2015.-P.784.

4. Fùrer M. Faste Integer Multiplication / M. Fùrer //SIAM Journal on Computing. Vol. 39.- № 3.-2009. p.979-1005. DOI.ORG/10. 1137/070711716.

5. Марковский О.П. Метод прискореного модулярного множення для ефективної реалізації механізмів криптографічного захисту з відкритим ключом / О.П. Марковський, Аль-Мріят Гассан Абдель Жаліль // Адаптивні системи автоматичного управління.- 2024.- Том.1 - № 44.- С.142-152. DOI: 10.20535/1560-8956. 44. 2024. 302429.

6. 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.

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

8. Bos J.W. Montgomery multiplication using vector instructions. In: Selected Areas in Cryptography — SAC, August 14–16, 2013, pp. 471–489 (2013). DOI: 10.1007/978-3-662-43414-724

9. Марковський О.П.,. Метод прискорення модулярного множення для механізмів криптографічного захисту з відкритим ключем / О.П. Марковський,

Аль-Мраят Гассан Абдель Жаліль // Проблеми управління та інформатизації.- 2023.- № 4 (76).- С.48-58. DOI: 10.18372/2073-4751. 76. 18240

10. Гуцуляк Н.А. Модулярне множення на постійне число з суміщенням групової обробки розрядів множника та редукції Монтгомері /Н.А. Гуцуляк, В.Л. Селіванов, В.Л. Володін // Проблеми управління та інформатизації.- 2025.- № 1 (81).- С.95-104. DOI:10.18372/2073-4751.81. 20135.

11. Markovskyi O., Borges J., Serhiichuk N. and Bardis N. Method for Power Analysis-Proof Implementation of Modular Exponentiation on IoT Terminal Microcontrollers /O. Markovskyi, Borges J., N.Serhiichuk, N. Bardis N. // 14th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2024, - P. 48-53, DOI: 10.1109/DESSERT65323.2024.11122248.

12. Markovskyi O. An Accelerate Approach for Public Key Cryptography Implemen-tation on IoT Terminal Platforms / O. Markovskyi, Al-Mrayat Ghassan Abdel Jalil Halil, N. Doukas, N.Bardis // In 13-th International Conference on Dependable system, Service and Technologies DESSERT-2023, 13-15 October, Greece, Athens. -2023.- P.62-67. DOI 10.1109/ DESSERT61349. 2023.10416516.

13. Марковський О.П.Метод прискорення модулярного піднесення до квадрату довгих чисел для криптографічних застосувань / О.П. Марковський , Аль-Мріят Гассан Абдель Жаліль // Проблеми управління та інформатизації.- 2024.- № 1 (77).- С.68-79. DOI: 10.18372/2073-4751.77.18659

Русанова О. В., Аль-Мраят Нізар Гассан Абдель Жаліль, Джура А. А., Аль-Мраят Гассан Абдель Жаліль

ПІДХІД ДО ПРИСКОРЕННЯ КОМП'ЮТЕРНОЇ РЕАЛІЗАЦІЇ МОДУЛЯРНОГО ЕКСПОНЕНЦІЮВАННЯ ДОВГИХ ЧИСЕЛ

В статті запропоновано метод прискорення базової операції криптографії з відкритим ключем – модулярного експоненціювання при її реалізації на термінальних компютерних платформх систем дистанційного управління обєктами реального світу. Зменшення часу екпоненціювання досягається шляхом прискорення операцій модулярного множення за рахунок використання передобчислень, які дозволяють сумістити додавання до суми часткових добутків зсунутого множимого та коду модулярної корекції. Відмінність запропонованого методу полягає в тому. що передобчислення виконуються в кожному циклі експоненціювання і використовуються для прискорення як модулярного піднесення до квадрату, так і множення.

Теоретично доведено і експериментально підтверджено, що запропонований метод дозволяє прискорити обчислення модулярної експоненти більш ніж вдвічі.

***Ключові слова**: модулярне множення, криптографія з відкритим ключем, модулярна редукція Монтгомері, модулярне експоненціювання, передобчислення.*

Rusanova O. V., Al-Mrayat Nezar Ghassan Abdel Jalil, Dzhura A. A., Al-Mrayat Ghassan Abdel Jalil

APPROACH TO ACCELERATE COMPUTER REALIZATION OF MODULAR EXPONENTIATION OF LONG NUMBERS

The article proposes an accelerating method of base cryptography with open key operation - modular exponentiation while being realized on terminal computer platforms of remote controlled systems for real-worlds objects. Reducing the time of exponentiation can be achieved by accelerating modular multiplying due to use of pre-computations that allow combining additions to a sum of partial products of shifted multiplicand and a modular correction code. The proposed method differs in that the pre-computations are being executed in every cycle of exponentiation and being used for accelerating both modular squaring and multiplying operations.

It has been theoretically showed and experimentally proven that proposed method allows an acceleration of calculating the modular exponent by more than twice.

***Keywords:** modular multiplication, public key cryptography, modular Montgomery reduction, modular exponentiation, precomputations.*

Стаття подана до редакції: 11*/03/2026*
Стаття прийнята до опублікування:18/03/2026
Стаття опублікована: 30/05/2026