УДК 004.056.5

СТІРЕНКО С. Г.,
МАРКОВСЬКИЙ О. П.,
ЗАХАРІУДАКІС ЛЕФТЕРИС .,
МІЩЕНКО Л. Д.

СПОСІБ ПРИСКОРЕНОГО ОБЧИСЛЕННЯ МОДУЛЯРНОЇ ЕКСПОНЕНТИ

В статті запропонована організація паралельного виконання модулярного експоненціювання. Доведено, що на рівні операцій модулярного множення двопотоковий паралелізм є найбільш доцільною формою паралельного виконання модулярного експоненціювання. Наведено математичне обґрунтування запропоновано підходу. Запропонована процедура паралельного обчислення модулярної експоненти детально викладена та ілюстрована чисельним прикладом. Виконано порівняльний аналіз продуктивності запропонованого методу обчислення модулярної експоненти Теоретично та експериментально доведено, що запропонований метод забезпечує прискорення обчислення модулярної експоненти приблизно вдвоє.

In article the organization of modular exponentiation parallel executing are presented. It has been shown that on modular multiplication level the two stream parallelism is best suited for parallel modular exponent calculation. The mathematical background of the proposed approach is presented. The proposed procedure of parallel modular exponent calculation are described in details and illustrated by numerical example. Performed comparative analysis of the proposed methods of modular exponent calculation has been executed. By the theoretical and experimental ways it is proved that the proposed method provides an acceleration of modular exponentiation by approximately two times.

Ключові слова: комп'ютерна арифметика, паралельні обчислення, модулярне множення, модулярне експоненціювання, мережові протоколи захисту даних.

Вступ

Визначальним напрямком розвитку інформаційних технологій є інформаційна інтеграція на основі комп'ютерних мереж. Ефективне використання таких технологій потребує надійного захисту даних та розділення прав доступу в інтегрованому інформаційному середовищі.

На сьогоднішній день такий захист та розподілення прав доступу забезпечується застосування ряду спеціальних протоколів мережевого обміну, в основі більшості з яких лежить криптографія відкритих ключів. Базовою обчислювальною операцією цих криптографічних перетворень є модулярне експонентціювання, тобто обчислення $A^E \bmod M$. Розрядність чисел, що регламентується існуючими протоколами значно більша за розрядність процесора і становить 2048 або 4096 [1]. Фактично швидкість виконання цієї операції визначає час реалізації протоколів криптографічного захисту інформації в мережах.

В останні роки широкого розповсюдження набуває використання хмарних технологій, які надають на комерційній основі кожному користувачеві значні за обсягом обчислювальні ресурси. Потенційно ці ресурси можуть бути використані зловмисниками для порушення існуючих протоколів захисту інформації в мережах. Це потребує адекватного підвищення рівня захищеності протоколів, в першу чергу, за рахунок збільшення розрядності чисел [2].

Проте, зі збільшенням розрядності чисел експоненційно зростає кількість операцій модулярної арифметики. При чому це зростання значно випереджає збільшості швидкості роботи процесорів.

Таким чином, підвищення ефективності операцій модулярного експоненціювання при реалізації мережових протоколів захисту інформації, являє собою актуальну проблему сучасного етапу розвитку інформаційних технологій.

Аналіз існуючих методів модулярного експоненціювання

Обчислювальну складність алгоритмів модулярного експоненціювання при їх реалізації на різних обчислювальних платформах зазвичай оцінють кількістю використаних в них опреацій процесорного множення.

Процесорними називають цілочисельні операції, які виконуються над k-розрядними чис-

лами, довжина яких відповідає розрядності процесора. Процес модулярного експоненціювання зводиться до послідовного виконання $\log_2 E = n$ циклів, у кожному із яких виконується операція піднесення до квадрату результату операції попереднього циклу й залежно від поточного біта степені E, виконується операція множення. Залежно від порядку, в якому аналізуються розряди степені E можна розглянути 2 типи алгоритмів експоненціювання [3]:

1) алгоритми, які передбачають аналіз розрядів степені E, починаючи зі старших розрядів (зліва-направо). У нотаціях мови С++ алгоритм цього типу може бути представлений у вигляді:

1. $R = 1$.
2. for (j=n-1; j >=0; j --)
{
2.1. $R = R \cdot R \bmod M$
2.2. if (ej == 1)
$R = R \cdot A \bmod M$
}
3. Результат: R.

При цьому, під час кожної ітерації циклу виконується модулярне піднесення числа в квадрат і множення на постійне число, рівне A, що створює потенційні передумови для підвищення швидкості множення. Недоліком є те, що оперції виконуються строго послідовно й належать критичному шляху. Це не дозволяє реалізувати паралельне обчислення операцій.

2) Алгоритми, які передбачають аналіз розрядів степені E починаючи із молодших (справаналіво). З використанням мови С++, алгоритм модулярного експоненціювання цього типу може бути представлений у вигляді:

1. $R = 1$, $Q = 1$.
2. for (j=0; j <n; j ++)
{
2.1. $R = R \cdot R \bmod M$
2.2. if (e_j == 1)
$Q = Q \cdot R$
}
3. Результат: Q.

При такій реалізації алгоритму є потерціальна можливість розпаралелити модулярне експоненціювання.

Аналіз вказаного алгоритму показує, що базовими операціями виконання модулярного експоненціювання є модулярне піднесення до квадрату і модулярне множення на фіксоване число, час виконання яких фактично визначається продуктивністю обчислення $A^E \bmod M$.

Більшість алгоритмів модулярного експоненціювання для реалізації згаданих двох операцій використовують єдину операцію модулярного множення. У свою чергу, час виконання модулярного множення визначається двома складовими: часом, що необідний для реалізації власне множення і часом, який витрачається на модулярну редукцію. У класичному множенні модулярна редукція реалізується з використанням операції ділення і, відповідно, друга складова відіграє значну роль. Значна ефективність обчислювальної реалізації модулярного множення досягається при використанні алгоритму Монтгомері, в якому модулярна редукція зводиться до здвигу на k розрядів. Тому, на практиці, при виконанні модулярного експоненціювання у більшості використовується алгоритм Монтгомері. Позначимо як Mont(A,B) - множення Монтгомері, яке формує результат $R = A \cdot B \cdot U \bmod M$, де U - модулярна інверсія числа 2^n по модулю M, тобто $U = (2^n)^{-1} \bmod M$.

Алгоритм модулярного експоненціювання Монтгомері можна представити з використання введених нотацій наступним чином:

1. $\tilde{x} = Mont(X, U^2 \bmod M) =$
$X \cdot U \bmod M$, $A = U \bmod M$
2. for ($j = n$; j >=0; j --)
{
2.1. A = Mont(A,A);
2.2. if ($e_j = 1$) $A \leftarrow Mont(A, \tilde{x})$
}
3. $A \leftarrow$Mont(A,1).

На сьогодняшній день розроблено ряд методів прискореної реалізації модулярного експоненціювання [4,5], які реалізують наступні можливості прискорення обчислень:

- зменшення кількості циклів при виконанні модулярного експоненціювання за рахунок раціональної організації обчислень з використанням адитивних ланцюжків;
- зниження часу виконання операції модулярного множення за рахунок обробки деяких суміжних розрядів множника;
- прискорення піднесення до квадрату за рахунок виключення збиткових операцій процессорного множення;
- зменшення обчислювальної складності модулярного експоненуіювання за рахунок спрощень, які враховують специфічні особливості практичного викростання цієї операції в алгоритмах захисту інформації, що використовуються в сучасних мережевих технологіях.

Мета досліджень полягає в прискоренні виконання критичної для протоколів захисту інформації операції модулярного експоненціювання за рахунок розпаралелювання обчислювального процесу.

Організація паралельного виконання модулярного експоненціювання

Операція модулярного експоненціювання складається з сукупності модулярних множень. В свою чергу, кожна із операцій модулярного множення складається з операцій процесорного множення, додавання та модулярної редукції. Відповідно, можна розглядати декілька рівнів обчислювального паралелізму при реалізації модулярного експоненціювання. Попередній аналіз показує, що при реалізації модулярного експоненціювання на сучасних багатоядерних процесорних засобах найбільш ефективним є розпаралелювання на рівні операцій модулярного множення.

Для досягнення поставленої мети виділено операції, що лежать на критичному шляху процесу модулярного експоненціювання. Як зазначалося вище, процес обчислення може бути організовано двома способами: шляхом послідовного аналізу двійкових розрядів експоненти починаючи зі старших розрядів, або починаючи з молодших розрядів. Аналіз показав, що при першому з наведених способів на критичному шляху лежать всі операції модулярного множення. Відповідно, цей спосіб мало придатний для організації розпаралелювання обчислювального процесу. Більш ефективний в цьому плані спосіб обчислення модулярної експоненти при аналізі двійкових розрядів експоненти починаючи з молодших розрядів. При організації експоненціювання з молодших розрядів виявлено, що критичний шлях утворюють операції модулярного піднесення до квадрату, результати яких утворюють послідовність: $A^2 \bmod M$, $A^4 \bmod M$, $A^8 \bmod M$,…, , де n – розрядність експоненти E. Всі інші операції модулярного множення за своєю кількістю менші за тих, що лежать на критичному шляху. Це дозволяє зробити висновок, що найбільш доцільним є використання двох процесорів.

Таким чином, мінімальний час виконання модулярного експоненціювання над n-розрядними числами визначається часом реалізації n операцій модулярного піднесення до квадрату та одного модулярного множення.

Число E може бути представлено у наступному вигляді $E = e_0 + e_1 \cdot 2 + e_2 \cdot 2^2 + \ldots + e_{n-1} \cdot 2^{n-1}$, де $\forall i \in \{0, 1, \ldots, n\text{-}2\}$: $e_i \in \{0,1\}$, $e_{n-1} = 1$.

Ступінь E пропонується розділити на дві частини α і β: $E = \alpha + \beta$. Перша частина відповідає значенню $\alpha = 2^{n-1}$.Друга $\beta = \sum_{i=0}^{n-2} e_i \cdot 2^i$ s. На першому процесорі пропонується обчислю-вати $R = A^{\alpha} \bmod M$, а на другому – $Q = A^{\beta} \bmod M$. Об'єднання результатів виконується шляхом модулярного множення $B = R \cdot Q \bmod M$ на другому процесорі.

Для збереження проміжних даних на першому ядрі використовується змінна R, на другому – Q. Початкове значення R визнача-ється кодом числа A, стартове значення Q дорівнює 1, якщо $e_0 = 0$, і A,. Запропонована організація модулярного експоненціювання може бути представлена наступною послідовністю дій.

1) встановлюється індекс циклу i в 0 ($i = 0$) і присвоюється початкове значення змінним на обох процесорах. Початкове значення змінної R першого процесора дорівнює A. Якщо $e_0 = 1$, то початкове значення змінної Q другого процесора також дорівнює A, інакше Q = 1.

2) на першому процесорі виконується модулярне піднесення до квадрату коду R: $R = R^2 \bmod M$. Якщо e_1=1, то отриманий результат передається на другий процесор і фіксується в змінній P. Значення індексу збільшується на одиницю.

3) якщо $e_{i-1} = 1$, то на другому процесорі обраховується модулярне множення $Q = P \cdot Q \bmod M$. На першому процесорі виконується модулярне піднесення до квадрату змінної R: $R = R^2 \bmod M$. Якщо e_{i+1}=1 або i=n-1, то отрима-ний результат передається на другий процесор і фіксується в змінній P.

4) індекс циклу збільшується на одиницю: i=i+1. Якщо $i < n$, то перехід на повторне виконання п.3.

5) на другому процесорі обраховується модулярне множення $Q = P \cdot Q \bmod M$, що є результатом модулярного експоненціювання.

Запропонована організація модулярного експоненціювання з розпаралелюванням обчислень може бути ілюстрована наступним прикладом.

Нехай потрібно обчислити $47^{43} \bmod 55 = 38$, тобто A = 47, E = 43 та M = 55 являють собою 6-розрядні двійкові числа (n = 6).

Згідно з запропонованою організацією модулярного експоненціювання, час обчислення

визначається часом виконання n+1 операцій множення.

Код експоненти E для цього прикладу може бути розділений на дві складові: $E = 43_{10} = 101011_2 = \varepsilon_1 + \varepsilon_2$, де $\varepsilon_1 = 100000_2 = 32_{10}$, а $\varepsilon_2 = 1011_2 = 11_{10}$. На першому процесорі обчислюється $A^{\varepsilon_1} \bmod M = 47^{32} \bmod 55$, а на другому - $A^{\varepsilon 2} \bmod M = 47^{11} \bmod 55$. Динаміка процесу обчислень для наведеного прикладу показана на рис.1.

До початку операції індекс циклу i установлюється в 0 (i=0). Значенню змінної R першого процесора присвоюється A=47. У початкове значення змінної Q другого проце-сора записується A, що відповідає числу 47.

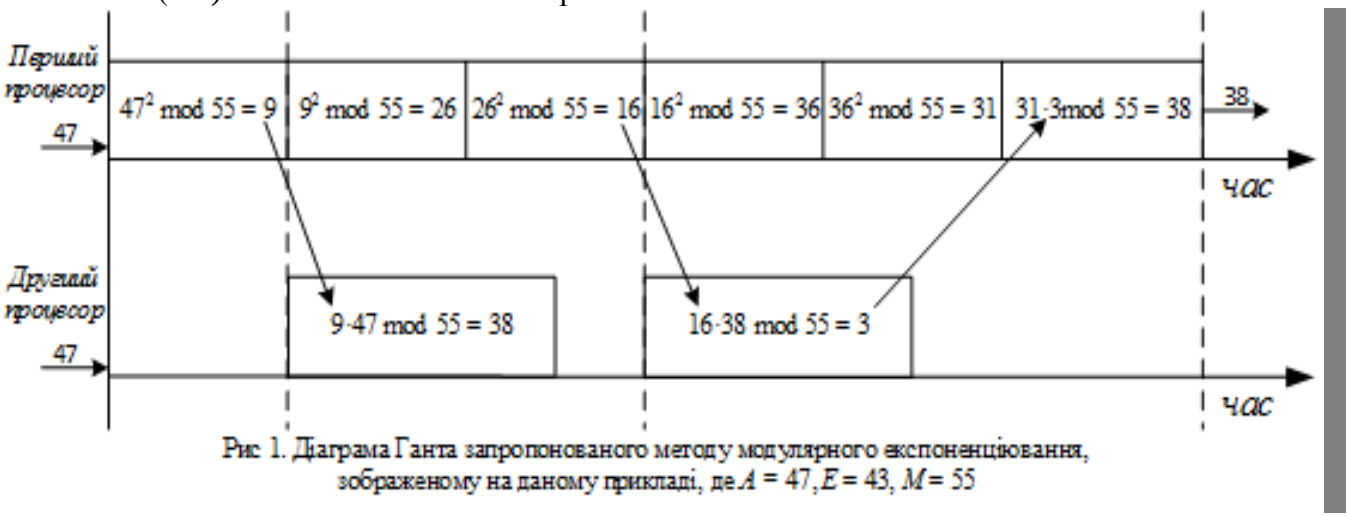

Рис 1. Діаграма Ганта запропонованого методу модулярного експоненціювання, зображеному на даному прикладі, де A = 47, E = 43, M = 55

На першому процесорі виконується модулярне піднесення до квадрату коду R, тобто обчислення $R = 47^2 \bmod 55 = 9$. На другому процесорі виконується операція модулярного множення $Q = Q \cdot P \bmod M = 47 \cdot 9 \bmod 55 = 38$. По закінченню операції модулярного піднесення до квадрату на першому процесорі, результат R передається на другий процесор і зберігається в змінній P.

Далі на першому процесорі виконується модулярне піднесення до квадрату коду R з отриманням результату $R = 9^2 \bmod 55 = 26$. На другому процесорі ніяких операцій не виконується. По закінченню операції модуляр-ного піднесення до квадрату на першому процесорі результат R передається на другий процесор і фіксується в змінній P.

На першому процесорі виконується модулярне піднесення до квадрату коду R з отриманням результату $R = 26^2 \bmod 55 = 16$. На другому процесорі виконується операція модулярного множення $Q = Q \cdot P \bmod M = 38 \cdot 16 \bmod 55 = 3$. По закінченню операції модулярного підне-сення до квадрату на першому процесорі, результат $R := P$ і передається на другий процесор.

Потім на першому процесорі виконується модулярне піднесення до квадрату коду R з отриманням результату $R = 16^2 \bmod 55 = 36$. На другому процесорі ніяких операцій не виконується.

На першому процесорі виконується модулярне піднесення до квадрату коду R з отриманням результату $R = 36^2 \bmod 55 = 31$. На другому процесорі в цей час ніяких операцій не виконується.

На перший процесор передається кінцеве значення змінної Q. На ньому виконується обчислення результату всіх обрахунків як модулярне множення коду R і Q: $R = R \cdot Q \bmod 55 = 31 \cdot 3 \bmod 55 = 38$.

Оцінка ефективності

Основною сферою практичного застосування модулярного експоненціювання є протоколи захисту інформації. Для цих застосувань розрядність чисел n становить 2048 або 4096 і значно більша за розрядність процесора r. Тому числа розділяються на m секцій, довжина яких визначається розрядністю процесора, тобто m дорівнює n/r. Базова операція модулярного множення n-розрядних чисел виконується як сукупність операцій множення секцій, з яких складаються числа.

Відповідно, для обчислення модулярного добутку двох чисел кожна секція першого числа перемножається із кожною секцією другого. Таким чином, загальна кількість операцій процесорного множення становить m^2.

При модулярному піднесенні до квадрату кількість операцій процесорного множення можна значно зменшити. За рахунок того, що

добуток i-тої секції першого числа на j-ту секцію другого дорівнює добутку i-тої секції другого числа на j-ту секцію першого. Загалька кількість операцій процесорних множень зменшується до $(m\text{-}1)\cdot m/2$.

При виконанні модулярного експоненціювання на процесорі виконується n операцій модулярного піднесення до квадрату та, у середньому, $n/2$ операцій модулярного множе-ння. Таким чином, загальна кількість N_1 операцій процесорного множення визначається формулою $n\cdot m^2 + n\cdot m/2$.

У пропонованому варіанті на першому процесорі виконується $N_{21}=n\cdot(m^2+m)/2$ операцій процесорного множення. На другому процесорі виконується $N_{22} = n\cdot m^2/2$ операцій процесорного множення. Оскільки $N_{21} > N_{22}$, то час виконання модулярного експоненціювання визначається часом обчислень N_{21} процесорних множень на першому процесорі $N_2 = N_{21}$.

Таким чином, запропонована організація модулярного експоненціювання дозволяє прискорити обчислення в ξ разів, причому значення ξ визначається формулою: $\xi = N_1/N_2 = (2\cdot m^2 + m) / (m^2 + m) = (2\cdot m + 1) / (m + 1) \approx 2$.

Наприклад, якщо розрядність чисел 2048, а процесора – 64, то $m = 32$ $\xi = 1.97$.

Проведені експериментальні дослідження на 32-розрядних процесорах в цілому підтвердили наведенні теоретичні викладки: експерименти показали, що практичне застосування запропонованого методу дозволило прискорити процес модулярного експоненціювання чисел розрядності 2048 в 1.82 рази.

Висновки

В результаті проведених досліджень, направлений на пошук шляхів прискорення виконання важливої для реалізації протоколів захисту інформації операції модулярного експоненціювання над числами, розрядність яких значно перевищує розрядність процесора можна зробити такі висновки.

Проведений аналіз обчислювальних процедур модулярного експоненціювання показав, прискорення їх виконання може бути досягнуто за рахунок розпаралелювання на різних рівнях. Якщо розглядати рівень операцій модулярного множення, як базової складової модулярного експоненціювання, то проведений аналіз показав, що рівень паралелізму не перевищує 2-х.

Для практичної реалізації цієї можливості запропонована організація прискореного обчислення модулярної експоненти на двоядерному процесорі. Теоретично та експериментально доведено, що розроблена організація забезпечує практично двократне прискорення виконання операції модулярного експоненціювання для розрядностей 2048 і 4096.

Більш значне прискорення виконання модулярного експоненціювання може бути досягнуто при переході на рівень операцій процесорного множення.

Список посилань

1. Самофалов К.Г. Ускоренная реализация модулярного экспоненцирования на малоразрядных микропроцессорах и встроенных микроконтроллерах / К.Г.Самофалов, Рамзи Анвар Салиба Сунна, Д.Ю. // Проблеми інформатизації та управління. Збірник наукових праць: Випуск 4(15).-К.,НАУ, 2005.- С.144-153.

2. Can Xiang. Verifiable and Secure Outsourcing Schemes of Modular Exponentiations Using One Untrusted Cloud Server and Their Application // IACR Cryptology ePrint Archive 2014: PP.500 .- https://eprint.iacr.org/2014/500.pdf

3. Markovskyi O.P. Secure Modular Exponentiation in Cloud Systems./ Oleksandr P. Markovskyi, Nikolaos Bardis, Nikolaos Doukas, Sergej Kirilenko // Proceedings of The Congress on Information Technology, Computational and Experimental Physics (CITCEP 2015), 18-20 December 2015, Krakow, Poland, C. 266-269.

4. Брей Б. Микропроцессоры Intel. Архитектура, программирование и интерфейсы. Пер.с англ.- СПб:БХВ-Петербург.-2014.- С.1328.

5. Марковський О.П. Спосіб прискореного обчислення модулярної експоненти / О.П.Марковський, Л.Д. Міщенко // Прикладна математика та комп'ютинг ПМК-2017. Збірник тез доповідей 9-ї наук. конференції магістрантів та аспірантів, Київ, 19-21 квіт.2017.- К.:Просвіта,2017 – С.200-203.