УДК 004.75

Міщенко Л. Д.,
Виноградов Ю. М.

МЕТОД ПАРАЛЕЛЬНОГО ОБЧИСЛЕННЯ МОДУЛЯРНОЇ ЕКСПОНЕНТИ З ВИКОРИСТАННЯМ ПЕРЕДОБЧИСЛЕНЬ

METHOD FOR PARALLEL MODULAR EXPONENTIATION BY USING PRECOMPUTATION

В статті запропонована організація паралельного виконання модулярного експоненціювання. Доведено, що на рівні операцій модулярного множення трьопотоковий 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 втроє.

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

In article the organization of modular exponentiation parallel executing are presented. It has been shown that on modular multiplication level the three 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 three times.

Key words: computer arithmetic, parallel calculation, modular multiplication, modular exponentiation, data security protocols.

Актуальність теми дослідження. З плином часузростає важливість забезпечення високої продуктивності реалізації термінальними пристроями комп'ютерних систем моніторингу та управління існуючих мережевих протоколів. Найбільш критичними з точки зору обчислювальної реалізації протоколів мережевого обміну є операції модулярної арифметики, що виконуються над числами великої розрядності, яка значно перевищує розрядність процесорів. На теперішній час, для досягнення прийнятного для більшості застосувань рівня захищеності, необхідна довжина чисел становить 1024-2048 розрядів з перспективою її зростання в найближчі роки до 4096.

Тенденція зростання пропускної здатності каналів передачі даних комп'ютерних мереж вимагає адекватного зростання швидкості реалізації протоколів захисту інформації як на комп'ютерах загального призначення, так і на малорозрядних мікроконтролерах.

Наведені фактори визначають розробку нових підходів до прискорення програмної реалізації операцій модулярної арифметики при їх застосування в протоколах захисту інформації, як важливу та актуальну проблему, від вирішення якої значною мірою залежить ефективність локальних та глобальних комп'ютерних мереж.

Постановка проблеми. Базовою обчислювальною операцією широкого кола мережевих протоколів захисту інформації є модулярне експоненціювання, тобто обчислення $A^E \bmod M$, де A, E і M – числа розрядністю n, значно більшою за розрядність процесора.

Процес модулярного експоненціювання зводиться до послідовного виконання n циклів, у кожному із яких виконується операція модулярного піднесення до квадрату результату операції попереднього циклу і, залежно від поточного біта степені E, здійснюється операція модулярного множення. Залежно від порядку, в якому організована обробка розрядів коду експоненти E можна існують два базових алгоритми експоненціювання [1]: зі старших та молодших розрядів коду експоненти.

Алгоритм обробки розрядів експоненти зі старших розрядів в нотаціях мови C++ алгоритм має такий вигляд:

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

Прицьому, підчас кожної ітерації циклу виконується модулярне піднесення числа вквадраті множення на постійне число, рівне A, що створює потенційні передумови для підвищення швидкості множення. Недоліком є те, що всі операції виконуються строго послідовно й лежать на критичному шляху [2].

Алгоритм, який передбачає обробку розрядів степені E починаючи із молодших розрядів в нотаціях мови C++ має вигляд:

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.

Аналіз останніх досліджень і публікацій. Аналіз обох базових алгоритмів показує, що час їх реалізації визначається сумою: $n \cdot t_{sq} + 0.5 \cdot n \cdot t_m$, де t_{sq} - час виконання операції модулярного піднесення до квадрату, а t_m – час виконання операції модулярного множення [3] .

В рамках другого з розглянутих алгоритмів модулярного експоненціювання існує потенційна можливість розпаралелювання обчислень. В роботі [4]

варіант такого розпаралелювання. Показано, що на рівні операцій модулярного множення максимальний рівень паралелізму не перевищує 2-х. Відповідно, прискорення обчислювання модулярної експоненти досягається за рахунок використання двох процесорів, один з яких реалізує потік операцій модулярного піднесення до квадрату, а інший – модулярного множення.

Проведений аналіз обчислювальних процедур модулярного експоненціювання показав, прискорення їх виконання може бути досягнуто за рахунок розпаралелювання на різних рівнях. Більшість робіт [5-7], присвячених вирішенню проблеми прискорення операції модулярного експоненціювання орієнтовані на рівень процесорних операцій, на які розкладаються операції модулярного піднесення до квадрату і модулярного множення. Зокрема, при експоненціювання 2048-розрядних чисел на 32-розрядних процесорів виконується 128 операцій процесорного множення, які можуть виконуватися паралельно [6] .

У свою чергу, час виконання модулярного множення визначається двома складовими: часом, необхідним для реалізації власне множення і часом, який витрачається на модулярну редукцію, тобто залишку від ділення результату множення на модуль М. 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 [8], в якому модулярна редукція зводиться до зсуву на k розрядів. Інший підхід до зменшення часу редукції запропоновано в роботі [3]. Цей підхід базується на тому, що на практиці, модуль, що є частиною відкритого ключа, практично не змінюється. Це надає змогу виділити операції, що залежать від модуля, обчислити їх результати один раз і використовувати при кожному модулярному експоненціюванні.

На сьогодняшній день розроблено ряд методів прискореної реалізації модулярного експоненціювання [8,9], які реалізують можливості розпаралелювання прискорення обчислення модулярної експоненти на рівні процесорних операцій.

Виділення недосліджених частин загальної проблеми. На основі досліджених літературних джерел можна зробити наступні висновки. Основним резервом підвищення швидкості обчислювальної реалізації базової операції мережевих протоколів захисту інформації є організація паралельної обробки. Проте цей підхід не може бути використаний для прискорення модулярного експоненціювання на малопотужних мікроконтролерах – термінальних пристроях широкого кола мереж, що використовуються на практиці. Таким чином, існуючі методи прискорення обчислення модулярної експоненти не вирішують цю проблему для вельми широкого класу обчислювальних пристроїв, які мають підтримувати протоколи мережевого захисту даних.

Постановка задачі. Мета досліджень полягає в прискоренні виконання критичної для протоколів мережового захисту інформації операції модулярного експоненціювання.

Метод паралельного обчислення модулярної експоненти з використанням передобчислень. Для досягнення поставленої мети, пропонується n-розрядний код експоненти $E = \{e_1,e_2,\ldots,e_n\}$, $\forall j \in \{0,1\}$, розділити на sm-розрядних фрагментів: $f=\{e_1,e_2,\ldots,e_m\},\ldots,f_s = \{e_{n-m}, \ldots, e_n\}$, очевидно, що $s=n/m$.

До безпосередніх обрахунків модулярної експоненти пропонується виконати $2^m - 2$ передобчислень значень $A^2 \bmod M$, $A^3 \bmod M$, …, $A^c \bmod M$, де $c = 2^{m-2}$ Передобчислення виконуються запропонованим деревовидним алгоритмом по заданому значенню A. На першому кроці обчислюється значення $A^2 \bmod M = A \cdot A \bmod M$. На наступному кроці паралельно на двох процесорах обраховуються значення $A^3 \bmod M = A^2 \bmod M \cdot A$ та $A^4 \bmod M = A^2 \bmod M \cdot A^2 \bmod M$. За третій крок одночасно обчислюються чотири значення: $A^5 \bmod M = A^3 \bmod M \cdot A^2 \bmod M$, $A^6 \bmod M = A^4 \bmod M \cdot A^2 \bmod M$, $A^7 \bmod M = A^4 \bmod M \cdot A^3 \bmod M$, $A^8 \bmod M = A^4 \bmod M \cdot A^4 \bmod M$. Таким чином, усі передобчислення виконуються за m кроків на 2^{m-1} процесорах. Результати передобчислень зберігаються в таблиці: $T[0] = 1$, $T[1] = A$, $T[2] = A^2 \bmod M$, … , $T[2^m - 1] = A^c \bmod M$.

Таким чином, теоретично, час передобчислень $T_{\text{тп}}$ визначається добутком $T_{\text{тп}} = m \cdot t_{\text{m}}$ де t_{m} – час виконання операції модулярного множення.

Безпосереднє обчислення модулярної експоненти пропонується виконувати на s-1 процесорах, кожен з яких оброблює відповідний фрагмент коду експоненти.

На j-тому процесорі, $j \in \{1,2,\ldots,s-1\}$, що оброблює j-ий фрагмент f_jкоду експоненти E пропонується виконувати:

- піднесення коду $T[f_j]$ до степені $2^{n-j.m}$;
- формування результату r_j обробки j-того фрагменту f_jшляхом модулярного множення отриманого коду на результат r_{j+1} обробки наступного фрагменту f_{j+1} коду експоненти: $r_j = r_{j+1} \cdot T[f_j]^{2^{n-j \cdot m}} \bmod M$. Для (s-1)-го фрагменту $r_{j+1} = r_s = T[f_s]$.

Запропонований метод розпаралелювання модулярної експоненти з виконанням передобчислень може бути показано наступним прикладом.

Нехай потрібно обчислити $48^{2542} \bmod 57 = 6$, тобто $A = 48$, $E = 2542$, $M = 57$.

До початку безпосередніх обрахунків виконуються передобчислення. Так як $E = 2542_{10} = 100111101110_2$, $n = 12$, запропоноване $m = 3$, тоді $s = n/m = 4$. Таблиця передобчислень заповнюється наступним чином.

Таблиця 1.

Таблиця передобчислень
для прикладу обчислення $48^{2542} \bmod 57 = 6$

f_j	$A^f \bmod M$	$T[f_j]$
2	$48^2 \bmod 57 = 24$	24
3	$48^3 \bmod 57 = 12$	12
4	$48^4 \bmod 57 = 6$	6
5	$48^5 \bmod 57 = 3$	3
6	$48^6 \bmod 57 = 30$	30
7	$48^7 \bmod 57 = 15$	15

Позначимо інтервал часу модулярного піднесення до квадрату як τ, тоді інтервал часу модулярного множення дорівнює $t_m = 2 \cdot \tau$.

Таблиця 2.

Часова діаграма обчислення прикладу $48^{2542} \bmod 57 = 6$.

τ	Процесори		
	1	2	3
1	$3^2 \bmod 57 = 9$	$15^2 \bmod 57 = 54$	$6^2 \bmod 57 = 36$
2	$9^2 \bmod 57 = 24$	$54^2 \bmod 57 = 9$	$36^2 \bmod 57 = 42$
3	$24^2 \bmod 57 = 6$	$9^2 \bmod 57 = 24$	$42^2 \bmod 57 = 54$
4	$30 \cdot 6 \bmod 57 = 9$	$24^2 \bmod 57 = 6$	$54^2 \bmod 57 = 9$
5		$6^2 \bmod 57 = 36$	$9^2 \bmod 57 = 24$
6		$36^2 \bmod 57 = 42$	$24^2 \bmod 57 = 6$
7		$9 \cdot 42 \bmod 57 = 36$	$6^2 \bmod 57 = 36$
8			$36^2 \bmod 57 = 42$
9			$42^2 \bmod 57 = 54$
10			$36 \cdot 54 \bmod 57 = 6$
11			

Цілком очевидно, що в запропонованому способі паралельного обчислення модулярної експоненти час експоненціювання визначається часом обробки першого фрагменту коду експоненти $T_1 = (n\text{-}m) \cdot t_q + t_m$, t_q – час модулярного піднесення до квадрату. Приймаючи до уваги, що $t_q \approx t_m/2$, сумарний час T_e обчислення модулярної експоненти з врахуванням часу передобчислень, визначається формулою:

$$T_e = m \cdot t_m + (n-m) \cdot t_q + t_m \approx n \cdot t_q + m \cdot t_q$$

Аналіз ефективності. Основною перевагою запропонованого способу модулярного експоненціювання є прискорення обчислення шляхом розпаралелювання. Тому, ефективніть запропонованого способу доцільно оцінювати коефіцієнтом прискорення, який розраховується відношенням часу виконання модулярного експоненціювання класичним способом – T_0, до часу виконання запропонованого способу паралельного обчислення модулярної експоненти – T:

$$\beta = \frac{T_0}{T}.$$

Для випадку, коли кількість процесорів не обмежена, коефіцієнт прискорення обраховується за формулою:

$$\frac{3 \cdot n \cdot t_q}{n \cdot t_q + m \cdot t_q} = \frac{3 \cdot n}{n + m}$$

Враховуючи, що$m \ll n$, реальне значення коефіцієнту прискорення при відсутності обмежень на кількість процесорів, дорівнює 3: $\beta \approx 3$.

Висновки. В результаті проведених досліджень, направлених 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 виконання передобчислень.

Для практичної реалізації цієї можливості запропонована організація прискореного обчислення модулярної експоненти на багатоядерному процесорі. Теоретично та експериментально доведено, що розроблена організація забезпечує практично трьократне 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 ВИКОРИСТАНИХ ДЖЕРЕЛ

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.

ДОВІДКА ПРО АВТОРІВ

Виноградов Юрій Миколайович – старший викладач, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Vinogradov Yurii – associate professor, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

Міщенко Людмила Дмитрівна – студентка, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Mishchenko Liudmyla - student, Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv PolytechnicInstitute".

E-mail: liudamishchenko@gmail.com

Mishchenko Liudmyla,
Vinohradov Yurii

METHOD FOR PARALLEL MODULAR EXPONENTIATION BY USING PRECOMPUTATION

Topic relevance. Development new ways of acceleration programmatic implementation of modular arithmetics. These can be used in information encryption protocols and can affect both local and global computer networks.

Problem setting. Computation of the modular exponent, which is the spreadest operation in communication protocols. This means calculations of A^E mod M where A, E and M are n-digit numbers, much greater than CPU bit depht.

Actual researches and issues analysis. The executed analysis of modular exponent calculation procedures showed, that acceleration of execution can be achieved by using parallelization on different levels. The majority of works in this topic are oriented on the level of CPU operations, which are the components of modular squaring and multiplication.

Uninvestigated parts of general matters defining. Relying on the investigated references, the next conclusions can be done. The main source of acceleration computation speed of modular exponent is organization parallel processing. But this solution cannot be implemented on slow microcontrollers. Thereby existing methods do not solve the problem for huge amount of devices, which require communication protocols of information encryption.

Target setting. The investigation endpoint is acceleration modular exponent calculation operation, which is critical for communication protocols of information encryption.

The statement of basic materials. The proposed solution is to split n-digit exponent code into s m-digit components. Computation of modular exponent is proposed to calculate using s-1 CPUs. Each is responsive for calculation of corresponding exponent fragment.

Conclusion. The executed analysis of calculation showed that acceleration can be achieved using precalculations. More significant acceleration of modular exponent calculation can be achieved with transition to the level of CPU multiplication operations.

Key words: computer arithmetic, parallel calculation, modular multiplication, modular exponentiation, data security protocols.