УДК 004.056.5

DOI: 10.18372/2073-4751.81.20130

Марковський О.П., к.т.н.,
orcid.org/0000-0003-3483-4233,
e-mail: markovskyy@i.ua,
Нікольський С.С.,
orcid.org/0000-0003-4893-3339,
e-mail: sergiy.nikolskiy@gmail.com

МЕТОД РОЗПОДІЛЕНОГО ОБЧИСЛЕННЯ НА БАГАТОЯДЕРНИХ ПРОЦЕСОРАХ ЕКСПОНЕНТИ НА ПОЛЯХ ГАЛУА ДЛЯ КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Вступ

Динамічний розвиток технологій Інтернету дозволив підняти в останні роки його технічні характеристики на якісно новий рівень. Це, в свою чергу, педалює розширення використання Інтернету в якості середовища оперативного обміну даними в системах контролю та управління віддаленими об'єктами реального світу.

Для широкого кола таких систем в якості вагомого чинника ефективності виступає рівень захищеності від стороннього втручання їх роботу. Застосування потенційно відкритого середовища Інтернету для обміну даними, крім безперечних переваг, таких, зокрема, як низька вартості та використання існуючої інфраструктури, створює певні загрози безпеці систем віддаленого управління.

Для ефективної протидії такого роду загрозам може бути використаний увесь арсенал існуючих механізмів мережевої безпеки, які з значною мірою базується на алгоритмах несиметричної криптографії. Зокрема, здатний найбільш ефективно протидіяти спробам стороннього втручання в роботу систем віддаленого контролю механізм цифрового підпису базується на засадах такої криптографії. Добре відомо [1], що з початком винайдення криптографії з відкритим ключем, основна проблема її практичного використання полягає в складності обчислень. Базовою обчислювальною операцією криптографії з відкритим ключем є модулярне експоненціювання, обчислювальна складність якого має кубічну залежність від розрядності n чисел. З іншого боку, розрядністю визначається рівень захищеності, яку забезпечує криптографія з відкритим ключем. На сьогоднішній день для більшості практичних застосувань використовуються числа розрядністю 4096. Реалізація модулярного експоненціювання над числами такої розрядності потребує порядку 1011 процесорних операцій і, відповідно, вимагає помітних часових ресурсів. Для комп'ютерних систем великої та середньої потужності вказана проблема вирішується за рахунок застосування криптопроцесорів, які реалізують операцію модулярного експоненціювання над довгими числами на апаратному рівні. Для термінальних обчислювальних платформ, які складають близько 65% комп'ютерних пристроїв, потрібно шукати інші шляхи прискорення реалізації криптографії з відкритим ключем.

Один із можливих шляхів прискорення комп'ютерної реалізації криптографії з відкритим ключем полягає в переході до альтернативної алгебри, зокрема до виконання криптографічних перетворень на кінцевих полях Галуа $GF(2^n)$ [2]. В силу специфічних властивостей цієї алгебри мультиплікативні операції над числами великої розрядності виконуються в цій алгебрі значно простіше і швидше [3]. Використання алгебри полів Галуа відкриває якісно ширші можливості для розпаралелювання реалізації механізмів криптографії з відкритим ключем на багатоядерних процесорах. Для втілення цих можливостей потрібно створити ефективні методи

реалізації базової операції – експоненціювання на кінцевих полях Галуа.

Таким чином, наукова задача прискорення комп'ютерної реалізації механізмів криптографічного захисту з відкритим ключем за рахунок використання алгебри полів Галуа і організації розпар алелювання відповідних обчислювальних процесів є актуальною для сучасного етапу розвитку інформаційних технологій.

Огляд та аналіз існуючих підходів до прискорення експоненціювання на полях Галуа

Поле Галуа $GF(2^n)$ яке включає в себе множину n-розрядних чисел, утворюється простим поліномом $P(x) = x^n + p_{n-1}\cdot x^{n-1}+\ldots+p_2\cdot x^2 + p_1\cdot x + p_0$, $\forall j \in \{0,1,2,\ldots,n-1\}$: $p_j \in \{0,1\}$, який співвідноситься з числом $P = 2^n + p_{n-1}\cdot 2^{n-1}+\ldots+p_2\cdot 2^2 + p_1\cdot 2 + p_0$.

Базові операції на кінцевих полях Галуа $GF(2^n)$ відрізняються від операцій а традиційній алгебрі. Зокрема операція додавання, яка позначається символом '$\oplus$' зводиться до побітового обчислення логічної суми (*XOR*) доданків. Операція редукції числа A на полі Галуа, заданого утворюючим простим поліномом $P(x)$, яка позначається як A *rem* P, являє собою обчислення залишку від поліноміального ділення поліному $A(x)$, що співвідноситься з числом A, на поліном $P(x)$. Множення $A \otimes B$ *rem* P на полях включає в себе дві операції: власне поліноміальне множення та редукцію отриманого добутку.

Операція обчислення експоненти на полях Галуа A^E *rem* P, де $E = e_{n-1}\cdot 2^{n-1}+\ldots+e_2\cdot 2^2 + e_1\cdot 2 + e_0$, $\forall j \in \{0,1,\ldots,n-1\}$: $p_j \in \{0,1\}$ як і модулярне експонентціювання, виконується за одним із двох різновидів класичного алгоритму [1]. В рамках першого із них реалізується послідовне обчислення всіх значень числа A в ступенях $1,2,4,\ldots,2^{n-1}$. Результат формується як добуток на полі Галуа тих ступенів, які відповідають одиничним значенням розрядів коду експоненти. Виходячи з послідовності в якому обчислюються ступені числа А, розряди коду експоненти Е аналізуються починаючи від молодших до старших. Фактично обчислювальний процес організується у вигляді двох потоків, я рамках першого із яких послідовно обчислюються ступені числа А, а в рамках другого реалізується формування добутку на полі Галуа тих із них, які співвідносяться з одиничними значеннями розрядів коду експоненти. Відповідно, в різновиді класичного алгоритму з порядком аналізу розрядів коду експоненти від молодших розрядів до старших, обчислення експоненти на полі Галуа відбувається у відповідності з наступною формулою:

$$A^E \, rem \, M = \left(\bigotimes_{j=0}^{n-1} A^{2^j \cdot e_j} \, rem \, P\right) rem \, P \,. \quad (1)$$

В рамках другого різновиду класичного алгоритму експоненціювання на полях Галуа аналіз розрядів коду експоненти здійснюється в напрямку від старших розрядів до молодших і організується один обчислювальний процес, який передбачає піднесення до квадрату попереднього результату і множення його на число А. В цьому різновиді класичного алгоритму обчислення експоненти на полі Галуа відбувається у відповідності з такою формулою:

$$A^E \, rem \, P = ((A^{e_{n-1}})^2 \otimes .. \otimes A^{e_1})^2 \otimes A^{e_0} \, rem \, P \,. (2)$$

Обидва різновиди класичного алгоритму обчислення експоненти на полях Галуа реалізуються за n операцій піднесення до квадрату на полях Галуа і, в середньому, за $n/2$ операцій множення.

Переважна частка досліджень, присвячених зменшенню часу комп'ютерної реалізації операцій експоненціювання на кінцевих полях Галуа $GF(2^n)$ направлена на прискорення операцій піднесення до квадрату та множення на полях Галуа [4]. Разом з тим, в роботі [5] запропоновано метод зменшення кількості операцій множення на полях Галуа при експоненціюванні за другим різновидом класичного алгоритму групами по k розрядів коду експоненти Е. Метод дозволяє зменшити кількість операцій множення на полях Галуа в $k/2$ раз за рахунок передобчислень значень результатів піднесення A в ступені $2,3,\ldots,2^k-1$ на полі Галуа.

З робіт, присвячених змешенню часу виконання множення на полях Галуа близько третини пропонують апаратні рішення, які, здебільшого, орієнтовані на реалізацію на програмованих матрицях [6-8]. В цих методах прискорення досягається за рахунок зменшення часу перестановок бітів та суміщенні операцій піднесення до квадрату і множення на полях Галуа.

Операція множення на полях Галуа складається власне з поліноміального множення та обчислення залишку від поліноміального ділення отриманого добутку на утворюючий поліном поля $P(x)$, тобто редукції. Прискорення першої із зазначених операцій може бути досягнуто за рахунок множення одночасно на групу з k розрядів множника [9]. Таке рішення дозволяє досягти k-кратного прискорення поліноміального множення, проте воно не вливає на іншу складову множення на полях Галуа – редукцію. Для уникнення цього недоліку в роботі [10] запропоновано комбіновано застосування множення на k бітів множника і групової редукції Монтгомері часткового добутку на таку ж кількість розрядів. Це дозволило реально прискорити множення на полях Галуа в k раз. При цьому в значення k обмежується об'ємом пам'яті для зберігання двох таблиць передобчислень. В роботі [11] пропонується підхід до зменшення об'єму таблиць передобчислень і, відповідно, прискорення реалізації мультиплікативних операцій на полях Галуа.

Для виконання редукції на полі Галуа може бути використана як технологія Монтгомері (редукція молодших розрядів) так і технологія прямого поліноміального ділення. На відміну від модулярної алгебри, в алгебрі полів Галуа операції поліно-міального множення та редукції можуть бути суміщені при застосуванні обох технологій реалізації останньої [12]. Це дозволяє обмежити розрядність проміжних результатів і зменшити час виконання операцій.

В роботі [13] запропоновано метод прискореного виконання мультиплікативних операцій на полях Галуа за рахунок суміщення піднесення до квадрату на множення на постійне число А. Для цього використано властивість поліноміального квадрату числа, яка полягає в тому, що ця операція може бути зведена до вставки нулі між двійковими розрядами числа. Запропонований в [13] дозволяє зменшити середню кількість операцій логічного додавання n-розрядних чисел для виконання одного циклу експонентціювання до рівня $0.68 \cdot n$.

Проведений аналіз відомих підходів до прискорення важливої для криптографічного захисту з відкритим ключем операції експоненціювання на кінцевих полях Галуа показав, що в свої більшості вони орієнтовані на одноядерну архітектуру процесору. Разом з тим, тенденцією сьогодення є використання багатоядерних процесорів, які надають змогу розпаралелити процес обчислення. Виходячи з цього перспективним напрямком подальшого прискорення експоненціювання на кінцевих полях Галуа є розробка підходів до організації паралельного обчислення цієї операції.

Мета досліджень

Мета досліджень полягає в прискоренні комп'ютерної реалізації важливої для криптографічних застосувань операції обчислення експоненти на полях Галуа шляхом організації розпаралелювання її виконання на багатоядерних процесорних платформах.

Організація паралельного обчислення експоненти на полях Галуа

Для досягнення поставленої мети пропонується організувати обчислення на k-ядерному процесорі експоненти $A^E \; rem \; P$ на полі Галуа у вигляді k незалежних процесів обчислення часткових експонент $A^{E_1} rem \, P, A^{E_2}, ..., A^{E_E} rem \, P$ добуток яких формує остаточний результат операції :

$$A^E rem \, P = (\bigotimes_{j=1}^{k} A^{E_j} rem \, P) rem \, P . \quad (3)$$

Очевидно, що при цьому сума часткових ступенів $E_1, E_2, \ldots, E_k$ дорівнює коду експоненти E: $E_1 + E_2 + \ldots + E_k = E$.

В теоретичному плані можна розглядати ряд варіантів розкладення коду експоненти E на адитивні складові $E_1,E_2,\ldots,E_k$:

1. Схема з побітовим розподіленням передбачає, що код кожної h-тої, $h\in\{1,2,\ldots,k\}$, часткової експоненти E_h включає в себе n/k значимих бітів експоненти E, відокремлених k-1 нульовими бітами. Тобто, код часткової експоненти E_h формується згідно наступної формули:

$$\forall h \in \{1,2...k\} : E_h = \sum_{j=0}^{n/h-1} e_{n-j\cdot h+h-1} \cdot 2^{n-j\cdot h+h-1} .(4)$$

Наприклад, якщо n=12, код експоненти $E = 3311 = 110\ 011\ 101\ 111_2$, а кількість k ядер дорівнює чотирьом: k=4, то утворені за описаною схемою часткові ступені у відповідністю з формулою (4) мають вигляд: E_1=100 010 001 000_2 = 2184; E_2 = 010 001 000 100_2 = 1092; E_3 = 000 000 100 010_2 = 34; E_4 = 000 000 000 001_2 = 1. Очевидно, що 2184+1092+34+1=3311=E.

2. Схема з груповим розподіленням також передбачає поділення коду експоненти E розділяється на k адитивних складові: $E=E_1+E_2+\ldots+E_k$. Перша складова E_1 містить в собі старші n/k значущі розряди коду повної експоненти E : $E_1 = e_{n-1}\cdot 2^{n-1}+e_{n-2}\cdot 2^{n-2}+ \ldots+e_{n-n/k}\cdot 2^{n-n/k}$. Відповідно, друга складова E_2 містить в собі наступні n/k значущі розряди коду повної експоненти E : $E_2 = e_{n-n/k-1}\cdot 2^{n-n/k-1}+ e_{n-n/k-2}\cdot 2^{n-n/k-2}+\ldots+ e_{n-2\cdot n/k}\cdot 2^{n-2\cdot n/k}$. Остання k-та складова E_k коду повної експоненти E складається з її k молодших розрядів : $E_k = e_{n/k-1}\cdot 2^{n/k-1} +e_{n/k-2}\cdot 2^{n/k-2}+\ldots+e_1\cdot 2+e_0$. Наприклад, якщо повний код експоненти має вигляд: $E = 3311 = 110\ 011\ 101\ 111_2$, а кількість k ядер дорівнює чотирьом: k=4, то утворені за описаною схемою складові експоненти мають такий вигляд: E_1=110 000 000 000_2 = 3072; E_2 = 011 000 000_2 = 192; E_3 = 101 000_2 = 40; E_4 = 111_2 = 7.

Вочевидь, що при застосуванні обох схем розкладення коду експоненти E на складові $E_1,E_2,\ldots,E_k$ і реалізації обчислення часткових експонент за одним із двох різновидів класичного алгоритму одночасно виконуються лише операції множення на полях Галуа, які співвідносяться з одиничними значеннями бітів кодів $E_1,E_2,\ldots,E_k$. Це дозволяє скоротити час виконання операцій множення в k раз. Натомість, кількість операцій піднесення до квадрату на полях Галуа на кожному із ядер процесора визначається позицією старшої значущої одиниці в відповідному із кодів $E_1,E_2,\ldots,E_k$. Для схеми з побітовим розподіленням при реалізації класичного алгоритму на кожному із ядер виконується близько n операцій піднесення до квадрату на полях Галуа. В схемі експоненціювання з груповим розподіленням на першому ядрі реалізується n таких операцій, на другому n-k, і, відповідно, на останньому k-му – лише k піднесень до квадрату на полі Галуа. Тобто, при застосуванні будь-якої з схем розкладення коду експоненти E на одному із ядер здійснюється n піднесень до квадрату на полях Галуа. Це означає, що загальний час обчислення експоненти A^E *rem* P на полі Галуа визначається тривалістю виконання n+0.5·n/k мультиплікативних операцій, що визначає теоретичну верхню межу прискорення як 1/3, тобто не більше 30%.

Саме ці резони роблять малоефективними спроби організувати паралельне обчислення модулярної експоненти [14]. Проте для експоненти на кінцевих полях Галуа з використанням передобчислень, що залежать лише від утворюючого поліному $P(x)$, можна піднести число A в будь-яку ступінь двійки за однаковий час. Саме ця властивість піднесення в ступінь двійки на полях Галуа робить можливим ефективне розпаралелювання обчислення експоненти A^E *rem* P на багатоядерних процесорах.

Для швидкого піднесення числа A в ступінь двійки A^{2^m} *rem* P на кінцевому полі Галуа $GF(2^n)$ може бути застосована технологія, подібна до поліноміального піднесення до квадрату, з використанням таблиці Q передобчислень. Формування зазначеної таблиці Q реалізується після вибору утворюючого поліному P та

зводиться до виконання наступної послідовності дій:

1. Нульовому рядку таблиці Q присвоюється значення одиниці: $Q[0]=1$. Першому рядку цієї ж таблиці присвоюється значення остачі від ділення на утворюючий поліном P двійки в ступені 2^m: $Q[1] = 2^{2^m} \; rem \; P$. Лічильник j номеру поточного рядка таблиці Q встановлюється в двійку: $j = 2$.

2. Значення j-того рядка таблиці Q, обчислюється як добуток на кінцевому полі Галуа значення попереднього $Q[j\text{-}1]$ та першого $Q[1]$ рядка таблиці: $Q[j] = Q[j\text{-}1] \otimes Q[1] \; rem \; P$.

3. Значення лічильника j збільшується на одиницю: $j = j+1$. Якщо j менше за n: $j<n$, то здійснюється перехід на п.2.

4. Кінець виконання процедури.

В результаті виконання описаної процедури отримується сформована таблиця Q передобчислень об'ємом n^2 біт. Функціонування описаної процедури формування таблиці Q передобчислень для піднесення числа в довільну ступінь m двійки може бути ілюстровано прикладом побудови такої таблиці для m=4. Тобто з використанням такої таблиці можна швидко підносити будь-яке число в ступінь 2^4 =16. Нехай, ступінь n утворюючого поліному $P(x)$ дорівнює 12-ти: $n = 12$, а сам поліном $P(x) = x^{12}+x^{11}+x^9+x^5+x^3+x+1$, двійковому представленню якого відповідає число $P = 0001\ 1010\ 0010\ 1011_2 = 6699$.

Згідно п.1 процедури, значення нульового рядка $Q[0]$ таблиці встановлюється в одиницю: $Q[0]=1$, а першому рядку присвоюється значення $2^{16} \; rem \; 6699 = 367$: $Q[1]=367$. Лічильник j встановлюється в двійку: $j = 2$.

Відповідно до п.2 другий рядок $Q[2]$ таблиці обчислюється наступним чином: $Q[2] = Q[1] \otimes Q[1] \; rem \; P = 367 \otimes 367 \; rem \; 6699 = 3857$. В рамках п.3 лічильник j інкрементується: $j = 2+1=3$, та здійснюється перехід на п.2, в силу того, що $3 < 12$.

Згідно з п.2 третій рядок $Q[3]$ таблиці обчислюється як добуток другого $Q[2]$ та першого $Q[1]$ рядків на полі Галуа: $Q[3] = Q[2] \otimes Q[1] \; rem \; P = 3857 \otimes 367 \; rem \; 6699 = 1432$. Відповідно до п.3 лічильник j збільшується на одиницю: $j = 3+1=4$, після чого здійснюється перехід на п.2. Визначення значень всіх інших рядків таблиці Q здійснюється аналогічним чином.

В результаті послідовного виконання запропонованої процедури результати передобчислень набувають вигляду представленого в табл. 1.

Таблиця 1. Приклад формування передобчислень Q для $m = 4$ та $P = 6699$

j	$Q[j]$	j	$Q[j]$
0	1	6	2597
1	367	7	2182
2	3857	8	2904
3	1432	9	2450
4	3635	10	1276
5	536	11	2539

Як зазначалось вище, таблиця Q передобчислень залежить лише від обраного значення m та утворюючого поліному $P(x)$,що входить в якості компоненти до складу відкритого ключа і тому може вважатися практично постійним. Це означає, що таблиця Q передобчислень практично формується один раз і, відповідно, час її побудови не вливає на часову ефективність обчислення експоненти на полях Галуа.

Процедура прискореного піднесення числа $A = a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} + \ldots + a_1\cdot 2 + a_0$, $\forall j=0,1,\ldots,n\text{-}1$: $a_j \in \{0,1\}$ в ступінь двійки на кінцевому полі Галуа $GF(2^n)$ з використанням таблиці Q передобчислень зводиться до виконання послідовності дій, описаної нижче:

1. Змінна D встановлюється в нуль: $D = 0$. Лічильник j поточного розряду a_j числа A встановлюється в n-1: $j = n\text{-}1$.

2. Якщо значення поточного розряду a_j числа A дорівнює одиниці: $a_j = 1$, то до змінної D логічно додається j-тий рядок таблиці Q передобчислень: $D = D \oplus Q[j]$:

3. Лічильник j декрементується: $j = j\text{-}1$. Якщо j більше або дорівнює нулю: $j \geq 0$, то здійснюється перехід на п.2.

4. Кінець процедури. В змінній D сформовано результат операції $A^{2^m} \; rem \; P$.

Робота наведеної процедури може бути ілюструвати таким прикладом.

Нехай, обчислюється 1393^{16} *rem* 6699 = 2475 з використанням таблиці 1; відповідно число A = 1393= 0101 0111 0001_2, m = 4, P = 6699 *а* n = 12.

Згідно п.1 запропонованої процедури: D = 0, а лічильник j встановлюється в n-1, тобто 12-1: j = 11.

В рамках п.2, оскільки, a_{11} = 0, то одразу здійснюється перехід на п.3, згідно з яким значення j зменшується на одиницю: j = 11-1=10, та виконується повернення на п.2.

Відповідно до п.2 значення десятого розряду числа A дорівнює одиниці: a_{10} = 1, тобто до змінної D логічно додається десятий рядок таблиці 1: $D = 0 \oplus 1276$=1276. В рамках п.3, значення лічильника j зменшується на одиницю: j = 9, та здійснюється перехід на повторне виконання п.2.

Подальше послідовне виконання процедури здійснюється аналогічно описаному вище. Динаміка зміни результату D, в залежності від значення поточного розряду a_j, для кожного j, наведена в табл. 2.

Таблиця 2. Динаміка зміни результату D в ході виконання процедури при обчисленні 1393^{16} *rem* 6699 = 2475

j	a_j	D	j	a_j	D
11	0	0	5	1	1945
10	1	1276	4	1	2474
9	0	1276	3	0	2474
8	1	4001	2	0	2474
7	0	4001	1	0	2474
6	1	1409	0	1	2475

В змінній D сформовано результат обчислення 1393^{16} *rem* 6699 = 2475.

Таким чином, викладена процедура піднесення будь-якого числа в довільну ступінь m двійки реалізується, в середньому, за $0.5 \cdot n$ операцій логічного додавання n-розрядних чисел незалежно від значення m.

Організація паралельного експоненціювання з побітовим еквідистанційним розподіленням

Схема з побітовим еквідистанційним розподіленням значимих бітів коду експоненти E по часткових ступенях $E_1,E_2,\ldots,E_k$ в які підноситься число A на кожному із ядер процесора передбачає що кожен із кодів $E_1,E_2,\ldots,E_k$ включає в себе n/k розрядів E, причому вони розділені проміжками з k-1 нульових бітів, тобто бітова дистанція між значущими бітами однакова. Це означає, що на кожному із k ядер реалізується послідовність із n/k ітерацій, в рамках кожної із яких виконується, за умови, що відповідний біт експоненти дорівнює одиниці, одна операція множення на полі Галуа, результат якої підноситься в ступінь 2^k на полі Галуа (крім останньої ітерації). На останній ітерації результат підноситься до ступеня 2^{k-h}, де h – номер ядра, $h \in \{1,2,\ldots,k\}$.

Піднесення до ступеня 2^k, яке виконується на кожному із ядер в рамках перших n/k-1 ітерацій здійснюється з використанням передобчислень за процедурою, викладеній в попередньому розділі. Для цього, в пам'ять кожного із ядер має бути завантажена однакова таблиця Q передобчислень для m=k.

Запропонований організація паралельного обчислення експоненти на полі Галуа передбачає використання одного ядра, для визначеності k-того, в якості провідного, для обчислення на ньому часткової експоненти A^{E_k} *rem* P, а також формування на цьому ядрі кінцевого результату, як добутку на полі Галуа результатів, отриманих на всіх ядер процесора. На всіх інших ядрах здійснюється обчислення відповідної часткової експоненти на полі Галуа та відправку результату на провідне ядро.

Таким чином, реалізація запропонованої схеми паралельного експонентціювання на полі Галуа з побітовим еквідистанційним розподіленням складається з двох процедур: процедури роботи провідного ядра, та процедури роботи підконтрольних ядра. Перед початком роботи цих процедур в пам’яті кожного з ядер передбачається наявність таблиці Q передобчислень для швидкого піднесення в ступінь 2^k на кінцевому полі Галуа, порядок побудови і використання якої описано в попередньому підрозділі.

Процедура роботи провідного ядра, для визначеності *k*-того, зводиться до виконання наступної послідовності дій:

1. Значення числа *A* надсилається всім ядрам.

2. Змінна *R* результату встановлюється в одиницю: *R* = 1, а лічильник *j* номеру розряду експоненти *E* встановлюється рівним значенню *n-k*: *j* = *n-k*.

3. Змінна *R* підноситься в ступінь 2^k на кінцевому полі Галуа з використанням таблиці *Q* передобчислень згідно процедури прискореного піднесення числа в ступінь двійки, описаної вище: $R = R^{2^k}$ *rem P*.

4. Якщо значення *j*-того розряду експоненти *E*, дорівнює одиниці: $e_j = 1$, то змінна *R* домножається на число *A* на кінцевому полі Галуа: $R = R \otimes A$ *rem P*.

5. Лічильник *j* зменшується на значення *k*: *j* = *j-k*. Якщо *j* більше або дорівнює *k-h* :*j*≥ *k -h*, то здійснюється перехід на повторне виконання п.3.

6. Лічильник і встановлюється в одиницю: i = 1.

7. Лічильник *s* встановлюється в *k*-1: *s* = *k*-1.

8. Перевіряється наявність сигналу завершення обчислення результату на *s*-тому ядрі. Якщо сигнал не надійшов, то здійснюється перехід на п.13.

9. Видається сигнал готовності до прийому результату від ядра *s*.

10. Отриманий, від ядра *s*, результат записується в змінну *X*.

11. Змінна *R* помножається на *X* з знаходженням остачі від ділення цієї операції на утворюючий поліном *P* поля Галуа: $R = R \otimes X$ *rem P*.

12. Лічильник *i* збільшується на одиницю: i = i +1. Якщо і рівне *k*: *i* = *k*, то здійснюється перехід на п.14.

13. Лічильник *s* зменшується на одиницю: *s* = *s*-1. Якщо *s* не дорівнює нулю: $s \neq 0$, то здійснюється перехід на п.8, інакше: *s* = 0, - перехід на п.7.

14. Кінець процедури. В змінній *R* сформовано результат A^E *rem P*.

Для обчислення A^{E_h} *rem P* запропонована процедура роботи підконтрольного ядра передбачає виконання наступної послідовності дій:

1. Ядро *h* отримує значення числа *A* від *k*-того ядра.

2. Змінна R_h результату встановлюється в одиницю: $R_h = 1$, а лічильник *j* номеру розряду експоненти *E* обчислюється як *n-h*: *j* = *n-h*.

3. Змінна R_h підноситься в ступінь 2^k на кінцевому полі Галуа з використанням таблиці *Q* передобчислень згідно процедури прискореного піднесення числа в ступінь двійки, описаної вище: $R_h = {R_h}^{2^k}$ *rem P*.

4. Якщо поточний розряд e_j експоненти *E*, дорівнює одиниці: $e_j = 1$, то змінна R_h домножається на число *A* на кінцевому полі Галуа: $R_h = R_h \otimes A$ *rem P*.

5. Лічильник *j* зменшується на значення *k*: *j* = *j-k*. Якщо *j* більше або дорівнює *h* :*j*≥*h* – здійснюється перехід на повторне виконання п.3.

6. Лічильник *s* встановлюється в одиницю: *s* = 1.

7. Змінна R_h підноситься до квадрату на кінцевому полі Галуа: $R_h = {R_h}^2$ *rem P*.

8. Лічильник *s* збільшується на один: *s* = *s*+1. Якщо *s* менше або дорівнює *k-h*: *s*≤*k-h*, то здійснюється перехід на п.7.

9. Видається сигнал про завершення обчислення результату R_h.

10. Приймається сигнал готовності від ядра *k* отримати результат R_h. За відсутності такого сигналу здійснюється перехід на повторне виконання п.9.

11. Результат R_h надсилається *k*-тому ядру. Знімається сигнал завершення обчислення.

12. Кінець процедури.

Роботу розроблених процедур схеми побітового розподілення експоненти можна проілюструвати наступним прикладом. Нехай, на чотирьох ядерному процесорі (k = 4) обчислюється експонента $1853^{3567} rem$ 6699 = 3724 на кінцевому полі Галуа $GF(2^{12})$, тоді число A = 1853 = 0111 0011 1101_2, E = 3567= 1101 1110 1111_2, а P = 6699.

Тоді, паралельне обчислення експоненти згідно з запропонованою схемою побітового розподілення на чотирьох ядрах здійснюється у наступному порядку:

Виконання п.1 обох процедур схеми побітового розподілення виконується синхронізовано та передбачає надсилання числа $A = 1853$ з k-того ядра іншим трьом ядрам. В рамках п.2-5 процедур роботи провідного та підконтрольних ядер проводиться асинхронне обчислення фрагментарної експоненти на кожному ядрі.

Приклад обчислення часткової експоненти 1853^{257} *rem* 6699 = 3824 на четвертому ядрі наведено нижче:

В рамках п.2 змінна R встановлюється в одиницю: $R = 1$, а лічильнику j присвоюється значення 12-4 = 8.

Згідно з п.3 змінна R підноситься в ступінь 2^4 з використанням таблиці.1: $R = 1$. Відповідно до п.4, восьмий розряд експоненти дорівнює одиниці: $e_8 = 1$, тобто змінна R помножається на значення A: $R = 1 \otimes 1853$ *rem* 6699 = 1853. В рамках п.5 лічильник j зменшується на чотири: $j = 8-4=4$, після чого здійснюється перехід на п.3.

Відповідно до п.3 змінна R підноситься в шістнадцяту ступінь з використанням таблиці.1: $R = 1853^{16}$ *rem* 6699 = 149. Згідно з п.4 значення четвертого розряду експоненти дорівнює нулю: $e_4 = 0$, тобто значення R залишається незмінним. Відповідно до п.5 лічильник j зменшується на чотири: $j = 4-4=0$, та виконується перехід на п.3.

В рамках п.3–п.5, значення змінної R спочатку підноситься до шістнадцятого ступеня з використанням передобчислень: $R = 149^{16}$ *rem* 6699 = 2469, після чого помножається на A, оскільки $e_0 = 1$: 2469·1853 *rem* 6699 = 3824; лічильник j зменшується на 4: $j = 0-4=-4$.

Послідовне виконання п.2-5 процедури роботи підконтрольного ядра для кожного h-того ядра, здійснюється аналогічно описаної вище процедури роботи провідного ядра. Динаміки зміни поточних результатів на кожному ядрі наведена в таблиці 3.

Після виконання п.2-5 процедури роботи підконтрольного ядра: перше, друге та третє – ядра, асинхронно виконують п.6-8 тієї ж процедури.

Паралельно з цим, четверте ядро формує остаточний результат у вигляді добутку на полі Галуа обчислених на всіх ядрах часткових експонент.

Обчислення часткової експоненти на третьому ядрі здійснюється таким чином: змінна R_3 підноситься до квадрату: $R_3 = 2573^2$ *rem* 6699 =727 один раз. Після цього третє ядро видає сигнал

готовності результату. четвертому ядру, яке в свою чергу приймає отримане значення та зберігає його в змінну X. Четверте ядро помножає змінну R та X на кінцевому полі Галуа: $R = 3824 \otimes 727$ *rem* 6699 = 18. Друге ядро виконує операцію піднесення до квадрату на полі Галуа два рази з видачею остаточного результату $R_2 = 2542$. Це значення передається на четверте ядро, на якому виконується множення: $R = 18 \otimes 2542$ *rem* 6699 = 158. Одночасно на першому ядрі послідовно виконуються три операції піднесення до квадрату, результатом яких є значення $R_1 = 3836$. Воно передається на четверте ядро, яке здійснює множення $R = 158 \otimes 3836$ *rem* 6699 = 3724.

Таблиця 3. Динаміка покрокової зміни поточних результатів на кожному ядрі при обчисленні 1853^{3567}*rem* 6699 = 3724

Пункт процедур	1 ядро			2 ядро			3 ядро			4 ядро		
	j	e_j	R_1	j	e_j	R_2	j	e_j	R_3	j	e_j	R
3	11	-	1	10	-	1	9	-	1	8	-	1
4	11	1	1853	10	1	1853	9	0	1	8	1	1853
3	7	-	149	6	-	149	5	-	1	4	-	149
4	7	1	2573	6	1	2573	5	1	1853	4	0	149
3	3	-	2801	2	-	2801	1	-	149	0	-	2469
4	3	1	2018	2	1	2018	1	1	2573	0	1	3824

Очевидна перевага розробленої організації паралельного обчислення експоненти на полі Галуа з побітовим еквідистанційним розподіленням значимих бітів експоненти полягає в тому, що на кожному із ядер використовується одні і та ж сама таблиця передобчислень Q при $m=k$.

Організація паралельного експоненціювання з груповим розподіленням бітів експоненти

Схема групового розподілення значущих бітів експоненти E по кодам часткових ступенів $E_1,E_2,\ldots,E_k$ полягає в тому, що кожна з них містить n/k значущих суміжних розрядів коду E. Відповідно на кожному h-тому ядрі процесора обчислювальний процес складається з двох фаз: піднесення числа A в ступінь, яка дорівнює значенню відповідної групи з n/k значущих суміжних розрядів коду E на піднесення отриманого результату в ступінь $2^{k\cdot(h-1)}$ на полі Галуа. В результаті виконання цих двох фаз на кожному h-тому ядрі процесора отримується код часткової експоненти $A^{E_h} rem P$. Обчислення експоненти A^E *rem P* здійснюється в вигляді добутку всіх фрагментарних експонент: $R = (A^{E_1}$ *rem P* $\otimes A^{E_2}$ *rem P* $\otimes \ldots \otimes A^{E_k}$ *rem P*) *rem P*, на кінцевому полі Галуа. Для цього, одне ядро, для визначеності k-те, використовується в якості провідного ядра, що формує кінцевий результату, а також обчислює A^{E_k} *rem P*. Інші, а саме кожне h-те ядро– підконтрольні, тобто виконують обчислення A^{E_h} *rem P* та надсилають результат провідному ядру.

Таким чином, реалізація запропонованої схеми складається з процедури роботи провідного ядра, та процедури роботи підконтрольного ядра. Остання передбачає наявність таблиці Q передобчислень в пам'яті h того ядра для піднесення в ступінь $2^{n-h\cdot n/k}$ на кінцевому полі Галуа, порядок побудови і використання якої описано вище.

Процедура роботи провідного ядра (в якості якого вибрано k-те) формалізована в вигляді наступної послідовності дій:

1. Значення числа A надсилається всім ядрам.
2. Змінна R результату встановлюється в одиницю: $R = 1$, а лічильник j номеру старшого розряду експоненти E_k встановлюється рівним n/k-1: $j = n/k$-1.
3. Змінна R підноситься до квадрату на кінцевому полі Галуа: $R = R^2$ *rem P*.
4. Якщо значення розряду e_j експоненти E, дорівнює одиниці: $e_j = 1$, то змінна R множиться на число A на кінцевому полі Галуа: $R = R \otimes A$ *rem P*.
5. Лічильник j декрементується: $j = j$-1. Якщо $j >= 0$ здійснюється перехід на повторне виконання п.3.
6. Лічильник і встановлюється в одиницю: i = 1.
7. Лічильник g встановлюється в k-1: $g = k$-1.
8. Перевіряється наявність сигналу завершення обчислення результату на g-тому ядрі. Якщо такий сигнал не надійшов, то здійснюється перехід на п.12.
9. Видається сигнал готовності до прийому результату R_g від ядра g.
10. Змінна R помножається на R_g з знаходженням остачі від ділення цієї операції на утворюючий поліном P поля Галуа: $R = R \otimes R_g$ *rem P*.
11. Лічильник i збільшується на одиницю: i = i+1. Якщо і рівне k: $i = k$, то здійснюється перехід на п.13.
12. Лічильник g зменшується на одиницю: $g = g$-1. Якщо g не дорівнює нулю: $g \neq 0$, то здійснюється перехід на п.8, інакше: $g = 0$, – повернення на п.7.
13. Кінець процедури. В змінній R сформовано результат A^E *rem P*.

Обчислення фрагментарних експонент A^{E_h} *rem P* на кожному h-тому ядрі здійснюється згідно процедури роботи підконтрольного ядра, описаної нижче:

1. Ядро h отримує значення числа A від k-того ядра.
2. Змінна R_h результату встановлюється в одиницю: $R_h = 1$, лічильнику j номеру поточного розряду експоненти E_h присвоюється значення n-$n/k\cdot(h$-1)-1: $j = n$-$n/k\cdot(h$-1)-1.

3. Змінна R_h підноситься до квадрату на кінцевому полі Галуа: $R_h = R_h^2$ *rem* P.

4. Якщо поточний розряд e_j експоненти E, дорівнює одиниці: $e_j = 1$, то змінна R_h домножається на число A на кінцевому полі Галуа: $R_h = R_h \otimes A$ *rem* P.

5. Лічильник j зменшується на одиницю: $j = j$-1. Якщо j більше або дорівнює n-$n/k \cdot h$: $j \geq n$-$n/k \cdot h$, то здійснюється перехід на повторне виконання п.3.

6. Змінна R_h підноситься в ступінь $2^{n-h \cdot n/k}$ на кінцевому полі Галуа з використанням таблиці Q_h передобчислень згідно процедури прискореного піднесення числа в ступінь двійки, описаної вище: $R_h = R_h^{2^{n-h \cdot n/k}}$ *rem* P.

7. Видається сигнал про завершення обчислення результату R_h.

8. Приймається сигнал готовності від ядра k отримати результат R_h. За відсутності такого сигналу здійснюється перехід на повторне виконання п.7.

9. Результат R_h надсилається k-тому ядру. Знімається сигнал завершення обчислення.

10. Кінець процедури.

В розробленій організації паралельного обчислення експоненти на полі Галуа з груповим розподіленням значимих бітів експоненти на кожному із ядер використовуються різні таблиці передобчислень Q для швидкого піднесення в різні ступені двійки.

Оцінка ефективності

Виходячи з поставленої мети, в якості основного критерію ефективності доцільно аналізувати прискорення обчислення експоненти на кінцевих полях Галуа за рахунок розпаралелювання обчислювального процесу. При цьому порівнянні часових характеристик реалізації доцільно проводити як по відношенню до виконання експоненціювання у вигляді єдиного обчислювального процесу, так і по обом запропонованим варіантам його розпаралелювання.

Обчислення експоненти A^2 *rem* P на кінцевих полях Галуа $GF(2^n)$ за умови його реалізації у вигляді єдиного обчислювального процесу потребує n операцій піднесення до квадрату та, в середньому, $0.5 \cdot n$ операцій множення. Якщо для реалізації обох цих мультиплікативних операцій використовується множення з суміщеною редукцією, то, в середньому, обчислення експоненти на полях Галуа потребує $N_0 = 3 \cdot n^2$ операцій логічного додавання (*XOR*) n– розрядних чисел.

Якщо операція експоненціювання на полях Галуа розподілено реалізується на k ядрах процесорах, то на кожному із k ядер, виконується, в середньому, $0.5 \cdot n/k$ операцій множення. Кожна така операція здійснюється з використанням, в середньому, $2 \cdot n$ операцій логічного додавання (*XOR*) n– розрядних чисел.

В варіанті розпаралелювання за схемою побітового розподілення значущих одиниць коду Е експоненти по частковим ступеням $E_1, E_2, \ldots, E_k$, операція піднесення в ступінь 2^k на полі Галуа з використанням передобчислень здійснюється n/k раз, причому її реалізація потребує, в середньому, $n/2$ логічних додавань n– розрядних чисел. Виходячи з наведеного, загальна середня кількість N_{10} таких операцій для реалізації експоненціювання на полях Галуа з розпаралелюванням за зазначеною схемою визначається наступним виразом:

$$N_{10} = \frac{n}{2 \cdot k} \cdot 2 \cdot n + \frac{n}{k} \cdot \frac{n}{2} = \frac{3 \cdot n^2}{2 \cdot k}. \qquad (5)$$

В варіанті розпаралелювання обчислювання експоненти за схемою групового розподілення значущих одиниць коду Е по частковим ступеням $E_1, E_2, \ldots, E_k$, операція піднесення в ступінь $2^{n-k \cdot h}$ на полі Галуа з використанням передобчислень здійснюється на кожному h-тому процесорі лише один раз і потребує, в середньому, $n/2$ логічних додавань n-розрядних чисел. Проте, в цьому варіанті в процесі обчислення на кожному із процесорних ядер експоненти від n/k значущих розрядів здійснюється ще n/k операцій піднесення до квадрату на полі Галуа. Вважаючи, що реалізація цієї операції потребує, в середньому, $0.86 \cdot n$

логічних додавань *n*– розрядних чисел [13], загальна середня кількість N_{20} таких операцій, що витрачаються для реалізації обчислення експоненти на кінцевих полях Галуа визначається формулою:

$$N_{20} = \frac{n^2}{k} + 0.86 \cdot \frac{n^2}{k} + \frac{n}{2} \approx 1.86 \cdot \frac{n^2}{k}. \quad (6)$$

Коефіцієнт прискорення β_{10} обчислення експоненти на полях Галуа за рахунок розпаралелювання на *k* процесорних ядрах за схемою побітового розподілення значущих одиниць коду Е експоненти по частковим ступеням $E_1,E_2,\ldots,E_k$, може бути визначений як співвідношення кількості N_0 операцій логічного додавання *n*–розрядних чисел при реалізації експоненціювання у вигляді єдиного обчислювального процесу до числа N_{10} таких операцій при паралельній реалізації:

$$\beta_{10} = \frac{N_0}{N_{10}} = 2 \cdot k. \quad (7)$$

Таким чином, використання запропонованої схеми розпаралелювання з організацією *k* обчислювальних процесів на відповідній кількості процесорних ядер дозволяє прискорити процес обчислення експоненти на кінцевих полях Галуа в $2 \cdot k$ раз. При цьому чинниками досягнутого прискорення виступає як зменшення в *k* раз кількості операцій множення в рамках кожного із процесів, так і організація піднесення в ступінь 2^k з застосуванням передобчислень.

При розпаралелювання за схемою групового розподілення значущих одиниць коду Е експоненти по частковим ступеням $E_1,E_2,\ldots,E_k$, коефіцієнт прискорення β_{20} визначається аналогічним чином і дорівнює:

$$\beta_{20} = \frac{N_0}{N_{20}} = \frac{3 \cdot k}{1.86} = 1.6 \cdot k. \quad (8)$$

Порівняння виразів (8) та (7) свідчить про те, що перша із запропонованих схем організації розпаралелювання обчислювального процесу на *k* процесорних ядрах є більш ефективною в ракурсі прискорення експоненціювання на кінцевих полях Галуа. Коефіцієнт прискорення β_{12} обчислення експоненти на полях Галуа, першої схеми в порівнянні з другою може бути визначений як співвідношення кількість операцій логічного додавання *n*-розрядних чисел, які реалізуються в другому – N_{20} та першому – N_{10} варіантах:

$$\beta_{12} = \frac{N_{20}}{N_{10}} = \frac{2 \cdot 1.86}{3} = 1.24. \quad (9)$$

Таким чином, варіант паралельної реалізації на *k* процесорних ядрах експоненціювання на полях Галуа за схемою побітового розподілення значущих одиниць коду Е експоненти по частковим ступеням $E_1,E_2,\ldots,E_k$ на 24% швидший за варіант з груповим розподілення.

Висновки

В результаті виконання досліджень, направлених на прискорення комп'ютерної реалізації операції обчислення експоненти на полях Галуа шляхом розпаралелювання її виконання на багатоядерних процесорних платформах виявлено можливості, що дозволяють, на відміну від модулярного експоненціювання, реалізувати ефективну паралельну обробку. Ці можливості полягають в використанні спеціальної схеми передобчислень, яка дозволяє за однаковий час здійснювати піднесення числа в довільну ступінь двійки.

На основі цієї властивості полів Галуа розроблено та досліджено метод паралельного обчислення експоненти на кінцевих полях Галуа, який відрізняється тим, що в рамках кожного із *k* обчислювальних процесів здійснюється піднесення числа в ступінь, код якої містить n/k значущих бітів експоненти, розділених k-1 нульовими бітами, за рахунок чого кількість операцій множення і піднесення в ступінь двійки на полі Галуа зменшується в *k* раз.

Теоретично показано та експериментально підтверджено, що розроблений метод дозволяє в $2 \cdot k$ раз прискорити комп'ютерну реалізацію важливої для криптографічних застосувань операцію обчислення експоненти на полях Галуа на *k*-ядерних процесорних платформах.

Метод орієнтовано на швидкої реалізації криптографічних алгоритмів з відкритим ключами для захисту від стороннього втручання в роботу систем віддаленого контролю та управління, які використовують Інтернет в якості середовища обміну даними.

Література

1. Schneier B. Appllied Cryptography. Protocols, Algorithms and Source Code in C. Wiley, 2015. 784 p.

2. Hachenbergen D., Jungnickel D. Topic in Galois Fields. *AACIM*. 2020. Vol. 29. 785. DOI: 10.1007/978-3-030-60806-4.

3. Ruiz A. L. et al. Two Galois Fields Cryptographic Applications. *Aritmetic and Algebraic Circuits*. 2021. Vol. 201. P. 551–565. DOI: 10.1007/978-3-030-67266-9_12.

4. Zholubak M., Hlukhov V. S. Galua Field Multipliers Core Generator. *International Journal of Computer Network and Information Security*. 2023. Vol. 15, no. 3. P. 1–14. DOI: 10.5815/ ijcnis. 2023.03.01.

5. Марковський О. П., Нікольський С.С. Метод швидкого обчислення експоненти на полях Галуа $GF(2^n)$ для криптографічних застосувань. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2025. Вип. 58. С. 188–196. DOI: 10.36910/6775-2524-0560-2925-58-23/.

6. Paar C., Fleischmann P., Rodriguez P.S. Fast aritmetic for public-key algoritms in Galois Fields with composite exponentes. *IEEE Transaction on Computers*. 1999. Vol. 48, no. 10. P. 1025–1034.

7. Hlukhov V. et al. Galoise Fields elements processing unit for cryptographic data protection in Ceber-phisical systems. *Advances in cyber-phisical systems*. 2017. Vol. 2, no. 2. P. 47–53.

8. Fitzpatrick P., Popovici E. M. Algorithm and Architecture for a Galois Fiels multiplicative Arithmetic Processor. *IEEE Trans. on Information Theory*. 2003. Vol. 49, no. 12. P. 3303–3307.

9. Марковський О. П., Дайко І. В. Метод швидкого експоненціювання на полях Галуа для систем криптографічного захисту інформації. *Проблеми управління та інформатизації*. 2024. Вип. 1(77). С. 80–88. DOI: 10.18372/2073-4751.77.18660.

10. Daiko I., Selivanov V. Fast exponential method on Galois field for cryptographic applications. *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)* : proceedings, Athens, Greece, 13–15 October 2023 / IEEE. 2023. P. 1–4. DOI: 10.1109/DESSERT61349.2023.10416519.

11. Dychka I. et al. Method of Performing Operations on the Elements of $GF(2^m)$ Using a Sparse Table. *International Journal of Computer Network and Information Security*. 2024. Vol. 16, no 1. P. 61–72. DOI: 10.5815/ijcnis.2024.01.05.

12 Al-Mrayt Ghassan Abdel Jalil Halil, Markovskyi O., Stupak A. Organization of fast exponentiation on Galois Fields for cryptographic data protection systems. *Information, Computing and Intelligent systems*. 2022. № 3. P. 17–25. DOI: 10.20535/2708-4930.3.2022.

13. Верба О. А., Нікольський С. С. Метод експоненціювання на полях Галуа для швидкої реалізації криптографічного захисту в IoT. *Проблеми управління та інформатизації*. 2024. Вип. 4(80). С. 21–31. DOI: 10.18372/2073-4751.80.19767.

14. Марковський О. П., Лефтеріс З., Міщенко Л. Д. Спосіб прискореного обчислення модулярної експоненти. *Вісник Національного технікного університету України "КПІ" Інформатика, управління та обчислювальна техніка*. 2017. № 65. С. 110–115.

Марковський О.П., Нікольський С.С.

МЕТОД РОЗПОДІЛЕНОГО ОБЧИСЛЕННЯ НА БАГАТОЯДЕРНИХ ПРОЦЕСОРАХ ЕКСПОНЕНТИ НА ПОЛЯХ ГАЛУА ДЛЯ КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ

Запропоновано організацію швидкого паралельного обчислення експоненти на полях Галуа для криптографічних застосувань на багатоядерних процесорах. Запропонований підхід до розпаралелювання базується на можливості піднесення числа в будь-яку ступень двійки на полях Галуа, з використанням передобчислень за однаковий час. Запропоновано дві схеми організації паралельних обчислювальних процесів експоненціювання на полях Галуа. Для обох схем детально розроблено процедури обчислення часткових експонент на k ядрах процесора та наведені числові приклади, що ілюструють їх роботу.

Теоретично доведено та експериментально підтверджено, що запропонований метод розподіленого експоненціювання на полях Галуа на багатоядерних процесорах дозволяє за рахунок розпаралелювання та застосування передобчислень скоротити час обчислення в 2·k раз.

***Ключові слова**: мультиплікативні операції на полях Галуа; криптографічні алгоритми на основі алгебри полів Галуа; експоненціювання на полях Галуа; паралельні обчислення.*

Markovskyi O.P., Nikolskyi S.S.

METHOD OF DISTRIBUTED CALCULATION ON MULTI-CORE PROCESSORS EXPONENTIAL ON GALOIS FIELDS FOR CRYPTOGRAPHIC APPLICATIONS

The organization of rapid parallel calculation of the exponential on Galois fields for cryptographic applications on multi-core processors is proposed.

The proposed approach to parallelization is based on the possibility of exponentiation a number to any degree of two on Galois fields by using precomputations over the same time. Two schemes of organization of parallel computing processes of exponentiation on Galois fields of were proposed. For both schemes, the procedures for calculating partial exponential on k processor cores are developed in detail and numerical examples that illustrate their work are given.

It has been theoretically proven and experimentally confirmed that proposed method of distributed exponentiation on Galoise fields on multi-core processors allows, due to the parallelization and use of precomputation, to reduce the calculation time by 2·k times.

***Keywords:** multiplication operation on Galois fields; cryptographic algorithms based on Galois Fields algebra; Galois Fields exponentiation; parallel computations.*