УДК 004.056.5

DOI: 10.18372/2073-4751.77.18660

Марковський О.П., к.т.н.,
orcid.org/0000-0003-3483-4233,
e-mail: markovskyy@i.ua,
Дайко І.В.,
orcid.org/0000-0002-5316-7080,
e-mail: igordaiko1604@gmail.com

МЕТОД ШВИДКОГО ЕКСПОНЕНЦІЮВАННЯ НА ПОЛЯХ ГАЛУА ДЛЯ СИСТЕМ КРИПТОГРАФІЧНОГО ЗАХИСТУ ДАНИХ

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

Вступ

Поява та швидкий розвиток хмарних технологій надали широкому колу користувачів можливості доступу до значних за обсягом обчислювальних потужностей, дозволивши реалізовувати якісно більш складні задачі та проекти прикладного характеру. З іншого боку, доступ до нових можливостей, які надають сучасні хмарні технології, отримали хакери, які активно використовують ці можливості для порушення існуючих на сьогоднішній день криптографічних механізмів захисту. Відомо, що задачі підбору ключів добре розпаралелюються і ефективно вирішуються на багатопроцесорних системах. В певному плані можна говорити про те, що розвиток хмарних технологій порушив існуючий баланс між затратами ресурсів на створення захисту та ресурсами на його порушення [1]. Такий стан потребує адекватних заходів для підсилення рівня захищеності існуючих криптографічних механізмів захисту даних.

Більшість існуючих протоколів захисту даних мають за основу криптографічні алгоритми з відкритим ключем, базовою обчислювальною операцією яких є модулярне експоненціювання, що виконується над числами, розрядність яких значно перевищує розрядність процесорів. Для алгоритмів цього класу єдиний шлях підвищення рівня захищеності полягає в збільшенні розрядності чисел. Проте це має наслідком суттєве збільшення об'єму обчислень і, відповідно, уповільнення реалізації функцій захисту [2].

Інший варіант вирішення зазначеної проблеми полягає в розширенні використання альтернативних алгебр і, зокрема алгебри кінцевих полів Галуа $GF(2^n)$, мультиплікативні операції в якій виконуються значно швидше. Для більш повного використання переваг алгебри кінцевих полів Галуа в криптографічних застосуваннях потрібно віднайти методи ефективної реалізації в цій алгебрі базової операції – експоненціювання на полях Галуа.

Таким чином, наукова задача прискорення обчислювальної реалізації експоненціювання на кінцевих полях Галуа є актуальною та значимою для практики на сучасному етапі розвитку інформаційних та комп'ютерних технологій.

Аналіз відомих методів швидкого експоненціювання на полях Галуа

Тенденція розширення використання експоненціювання на полях Галуа в сучасних механізмах криптографічного захисту інформації стимулює інтенсивні дослідження направлені на прискорення виконання мультиплікативних операцій над числами, розрядність яких значно перевищує розрядність процесора. Для прискорення виконання операцій цього класу на полях Галуа активно досліджуються можливості оптимізації обчислювального процесору, його розпаралелювання на багатоядерних комп'ютерних платформах, використання передобчислень та апаратної реалізації швидкого множення на полях Галуа.

При переході від традиційної алгебри до алгебри полів Галуа операція арифметичного додавання замінюється на логічне додавання (*XOR*), яке позначається символом '⊕', арифметичне множення трансформуться в поліноміальне множення (множення без переносів), яке позначається символом '⊗' [3]. Операція арифметичного піднесення числа А до степені Е, яка традиційно позначається як A^E при переході в алгебру полів Галуа позначається як $A\,|^{E}$. Операція редукції числа *Y* по модулю *M*: *Y mod M* (віднаходження залишку від ділення *Y* на число М) змінюється на редукцію на полях Галуа, що позначається як *Y rem P* і означає віднаходження залишку від поліноміального ділення поліному *Y*(*x*), що співвідноситься з числом *Y*, на утворюючий поліном *P*(*x*) поля Галуа. Відповідно, операція модулярного експоненціювання, тобто обчислення $A^E \bmod M$, при переході до алгебри кінцевих полів Галуа, трансформується в експоненціювання на полях Галуа, яка позначається як $A\,|^{E}\ rem\ P$ [4].

Обчислення експоненти на полях Галуа $A\,|^{E}\ rem\ P$, як і модулярної експоненти $A^E \bmod M$ в традиційній алгебрі організується у вигляді *n* циклів, операції в яких визначаються значеннями відповідних бітів коду експоненти. Відповідно, існує два різновиди алгоритмів експоненціювання: з молодших та зі старших розрядів коду експоненти Е. При експоненціюванні з молодших розрядів в кожному *j*-тому із циклів, *j*=0,2,…,*n*-1, виконується обчислення $A^{2^j} remP$, шляхом піднесення до квадрату попереднього результату, та множення проміжного результату *Y* на $A^{2^j} remP$, якщо поточний розряд e_j коду експоненти дорівнює одиниці. Істотна перевага експоненціювання з молодших розрядів коду експоненти полягає в тому, що існує принципова можливість розпаралелювання, тобто одночасного виконання операцій множення *Y* на $A^{2^j} remP$ та обчислення наступного значення $A^{2^j} remP$ [5]. Це дозволяє практично вдвоє прискорити час експоненціювання при його реалізації на багатоядерних процесорах.

При експоненціюванні зі старших розрядів коду експоненти, в кожному із *n* циклів здійснюється піднесення до квадрату попереднього проміжного результату *Y* та множення його на число А за умови, що поточний біт e_j коду експоненти дорівнює одиниці. Обидві ці операції виконуються послідовно.

Відповідно, середній час T_1 обчислення експоненти на полях Галуа з молодщих розрядів коду експоненти при організації розпаралелювання визначається формулою [6]:

$$T_1 = 0.5 \cdot n \cdot (t_s + t_m), \qquad (1)$$

де t_s – час піднесення до квадрату на полях Галуа, а t_m – час множення на полях Галуа. На практиці $t_s \leq t_m$. Середній час T_2 експоненціювання на полях Галуа зі старших розрядів коду експоненти або обчисленні експоненти з молодших розрядів без розпаралелювання, визначається формулою:

$$T_2 = n \cdot t_s + 0.5 \cdot n \cdot t_m\ . \qquad (2)$$

Якщо операція піднесення до квадрату на полях Галуа виконується так само, як і операція множення, то формула (2) трансформується до наступного вигляду:

$$T_2 = 1.5 \cdot n \cdot t_m. \qquad (3)$$

З викладеного вище можна зробити висновок про те, що реально не існує шляхів прискорення експонентціювання на полях Галуа на рівні класичних алгоритмів його реалізації. Це означає, що прискорення операції експоненціювання на полях Галуа може буди досягнуто за рахунок зменшення часу виконання самих мультиплікативних операцій на полях Галуа: множення та піднесення до квадрату [7].

Ці операції складаються з двох фаз: поліноміального множення (поліноміального піднесення до квадрату) та редукції, тобто віднаходження залишку поліноміального ділення результату першої фази на утворюючий поліном Р(х) поля Галуа. Операція поліноміального множення *n*-розрядних чисел потребує для обчислення

добутку $0.5 \cdot n$ операцій логічного додавання і n операцій зсуву та n операцій тестування значення біту. Беручи до уваги, що час виконання команди логічного додання приблизно однаковий з часом виконання команди зсуву, можна вважати, що реалізація поліноміального множення визначається часом виконання $2.5 \cdot n$ логічних операцій.

Операція поліноміальної редукції здійснюється шляхом додавання числа, що співвідноситься з утворюючим поліномом до поточного залишку. Ця операція включає в себе визначення позиції старшого розряду поточного залишку, зсув коду утворюючого поліному, логічного додавання його до поточного залишку. Таким чином, для виконання редукції потрібно здійснити в середньому n операцій тестування біту, $2 \cdot n$ операцій зсуву (зсувається код утворюючого поліному та тестового коду, що містить одну одиницю), а також $0.5 \cdot n$ операцій логічного додавання. Загальна середня кількість логічних операцій для виконання редукції шляхом ділення поліномів становить $3.5 \cdot n$. Таким чином, загальна кількість логічних операцій, потрібних для реалізації множення n–розрядних чисел на полях Галуа, що утворюються поліномом $P(x)$ ступеню n становить $6 \cdot n$.

Операція поліноміального множення зводиться до логічного додавання максимум n зсунутих відповідним чином кодів множимого, тобто теоретично мінімальний час виконання цієї операції визначається часом $log_2 n$ операцій логічного додавання. Вважаючи на те, що в реальних застосуваннях значення n складає декілька тисяч, вказаний підхід до прискорення поліноміального множення може біти застосований лише в рамках апаратних реалізацій [8].

В якості основного резерву прискорення множення на полях Галуа більшість дослідників розглядають операцію редукції, тобто подальше зменшення часу множення на полях Галуа досягається за рахунок прискорення редукції. Більшість відомих методів [7-9] базується на використанні передобчислень, залежних від незмінного поліному P(x), який в системах криптографічного захисту інформації є частиною відкритого ключа і, відповідно, змінюється рідко.

В методах прискорення, що базуються на використанні цієї властивості утворюючого поліному попередньо обчислюються залишки від ділення кодів $2^{n+1},\ldots,2^{2 \cdot n}$ на утворюючий поліном $P(x)$: $Q_1 = 2^{n+1}\ rem\ P$, $Q_2 = 2^{n+2}\ rem\ P,\ldots,Q_n = 2^{2 \cdot n}\ rem\ P$. Обчислені коди зберігаються в табличній пам'яті передобчислень. Редукція зводиться до додавання табличних кодів, які співвідносяться з одиницями в старших n розрядах коду поліноміального добутку. Для цього потрібно здійснювати аналіз старших n розрядах коду поліноміального добутку, що потребує $2 \cdot n$ логічних операцій (n операцій тестовування значення біту та n операцій зсуву тестового коду). Ще $0.5 \cdot n$ операцій потрібно, в середньому, для додавання результатів передобчислень. Таким чином, за рахунок використання передобчислень можна зменшити середню кількість логічних операцій для реалізації редукції до $2.5 \cdot n$. При цьому загальна середня кількість логічних операцій для множення на полях Галуа становить $5 \cdot n$.

Інший шлях прискорення множення на полях Галуа полягає в суміщенні виконання обох його фаз: поліноміального множення та редукції з використанням технології Монтгомері [11]. В роботі [12] запропонована модифікація відомої в традиційній алгебрі технології Монтгомері до особливостей алгебри полів Галуа. При застосуванні модифікованої під особливості полів Галуа технології Монтгомері, середню кількість логічних операцій для обчислювальної реалізації множення на полях Галуа вдалося знизити до рівня $4.5 \cdot n$.

На основі проведеного огляду та аналізу літературних джерел можна зробити наступні висновки. Розроблені до теперішнього часу методи виконання базової для широкого кола криптографічних застосувань операції експоненціювання на кінцевих полях Галуа не забезпечують прийнятної для широкого кола практичних

застосувань швидкості обчислювальної реалізації.

Аналіз обох класичних алгоритмів експоненціювання на кінцевих полях Галуа показує, що 67% об'єму обчислень приходиться на операції піднесення до квадрату. Тому найбільш перспективним шляхом прискорення цих важливих для криптографічних застосувань обчислень є зменшення часу програмної та апаратної реалізації домінуючої операції піднесення до квадрату на полях Галуа.

Мета досліджень

Мета досліджень полягає в прискоренні обчислювальної реалізації базової для широкого кола практичних застосувань операції експоненціювання на кінцевих полях Галуа за рахунок комплексного використання властивостей поліноміального квадрату, передобчислень, що залежать тільки від утворюючого поліному поля та редукції Монтгомері .

Теоретичне обґрунтування можливості прискорення піднесення до квадрату на полях Галуа

Для досягнення поставленої мети пропонується метод швидкого піднесення до квадрату на кінцевих полях Галуа. Метод має за основу властивість поліноміального квадрату, яка полягає в тому, що поліноміальний квадрат $B|^2 = B \otimes B$ числа $B = b_{n-1}\cdot 2^{n-1} + b_{n-2}\cdot 2^{n-2} + \ldots + b_2\cdot 2^2 + b_1\cdot 2 + b_0$, де $\forall j \in \{0,1,\ldots,n\text{-}1\}$: $b_i \in \{0,1\}$ дорівнює числу $b_{n-1}\cdot 2^{2\cdot(n-1)} + b_{n-2}\cdot 2^{2\cdot(n-2)} + \ldots + b_1\cdot 4 + b_0$, що означає, що операція поліноміального піднесення до квадрату зводиться до вставки нулів між двійковими розрядами $b_{n-1},b_{n-2},\ldots,b_2,b_1,b_{0-1}$ числа B. Вказана властивість може бути доволі просто доведена наступним чином: нехай поліном $B(x) = b_{n-1}\cdot x^{n-1} + b_{n-2}\cdot x^{n-2} + \ldots + b_2\cdot x^2 + b_1\cdot x + b_0$ містить два не рівні нулю коефіцієнти b_l та b_q: $b_l=b_q=1$, $l,q \in \{0,1,\ldots,n\text{-}1\}$. Якщо $l \neq q$, то поліноміальний квадрат $B(x)\otimes B(x)$ містить дві компоненти x^{l+q} та x^{q+l} , логічна сума яких дорівнює нулю; якщо $l=q$, то поліноміальний квадрат включає лише одну компоненту $x^{2\cdot q}$. Це означає, що квадрат поліному містить лише компоненти, що є квадратами компонентів заданого поліному. Наприклад, якщо $B= 13 = 1101_2$, то його поліноміальне представлення має вигляд: $B(x) = x^3+x^2+1$. Відповідно, поліноміальний квадрат цього числа може бути представлений у вигляді: $B(x)\otimes B(x) = (x^3+x^2+1)\cdot(x^3+x^2+1)=$
$x^6+x^5+x^3+x^5+x^4+x^2+x^3+x^2+1=x^6+x^4+1$. Вставка "нулів" між двійковими розрядами числа B дає аналогічний результат: $B\,|^2 = 1\ 0\ 1\ 0\ 0\ 0\ 1_2 = 81$.

З наведеного випливає, що поліноміальне множення не потребує для своєї реалізації ніяких операцій крім зсувів [13]. Якщо об'єднувати першу та другу фази піднесення до квадрату на полях Галуа з використанням технології Монтгомері, то з відомого [11] алгоритму множення на полях Галуа можна виключити дві групи дій: аналіз значення поточного розряду множника та логічне додавання множимого в разі якщо поточний біт дорівнює одиниці. Це дозволяє скоротити кількість логічних операцій для реалізації піднесення до квадрату на полях Галуа в порівнянні з множенням різних чисел. Якщо вважати, що значення бітів чисел великої розрядності з рівною ймовірністю дорівнюють нулю або одиниці, то середня кількість операцій логічного додавання, які можна виключити завдяки використанню наведеної властивості поліноміального квадрату становить $0.5\cdot n$. Крім того, виключаються операції зсуву множимого та аналізу його молодшого біту, які виконуються на кожному з n циклів. Тобто, за рахунок виключення операцій, пов'язаних з аналізом розрядів множника і додавання множимого можна скоротити $2.5\cdot n$ логічних операцій [14].

Проте використання специфічної властивості поліноміального квадрату дозволяє не тільки виключити певні обчислювальні операції в порівнянні з множенням на полях Галуа, але й відкриває можливості для прискорення за рахунок передобчислень. Оскільки з обчислювального процесу виключаються дії, пов'язані з аналізом множника, то корекція Монтгомері може проводитися відразу з урахуванням значень двох розрядів поточного коду залишку. Останні два розряди поточного

коду залишку можуть приймати лише чотири можливих значення. При нульових їх значення корекція Монтгомері не виконується. Оскільки утворюючий поліном P(x) = $p_{n-1}\cdot x^{n-1} + p_{n-2}\cdot x^{n-2} + \ldots + pb_2\cdot x^2 + p_1\cdot x + p_0$ будь-якого поля Галуа є простим, тобто таким, що не можу бути утвореним в результаті множення будь-яких інших поліномів [3], то його компонента $p_0 = 1$. За цієї умови неважно показати, що при будь-якому значенні компоненти p_1, молодші компоненти поліномів $P(x)$, $2\otimes P(x)$ та $3\otimes P(x)$ не рівні між собою і приймають значення із множини Ω={<0,1>, <1,0>, <1,1>}. З цього випливає, що при будь-яких значеннях двох молодших розрядів поточного результату вони можуть бути компенсовані логічним додаванням одного із трьох кодів $P(x)$, $2\otimes P(x)$ або $3\otimes P(x)$. Таким чином, теоретично доведено, що при використанні специфічної властивості поліноміального квадрату існує можливість обробки відразу двох розрядів коду поточного результату піднесення до квадрату, що, в свою чергу, дозволяє вдвічі зменшити кількість циклів піднесення до квадрату на полях Галуа у порівнянні з множенням різних чисел.

Метод прискореного піднесення до квадрату на полях Галуа

Для організації прискореної редукції поліноміального квадрату з обробкою двох розрядів, пропонується попередньо створити таблицю передобчислень, яка складається з трьох значень: T_{01},T_{10},T_{11}. Оскільки утворюючий поліном P(x)=$p_n\cdot x^n + p_{n-1}\cdot x^{n-1}+\ldots+ p_1\cdot x+p_0$ поля Галуа є простим, тобто не може бути представленим у вигляді добутку двох поліномів, то p_0=1 число P= $p_n\cdot 2^n + p_{n-1}\cdot 2^{n-1}+\ldots+ p_1\cdot 2+p_0$ непарне. Якщо p_1=0, то T_{01} = P, T_{10} =2·P, T_{11} = P⊕ 2·P = 3⊗P. Якщо p_1=1, то T_{11} = P, T_{10} =2·P, T_{11} = P⊕ 2·P = 3⊗P. Таким чином табличне значення T_{01} – це число, що є лінійною комбінацією P, молодший розряд якого дорівнює одиниці, а попередній – нулю. Аналогічно, T_{10} – це подвоєне значення P, молодший розряд T_{10} дорівнює нулю, а той, що передує йому – одиниця. Код T_{11} = 3⊗P, два молодші розряди якого дорівнюють одиниці.

Очевидно, що табличні значення T_{01},T_{10}, T_{11} залежать лише від утворюючого поліному поля Галуа, який, в криптографічних застосуваннях є частиною відкритого ключа, і, відповідно може розглядатися як постійна величина. Це означає, що таблиця передобчислень обчислюється лише один раз.

Розроблена процедура прискореного піднесення до квадрату $A|^2$ *rem* P зводиться до наступної послідовності дій.

1. Утворити $2\cdot n$-розрядне число B з n-розрядного числа A шляхом вставки нулів між значущими розрядами числа A. Технологічно ця операція може бути швидко виконана з використанням таблиць заміщення.

2. Якщо n парне, тобто n *mod* 2 = 0, перехід на п.4. Якщо n непарне і молодший розряд B дорівнює одиниці, тобто b_0=1, то виконали логічне додавання до B числа P: B = B ⊕ P.

3. Зсув праворуч коду B: $B = B >> 1$. Декремент n: $n = n$-1.

4. Початкове значення лічильника j циклів встановлюється в n/2.

5. Якщо два молодші розряди проміжного результату B дорівнюють нулю, тобто b_0=b_1=0, то перехід на п.6. Якщо b_1=0 і b_0=1, то виконати логічне додавання до коду B значення T_{01}: B=B⊕T_{01}, після чого здійснити перехід на п.6. Якщо b_1=1 і b_0=0, то виконати логічне додавання до коду проміжного результату B значення T_{10}: B=B⊕T_{10} і перейти на п.6. Якщо два молодших розряди B дорівнюють одиниці, тобто b_1=1 і b_0=1, то додати до коду B значення T_{11}: B=B⊕T_{11}.

6. Виконати зсув праворуч на два двійкових розряди коду B: $B = B >> 2$.

7. Декремент лічильника циклів j: $j = j$-1. Якщо j більше нуля, тобто $j > 0$, перехід на повторне виконання п.5.

8. Якщо n–тий біт отриманого результату B дорівнює одиниці, тобто b_n=1, то здійснити логічне додавання до нього коду P: B=B⊕P.

В результаті виконання описаної процедури отримується код B, який являє собою добуток $A \otimes A \otimes R^{-1}$ *rem P*. Для отримання правильного значення квадрату числа A, обчислене значення B потрібно помножити на *R*: $B \otimes R$ *rem* $P = A \otimes A$ *rem P*.

Робота запропонованого методу прискореного обчислення квадрату на полі Галуа може бути ілюстрована наступним прикладом. Нехай, n=5, а утворюючий поліном поля Галуа має наступний вигляд: $P(x)=x^5 + x^2+1$. Наведений утворюючий поліном співвідноситься з числом P=37 = 100101_2. Відповідно, передбачені пропонованим методом передобчислення, що залежать тільки від утворюючого поліному поля Галуа мають наступний вигляд: $T_{01} = P = 37 = 100101_2$, $T_{10} = 2 \cdot P = 74 = 1001010_2$, $T_{11} = 3 \otimes P = 111 = 1101111_2$. Для n=5 технологія Монтгомері передбачає використання допоміжного поліному $R(x)=x^5$, з яким співвідноситься число $\underline{R}=2^5 = 32$. Його мультиплікативна інверсія R^{-1} на полі Галуа, утворена поліномом, яке співвідноситься з числом P=37 дорівнює R^{-1} =23, тобто $R \otimes R^{-1}$ *rem* $P = 32 \otimes 23$ *rem* 37 =1.

При обчисленні квадрату числа A=25=11001_2 : A⊗A *rem P* = 25⊗25 *rem* 37 згідно п.1 формується число B у вигляді: B = 101000001_2 = 321.

Оскільки n=5 непарне, і молодший розряд B дорівнює одиниці, то в рамках п.2 описаної процедури здійснюється логічне додавання до коду B числа P: B = 101000001⊕100101 = 101100100. Після цього, згідно п.3 виконується зсув праворуч коду B: 10110010, а n зменшується на одиницю: $n = n$-1 =4.

Наступним п.4 процедури лічильник j циклів встановлюється рівним $j=n/2=2$.

В рамках п.5 перевіряється значення двох молодших розрядів коду B: оскільки b_1=1 і b_0=0, то здійснюється логічне додавання до коду проміжного результату B значення T_{10}: $B=B \oplus T_{10}$ = 10110010 ⊕ 1001010 = 11111000. В наступному п.6 реалізується зсув B праворуч на 2 розряди: B = 111110. Далі в п.7 зменшується на одиницю лічильник циклів j: $j=j$-1=1. Так як j>0, то здійснюється повернення на повторне виконання п.5

В п.5 знов аналізується значення двох молодших розрядів коду B: оскільки b_1=1 і b_0=0, то здійснюється логічне додавання до коду проміжного результату B значення T_{10}: $B=B \oplus T_{10}$ = 111110 ⊕ 1001010 = 1110100. В наступному п.6 реалізується зсув коду B праворуч на 2 розряди: B = 11101. Далі в п.7 зменшується на одиницю лічильник циклів j: $j=j$-1=0. Так як після декременту j =0, то наступним п. 8 перевіряється значення старшого розряду отриманого результату B: оскільки b_5=0, то корекція не виконується.

В результаті виконання процедури отримано код B = $11101_2 = 29_{10}$. який являє собою добуток $A \otimes A \otimes R^{-1}$ *rem P* = 25⊗25⊗23 *rem* 37. Для отримання правильного значення квадрату числа A на полі Галуа, утвореного поліномом $P(x)=x^5+x^2+1$, отримане значення B потрібно помножити на *R*, яке для цього прикладу дорівнює 32: $B \otimes R$ *rem* $P = 29 \otimes 32$ *rem* 37 = 6.

Аналіз ефективності

Множення на полях Галуа складається двох з базових операцій: логічного додавання та зсувів. Час виконання обох цих операцій приблизно однаковий тому, оцінювання часу множення може здійснюватися через кількість таких операцій.

Найбільш швидким варіантом множення на полях Галуа $A \otimes B$ *rem* $\underline{P}$ є використання для реалізації редукції технології Монтгомері [11]. В цьому варіанті для множення виконується n циклів, в кожному з яких виконується операції зсуву множника A і поточного проміжного результату. Крім того, в залежності від поточного біту b_0 множника та молодшого біту проміжного результату, виконуються операції логічного додавання множимого та числа P, що співвідноситься з утворюючим поліномом поля Галуа. Таким чином, виконання множення на полях Галуа потребує, в середньому, $4.5 \cdot n$ операцій (в циклі виконуються два зсуви, дві операції тестування бітів і, в середньому, одна операція логічного додавання – *XOR*).

Запропонований метод прискореного піднесення до квадрату на кінцевих полях Галуа передбачає виконання $n/2$ циклів, в кожному з яких виконується одна операція тестування пари бітів, одна операція зсуву і в трьох ситуаціях із чотирьох – логічне додавання одного із табличних значень: T_{01}, T_{10} або T_{11}. Тобто, в середньому, в рамках одного циклу здійснюється 2.75 операцій, а всього, для операції піднесення до квадрату за запропонованим методом – $1.375 \cdot n$ операцій. На практиці значення n вимірюється тисячами (2048 або 4096), тому допоміжні одиночні операції, що виконуються до і після циклів можна не враховувати.

Таким чином, запропонований метод піднесення до квадрату на полях Галуа, який ґрунтується на властивості поліноміального квадрату, редукції Монтгомері та обробці відразу пари розрядів дозволяє в 4.5/1.375 = 3.3 рази прискорити виконання піднесення до квадрату в порівнянні з найбільш швидкодіючим варіантом множення на полях Галуа.

Як зазначалося вище, при використанні класичного алгоритму експоненціювання зі старших розрядів коду експоненти середній час T_2 виконання визначається формулою (2). Якщо вважати, що час t_S виконання операції піднесення до квадрату на полях Галуа за запропонованим методом в 3.3 менша за час множення на полях Галуа, то середній час T_3 експоненціювання визначається формулою:

$$T_3 = n \cdot t_s + 0.5 \cdot n \cdot t_m = \\ = n \cdot 0.3 \cdot t_m + 0.5 \cdot n \cdot t_m = 0.8 \cdot n \cdot t_m \quad . \quad (10)$$

Якщо операція піднесення до квадрату на полях Галуа виконується так само, як і операція множення, то згідно (3) середній час експоненціювання $T_2 = 1.5 \cdot n \cdot t_m$. Відповідно, застосування запропонованого методу для швидкого піднесення до квадрату на полях Галуа дозволяє в η раз прискорити експонентціювання, причому чисельне значення коефіцієнту η прискорення визначається формулою:

$$\eta = \frac{T_2}{T_3} = \frac{1.5 \cdot n \cdot t_m}{0.8 \cdot n \cdot t_m} = 1.875 . \quad (11)$$

Таким чином, запропонований метод швидкого піднесення до квадрату на полях Галуа з обробкою двох розрядів проміжного результату дозволяє майже вдвоє прискорити обчислювальну реалізацію експоненціювання на полях Галуа. Проведені експериментальні дослідження програмної реалізації експоненціювання на полях Галуа з використанням запропонованого методу показали, що реальне прискорення близьке до значення (11) отриманого теоретичним шляхом.

Можливість обробки пари розрядів проміжного результату зумовлена тим, що в розробленому методі використана специфічна властивість поліноміального квадрату. В методі не передбачається аналіз розрядів числа, що підноситься до квадрату, Це робить можливість введення корекції Монтгомері в залежності від значень лише пари молодших розрядів проміжного результату. Це, в свою чергу відкрило можливості використання передобчислень, які залежать лише від утворюючого поліному поля Галуа, який в системах криптографічного захисту є частиною відкритого ключа і, відповідно, може розглядатися як незмінний. Відповідно, перед обчислення можна виконувати лише один раз і час їх виконання на впливає на час експоненціювання на полях Галуа, яке здійснюється при кожному застосуванні криптографічних механізмів захисту.

Висновки

В результаті проведених досліджень, направлених на прискорення обчислювальної реалізації базової для криптографічних застосувань операції експоненціювання на кінцевих полях Галуа, теоретично обґрунтовано, розроблено та досліджено метод виконання цієї операції, який дозволяє в 1.875 раз, тобто майже вдвоє, зменшити кількість потрібних для її реалізації обчислень.

Ефект прискорення обчислення експоненти на полях Галуа, досягнутий в запропонованому методі базується на

зменшенні об'єму обчислень при виконанні піднесення до квадрату на полях Галуа, який зумовлений використанням специфічних властивостей поліноміального квадрату, адаптованою для полів Галуа технології редукції Монтгомері та передобчислень, які залежать від утворюючого поліному поля. Останній в практичних застосуваннях являє собою частину відкритого ключа криптографічних механізмів і, відповідно, є практично незмінним, що дозволяє багато раз використовувати результати передобчислень. Показано, що за рахунок комплексного використання в запропонованому методі наведених вище чинників, об'єм обчислень при піднесенні до квадрату на полях Галуа вдалося зменшити в 3.3 рази в порівнянні з піднесенням до квадрату з застосуванням алгоритму множення на полях Галуа.

Запропонований метод може бути ефективно використаних в системах криптографічного захисту інформації на основі еліптичних кривих та при ідентифікації віддалених користувачів в рамках концепції нульових знань.

Література

1. Thangaval M., Varalakshmi P. Improved secure rsacryptosystem (ISRSAC) for data confidentiality in cloud. *International Journal of Information Systems and Change Management*. 2017. Vol. 9, no. 4. P. 46–53.
2. Schneier B. Applierd Cryptography. Protocols, Algorithms and Source Codes in C. New York : John Wiley & Son, Inc., 2009. 816 p.
3. Николайчук Я. М. Коди полів Галуа: теорія і застосування. Тернопіль : Вид-во ТНУ, 2012. 576 с.
4. Марковський О. П., Захаріудакіс Ліфтеріс, Максимук В. Р. Використання алгебри полів Галуа для реалізації концепції «нульових знань» при ідентифікації та автентифікації віддалених користувачі. *Електронне моделювання*. 2017. Т. 6, № 39. С. 33–45.
5. Марковський О. П., Саїдреза Махмали, Ісаченко Г. В. Технологія цифрового підпису DSA на основі арифметики полів Галуа. *Вісник національного технічного університету України "КПІ". Інформатика, управління та обчислювальна техніка*. 2012. № 55. С. 34–41.
6. Калмиков І. А., Степанова Е. С., Тинчеров К. Т. Розробка методу нелінійного шифрування інформації з використанням операції піднесення до степеня для кінцевого поля Галуа. *Сучасні наукомісткі технології*. 2019. № 9. С. 84–89.
7. 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.
8. Wu H. et al. Finite field multiplier using redundant representation. *IEEE Trans. Computers*. 2002. Vol. 51, no. 5. P. 1306–1316.
9. Daiko I., Selivanov V. Fast exponential method on Galois fields for cryptographic applications. *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)* : proceedings, Athens, Greece, 13–15 October, 2023 / IEEE. Danvers, 2023. P. 1–4. DOI: 10.1109/DESSERT61349.2023.10416519.
10. Osadchyy V. The Order of Edwards and Montgomery Curves. *WSEAS Transactions on Mathematics*. 2020. Vol. 19, no. 25. P. 253–264.
11. Кот О. С., Марковський О. П. Організація прискореного експоненціювання на полях Галуа з використанням редукції Монтгомері. *Альманах науки*. 2020. № 3(36). С. 34–37.
12. Markovskyi O., Masimyk V., Kot O. The Employment of Montgomery reduction for acceleration of exponent on Galoise fields calculation. *The International Conference on Security, Fault Tolerance, Intelligence" (ICSFTI2020)* : proceedings, Kyiv, Ukraine, 13–14 May, 2020, / Department of Computer Engineering (OT), FIOT, NTUU "Sikorsky KPI". Kyiv, 2020. P. 44–49.
13. Hachez G., Quisquater J.-J. Montgomery multiplication with no final subtraction. *Lecture Notes in Computer Science. Vol. 1965. Cryptographic Hardware and Embedded System – CHES'2000. Second International Workshop Worcester, MA, USA,*

August 17–18, 2000 Proceedings / ed. by Ç. Koç et al. Berlin, 2000. P. 293–301.

14. Elfard S. Justification of Montgomery Modular Reductions. *Advanced Computing*. 2012. No. 11. P. 41–45.

Марковський О.П., Дайко І.В.

МЕТОД ШВИДКОГО ЕКСПОНЕНЦІЮВАННЯ НА ПОЛЯХ ГАЛУА ДЛЯ СИСТЕМ КРИПТОГРАФІЧНОГО ЗАХИСТУ ДАНИХ

Запропоновано та досліджено метод прискореного обчислення експоненти на полях Галуа – базової операції широкого кола алгоритмів криптографічного захисту даних. Прискорення досягнуто за рахунок зменшення часу виконання піднесення до квадрату на полях Галуа, яке базується на використанні властивостей поліноміального квадрату, редукції Монтгомері та використанні передобчислень, які залежать лише від утворюючого поліному поля. Наведено математичне обґрунтування запропонованого метод та числові приклади, які ілюструють його роботу.

Теоретично та експериментально доведено, що запропонований метод дозволяє майже вдвічі прискорити експоненціювання на полях Галуа в порівнянні з відомими методами.

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

Markovskyi O.P., Daiko I.V.

METHOD OF FAST EXPOSURE IN GALOIS FIELDS IN CRYPTOGRAPHIC DATA PROTECTION SYSTEMS

The method of accelerated calculation of the exponent on Galois fields – the basic operation of a wide range of cryptographic data protection algorithms is proposed and investigated. Acceleration is achieved by reducing the execution time of the ascent to the square in Galois fields, which is based on the use of the properties of the polynomial square, Montgomery reduction and the use of precalculations that depend only on the Galois Field base polynomial. The mathematical substantiation of the offered method and numerical examples which illustrate its work are resulted.

It is theoretically and experimentally proved that the proposed method allows to almost twice accelerate the exponentiation on Galois fields in comparison with known methods.

***Keywords**: multiplication operation on Galois fields; cryptographic algorithms based on Galois fields algebra; Galois fields exponentiation; Montgomery reduction.*