DOI: https://doi.org/10.36910/6775-2524-0560-2025-58-23
УДК 004.056.5
Марковський Олександр Петрович, к.т.н., доцент.
https://orcid.org/0000-0003-3483-4233
Нікольський Сергій Сергійович, асистент
https://orcid.org/0000-0003-4893-3339
Національний університет "Київський політехнічний інститут имені Ігоря Сікорського", м.Київ, Україна

МЕТОД ШВИДКОГО ОБЧИСЛЕННЯ ЕКСПОНЕНТИ НА ПОЛЯХ ГАЛУА GF(2^n) ДЛЯ КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ

Марковський О.П., Нікольський С.С. Метод швидкого обчислення експоненти на полях Галуа GF(2^n) для криптографічних застосувань. Розроблено метод прискореного обчислення експоненти на полях Галуа GF(2^n), який реалізує групову обробку *k* розрядів коду експоненти з використання передобчислень, що дозволяє зменшити кількість операцій піднесення до квадрату та множення і, тим самим, прискорити процес обчислення. В теоретичному плані, запропонований метод базується на властивості операції експоненціювання числа на полях Галуа за умови, що код експоненти є ступенем числа два. Детально описано процедури побудови двох таблиць передобчислень, а також процедуру експоненціювання на полях Галуа з їх використанням. Виклад ілюстровано числовими прикладами.

Теоретично показано та експериментально доведено, що розроблений метод забезпечує прискорення обчислення експоненти на полях Галуа практично в *k* раз.

Ключові слова: мультиплікативні операції на полях Галуа, криптографічні алгоритми на основі алгебри полів Галуа, експоненціювання на полях Галуа, передобчислення.

Markovskyi O., Nikolskyi S. Method for fast exponential on Galose Fields GF(2^n) calculation for cryptographic applications. The accelerated calculation method of the exponentiation one Galoise Fields GF (2^n), which sells group processing *k* exponent code bits using precomputations for reducing the number of the square and multiplication operations and, thus, accelerate the calculation process. In the theoretical terms, the proposed method is based on the properties of the operation of exponentiation on Galoise Fields in case the code of exponent is the degree of number two. The procedures for constructing two precomputations tables, as well as the exponentiation on the Galoise Fields procedure with their use, are described in detail. The presentation is illustrated with numerical examples.

It is theoretically shown and experimentally proven that the developed method provides acceleration of the calculation of the exponential on Galoise Fields practically on *k* times.

Key words: multiplication operation on Galois fields, cryptographic algorithms based on Galois Fields algebra, Galois Fields exponentiation, precomputations.

Постановка наукової проблеми. Досягнуте в останнє десятиліття якісне зростання показників швидкодії та надійності Інтернету має наслідком розширення сфери застосування технологій IoT, яке вийшло на сьогоднішній день далеко за межі віддаленого управління побутовими приладами. Зримі переваги використання Інтернету в якості середовища обміну даних з готовою інфраструктурою, такі, як на порядки нижча вартість, висока гнучкість при реконфігуруванні, відсутність обмежень на відстань, стимулюють використання технологій IoT для побудови систем контролю та управління віддаленими об'єктами реального світу в багатьох галузях людської діяльності [1].

Сьогодні технології IoT знаходять широке застосування для віддаленого моніторингу стану здоров'я, об'єктів інфраструктури, дорожнього трафіку, в системах охоронної сигналізації і відео спостереження, а також для дистанційного управління технологічними процесами, транспортними засобами, тощо. При використанні Інтернету в якості середовища обміну даними в цих, та багатьох інших застосуваннях об'єктивно виникає потреба в організації захисту інформаційних потоків від зовнішнього втручання [2].

Для цього на практиці застосовується увесь арсенал сучасних механізмів криптографічного захисту. Для безпеки віддаленого моніторингу та управління на базі IoT найбільше значення мають механізми цифрового підпису, які гарантують автентичність як самих даних, так і їх відправників, в якості яких виступають компоненти системи керування. Базовою обчислювальною операцією цих механізмів витупає модулярне експоненціювання над числами, довжина яких на порядок перевищує розрядність процесора. Це зумовлене тим, що рівень захищеності напряму визначається розрядністю чисел, з якими оперують механізми цифрового підпису. На сьогодні, для більшості застосувань використовуються числа розрядністю 4096. Реалізація модулярного експоненціювання над такими довгими числами потребує сотень мільйонів процесорних операцій і, відповідно,

значних часових ресурсів. Для систем дистанційного управління, що працюють в реальному часі і, в яких в якості термінальних платформ використовуються малопотужні мікроконтролери, час перевірки автентичності команд є критичним. Це зумовлює практичну важливість пошуку шляхів прискорення реалізації механізмів цифрового підпису.

Одним із найбільш перспективних варіантів виступає перехід до альтернативних алгебраїчних базисів, в яких мультиплікативні операції над довгими числами виконуються на порядок простіше. Зокрема, мова йде про алгебру кінцевих полів Галуа GF(2^n), яка широкого використовується в механізмах цифрового підпису на базі ECC [3], а також при ідентифікації учасників віддаленої інформаційної взаємодії [4].

Таким чином, наукова задача прискорення комп'ютерної реалізації експоненціювання на полях Галуа є актуальною та має практичну значимість для сьогоднішнього етапу розвитку інформаційних технологій.

Аналіз досліджень. Використання алгебри кінцевих полів Галуа GF(2^n) в якості математичної основи побудови криптографічних систем з відкритим ключем, має певні переваги в порівнянні з традиційною модулярної арифметикою [5]. Насамперед, це суттєве спрощення всіх операцій: арифметичне додавання замінюється на логічне (XOR), яке позначається символом '$\oplus$', арифметичне множення замінюється поліноміальним, яке позначається символом '$\otimes$'. В цих операціях не використовується перенос і, відповідно, розряди оброблюються автономно. Це спрощує і прискорює виконання цих базових операцій, особливо при апаратній реалізації.

Операція множення на полях Галуа $A \otimes B$ rem P включає, крім поліноміального множення, редукцію отриманого добутку, тобто знаходження остачі поліноміального ділення добутку $A \otimes B$ на утворюючий поліном поля Галуа $P(x)$. Редукція на полях Галуа, яка позначається як 'rem', до певної міри являє собою аналог модулярної редукції, яка позначається як 'mod' [6]. Базовою для криптографічних застосувань операцією на полях Галуа виступає обчислення експоненти: A^E rem P. Ця операція здійснюється над числами, довжина n яких забезпечує необхідний для задач практики рівень захищеності і визначається ступенем n утворюючого поліному $P(x)=x^n + p_n \cdot x^{n-1} + \ldots + p_2 \cdot x + p_1$, $\forall i \in \{1,2,\ldots,n\}$: $p_i \in \{0,1\}$ поля Галуа. На сьогоднішній день для більшості застосувань використовується розрядність 4096 [7].

В силу того. що архітектура сучасних процесорів не пристосована до ефективного виконання мультиплікативних операцій в алгебрі полів Галуа, значна частина публікацій присвячена їх реалізації спеціалізованими апаратними засобами, в більшості випадків з використанням програмованих матриць [8, 9]. Простота та специфічні властивості операцій на полях Галуа дозволяють створювати апаратні засоби криптографічного захисту, які за показниками швидкодії значно перевершують аналоги, що реалізують базові операції криптографії з відкритим ключем в традиційній алгебрі.

Експоненціювання на полях Галуа, так само, як і модулярне експоненціювання, реалізується за одним із двох різновидів класичного алгоритму [10]. За цим алгоритмом процес обчислення організовано у вигляді n-кратного циклу, дії в рамках кожного j-того з яких, $j \in \{1,2,\ldots,n\}$, залежать від значення відповідного j–го двійкового розряду коду експоненти $E = e_n \cdot 2^{n-1} + e_{n-1} \cdot 2^{n-2} + \ldots + e_2 \cdot 2 + e_1$, $e_j \in \{0,1\}$. Згадані вище різновиди відрізняються напрямком проходу розрядів коду експоненти E. Так, якщо при виконанні n циклів індекс j номеру поточного біту коду експоненти E послідовно змінюється від n до одиниці, тобто від старших разрядов до молодших, класичний алгоритм передбачає використання лише одної змінної R, яка перед початком циклів встановлюється в одиницю: R=1. В кожному циклі ця зміна спочатку підноситься до квадрату на полі Галуа: $R = R^2$ rem P, а потім, за умови одиничного значення поточного біту коду експоненти E, множиться на число A: якщо e_j=1 то $R=R \otimes A$ rem P. Після завершення циклів в змінній R сформовано значення $R=A^E$ rem P.

В другому різновиді класичного алгоритму, в якому розряди коду експоненти скануються в напрямку від молодших до старших, індекс j номеру поточного біту коду експоненти E послідовно змінюється від одиниці до n. Цей різновид класичного алгоритму обчислення експоненти на полях Галуа передбачає використання двох змінних: D та R, які на початку встановлюються рівними A та одиниці відповідно: $D=A$, R=1. В кожному циклі спочатку, за умовити, що поточний біт e_j коду E дорівнює одиниці, здійснюється множення на полі Галуа значення R на D: якщо e_j=1, то $R=R \otimes D$ rem P. Після цього, незалежно від значення e_j реалізується піднесення до квадрату змінної D: $D=D^2$ rem P. Після завершення всіх n циклів в змінній R фіксується результат: $R=A^E$ rem P.

Реалізація обох різновидів класичного алгоритму потребує виконання, в середньому $1.5 \cdot n$ операцій множення на полі Галуа n-розрядних чисел.

Аналіз класичного алгоритму обчислення експоненти на полях Галуа свідчить про те, що він має послідовну структуру, яка виключає можливість розпаралелювання обчислень. Тому переважна більшість запропонованих на сьогоднішній день підходів направлені на прискорення базових його мультиплікативних складових: піднесення до квадрату на полях Галуа та множення на полях Галуа.

Ці операції мають в своєму складі дві фази: мультиплікативну, тобто обчислення поліноміального квадрату чи добутку та редукційну, яка полягає в обчисленні залишку поліноміального ділення результату мультиплікативної фази на утворюючий поліном поля Галуа. При застосуванні технології Монтгомері виконання обох зазначених вище фаз суміщається у часі [11].

Основний позитивний ефекти від такого суміщення полягає в тому, що довжина проміжних результатів практично не перевершує n+1, В той час, як при роздільній реалізації фаз поліноміального множення та редукції розрядність проміжного результату становить $2 \cdot n$. з огляду на те, що обробка здійснюється фрагментами, довжина r яких дорівнює розрядності процесора, оперування з майже вдвоє коротшими числами при суміщенні виконання фаз дозволяє суттєво прискорити мультиплікативні операції на полях Галуа.

Множення на полях Галуа $A \cdot B$ rem P за методом Монтгомері організується у вигляді n циклів, з послідовною зміною його індексу j від 1 до n та формуванням результату в змінній R, стартов значення якої дорівнює нулю: R=0. В кожному j–тому циклі виконуються такі дії: якщо j-тий біт множника дорівнює одиниці, до поточного результату R логічно додається множене A: $R=R \oplus A$; якщо після цього молодший розряд R дорівнює одиниці, то до R логічно додається код утворюючого поліному P: $R=R \oplus P$; на кінець, код R зсувається на один біт праворуч : $R=R>>1$. Очевидно, що, в середньому, описана операція потребує $2 \cdot n$ логічних операцій над n-розрядними числами.

В якості основного резерву прискорення обчислення мультиплікативних операцій на кінцевих полях Галуа в відомих рішеннях застосовуються передобчислення, які дозволяють багатократно використовувати попередньо вирахувані результати.

Зокрема, в криптографічних застосуваннях утворюючий поліном $P(x)$ поля Галуа $GF(2^n)$ є компонентою відкритого ключа і, відповідно, його можна вважати практично незмінним [6]. Це означає, що можна певним чином виокремити обчислення, які залежать тільки від $P(x)$, виконати їх лише один раз зі збереженням результатів в постійній пам'яті для їх використанні при кожному множенні на полях Галуа. Такий підхід використовується для прискорення виконання редукції.

Окремий клас складають відомі рішення до прискорення обчислення експоненти на полях Галуа, які базуються на властивості поліноміального квадрату $A \otimes A$. Ця властивість полягає в тому, що поліноміальний квадрат числа $A= a_n \cdot 2^{n-1} + \ldots + a_2 \cdot 2 + a_1$, $\forall i \in \{1,2,\ldots,n\}$: $a_i \in \{0,1\}$ може бути отриманий вставкою нулів між двійковими розрядами числа A. Іншими словами, $A \otimes A= a_n \cdot 2^{2 \cdot (n-1)} + \ldots + a_2 \cdot 2^2 + a_1$. В одному із відомих рішень [12] піднесення до квадрату на полі Галуа $A \otimes A$ rem P зводиться до логічного додавання табличних значень $T[i] = 2^{2 \cdot i}$ rem P для значень i=1,2,…,n для яких a_i=1. Очевидно, що в цьому варіанті піднесення до квадрату на полі Галуа потребує, в середньому, $n/2$ операцій логічного додавання (XOR). Відповідно, час обчислення експоненти на полях Галуа дорівнює $0.5 \cdot n^2 \cdot t_{XOR} + 0.5 \cdot n \cdot t_m$, де t_{XOR} – час виконання операції логічного множення, а t_m – множення на полі Галуа n-розрядних чисел. Головний недолік такого рішення полягає в використанні доволі значного для термінальних мікроконтролерів об'єму пам'яті для зберігання результатів передобчислень: n^2 бітів.

В роботі [13] запропоновано варіант використання властивості поліноміального квадрату для організації прискореного обчислення квадрату на полях Галуа. В силу того, що процес обчислення поліноміального квадрату не передбачає дій, пов'язані з аналізом множника, то корекція Монтгомері здійснюється з урахуванням значень двох розрядів поточного коду результату і, відповідно, зі зсувом відразу на два розряди. Відповідно цей метод передбачає для обчислення квадрату на полях Галуа з використанням $n/2$ циклів, в кожному з яких зі ймовірністю 0.75 виконується логічне додавання одного із трьох табличних значень і дві операція зсуву (поточного результату та числа, яке підноситься до квадрату). Тобто, в середньому, в рамках одного циклу здійснюється 2.75 операцій над n-розрядними числами, а всього, для операції піднесення до квадрату на полі Галуа – $1.375 \cdot n$. Перевага методу полягає в тому, що в таблиці передобчислень зберігається всього три n-розрядних числа.

В роботі [14] запропоновано більш узагальнену версію швидкого піднесення до квадрату на полі Галуа без множення, в якій редукція здійснюється відразу на k розрядів. Це вимагає створення таблиці передобчислень об'ємом 2^k $(n+k)$-розрядних чисел. В одному циклі виконується три операції над n-розрядними числами і, відповідно, загальна середня їх кількість для піднесення до квадрату на полі Галуа складає $3 \cdot n/k$. Оскільки в розглянутому методі час виконання множення на полі Галуа не змінюється, загальний час T_M виконання експоненціювання на полі Галуа становить:

$$T_M = \frac{3 \cdot n^2}{k} \cdot t_{XOR} + \frac{n}{2} \cdot t_m . \qquad (1)$$

Якщо виходити з того, що операція множення на полях Галуа з використанням редукції Монтгомері потребує, в середньому, $2 \cdot n$ операцій типу логічного додавання чи зсуву n–розрядних чисел [13], тобто $t_m = 2 \cdot n \cdot t_{\text{XOR}}$, формула (1) може бути перетворена до вигляду:

$$T_M = \frac{3+k}{k} \cdot n^2 \cdot t_{XOR} . \qquad (2)$$

Виділення невирішених задач. Проведений аналіз відомих рішень, направлених на прискорення комп'ютерної реалізації експоненціювання на полях Галуа показав, що всі вони, в переважній більшості, орієнтовані на зменшення часу виконання складових цієї операції: піднесення до квадрату та множення. Іншими словами, вони не змінюють кількості операцій в процесі обчислення експоненти на полях Галуа. Це суттєвим чином обмежує ефективність відомих методів в плані подальшого прискорення програмної реалізації важливої для криптографічних застосувань операції насамперед на малопотужних термінальних обчислювальних платформах, таких як мікроконтролери систем дистанційного контролю та управління на базі технологій IoT.

Мета досліджень полягає в прискоренні комп'ютерної реалізації обчислення експоненти на полях Галуа шляхом організації групової обробки розрядів коду експоненти з використанням передобчислень

Виклад основного матеріалу й обгрунтування отриманих результатів дослідження. Для досягнення поставленої мети запропоновано метод швидкого обчислення експоненти на кінцевих полях Галуа, який базується на організації групової обробки коду експоненти.

В основі пропонованого методу швидкого обчислення експоненти A^E rem P на кінцевих полях Галуа GF(2^n) лежить версія класичного алгоритму виконання цієї операції, яка передбачає сканування розрядів коду експоненти в напрямку від старших до молодших. Прискорення експоненціювання досягається за рахунок одночасної обробки відразу k розрядів коду експоненти. Для реалізації цієї ідеї запропоновано спеціальні таблиці передобчислень, перша із яких залежить лише від відтворюючого поліному $P(x)$ поля Галуа і дозволяє швидко виконувати піднесення числа R відразу в ступінь 2^k на полі Галуа $R^{2^k} remP$ без обчислення проміжних ступенів: R^2 rem P, R^4 rem P,…, $R^{2^{k-1}} remP$. Друга таблиця передобчислень ступенів A залежить від числа A, яке постійне для кожного із циклів експоненціювання на полях Галуа. Ця таблиця забезпечує можливість суміщення групи операцій множення на постійне число A в рамках обробки k розрядів коду експоненти.

Перша таблиця U редукція ступенів двійки створюється при зміні утворюючого поліному $P(x)$, який є частинного відкритого ключа криптосистеми. На практиці останній змінюється відносно рідко, так, що за періоди між його оновленнями операція експоненціювання на полях Галуа виконується сотні тисяч і мільйони раз. Це означає, що час формування таблиці U практично не впливає на показники ефективності обчислення експоненти на полях Галуа.

Друга таблиця T передобчислень ступенів A залежить від числа A, яке змінюється при кожній операція обчислення експоненти A^E rem P, створюється перед початком циклу експоненціювання.

Для створення таблиці U передобчислень редукцій ступенів двійки, що містить n-1 n-розрядних чисел пропонується процедура, яка передбачає виконання наступної послідовності дій:

1. Індекс j встановлюється в одиницю: j=1; в першу комірку таблиці U[1] заноситься код одиниці: U[1] = 1.

2. Індекс j збільшується на одиницю: j=j+1; якщо $j > n$-1, перехід на кінець;

3. j-тий елемент таблиці U заповнюється значенням (U[j-1]<< 2^k) rem P – залишком від поліноміального ділення на утворюючий поліном $P(x)$ поля Галуа зсунутого на 2^k розрядів ліворуч коду попереднього табличного значення U[j-1]: U[j] = U[j]$\cdot 2^k$ rem P. Повернення на повторне виконання п.2.

Робота запропонованої процедури побудови таблиці передобчислень редукцій ступенів двійки ілюструється наступним прикладом. Нехай, n=12, а утворюючий поліном поля Галуа GF(2^{12}) має наступний вигляд: $P(x) = x^{12} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^2 + x + 1$. Цьому поліному відповідає числове значення P=7079. Кількість k розрядів коду експоненти E, що оброблюються одночасно, нехай дорівнює трьом: k=3. Відповідно до п.1. розробленої процедури, індекс j номеру рядка таблиці встановлюється в одиницю, а в перший рядок таблиці записується одиниця U[1]=1. Наступним п.2 процедури індекс j номеру рядка таблиці збільшується на одиницю і стає рівним двом. Згідно з п.3, код табличного значення U[j]=U[2] формується як зсунутий на $2^k = 2^3 = 8$ двійкових розрядів код попереднього рядка таблиці U[j-1]=U[1]: U[2]=U[1]<<8 = 1<<8 = 256. Через те, що цей код має розрядність меншу за n=12 U[2] rem P = U[2]. Аналогічним чином здійснюється заповнення наступного рядка U[3] таблиці передобчислень: U[3]=(U[2]<<8) rem P = U[2]<<8. При формуванні табличного значення U[4] довжина зсунутого на 8 розрядів коду U[3]=2516 перевищує n: U[3]<<8 = 2516·256 = 644096. Тому над виконується редукція: обчислюється залишок від поліноміального ділення 644096 на P=7079: U[4]=644096 rem 7079 = 3129. Формування всіх наступних табличних значень U[5],…,U[8] здійснюється аналогічним чином. Підсумкова таблиця передобчислень для заданих значень P=7079 та k=3 наведена в таблиці 1.

Таблиця 1. Значення передобчислень U редукцій ступенів двійки для P=7079 та k=3

j	U[j]	j	U[j]	j	U[j]
1	1	5	2338	9	1508
2	256	6	3048	10	3099
3	2516	7	3960	11	1995
4	3129	8	2956	12	2045

Виходячи з того, що таблиця U містить n n-розрядних чисел, її об'єм становить n^2 біт.

Передбачена запропонованим методом швидкого обчислення експоненти на полях Галуа друга таблиця T передобчислень ступенів A містить 2^k n-розрядних чисел. Кожен j-тий із 2^k рядків таблиці T[j], $j \in \{0,1,\ldots,2^k-1\}$ містить залишки від ділення поліноміального номера j на A : T[j]= A^j rem P.

Процедура формування таблиці T передобчислень ступенів A зводиться до виконання наступної послідовності дій:

1. Табличні значення T[0] та T[1] встановлюється рівним відповідно одиниці та A: T[0]=1 та T[1]=A. Значення індексу i встановлюється рівним двійці : i =2.
2. Значення i – го рядка таблиці T заповнюється залишком від поліноміального ділення на утворюючий поліном P поля Галуа поліноміального добутку попереднього табличного значення на число A: T[i] = (T[i-1] ⊗ A) rem P.
3. Значення індексу i збільшується на одиницю : i=i+1. Якщо $i < 2^k$, здійснюється перехід на повторне виконання п.2

Робота запропонованої процедури формування таблиці T передобчислень може бути ілюстрована наступним прикладом: нехай значення A=4021, P=7079, а k=3. При виконанні п.1 в перші два рядка таблиці заносяться коди одиниці та A=4021: T[0]=1 та T[1]=4021, а індекс i встановлюється рівним двом: i=2. В п.2 процедури наступне табличне значення T[2] обчислюється у вигляді: T[2]=(T[0]⊗A) rem P = (4021⊗4021) rem 7079 = 3888. В рамках виконання п.3 значення і збільшується на одиницю, тобто стає рівним трьом: i=3. В силу того, що це значення менше 2^3=8, здійснюється повторне виконання п.2. В подальшому заповнення таблиці відбувається аналогічним описаному чином. Результат – сформована таблиця T представлена в таблиці 2.

Таблиця 2. Результати передобчислень T для значень A=4021, k=3, P=7079

j	A^j rem P	j	A^j rem P
0	0	4	3412
1	4021	5	1329
2	3888	6	3022
3	1221	7	3678

Пропонована процедура прискореного обчислення експоненти на полях Галуа A^E rem P включає наступну послідовність дій:

1. За заданими значеннями A та P та обраним значенням k формується таблиця T передобчислень ступенів A, яка зберігається в пам'яті.

2. Стартове значення номеру j молодшого розряду поточного k–розрядного фрагменту коду експоненти E встановлюється рівним $j=n-k+1$. Стартове значення поточного коду R результату встановлюється рівним одиниці: $R=1$.

3. Індекс i номеру розряду поточного результату R встановлюється рівним одиниці: $i=1$. Початкове значення коду Q встановлюється рівним нулю: Q=0.

4. Якщо i-тий двійковий розряд r_i коду $R= r_1 + r_2\cdot 2 + r_2\cdot 2^2 +\ldots+ r_n\cdot 2^{k-1}$, $\forall l\in\{1,2,\ldots,n\}: r_l\in\{0,1\}$, поточного результату дорівнює одиниці: $r_i=1$, то до коду Q виконується логічне додавання (XOR) i-го значення таблиці U[i] передобчислень: $Q=Q\oplus U[i]$.

5. Індекс i номеру розряду результату R збільшується на одиницю: $i=i+1$. Якщо після цього значення індексу і менше $n+1$: $i < n+1$, то здійснюється повернення на повторне виконання п. 4.

6. Виконується множення на полі Галуа обчисленого коду Q на табличне значення T[f_j], яке адресується чисельним значенням $f_j = e_j + e_{j+1}\cdot 2+\ldots+e_{j+k-1}\cdot 2^{k-1}$ поточного j-го фрагменту коду експоненти E зі збереженням результату в R : $R = Q \otimes T [f_j]$ rem P.

7. Здійснюється перехід до обробки наступного k–розрядного фрагменту коду експоненти E шляхом зменшення номеру j молодшого розряду поточного фрагменту на величину k : $j = j - k$. Якщо після цього $j > 0$ реалізується повернення на повторне виконання п. 3.

8. Кінець. Результат обчислення експоненти на полі Галуа зафіксовано в змінній R: $R=A^E$ rem P.

Робота запропонованої процедури швидкого експоненціювання на полях Галуа може бути ілюстрована прикладом обчислення 4021^{3758} rem 7079 = 2185. Відповідно, утворюючий поліном $P(x) = x^{12} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^2 + x +1$, який співвідноситься з числом P=7079, формує поле Галуа GF(2^{12}) 12-розрядних чисел, тобто n=12. Число, над яким виконується експоненціювання A=4021, код експоненти E=3758, розряди якої оброблюються фрагментами по три розряди, тобто k=3.

Оскільки в рамках прикладу поле Галуа утворено поліномом $P(x) = x^{12} + x^{11} + x^9 + x^8 + x^7 + x^5 + x^2 + x +1$, то для експоненціювання використовується наведена в таблиці 1 значення передобчислень.

В ході виконання п.1 запропонованої процедури формуються наведені в таблиці 2 передобчислення T, які залежить від значення A=4021 та довжини фрагменту k=3.

Згідно п.2 значення номеру j молодшого розряду поточного фрагменту експоненти E встановлюється рівним $j=n-k+1$ = 12-3+1=10. Значення R встановлюється рівним одиниці: R=1.

У відповідності з п.3 процедури, індекс i номеру розряду поточного результату R встановлюється рівним одиниці: i=1, а код Q обнуляється: Q=0.

В силу того, що R=1, при i=1 в п.4 до коду Q додається рівне одиниці значення U[1], в результаті чого Q стає рівним одиниці. При всіх інших значеннях індексу і від 2-х до 12-ти, значення Q не змінюється оскільки відповідні розряди R дорівнюють нулю.

Наступним п.6 одиничне значення Q множиться на табличне значення T[f_j], яке адресується кодом 3-х старших розрядів експоненти E, який дорівнює f_j = 7 (111_2), в результаті чого в змінну R заноситься код $R = Q \otimes T [7]$ rem P = 1⊗ 3678 rem 7079 = 3678. Значення j зменшується на k=3: $j=j-k$ =10-3 = 7. Оскільки це значення більше нуля, здійснюється перехід на повторне виконання п.3.

В рамках реалізації п.3-4, індекс і змінюється від 1 до 12 і в змінній Q формується логічна сума табличних значень U, які співвідносяться з одничними компонентами двійкового коду R. В силу того, що поточне значення R=3678 = 1110 0101 1110_2, обчислений в циклі виконання в п.3-4 код Q являє собою логічну суму: Q = U[2] ⊕U[3] ⊕ U[4] ⊕ U[5] ⊕ U[7] ⊕ U[10] ⊕U[11]⊕U[12] = 256 ⊕ 2516 ⊕ 3129 ⊕ 2338 ⊕ 3960 ⊕ 3099 ⊕ 1995 ⊕ 2045 = 3738.

Наступним п.6 обчислене значення Q = 3738 множиться на табличне значення T[f_j], яке адресується кодом, утвореним 7-м, 8-м і 9-м розрядами експоненти E, який при e_9=0, e_8=1, e_{10}=0 дорівнює f_j = 2 (010_2). Відповідно виконується множення на полі Галуа: $R = Q \otimes T [2]$ rem P = 3738⊗3888 rem 7079 = 3420.

В подальшому, обробка другого та першого фрагментів коду експоненти E здійснюється по аналогії . В таблиці 3 наведена покрокова динаміка трансформацій змінних процедури.

Таблиця 3. Динаміка прокрокових значень змінних процедури

j	f_j	Q	R
10	111_2=7	1	1⊗3678 rem 7079 = 3678 =1110 0101 1110_2
7	010_2=2	3738	3738⊗3888 rem 7079= 3420 = 1101 0101 1100_2
4	101_2=5	1637	1637⊗1329 rem 7079= 253 = 1111 1101_2

1	110_2=6	978	978⊗3022 rem 7079= 2185

В запропонованому методі для обчислення $R=A^E$ rem P використовується, в середньому, $0.5\cdot n^2/k$ операцій логічного додавання (XOR) n-розрядних чисел та n/k операцій множення чисел такої довжини на полі Галуа.

Операція множення n-розрядних чисел на полі Галуа складається з власне поліноміального множення та редукції (отримання залишку від поліноміального ділення на утворюючий поліном P поля Галуа) отриманого добутку. Перша частина зводиться до n-кратного виконання двох операцій: логічного додавання (за умови одиничного значення поточного розряду множника) та зсуву. Враховуючи, що час виконання цих логічних операцій для переважної більшості процесорів близький за значенням, можна вважати, що поліноміальне множення потребує виконання $1.5\cdot n$ операцій логічного додавання. Поліноміальне ділення потребує для реалізації, в середньому, $n/2$ логічних додавань та n зсувів. В цілому, час множення на полях Галуа оцінюється витратами часу на виконання $3\cdot n$ операцій логічного додавання n-розрядних чисел.

З урахуванням наведеного, час T обчислення експоненти на полях Галуа в запропонованому методі може бути оцінений витратами часу на реалізацію $T= 0.5\cdot n^2/k + 1.5\cdot n^2/k = 2\cdot n^2/k$ логічних додавань.

При використанні одного із двох різновидів класичного алгоритму експоненціювання на полях Галуа, потрібно реалізувати $1.5\cdot n$ операцій множення на полях Галуа n-розрядних чисел, або $2.25\cdot n^2$ операцій логічного додавання n–розрядних чисел. Іншими словами, час T_C обчислення експоненти на полі Галу за класичним алгоритмом становить $T_C=2.25\cdot t_{XOR}$. Тобто, в порівнянні з класичним алгоритмом обчислення експоненти на полях Галуа, запропонований метод дозволяє прискорити її реалізацію в β_C раз, при тому, що чисельне значення β_C визначається формулою:

$$\beta_C = \frac{T_C}{T} = \frac{2.25\cdot n^2}{2\cdot \frac{n^2}{k}} = 1.125\cdot k\,. \qquad (3)$$

На практиці значення коефіцієнту β_C пропорційне значенню k, яке обмежується об'ємом пам'яті для зберігання результатів передобчислень. Оскільки таблиця U передобчислень редукцій ступенів двійки залежить тільки від утворюючого поліному $P(x)$ поля Галуа, який є частиною відкритого ключа і, відповідно, незмінний, для її зберігання може бути використана вбудована флеш-пам'ять мікроконтролера. Як зазначалося вище, об'єм таблиця U становить n^2 біт, що при значенні n=4096=2^{12} складає 2^{24} біти або 2^{21} байти (2 Мбайти). Тобто,об'єм вбудованої флеш-пам'яті більшості сучасних термінальних мікроконтролерів, таких, зокрема як ARM Cortex-M чи PIC, цілком дозволяє зберігати таблицю U передобчислень.

Таблиця T об'ємом $2^k\cdot n$ біт, формується безпосередньо перед кожним обчисленням експоненти A^E rem P і, відповідно, має зберігатися в оперативній пам'яті. Враховуючи, що об'єм оперативної пам'яті сучасних мікроконтролерів становить 256-512 Кбайт (2^{21}-2^{22} біт), при n=4096, граничне значення k визначається як $\log_2 2^{21}/2^{12} = 9$. В рамках проведених експериментальних досліджень використовувалося k=8, що забезпечило реальне прискорення обчислення експоненти по полі Галуа в 8.5 раз.

Якщо порівнювати запропоноване рішення, яке полягає в одночасній обробці k розрядів коду експоненти з відомим [14], в якому організується одночасна обробка k розрядів числа, при його піднесенні до квадрату, то відповідне значення коефіцієнту β_k прискорення визначається у наступному вигляді:

$$\beta_k = \frac{T_M}{T} = \frac{(3+k)\cdot n^2}{2\cdot n^2} = 1.5 + \frac{k}{2}\,. \qquad (4)$$

Це означає, що при однакових значеннях k=8, запропонований метод забезпечує прискорення обчислення експоненти на полях Галуа в п'ять раз.

Висновки. В результаті досліджень, направлених на прискорення комп'ютерної реалізації механізмів захисту інформації на базі криптографії з відкритим ключем, в основі яких лежить алгебра кінцевих полів Галуа GF(2^n), теоретично обґрунтовано, розроблено та досліджено метод прискореного обчислення експоненти.

Запропонований метод експоненціювання над полях Галуа, який відрізняється організацією групової обробки розрядів коду експоненти з використання передобчислень, що дозволяє зменшити

кількість мультиплікативних операцій піднесення до квадрату та множення і, тим самим, прискорити процес обчислення.

В основу методу покладено властивості операції експоненціювання числа на полях Галуа за умови, що код експоненти є ступенем числа два. Ці властивості дозволяють об'єднати групу операцій піднесення до квадрату на полі Галуа, якщо число елементів в групі є ступенем числа два. З використанням цього в рамках запропонованого методу організовано одночасну обробку груп із k розрядів коду експоненти.

Теоретично показано та експериментально доведено, що розроблений метод забезпечує прискорення обчислення експоненти на полях Галуа практично в k раз. При цьому значення k обмежується ресурсами пам'яті комп'ютерної платформи для зберігання таблиць передобчислень. Зокрема при використанні в якості такої платформи термінальних мікроконтролерів досягнуто прискорення обчислення в 8.5 раз.

Розробка орієнтована для швидкої реалізації криптографічних механізмів захисту від зовнішнього втручання в роботу систем віддаленого контролю та управління, в яких якості середовища обміну даними використовується Інтернет.

Список бібліографічного опису:

1. Kopetz H. Internet of Things /H.Kopetz, W. Steiner// Real-Time Systems. Springe, Cham.-2022.-P.325-341. DOI: 10.1007/978-3-031-11992-7_13.
2. Rayer A. Internet of Things Security and Privacy /A.Rayer, S.Salam // Internet of Things from Hype to Reality. Springer, Cham.-2022.- P.213-246. DOI:10.1007/978-3-030-90158-5_8.
3. Rezai A. A New Finite Field Multiplication Algorithm to Improve Elliptic Curve Cryptosystem Implementations / Abdalhossein Rezai, Parviz Keshavarzi // Journal of Information Systems and Telecommunication, - 2013.-Vol. 1, No. 2, P.119-129.
4. Марковський О.П. Використання алгебри полів Галуа для реалізації концепції «нульових знань» при ідентифікації та автентифікації віддалених / О.П.Марковський, Захаріудакіс Ліфтеріс, Максимук В.Р. // Електронне моделювання. Збірник наукових праць. - 2017.- Т.6, №39. - С.33-45.
5. Калмиков І.А. Розробка методу нелінійного шифрування інформації з використанням операції піднесення до степеня для кінцевого поля Галуа / І.А. Калмиков, Е.С. Степанова, К.Т. Тинчеров// Сучасні наукомісткі технології. — 2019.- № 9.- 2019. - С.84—89.
6. Daiko I. Fast exponential method on Galois fields for cryptographic applications / Ihor Daiko, Victor Selivanov // 13-th International Conference on Dependable system, Service and Technologies DESSERT-2023,13-15 October, Greece, Athens. 2023.- P.648-650. DOI :10.1109/DESSERT61349.2023.10416519.
7. Русанова О.В. Метод розподіленого модулярного експоненціювання на термінальних мікроконтролерах IoT із захищеним залученням хмарних обчислень / О.В. Русанова, М.А. Гайдукевич // Проблеми управління та інформатизації - 2024.- № 2 (78).- С.91-103. DOI: 10.18372/2073-4751.78.18966.
8. Fitzpatrick P. Algorithm and Architecture for a Galois Fiels multiplicative Arithmetic Processor/ P. Fitzpatrick, Popovici E. M. // IEEE Trans. on Information Theory. — 2003— V.49, - № 12, — P. 3303— 3307.
9. Жолубак І. Аналіз алгоритмів множення в полях Галуа для криптографічного захисту інформації / Bulletin of the Lviv Polytechnic National University "Information systems and networks"-2023.- Вип.13.- С,338-349. DOI: 10.23939/sisn2023.13.338.
10. Николайчук Я.М. Коди полів Галуа: теорія і застосування / Я.М. Николайчук //Тернопіль.-Вид-во ТНУ. — 2012. - 576 с.
11. Osadchyy V. The Order of Edwards and Montgomery Curve / V.Osadchyy // WSEAS Transactions on Mathematics, - 2020.- Vol. 19.- № 25, - P. 253-264.
12. Wu H. Finite field multiplier using redundant representation/ H. Wu, M.A. Hasan, I.F. Blake, S.Gao // IEEE Trans. Computers. —2002 — V.51,- № 51. — P. 1306 — 1316.
13. Марковський О.П. Метод швидкого експоненціювання на полях Галуа для систем криптографічного захисту інформації / О.П. Марковський, І.В. Дайко // Проблеми управління та інформатизації - 2024.- № 1 (77.- С.80-88. DOI: 10.18372/2073-4751.77.18660
14. Al-Mrayt Ghassan Abdel Jalil Halil. Organization of fast exponentiation on Galois Fields for cryptographic data protection systems / Al-Mrayt Ghassan Abdel Jalil Halil, O. Markovskyi, A. Stupak // Information, Computing and Intelligent systems. – 2022. – № 3.- P.17-25. DOI: 10.20535/2708-4930.3.2022.

References:

1. Kopetz H. Internet of Things /H.Kopetz, W. Steiner// Real-Time Systems. Springe, Cham.-2022.-P.325-341. DOI: 10.1007/978-3-031-11992-7_13.
2. Rayer A. Internet of Things Security and Privacy /A.Rayer, S.Salam // Internet of Things from Hype to Reality. Springer, Cham. 2022.- P.213-246. DOI:10.1007/978-3-030-90158-5_8.
3. Rezai A. A New Finite Field Multiplication Algorithm to Improve Elliptic Curve Cryptosystem Implementations / Abdalhossein Rezai, Parviz Keshavarzi // Journal of Information Systems and Telecommunication, - 2013.-Vol. 1, No. 2, P.119-129.

4. Markovskyi O.P. Galois Fields Algwbra Utilization for Implementation of the Conception of Zero-Knowledge Under Identification and Authentification of Remote Users / O.P. Markovskyi, Zacharioudakis Leftherios, V.R. Maksymuk //Electronic Modeling.-2017.- Vol.6.- №39. - P.33-45.
5. Kalmikov I.A. Rozrobka metody nelinijnogo shifruvannja informacii z vikoructannjam operacij pidnesennja do stepeni dlja kincevogo polja Galua /I.A.Kalmikov, E.C.Stepanova, K.T.Tincherov// Sychasni Naukomisni technologiji. — 2019.- № 9.- 2019. - C.84—89.
6. Daiko I. Fast exponential method on Galois fields for cryptographic applications / Ihor Daiko, Victor Selivanov // 13-th International Conference on Dependable system, Service and Technologies DESSERT-2023,13-15 October, Greece, Athens.-2023.- P.648-650. DOI :10.1109/DESSERT61349.2023.10416519.
7. Rusanova O.V. Method of distributed modilar exponentiation on IoT terminal micricontrollers with protected involvement of cloud computing / O.V.Rusanova, M.A. Haidukevych // Problems of Informatization and Managements.- 2024.- № 2 (78).- P.91-103. DOI: 10.18372/2073-4751.78.18966.
8. Fitzpatrick P. Algorithm and Architecture for a Galois Fiels multiplicative Arithmetic Processor./ P. Fitzpatrick, Popovici E. M. // IEEE Trans. on Information Theory. — 2003— V.49, - № 12, — P. 3303— 3307.
9. Golubak I. Analysis of multiplication algorithms on Galoise Fields for cryptographic data protection / Bulletin of the Lviv Polytechnic National University "Information systems and networks" -2023.- Ed.13.- P.338-349. DOI: 10.23939/sisn2023.13.338.
10. Nikolajchuk J.M. Kodi poliv Galua: teorija i zstosuvannja / J.M. Nikolajchuk //Ternopil.-Ed TNY. —2012. - 576 c.
11. Osadchyy V. The Order of Edwards and Montgomery Curve / V.Osadchyy // WSEAS Transactions on Mathematics, - 2020.- Vol. 19.- № 25, - P. 253-264.
12. Wu H. Finite field multiplier using redundant representation./ H. Wu, M.A. Hasan, I.F. Blake, S.Gao // IEEE Trans. Computers. —2002 — V.51,- № 51. — P. 1306 — 1316.
13. Markovskyi O.P. Method of fast exposure in Galoise Fields in cryptographic data protection systems // O.P. Markovskyi, I.V. Daiko // Problems of Informatization and Managements.- 2024.- № 1 (77).- P.80-88. DOI: 10.18372/2073-4751.77.18660
14. Al-Mrayt Ghassan Abdel Jalil Halil. Organization of fast exponentiation on Galois Fields for cryptographic data protection systems / Al-Mrayt Ghassan Abdel Jalil Halil, O. Markovskyi, A. Stupak // Information, Computing and Intelligent systems. – 2022. – № 3.- P.17-25. DOI: 10.20535/2708-4930.3.2022