Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

На правах рукопису

УДК 004.9:004.056:681.51

ЛІТАВА ГЖЕГОЖ ВЛАДИСЛАВ

МОДЕЛІ ТА ЗАСОБИ ПІДВИЩЕННЯ ЖИВУЧОСТІ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ НА ОСНОВІ ЕЛІПТИЧНИХ КРИВИХ

05.13.06 – інформаційні технології

Дисертація на здобуття наукового ступеня
кандидата технічних наук

Наукові керівники:
Карпінський Микола Петрович
доктор технічних наук, професор
Алєксандер Марек Богуслав
кандидат технічних наук (Польща)

Тернопіль – 2014

Зміст

ВСТУП

Актуальність теми. Розвиток економічного потенціалу кожної країни нерозривно пов'язаний з техногенною безпекою.

Для її забезпечення одним з ключових чинників являються інформаційні технології (ІТ), як інструмент для розробки та впровадження систем управління гарантоздатності критично важливої інфраструктури. На сьогоднішній день щораз більше уваги приділяється захисту інформації та конфіденційності. Нехтування додатковими дефіцитами гарантоздатності, такими як інформаційно-управляючі системи (ІУС), може призвести до що п’ятої відмови атомного енергетичного обладнання і що п'ятої аварії космічно-ракетної техніки, з тенденцією їх зростання за останні роки. Цій темі притаманний міжнародний характер та її розгляду присвячено низку міжнародних конференцій, пов’язаних з гарантоздатністю, зокрема з її складовими – живучістю та функціональною безпекою ІТ: DESSERT, DSN, EDCC, ESREL, SAM, SAFECOMP, тощо. Важливим питанням є дослідження спеціальних режимів функціонування ІУС з точки зору їх живучості, які, у поєднанні з розвитком комплексних інтегрованих дистанційних сенсорів, належать до одного з провідних наукових напрямів найбільших інженерних викликів XXI століття, визначених науковим фондом NSF, а також передбачених до виконання згідно з Постановою Президії НАН України. Доцільно виділити окремий клас ІУС, що базуються на еліптичних кривих (ІУСЕК), тобто ІУС, в пристроях яких застосовуються обчислювальні операції на еліптичних кривих (ЕК), включаючи Elliptic Curve Cryptography Device (ECCD або криптографічні пристрої на ЕК), що веде до необхідності удосконалення, опрацювання і впровадження відповідних моделей живучості ІУСЕК.

Істотним завданням постає оцінювання та підвищення живучості ІУСЕК, а також самих алгоритмів шифрування. Від криптографічних пристроїв вимагається також можливість шифрування та розшифрування в реальному часі. Крім цього, не слід скидати з рахунків одну з основних проблем безпеки, яка зводиться до розв’язання задачі дискретного логарифмування. Кожне збільшення швидкості

виконання основних обчислень на ЕК зумовлює скорочення часу, необхідного для знаходження цього логарифму. Одним з основних факторів, який визначає ефективність обчислювальних методів, моделей та технологій не тільки щодо швидкості шифрування/розшифрування, а також непрямо рівень гарантоздатності ECCD і відповідно параметри живучості ІУСЕК, є швидкодія додавання точок на ЕК, від якої безпосередньо залежить час, необхідний для розв’язання дискретного логарифма. З огляду на це, удосконалення моделей та засобів ІУС, що базуються на пристроях для виконання криптографічних операцій на ЕК, для підвищення живучості цих систем є актуальною науковою задачею, що має важливе наукове та практичне значення.

Зв’язок роботи з програмами, планами, науковими темами. Дисертаційна робота виконувалася у Тернопільському національному технічному університеті імені Івана Пулюя (ТНТУ ім. І. Пулюя) за держбюджетним договором на виконання науково-дослідних робіт за темою "Розробка, дослідження та впровадження методів і засобів контролю та управління якістю програмних продуктів" (ДР № 0113U000258) (2012-2013 рр., виконавець).

Мета і задачі дослідження. Метою дисертації є удосконалення моделей та засобів для підвищення живучості інформаційно-управляючих систем на основі еліптичних кривих.

Досягнення цієї мети вимагає розв’язання наступних завдань:

1. Провести аналіз моделей живучості, засад побудови та технологічних рішень ІУСЕК з точки зору їх адекватності та можливого використання для розв’язання поставленої задачі.

2. Здійснити вибір обчислювальних платформ і розробити, ґрунтуючись на ефективних методах розв'язання дискретного логарифма, моделі засобів та технології обчислень на ЕК *GF(p)* у пристроях ІУСЕК з врахуванням дефектів зовнішніх впливів.

3. Створити апаратно-програмні засоби для виконання обчислень на ЕК *GF(p)* і розв’язання дискретного логарифма вищої швидкодії для живучих ІУСЕК.

4. Дослідити вплив методів виконання основних операцій на ЕК над скінченним полем вищих порядків *GF(p)* на параметри живучості ІУСЕК для кожної з опрацьованих моделей та обчислювальних технологій.

5. Провести симуляційні дослідження (імітаційне моделювання) для верифікації отриманих теоретичних залежностей щодо живучості ІУСЕК та реалізувати імплементацію запропонованих рішень в практиці.

Об'єкт дослідження – процеси моделювання оцінки живучості інформаційно-управляючих систем на основі еліптичних кривих.

Предмет дослідження – моделі та засоби забезпечення живучості інформаційно-управляючих систем на основі еліптичних кривих.

Методи досліджень. Теорія ЕК, теорія математичного моделювання, теорія алгебри та теорія криптографії для створення технологій обчислень й побудови моделей пристроїв ІУСЕК на основі ЕК над полем вищих порядків, оцінювання рівня живучості таких систем, розроблення апаратно-програмних засобів для ефективних обчислень на ЕК і розв’язання задачі дискретного логарифмування для визначення рівня живучості ІТ, що забезпечується пристроями захисту на основі ЕК.

Наукова новизна дисертації.

1. Удосконалено структурну модель підвищення живучості ІУСЕК з врахуванням дефектів зовнішніх впливів за ознаками ймовірності та детермінованості, що дало змогу створити підґрунтя для проектування ефективних засобів цих систем.

2. Удосконалено технології обчислень на ЕК шляхом заміни множення за модулем в алгоритмі Крестенсона, що дало змогу звести множення до операції додавання, яка відрізняється від відомих новою архітектурою і меншою складністю у порівнянні з підходом, який ґрунтується на традиційному множенні, завдяки чому збільшено швидкість обчислень і прискорено дію алгоритму розв’язання задачі дискретного логарифмування у пристроях ІУСЕК.

3. Вперше одержано модифіковані моделі засобів обчислень на ЕК у пристроях ІУСЕК з врахуванням дефектів зовнішніх впливів на них, що дозволило

по-новому визначити розмір кривих, які застосовуються в інтегрованих інформаційних системах, і забезпечити підвищену ефективність засобів протидії потенційним впливам.

4. Вперше, ґрунтуючись на алгоритмі модульного множення в теоретико-числових базисах (ТЧБ) Радемахера-Крестенсона щодо розв'язання дискретного логарифму, розроблено моделі, технології та структури засобів обчислень на ЕК в ІУСЕК для виконання криптографічних операцій, завдяки чому здійснено верифікацію параметрів живучості ІУСЕК та доведення збільшення його рівня.

Практичне значення отриманих результатів. На підставі отриманих теоретичних результатів розроблено математичні моделі, технології обчислень і засоби на ЕК вищих порядків, які можуть використовуватися для підвищення живучості ІУСЕК, а також вибору відповідних ЕК, які забезпечують функціонування ІУСЕК в аномальних умовах, викликаних дефектами зовнішніх впливів на неї.

На основі запропонованих моделей та технологій обчислень створено, з імплементацією в середовищі програмованих користувачем вентильних матриць FPGA (Field Programmable Gate Array), апаратні засоби для реалізації стежок блукання ро-алгоритму Полларда, які використано складовою частиною комплексної апаратно-програмної системи розв'язання дискретного логарифма на ЕК, що дало змогу отримати відомості про параметри живучості засобів ІУСЕК.

Практична цінність роботи полягає в тому, що на отриманій експериментальній базі модельованої атаки на ІУСЕК визначено час захисту засобів в залежності від розміру застосованих кривих, завдяки чому оцінено максимальне зростання ризику в короткостроковій перспективі, ґрунтуючись на особливостях симульованих дефектів впливу і даних щодо підвищення продуктивності нових апаратних рішень. Це також забезпечило вибір оптимальних розмірів ЕК в спеціалізованих рішеннях, де важливим є компроміс між обчислювальною потужністю та рівнем захисту з врахуванням часу, протягом якого даний засіб повинен бути живучим.

Додатковим фактором практичної цінності роботи є здійснення додавання точок та реалізації ро-алгоритму Полларда за допомогою створеної системи, побудованої на основі програмованих матриць FPGA типу Stratix III, завдяки чому забезпечено високу швидкодію і втричі зменшено час, необхідний для розв'язання дискретного логарифма на тій самій кривій, в порівнянні з програмною системою, яка працює на процесорі типу Itanium2. Побудовано паралельну систему, в якій швидкість зростає пропорційно до кількості застосованих програмованих матриць FPGA, здійснено також перенесення і симуляцію функціонування заімплементованої моделі на кластері FPGA (120 компонентів), який базується на програмованій матриці серії Virtex-4.

Теоретичні та практичні результати дисертаційної роботи використані та впроваджені: при виконанні науково-дослідної роботи "Розробка, дослідження та впровадження методів і засобів контролю та управління якістю програмних продуктів" (ДР № 0113U000258), що виконувалася в ТНТУ ім. І. Пулюя, ТОВ "Шредер" для повноцінного функціонування ІУС із збереженням їх живучості за наявності потенційних загроз, захисту ІУС від несанкціонованого доступу і підвищення відмовостійкості та продуктивності ІУС, а також у навчальному процесі ТНТУ ім. І. Пулюя в курсах „Технології захисту інформації", "Інформаційна безпека" та Державної вищої професійної школи у Новому Сончі, Польща (ДВПШ) при викладанні дисциплін "Безпека інформаційних технологій", "Криптографія і теорія кодів" та "Мережеві технології", згідно з Договором про співпрацю між ДВПШ і ТНТУ ім. І. Пулюя.

Особистий внесок здобувача. Усі результати, які формують основний зміст кандидатської дисертації, автор отримав особисто. У друкованих працях, опублікованих в співавторстві, здобувачеві належать: [3] – проведено тестування системи безпечної передачі даних в телекомунікаційній мережі та виконано аналіз отриманих результатів, [4] – сформульовано та обґрунтовано підхід з використанням паралельного додавання і ТЧБ Крестенсона до виконання основних операцій на ЕК *GF(p)* та застосування їх у реалізації паралельного ро-алгоритму Полларда за допомогою програмованих матриць FPGA, [5] – побудовано і

впроваджено структуру, виконано функціональну симуляцію і проведено верифікацію живучості ІУСЕК за допомогою системи, яка реалізує розв'язання дискретного логарифма на ЕК вищих рядів *GF(p)*, ґрунтуючись на паралельному ро-алгоритмі Полларда із застосуванням модульного множення на підставі ТЧБ Радемахера-Крестенсона, реалізованого на програмованих матрицях типу Stratix III, [6] – розроблено моделі та технології обчислень на ЕК у пристроях ІУСЕК для виконання криптографічних операцій, [7] – удосконалено модель обчислень на ЕК вищих порядків шляхом введення в основні операції на ЕК вищих порядків модульного множення, що ґрунтується на ТЧБ Радемахера-Крестенсона, та паралельного додавання, [26] – доведено доцільність застосування моделі, в якій використано гаусівські нормальні базиси, для виконання обчислень на ЕК в апаратних рішеннях, базованих на програмованих матрицях FPGA, для дослідження живучості ІУС, що ґрунтуються на ЕК другого порядку $GF(2^m)$, [51] – проаналізовано безпеку передавання інформації в комп'ютерних мережах, під час якого застосовано методи шифрування на базі ЕК, [53] – розроблено паралельні розподілені системи на основі бібліотек MPI2 і MIRACL для оцінювання часу розв'язування задачі дискретного логарифмування на ЕК в багатопроцесорних середовищах, [65] – одержано модифіковані моделі живучості ІУСЕК з врахуванням атак на них, [98] – виконано симуляційні дослідження для верифікації6+ отриманих теоретичних залежностей щодо живучості ІУСЕК.

Апробація результатів дисертації. Основні положення і результати дисертаційної роботи доповідалися та обговорювалися на: Дванадцятій науковій конференції ТНТУ ім. І. Пулюя (14-15 травня 2008, Тернопіль), 9th International Workshop "Computational Problems of Electrical Engineering" (CPEE'08) (September 16-20, 2008, Alushta (Crimea), Ukraine), III Міжнародній Науковій Конференції з циклу Інформатика в XXI столітті. Інформаційні технології в науці, техніці та інформатиці (Радом, Польща, 2009 р.), Xth International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics" (CADSM 2009) (Львів-Поляна, 2009), International Science-Practical Conference «Information Technologies and Security in Administration» (ITSM'2008-ITSM'2012) (Crimea, 2008-

2012), Науковому семінарі НАН України «Технічні засоби захисту інформації» (Одеське відділення) (2013), III Міжнародній науково-технічній конференції «Захист інформації і безпека інформаційних систем» (05-06 червня 2014 р., Львів, Україна).

Наукові результати дисертаційної роботи розглядалися та обговорювалися в Державному технологічному університеті (м. Черкаси), Університеті в Бєльську-Бялій (Польща), ДВПШ у Новому Сончі. В цілому роботу апробовано у ТНТУ ім. І. Пулюя (м. Тернопіль), Східноукраїнському національному університеті ім. В. Даля (м. Луганськ), Державному економічному університеті (м. Одеса).

Публікації. Основні результати дисертаційних досліджень опубліковані в 14 наукових роботах, 8 із них – наукові статті, серед яких 6 ([4-7, 26, 65]) – статті у наукових фахових виданнях України, 2 наукові статті ([66, 98]) – у провідних закордонних журналах, які входять до міжнародних наукометричних баз (IEEE, Inspec, Scopus, UlrichsWeb, Index Copernicus, Google Scholar, Baztech, ISI Master Journal List тощо), 2 розділи у закордонній монографії ([3, 13]), а також 4 публікації в матеріалах конференцій.

Структура та обсяг дисертації. Дисертаційна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел із 119 найменувань і додатків. Загальний обсяг дисертації становить 142 сторінки, з яких основний зміст викладений на 133 сторінках, містить 39 рисунків, 28 таблиць.

РОЗДІЛ 1
АНАЛІЗ МОДЕЛЕЙ ЖИВУЧОСТІ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ

У першому розділі на основі аналітичного огляду літературних джерел розкрито стан досліджуваної проблеми. Проведено аналіз математичних моделей оцінювання живучості інформаційно-управляючих систем з точки зору ймовірності та детермінованості, побудованих на основі засобів для виконання обчислювальних операцій на еліптичних кривих, який дозволив сформулювати відповідні технічні і наукові дані, що лежать в основі дисертаційних завдань для розв'язання. Зроблено огляд основних типів еліптичних кривих криптографічних систем, що використовуються в засобах ІУС. Показано важливість дискретного логарифма для живучості ІУС, в яких складовими елементами виступають криптографічні засоби. Піддано аналізу ефективні методи розв'язання дискретного логарифма. Обґрунтовано необхідність досліджувати нові, ефективніші моделі і технології обчислень та їх вплив на живучість ІУС.

1.1. Аналіз моделей оцінювання живучості інформаційно-управляючих систем на підставі еліптичних кривих

Під інформаційно-управляючими системами на основі еліптичних кривих розуміється клас ІУС, у засобах яких застосовуються обчислювальні операції на ЕК, включно із криптографічними пристроями на еліптичних кривих ECCD.

1.1.1 Аналіз загальних підходів до питання живучості інформаційно-управляючих систем.

Оптимальну функціональну безпеку отримуємо з урахуванням логічного і фізичного аспектів безпеки ІУС. Найчастіше вживане визначення функціональної безпеки вказує на те, що вона є складовою безпеки, яка залежить від правильного

функціонування системи у відповідь на вхідні дані [92]. Розглядаючи функціональну безпеку ІУС варто зазначити ще одну концепцію – цілісність безпеки, яка показує, що система протягом певного періоду часу і за певних умов правильно виконає необхідні функції безпеки [4, 8, 9, 15, 24, 25, 27, 89].

Даючи означення живучості, можна відзначити, що вона є складовою гарантоздатності, а саме властивістю для систем, зокрема ІУС, що застосовуються у сфері техногенної та природної безпеки, мінімізувати зниження працездатності і зберігати в прийнятних межах обсяг та якість надаваних послуг за відмов, обумовленими зовнішніми впливами різної природи, та дуже часто характеризує здатність системи виконувати своє призначення в часі, під час атак або збоїв [3, 10, 104, 107, 114, 116]. Збої або відмови можуть бути викликані несприятливими факторами в засобах, які ззовні впливають на залежний елемент. Небажані ефекти можуть бути зумовлені помилками в розробці програмного забезпечення, деградацією обладнання, людськими хибними діями або пошкодженням даних. Пошкодження даних може бути пов'язане з помилками як в середовищі зберігання і передавання даних, так і може також бути цілеспрямованим з метою спричинити дезінтеграцію системи. Живучість ІУС можна охарактеризувати як здатність виконувати завдання, що було поставлене перед системою, правильним способом, наприклад, в умовах атаки протягом певного періоду часу. Аналізуючи загрози, які є важливою частиною таксономії гарантоздатності і можуть зумовити ненадання послуг – невиконання функцій, слід враховувати, передусім, дефекти або несправності, що викликані різними причинами. Серед класифікованих згідно з [3] за ознаками множини дефектів, поділених на три групи, – розроблення або проектування (ДР), фізичних (ДФ) і зовнішніх впливів (ДВ), надалі зосереджено на третій групі ДВ, яка притаманна розглянутим в роботі ІУС та є наслідком зовнішніх впливів (несанкціонованого втручання або інформаційних атак включно з хакерськими і кракерськими атаками та спамом, помилковими діями персоналу, екстремальних впливів фізичного характеру), які можуть призводити до кратних відмов апаратних і програмних засобів ІУС.

Стандартизацію та класифікацію показників живучості протягом останнього часу здійснювалося з точки зору ймовірності та детермінованості [2, 22, 26]. Згідно з класифікацією за систематизованими двома ознаками показники можна поділити таким чином [26]:

1) за першою ознакою виділено 2 групи – показники для оцінювання живучості за станом системи (тобто зберігання працездатності після ДВ) та за результатами виконання завдання (тобто здатність не тільки протистояти ДВ, але й надалі, незважаючи на ДВ, успішно виконати встановлене завдання);

2) адитивні та мінімаксні, які відрізняються один від одного за способом зведення векторного показника до скалярного, причому до адитивних відносяться і ймовірнісні показники, що ґрунтуються на формулі повної ймовірності.

Ґрунтуючись на викладеному в [2, 22, 26], оцінити живучість за станом системи можна за допомогою таких показників:

- умовний закон уразливості (інакше ймовірність втрати працездатності ІУС за умови *k*-кратного ДВ)

$$Q(k) = p(f = 0 / A_k), \tag{1.1}$$

- виживаність ІУС для *k*-кратного ДВ

$$R_s(k) = 1 - Q(k) = p(f = 1 / A_k), \tag{1.2}$$

- запас живучості (*d*-живучість, інакше критична кількість ДВ, зменшена на одиницю)

$$d = C - 1, \tag{1.3}$$

- запас живучості (m_d-живучість, інакше максимальна кількість ДВ, яку ще може витримати ІУС без втрати працездатності)

$$m_d = max\ (i)\ m_{di}, \tag{1.4}$$

- середня кількість ДВ, що призводять до втрати працездатності (або математичне сподівання кількості ДВ, що задається розподілом Q*(k)*)

$$\overline{\omega} = \sum_{k=0}^{\infty} R_s(k), \tag{1.5}$$

- середній запас живучості

$$\overline{d} = \overline{\varpi} - 1, \tag{1.6}$$

причому у вищенаведених виразах A_k – подія, яка полягає у k-кратній появі ДВ; f – функція працездатності ІУС, яка приймає значення 1, якщо ІУС працездатна, та 0, якщо непрацездатна; С – критична кількість ДВ, тобто мінімальна кількість дефектів, поява яких призводить до втрати працездатності.

Слід зазначити, що перший, другий, п’ятий і шостий показники є ймовірнісними, тоді як третій і четвертий – детермінованими.

Оцінювання живучості за результатами виконання завдання протягом часу t базової S_0 та нової S_i структур ІУС можна здійснити за допомогою наступних показників [2, 22, 26]:

- умовна функція живучості (інакше відношення ймовірностей виконання завдання ІУС, визначених для двох випадків – для структур S_0 та S_i)

$$G(t/S_i) = G_i(t) = p(t/S_i) / p(t/S_0), \tag{1.7}$$

- функція виживаності ІУС для k-кратного впливу (подія A_k) (інакше усереднена по всіх можливих структурах функція живучості, враховуючи $p_k(r)$ – ймовірність виникнення структури S_r після k-кратного ДВ)

$$G(t/A_k) = G(t, k) = \sum_{r=1}^{K} p_k(r) G_r(t), \tag{1.8}$$

- безумовна функція живучості (тобто усереднена по всіх можливих подіях A_k функція виживаності ІУС)

$$G(t) = \sum_{k=1}^{\infty} p(A_k) G(t / A_k) = \sum_{r=1}^{K} p(S_r) G_r(t), \qquad (1.9)$$

причому ймовірність $p(S_r)$ в останній формулі визначається згідно з виразом

$$p(S_r) = \sum_{k=1}^{\infty} p(A_k)\, p_k(r). \qquad (1.10)$$

Передостанні два показники відносяться до класу адитивних і забезпечують згортку векторного показника $\{G_r(t), r = 1, ..., K\}$ в скалярний. За відсутності впевненої інформації про ймовірності $p_k(r)$ та $p(S_r)$ їх можна замінити на відповідні вагові коефіцієнти, призначені експертно. Якщо ж і це зробити важко, то необхідно переходити до мінімаксних показників.

На основі зазначеної в [1, 20, 97, 98] ризик-орієнтованої моделі оцінювання безпеки ІУС можна записати

$$R(t) = p(t)\, D, \qquad (1.11)$$

де безрозмірна величина $R(t)$ – ризик за час t, пов’язаний з деякою подією та значення якого часто використовують показником безпеки; $p(t)$ – імовірність цієї події, інакше імовірність переходу ІУС до небезпечного стану на часовому інтервалі t; D – коефіцієнт, який визначається об’єктом управління та відображає сукупність всіх можливих негативних наслідків, зумовлених вище згаданою подією.

1.1.2 Огляд результатів досліджень щодо розв'язків, базованих на еліптичних кривих.

1.1.2.1 Еліптичні криві та їх арифметика.

Інтеграли з виразів у вигляді [41, 59]

$$\int R\left(x, \sqrt{ax^3 + bx^2 + cx + d}\right)dx, \ \int R\left(x, \sqrt{ax^4 + bx^3 + cx^2 + dx + e}\right)dx \qquad (1.12)$$

були названі еліптичними інтегралами. Ці інтеграли отримали назву еліптичних інтегралів, оскільки вперше з ними зіткнулися під час розрахунку довжини еліпса. Конкретизуючи поняття еліптичних інтегралів, доцільно додати, що це є тільки ті інтеграли, які не можна обчислити у скінченному виді [6, 11, 23, 42, 58, 59]. За приклад еліптичної кривої найчастіше наводиться крива, описана рівнянням [47]

$$y^2 = 4x^3 - cx - d \qquad (1.13)$$

з нескінченно віддаленою точкою O.

Нехай E – це еліптична крива, задана проективними координатами над полем K та $P \in E(K)$, причому K^* є мультиплікативною групою [41, 47, 108]. Розглянемо узагальнене рівняння Веєрштраса у вигляді [41]

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3, \qquad (1.14)$$

де $a_1, \ldots, a_6 \in K$.

Тоді еліптична крива над полем K – це множина розв'язків узагальненого рівняння Веєрштраса (1.14) на площині P_K^2 і точка $O = (0, 1, 0)$ [41]. Будь-яка друга координата задовольняє рівняння 1.4, а 1 як частковий випадок.

Еліптична крива E також може бути задана в афінних координатах, для такого подання кривої проективним координатам (X, Y, Z) відповідають координати $(X/Z, Y/Z)$, якщо $Z \neq 0$. Беручи до уваги, що для еліптичної кривої точкою,

координата якої Z дорівнює нулю, є точка O (нескінченно віддалена), тому у випадку такого представлення точкою O розглядається точка, яка розташована в нескінченності. Узагальнене рівняння Веєрштраса для афінного подання еліптичної кривої прийме вигляд [41]:

$$Y^2 + a_1XY + a_1Y = X^3 + a_2X^2 + a_4X + a_6, \quad (1.15)$$

де $a_1, \ldots, a_6 \in K$.

Відомо, що еліптичні криві з точкою в нескінченності утворюють абелеву або комутативну групу з точки зору на додавання. Нейтральним елементом цієї групи є точка O.

Показана на рисунку 1.1 точка *P(x, y)* має координати *x* і *y* та $P \neq O$. Протилежна до неї точка *-P* має координати *(x, - y)*, що належать кривій *E*.

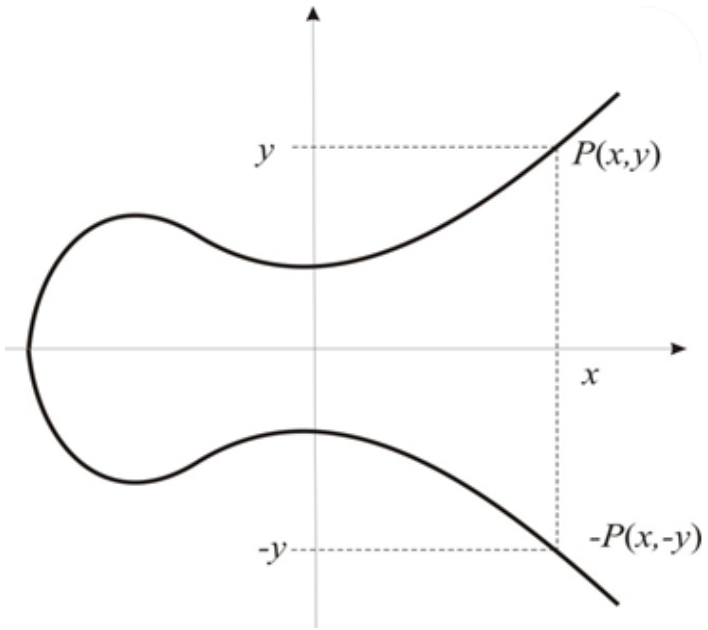

Рисунок 1.1 – Еліптична крива

Нехай точки P і Q належать кривій E та мають також відповідні координати $P(x_1, y_1)$ і $Q(x_2, y_2)$. Приймається, що точка $P \neq Q$ та $P, Q \neq O$. Сума точок $P + Q$ утворює точку $R(x_3, y_3)$.

На рисунку 1.2 показано графічну інтерпретацію додавання точок на еліптичній кривій. Пряма, яка проходить через точки P і Q, перетинає криву в точці $X = -R$, отже є протилежною точкою до точки R.

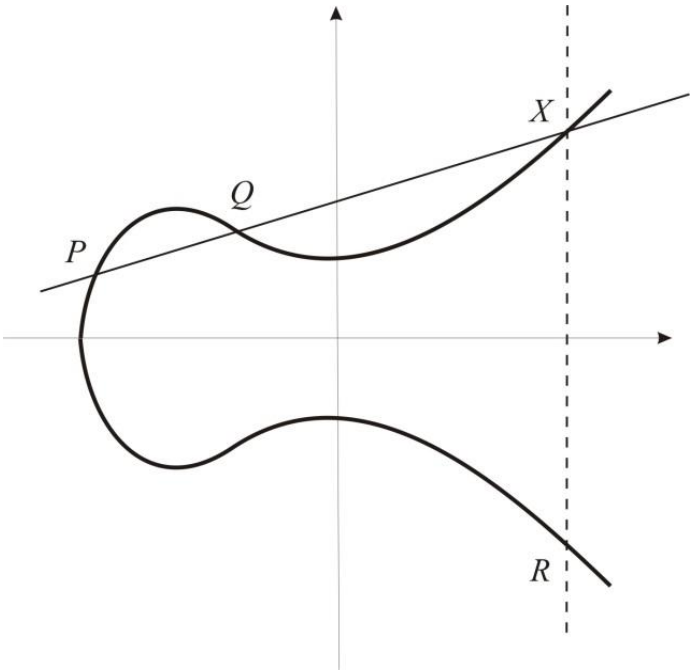

Рисунок 1.2 – Додавання точок

Для подальшого розгляду проаналізовано рівняння кривої виду:

$$y^2 = x^3 + ax + b. \tag{1.16}$$

Прийнято також, що коефіцієнти a та b, а також координати точок належать полю K. За визначенням, необхідно, щоб така крива не мала особливих точок. Геометрично це значить, що графік не повинен мати точок повернення і перетинів. Алгебрично це означає, що дискримінант $4a^3+27b^2$ не повинен дорівнювати нулю.

Опишемо пряму за допомогою рівняння

$$y = Ax + B, \tag{1.17}$$

де $A, B \in K$.

Оскільки точки P і Q лежать на одній прямій, тому $A = \frac{y_2 - y_1}{x_2 - x_1}$ та $B = y_1 - Ax_1$ і повинні задовольнятися співвідношення:

$$x_3 = A^2 - x_1 - x_2, \tag{1.18}$$

$$y_3 = -Ax_3 - B, \tag{1.19}$$

де x_3, y_3 – координати суми точок $P + Q$.

В окремих випадках постає необхідність додавання точки до самої себе, а саме: $X = P + P$, графічну інтерпретацію даної операції наведено на рисунку 1.3. Через точку P проведено дотичну до графіка E. Подальша частина дій є аналогічною, як і для випадку з додаванням двох різних точок.

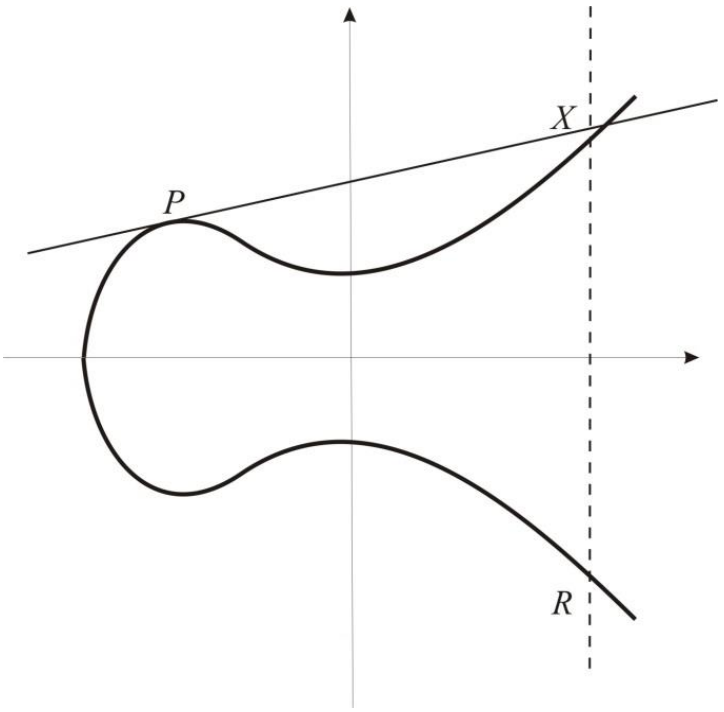

Рисунок 1.3 – Подвоєння точки

Враховуючи пряму, представлену рівнянням (1.17) та яка є дотичною до кривої (1.6) в точці P, можна знайти коефіцієнти цієї прямої, користуючись залежністю:

$$A = \frac{3x_1^2+a}{2y_1} \quad , \tag{1.20}$$

де a – коефіцієнт ЕК в (1.16).

Знаючи коефіцієнти A і B, після подальших виконань операцій як і для випадку додавання, отримуємо рівняння для координат точки $R = 2P = P + P$. Використовуючи наведені вище рівняння, можна здійснити множення на ціле число. Якщо τ є цілим числом, тоді отримується рівняння $\tau P = P + P + \cdots P$, де кількість точок, які підлягають додаванню, дорівнює числу τ .

Віднімання виконується шляхом додавання протилежної точки $Q - P = Q + (-P)$, а множення на від'ємне ціле число $-\tau$, як показано в рівнянні $-\tau P = -(\tau P)$.

Нижче описано еліптичні криві над скінченним полем характеристики два $GF(2^m)$ та над полем характеристики p, $GF(p)$, де p – просте число великої розрядності. Еліптичні криві над скінченним полем характеризуються скінченною кількістю раціональних точок на кривій.

1.1.2.2 Основні операції на еліптичних кривих.

Сучасна криптографія базується на еліптичних кривих над скінченним розширеним полем Галуа характеристики два $GF(2^m)$ та p, $GF(p)$ [11]. З цієї причини подальший розгляд буде стосуватися таких випадків. Еліптичною кривою над скінченним полем $GF(p)$ називається ЕК з редукцією по модулю p кривої E [53, 54]. Нехай E крива виражається рівнянням (1.15), тоді еліптична крива над скінченним полем $GF(p)$ описуватиметься рівнянням

$$y^2 \bmod p = g(x) \bmod p, \tag{1.21}$$

причому за рахунок зменшення за модулем p коефіцієнтів кривої E отримано їх зображення в полі $GF(p)$ [40].

Еліптичні криві над скінченним полем характеристики два $GF(2^m)$ визначаються множиною розв'язків рівняння $y^2 + xy = x^3 + ax^2 + b$, де $x, y, a, b \in GF(2^m)$ і $b \neq 0$ з врахуванням точки нескінченності [67].

1.1.2.3 Еліптичні криві над полем характеристики p.

З точки зору здійснюваних обчислень, важливо оцінити обчислювальну складність. Обчислення приймають різні форми в залежності від характеристики поля, над яким визначено еліптичну криву E [11, 41, 47]. Основний акцент в цій роботі поставлено на розгляд кривих над полем характеристики p>3. Для цього аналіз обчислювальної складності розпочато з цих кривих.

Для випадку $P_1 = P_2$ або подвоєння точки отримується:

$$x_3 = A^2 - 2x_1, \tag{1.22}$$

$$y_3 = (x_1 - x_3)A - y_1, \tag{1.23}$$

де $A = \frac{3x_1^2 + a}{2y_1}$.

Розглядаючи витрати на розрахунок суми точок в афінних координатах, отримуємо, що виконується три множення та одне обчислення оберненого елемента. Для випадку подвоєння необхідно здійснити відповідно чотири множення та обчислення оберненого елемента в полі, при цьому звичайно ігноруються додавання та множення на невеликі константи з точки зору незначних затрат. Відомо, що витрати на обчислення оберненого елемента в полі у багато разів вищі, ніж на множення, тому в здійсненні комп'ютерних розрахунків корисним вбачається використання таких способів подання кривих, які дозволяють уникнути необхідності обчислення оберненого елемента в полі. Одним з них є представлення кривої в проективних координатах, тоді отримання суми зводиться до здійснення послідовності операцій, показаних у таблиці 1.1 [41].

Таблиця 1.1

Модель обчислень при додаванні точок на кривій *GF(p)* в проективних координатах

$\lambda_1 = X_1 Z_2^2$	$\lambda_7 = \lambda_1 + \lambda_2$
$\lambda_2 = X_2 Z_1^2$	$\lambda_8 = \lambda_4 + \lambda_5$
$\lambda_3 = \lambda_1 - \lambda_2$	$Z_3 = Z_1 Z_2 \lambda_3$
$\lambda_4 = Y_1 Z_2^3$	$X_3 = \lambda_6^2 - \lambda_7 + \lambda_3^2$
$\lambda_5 = Y_2 Z_1^3$	$\lambda_9 = \lambda_7 \lambda_3^2 - 2X_3$
$\lambda_6 = \lambda_4 - \lambda_5$	$Y_3 = \left(\lambda_9 \lambda_6 + \lambda_8 \lambda_3^3\right)/2$

Можна зауважити, що витрати на виконання операцій зводяться до 16 множень. Подвоєння точок у проективних координатах зводиться до здійснення операцій, наведених в таблиці 1.2 [41].

Таблиця 1.2

Модель подвоєння точок на кривій *GF*(*p*) в проективних координатах

$\lambda_1 = 3X_1^2 + aZ_1^4$	$X_3 = \lambda_1^2 - 2\lambda_2$
$Z_3 = 2Y_1 Z_1$	$\lambda_3 = 8Y_1^4$
$\lambda_2 = 4X_1 Z_1^2$	$Y_3 = \lambda_1(\lambda_2 - X_3) - \lambda_3$

Нижче проаналізовано випадок, коли точка подана в афінних координатах, тоді як друга точка записана в проективних координатах. Перетворення точки між представленнями є тривіальним завданням. Заміна афінних координат на проективні полягає на добавленні третьої координати, що дорівнює 1. З іншого боку, заміна проективних координат на афінні зводиться до приписування афінним координатам $X/Z^2, Y/Z^3$, причому в проективних координатах точка має координати X, Y, Z. Затрати такого переходу – це одне знаходження оберненого елемента в полі, три дії множення і одне піднесення до квадрату.

Прийнято, що координата однієї з точок додавання, наприклад, точки P1 записана в афінних координатах, а іншої P2 – в проективних координатах.

Припускаючи тоді, що третьою координатою точки P2 є Z2=1, а також користуючись формулами, наведеними в таблиці 1.1, отримано послідовність дій, необхідних для додавання точок змішаним способом [41, 47]. Модель обчислень представлена у таблиці 1.3.

Таблиця 1.3

Модель обчислень для додавання точок на кривій $GF(p)$ в змішаному записі.

$\lambda_1 = X_1 Z_2^2$	$\lambda_8 = \lambda_4 + Y_2$
$\lambda_3 = \lambda_1 - X_2$	$Z_3 = Z_2 \lambda_3$
$\lambda_4 = Y_1 Z_2^3$	$X_3 = \lambda_6^2 - \lambda_7 \lambda_3^2$
$\lambda_6 = \lambda_4 - Y_2$	$\lambda_9 = \lambda_7 \lambda_3^2 - 2X_3$
$\lambda_7 = \lambda_1 + X_2$	$Y_3 = \left(\lambda_9 \lambda_6 - \lambda_8 \lambda_3^3\right)/2$

Аналізуючи вище зазначену послідовність дій для отримання загальної суми точок змішаним способом, можна визначити витрати 11 множень. Отже, затрати на додавання точок у змішаних координатах є найнижчі.

1.1.2.4 Еліптичні криві над полем характеристики 2.

Для поля характеристики 2, вище наведені рівняння приймають інший вид, іншою є також форма подання. Представлення базується на поліноміальних або нормальних базисах. Детальніша інформація цю тему міститься в [11, 41, 95]. Обчислення для характеристики два приймають дещо іншу форму. Вона може бути записана таким чином [41]:

$$x_3 = A^2 + A + a_2 - x_1 + x_2, \tag{1.24}$$

де $A = \frac{y_2 + y_1}{x_2 + x_1}$,

$$y_3 = (x_1 + x_3)A + x_3 + y_1, \tag{1.25}$$

якщо $P_1 \neq P_2$ і $P_1, P_2 \neq$ O, а крива задана рівнянням $x^2 + xy = x^3 + a_2 x^2 + a_6$.

Для випадку подвоєння точок, коли $X_1 = X_2 \neq 0$, одержується:

$$A = \frac{x_1^2 + y_1}{x_1}. \tag{1.26}$$

Аналогічно, як і в попередньому пункті, надалі проаналізовано затрати на отримання суми двох точок. Для еліптичних кривих над полем порядку два в афінних координатах затрати на отримання суми точок зумовлені однією оберненістю в полі та двома множеннями. З точки зору способу представлення точок, базованого на базисах, чи то експонентних, чи то нормальних, упущено затрати на піднесення до квадрату у зв'язку із незначними затратами на виконання цієї операції.

В подальшому розглядається задача представлення кривої в проективних координатах (X, Y, Z), де при перетворенні афінних координат третя координата Z отримує значення 1. Так як і для випадку кривих над полем вищих порядків, так і для кривих над полем другого порядку в розрахунках особливо затратною є операція знаходження оберненого елемента в полі. Використання проективних координат дозволяє позбутися від знаходження оберненого елемента в полі за рахунок збільшення інших операцій. У таблиці 1.4 показано схему способу отримання суми двох точок, записаних в проективних координатах.

Таблиця 1.4

Модель обчислень для додавання точок на кривій $GF(2^m)$у проективних координатах.

$\lambda_1 = X_1 Z_2^2$	$\lambda_7 = Z_1 \lambda_3$
$\lambda_2 = X_1 Z_1^2$	$\lambda_8 = \lambda_6 X_2 + \lambda_7 Y_2$
$\lambda_3 = \lambda_1 + \lambda_2$	$Z_3 = \lambda_7 Z_2$
$\lambda_4 = Y_1 Z_2^3$	$\lambda_9 = \lambda_6 + Z_3$
$\lambda_5 = Y_1 Z_1^3$	$X_3 = a Z_3^2 + \lambda_6 \lambda_9 + \lambda_3^3$
$\lambda_6 = \lambda_4 + \lambda_5$	$Y_3 = \lambda_9 X_3 + \lambda_8 \lambda_7^2$

Можна зазначити, що затрати на операцію додавання зумовлені виконанням 15 множень. Подвоєння точок у проективних координатах зводиться до здійснення операцій, наведених в таблиці 1.5 згідно з [41], і передбачає реалізацію п'яти множень.

Таблиця 1.5

Модель обчислень для подвоєння точок на кривій GF(2^m) записаних в проективних координатах.

$Z_3 = X_1 Z_1^2$	$\lambda = Z_3 + X_1^2 + Y_1 Z_1$
$X_3 = (X_1 Z_1^2)^4$	$Y_3 = X_1^4 Z_3 + \lambda X_3$

Прийнято припущення, що координата однієї з точок додавання записана у проективних координатах, а другої – в афінних. Підлягли операції додавання точки, записані змішаним чином [41]. У таблиці 1.6 наведено послідовність операцій, необхідних для додавання точок.

Таблиця 1.6

Модель обчислень для додавання точок на кривій $GF(2^m)$ у змішаних координатах.

$\lambda_1 = X_2 Z_1^2$	$\lambda_6 = \lambda_4 X_2 + \lambda_5 Y_2$
$\lambda_2 = X_1 + \lambda_1$	$Z_3 = \lambda_5$
$\lambda_3 = Y_2 + Z_1^3$	$\lambda_7 = \lambda_4 + Z_3$
$\lambda_4 = \lambda_1 + \lambda_3$	$X_3 = aZ_3^2 + \lambda_7 \lambda_4 + \lambda_2^3$
$\lambda_5 = Z_1 \lambda_2$	$Y_3 = \lambda_7 X_3 + \lambda_6 \lambda_5^2$

Аналізуючи вищевказану послідовність дій для отримання суми точок змішаним чином, можна визначити затрати, необхідні на виконання 11 множень. З цього випливає, що затрати на додавання точок, записаних в змішаних координатах, є найнижчими.

Підводячи підсумки вищенаведеного розгляду щодо еліптичних кривих, ключову інформацію з точки зору подальшого аналізу зібрано в таблиці 1.7 на основі [41].

Таблиця 1.7

Затрати на виконання основних дій на ЕК в залежності від прийнятого подання.

Дія	Крива	Координати		
		афінні	проективні	мішані
Додавання	$GF(p)$	$1O+3M$	$16M$	$11M$
	$GF(2^m)$	$1O+2M$	$15M$	$11M$
Подвоєння	$GF(p)$	$1O+4M$	$10M$	–
	$GF(2^m)$	$1O+2M$	$5M$	–
Конверсія	Проективні	$1O+3M$	–	–
	Афінні	–	*Відсутні дані*	–

Тут O – операція знаходження оберненого елемента в полях $GF(p)$ та $GF(2^m)$;
M – множення в полях $GF(p)$ та $GF(2^m)$.

1.2. Дискретне логарифмування на еліптичній кривій до розв'язання задач інформаційно-управляючих систем

1.2.1 Дискретний логарифм на еліптичних кривих.

Ключовим питанням з точки зору безпеки ECC є задачі дискретного логарифмування на еліптичній кривій ECDLP (Elliptic Curve Discrete Logarithm Problem), представлені, наприклад, в [11, 12, 45, 54, 63, 68, 106, 112, 113] наступним чином. Дано еліптичну криву E, визначену над скінченним полем, точку P порядку n і точку Q, яка є кратна точці P. Слід знайти ціле число $k \in <0, n-1>$, таке, що

$$Q = k \cdot P. \tag{1.27}$$

Число k – дискретний логарифм Q з основою P

$$k = log_P Q. \tag{1.28}$$

За даними літератури [58, 59], є багато можливостей для вирішення цього завдання. Методами знаходження дискретних логарифмів, є між іншим:

- використання добутку Вейля, представленого в [53, 67, 69, 119], на $E[n]$ уможливлює редукцію задачі дискретного логарифму на кривій в мультиплікативній групі поля (розвинене в працях [49, 61, 88]),
- атака для аномальної еліптичної кривої, подана в [41, 70, 90], передбачає розв’язок ECDLP та вимагає лінійного часу (обговорено в [85] і [95]). Цей метод є зближеним методові Семаєва, наведеному в [99] для ECDLP;
- метод малих і великих кроків (BSGS), наведений в [41], який може бути використаний для вирішення проблеми дискретного логарифму в довільній скінченній абелевій групі. Цей метод є стандартним прикладом виграшу в часі за рахунок пам'яті;
- ро-метод Полларда, в якому використано випадкові блукання. Даний метод детальніше розглянуто в подальшій частині цього розділу;
- лямбда-метод Полларда.

Беручи до уваги наведені в п. 1.1.1 показники живучості ІУСЕК, вбачається зв'язок між дискретним логарифмом, зокрема часом його розв’язку та цією складовою гарантоздатності системи, а саме – рівень живучості ІУСЕК знаходиться в прямопропорційній залежності від часу, необхідного на проведення атаки на криптографічний пристрій на еліптичних кривих ECCD, що входить до складу системи.

1.2.2 Розв’язок дискретного логарифма.

Ро-метод Полларда є одним з декількох методів, розроблений Джоном Поллардом, який слугує для розв’язку дискретного логарифму [80, 93]. Цей метод базується на одному випадковому блуканні поодинокою стежкою аж до закриття циклу, колізії точок. Він полягає на знаходженні таких двох пар чисел (α_1, β_1) і (α_2, β_2), щоб для двох точок P і Q, де $Q = k \cdot P$, отримати (що детальніше описано в [68]):

$$\alpha_1 P + \beta_1 Q = \alpha_2 P + \beta_2 Q \ . \quad (1.29)$$

З рівняння (1.29) обчислюється дискретний логарифм

$$k = {}^{(\alpha_1 - \alpha_2)}/_{(\beta_2 - \beta_1)} \ (mod\ n). \quad (1.30)$$

Припущення, представлені в джерелах [41, 68, 105] стосовно коефіцієнтів (α, β), передбачають, що вони є цілими випадковими числами в діапазоні $[0; n-1]$. Вони слугують для обчислення

$$X = \alpha P + \beta Q \ . \quad (1.31)$$

Для того щоб знайти колізію, відповідно до джерел [41, 68, 105], розраховано від 16 до 32 пар випадкових дійсних чисел $a_j, b_j \in_R [0, n-1] \wedge j \in [0; l]$, де l є числом випадкових пар a_j, b_j. Потім обчислено l точок згідно з рівнянням

$$R_j = a_j P + b_j Q \ . \quad (1.32)$$

Припускаючи, що X належить до кривої GF(p), обчислено наступні точки відповідно до виразу

$$X' = X' + R_j \ , \quad (1.33)$$

де $j = H$(X) та

$$\alpha' = \alpha' + a_j mod n \ , \quad (1.34)$$

$$\beta' = \beta' + b_j mod n \quad . \tag{1.35}$$

Рисунок 1.4 ілюструє послідовні ітерації ро-алгоритму Полларда до досягнення точки колізії.

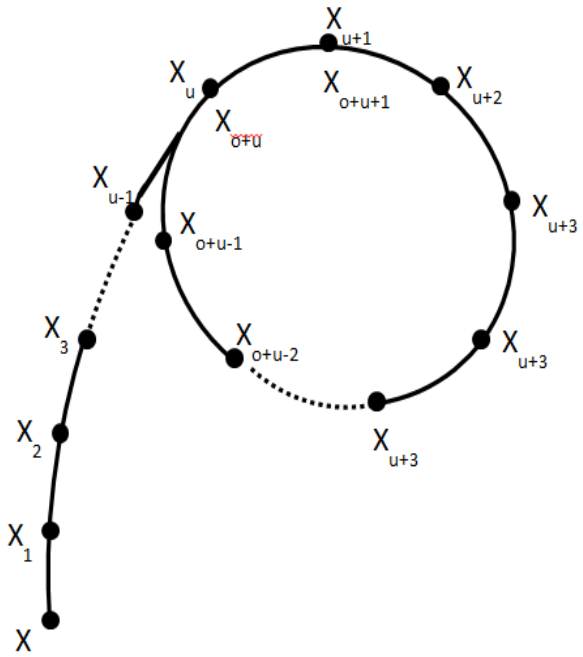

Рисунок 1.4 – Ілюстрація ро-алгоритму Полларда

З рисунка 1.4 можна зазначити, що від точки зіткнення всі наступні отримані точки у відповідності з ро-алгоритмом Полларда збігаються. Дискретний логарифм обчислюється за такою формулою:

$$k = (\alpha' - \alpha'')(\beta'' - \beta')^{-1} mod n, \tag{1.36}$$

якщо $\beta' \neq \beta''$.

З метою прискорення обчислень дискретного логарифму надалі застосований модифікований ро-метод Полларда, так званий паралельний метод, запропонований і розвинений в працях [66, 68, 90]. Ідея цього методу полягає в урухомленні багатьох стежок блукання, реалізованих в тому самому часі, причому стартова (відправна) точка X є різною для кожної стежки блукання та її вибрано випадково в таблиці В.1,

точки 9 і 10 (Додаток В). Таблиця В.1 містить опис паралельного алгоритму згідно з ро-методом Полларда, складений на основі алгоритму, поданого в праці [90]. Іншим припущенням, що дозволяє лінійний приріст [68, 90], є доручення роботи над одною стежкою блукання одному процесору. За даними [68], кількість кроків *D*, необхідних для знаходження колізії, дорівнюватиме значенню, обчисленому з рівняння

$$D = \left. \sqrt{\pi n / 2} \right/ M, \tag{1.37}$$

де *M* – кількість процесорів, які реалізують стежки блукання.

Отже, постає необхідність удосконалення моделей та технологій обчислень на ЕК у засобах ІУСЕК для покращення показників їх живучості.

1.2.3 Параметри кривих і дискретний логарифм.

Параметри еліптичних кривих вибираються таким чином, щоб протистояти всім відомим атакам, які базуються в основному на здійсненні обчислення кратності точки *P* аж до отримання *Q* [11]. Необхідний час для проведення атаки становить від *n* кроків, де *n* представляє собою порядок точки, тобто кількість точок в циклічній підгрупі групи точок еліптичної кривої, в середньому визначається кількість кроків на *n*/2. Запобігання атаці на сучасному рівні знань можна досягнути за рахунок задання досить великого порядку точки *n* і вибору відповідних параметрів для кривої, особливості чого описано в праці [68].

В наступній частині цього розділу представлено ті атаки, які не вимагають якихось додаткових припущень щодо групи, тому що з точки зору криптографії таким атакам нескладно запобігти. Слід згадати тут наступні атаки: Pohlig і Hellman, MOV або ро-метод Полларда чи його паралельна версія (MPPR).

1.3. Аналіз атак на системи, базовані на еліптичних кривих

1.3.1 Атака, що ґрунтується на ро-методі Полларда.

Цей метод базується на випадковому блуканні поодинокою стежкою аж до закриття циклу, колізії точок. Він полягає на знаходженні таких двох пар чисел (α_1, β_1) і (α_2, β_2) , щоб для двох точок *P* і *Q* отримати рівняння яке відповідає (1.29). Знаючи, що $Q = K \cdot P$, отримується рівняння

$$(\alpha_1 - \alpha_2)P = (\beta_2 - \beta_1)kP \ , \tag{1.38}$$

так що

$$(\alpha_1 - \alpha_2) \equiv (\beta_2 - \beta_1)k \pmod n. \tag{1.39}$$

З наведеного вище рівняння можна обчислити дискретний логарифм (1.30).

Пари коефіцієнтів (α, β) є випадковими цілими числами, які знаходяться в діапазоні $[0; n-1]$. Маючи випадково отримані коефіцієнти, здійснюється обчислення *X* згідно з (1.31).

На підставі отриманої точки *X* запам'ятовується трійка (α, β, X), причому такі обчислення здійснюються доти, доки точка не повториться – тобто має місце колізія чи зіткнення точок. В розрахунках припускається, що зіткнення відбувається в середньому після $\sqrt{\pi n/2}$ кроків, де *n* є порядком групи [68, 101]. Нижче пояснено ідею атаки, основаної на ро-методі Полларда. Припускається, що *G* є циклічною групою, генерованою через *P,* точка *X* є елементом групи $\langle P \rangle$. Визначено функцію *F* таку, що для $X = aP + bQ, \qquad X' = a'P + b'Q$, де $a, b', b, b' \in [0, n-1]$, отримується $X' = f(X)$. Згодом, відповідно до [57, 64], виконується випадковий поділ групи $\langle P \rangle$. Результати, які отримані експериментальним шляхом та представлені в праці [102], дозволяють вибрати поділ на близько 20 підгруп.

Вибирається функція поділу *H* така, що для $H(X) = i$. Одержується рівняння:

$$f(X) = X + a_i P + b_i Q \tag{1.40}$$

З даного рівняння отримуються: $\alpha' = \alpha + a_i mod\, n$ і $\beta' = \beta + b_i mod\, n$.

Випадкове блукання відбувається відповідно до прийнятого алгоритму, по встановленій стежці, тому недолік цього методу можна усунути, а саме зменшити обсяг пам'яті, необхідної для зберігання обчислених точок X_i. Обмеження обсягу пам'яті досягається завдяки запису в базу даних лише деяких точок, які називаються виділеними точками. За критерій виділення зазвичай приймається значення найменш значущих бітів координати x_i точки X_i [94]. Зменшення пам'яті відбувається за рахунок додаткових додавань. Обсяг пам'яті, яка необхідна для зберігання трійок (a_i, b_i, X_i), становить [41]:

$$M = \sqrt{n\pi/2}\Big/ 2^s. \tag{1.41}$$

Як можна зауважити, при розгляді виразу (1.37), остаточний обсяг пам'яті можна зменшити завдяки використанню відповідного значення коефіцієнта *s*. На жаль, наслідком цієї дії є збільшення кількості додавань, які необхідно виконати для досягнення зіткнення, в 2^s разів.

1.3.2 Атака на основі паралельного ро-методу Полларда.

У цьому методі використовується не одна, а багато стежок блукання, реалізованих паралельно. Припущенням є реалізація одної стежки блукання за допомогою одного обчислювального процесора з передаванням виділених точок до спільної бази, де в подальшому виконується перевірка точок і пошук колізії. В паралельному ро-методі Полларда здійснюється пошук колізії двох стежок блукання,

причому не обов'язково до закриття циклу. Схематично дію паралельного ро-методу Полларда представлено на рисунку 1.5.

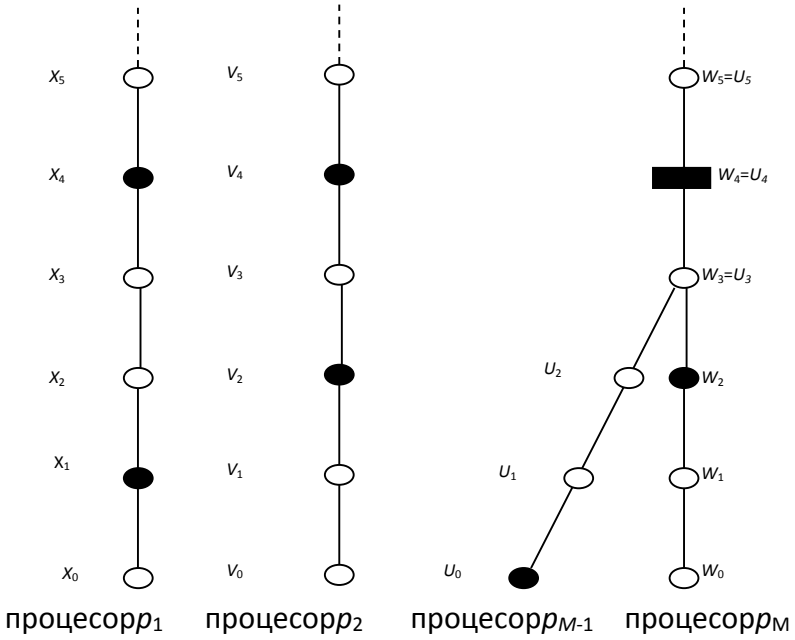

Рисунок 1.5 – Принцип дії паралельного ро-методу

На даний час це найшвидша відома атака на ЕК. Використання багатьох стежок блукання, реалізованих окремими процесорами, призводить до лінійного збільшення продуктивності [41, 67, 75]. В середньому зіткнення точок для випадку одної стежки блукання можна знайти після

$$C = \sqrt{{}^{1}/_{2}\,\pi n} + 2^{s} \tag{1.42}$$

виконаних додавань.

В паралельній версії ро-методу Полларда є змога отримати лінійний ріст швидкості дій. Тому, якщо припускається, що наявні M процесорів p_i, за допомогою яких виконується ро-метод Полларда, то можна оцінити остаточний час для досягнення колізії згідно з виразом

$$C = \frac{\sqrt{1/2^{\pi n}}}{M} + 2^{s}. \quad (1.43)$$

Завдяки методам прискорення виконання основних операцій на еліптичних кривих і розв’язання дискретного логарифма, а також введенню до обчислень спеціально побудованих апаратно-програмних засобів можна точніше оцінити живучість ІУСЕК і підвищити ймовірність зниження його рівня внаслідок суттєвого скорочення часу, необхідного на проведення атаки. Це дає змогу довести необхідність верифікації та покращення оцінювання живучості як важливої складової гарантоздатності даних систем.

Висновки до розділу 1

1. Аналіз підходів до загальних питань живучості інформаційно-управляючих систем, включаючи системи на основі еліптичних кривих, дозволив сформулювати відповідні технічні і наукові дані, що лежать в основі цієї складової гарантоздатності, та визначити основні завдання дослідження.

2. Подано основні типи еліптичних кривих та їх застосування в криптографії з точки зору живучості інформаційно-управляючих систем, що дало змогу проаналізувати обчислювальну складність для різних представлень кривої.

3. Визначено зв'язок дискретного логарифма на еліптичній кривій з живучістю ІУСЕК та вплив на неї способів розв’язку логарифма в криптографічних засобах систем, завдяки чому обґрунтовано вибір методу для розв’язку дискретного логарифму підвищеної швидкодії.

4. На підставі аналізу методів атак на системи, які базуються на еліптичних кривих, та стійкості до них виявлено, що завдяки методам прискорення виконання основних операцій на еліптичних кривих, а також введенню до обчислень спеціально побудованих засобів підвищується імовірність зниження рівня живучості ІУСЕК внаслідок суттєвого скорочення часу, необхідного для виявлення атаки. Це дало змогу довести необхідність верифікації та покращення оцінювання живучості даних систем.

Отже, завдання дослідження представимо так:

1. Провести аналіз принципів побудови, технологічних рішень і напрямів розвитку ІУСЕК з точки зору їх адекватності та можливого використання для розв’язання поставленої задачі.

2. Розробити, ґрунтуючись на покращених та ефективніших методах розв'язання дискретного логарифма, моделі та технології обчислень на ЕК в *GF(p)* у пристроях ІУСЕК для виконання криптографічних операцій.

3. Створити апаратно-програмні засоби для виконання обчислень на ЕК в *GF(p)* і розв’язання дискретного логарифма підвищеної швидкодії для точнішого оцінювання та покращення живучості як важливої складової гарантоздатності ІУСЕК.

4. Оцінити вплив способів збільшення швидкості виконання основних операцій на ЕК над скінченним полем вищих порядків GF(p) на зміну рівня живучості ІУСЕК для кожної з опрацьованих моделей та обчислювальних технологій.

5. Провести симуляційні дослідження (імітаційне моделювання) для верифікації отриманих теоретичних залежностей щодо живучості ІУСЕК.

6. На підставі отриманих моделей підвищення живучості ІУСЕК реалізувати імплементацію запропонованих рішень в практиці.

РОЗДІЛ 2
ВИБІР ОБЧИСЛЮВАЛЬНИХ ПЛАТФОРМ І ЗАСОБІВ ДЛЯ СТВОРЕННЯ ЖИВУЧИХ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ

У другому розділі представлено структурну модель підвищення живучості ІУС з врахуванням дефектів зовнішніх впливів, обчислювальні платформи та компоненти, за допомогою яких здійснюються основні обчислення на еліптичних кривих і цілих числах великої розрядності на потреби створення живучих ІУС. Розглянуто питання розпаралелювання операцій, що виконуються як на числах великої розрядності, так і на кривих. Висвітлено способи побудови апаратних та апаратно-програмних систем, що реалізують паралельний ро-метод Полларда. Подано моделі та технології обчислень для випадку сумування і множення чисел великої розрядності та реалізації основних операцій на еліптичних кривих. Продемонстровано, що вибір відповідно швидких обчислювальних алгоритмів та ефективна імплементація в специфічних умовах можуть помножити швидкість розв’язання дискретного логарифма, а отже призвести до порушення безпеки ІУСЕК. Обґрунтовано, що для визначення живучості ІУСЕК слід змодифікувати алгоритми та імплементаційні рішення в специфічних обчислювальних середовищах, а також оцінити швидкість розв’язання логарифму.

2.1. Структурна модель підвищення живучості інформаційно-управляючих систем

Розроблення структурної моделі підвищення живучості ІУС проведено з врахуванням дефектів ДВ, що згідно з [3] дає сформулювати завдання вибору (пошуку) оптимальної стратегії обслуговування гарантоздатної ІУС за показниками готовності технічного та інформаційного станів з огляду на всю множину дефектів. При цьому взято до уваги зміну інформаційно-технічних станів, в одному з восьми яких може перебувати ІУС. Скористаємося напрямленим графом інформаційно-технічних станів (рисунок 2.1), наведеним в [3].

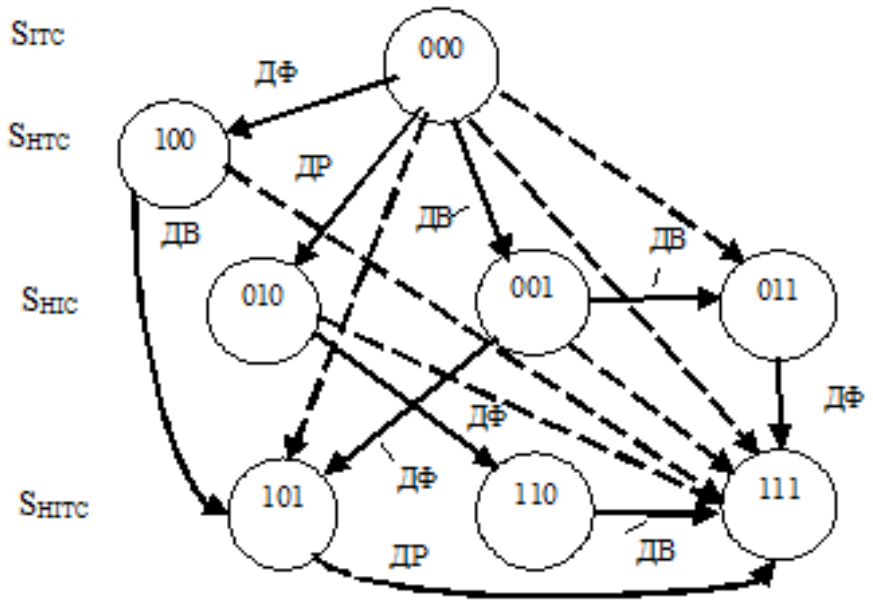

Рисунок 2.1 – Напрямлений граф інформаційно-технічних станів ІУС

На рисунку 2.1 позначено: $S_{ІТС}$ – інформаційно-технічний стан, $S_{НТС}$ – непрацездатний технічний стан, $S_{НІС}$ – непрацездатний інформаційний стан, $S_{НІТС}$ – несправний (непрацездатний, небезпечний) інформаційно-технічний стан ІУС. Кожному стану та відповідній вершині графу станів надано трирозрядний код, який описує наявність відповідних дефектів ДФ, ДР і ДВ (1 – дефект присутній, 0 – відсутній). Переходи на графі, на яких виникають два або три дефекти різних типів, показані пунктиром як менш ймовірні. Приміром, у множину $S_{НІТС}$ входять підмножини $S_{НІТС1}$(101), $S_{НІТС2}$(110), $S_{НІТС3}$(111), залежно від комбінації дефектів ДР і ДВ, за наявності ДФ, причому в розглянутих автором ІУС, включно з вбудованими криптографічними пристроями на еліптичних кривих, та їх моделях враховуються лише ДВ.

Модель живучості ІУС, зокрема ІУСЕК можна представити сукупністю відповідної кількості часткових моделей різного призначення, в яких для опису процесів застосовуються як детерміновані, так і імовірнісні методи. Введемо позначення для множини *MS* інформаційно-технічних станів ІУСЕК: *ПС* – працездатний стан, *ЧП* – частково працездатний стан, *НБС* – працездатний безпечний стан, *НС* – небезпечний стан. Один із підходів оцінювання

гарантоздатності ґрунтується на розробленні та аналізуванні структурних моделей та схем живучості системи. Беручи за основу наведене в стандарті [3], можна отримати об'єднання підмножин елементів ІУС, включно з елементами на ЕК, відмови яких призводять до її переходу в інший стан, для різних вихідних станів системи, а саме:

$$E_{ПС} = E_{ПС,ПС} \cup E_{ПС,ЧП} \cup E_{ПС,НБС} \cup E_{ПС,НС}, \quad (2.1)$$

$$E_{ЧП} = E_{ЧП,ЧП} \cup E_{ЧП,НБС} \cup E_{ЧП,НС}, \quad (2.2)$$

$$E_{НБС} = E_{НБС,\,НБС} \cup E_{НБС,НС}. \quad (2.3)$$

Приміром, перше із співвідношень стосується випадку перебування ІУС у працездатному (вихідному) стані, для котрого множину її елементів $E_{ПС}$, включно з криптографічними на ЕК, можна подати у вигляді об'єднання підмножин елементів, відмови яких призводять до її переходу в інший працездатний стан – $E_{ПС,ПС}$, частково працездатний стан – $E_{ПС,ЧП}$, непрацездатний безпечний стан – $E_{ПС,НБС}$, і небезпечний стан – $E_{ПС,НС}$. Тоді для кожної із груп станів $MS_{ПС}$ і $MS_{ЧП}$ ІУС згідно з виразами (2.1)-(2.3) запропоновано побудувати відповідні структурні моделі живучості, враховуючи наявність криптографічних елементів на ЕК і беручи до уваги оцінювання живучості за станом системи та результатами виконання завдання:

1) для станів $MS_{ПС}$ схему живучості утворено елементами підмножин $E_{ПС,ПС}$ і $E_{ПС,ЧП}$ з можливим паралельним ввімкненням, $E_{ПС,НБС}$ і $E_{ПС,НС}$ з послідовним ввімкненням, модулем 1 оцінювання живучості ІУС за її станом, модулем 2 оцінювання живучості ІУС за результатами виконання нею завдання, моделлю прийняття рішення (ІПР) про способи підвищення живучості (включно із змінами структури та параметрів ІУС, а також додатковим удосконаленням пасивних та активних засобів забезпечення живучості), якщо оцінки вказують на її незадовільний рівень (рисунок 2.2);

2) для станів $MS_{ЧП}$ схему живучості утворено елементами підмножин $E_{ЧП,ЧП}$ з можливим паралельним ввімкненням, $E_{ЧП,НБС}$ і $E_{ЧП,НС}$ з послідовним ввімкненням, модулем 1 оцінювання живучості ІУС за її станом, модулем 2 оцінювання живучості ІУС за результатами виконання нею завдання, моделлю прийняття рішення (ІПР) про способи підвищення живучості (включно із змінами структури та параметрів ІУС, а також додатковим удосконаленням пасивних та активних засобів забезпечення живучості), якщо оцінки вказують на її незадовільний рівень (рисунок 2.3).

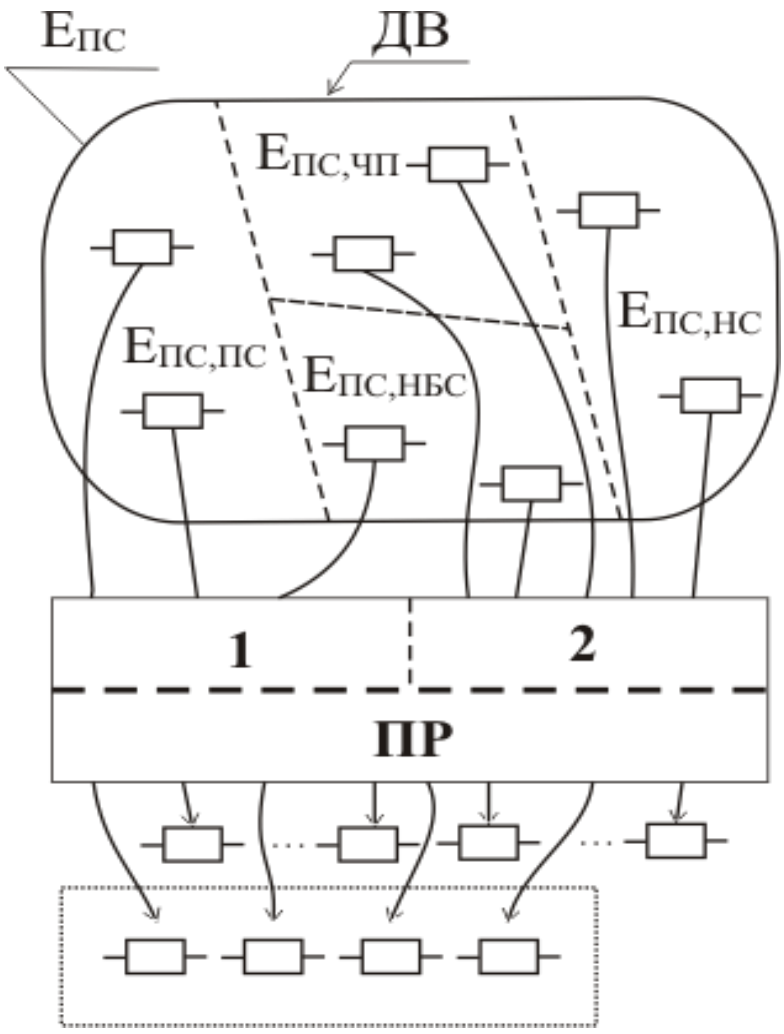

Рисунок 2.2 – Структурна модель живучості ІУСЕК для вихідного стану $MS_{ПС}$

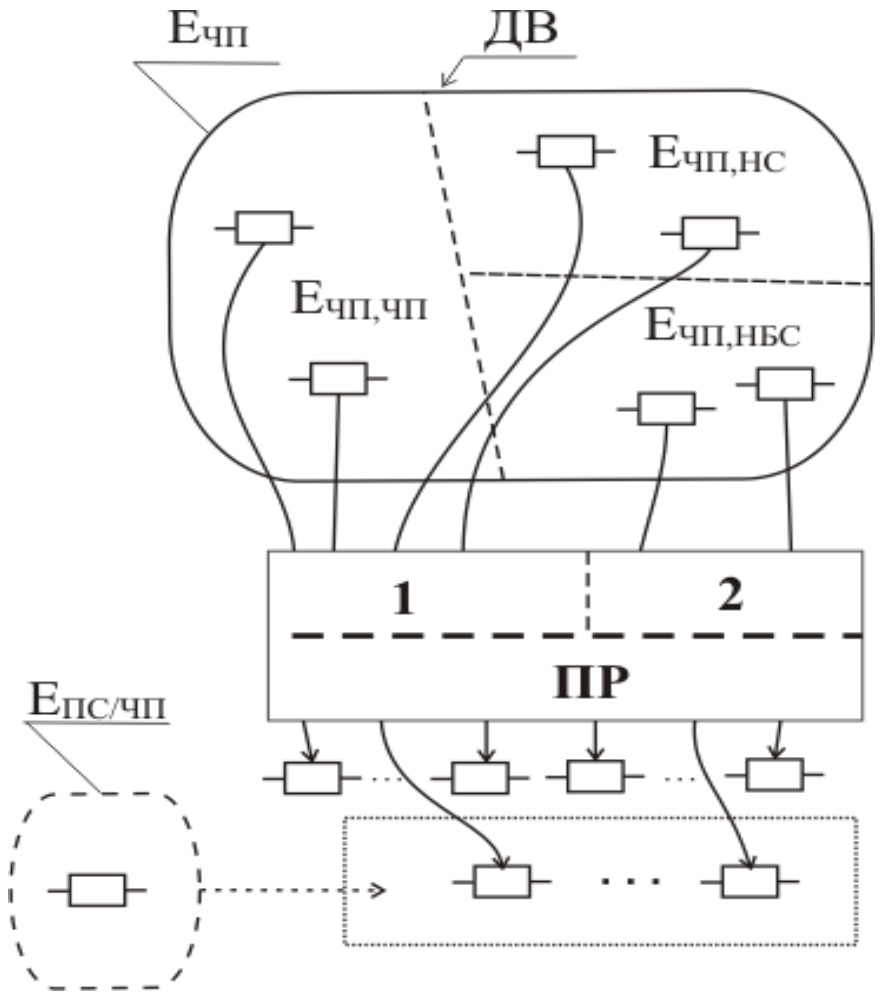

Рисунок 2.3 – Структурна модель живучості ІУСЕК для вихідного стану $MS_{ЧП}$

В наведених на рисунках 2.2 та 2.3 моделях дефекти ДВ можна подати у вигляді точкової та просторової моделі, якщо класифікувати їх за областю дії [26]. У точкових моделях ДВ викликає відмову одного або декількох елементів. Для одноточкової області за наявності K елементів ІУС одним з можливих розподілів є рівномірний розподіл $\alpha_i = 1/K$, тоді як для багатоточкової області в моделях можна застосувати зрізаний біноміальний розподіл

$$\beta_i = C_K^i p^i (1-p)^{K-i} / (1- p^K),\ i = 1, \ldots, K, \qquad (2.4)$$

та зрізаний розподіл Пуассона

$$\beta_i = \frac{a^i}{i!} / \sum_{j=1}^{K} \frac{a^j}{j!}, a = -\ln p \,, \qquad (2.5)$$

де p – ймовірність виживання одиничного елемента в точковій моделі.

В просторових моделях можна задати двомірний розподіл декартових координат епіцентру ДВ $p_2(x_0, y_0)$ і розподіл радіуса круга $p_0(r_0)$, в якому діє ДВ. За законом розподілу інтенсивності ДВ можна врахувати в моделі дефекти з нескінченною інтенсивністю, з постійною інтенсивністю I по всій площі області дії та зі спадною від епіцентру за певним законом $I(r,\varphi)$ інтенсивністю, зокрема, за законом Релея:

$$I(r, \varphi) = I_0 \exp(-r^2 / a r_0^2), \qquad (2.6)$$

де I_0 – максимальна інтенсивність в епіцентрі, r_0 – радіус круга - області дії ДВ, a – постійний параметр, r і φ – полярні координати точки при розташуванні початку координат в епіцентрі.

Для введення в модель тривалості дії ДВ слід врахувати їх імпульсний характер або дефекти с з постійною і випадковою тривалістю. Пунктирними лініями на рисунках показано паралельно ввімкнені (резервні) групи елементів або вимкнені групи елементів у структурній схемі живучості ІУС. Оцінювання живучості ІУС за її станом (модуль 1) та за результатами виконання нею завдання (модуль 2) пропонується проводити на підставі таких моделей:

- модель технічної та функціонально-алгоритмічної структури ІУС, включно з моделями функціонування і характеристик криптографічних елементів на ЕК, топології системи, маршрутів інформаційних потоків, функціональної та структурної ієрархії;
- модель фізичних процесів, зокрема для аналізу перехідних процесів в ІУС після ДВ;

- модель первинних наслідків, що отримується шляхом взаємодії моделі фізичних процесів з моделлю ДВ, але без врахування впливів керування зі сторони засобів забезпечення живучості;

- модель засобів забезпечення живучості, яка відображає характеристики, в тому числі ймовірнісні, засобів контролю, аварійного захисту, реконфігурування та керування, причому за допомогою агоритмів прийняття рішень за забезпечення живучості, що входять в цю модель, формуються дії керування, спрямовані на зміну структури і параметрів ІУС і на використання внутрішніх резервів, створених для роботи в екстремальних ситуаціях;

- модель розвитку первинних наслідків, що отримується внаслідок поєднання моделі первинних наслідків і моделі засобів забезпечення живучості та завдяки якій можна визначити новий стійкий стан ІУС, причому результати аналізу цієї моделі також можуть бути подані в ймовірнісній формі, оскільки деякі характеристики засобів забезпечення живучості є ймовірнісними;

- модель вторинних наслідків, що відображає ті віддалені наслідки ДВ, які можуть виникати в системі внаслідок скорочення обсягу виконуваних функцій і погіршення технічних характеристик ІУС, причому до цих вторинних наслідків можна віднести збільшення часу виконання функцій, додаткове поширення помилок в системі, підвищене споживання для виконання тих самих функцій та інші наслідки, які зумовлюють скорочення резервів, що залишилися після ДВ в ІУС, і подальше погіршення технічних характеристик;

- модель відновлення, яка містить опис аварійних ресурсів, правил і способів їх використання в екстремальних ситуаціях з метою відновлення технічної та функціонально-алгоритмічної структури тієї частини ІУС, яка зайнята у виконанні встановленого завдання, причому цю модель можна вважати моделлю розвитку ІУС після закінчення ДВ;

- модель процесів виконання завдання, що отримується в результаті об’єднання таких моделей – моделі технічної та функціонально-алгоритмічної структури, моделі фізичних процесів, моделі вторинних наслідків і моделі

відновлення, причому завдяки аналізу цієї моделі можна оцінити живучість ІУС за результатами виконання нею завдання.

Для оцінювання та підвищення живучості ІУСЕК доцільно: а) провести аналіз відмов елементів та оцінити вплив їхніх наслідків на працездатність ІУСЕК і сформулювати множини інформаційно-технічних станів $MS_{ПС}$, $MS_{ЧП}$, $MS_{НБС}$ і $MS_{НС}$; б) в залежності від наслідків відмов здійснити розбиття множини елементів E на підмножини $E_{ПС}$ ($E_{ПС,\,ПС}$, $E_{ПС,ЧП}$, $E_{ПС,НБС}$, $E_{ПС,НС}$), $E_{ЧП}$ ($E_{ЧП,ЧП}$, $E_{ЧП,НБС}$, $E_{ЧП,НС}$) і $E_{НБС}$ ($E_{НБС,НБС}$, $E_{НБС,НС}$); в) побудувати структурну модель живучості ІУСЕК для всіх груп інформаційно-технічних станів; г) одержати відповідно до цих моделей вирази для обчислення показників живучості. До найскладніших із цих задач, як свідчать результати досліджень [3], належать етапи а) та б). Їх можна виконати за допомогою відповідної методики аналізу.

2.2. Порівняльна характеристика моделей виконання арифметичних операцій на еліптичних кривих в задачах живучих інформаційно-управляючих систем

2.2.1 Модель модульного множення, основаного на теоретико-числовому базисі Радемахера-Крестенсона.

Необхідність множення чисел великої розрядності в комп'ютерних системах призвела до того, щоб сягнути до нестандартних рішень. У даній роботі застосовано рішення на основі теоретико-числових базисів Радемахера-Крестенсона [83, 103]. Використання алгоритму множення Крестенсона уможливлює зведення множення до операції додавання, яка реалізовується в інформаційних системах на основі попередньо згенерованих таблиць. Прийнято припущення, що необхідно помножити два числа x і y за модулем p. Числа x і y представлено у вигляді:

$$\begin{aligned} x &= x_{r-1}2^{r-1} + x_{r-2}2^{r-2} + \ldots + x_i 2^i + \ldots + x_1 2^1 + x_0 2^0 \\ y &= y_{r-1}2^{r-1} + y_{r-2}2^{r-2} + \ldots + y_j 2^j + \ldots + y_1 2^1 + y_0 2^0 \end{aligned}, \qquad (2.7)$$

де r – позиція чисел x і y, а $x_i, y_j = 0$ або 1.

Для визначення результату множення за модулем p побудовано матрицю, показану в таблиці 2.1, де $m_{ij} = 2^{i+j} \bmod p$.

Таблиця 2.1

Матриця визначення модуля перетворення в теоретико-числовому базисі Радемахера-Крестенсона

	y_{r-1}	**…**	**y_j**	**…**	**y_1**	**y_0**
x_{r-1}	$m_{r-1\ r-1}$	…	$m_{r-1\ j}$	…	$m_{r-1\ 1}$	$m_{r-1\ 0}$
…	…	…	…	…	…	…
x_i	$m_{i\ r-1}$	…	m_{ij}	…	m_{i1}	m_{i0}
…	…	…	…	…	…	…
x_1	$m_{1\ r-1}$	…	m_{1j}	…	m_{11}	m_{10}
x_0	$m_{0\ r-1}$	…	m_{0j}	…	m_{01}	m_{00}

Добуток чисел x і y отримано з моделі, представленої формулою:

$$(X \cdot Y) \bmod p = \left(\sum_{s,k=1}^{r-1} m_{sk} \right) \bmod p, \qquad (2.8)$$

де $x_s, y_k = 1$, тобто m_{sk} знаходиться на перетині стовпця і рядка, для яких відповідні x_i та y_j дорівнюють 1.

Проведені нами теоретичні дослідження вказують на те, що використання теоретико-числового базису Крестенсона для здійснення обчислень на числах великої розрядності забезпечить ефективне виконання операцій множення не тільки в програмній версії, але також дозволить реалізувати множення чисел

великої розрядності у апаратних компонентах, наприклад, таких як програмованих матрицях FPGA [34]. Наведений вище алгоритм дає змогу замінити множення, для якого притаманна квадратна обчислювальна складність, на додавання, яке характеризується лінійною обчислювальною складністю. Завдяки цьому можна досягти високої продуктивності в обчисленнях на еліптичних кривих над скінченним полем вищих порядків.

2.2.2 Методи розпаралелювання арифметичних операцій в інформаційних системах.

Для криптографічного забезпечення ІУС є потреба здійснювати розрахунки на числах великої розрядності, які занадто великі, щоб помістити їх безпосередньо в регістрах процесорів. Наявні різні заімплементовані методи, які призначені для досягнення цієї мети, в багатьох програмних бібліотеках. На жаль, для випадку застосування апаратних модулів до виконання операцій над числами великої розрядності немає можливості використання цих бібліотек. Стандартні методи додавання чисел великої розрядності не дозволяють розділити їх на коротші слова та додавати їх паралельно.

Таким чином, створюючи апаратні або програмні системи, які функціонують паралельно, слід використати методи, які дозволять поділити довільно числа великої розрядності на сегменти розміром, що дозволяє побудувати апаратні компоненти для їх обслуговування. Слід зазначити, що відомі методи, які уможливлюють запам’ятовувати числа у вигляді списку $A=(a_1, a_2, \ldots, a_n)$, де кожна з n комірок a_i містить фрагмент числа відповідного розміру [16, 17, 19, 34, 44, 86]. Вибране рішення дозволяє подавати натуральні числа великої розрядності у двійковій формі або у формі цифр, які заповнюють комірки a_i. Ця інтерпретація уможливлює паралельну роботу з кожною коміркою, що здійснюється незалежними процесами.

Приймається, що X, Y, Z є цілими числами великої розрядності. Виконано поділ цих чисел на слова відповідної довжини m у теоретико-числовому базисі δ, де $\delta=p^m$:

$$X = x_n\delta^n + x_{n-1}\delta^{n-1} + \cdots + x_1\delta^1 + x_0$$
$$Y = y_r\delta^r + y_{r-1}\delta^{r-1} + \cdots + y_1\delta^1 + y_0, \quad (2.9)$$
$$Z = z_k\delta^k + x_{k-1}\delta^{k-1} + \cdots + x_1\delta^1 + x_0$$

причому всі коефіцієнти $x_\propto, y_\beta, z_\gamma \in [0; \delta)$.

В подальшому аналізі числа *X, Y, Z* розглядаються без знаку, за умови, що $Y \leq X$.

Для випадку додавання отримано $X + Y = Z,\ r \leq k \leq n + 1$. Нескладним способом можна перенести підхід до додавання даних за допомогою вищенаведеного рівняння на додавання чисел великої розрядності. При цьому доцільно брати до уваги, що потрібно оцінити максимальну кількість комірок *k,* яка може бути зайнята під час додавання чисел великої розрядності. Виконуючи операцію віднімання, кількість зайнятих комірок для різниці двох чисел можна прийняти $X - Y = Z,\ 0 \leq k \leq n$.

Провівши поділ числа великої розрядності, можна перейти до розгляду розробки схем алгоритмів, завдяки яким можна здійснити паралельну операцію додавання та віднімання чисел, записаних у вигляді попередньо створених слів. У кожному з процесів ініціюється подвійне слово, причому в молодшому слові записується складова a_i, в старшому – b_i, за умови, що для кожного $i > r\ b_i = 0$.

В основу моделей додавання для кожного з процесів покладено наступний алгоритм.

Алгоритм 2.1. Алгоритм паралельного додавання чисел,
поділених на слова:

1. Додавання *i*-тої складової $(x_i + y_i)$
2. Запис в молодшому слові відповідного пристрою значення, обчисленого за формулою $\vartheta_i = (x_i + y_i) mod \delta$
3. На місце $(i + 1)$ в старшому слові записується значення, розраховане згідно з виразом $\omega_{i+1} = (x_i + y_i) div \delta$

4. Аналіз старшого слова з *i*-тої позиції дає змогу закінчити алгоритм при відсутності переносу або повернутися до кроку 1, якщо є перенос.

Як можна зауважити, результат Z отримується після виконання щонайбільше $(n + 1)$ кроків.

Метод віднімання для кожного з процесів ґрунтується на нижче наведеному алгоритмі.

Алгоритм 2.2. Алгоритм паралельного віднімання чисел, поділених на слова:

1. Віднімання *i*-го компонента $x_i - y_i$
2. Запис в молодшому слові відповідного пристрою значення, обчисленого згідно з виразом

$$\vartheta_i = \begin{cases} \delta + x_i - y_i, \text{якщо } x_i < y_i \\ x_i - y_i, \text{якщо } x_i \geq y_i \end{cases} \tag{2.10}$$

2. На місце $(i + 1)$ в старшому слові записується значення, розраховане за виразом

$$\omega_{i+1} = \begin{cases} 1, \text{якщо } x_i < y_i \\ 0, \text{якщо } x_i \geq y_i \end{cases} \tag{2.11}$$

4. Аналіз старшого слова з *i*-ої позиції уможливлює закінчення алгоритму при відсутності перенесення або повернення до кроку 1, якщо наявний перенос.

З вищенаведеного алгоритму випливає. що результат Z одержується після виконання щонайбільше $(n + 1)$ кроків.

2.3. Принципи вибору обчислювальних платформ для виконання арифметичних дій на еліптичних кривих в інформаційно-управляючих системах

2.3.1 Інформаційні системи для реалізації паралельного ро-методу Полларда.

Згідно з [13, 57, 62, 64, 75, 110] нижче розглянуто технологію обчислень із застосуванням паралельної системи, що реалізує ро-метод Полларда та побудованої згідно із зображеною на рисунку 2.4 схемою, особливості якої висвітлено в [66].

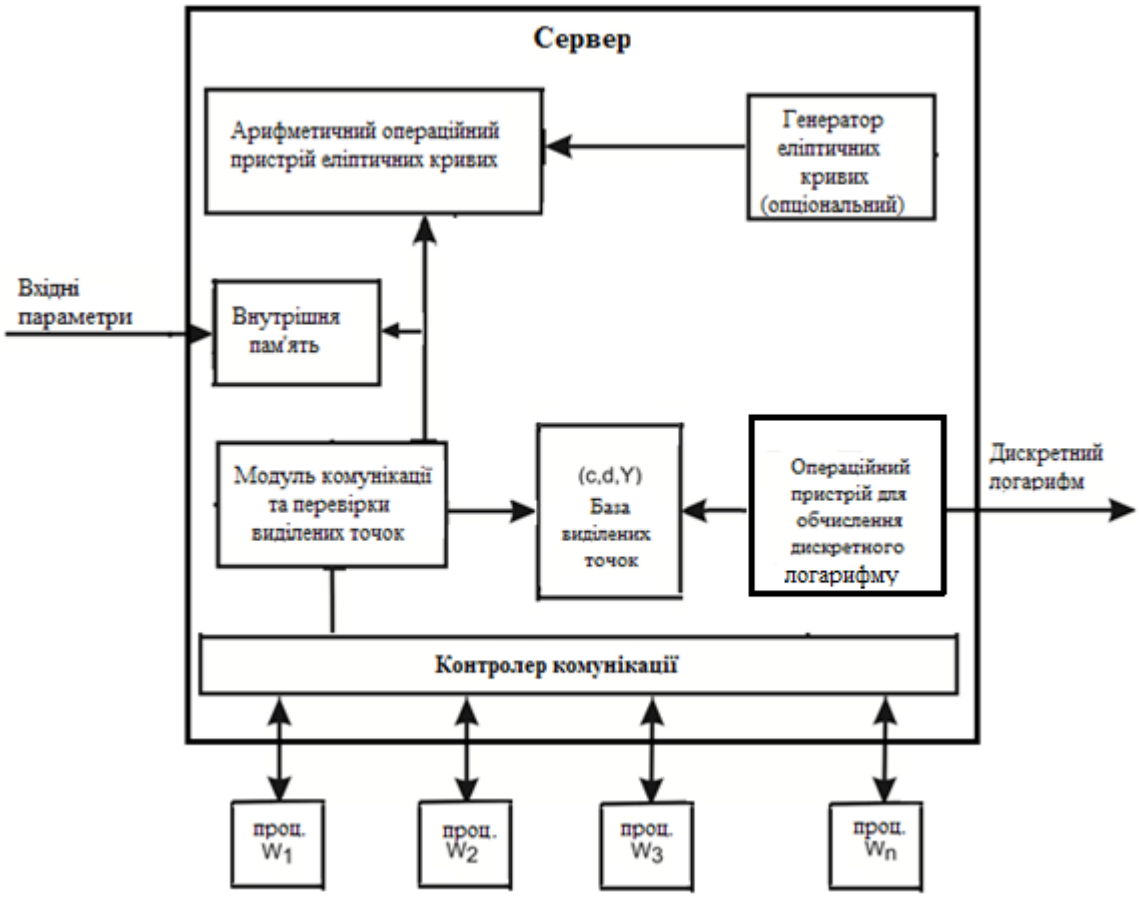

Рисунок 2.4 – Схема системи для виконання паралельного ро-методу Полларда

Така система складається з сервера, функцією якого є надання доступу до параметрів еліптичної кривої, необхідних для реалізації стежки блукання під'єднаними до нього незалежними засобами. Іншим характерним компонентом сервера є база даних, до якої записуються вибрані точки – так звані виділені точки, які розраховуються окремими пристроями, що здійснюють стежки блукання, та

якщо вони задовольняють встановленим критеріям виділення. Сервер системи оснащений також такими засобами:

а) програмне забезпечення, за допомогою якого здійснюється порівняння з базою даних точок, отриманих з пристроїв, що реалізують стежки блукання,

б) програмне забезпечення, яке дає змогу обчислювати дискретний логарифм в разі виявлення колізії. Конструкція сервера дозволяє під'єднати до нього в будь-який час новий пристрій, причому на функціонування системи не впливає від'єднання від сервера довільного засобу, що реалізує стежку блукання. Аналогічна архітектура притаманна системі, запропонованій в [43].

2.3.2. Процесори до виконання стежок блукання ро-методу Полларда для кривих $GF(2^m)$.

За основу прийнято обчислювальну платформу, яка базується на компоненті, що реалізує основні операції на еліптичній кривій над скінченним полем другого порядку $GF(2)^m$ [43, 84].Зазначена структурна одиниця реалізовує стежку блукання ро-методу Полларда, причому, будучи окремим процесором, вона може бути складовою частиною паралельної системи, розглянутої в попередньому пункті (рисунок 2.5) [43].

В цьому компоненті виконання найзатратнішої за часом та кількістю задіяних модулів операції, якою є множення, ґрунтується на алгоритмі КОА (Karatsuba-Ofman Algorithm). Останній полягає у зменшенні кількості множень, необхідних для визначення добутку двох чисел великої розрядності, за рахунок введення додаткових обчислень суми та різниці при розкладі чисел до такого вигляду

$$x = x_1 P^m + x_2, \qquad y = y_1 P^m + y_2, \tag{2.12}$$

де x_2 і y_2 менші, ніж P^m.

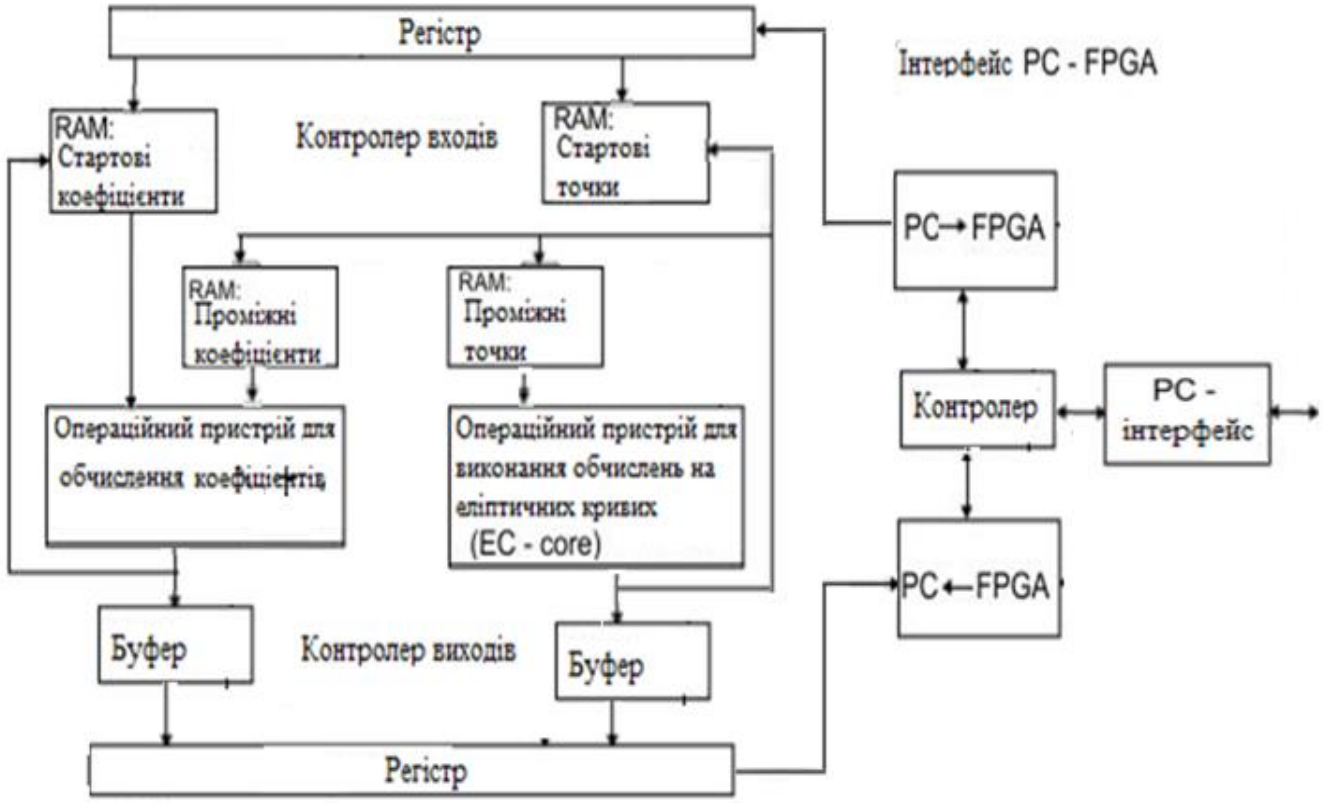

Рисунок 2.5 – Архітектура паралельної апаратної системи

В цьому компоненті виконання найзатратнішої за часом та кількістю задіяних модулівоперації, якою є множення, ґрунтується на алгоритмі КОА (Karatsuba-Ofman Algorithm). Останній полягає у зменшенні кількості множень, необхідних для визначення добутку двох чисел великої розрядності, за рахунок введення додаткових обчислень суми та різниці при розкладі чисел до такого вигляду:

$$x = x_1 P^m + x_2, \qquad y = y_1 P^m + y_2, \tag{2.13}$$

де x_2 і y_2 менші, ніж P^m.

Для реалізації алгоритму в програмних матрицях FPGA запропоновано рекурсивний поділ чисел до відповідного розміру, а згодом використання паралельних спеціально оптимізованих помножувачів розміром, адаптованим до завдання, яке в даний час виконується. Подробиці цього підходу разом з моделлю обчислення оберненості числа, необхідного для додавання точок в афінних координатах, можна знайти в [43, 96]. Розрахунок оберненості заснований на

алгоритмі Евкліда і є дуже затратною в часі та кількістю складових компонентів операцією. Наприклад, зайнятість комірок програмованої матриці, необхідних для побудови компонента, що обчислює оберненість, майже в сім разів вища, ніж кількість, необхідна для створення мультиплікатора для чисел такого ж розміру.

Показано, в поєднанні з порівнянням відносно реалізованої таким же чином програмної системи, базованої на бібліотеках NTL для процесора Xeon 3,2 ГГц [43], що збудований аналогічно процесор, який виконує поодиноку стежку випадкового блукання за ро-методом Полларда, знайде дискретний логарифм протягом часу, наведеного в таблиці 2.2.

Таблиця 2.2

Час, передбачений для знаходження логарифму на різних платформах

Крива/платформа	Програмна платформа	Апаратна платформа
$GF(2^{79})$	1000 років	3 години
$GF(2^{163})$	$540 \cdot 10^{12}$ років	$700 \cdot 10^{6}$ років

2.3.3 Операційні пристрої до реалізації стежки блукання ро-методу Полларда для кривих $GF(p)$.

Ґрунтуючись на результатах проведених тестів і досліджень стійкості до атак з використанням ро-методу Полларда, обґрунтовано, що ефективним підходом є застосування процесора, який реалізує обчислення на еліптичній кривій над скінченним полем вищих порядків $GF(p)$ [28, 37, 66, 84]. Подібно, як і в описі попереднього пункту, побудований операційний пристрій може стати частиною більшої, криптографічно захищеної інформаційної системи, яка реалізовує паралельний ро-метод Полларда, розглянутий в пункті 2.3.1.

Рисунок 2.6 ілюструє структуру операційного пристрою, за допомогою якого реалізується частину ро-методу Полларда, що полягає на циклічному додаванню точок на еліптичній кривій.

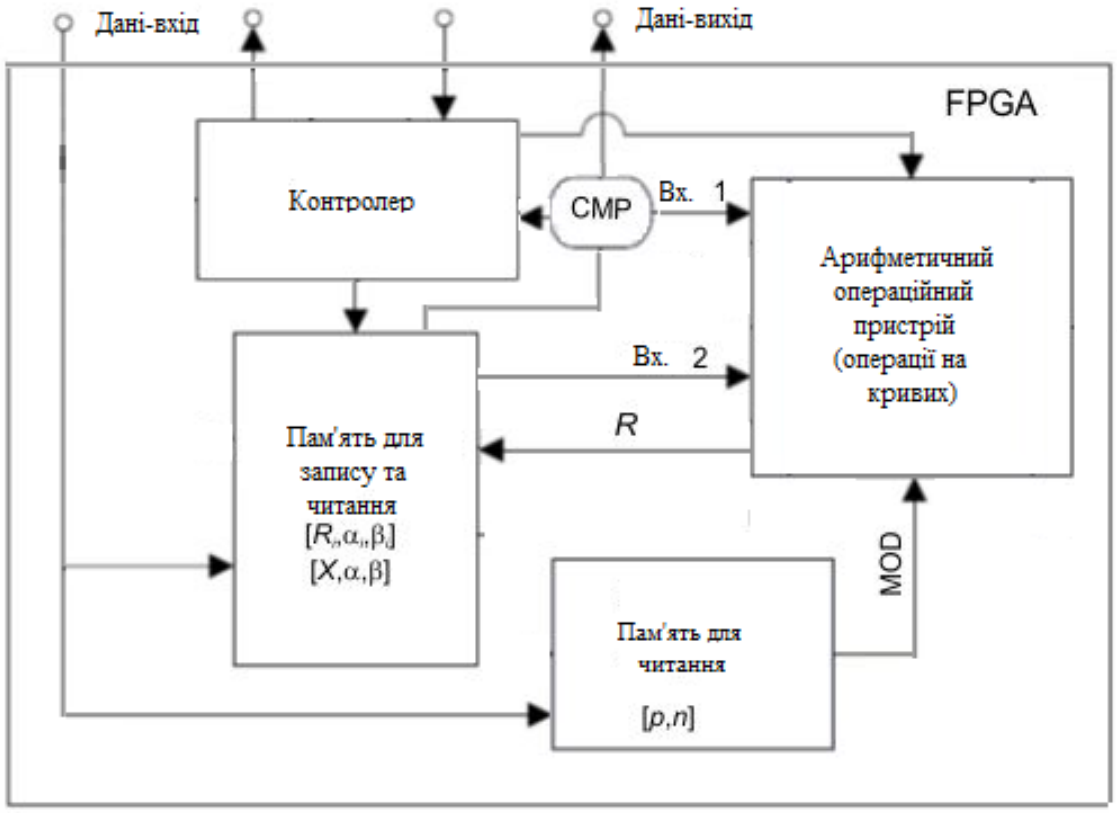

Рисунок 2.6 – Структура операційного пристрою для реалізації додавання точок на еліптичній кривій ро-методу Полларда

В структурі цього операційного пристрою відзначено пам'ять, в якій зберігаються координати точок, що потрібні для здійснення ро-методу Полларда, стартові коефіцієнти і порядок групи. Пропозиція такого рішення міститься також в інших працях та вважається для досягнення цієї мети оптимальною. З точки зору швидкодії виконання основних операцій на еліптичній кривій становить інтерес операційний пристрій для виконання арифметичних операцій. Операції додавання і віднімання здійснюються модулярно [78, 87]. Натомість операції множення та інверсії реалізуються за допомогою методу Монтґомері з модифікацією Каліскі (B. S. Kaliski Jr.) [52]. Для імплементації застосовано програмовані матриці FPGA компанії Xilinx. Операції додавання та віднімання здійснено з використанням Xilinx IP cores. Для виконання операції множення методом Монтґомері збудовано операційний пристрій на підставі одного (k + 1)-бітного суматора/віднімача, двох (k + 1)-бітних регістрів переносу та одного (k + 1)-бітного мультиплексора.

Обчислення оберненості методом Калісі вимагає використання трьох (k + 1)-бітних суматорів/віднімачів, чотирьох (k + 1)-бітних регістрів перенесення і шести (k + 1)-бітних мультиплексорів [65].

Функціонування засобу XC3S1000 типу FPGA, в якому застосовано вище зазначені методи, дозволило отримати результати, які представляють кількість додавань точок за одну секунду для кривих з параметрами різної довжини та зібрані в таблиці 2.3.

Таблиця 2.3

Швидкодія додавання для поодинокого операційного пристрою на підставі програмованої матриці FPGA

Крива *GF(p)*	160	128	96	80	64
Кількість додавань/с	46 800	80057	90 000	111 900	148 200

Наступна таблиця 2.4 містить передбачуваний час, який потрібний для знаходження дискретного логарифму для різних кривих за допомогою різних операційних пристроїв, які реалізують ро-метод Полларда. Результати, підготовлені в таблиці 2.4 на основі даних, що містяться в [66], відносяться до одного операційного пристрою, в якому застосовано рішення, описане в цьому пункті та яке стосується додавання точок і реалізації ро-методу Полларда.

Таблиця 2.4

Прогнозний час для обчислення дискретного логарифму окремим операційним пристроєм

Крива *GF*(p) / Засіб	Pentium M	XC3S1000
64	5,14 год	78,1 хв.
80	70,5 днів	21,5 днів
96	55,1 років	20,4 років
128	$4{,}86 \cdot 10^6$ років	$2{,}55 \cdot 10^6$ років
160	$3{,}78 \cdot 10^{11}$ років	$2 \cdot 10^{11}$ років

2.3.4 Кластер програмованих вентильних матриць.

Узагальнюючи теоретичний розгляд, можна передбачати, що перспективним є підхід, який ґрунтується на застосуванні кластера програмованих матриць FPGA до обчислень на еліптичних кривих, де основне завдання полягає у знаходженні дискретного логарифма (рисунок 2.7) [31, 34, 71, 110].

Кластер, схему якого ілюструє рисунок 2.7, побудований на програмованих матрицях FPGA типу Copacobana. Кластер складається з контролера USB, який дозволяє підключити до комп'ютера і контролера FPGA, що взаємодіє з окремими компонентами, максимально 120 програмованих матриць FPGA, зібраних в модулі. Модулів може бути максимум 20 по 5 програмованих матриць в кожному модулі. Застосовано відповідні моделі, методи та алгоритми з метою обчислення дискретного логарифма на еліптичних кривих вищих порядків *GF*(p). Можна також здійснити розрахунки на еліптичних кривих над полем другого порядку $GF(2^m)$ [110].

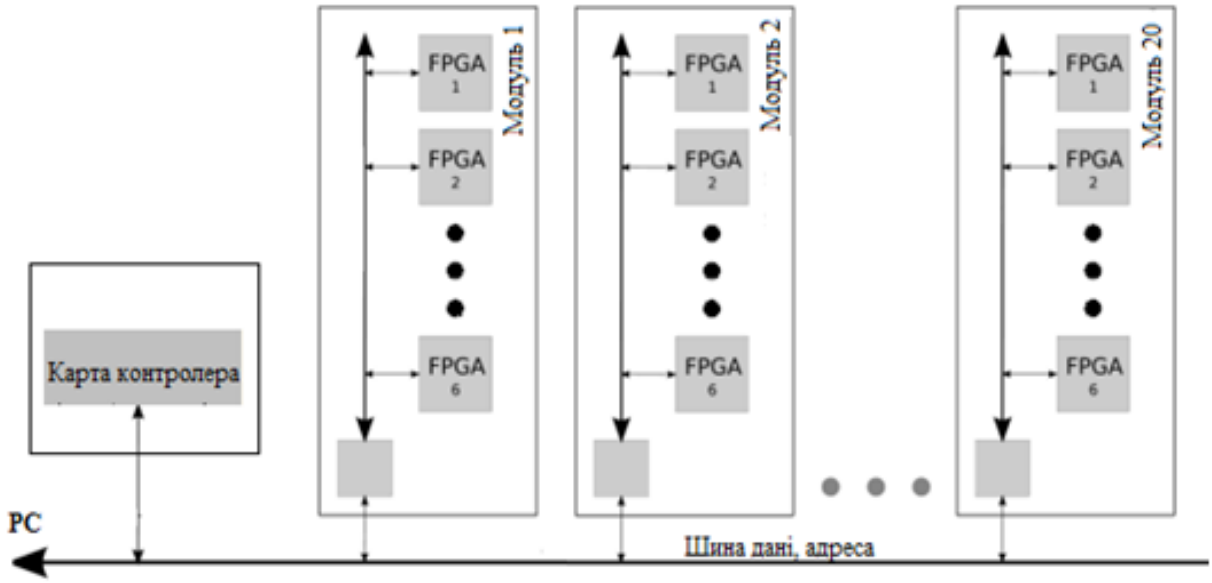

Рисунок 2.7 Схема побудови кластера програмованих матриць FPGA

На увагу заслуговує також порівняння результатів функціонування кластера, побудованого на комп'ютерах класу ПК вартістю 10 тис. ам. доларів, та кластера програмованих матриць FPGA тієї ж вартості. Отримані результати показано на нижче наведеному графіку (рисунок 2.8) [66].

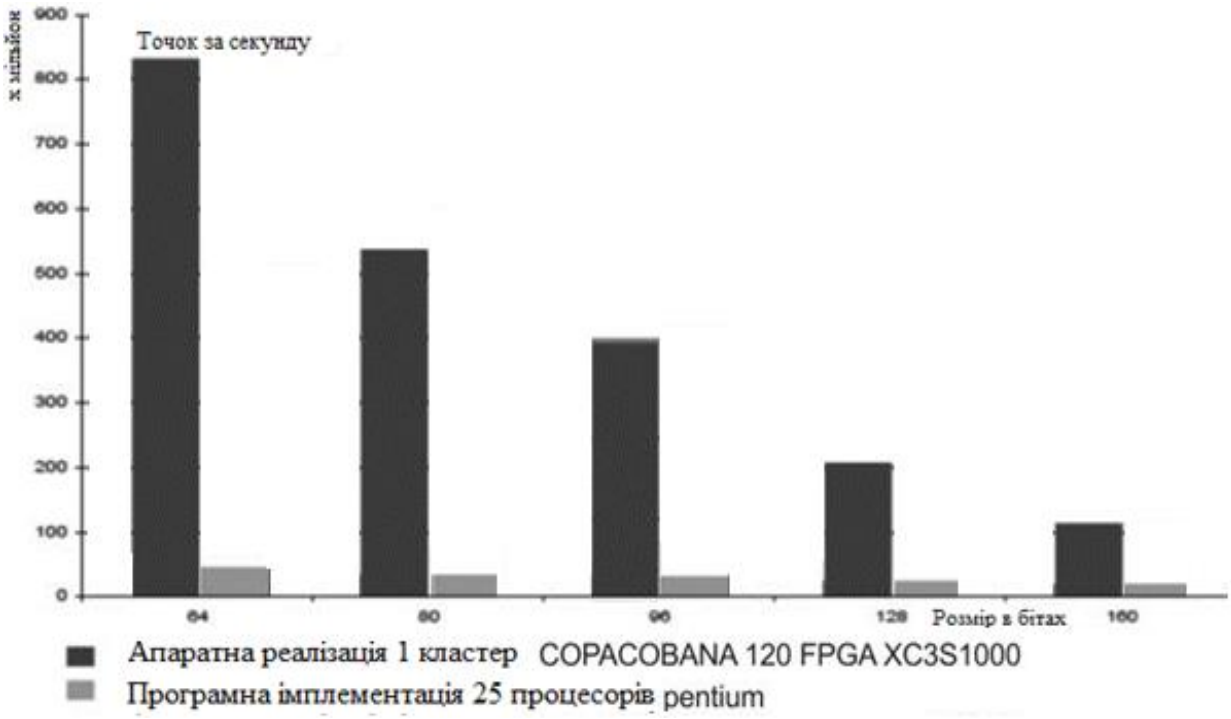

Рисунок 2.8 – Кількість додавань точок за секунду для кластерів, вартістю 10 тис. ам. доларів

В імплементаціях додавання точок застосовано вищеописаний метод Монтґомері. Для здійснення операцій ADD/SUB/MUL використано арифметичні компоненти, що містяться в програмованих матрицях FPGA, – так звані блоки DSP.

2.4. Засади прискорення обчислень на еліптичних кривих для підвищення живучості інформаційно-управляючих систем

2.4.1 Методи прискорення розрахунків на еліптичних кривих.

Методи базуються на підході, що відноситься до створення засобів, які допомагають здійснити криптоаналіз шифрів на основі еліптичних кривих в репрограмувальних структурах [51, 69, 95]. Існує можливість побудувати структурну одиницю – операційний пристрій, який реалізовує ро-метод Полларда для еліптичних кривих над полем другого порядку $GF(2^m)$. Застосування відповідно підібраних представлень, а також подання точок на еліптичній кривій у відповідних координатах призвело до того, що отримано кращі результати, ніж ті, які наведені в більш ранніх працях.

В подальшій частині даної дисертації представлено операційний пристрій, який реалізує ро-метод Полларда, подібно як компонент, розглянутий в праці [66]. Однак те, що, безумовно, відрізняє ці два засоби – це спосіб виконання обчислень, реалізований в арифметичному пристрої. Для того, щоб прискорити виконання операції додавання точок, застосовано змішане представлення. Останнє вимагає, зокрема для апаратної реалізації операції додавання двох точок, виконання лише одинадцять множень і повністю ігнорує необхідність обчислення оберненості. У пропонованому рішенні для представлення елементів поля використано нормальні базиси. Множення елементів основане на матриці множення, завдяки чому побудовано паралельний перемножувач, який дозволяє отримати результат множення за один такт [25, 84, 95]. Результат цієї імплементації – це запроектування операційного пристрою в програмованій матриці FPGA типу EP2C35F672C6 Cyclone II. Створений операційний пристрій реалізує ро-метод Полларда для кривої ECC2-89. Результати, які отримано авторами праці [95], зібрано в таблиці 2.5.

Таблиця 2.5

Основні результати дії операційного пристрою для додавання точок

Частота праці, МГц	137,25
Кількість додавань / с	12480000

В результаті проведених досліджень виявлено, що прогнозний час, потрібний для знаходження дискретного логарифму на основі використання ро-методу Полларда для кривої ECC2-89, становить приблизно 20,5 днів, якщо застосовано лише один операційний пристрій для додавання точок [95, 109].

2.4.2 Технологія обчислень на основі системи HBTNS.

Результати теоретичного аналізу вказують, що прискорення виконання математичних операцій на еліптичних кривих з метою підвищення живучості інформаційно-управляючих систем можна досягнути за допомогою моделей обчислень, імплементованих в систему HBTNS (Hybrid Binary-Ternary Number System) [55]. На базі розробленої системи HBTNS можна застосувати три методи, які дають змогу підвищити швидкодію здійснення основних операцій на еліптичних кривих [29]. Їх суть полягає у скороченні запису числа по відношенню до двійкового запису числа за рахунок мішаного запису в подвійному теоретико-числовому базисі, що ілюструє рисунок 2.9 [29].

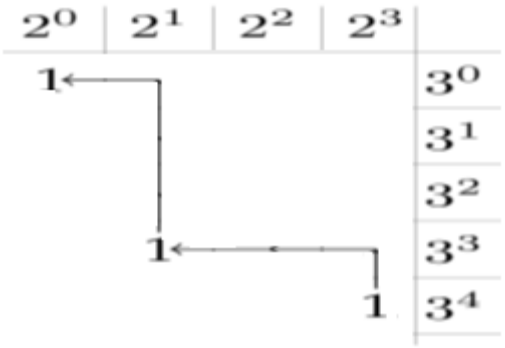

Рисунок 2.9 – Приклад запису числа 703 в подвійному теоретико-числовому базисі

Слід зазначити, що вищезгадані три методи, які ґрунтуються на системі HBTNS та можуть бути використані в обчисленнях на еліптичних кривих,

вимагають однак дослідження швидкості виконання арифметичних операцій для конкретних кривих.

2.5. Технологія обчислень для прискореного здійснення основних операцій на еліптичних кривих GF(p) і GF(2^m) у спеціалізованих компонентах FPGA і GPU

Результати проведеного теоретичного дослідження свідчать про те, що перспективною вбачається реалізація технології обчислень для прискореного виконання основних операцій на еліптичних кривих над полем вищих порядків *GF*(p) на базі вбудованих модулів VIRTEX [36]. При цьому точки на еліптичних кривих ще перед початком виконання операції додавання записано у проективних координатах, що дозволило уникнути необхідності обчислення оберненості. Операція обчислень на числах великої розрядності ґрунтується на модулярних додаванні, відніманні та множенні. Архітектура операційного пристрою для додавання точок базується на двох суматорах/віднімачах і двох помножувачах (рисунок 2.10) [73].

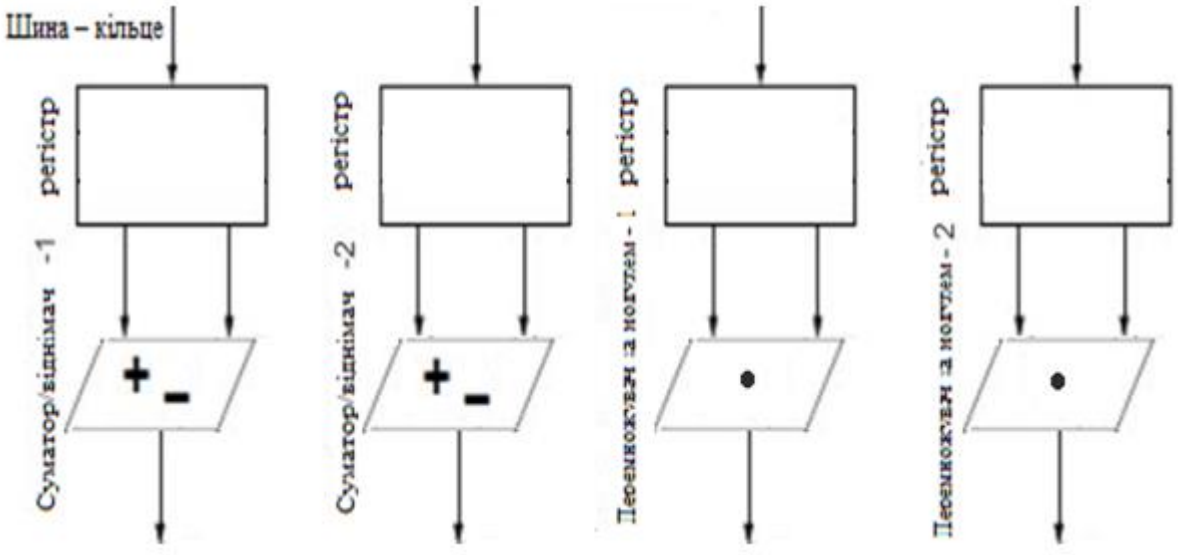

Рисунок 2.10 – Схема операційного пристрою для додавання точок на еліптичних кривих

Операцію множення методом Монтґомері здійснено за допомогою алгоритму Radix-4. В скороченому вигляді множення двох k-бітних чисел X та Y відповідно до цього алгоритму можна подати так [36]:

$$A = X \cdot Y = \{2^{k-1}x_{k-1} + 2^{k-2}x_{k-2} + x_{k-3,k-4}\} \times \{2^{k-1}x_{k-1} + 2^{k-2}y_{k-2} + y_{k-3,k-4}\}. \quad (2.14)$$

Ґрунтуючись на [20, 114], графічне представлення операції множення можна проілюструвати рисунком (рисунок 2.11) [36].

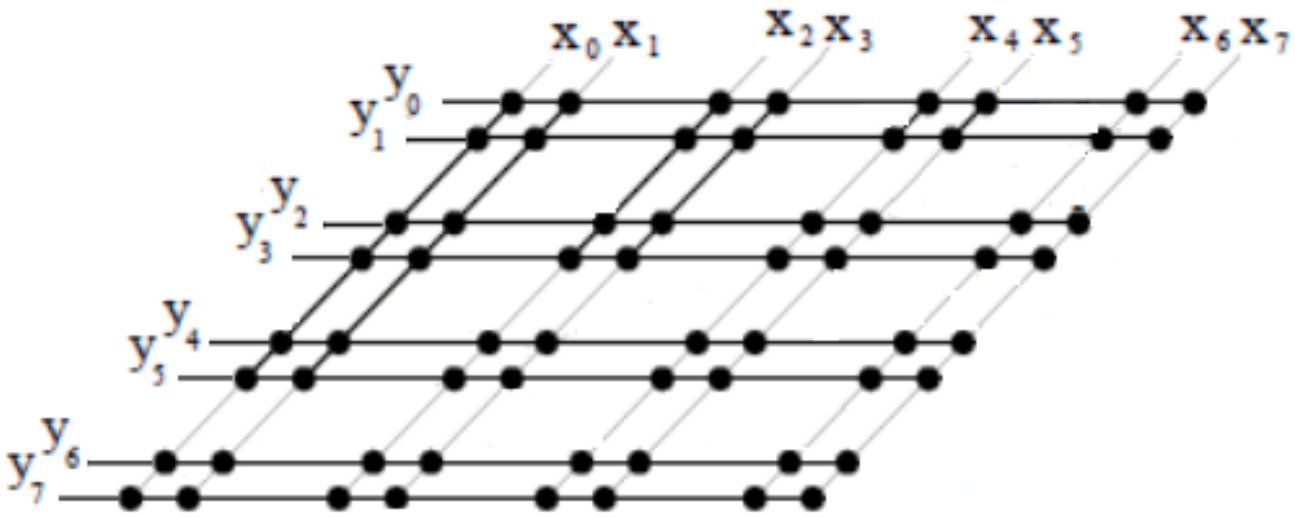

Рисунок 2.11 – Схема множення на основі алгоритму Radix-4

Дослідження показали, що час, необхідний для виконання одного додавання, в залежності від довжини модуля еліптичної характеристики, приймає значення, наведені в таблиці 2.6 [36].

Таблиця 2.6

Час виконання операції додавання двох точок на кривій GF(p)

$GF(p)$	Час додавання двох точок, мкс
128	4,45
256	10,34

Доведено, що прискорення обчислень на ЕК може бути досягнуто шляхом подальшої імплементації відповідного методу для еліптичних кривих виду $GF(2^m)$ [50]. Додавання точок еліптичної кривої $GF(2^m)$ проведено в проективних координатах. Додавання двох елементів, що належать еліптичній кривій $GF(2^m)$, полягає на додаванні відповідних коефіцієнтів за модулем 2. З цією метою застосовано операцію XOR для відповідних коефіцієнтів. Беручи до уваги [60], додавання точок A і B, результат якого дорівнює C, можна подати моделлю, представленою такою формулою:

$$C = \sum_{i=0}^{m-1} a_i + b_i \bmod 2, \tag{2.15}$$

якщо

$$A = (a_{m-1}a_{m-2} \ldots a_1a_0) \tag{2.16}$$

і

$$B = (b_{m-1}b_{m-2}\ldots b_1b_0). \tag{2.17}$$

Передбачено множення двох елементів A і B за модулем третій елемент P, причому всі елементи представлено як і для випадку додавання. Множення виконується одночасно з обчисленням модуля згідно з методом, запропонованим в праці [98]. Схематично перемножувач за модулем наведено на рисунку 2.12 [50].

Результати імплементації вищевказаних методів дозволили досягти виконання операції додавання двох точок на еліптичній кривій $GF(2^{163})$ протягом часу 3,05 мкс.

Результати проведених досліджень доводять, що не повинна залишатися поза увагою науковців наявність ще однієї можливості реалізації методу паралельних обчислень, яка базується на використанні процесорів GPU (Graphics Processing Units) [48].

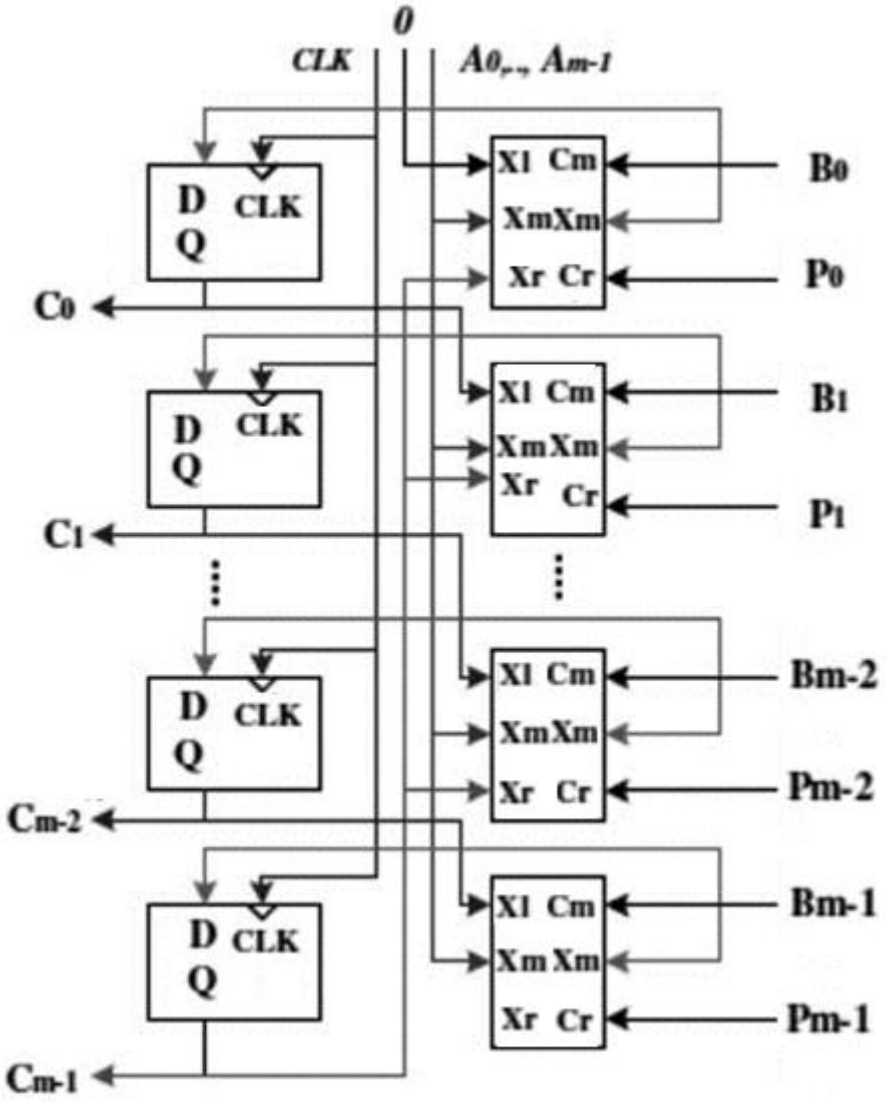

Рисунок 2.12 – Архітектура перемножувача по модулю для еліптичної кривої $GF(2^m)$

Показано, що алгоритм LSB можна застосувати із задовільними результатами також для виконання операції множення елементів в програмованих матрицях FPGA, що знайшло відображення в працях [14, 81, 100, 115] (рисунок 2.13).

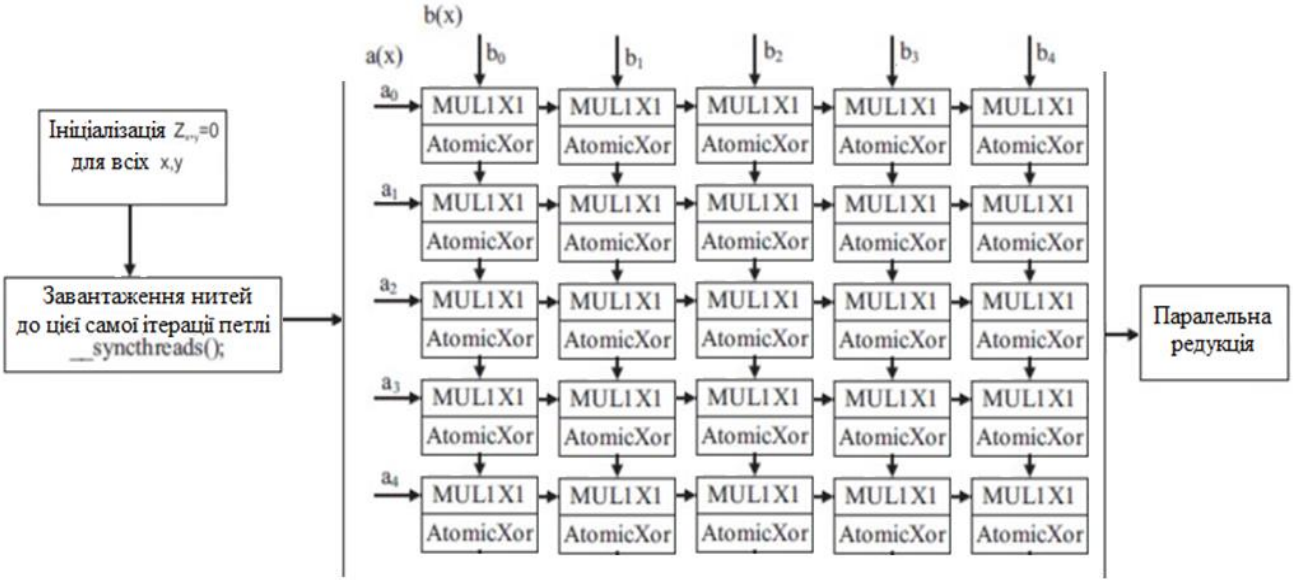

Рисунок 2.13 – Спосіб виконання операції множення елементів на еліптичній кривій $GF(2^m)$ із використанням алгоритму LSB

Про це свідчать також результати випробувань, проведених в [48], а також власні дослідження, порівнюючи швидкодію функціонування операційних пристроїв [105], побудованих на основі процесорів GPU та імплементацій, здійснених в середовищі програмованих матриць FPGA.

Висновки до розділу 2

1. Розроблено структурну модель підвищення живучості ІУСЕК з врахуванням дефектів зовнішніх впливів за ознаками ймовірності та детермінованості і досліджено алгоритми, які використовуються в моделях виконання основних операцій на еліптичних кривих, що дало змогу оцінити їх продуктивність.

2. Розглянуто та порівняно різні імплементації ро-методу Полларда розв’язування дискретного логарифму, звертаючи увагу в основному на швидкодію знаходження розв’язку.

3. Здійснено вибір апаратно-програмної системи для реалізації ро-методу Полларда на еліптичних кривих, базованих на скінченному полі характеристики вищої від 3, завдяки чому доведено оптимальність такої системи із досягненням найбільшої ефективності виконуваних операцій.

4. Обґрунтовано вибір та доведено продуктивність методу модулярного множення, що ґрунтується на теоретико-числовому базисі Радемахера-Крестенсона, та паралельного додавання, що дозволило отримати базу для створення моделей та алгоритмів виконання арифметичних операцій знаходження суми та кратності точок на кривій.

РОЗДІЛ 3
МОДЕЛІ ТА ЗАСОБИ ОБЧИСЛЮВАЛЬНИХ ПЛАТФОРМ НА ЕЛІПТИЧНИХ КРИВИХ З ВИКОРИСТАННЯМ ТЕОРЕТИКО-ЧИСЛОВИХ БАЗИСІВ РАДЕМАХЕРА-КРЕСТЕНСОНА ІЗ АПАРАТНОЮ РЕАЛІЗАЦІЄЮ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ ВИЩОЇ ЖИВУЧОСТІ

В третьому розділі представлено моделі обчислювальних платформ, за допомогою яких виконуються основні операції на еліптичних кривих із застосуванням розроблених автором перемножувачів, що грунтуються на ТЧБ Радемахера-Крестенсона, і суматорів, базованих на паралельному сумуванню чисел великої розрядності, наприклад, довжини 100 і більше бітів. Побудовано і проаналізовано моделі та апаратні імплементації для виконання основних операцій на ЕК. Розроблено операційні пристрої для здійснення обчислень на еліптичних кривих. Реалізовано апаратний засіб, який виконує поодиноку стежку ро-методу Полларда, а також його паралельну версію. Проведено аналіз та виконано оцінювання особливостей функціонування та застосування побудованих засобів і удосконалених методів щодо живучості ІУСЕК, здійснюваної шляхом розв'язання дискретного логарифма. Показано придатність розроблених моделей та алгоритмів в шифруванні і розшифруванні.

3.1. Моделі виконання операцій на еліптичних кривих за допомогою теоретико-числових базисів Радемахера-Крестенсона і паралельного сумування чисел великої розрядності

3.1.1 Моделі та живучість.

Покладемо в основу, по аналогії з відмовостійкою багатопроцесорною ІУС [21, 97], її продуктивність P для оцінювання стану ІУСЕК, базованих на пристроях, що реалізують криптографічні операції на еліптичних кривих, та введемо наступні складові:

p_{wwf} – імовірність безвідмовної роботи ІУСЕК;

f_s – функція живучості, тобто мінімальна підмножина функцій ІУСЕК з найвищим пріоритетом, виконання яких необхідно для того, щоб система не перейшла до небезпечного стану;

P_s – продуктивність ІУСЕК, необхідна для виконання функцій живучості та нижче за яку виникає аварія;

m – загальна кількість функцій живучості;

p_{ds} – ймовірність переходу ІУСЕК на часовому інтервалі t до небезпечного стану;

n – загальна кількість процесорних пристроїв в ІУСЕК;

n_{ECCD} – кількість пристроїв ECCD у системі, що виконують криптографічні операції на еліптичних кривих;

x – вектор стану ІУСЕК;

X_{ds} – множина, яка відповідає небезпечним станам ІУСЕК;

x_i та x_j – компоненти вектора x, які відповідають стану i-гопроцесорного пристрою та j-го ECCD ($x_i = x_j = 0$ – для відмови; $x_i = x_j = 1$– для працездатності);

P_i та P_j – продуктивність i-го процесорного пристрою та j-го ECCD;

P_x – продуктивність ІУСЕК в стані, що відповідає вектору x.

Беручи до уваги наведені в [97] методи, можна отримати модель живучості ІУСЕК, представлену формулами:

$$P_x = \sum_{i=1}^{n-n_{ECCD}} \alpha_i \; P_i + \sum_{j=1}^{n_{ECCD}} \alpha_j \; P_j \;, \tag{3.1}$$

$$p_{ds}(t) = \sum_{x \in X_{ds}} p_x(t)\Big|_{\forall x \in X_{ds}} \;, \tag{3.2}$$

де

$$p_x(t)\Big|_{\forall x \in X_{ds}} = \prod_{i=1}^{n} p_i^{x_i}(t)\big(1 - p_i(t)\big)^{1-x_i} - \prod_{j=1}^{n-n_{ECCD}} p_j^{x_j}(t)\big(1 - p_j(t)\big)^{1-x_j} \;, \tag{3.3}$$

причому $p_{wwf} = p\big(P_x \geq P_s\big)$, $P_s = \sum_{f=1}^{m} P_{s.f}$.

завдяки чому можна обчислити ймовірність переходу ІУСЕК в часовому інтервалі *t* до небезпечного стану, викликаного зниженням її продуктивності внаслідок відмов ECCD.

Основна особливість моделі полягає в заміні операції модулярного множення цілих чисел великої розрядності додаванням за модулем, яке проводиться на основі використання теоретико-числового базису Крестенсона, а також застосуванням паралельного додавання чисел, що дає змогу прискорити виконання операції на відміну від традиційного підходу. Паралельне сумування полягає на поділі цих чисел великої розрядності на слова, розмір яких дозволить безпосередньо виконати операцію додавання за допомогою вбудованих в процесори суматорів з використанням стандартних типів даних. Зазвичай в математичних операціях на еліптичних кривих виконується додавання точок або їх подвоєння. В такого типу операціях застосовуються дії над числами, такі як сумування по модулю та модулярне множення чисел великої розрядності.

3.1.2 Узагальнена модель паралельного суматора багаторозрядних чисел за модулем.

Модель суматора ґрунтується на теоретичних основах, що містяться в [19, 86, 118], з деякими удосконаленими модифікаціями, які дають змогу додавання багатьох чисел. Такий підхід необхідний для створення моделі перемножувача, базованого на теоретико-числовому базисі Крестенсона. Передбачається, що на суматор подаються числа у вигляді двійкових послідовностей. Основною частиною функціонування цієї моделі є поділ цілих чисел великої розрядності на слова потрібної довжини *m* в базисі $\delta = p^m$, зокрема у цьому випадку $p = 2$. Відповідно до формули $\boldsymbol{X = x_n \delta^n + x_{n-1} \delta^{n-1} + \cdots + x_1 \delta^1 + x_0}$ довжина слова *m*, на яку будуть поділені числа, які підлягають додаванню, становить 23 біти. Наприклад, число *X* розміром 92 біти поділено на чотири слова довжиною 23 біти $\boldsymbol{x_3, x_2, x_1, x_0}$, так що кожне слово є типу integer. Сумуючи два слова $\boldsymbol{X + Y = Z}$, додаються відповідні слова $x_i + y_i$, подані у вигляді чисел типу integer, причому на позиції *i* знаходиться

$(x_i + y_i) \bmod \delta$, а до старшого слова z_{i+1} переноситься $(x_i + y_i) \operatorname{div} \delta$ згідно з алгоритмом 2.1. В цьому випадку використано факт, що в основі поділу лежить число 2. Запис чисел у вигляді двійкових послідовностей дозволяє виконувати операції *div* і *mod* тривіальним способом. Недорогим чином можна також здійснювати конвертацію даних між типом integer і записом у вигляді вектора.

Одночасно з додаванням відбувається обчислення модуля з числа, що представляє суму, тобто *Z mod n*. В розглянутих розв'язках числа *X* і *Y* є меншими, ніж модулі *n*, так що $X+Y<2n$. Таким чином, обчислення модуля числа Z, який є сумою чисел*X* і *Y*, зводиться до перевірки умови $Z>n$; *я*кщо умова виконується, то вистачить здійснити операцію віднімання $Z–n$. На практиці обчислення модуля виконується паралельно зі знаходженням кожного слова $z_{i.}$, що схематично показано на рисунку 3.1.

Віднімання чисел також виконується паралельним способом з попереднім поділом чисел на слова довжиною 23 біти відповідно до алгоритму 2.2.

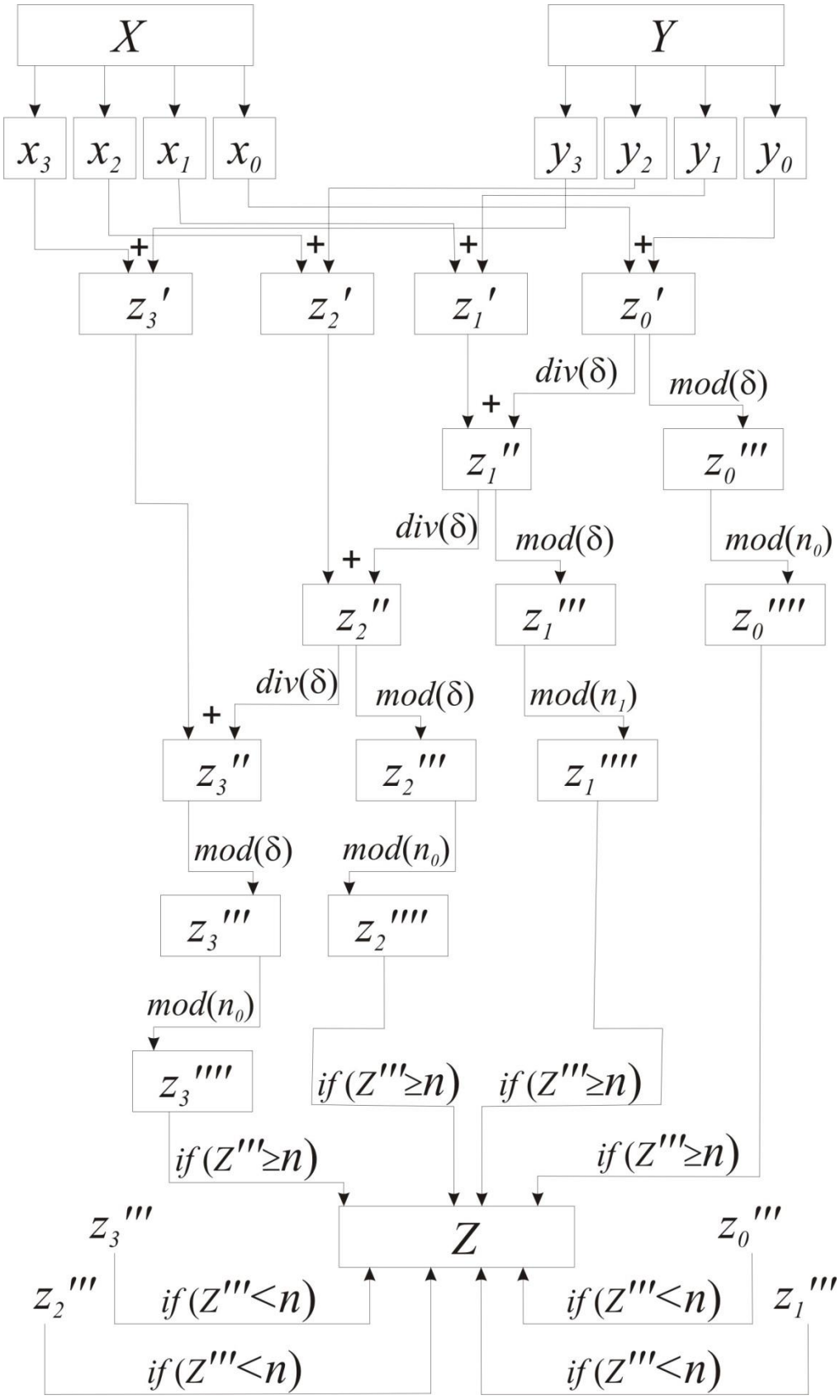

Рисунок 3.1 – Алгоритм паралельного додавання по модулю

3.1.3 Модель паралельного суматора багаторозрядних чисел за модулем в апаратних компонентах.

Модель апаратної реалізації паралельного додавання по модулю в програмованих матрицях FPGA передбачає два входи, які дозволяють завантажити сумовані числа, та один вихід. Всі обчислення проводяться по відношенню до одного і того ж модуля, так що для модуля не передбачено окремого входу, він записаний сталою в процесі перетворень.

Для чисел довжиною 92 біти компонент gfmkadd описує модель операційного пристрою додавання за модулем:

```
entity gfmkadd is port(
a, b : in std_logic_vector(91 downto 0);
clk, rst, next_ : in std_logic;
h: out std_logic_vector(91 downto 0)
);
end gfmkadd;
```

Нижче наведено основні елементи фізичної моделі, записані мовою опису апаратури, що реалізує паралельне додавання:

```
architecture gfkm_a1 of gfmkadd is

TYPE array4 IS ARRAY (3 downto 0) OF natural;
TYPE array5 IS ARRAY (4 downto 0) OF natural;
:
:
constant modul : array5 :=(0,0,0,0,0); -- moduł n
:
:
begin
:
kesm(0)<=to_integer(unsigned(a(22 downto 0)))+ \\
         to_integer(unsigned(b(22 downto 0)));-sumowanie słowa 0
```

Аналогічним способом додаютьсяінші слова:

```
:
.
```

Обчислення $(x_i + y_i) \bmod \delta$:

```
ke_add(0)<=to_integer(unsigned(std_logic_vector(to_unsigned(
           kesm(0), 32))(22 downto 0)));
```

Обчислення $(x_i + y_i)\text{div}\,\delta$:

```
aa<=kesm(1)+to_integer(unsigned(std_logic_vector(to_unsigned(
         kesm(0), 32))(30 downto 23)));
     :
     .
```

Обчислення модуляn з числа Z для словамодуляn_0 та числаz_0:

```
if ke_add(0)< modul(0)then
ke_min (0)<= 8388608+ke_add(0)-modul(0);
ke_mod(1)<=to_integer(unsigned(std_logic_vector(to_unsigned(aa,32))(
22 downto 0)))-1;
     else
ke_min(0)<= ke_add(0)-modul(0);
ke_mod(1)<=to_integer(unsigned(std_logic_vector(to_unsigned(aa,32))(
22 downto 0)));
      end if;
     :
     .
```

Остаточний результат отримується на підставі часткового порівняння окремих слів. Якщо число Z є більшим, ніж модуль, то видається результат, який зменшений на модуль n, в іншому випадку одержується Z.

Модель апаратного додавання дає змогу отримати суму за модулем двох чисел, поділених на чотири слова, в 7-ох циклах, що відповідають тактовому генератору програмованої матриці FPGA.

3.1.4 Модель паралельного віднімача багаторозрядних чисел за модулем.

Модель віднімача ґрунтується на засадах, що містяться в [86]. Ідея моделі полягає в тому, щоб отримати числа у вигляді двійкових послідовностей. Істотною частиною даної моделі є поділ цілих чисел великої розрядності на слова бажаної довжини, так само, як і у випадку суматора, який було розглянуто у попередньому пункті. Подібно, як і у випадку моделі суматора, паралельно з відніманням

відбувається обчислення встановленого модуля з різниці. Фізична модель, що забезпечує виконання алгоритму паралельного віднімання, базується на програмованій матриці FPGA. Розроблена апаратна модель паралельного віднімання за модулем має два входи, які дозволяють завантажувати числа, що віднімаються, та один вихід. Всі обчислення проводяться по відношенню до одного і того ж модуля, так що модуль не подається на вхід. Цей модуль зберігається у вигляді константи в процесі проектування модуля віднімача. Для чисел довжиною 92 біти компонент gfmksub описує операційний пристрій віднімання за модулем:

```
entity gfmksub is port(
a, b : in std_logic_vector(91 downto 0);
clk, rst, next_ : in std_logic;
h: out std_logic_vector(91 downto 0)
);
end gfmksub;
```

Особливості щодо реалізації віднімання є подібні, як це вже обговорювалося для випадку суматора та операцій за модулем, тому вони не будуть розглядатися в цьому пункті. Це дозволяє застосувати еліптичні криві для забезпечення заданих показників живучості ІУСЕК.

3.1.5 Модель перемножувача цілих чисел великої розрядності за модулем на основі теоретико-числових базисів Радемахера-Крестенсона.

Модель перемножувача, базована на системі залишкових класів Крестенсона, дає змогу виконувати множення за модулем натуральних чисел великої розрядності без необхідності здійснення множення традиційним способом. Використання системи залишкових класів дозволяє записати результат множення у матричному вигляді, а знаходження добутку числа полягає на додаванні відповідних елементів створеної матриці.

Теоретичні основи моделі перемножувача представлено в розділі 2 і в [5, 38, 117]. Розглянемо два числа X та Y і модуль n: $Z = X*Y \bmod n$.

Модель передбачає подання чисел X і Y у вигляді двійкових послідовностей

$$X = x_{r-1}2^{r-1} + x_{r-2}2^{r-2} + x_i 2^i + \ldots + x_1 2^1 + x_0 2^0,$$

$$Y = y_{r-1}2^{r-1} + y_{r-2}2^{r-2} + y_j 2^j + \ldots + y_1 2^1 + y_0 2^0.$$

Для визначення результату їх множення побудовано матрицю, представлену в таблиці 3.1, де $m_{ij} = 2^{i+j} \bmod n$.

Таблиця 3.1

Матриця Крестенсона

…	…	…	…	…	2^{r-1}
…	$(2^{j+i}) \bmod n$	…	…	…	2^i
…	…	…	…	…	|
…	…	…	$(2^{1+1}) \bmod n$	…	2^1
…	…	…	…	…	2^0
2^{r-1}	2^j	….	2^1	2^0	

Добуток чисел або координат *X* та *Y* можна отримати з формули:

$$X \cdot Y \bmod n = \sum_{s,k=0}^{r-1} m_{sk} \bmod n, \qquad (3.4)$$

де $x_s = 1, y_k = 1,$ тобто m_{sk} знаходиться на перетині стовпця і рядка, для яких відповідні x_s та y_k y_j дорівнюють 1.

Числа, які записано в таблицю, є меншими, ніж заданий модуль *n.* Сума чисел рядка або стовпця, з прийнятими раніше припущеннями, є меншою від подвійного модуля, тому для обчислення за модулем достатньо порівняння та віднімання за модулем. Виходячи з цього, надалі автором створено модель, за допомогою якої проводяться обчислення із використанням стандартних типів даних, що обслуговуються безпосередньо даним процесором. Алгоритм обчислень виглядає наступним чином:

1. Генерування матриці Крестенсона згідно з таблицею 3.1 та її запис у тривимірному масиві (рисунок 3.2), де третій вимір залежить від кількості слів, на

які було поділено числа. Закладено виконання обчислень на 92-бітних числах. Застосовано тип чисел integer, тому прийняті числа поділено, як і раніше, на чотири слова по 23-біти. Таким чином побудований третій вимір масиву.

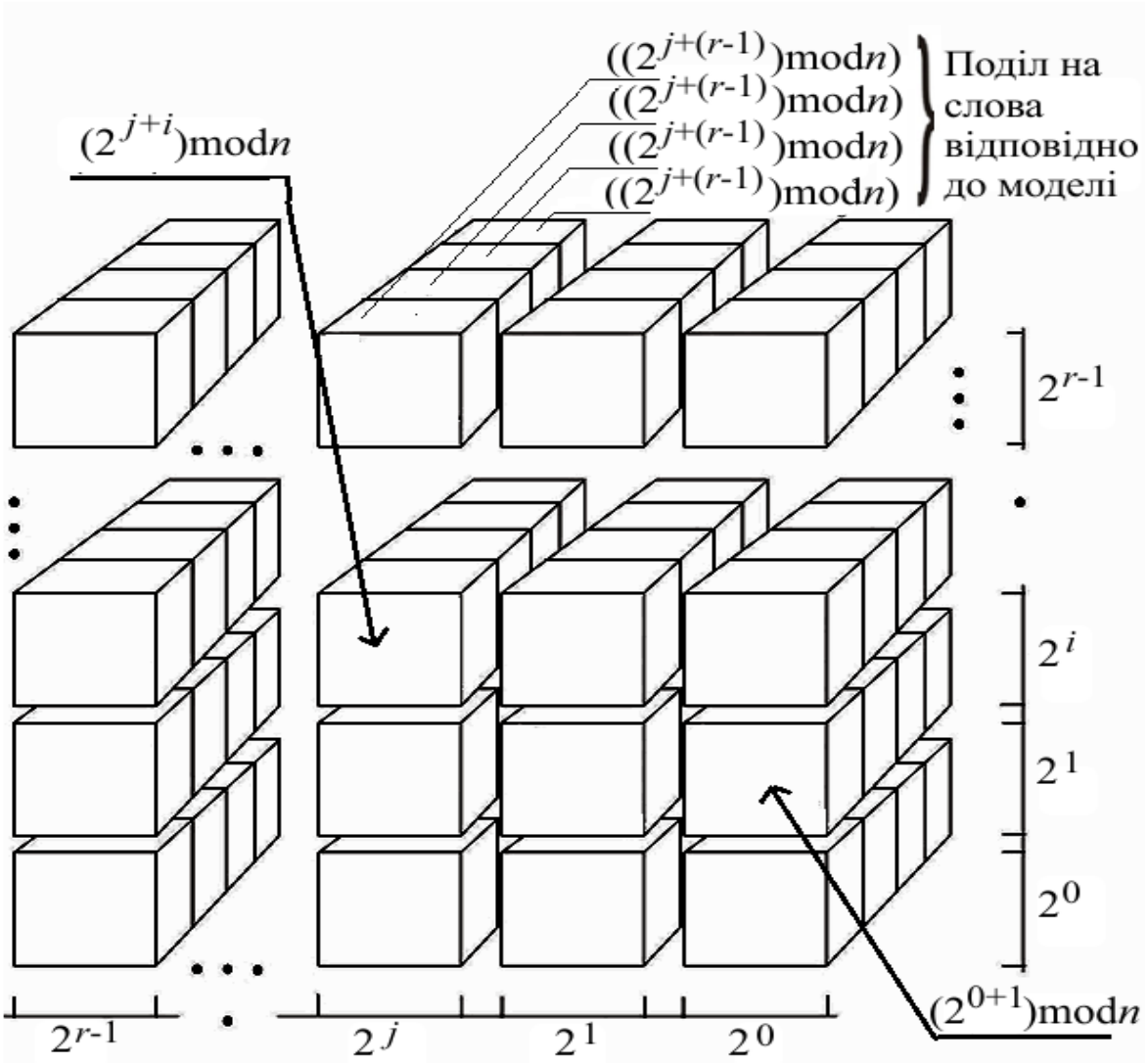

Рисунок 3.2 – Алгоритм запису матриці Крестенсона та обчислення для рядків моделі

2. Додавання за модулем n рядків матриці Крестенсона відповідно до виразу $\sum_{i;\,j=0}^{j=r-1} m_{sk} \bmod n$. Додавання рядків відбувається в паралельному режимі, тобто в тому самому часі здійснюється додавання кожного з i рядків відповідно до алгоритму, поданого на рисунку 3.2. Розглянуте додавання відповідає методу, описаному в

попередньому розділі. Єдина модифікація полягає у додаванні на першому кроці не двох слів, а *r–1* слів, що містяться в матриці, як показано на рисунку 3.3. Ця залежність визначає обмеження розміру одного слова до 23-ох бітів. Обчислення за модулем для кожного рядка здійснюється відповідно до рисунку 3.1.

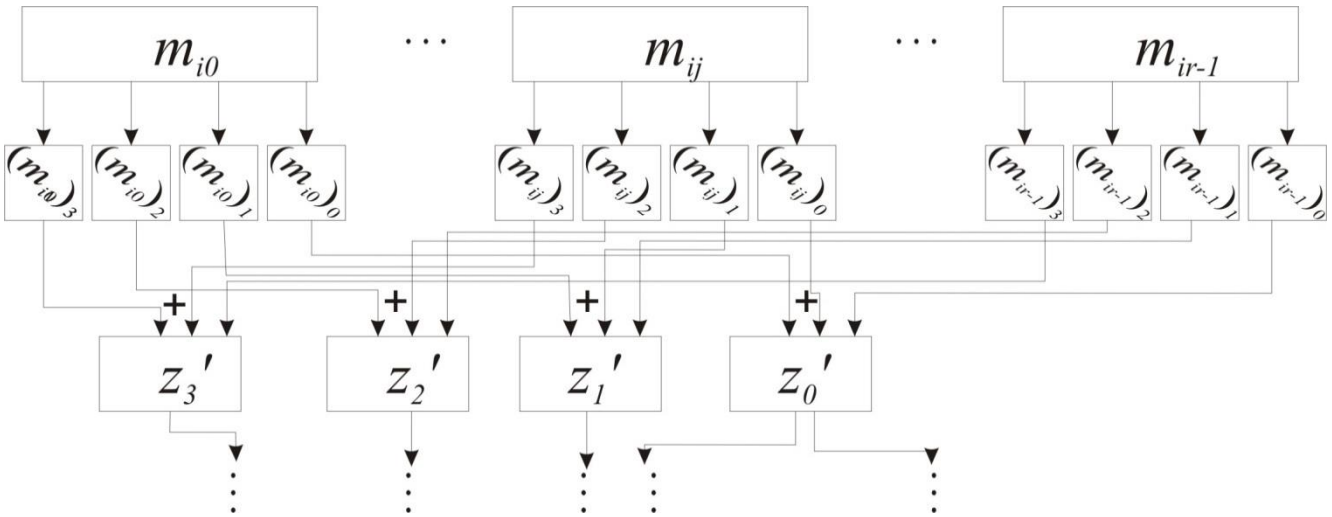

Рисунок 3.3 – Алгоритм паралельного додавання багатьох чисел

Рисунок 3.4 ілюструє концепцію додавання рядків матриці. Результатом цієї операції є вектор розміру *r* сум для кожного рядка, що показано на рисунку 3.4.

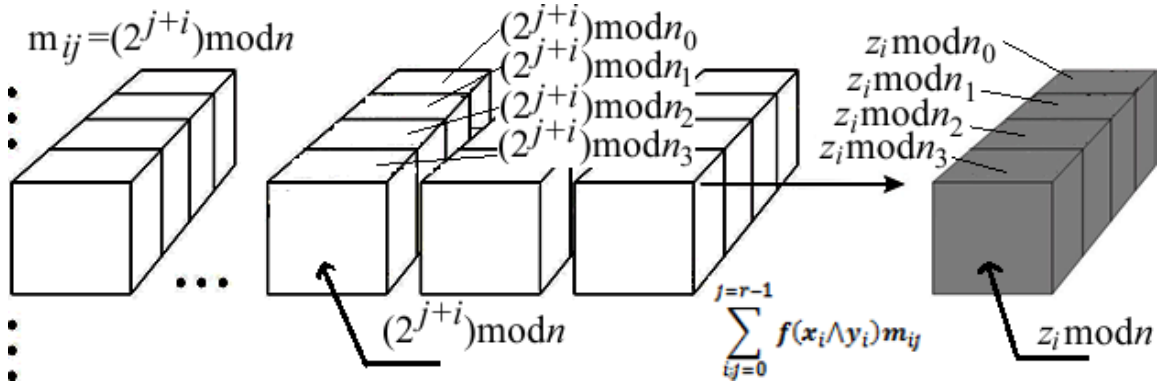

Рисунок 3.4 – Концепція додавання вмістимого рядків матриці Крестенсона

3. Додавання вектора, наведеного на рисунку 3.5, здійснюється в спосіб, представлений на рисунках 3.3 та 3.4. Єдиною відмінністю є те, що додаються всі елементи вектора, як це показано на рисунку 3.5. В результаті виконання цієї операції одержується добуток чисел *X* та *Y* за модулем *n*.

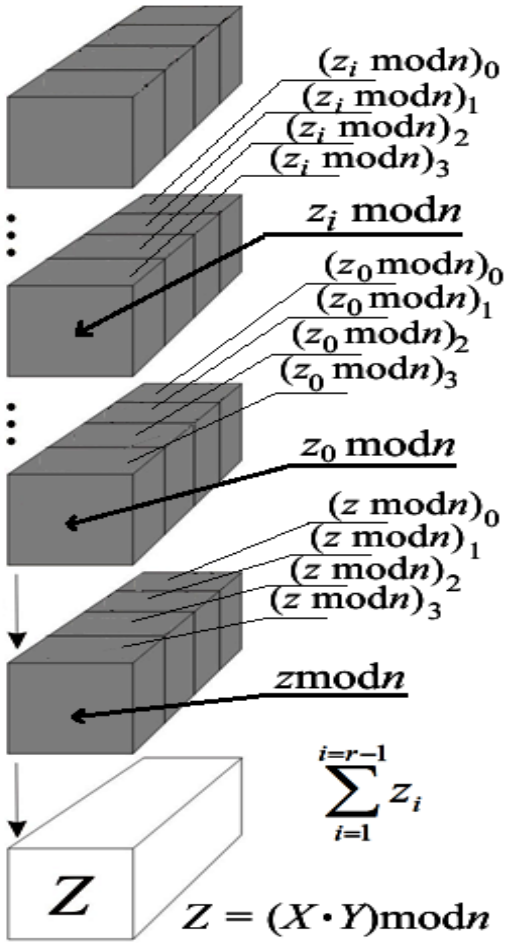

Рисунок 3.5 – Додавання елементів остаточного вектора матриці Крестенсона

Такий підхід дає змогу досягти мінімальних розмірів еліптичних кривих, які можуть бути застосовані для забезпечення необхідного рівня живучості ІУСЕК.

3.1.6 Модель перемножувача цілих чисел великої розрядності за модулем на основі теоретико-числових базисів Радемахера-Крестенсона в програмованих вентильних матрицях.

Надалі представлено модель апаратної реалізації обчислень добутку чисел за модулем з використанням теоретико-числових базисів Крестенсона в компонентах програмованих матриць FPGA. Обгрунтовано, що створення апаратної моделі слід розпочати з побудови матриці Крестенсона для чисел із заданою довжиною і заданого модуля. Матриця генерується та записується до пам’яті ROM. У апаратній моделі матриця зберігається в пам'яті у вигляді константи. Нехай *i* є номером рядка

матриці, а j – номером її стовпця. Слід відзначити закономірність, що якщо $\boldsymbol{i} + \boldsymbol{j} = \boldsymbol{i}' + \boldsymbol{j}'$, то $\boldsymbol{m_{ij}} = \boldsymbol{m_{i'j'}}$, де $\boldsymbol{m_{ij}}$ – елемент матриці з координатами ij. Таким чином, відповідно до вищенаведеного факту немає необхідності розміщення в пам'яті всіх r^2 елементів таблиці, а тільки $2r$. Числа в пам'яті зберігаються у вигляді двійкових векторів, для адресації використовується сума $i+j$.

Блок множення отримує числа у вигляді двійкових векторів. В тій же формі виводиться результат обчислень:

```
entity gfkm is port(
a, b : in std_logic_vector(91 downto 0);
clk, rst, next : in std_logic;
h : out std_logic_vector(91 downto 0));
end gfkm;
```

В апаратному перемножувачі можна виділити складові компоненти, показані на рисунку 3.6, і пам’ять ROM I, про яку йшлося раніше.

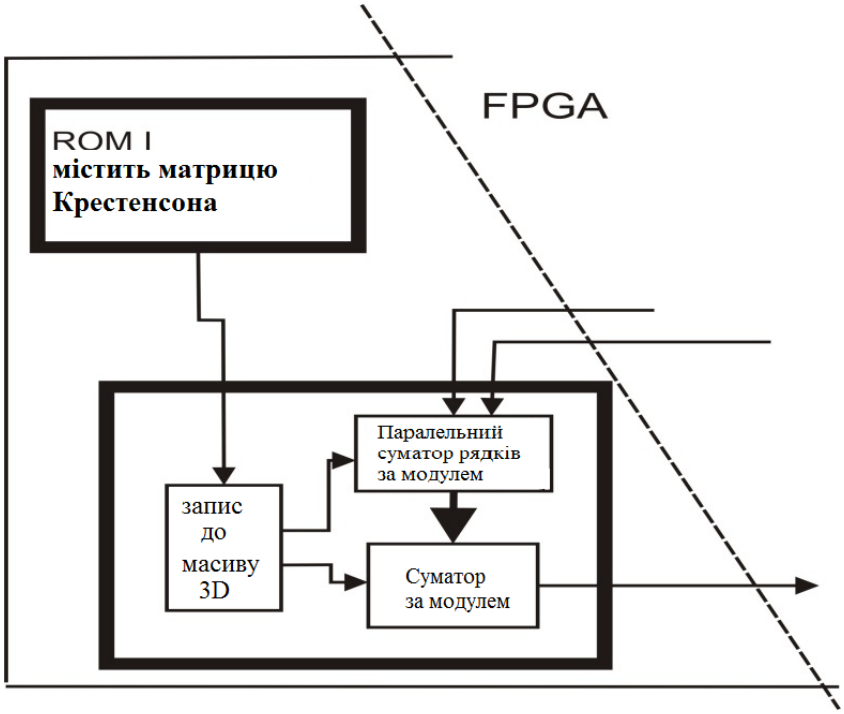

Рисунок 3.6 – Модель апаратного блоку множення, який ґрунтується на теоретико-числовому базисі Крестенсона

Зчитуванням даних з постійної пам'яті займається компонент, код якого представлено нижче. До нього передаються такі параметри: адреса, натуральне

число, що є сумою рядків і стовпців матриці, в якій дане значення в подальшому записується. Дані повертаються, проте у вигляді `STD_LOGIC_VECTOR`. Ці дані потім поміщаються всередині відповідної таблиці.

```
entity gfkmem is port(
       address   : in integer range 0 to 182;
       clock     : in std_logic ;
       q         : out std_logic_vector (91 downto 0));
end gfkmem;
```

Усередині апаратної моделі перемножувача дані, які завантажені з пам'яті згідно з вищенаведеним способом, піддаються обробці, тобто діляться на слова відповідного розміру. Поділ описано на початку цього розділу. Потім дані переконвертовуються в цілі числа і зберігаються у відповідній таблиці, як це показано на рисунку 3.2.

На підставі так підготовленої матриці Крестенсона відбувається додавання її окремих рядків. Всі рядки сумуються одночасно. Додаючи вміст окремих комірок в даному рядку, потрібно вибрати тільки поля з такими координатами *ij*, для яких значення вектора *a(i)* та *b(j)* приймає значення 1. Вхідні вектори *a* та *b* містять відповідно числа *X* і *Y*. Таким чином, якщо значення *a(i)* приймає значення 0, то не відбувається додавання по всьому рядку.

```
if a(0)='1' then
   kesm(0,3)<=matrixk(0)(aa(0),3)+matrixk(0)(aa(1),3)+ … ;
   kesm(0,2)<=matrixk(0)(aa(0),2)+matrixk(0)(aa(1),2)+ … ;
   kesm(0,1)<=matrixk(0)(aa(0),1)+matrixk(0)(aa(1),1)+ … ;
   kesm(0,0)<=matrixk(0)(aa(0),0)+matrixk(0)(aa(1),0)+ … ;
end if;
```

Спочатку перевіряється значення для стовпців і відповідно сформованого допоміжного вектора *aa(j),* за допомогою якого вибираютьсмя дані з таблиці. Потім відбувається додавання відповідно до алгоритмів, показаних на рисунках 3.3 і 3.4.

Наступним кроком є додавання всіх елементів вектора, який сформовано в результаті додавання всіх рядків матриці з одночасним обчисленням модуля, як і на попередньому етапі.

Перед тим як вивести результат, він спочатку конвертується до вигляду вектора типу Std_Logic_Vector, а вже потім подається на вихід.

Слід відзначити, що для прискорення достосування коду VHDL до відповідного розміру перемножуваних чисел та модуля доцільним вбачається розроблення спеціальної функції, за допомогою якої код генеруватиметься автоматично.

3.2. Моделі суматора точок на еліптичній кривій з використанням обчислень, основаних на теоретико-числових базисах Радемахера-Крестенсона та паралельному додаванні

3.2.1 Загальна модель суматора точок на кривій *GF*(p).

Застосування моделі обчислень на основі теоретико-числового базису Крестенсона, зокрема в криптографії, що грунтується на еліптичних кривих, вимагає побудови насамперед суматора точок. Обчислення суми точок на кривій чи подвоєння точки належать до основних операцій в цій криптографії. Тому в подальшому піддано аналізу суматор точок, що оперує на проективних або змішаних координатах. Причина такого вибору випливає з матеріалу, обговореного в розділі 2. В наступній частині роботи автором проектується суматор, оснований на змішаних координатах. Вибір такого підходу викликаний тим, що даний суматор призначений для обчислень дискретного логарифму, а структура ро-методу Полларда дозволяє використовувати цей тип додавання. Таким чином, щоб виконати операцію додавання двох точок, поданих у змішаній формі, належить виконати послідовность операцій, показаних у таблиці 1.3, причому в цьому випадку елімінується необхідність обчислення λ_2 і λ_5.

Проведений теоретичний аналіз свідчить, що модель суматора точок на еліптичній кривій може бути побудована з одинадцяти незалежних множників, двох суматорів та п’яти перемножувачів. Такий спосіб може виявитися неможливим для фізичної реалізації суматора в програмованих матрицях, оскільки для побудови такої складної системи не вистачило б необхідної логіки. Другий

спосіб полягає у використанні логічного пристрою, який керує виконанням операцій в потрібній послідовності, як це показано на рисунку 3.7. Крім цього, цей пристрій керування виконує операції зсуву бітів направо та наліво shl1 і shr1, що відповідає множенню та діленню даного числа на 2, а також є тривіальним за рахунок використання чисел у вигляді двійкових векторів. Слід звернути увагу на припущення, яке було прийняте при створенні моделі та передбачає, що суматор і перемножувач є незалежними пристроями або процесорами, які можуть працювати незалежно. Виходячи з цього, модель суматора точок схематично може бути представлено в спосіб, як це наведено на рисунку 3.7.

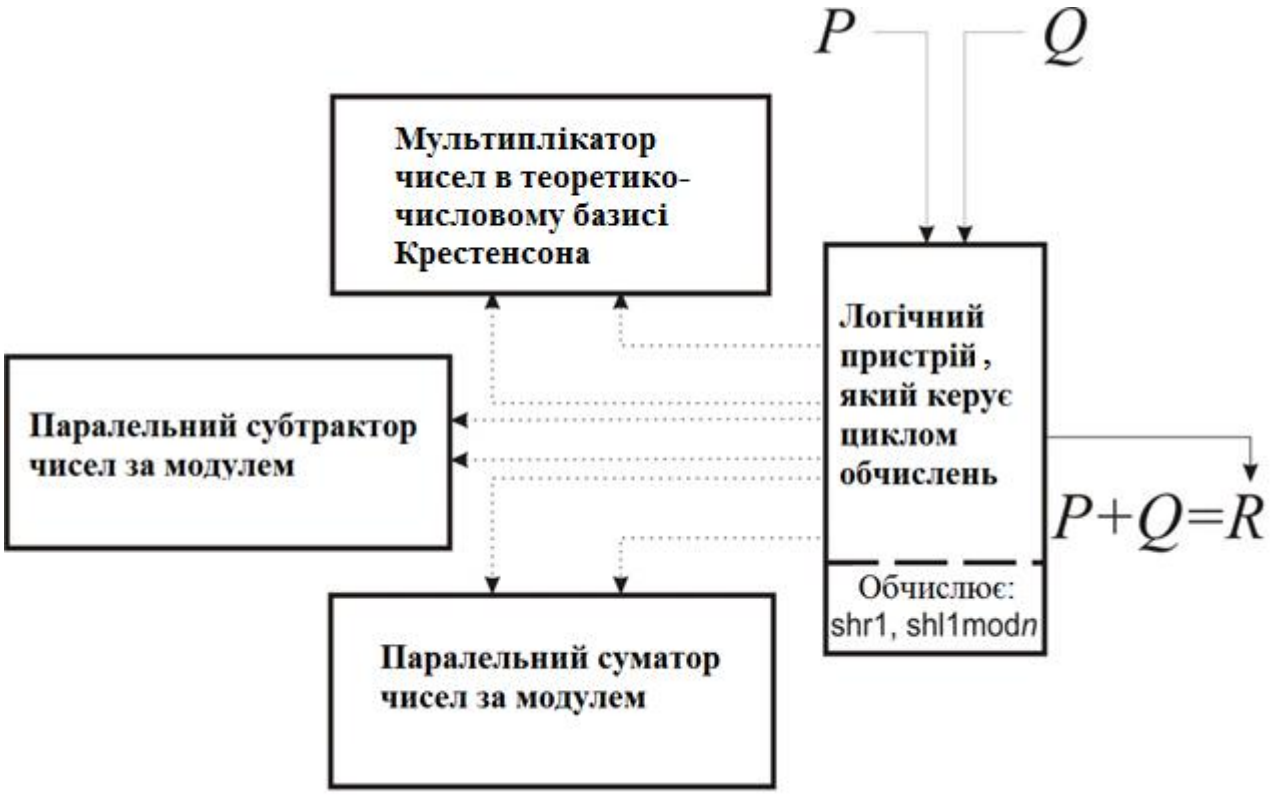

Рисунок 3.7 – Модель суматора точок на еліптичній кривій

3.2.2 Модель апаратної реалізації додавання точок на еліптичних кривих GF(p) в програмованих вентильних матрицях.

Для її побудови апаратної моделі додавання точок на еліптичній кривій використано складові фізичних моделей, запропонованих у попередньому розділі. Обидва блоки апаратної моделі для відповідної складової криптографічного

пристрою ІУСЕК запроектовано таким чином, що працюють на даних того самого типу, зокрема як вхідні, так і вихідні дані мають тип Std_Logic_Vector, що дозволяє уникнути труднощів виконання перетворень під час конверсії.

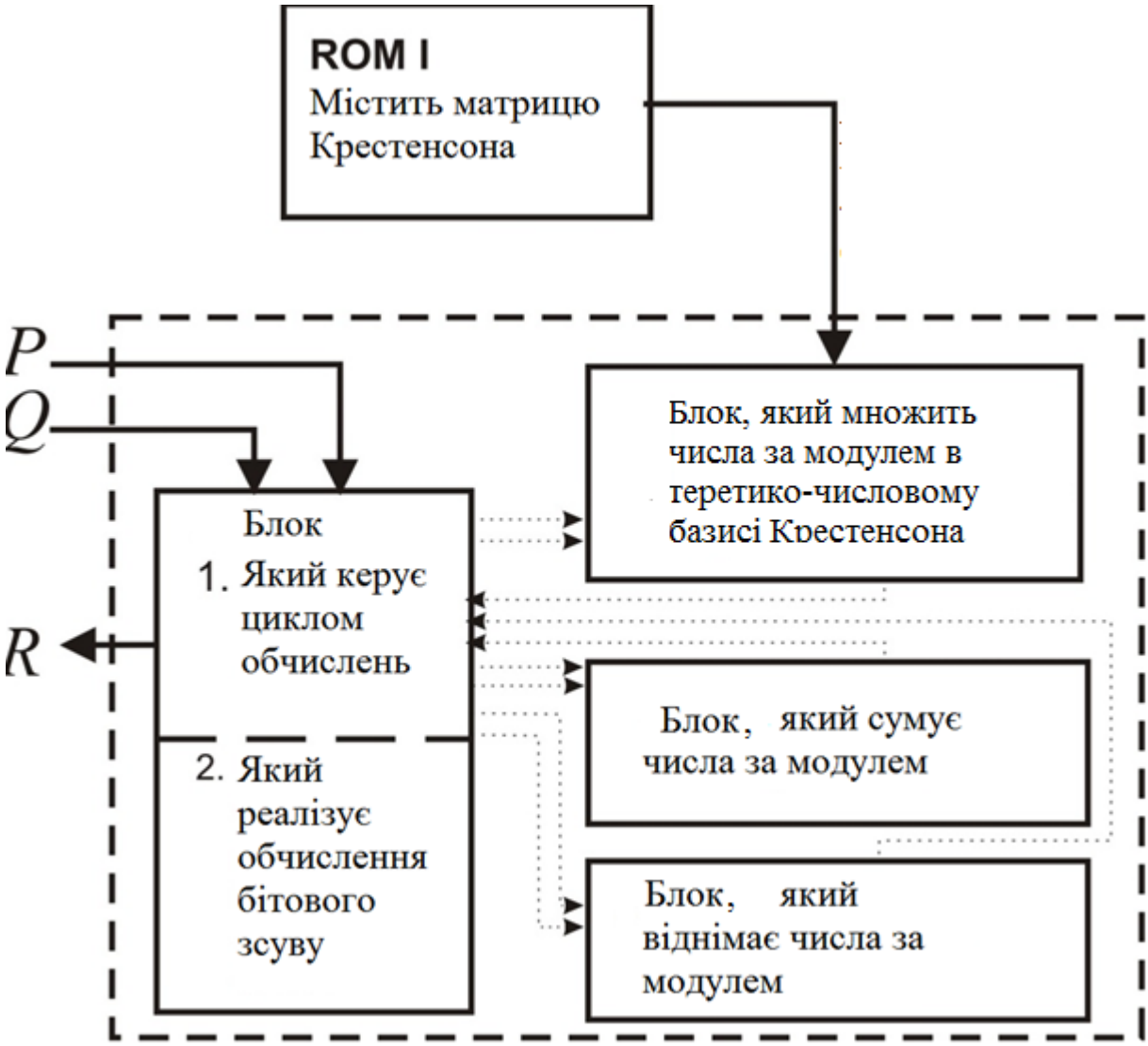

Рисунок 3.8 – Модель апаратного суматора точок

Операційний пристрій додавання точок, що показаний на рисунку 3.8, завантажує числа у вигляді двійкових векторів, причому вектори *a* та *b* є відповідними координатами точки *P*, а вектори *c* та *d* – це координати точки *Q*. Тоді як сума *R* вивантажується у вигляді двох векторів *x* та *y*. Операційний пристрій додавання точок описано за допомогою компонента gfmkadd_p, а саме:

```
entitygfkadd_pisport(
a, b, c, d : in std_logic_vector(91 downto 0);
clk, rst, next : in std_logic;
x, y : out std_logic_vector(91 downto 0));
end gfkadd_p;
```

Обчислення результату множення та ділення на 2, тобто зсув бітів для випадку запису у вигляді векторів, є тривіальним і згідно з визначенням таких дій полягає у зсуві значень окремих бітів. Обчислення модуля для випадку множення здійснюється відповідно до попереднього опису. Більше технічних особливостей щодо додавання точок в апаратній моделі представлено в четвертому розділі.

3.3. Моделі обчислень дискретного логарифма із застосуванням теоретико-числових базисів Радемахера-Крестенсона та паралельного додавання на підставі ро-методу Полларда

3.3.1 Моделі реалізації ро-методу Полларда розв’язання дискретного логарифма.

Побудову моделі, що реалізує розв’язок дискретного логарифму для оцінювання рівня стійкості криптографічних пристроїв на еліптичних кривих ECCD в ІУСЕК, автором розпочато з аналізу алгоритму, що міститься в таблиці Б.1 (Додаток Б). Цей алгоритм характеризує хід знаходження дискретного логарифма на еліптичній кривій з можливістю паралельного проведення обчислень.

Один із способів прискорення роботи такого алгоритму полягає устворенні, зокрема для повторюваних обчислень, спеціалізованого програмного або апаратного блоку. Таке рішення представлено в [95] і торкається розв’язання дискретного логарифма для кривих $GF(2^m)$. Модель циклічних обчислень передбачає реалізацію тільки частини обчислень, що містяться між пунктами 11 та 17 ро-методу Полларда. Таким чином, виконуються обчислення, які наведено в таблиці 3.2 між директивами repeat та until.

Таблиця 3.2

Циклічне обчислення ро-методу Полларда

11.	**repeat**
12.	**if** $X \in W$ ця точка виділена **then**
13.	Передай (α , β , X) до центральної бази точок
14.	**end if**
15.	Обчисли поточний поділ $i = g(X)$
16.	Обчисли наступну точку $X \leftarrow X + R_i$; $\alpha \leftarrow \alpha + a_i \bmod n$; $\beta \leftarrow \beta + b_i \bmod n$
17.	**until** якщо в базі виявиться колізія двох точок

Подальша частина алгоритму виконується за межами запропонованої моделі стандартним способом. Поза межами також обчислюються стартові точки α і β та таблиці R_i, a_i, b_i. Запропоновану модель циклічних обчислень представлено схемою, зображеною на рисунку 3.9. У ній не застосовано раніше побудованої апаратної моделі додавання точок, оскільки в подальшому її буде інтегровано та скомпоновано в ро-метод Полларда, що значно спростить логіку дії.

Точки R_i, які записано в таблиці, генеруються поза моделлю за допомогою рівняння

$$R_i \leftarrow a_i P + b_i Q \tag{3.5}$$

аналогічно до того, як відповідні рівнянню коефіцієнти a_i, b_i, які вибираються випадковим чином з діапазону $[1,\ldots, n-1]$. Стартові параметри генеруються за межами системи на основі

$$X \leftarrow \alpha P + \beta Q. \tag{3.6}$$

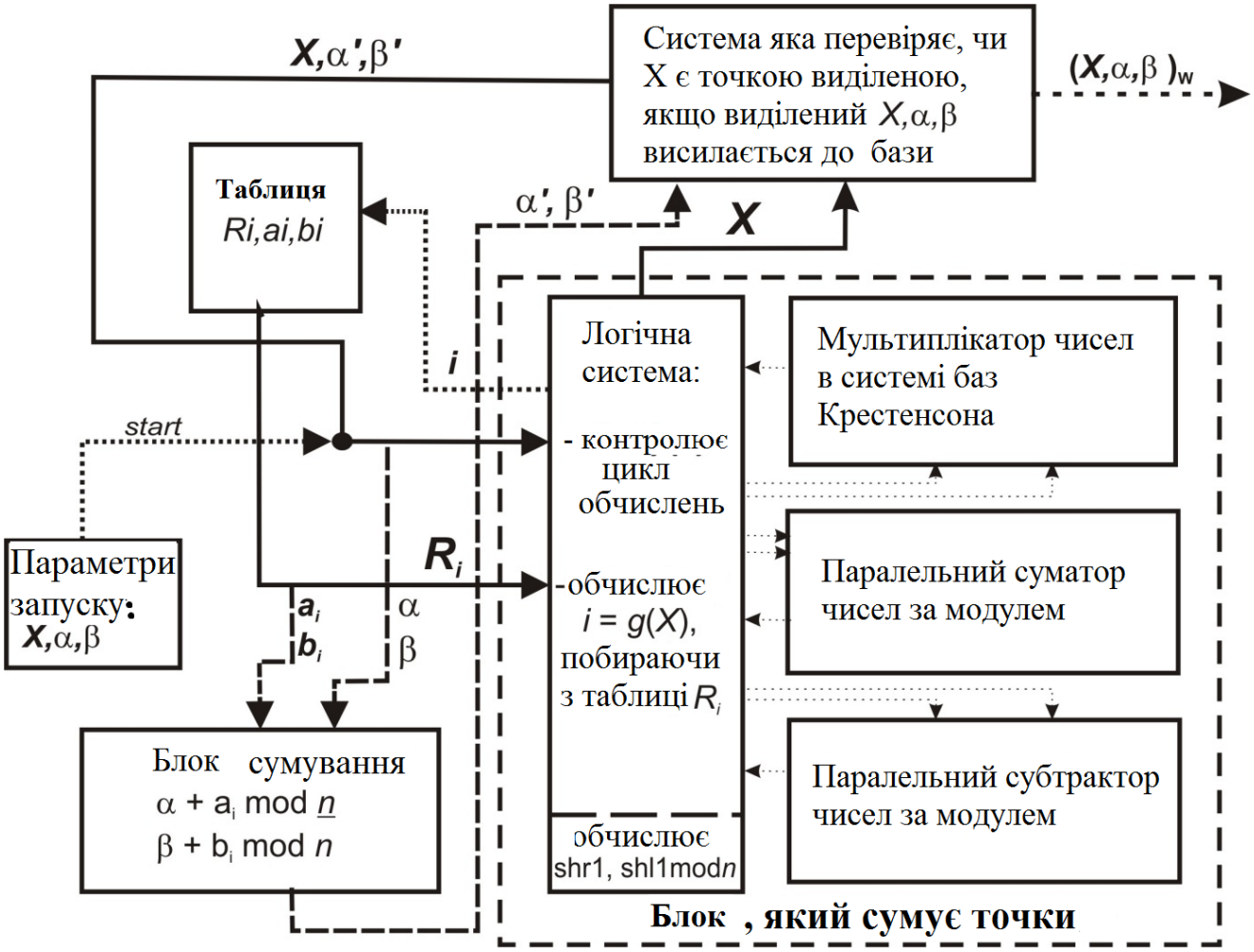

Рисунок 3.9 – Загальна модель обчислень дискретного логарифма за допомогою ро-методу Полларда з використанням дій на основі теоретико-числового базису Крестенсона

Слід зазначити, що таблицю точок R_i записано в афінних координатах, тоді як стартову точку X подано у проективних координатах. Тому додавання виконується у змішаних координатах, а результат X' згідно з алгоритмом змішаного додавання виводиться в проективних координатах. Отже, в наступному циклі отримується афінне R_i та проективне X'. Кожна точка X перевіряється і якщо вона не відповідає критеріям виділення, то $X \in W$ висилається до бази, де перевіряється можлива колізія та обчислюється дискретний логарифм.

3.3.2 Апаратна модель реалізації ро-методу Полларда розв’язання дискретного логарифма.

У цьому пункті представлено апаратно-програмну модель реалізації алгоритму обчислень ро-методу Полларда, що є складовою для оцінювання рівня живучості ІУСЕК з точки зору стійкості криптографічних пристроїв на еліптичних кривих ECCD на атаки криптоаналізу. Подібне рішення, яке стосується кривих $GF(2^m)$, наведено в [95]. У розглянутій моделі вхідними даними є тільки сигнали керування, тоді як вихідними даними є виділені точки разом з додатковими параметрами, що передаються у вигляді бітів. Цей компонент також відповідає за керування обчисленнями:

```
Entity gfkp is port(
        clk, rst : in std_logic;
        TxD: out std_logic );
end gfkp
```

Як вже було описано у другому розділі, в пункті 2.3.1 компонентом здійснюватиметься лише частина алгоритму, яка представлена в таблиці 3.2. Інші елементи реалізовано програмно за межами компонента. В апаратній моделі розроблено два блоки, що виконують паралельне додавання чисел великої розрядності за модулем:

```
component gfmkadd
 port( a, b : in std_logic_vector(91 downto 0);
       clk, rst, next_ : in std_logic;
       h: out std_logic_vector(91 downto 0)
      );
end component;
```

блок перевірки точок (виділені точки):

```
component compare_p
 port( bgp : in std_logic_vector(3 downto 0);
       bip : out std_logic);
end component;
```

Отже, перевіряються тільки три біти. Для перевірки критерію виділеності точок застосовано перевірку трьох наймолодших бітів координати *x* точки.

Блок, в якому перемножуються числа великої розрядності за модулем, базується на діях із використанням теоретико-числових базисів Крестенсона. Спосіб множення з використанням теоретико-числових базисів Радемахера-Крестенсона обговорено у другому розділі, підрозділ 2.3.1. Отже

```
component gfkm
 port( a, b : in std_logic_vector(91 downto 0);
       clk, rst, next_ : in std_logic;
       h : out std_logic_vector(91 downto 0));
end component;
```

Також, необхідним є використання двох блоків пам'яті, один з яких служить для зберігання таблиць R_i, a_i, b_i, що містять постійні дані, потрібні для реалізації ро-методу Полларда, та які обчислюються поза межами системи. Дані в пам'яті зберігаються у вигляді векторів, що складаються з координат *x* та *y* точки R_i, а також коефіцієнтів a_i та b_i, записаних у вигляді векторів.

Компонент завантажує точки R_i з пам’яті ROM на підставі адреси:

```
component gf2f3mem
 port( address: in std_logic_vector (3 downto 0);
       clock: in std_logic ;
       q: out std_logic_vector (367 downto 0));
 end component;
```

З метою адресації для функції $i=g(x)$ вибрано найменш значущі три біти координати *x* точки *X*.

Другий блок пам'яті призначено для зберігання обчисленого теоретико-числового базису Крестенсона для даного модуля. Архітектуру цього блоку пам'яті представлено в розділі 3.1.6, обслуговує її компонент:

```
component gfkmem
 port( address   : in integer range 0 to 182;
       clock     : in std_logic ;
       q         : out std_logic_vector (91 downto 0));
end component;
```

Комунікацію з програмною частиною системи реалізовано за допомогою окремого компонента, показаного нижче:

```
component uart_top
 port( clk : in  std_logic;
       TxD : out  std_logic;
       start : in std_logic;
       data_big_1 : in std_logic_vector (367 downto 0);
       gotowy : out std_logic);
end component;
```

Слід зазначити, що зв’язок з програмною частиною може бути реалізований довільним способом, оскільки в цьому випадку швидкість роботи не відіграє особливо важливої ролі, тому що передається інформація тільки про деякі точки, а саме – про виділені точки. Програмну частину моделі забезпечено базою даних, в якій відбувається запис виділених точок і перед кожним дописуванням здійснюється перевірка з базою записуваної точки.

Апаратну модель, яка реалізує ро-метод Полларда, показано на рисунку 3.10.

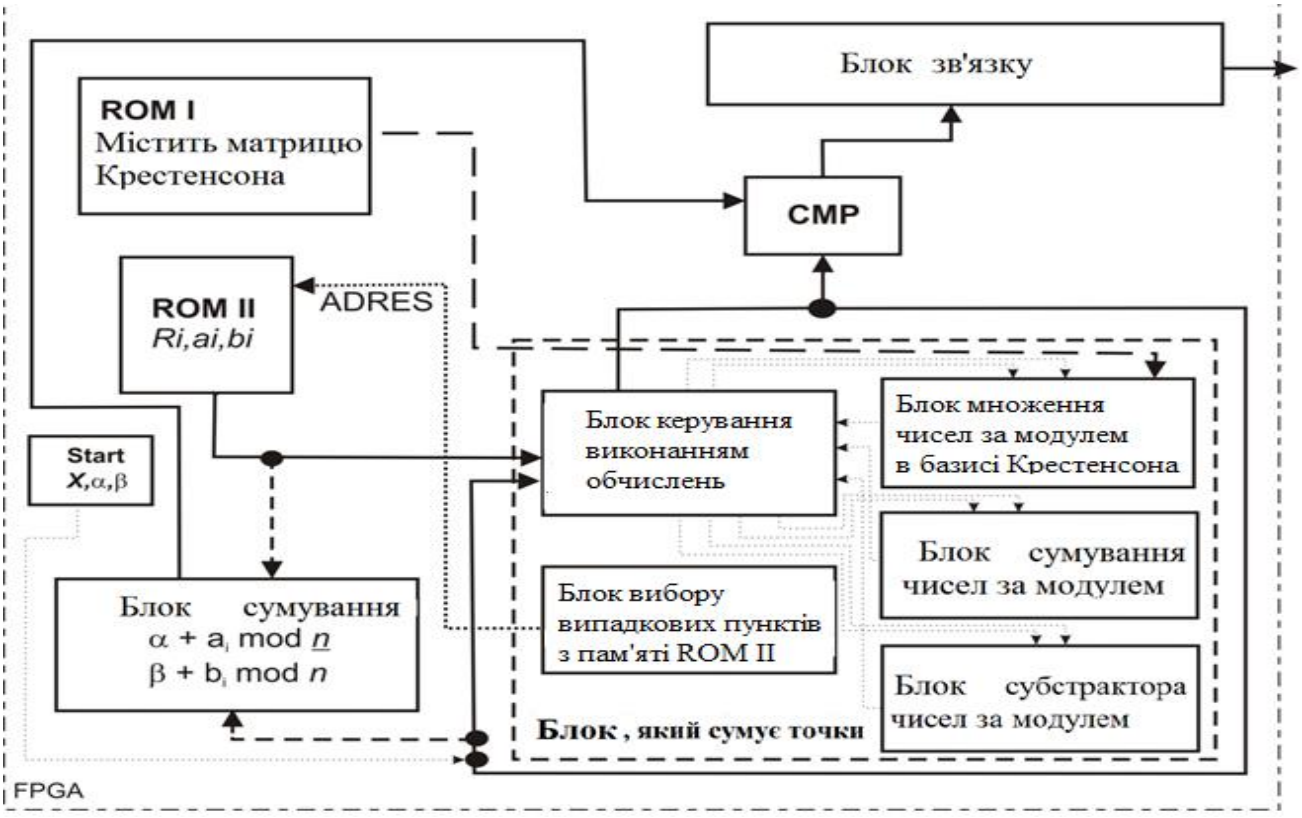

Рисунок 3.10 – Апаратна модель реалізації ро-методу Полларда із застосуванням дій, основаних на теоретико-числових базисах Крестенсона

До програмної частини моделі висилаються лише дані про виділені точки, які потім передаються в базу даних, де відбувається перевірка точок і пошук колізії відповідно до ро-методу Полларда. Код VHDL всієї структури достосовано до вимог щодо довжини параметрів кривих, модуля, причому повністю автоматизовано. Додаткові коефіцієнти, які стосуються записаної в проективних координатах стартової точки X (таблиця R_i записана в афінних координатах), та стартові коефіцієнти α і β обчислюються поза системою, а потім записуються в структурі сталими.

3.3.3 Модель паралельної реалізації ро-методу Полларда обчислення дискретного логарифма з використанням теоретико-числового базису Крестенсона.

Характерна особливість моделі паралельної реалізації ро-методу Полларда полягає у застосуванні багатьох шляхів випадкового блукання, що здійснюються паралельно. Суть його функціонування зводиться до використання багатьох компонентів, які реалізують свої власні стежки блукання та записують дані до спільної бази даних. Усі компоненти, які здійснюють стежки блукання, отримують та завантажують однакові таблиці точок R_i та таблиці значень a_i і b_i. Побудова кожного компонента є ідентичною, єдина відмінність полягає в наборі трьох стартових параметрів, а саме стартової точки X та параметрів α і β. Це дає змогу побудувати продуктивні криптографічні пристрої на еліптичних кривих ECCD ІУСЕК та підвищити ефективність розв’язання дискретного логарифму, а тим самим пришвидшити виявлення атаки криптоаналізу. Апаратно-програмна модель такої системи зображена на рисунку 3.11.

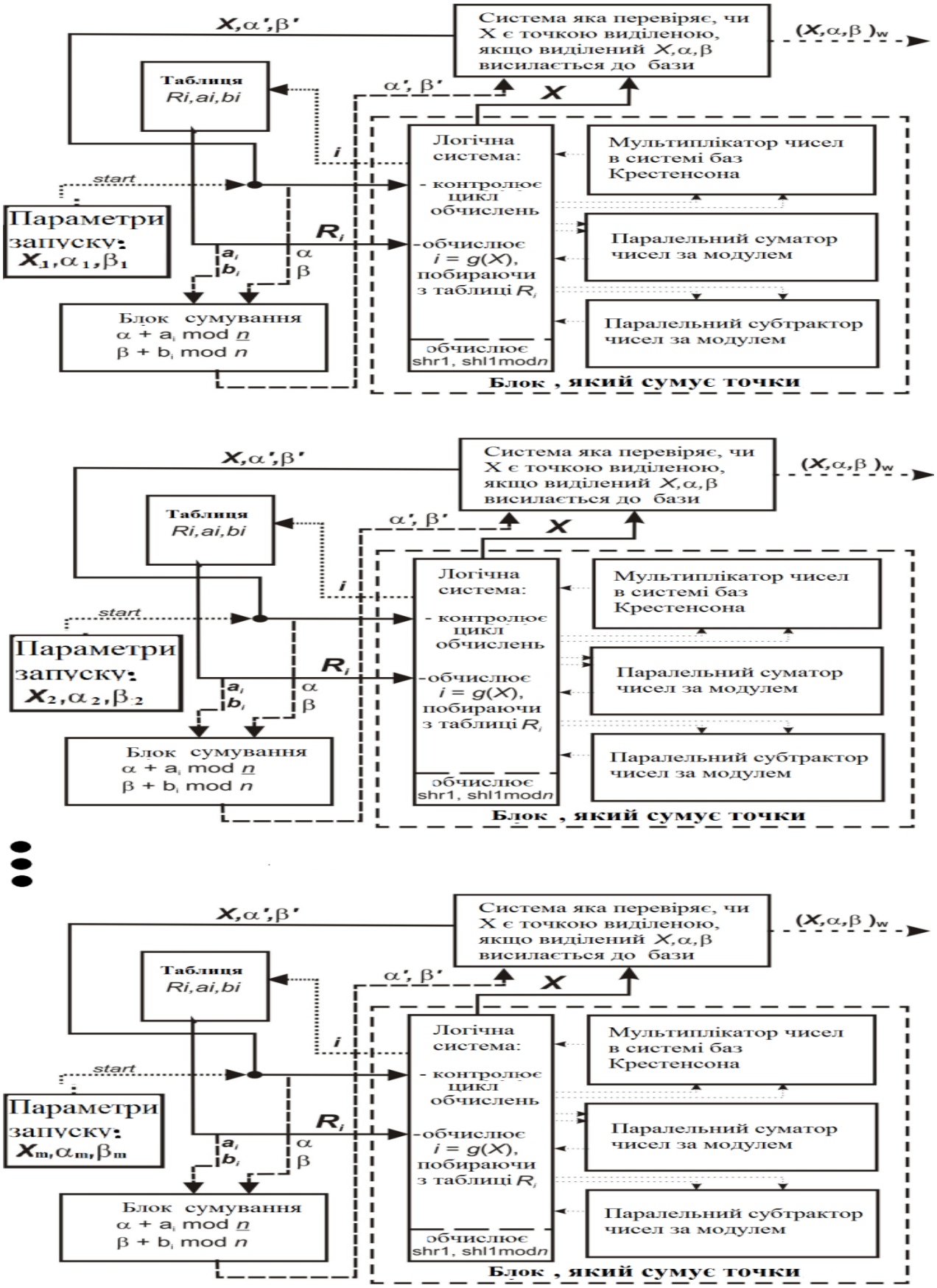

Рисунок 3.11 – Апаратно-програмна модель паралельної реалізації ро-методу Полларда з використанням теоретико-числового базису Крестенсона

3.3.4 Реалізація паралельного ро-методу Полларда обчислення дискретного логарифма з використанням теоретико-числового базису Крестенсона в апаратній моделі.

Реалізація паралельного ро-методу Полларда в апаратній моделі полягає в інтегруванні в систему багатьох блоків, кожен з яких працює над окремою стежкою випадкового блукання паралельно, зокрема здійснюється дублювання в блоках компонента `gfkp`. Цей компонент реалізує окрему поодиноку стежку блукання з модифікацією трьох стартових сталих (X, α, β). Кожен з компонентів, що беруть участь у реалізації ро-методу Полларда обчислення дискретного логарифма з використанням теоретико-числових базисів Крестенсона, отримує та завантажує також інші вище зазначені випадкові початкові коефіцієнти. Завдяки цьому можна розробити високопродуктивні криптографічні засоби на еліптичних кривих ECCD, зменшити час розв’язання дискретного логарифму, підвищити ефективність виявлення атаки криптоаналізу та оцінювання живучості ІУСЕК.

Висновки до розділу 3

1. Проведено дослідження впливу змодифікованих моделей, алгоритмів і мотодів, реалізованих на основі теоретико-числових базисів Радемахера-Крестенсона, на живучість ІУСЕК шляхом розв’язання дискретного логарифма на еліптичній кривій, що дає змогу досягти мінімальних розмірів еліптичних кривих, які можуть бути застосовані для забезпечення необхідних показників живучості ІУСЕК.

2. Побудовано і проаналізовано моделі та апаратні імплементації для виконання основних операцій на еліптичних кривих, в яких замінено множення цілих чисел великої розрядності за модулем на додавання за модулем за допомогою теоретико-числового базису Крестенсона, а також обґрунтовано здійснення поділу чисел великої розрядності на слова довжиною, котра відповідає розміру регістра процесора, та застосовано паралельне сумування складових виразів, завдяки чому зменшено обчислювальну складність, оптимально підібрано розмір слів до

регістрів, зменшуючи кількість тактів, необхідних для виконання основних операцій на еліптичних кривих.

3. Створено моделі та апаратні імплементації, основані на теоретико-числовому базисі Крестенсона і паралельному додаванні та за допомогою яких виконано ро-метод Полларда розв’язання дискретного логарифма, що дозволило побудувати продуктивні криптографічні засоби на еліптичних кривих ECCD, виявити атаки криптоаналізу з меншим часом та оцінити живучість ІУСЕК з вищою ймовірністю.

4. Визначено середовища, в яких застосування розроблених моделей та алгоритмів зумовлює вищу ефективність обчислень на кривих $GF(p)$ в ІУСЕК. Такими середовищами є компоненти вентильних матриць FPGA, комп’ютерні системи з багатоядерними процесорами чи багатопроцесорні сервери та комп’ютерні кластери, а також системи, в яких заімплементовано паралельну обчислювальну архітектуру CUDANVIDIA.

РОЗДІЛ 4
ДОСЛІДЖЕННЯ ТА ОЦІНЮВАННЯ РОЗРОБЛЕНИХ МОДЕЛЕЙ ТА ЗАСОБІВ ПІДВИЩЕННЯ ЖИВУЧОСТІ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ, БАЗОВАНИХ ЕЛІПТИЧНИХ КРИВИХ

У четвертому розділі здійснено симуляцію роботи моделей та технологій обчислень, запропонованих в попередньому розділі. Проведено аналіз симуляції роботи апаратно-програмних систем для розв'язування дискретного логарифма, який є підставою для підвищення живучості ІУСЕК. На основі результатів розв'язування дискретного логарифма для кривих різних розмірів здійснено оцінювання живучості ІУСЕК.

4.1. Дослідження та аналіз функціонування систем реалізації обчислень з використанням базисів Радемахера-Крестенсона і паралельного сумування чисел великої розрядності

4.1.1 Реалізація моделей паралельних суматора та віднімача за модулем.

У попередньому розділі побудовано паралельний суматор натуральних чисел великої розрядності. В цьому пункті представлено спосіб та особливості його роботи, швидкодія та обмеження, які залежать від прийнятої структури.

Сумування виконується за модулем n, де n – задане натуральне число. При цьому числа, які додаються, – це X ∧ Y (тут X < n, Y < n), причому сума $X + Y < 2n$, так що обчислення модуля n зводиться до знаходження різниці $(X + Y) - n$. Рисунок 4.1 ілюструє спосіб сумування з поділом на кроки, які не можуть виконуватися паралельно.

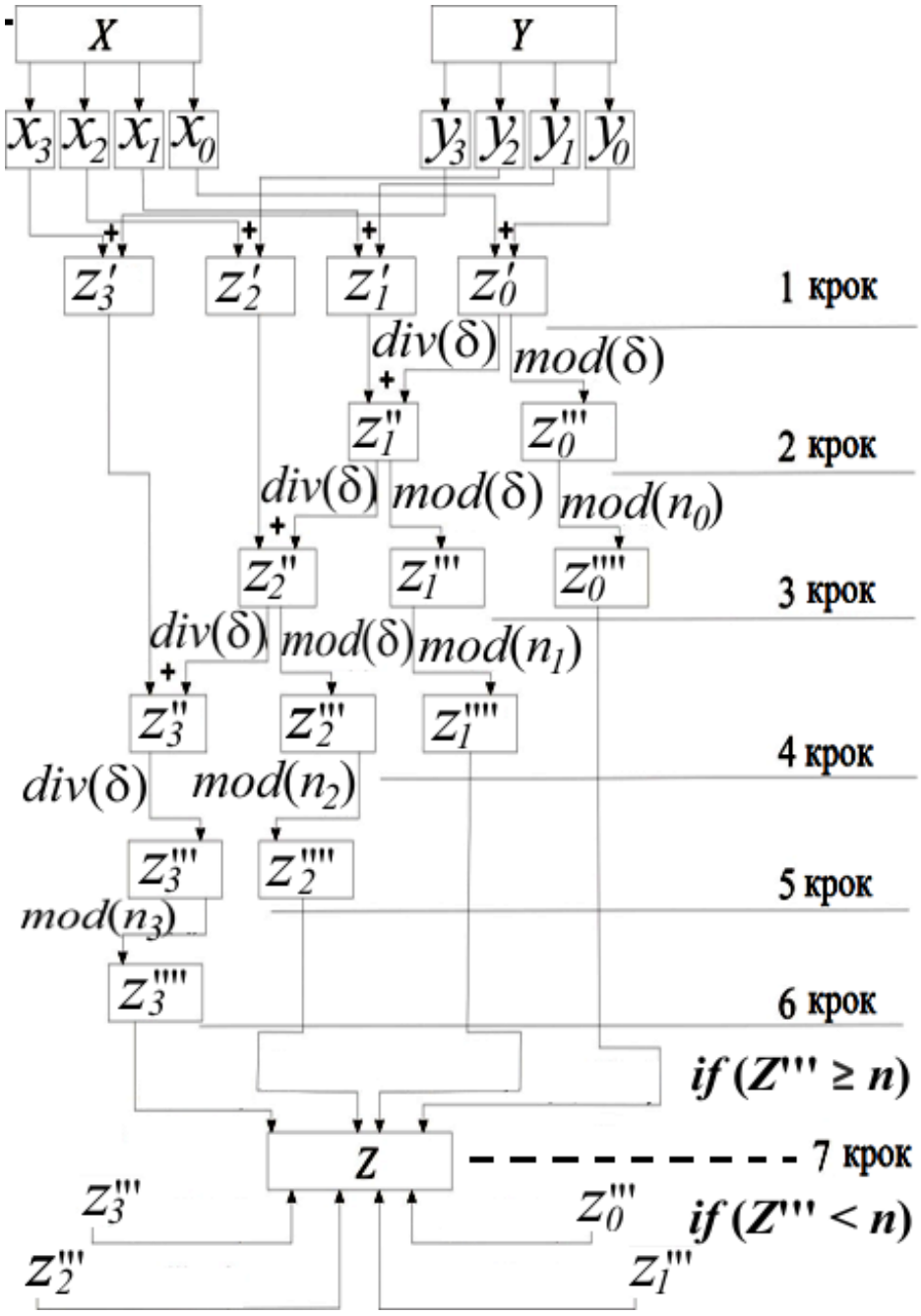

Рисунок 4.1 – Схема алгоритму сумування двох чисел за модулем

Обчислення суми двох чисел, яке складається з чотирьох слів, вимагає виконання семи кроків. Під час додавання, як показано на рисунку 4.1, проводиться також операція паралельного віднімання відповідно до алгоритму, який наведено у третьому розділі, пункт 3.1.4. При цьому слід відзначити, що порівнюється не ціле число, а окремі слова, щоб на підставі порівняння всіх слів здійснити порівняння чисел. Паралельне сумування чисел в поєднанні з відніманням за модулем дозволяє зменшити кількість кроків, які повинні бути виконані для того, щоб додати числа

за модулем. Досі обговорено числа, поділені на чотири слова. Надалі проаналізовано випадок зміни складності обчислень для більших чисел, які поділено не на чотири, а на п'ять, шість і т.д. слів. Доведено, що якщо в схему алгоритму, зображену на рисунку 4.1, ввести додатково одне слово, то тоді потрібно збільшити кількість кроків на два, модифікуючи п'ятий крок, і після нього додати додатково два кроки. Звідси випливає, що для випадку поділу числа на п’ять слів кількість необхідних кроків потрібно збільшити до дев'яти. Поширюючи даний підхід на більшу кількість слів поділу та узагальнюючи отримані дані, можна констатувати, що збільшення довжини числа на одне слово зумовлює зростання кількості кроків на два. Описується це залежністю:

$$k = 2w - 1, \tag{4.1}$$

де k – кількість кроків,

w – кількість слів, на які поділено число.

Для випадку віднімання чисел за модулем, що означає використання віднімача, будову якого наведено у третьому розділі в пункті 3.1.4, отримано такий результат: кількість кроків, необхідних для обчислення двох чисел за модулем, збігається з кількістю кроків, потрібних для суматора. Доведено, що таким же чином також змінюється кількість кроків, тому щоб обчислити кількість кроків, необхідних для числа, поділеного на задану кількість слів, слід використовувати залежність 4.1.

4.1.2 Дослідження моделі суматора, реалізованого в програмованих вентильних матрицях.

Аналіз роботи здійснено на підставі результатів досліджень обчислювальних засобів, побудованих на програмованих матрицях типу FPGA, додаючи 92-бітні числа, поділені на чотири слова. В подальшому проведено дослідження суматорів, за допомогою яких здійснюється додавання більших чисел, впливу збільшення швидкості додавання чисел на продуктивність виконуваних обчислень над

еліптичними кривими $GF(p)$. При цьому візьмемо до уваги вище наведену залежність між збільшенням розміру числа та зростанням кількості обчислень для знаходження суми чисел за модулем. Проте, це не єдина причина зниження швидкості обчислень. Друга причина полягає у зростанні складності побудови апаратної моделі на основі програмованих вентильних матриць, видовження доріжок в програмованій мікросхемі, що призводить до зменшення робочої частоти. Симуляція роботи апаратної моделі суматора, що базується на програмованих мікросхемах, дозволяє отримати суму чисел довжиною 92 біти протягом восьми тактів. Логічний синтез апаратної моделі, виконаній на програмованій матриці фірми Altera типу Stratix III 3SL150, дав змогу забезпечити робочу частоту на рівні 366 МГц. Робота на такій частоті дозволила виконати 45,75 млн. додавань за секунду для чисел, які складаються з чотирьох слів. Швидкодія сумування для числа, яке складається з п'яти слів, становить 35 млн. додавань за секунду для частоти 350 МГц, а для шести слів – 27,9 млн. додавань.

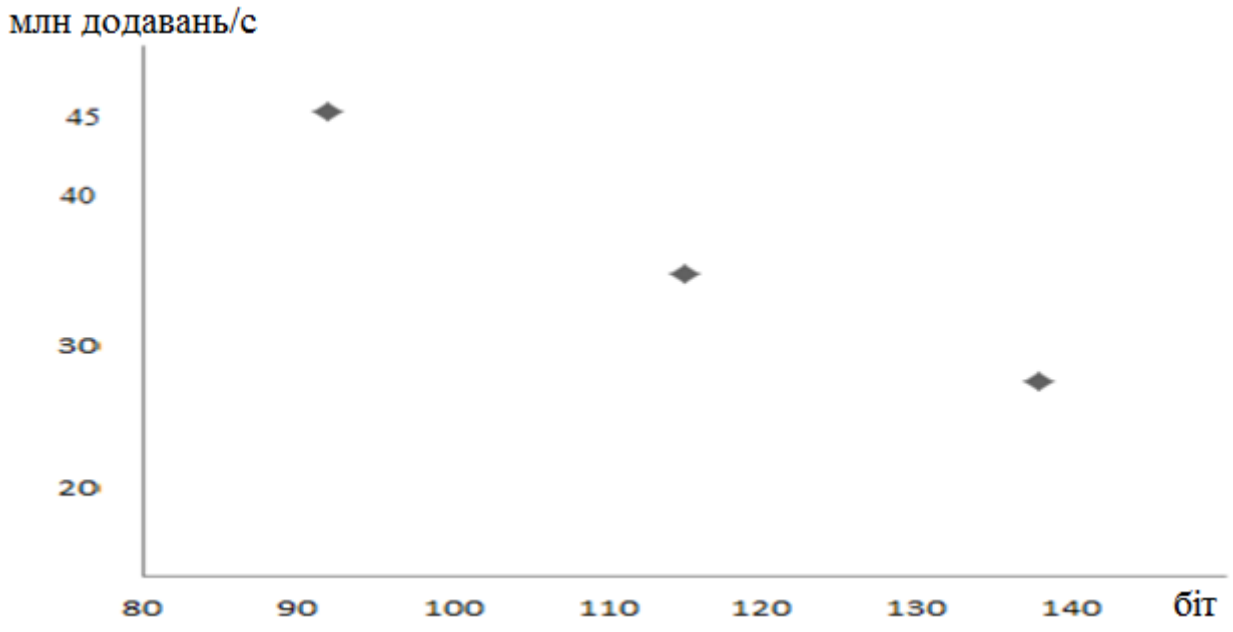

Рисунок 4.2 – Залежність швидкості додавання чисел від їх розміру

Можна зауважити майже лінійне зменшення швидкості обчислень в залежності від збільшення кількості слів. Результати проведених досліджень показують, що для випадку побудови апаратної моделі суматора точок на еліптичній кривій $GF(p)$,

робоча частота значно нижча. У зв'язку з цим результати, що отримуються під час симуляції функціонування апаратної моделі сумування чисел, відрізнятимуться.

4.1.3 Дослідження апаратної моделі перемножувача цілих чисел великої розрядності за модулем на основі теоретико-числових базисів Радемахера-Крестенсона.

Піддано тестуванню побудовану в п. 3.1.6 модель перемноження чисел великої розрядності на предмет визначення швидкодії. Для цього досліджено апаратну модель перемножувача чисел розміром 92 біти, поділених на чотири слова. Схема алгоритму, зображеного на рисунку 4.3, представляє дії, необхідні для обчислення добутку двох чисел за модулем, виконуючи відповідні сумування елементів матриці Крестенсона.

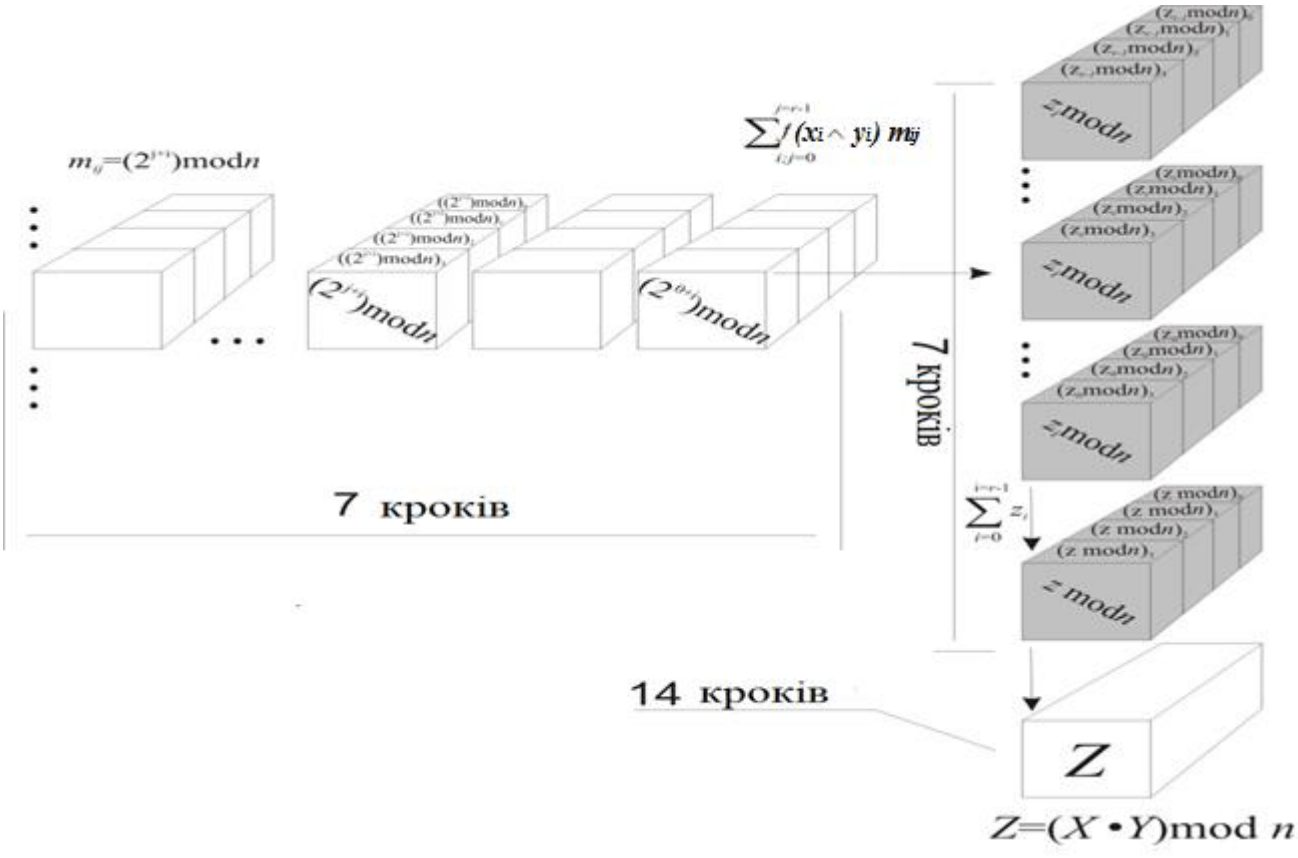

Рис 4.3. Схема алгоритму виконання операції множення в теоретико-числових базисах Крестенсона

Додавання значень всіх рядків матриці здійснюється паралельним способом, причому обчислення суми рядків отримується протягом виконанню семи кроків. Сумування рядка здійснюється аналогічно тому, яке описане попередньо для випадку суматора, проте єдиним винятком є додавання на першому кроці багатьох чисел, що наведено в п. 3.1.4. На наступному кроці виконано сумування вектора, що показаний «затемненим» на рисунку 4.3, також аналогічним чином протягом 7 кроків. Для того щоб обчислити добуток двох чисел, поділених на чотири слова, потрібно виконати 14 кроків. Доведено, що збільшення довжини числа на одне слово призводить до збільшення на 4 кроки виконуваних операцій, які потрібні для обчислення добутку. Це можна записати у вигляді рівняння:

$$k = 4w - 2 \tag{4.2}$$

де k – кількість кроків,
w – кількість слів, на які поділено число.

4.1.4 Симуляція роботи апаратної моделі перемножувача, реалізованого на основі програмованих вентильних матриць.

Дослідження проведено на апаратній моделі перемножувача, будову якого представлено в п. 3.1.6. Аналіз роботи блоку множення розпочато з виконання операцій над 92-бітовими числами. Проведено симуляцію для апаратної моделі, яку передбачено для обчислень, що здійснюються на числах великої розрядності. Для блоку, пристосованого для виконання операцій на 92-бітних числах, а отже поділених на чотири слова, результат множення отримано протягом 17-ти тактів. Відповідне генерування і синтез операційного пристрою проведено для програмованої матриці FPGA типу Stratix III EP3SL150F1152I4SL. При цьому отримано тактову частоту на рівні 44 МГц. Для тестування використано раніше описаний операційний пристрій. Його робота дозволяє виконати 2,6 млн. множень за секунду чисел розміру 92 біти. Для інших розмірів чисел результати генерування та синтезу дали змогу отримати аналогічні частоти функціонування операційного

пристрою. У таблиці 4.1 зведено результати функціонування перемножувача чисел різного розміру.

Таблиця 4.1

Швидкодія множення для апаратної моделі, базованої на програмованих матрицях FPGA, для чисел різного розміру

Довжина числа, біт	69	92	115	138	161	184
Кількість слів	3	4	5	6	7	8
Кількість множень / с	3538462	2588235	2047619	1640000	1379310	1181818

Показаний на рисунку 4.4 графік ілюструє зменшення швидкості додавання відповідно до розміру сумованих цифр.

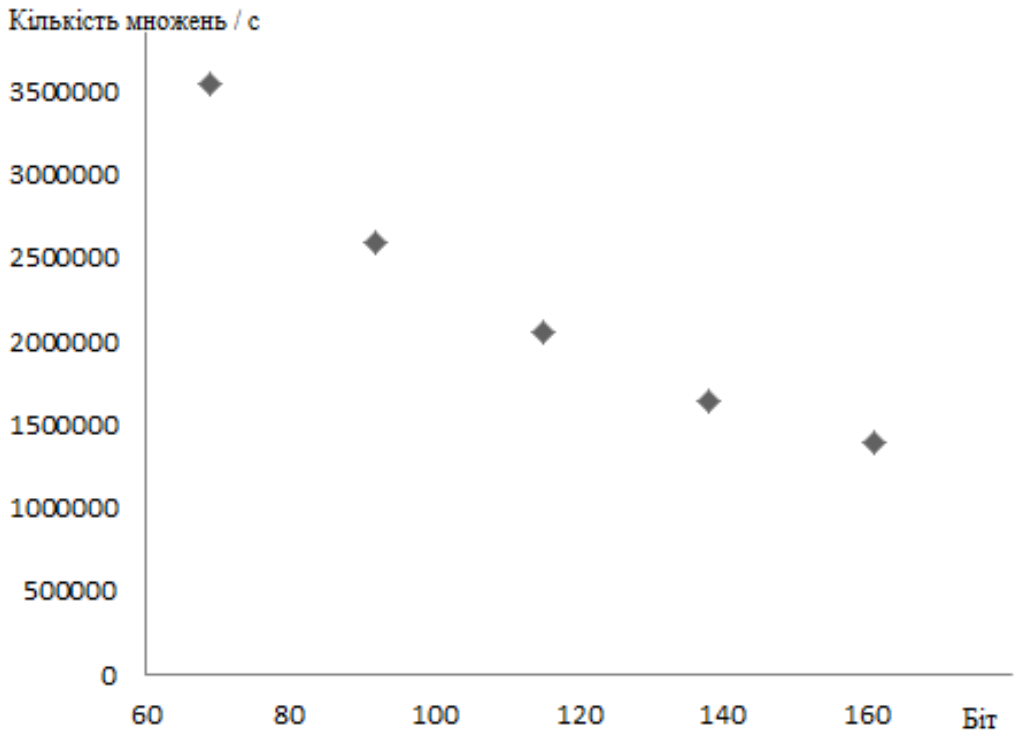

Рисунок 4.4 – Залежність швидкості додавання чисел від їх розміру

4.2. Дослідження та аналіз роботи суматора точок на еліптичній кривій із застосуванням обчислень в теоретико-числових базисах Рademахера-Крестенсона та паралельного додавання

4.2.1 Процеси у моделі суматора точок на еліптичній кривій *GF(p)*, побудованого на основі обчислень в теоретико-числових базисах Крестенсона і паралельного додавання.

Сумування точок у суматорі складається із виконання операцій додавання та множення чисел. Сумування точок у змішаному поданні зводиться до виконання наступних операцій над числами: 11-ти множень, 2-ох додавань, 5-ти віднімань та 2-ох бітових зсувів. Згідно з попередніми припущеннями можна виконати незалежно кожну з чотирьох дій у тому ж самому часі, оскільки це дозволено алгоритмом додавання точок (таблиця 4.2).

Таблиця 4.2

Послідовність операцій сумування точок в змішаних координатах

$\lambda_1 = X_1 Z_2^2$	$\lambda_8 = \lambda_4 + Y_2$
$\lambda_3 = \lambda_1 - X_2$	$Z_3 = Z_2 \lambda_3$
$\lambda_4 = Y_1 Z_2^3$	$X_3 = \lambda_6^2 - \lambda_7 \lambda_3^2$
$\lambda_6 = \lambda_4 - Y_2$	$\lambda_9 = \lambda_7 \lambda_3^2 - 2X_3$
$\lambda_7 = \lambda_1 + X_2$	$Y_3 = \left(\lambda_9 \lambda_6 - \lambda_8 \lambda_3^3\right)/2$

Всі операції сумування точок змішаним способом можуть бути здійснені за 11 кроків. Підхід до вибору порядку обчислень наведено в таблиці 4.3 Можна зауважити, що процес сумування точки зводиться до виконання операцій, які містяться в таблиці 4.2. Таблиця 4.3 ілюструє метод групування операцій, який дозволяє їх оптимальне виконання за критерієм найменшої кількості кроків.

Таблиця 4.3

Метод реалізації обчислень в суматорі під час додавання двох точок

№ кроку	Множення	Додавання	Віднімання	Зсув
1	Z_2^2			
2	$X_1 Z_2^2$			
3	Z_2^3	$\lambda_1 + X_2$	$\lambda_1 - X_2$	
4	$Y_1 Z_2^3$			
5	λ_3^2	$\lambda_4 + Y_2$	$\lambda_4 - Y_2$	
6	λ_6^2			$\frac{\lambda_8}{2}$
7	$\lambda_7 \lambda_3^2$			$\frac{\lambda_6}{2}$
8	λ_3^3	$\lambda_6^2 - \lambda_7 \lambda_3^2$		$\frac{\left(\lambda_7 \lambda_3^2\right)}{2}$
9	$\frac{\lambda_8}{2} \lambda_3^3$		$\frac{\left(\lambda_7 \lambda_3^2\right)}{2} - X_3$	
10	$\frac{\lambda_6}{2} \lambda_9$			
11	$Z_2 \lambda_3$		$\frac{\lambda_9}{2} \lambda_6 - \frac{\lambda_8}{2} \lambda_3^3$	

Відповідне групування операцій, а також прийняття відповідного порядку їх виконання дозволяє здійснювати сумування двох точок на еліптичній кривій *GF(p)* за одинадцять кроків. Зміна розміру чисел не впливає на зміну кількості кроків, які необхідно виконати під час додавання точок, а має тільки вплив на кількість операцій, які проводяться над числами. Сума двох точок на еліптичній кривій отримується, якщо:

$$k = (4w - 2) \cdot 11, \qquad (4.3),$$

де k – кількість кроків,

w – кількість слів, на які поділено число.

4.2.2 Процеси у моделі суматора точок *GF(p)*, реалізованого в програмованих вентильних матрицях.

Симуляцію процесів здійснено на апаратній моделі суматора точок *GF*(p), структуру якого представлено в п. 3.2.2. Аналіз симуляції процесів у моделі суматора точок розпочато від блоку, що виконує операції на 92-бітних числах. Для того щоб обчислити суму двох точок, потрібно зробити одинадцять кроків, представлених у таблиці 4.3, та, крім цього, слід призначити ще один крок для запам'ятовування числа. Отже, шукана сума отримується протягом дванадцяти кроків. Найкоштовнішою операцією є множення чисел, яка здійснюється блоком множення, – його роботу описано в п. 4.1.4. Апаратна модель суматора вимагає 17 тактів для обчислення добутку. Обчислення суми, різниці або бітового зсуву є операціями набагато дешевшими. Обчислення суми двох точок на еліптичній кривій *GF(p),* в яких координати поділено на чотири слова, а отже максимально 92-бітного числа в базованій на програмованій матриці FPGA моделі, виконується протягом 188 тактів. Симуляцію проведено для програмованої матриці FPGA типу Stratix III EP3SL150F1152I4SL. Результати симуляції дозволили отримати тактову частоту на рівні 44 МГц. Процеси, які протікають в апаратній моделі, яка базується на вищезгаданій програмованій матриці, дозволяють виконувати 234 000 сумувань за секунду для 92-бітного розміру координат. Слід відзначити, що для інших розмірів чисел результати моделювання дозволили отримати аналогічні частоти роботи. В таблиці 4.4 наведено результати досліджень процесів, які мають місце в апаратній моделі суматора для чисел різного розміру.

Таблиця 4.4

Продуктивність сумування точок *GF(p)* в апаратній моделі,
основаній на програмованій матриці FPGA, для чисел різного розміру

Крива *GF(p)*	69	92	115	138	161	184
Кількість слів	3	4	5	6	7	8
Кількість сумувань / с	319444,4	234042,6	185344,8	148550,7	125000	107142,86

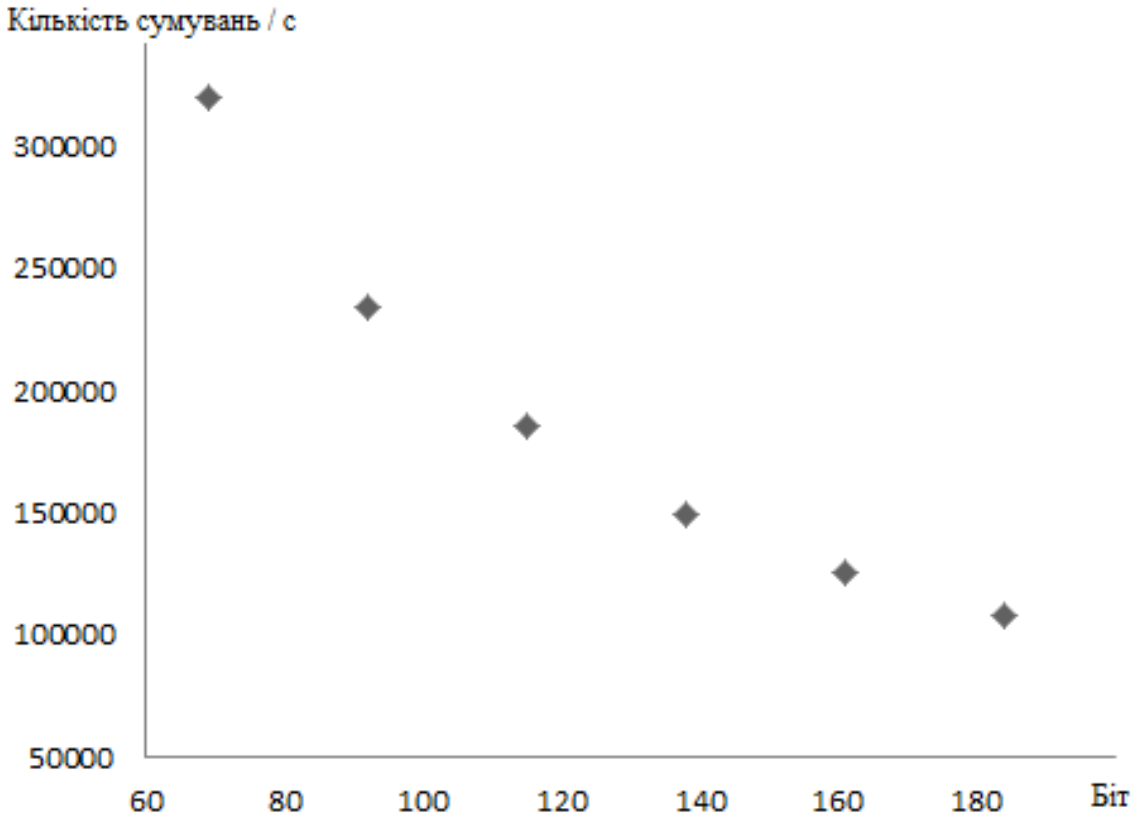

Рисунок 4.5 – Залежність швидкодії додавання точок на кривій $GF(p)$ від її розміру

4.3. Дослідження та аналіз функціональних характеристик суматора точок на еліптичній кривій при реалізації алгоритму шифрування Ель-Гамаля

4.3.1 Перебіг алгоритму Ель-Гамаля при застосуванні модифікованого суматора точок.

Виконано порівняння швидкодії асиметричного алгоритму шифрування Ель-Гамаля [7, 35, 76, 88], в якому застосовано сумування точок згідно з методом, що розглянутий у попередніх пунктах та ґрунтується на теоретико-числовому базисі Крестенсона. Для імплементації застосовано бібліотеку MIRACL. Допоміжним модулем використано програмовану матрицю FPGA, що реалізує додавання точок. Симуляція роботи системи, основаної на традиційних підходах до сумування

точок, дозволила визначити для кожної з розглянутих кривих час шифрування файлу обсягом 1024 кБ, що зведено в таблиці 4.5.

Таблиця 4.5

Швидкість шифрування файлу інформаційним обсягом 1024 кБ методом Ель-Гамаля із використанням стандартних алгоритмів сумування точок на еліптичній кривій

Крива $GF(p)$	89	97	109	131	163	191
Час, с	104	131	145	243	339	440

Симуляція алгоритму Ель-Гамаля, до якого було впроваджено алгоритм сумування точок на еліптичних кривих на основі теоретико-числового базису Крестенсона, дозволила отримати для тих же кривих час шифрування, значення якого наведені у таблиці 4.6.

Таблиця 4.6

Швидкість шифрування файлу інформаційним обсягом 1024 кБ методом Ель-Гамаля на основі системи залишків Радемахера-Крестенсона

Крива $GF(p)$	89	97	109	131	163	191
Час, с	37	40	43	79	111	127

З аналізу таблиць 4.5 та 4.6 можна зробити висновок, що введення сумування на основі теоретико-числових базисів Крестенсона у відповідному середовищі збільшує швидкість шифрування приблизно вдвічі порівняно з традиційним підходом, що візуально відтворено на рисунку 4.6.

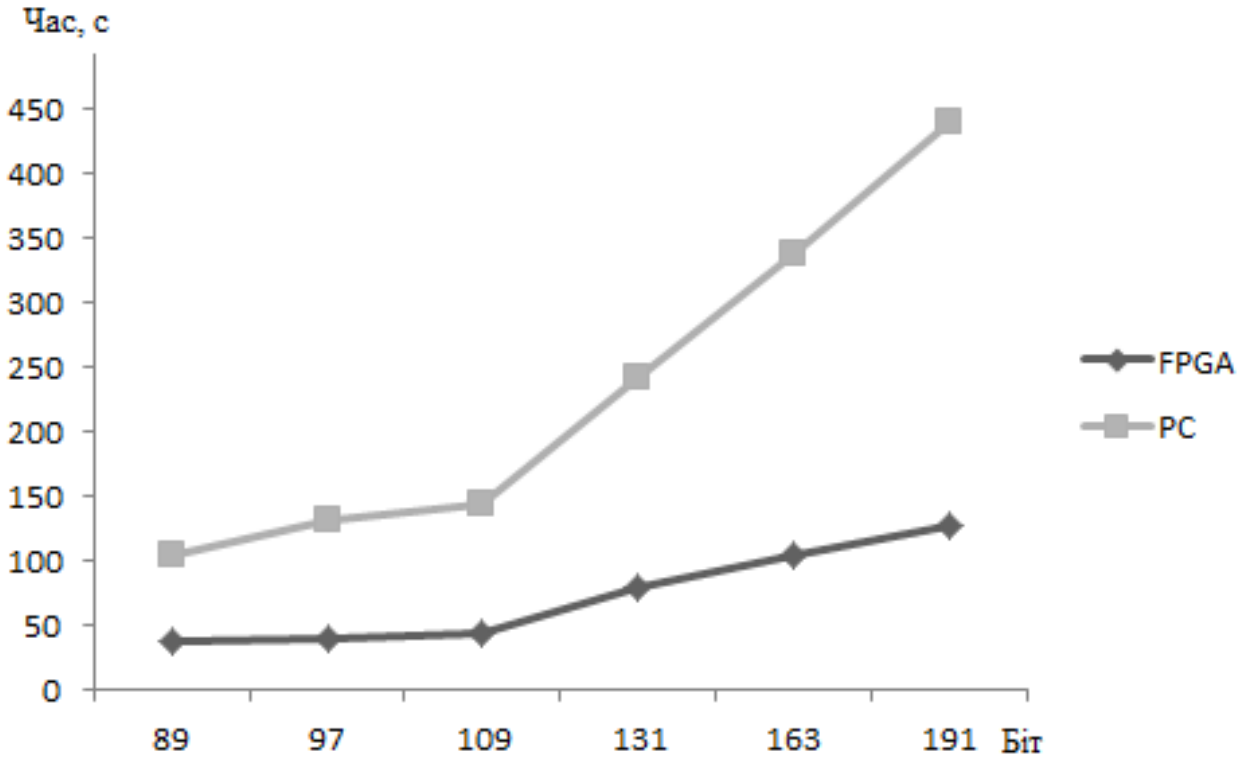

Рисунок 4.6 – Порівняння швидкості шифрування файлів розміру 1024 кБ методом Ель-Гамаля з використанням стандартних алгоритмів, реалізованих програмно (PC) та апаратно (FPGA) на еліптичній кривій на основі системи залишків Радемахера-Крестенсона

4.4. Дослідження та аналіз живучості ІУС, базуючись на розв'язанні дискретного логарифма модифікованим ро-методом Полларда

4.4.1 Дослідження процесів у моделі обчислення дискретного логарифма за допомогою ро-методу Полларда із застосуванням сумування на основі теоретико-числового базису Крестенсона.

Проведено дослідження процесів, які протікають в іншій моделі, ніж досі розглянуто, та в якій застосовано обчислення в теоретико-числових базисах Крестенсона, а також паралельні операції віднімання та додавання. Цією моделлю є модель обчислення дискретного логарифма з використанням ро-методу Полларда. Така модель обчислення, з точки зору наголошення акценту на швидкість виконання сумування, дозволяє визначити живучість різного типу

шифрів на основі еліптичних кривих *GF*(p). Досліджуючи та аналізуючи процеси у такій моделі, зроблено оцінювання продуктивності виконуваних операцій, що дало змогу оцінити час, необхідний для розв'язання дискретного логарифма. Згідно з цією моделлю основна задача полягає у виконанні циклічного сумування точок на еліптичній кривій. У циклі обчислень передбачено обчислення суми точок, в яких одна з обчислюваних точок міститься в таблиці, а друга – поза першим додаванням – є результатом, отриманим в попередньому циклі.

Таблиця 4.7

Спосіб реалізації циклічних обчислень сумування точок

Крок	Множення	Додавання	Віднімання	Зсув	Поділ	Перевірка точки
1	Z_2^2					
2	$X_1 Z_2^2$					
3	Z_2^3	$\lambda_1 + X_2$	$\lambda_1 - X_2$			
4	$Y_1 Z_2^3$					
5	λ_3^2	$\lambda_4 + Y_2$	$\lambda_4 - Y_2$			
6	λ_6^2			$\frac{\lambda_8}{2}$		
7	$\lambda_7 \lambda_3^2$			$\frac{\lambda_6}{2}$		
8	λ_3^3	$\lambda_6^2 - \lambda_7 \lambda_3^2$		$\frac{\left(\lambda_7 \lambda_3^2\right)}{2}$		
9	$\frac{\lambda_8}{2} \lambda_3^3$		$\frac{\left(\lambda_7 \lambda_3^2\right)}{2} - X_3$		$g(X)$	CMP(X)
10	$\frac{\lambda_6}{2} \lambda_9$					
11	$Z_2 \lambda_3$		$\frac{\lambda_9}{2} \lambda_6 - \frac{\lambda_8}{2} \lambda_3^3$			

У таблиці 4.7 наведено всі операції, необхідні для здійснення одного шляху випадкового блукання. На підставі аналізу даних, що містяться в таблиці 4.7, отримано, що всі необхідні операції можна виконати на протязі 11-ти кроків. Найкоштовнішою операцією в процесі є операція множення, яка реалізується за допомогою використання алгоритму на основі теоретико-числових базисів Крестенсона протягом (2 *w* - 1) кроків, де *w* – кількість слів, на які поділене число. Суми послідовних точок на стежці випадкового блукання отримуються після 11 кроків – відповідно до алгоритму, показаного в таблиці 4.7. Зміна розміру чисел не впливає на зміну кількості кроків, які потрібно здійснити при додаванні точок, тільки має вплив на кількість операцій, виконуваних над числами. Суми наступних точок на шляху випадкового блукання для кривих GF*(p),* отримуємо протягом виконання *k* кроків, які визначаються з виразу:

$$k = (4w - 2) \cdot 11, \tag{4.4}$$

де *w* – кількість слів, на яку було поділено число.

В обчисленнях прийнято, що до колізії доходить в середньому після $\sqrt{\pi n/2}$ ітерацій (сумувань точок). Виходячи з цього, можна припустити, що очікувана колізія відбувається після виконання такої кількості кроків, яка задовольняє рівнянню:

$$k = 11(4w - 2)\sqrt{\pi n/2} \tag{4.5}$$

4.4.2 Дослідження процесів в апаратній моделі реалізації ро-методу Полларда для еліптичних кривих GF(p).

Для дослідження використано розроблений операційний пристрій, за допомогою якого виконується ро-метод Полларда та проводяться обчислення, що ґрунтуються на методі множення, реалізованому в теоретико-числовому базисі Крестенсона, та паралельному сумуванню чисел. Аналіз результатів симуляції

апаратної моделі, що реалізує сумування точок, розпочато від моделі, що виконує операції над 92-бітними числами. Розглянута модель передбачає виконання операції сумування точок, де першу точку, яку призначено для додавання, обчислено на попередньому кроці і записано в проективних координатах. Слід зазначити, що перебіг такого процесу стосується всіх кроків, окрім першого кроку, в якому точка задається у вигляді сталої та визначається індивідуально. За допомогою моделі виконується також вибір точки з пам'яті, так званого поділу. Поділ здійснюється на основі останніх чотирьох бітів координати x обчисленої точки. Це є друга точка суми, яка записана в афінних координатах. Більше того, модель також дає змогу перевірити, чи обчислена точка задовольняє критерію виділення, та посилає точку до бази, якщо ця точка відповідає критерію виділення. Крім цього, виконується також сумування за модулем додаткових коефіцієнтів. Згадані обчислення здійснюються спеціально розробленим для реалізації цієї мети суматором. Обчислення суми точок виконується кожні 11 кроків, окрім визначення першої суми, де повинно бути додано ще один додатковий крок. Циклічні обчислення кожної суми точок на кривій $GF(p)$ проведено для випадку поділу координат точок на чотири слова, що, отже, відповідає числу з максимальною кількістю бітів – 92. Ці обчислення виконано за допомогою апаратної моделі, основаної на програмованій матриці FPGA, протягом 187 тактів, окрім знаходження першої суми. Симуляцію роботи здійснено на базі програмованої матриці типу Stratix III EP3SL150F1152I4SL. При цьому отримано тактову частоту значенням 44 МГц. Для проведення тестів використано вище наведений операційний пристрій, який, як показали результати досліджень, дозволяє виконувати 235 000 сумувань за секунду для чисел розміру 92 біти. Для інших довжин чисел результати генерування та синтезу дали змогу отримати аналогічні частоти функціонування розробленого операційного пристрою. В таблиці 4.8 наведено результати дослідження процесів, які протікають у апаратно–програмній моделі для кривих різного розміру.

Таблиця 4.8

Кількість ітерацій алгоритму для різних кривих GF(p)

Крива *GF*(*p*)	69	92	115	138	161	184
Кількість слів	3	4	5	6	7	8
Кількість ітерацій / с	321678,3	235294,1176	186147,2	149090,9	125391,8	107438

Використовуючи результати компіляції, зібрані в таблиці 4.8, та вираз для визначення середнього очікуваного часу знаходження дискретного логарифма за допомогою ро-методу Полларда $\sqrt{\pi n/_2}$ для– порядку еліптичної кривої *n*, в таблиці 4.9 показано приблизний прогнозований час обчислення дискретного логарифма для еліптичних кривих різного розміру.

Таблиця 4.9

Прогнозований час знаходження дискретного логарифма для різних еліптичних кривих *GF*(p)

Крива	*GF*(69)	*GF*(92)	*GF*(115)	*GF*(138)	*GF*(161)	*GF*(184)
Час (дні)	0,84	3295	$12*10^6$	$4,2*10^{10}$	$1,7*10^{14}$	$5,8*10^{17}$

На рисунку 4.7 наведено збільшення часу, необхідного для розв’язання дискретного логарифма за допомогою однієї програмованої матриці FPGA залежно від розміру ЕК.

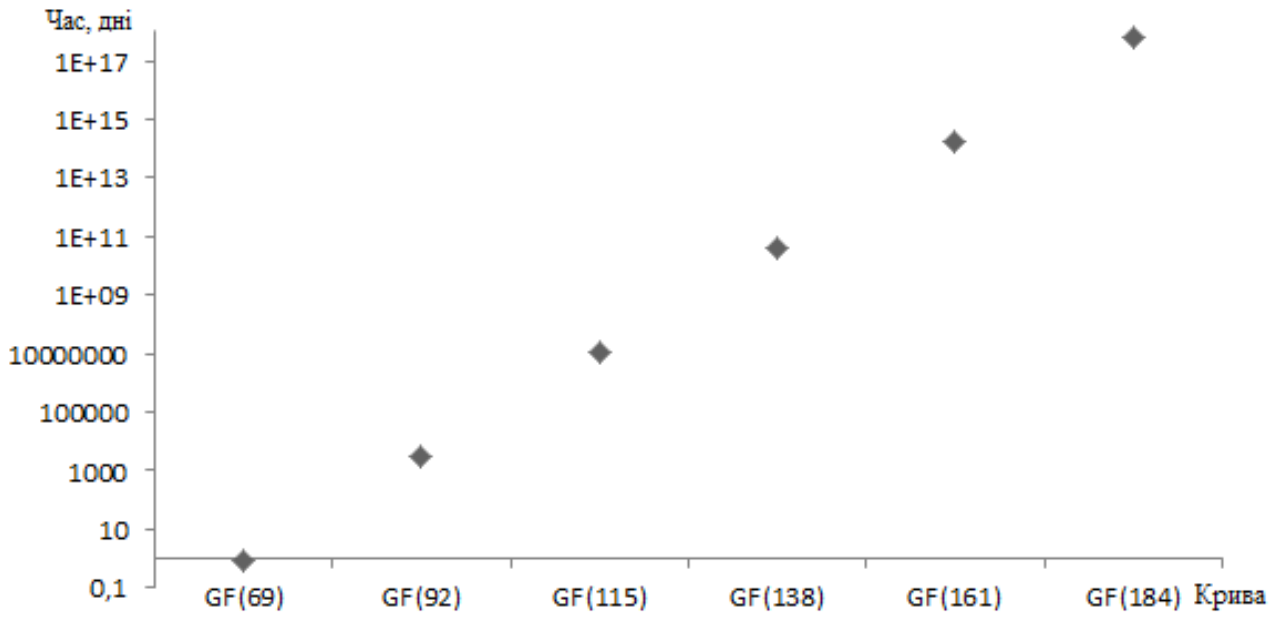

Рисунок 4.7 – Залежність часу, необхідного для розв’язання дискретного логарифма для кривої $GF(p)$: час подано у логарифмічній шкалі

4.4.3 Дослідження процесів в апаратній моделі обчислень дискретного логарифма на основі паралельного ро-методу Полларда.

Апаратно–програмна модель реалізації паралельного ро-методу Полларда передбачає паралельне обчислення багатьох стежок блукання із застосуванням одних і тих же параметрів, бази точок. Єдиним елементом, яким відрізняються окремі компоненти моделі виконання паралельного алгоритму, є стартові параметри, а отже випадково генеровані координати стартової початкової точки та два коефіцієнти. Паралельний запуск декількох компонентів в моделі призводить до лінійного збільшення швидкості обчислення чергових точок на стежках блукання. Отже, в k кроках, які виконуються, налічується J ітерацій, де J – кількість компонентів, що реалізують одну стежку блукання. Таким чином, можна припустити, що очікувана колізія відбувається після виконання k кроків, обчислених з формули:

$$k = \frac{11(4w-2)\sqrt{\pi n/2}}{J}. \quad (4.6)$$

Реалізація алгоритму паралельного блукання полягає на створенні J компонентів, за допомогою яких реалізуються окремі стежки блукання. Для випадку впровадження до обчислень більшої кількості компонентів, що реалізують шляхи блукання, кількість ітерацій алгоритму збільшується пропорційно до кількості впроваджених компонентів. Розглянемо випадок заімплементування моделі, яка дає змогу реалізувати паралельний ро-метод Полларда, на кластері FPGA типу COPACOBANA, особливості будови та функціонування якого наведено в [110]. Він складається зі 120 програмованих матриць FPGA. Рисунок 4.8, що походить з [110], ілюструє згаданий кластер.

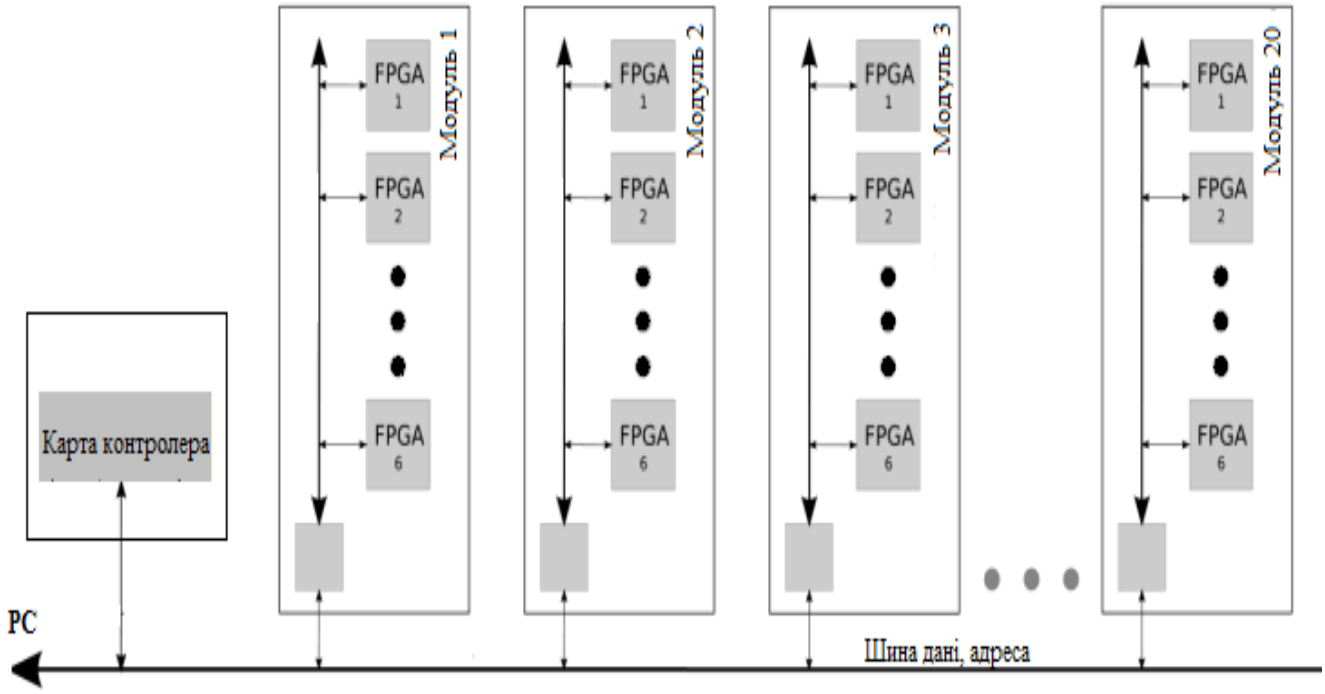

Рис 4.8. Архітектура кластера FPGA типу COPACOBANA

В результаті аналізу проведених досліджень отримано, що для застосованих 120 стежок випадкового блукання, які виконуються паралельно, продуктивність обчислювальної системи, виражена в кількості ітерацій за секунду, для такого рішення зросте до рівня, представленого в таблиці 4.10.

Таблиця 4.10

Продуктивність обчислювальної системи для різних кривих GF (p)
під час реалізації 120 стежок блукання

Крива *GF*(*p*)	69	92	115	138	161	184
Кількість слів	3	4	5	6	7	8
Кількість ітерацій / с	38601398	28235294	22337662	17890909	15047022	12892562

Отримані результати досліджень дали змогу оцінити час, який необхідний для розв'язку дискретного логарифма для кривих *GF*(p) різного розміру при використанні 120 стежок блукання (таблиця 4.11).

Таблиця 4.11

Прогнозний час знаходження дискретного логарифма для різних еліптичних
кривих *GF*(p) при 120 програмованих компонентах

Крива	*GF*(69)	*GF*(92)	*GF*(115)	*GF*(138)	*GF*(161)	*GF*(184)
Час, дні	0,007	27	100790	$3,5*10^{8}$	$1,4*10^{12}$	$4,9*10^{15}$

На рисунку 4.9 показано збільшення часу, необхідного для розв'язання дискретного логарифма за допомогою системи, побудованої із 120 програмованих матриць FPGA, в залежності від розміру ЕК. При цьому час подано у логарифмічній шкалі.

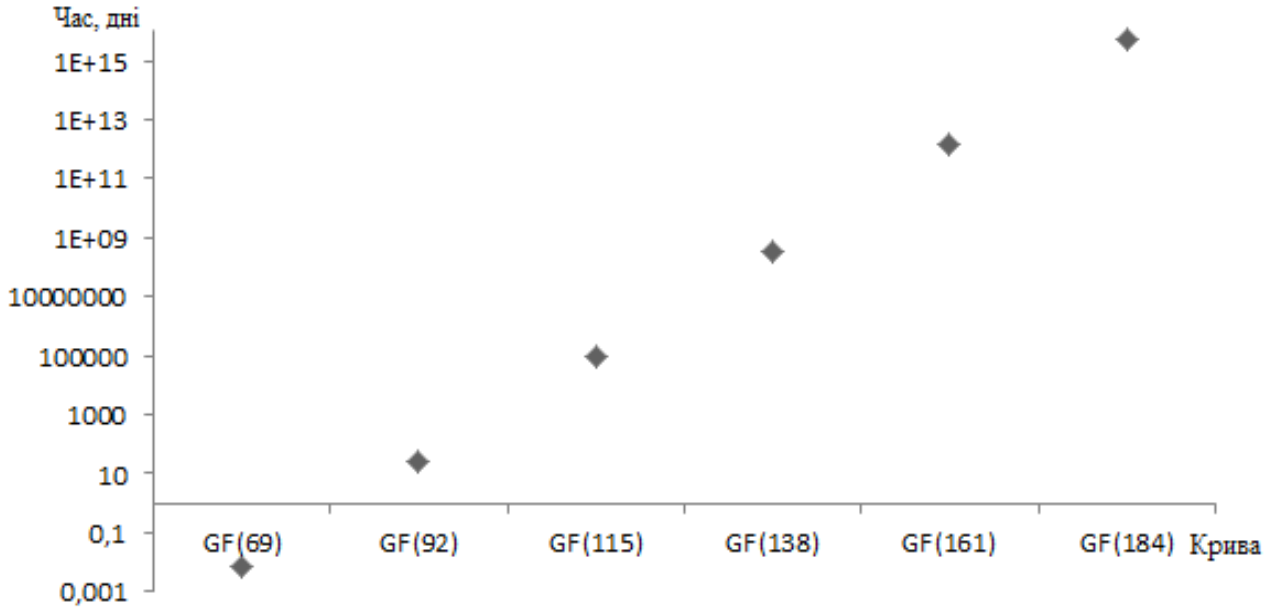

Рисунок 4.9 – Залежність часу, необхідного для розв’язання дискретного логарифма із застосуванням 120 програмованих матриць FPGA, в залежності від розміру кривої $GF(p)$

4.4.4 Дослідження процесів в апаратній моделі реалізації ро-методу Полларда для підтримки системи криптоаналізу.

Симуляцію процесів проведено на побудованій апаратній моделі реалізації ро-методу Полларда. Цю модель застосовано складовим компонентом для підтримки функціонування криптографічної системи, що ґрунтується на програмному рішенні. Програмне рішення в своїй роботі використовує технологію MPI (Message Passing). Операції на еліптичних кривих виконуються з використанням бібліотеки MIRACL (Multiprecision Integer and Rational Arithmetic C/C++ Library). Застосунок побудований таким чином, що передбачає можливість свого запуску в багатьох середовищах. Характерна особливість ро-методу Полларда полягає в тому, що для його реалізації можна використовувати мережеві під’єднання для загальної паралельної роботи над обчисленням дискретного логарифма за допомогою різних географічно віддалених комп’ютерних систем та компонентів. Це рішення успішно використано в різних проектах, не тільки в криптографії. Такий підхід дозволяє ввести в систему апаратне рішення, яке описано раніше. Подібно як для випадку програмного рішення, апаратні

компоненти, що реалізують стежки блукання, можуть бути географічно віддаленими і комунікуватися через мережу. Процеси як у апаратно–програмній, так і програмній моделях симульовано в різних середовищах. Одні з них – це багатопроцесорний суперкомп'ютер типу ALTIX 3700 (АСК Cyfronet), обладнаний процесорами Itanium 2 тактової частоти 1,5 ГГц, та кластерне середовище OpenMosix на основі мікропроцесора Pentium IV 2,8 ГГц. Результати роботи, які представлено в таблиці 4.12, відносяться до кількості ітерацій, необхідних для реалізації одної стежки блукання. Крім цього, в таблиці додатково подано кількість ітерацій для апаратного рішення на базі програмованої матриці FPGA.

Таблиця 4.12

Залежність кількості ітерацій алгоритму для різних кривих GF (p)
в залежності від обчислювальних середовищ

Обчислювальне середовище \ Крива $GF(p)$	69	92	115	138	161	184
Itanium2	109169	67142	53791	47921	42052	32543
Pentium IV	117496	73157	56045	48659	41274	31235
FPGA	321678	235294	186147	149091	125392	107438

Як випливає з даних, наведених в таблиці 4.12, швидкість обчислення на заданій еліптичній кривій при застосуванні апаратно-програмної моделі збільшується принаймні в три рази в порівнянні з обчисленнями, виконаними в програмній моделі. Застосування апаратної моделі обчислень з використанням теоретико-числових базисів Крестенсона дає реальне збільшення швидкості обчислень на еліптичній кривій і значно пришвидшує продуктивність ро-методу Полларда.

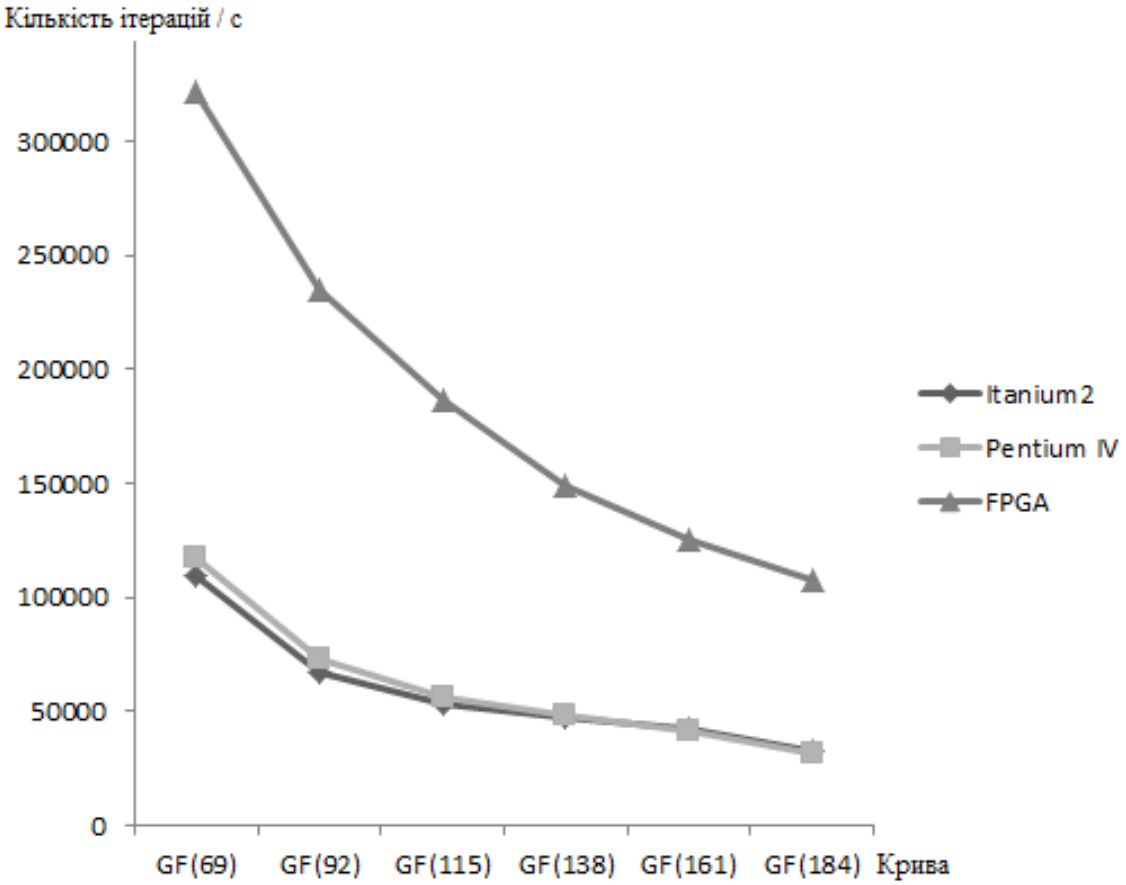

Рисунок 4.10 – Продуктивність ро-методу Полларда для різних реалізацій

Висновки до розділу 4

1. Побудовано суматори точок, основані на теоретико-числових базисах Крестенсона і паралельному додаванні, та виконано симуляцію процесів в суматорах, а також визначено швидкодію сумування точок для кривих різного розміру $GF(p)$, що дало змогу в застосуваннях до криптографічних алгоритмів в ІУСЕК оцінити час шифрування даних відкритим ключем і збільшення продуктивності шифрування. Отримано, що для кривої $GF(163)$ час, необхідний для шифрування інформації розміру 1024 кБ з використанням системи залишків Радемахера-Крестенсона, стоновить 111 с або втричі менший, ніж шифрування тієї ж інформації із застосуванням стандартних алгоритмів.

2. Досліджено та проведено аналіз процесів в системі ІУСЕК, що реалізує ро-метод Полларда, як в базовій, так і паралельній версіях. Симуляцію здійснено на основі програмованої матриці FPGA типу Stratix III EP3SL150F1152I4SL, що дозволило оцінити швидкодію алгоритму для базової та паралельної версій, а також

час, необхідний для розв’язання дискретного логарифма. Дослідження дозволило визначити час, необхідний для розв’язання дискретного логарифму, зокрема в паралельній версії за допомогою 120 програмованих матриць цей час становить 270 років для кривої GF(115), тоді як для кривої GF(138) час збільшується в $3 \cdot 10^3$ разів.

3. На підставі розв’язання дискретного логарифма визначено показники живучості ІУСЕК, завдяки чому розраховано прогнозований час живучості даних в залежності від розміру використаних еліптичних кривих в криптографічних системах. Отримано, що для випадку безпосередньої атаки на ЕК, приміром, криву *GF*(115), ІУСЕК функціонуватиме правильно і виконуватиме свої завдання більше 200 років, в той час як ІУСЕК, в яких застосовуються пристрої, що виконують операції на кривій *GF*(138), живучість зростає в $4{,}5 \cdot 10^3$ разів.

Висновки

У дисертації отримано науково-обґрунтовані результати розв’язання актуального наукового завдання – побудови моделей та засобів підвищення живучості інформаційно-управляючих систем на основі еліптичних кривих, базованих на полях більш високого порядку, що має істотне значення для визначення гарантоздатності систем, вибору потрібних розмірів і типів кривих, збільшення стійкості до дефектів зовнішніх впливів. Найважливіші наукові результати, висновки та рекомендації такі:

1. Базуючись на результатах аналізу принципів побудови, технологічних рішень і напрямів розвитку інформаційно-управляючих систем, доведено необхідність створення моделей та засобів ІУСЕК, що дає змогу забезпечити високі продуктивність і живучість їх функціонування.

2. На підставі ефективних методів розв'язання дискретного логарифма на ЕК розроблено моделі підвищення живучості ІУСЕК з врахуванням дефектів зовнішніх впливів за ознаками ймовірності та детермінованості та технології обчислень у ТЧБ Крестенсона на ЕК *GF(p)* у пристроях для виконання криптографічних операцій із реалізацією ро-методу Полларда, що дало змогу забезпечити низьку тривалість факторизації, верифікацію та визначення розміру ЕК в залежності від часу, протягом якого інформація повинна бути конфіденційною, і в той же мінімізацію обчислювальних затрат та ефективну протидію потенційним загрозам.

3. Виконано оцінювання часу, потрібного для одержання розв’язання дискретного логарифма ро-методом Полларда при застосуванні запропонованих моделей та технологій обчислення, а також визначено середовища, в яких можна використовувати ці підходи, завдяки чому можна вибрати оптимальний розмір ЕК для засобів підвищення живучості ІУСЕК.

4. Удосконалено технологію обчислення добутку за модулем багатобітних чисел з використанням ТЧБ Радемахера-Крестенсона, що дає змогу позбутися операції множення в традиційній формі шляхом заміни операцією

додавання та, на підставі поділу чисел на слова потрібної довжини, усунути обмеження щодо максимальної довжини чисел, які додаються, у засобах підвищення живучості ІУСЕК.

5. Модифіковано модель обчислення точок на ЕК *GF*(p), впроваджуючи удосконалену технологію обчислення добутку та сумування чисел, змішане представлення точок, що дозволило кількість необхідних операцій обчислення добутку зменшити до значення 11, уникаючи додатково потреби знаходження оберненого елемента в полі у засобах підвищення живучості ІУСЕК.

6. Удосконалено технологію обчислень для реалізації паралельного ро-методу Полларда для ЕК вищих порядків *GF*(p) на основі модифікованої моделі суматора точок в ІУСЕК, що забезпечило суттєве зменшення часу, необхідного для розв’язання дискретного логарифма.

7. Створено дві апаратно-програмні системи для ІУСЕК:

а) перша з них працює на програмованих матрицях FPGA та комп’ютерах класу ПК, дія якої ґрунтується на основі модифікованої моделі обчислень, причому на підставі досліджень побудованого суматора точок на програмованих матрицях FPGA Stratix III EP3SL150F1152I4SL одержано збільшення швидкості шифрування методом Ель-Гамаля приблизно втричі в порівнянні до традиційного підходу сумування точок;

б) друга з них реалізує ро-алгоритм Полларда, використовуючи технологію обчислень на ТЧБ Крестенсона, та ґрунтується на аналогічних апаратних засобах. За результатами дослідження роботи паралельної системи на декількох програмованих матрицях FPGA, дослідження її роботи для розв’язання логарифма для ЕК різного розміру та проведених вимірювань визначено, що застосування поодинокої стежки алгоритму, який реалізовано на одній програмованій матриці FPGA, зумовлює трикратне збільшення швидкості роботи алгоритму в порівнянні з системою, побудованою на базі процесора Itanium 2.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безопасность атомных станций: Информационные и управляющие системы / М. А. Ястребенецкий, В.Н. Васильченко, С. В. Виноградская [и др.]; Под ред. М.А. Ястребенецкого. – К.: Техніка, 2004. – 472 с.
2. Березюк Н.Т. Живучесть микропроцессорных систем управления / Н. Т. Березюк, А. Я. Гапунин, Н. И. Подлесный. – К.: Тэхника, 1989. – 143 с. – ISBN 5-335-00245-Х.
3. Галузева система управління якістю. Гарантоздатність програмно-технічних комплексів критичного призначення: СОУ-Н НКАУ 0060:2010. – [Чинний від 2010-04-01]. – К.: НКАУ, 2010. – 60 с. – (Галузевий стандарт України).
4. Голуб С. В. Методологія створення автоматизованих систем багаторівневого соціоекологічного моніторингу : автореф. дис на здобуття наук. ступеня докт. техн. наук: спец. 05.13.06 “Інформаційні технології” / Голуб Сергій Васильович ; Київ, Інститут проблем математичних машин і систем НАН України. – Київ, 2008. – 35 с.
5. Горбенко И. Д. Классы сложностей алгоритмов на основе билинейных отображений / И. Д. Горбенко, К. А. Погребняк // Радіоелектронні і комп’ютерні системи. – 2007. – № 7 (25). – С. 125-128.
6. Горбенко І. Д. Прикладна криптологія / І. Д. Горбенко, Ю. І. Горбенко // Харків: Форт, 2012. – 867 с.
7. Горбенко Ю. І. Інфраструктура відкритих ключів. Електронний цифровий підпис. Теорія та практика / Ю. І. Горбенко, І. Д. Горбенко. – Харків: Форт, 2010. – 593 с.
8. Дудикевич В. Б. Концептуальні моделі захисту інформації для технологій стаціонарного, стільникового, супутникового зв’язку / В. Б. Дудикевич, Ю. Р. Гарасим, Г. М. Микитин // Вісник Національного університету “Львівська політехніка”: Автоматика, вимірювання та керування. – 2010. – № 665. – С 18-26.

9. Задирака В. К. Анализ стойкости криптографических и стеганографических систем на основе общей теории оптимальных алгоритмов [Electronic resource] / В. К. Задирака, А. М. Кудин // Journal of Qafgaz University Mathematics and Computer Science. – 2010. – № 2. – С. 47-49. – Режим доступу : http://journal.qu.edu.az
10. Згуровский М. З. Принятие решений в сетевых системах с ограниченными ресурсами / М. З. Згуровский, А. А. Павлов; Нац. техн. ун-т "Киев. политехн. ин-т". – К.: Наукова думка, 2010. - 575 с. – ISBN 978-966-00-1040-6.
11. Інформаційні технології. Методи захисту. Криптографічні перетворення, що ґрунтуються на еліптичних кривих: Частина 3 – Встановлення ключів: ДСТУ ISO/IEC 15946-3:2006. – [Чинний від 2008-01-01]. – К.: Держспоживстандарт України, 2006. – 34 с. – (Національний стандарт України).
12. Карпінський М. Показники оцінки ефективності алгоритмів шифрування на еліптичних кривих / М. Карпінський, І. Васильцов, І. Якименко // Науково-технічний збірник «Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні». – К.: НТУУ «КПІ». – 2004. – Вип. 8. – С. 121–124.
13. Комп’ютерні мережі / Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. – Львів: Магнолія, 2013. – 256 с. – ISBN 978-617-574-087-3.
14. Комп’ютерні технології криптографічного захисту інформації на спеціальних цифрових носіях. Навчальний посібник / В. К. Задірака, А. М. Кудін, В. О. Людвиченко, О. С. Олексюк. – Київ-Тернопіль: Підручники і посібники, 2007. – 272 с.
15. Корченко О. Г. Метод перетворення еталонів параметрів для систем аналізу і оцінювання ризиків інформаційної безпеки / О. Г. Корченко, С. В. Казмірчук, А. Ю. Гололобов // Захист інформації. – 2013. – Том 15, № 4. – С. 359-366.
16. Оптимальні алгоритми обчислення інтегралів від швидкоосцилюючих функцій та їх застосування. Том 1. Алгоритми / Сергієнко І.В., Задірака В.К., Литвин О.М. [та ін.]. – К.: Наук. думка, 2011. – 448 с.

17. Оптимальні алгоритми обчислення інтегралів від швидкоосцилюючих функцій та їх застосування. Том 2. Застосування / Сергієнко І.В., Задірака В.К., Литвин О.М. [та ін.]. – К.: Наук. думка, 2011. – 348 с.
18. Отказобезопасные информационно-управляющие системы на программируемой логике / Е. С. Бахмач, А. Д. Герасименко, В. А. Головир [и др.]; под ред. В.С. Харченко, В.В. Скляра. – Х.: Национальный аэрокосмический университет «ХАИ»; Кировоград: НПО «Радий», 2008. – 380 с.
19. Параллельные вычисления / А. Корченко, Г. Жангисина, С. Гнатюк, А. Шайханова // Ізденіс – Поиск. – 2013. – № 2 (3). – С. 195-198. – ISSN 1560-1730.
20. Потій О. В. Аналіз систем показників безпеки інформації / О. В. Потій, Д. Ю. Пилипенко // Прикладная радиоэлектроника. – 2010. – Том 9, № 3. – С. 435-443.
21. Потій О. В. Формалізована модель діяльності із захисту інформації / О. В. Потій // Радіоелектронні і комп'ютерні системи. – 2007. – № 6 (25). – С. 96-102.
22. Сербін В. Г. Визначення і формалізація основних показників гарантоздатності живучих комп’ютерних систем керування на основі ймовірнісно-фізичного підходу для їх проектної оцінки і прогнозування / В. Г. Сербін, А. І. Сухомлин // Математичні машини і системи. – 2012. – № 4. – С. 182-189.
23. Сергієнко І. В. Алгебраїчні аспекти інформаційних технологій. Частина 1 / І. В. Сергієнко, С. Л. Кривий, О. І. Провотар. – К.: Наук. думка, 2011. – 400 с.
24. Системи менеджменту інформаційної безпеки / Ромака В. А., Дудикевич В. Б., Гарасим Ю. Р. [та ін.]. – Львів: Вид-во Львівської політехніки, 2012. – 232 с.
25. Технологія багатофункціональної обробки і передачі інформації в моніторингових мережах / Б. М. Шевчук, В. К. Задірака, Л. О. Гнатів, С. В. Фраєр. – К.: Наук. думка, 2010. – 370 с.
26. Черкесов Г. Н. Методы и модели оценки живучести сложных систем / Г. Н. Черкесов. – М.: Знание, 1987. – 32 с.

27. Юдін О. К. Захист інформації в мережах передачі даних / О. К. Юдін, О.Г. Корченко, Г.Ф. Конахович. – К.: Інтерсервіс, 2009. – 714 с. – ISBN 978-966-97108-6-4.
28. A New Model for Error-Tolerant Side-Channel Cube Attacks / Z. Li, B. Zhang, J. Fan, I. Verbauwhede // Cryptographic Hardware and Embedded Systems - CHES 2013. LNCS. – 2013. – Vol. 8086. – P. 453-470.
29. Adikari J. Hybrid Binary-Ternary Number System for Elliptic Curve Cryptosystems / J. Adikari, V. Dimitrov, L. Imbert // IEEE transactions on computers. – 2011. – Vol. 60, No 2. – Режим доступу : http://www.lirmm.fr/~imbert/pdfs/hybrid_ieeetc_2011.pdf
30. Aleksander M. Bezpieczeństwo kryptosystemów opartych na krzywych eliptycznych / M. Aleksander, G. Litawa // Bezpieczeństwo informacji / M. Karpiński. – Warszawa: Wydawnictwo Pomiary Automatyka Kontrola, 2012. – 280 s. – Rozd. 5. – S. 183-196. – ISBN 978-83-930505-3-6.
31. Aleksander M. Calculation of GF (p) Elliptic Curves in FPGA / M. Aleksander, M. Karpinskyy, G. Litawa // Computing. – 2011. – Vol. 10. – Issue 2. – P. 91-96. – ISSN 1727-6209.
32. Aleksander M. Functional safety and survivability of information control elliptic-curve-based systems: models and methods / M. Aleksander, M. Karpiński, G. Litawa // Ukrainian Scientific Journal of Information Security. – 2013. – Vol. 19, issue 1. – P. 51-55. – ISSN: 2225-5036.
33. Aleksander M. Implementation and testing of methods parallel computation on elliptic curves GF(p) / M. Aleksander, M. Karpinski, G. Litawa // Scientific Journal of the East Ukrainian National University. –2011. – No 7 [161], Part 1. – P. 304-310. – ISSN 1998-7927.
34. Aleksander M. Implementation in FPGA of Computations on Elliptic Curves GF(p) based on Rademacher–Krestenson's Bases / M. Aleksander, M. Karpinski, G. Litawa // Informative safety. – 2012. – No 1 (7). – P. 12-17. – ISSN 2224-9613.

35. Amounas F. Proposed Developements of Blind Signature Scheme Based on ECC / F. Amounas, E.H. El Kinani // Computer Engineering and Applications. – 2013. – Vol. 2, No. 1. – P. 151-160.
36. An Elliptic Curve Cryptosystem Design Based on FPGA Pipeline Folding [Electronic resource] / O. Al-Khaleel, C. Papachristou, F. Wolff, K. Pekmestzi // IOLTS '07 Proceedings of the 13th IEEE International On-Line Testing Symposium IEEE Computer Society Washington, DC, USA. – 2007. – Режим доступу : http://bear.ces.cwru.edu/Recent_Papers /iolts07.pdf.
37. Bailey D. V. Breaking elliptic curve cryptosystems using reconfigurable hardware / D. V. Bailey, L. Batina, T. Guneysu // Field Programmable Logic and Applications, 2010 International Conference. – 2010. – P. 133–138.
38. Bernstein D. J. Explicit-formulas database [Electronic resource] / D.J. Bernstein, T. Lange. – Режим доступу : http://www.hyperelliptic.org/EFD/. 2013.
39. Bezpieczeństwo bezprzewodowych sieci spontanicznych i sensorowych / M. Aleksander, J. Kinach, G. Litawa [et al.] // Bezpieczeństwo informacji / M. Karpiński. – Warszawa: Wydawnictwo Pomiary Automatyka Kontrola, 2012. – 280 s. – Rozd. 2. – S. 82-129. – ISBN 978-83-930505-3-6.
40. Bissona G. Computing the endomorphism ring of an ordinary elliptic curve over a finite field / G. Bissona, A. V. Sutherlandb // Journal of Number Theory. – 2013. – Volume 131, Issue 5. – P. 815–831.
41. Blade I. Krzywe eliptyczne w kryptografi / I. Blade, G. Seroussi, N. Smart. – Warszawa: TAO, 2004. – 239 p.
42. Bos J. W. Elliptic Curve Cryptography in Practice / J. W. Bos, J. A. Halderman, N. Heninger [et al.] // Microsoft Research. – 2013. – November. – 16 p.
43. Bulens P. Hardware for Collision Search on Elliptic Curve over GF(2^m) [Electronic resource] / P. Bulens, G. Meurice, J. J. Quisquater. – 2006. – Режим доступу : http://www.hyperelliptic.org/tanja/SHARCS/talks06/bulens.pdf.
44. Calculation of Multiplication, Using Walsh Transform / A. N. Tereshchenko, S. S. Melnikova, L. A. Hnativ [et al.] // Journal of Automation and Information Sciences. – 2010. – Vol. 42, No 4. – P. 37-65.

45. Chakraborty K. A Stamped Blind Signature Scheme based on Elliptic Curve Discrete Logarithm Problem / K. Chakraborty, J. Mehta // International Journal of Network Security. – 2012. – Vol.14, No.6. – P. 316-319.
46. Chen J. On-Demand Security Architecture for Cloud Computing / J. Chen, Y. Wang, X. Wang // Computer. – 2012. – Vol. 45, No 7. – P. 73-78. – ISSN 0018-9162.
47. Chocianowicz W. Kryptologia i zaawansowane metody kryptografii [Electronic resource] / W. Chocianowicz. – Режим доступу : http://uznam.net.pl/~blondasek/iuz/materialy/ Krypto-2007-2008-2MUDZ.pdf.
48. Cohen A. GPU accelerated elliptic curve cryptography in GF(2^m) / A. Cohen, K. Parhi // IEEE International Midwest Symposium on Circuits and Systems. – 2010. – P. 57-60.
49. Costello C. Faster compact Diffie–Hellman: Endomorphisms on the *x*-line advances in cryptology / C. Costello, H. Hisil, B. Smith // EUROCRYPT 2014. LNCS. – 2014. – Vol. 8441. – P. 183-200.
50. Coupled fpga/asic implementation of elliptic curve crypto-processor / M. Machhout, Z. Guitouni, K. Torki [et al.] // International Journal of Network Security & Its Applications. – 2010. – Vol. 2, No 2. – P. 100-112.
51. Cryptographic system security level based on elliptic curves / M. Karpinski, M. Aleksander, G. Litawa, V. Karpinskyi // Scientific Journal of the East Ukrainian National University. – 2008. – No 8 (126), Issue 1. – P. 94-98.
52. Daly A. Fast modular inversion in the Montgomery domain on reconfigurable logic / A. Daly, L. Marnaney, E. Popovici // Technical report. – Cork, Ireland: University College Cork, 2004. – P. 362-367.
53. Diem C. On the discrete logarithm problem in class groups of curves / C. Diem // Math. Comp. – 2011. No 80. – P. 443-475.
54. Diem C. On the discrete logarithm problem in elliptic curves / C. Diem // Compos. Math. – 2011. – No 147(1). – P. 75–104.
55. Dimitrov V. S. Two Algorithms for Modular Exponentiation Based on Nonstandard Arithmetics [Electronic resource] / V. S. Dimitrov, T. V. Cooklev // IEICE Trans. Fundamentals of Electronics, Comm. and Computer Science. – 1995. – Vol. E78-A,

No 1, special issue on cryptography and information security. – P. 82-87. – Режим доступу : http://eprint.iacr.org/2008/285.pdf.

56. Distributed computing system which solve an elliptic curve discrete logarithm problem / Aleksander M., Litawa G., Karpinskyi V. // The Experience of Designing and Application of CAD Systems in Microelectronics : Xth International Conference CADSM 2009, 24-28 February 2009 : Proceedings of the Conference. – Lviv-Polyana, Ukraine: Publishing House Vezha&Co, 2009. – P. 378-380. – ISBN 978-966-2191-05-9.
57. Fast Cryptography in Genus 2 / J. W. Bos, C. Costello, H. Hisil, K. Lauter // EUROCRYPT 2013. LNCS. – Vol. 7881. – Heidelberg: Springer. – 2013. – P. 194–210.
58. Faure C. Cryptanalysis of the McEliece cryptosystem over hyperelliptic curves / C. Faure, L. Minder // Proceedings of the Eleventh International Workshop on Algebraic and Combinatorial Coding Theory. – 2008. – P. 99–107.
59. Fichtenholz G. M. Rachunek różniczkowy i całkowy. Tom 2. – Warszawa: Wydawnictwo Naukowe PWN, 2011. – 696 s.
60. Fpga-oriented secure data path design: Implementation of a public key coprocessor / N. Mentens, K. Sakiyama, L. Batina [et al.] // IEEE Field Programmable Logic and Applications. – 2006. – P. 133-138.
61. Frey G. A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves / G. Frey, H. G. Ruck // Math. Comp. – 1994. – No 62. – P. 865-874.
62. Galbraith S. D. Endomorphisms for faster elliptic curve cryptography on a large class of curves / S. D. Galbraith, X. Lin, M. Scott // Journal of Cryptology. – 2011. – July, Vol. 24, Issue 3. – P. 446-469.
63. Galbraith S. D. Using Equivalence Classes to Accelerate Solving the Discrete Logarithm Problem in a Short Interval / S. D. Galbraith, R. S. Ruprai // PKC 2010. LNCS. – 2010. – Vol. 6056. – P. 368-383.

64. Gampala V. Data Security in Cloud Computing with Elliptic Curve Cryptography / V. Gampala, S. Inuganti, S. Muppidi // IJSCE. – 2012. – Vol. 2, Issue 3. – P. 138-141. – ISSN 2231-2307.
65. Guneysu T. E. Efficient hardware architectures for solving the discrete logarithm problem on elliptic curves [Electronic resource] / T. E. Guneysu // Horst Gortz Institute, Ruhr University of Bochum. February 2006. – P. 119. – Режим доступу : http://scribd.com/doc/164166903/Thesis-Gueneysu-Mppr.
66. Guneysu T. On the Security of Elliptic Curve Cryptosystems against Attacks with Special-Purpose Hardware [Electronic resource] / T. Guneysu, Ch. Paar, J. Pelzl // SHARCS'06. – 2006. – P. 20. – Режим доступу : http://www.hyperelliptic.org/tanja/ SHARCS/talks06/ecc_rub.pdf.
67. Hankerson D. Elliptic Curve Discrete Logarithm Problem / D. Hankerson, A. Menezez // Encyclopedia of Cryptography and Security. – 2011. – P. 397-400.
68. Hankerson D. Guide to elliptic curve cryptography [Electronic resource] / D. Hankerson, A. Menezes, S. Vanstone. – NY: Springer, 2004. – 332 p. – Режим доступу : http://math.boisestate.edu/~liljanab/Crypto2Spring10 /GuideToECC.pdf.
69. Joux A. Cover and Decomposition Index Calculus on Elliptic Curves made practical: Application to a seemingly secure curve over F_p^6 / A. Joux, V. Vitse // Eurocrypt 2012. LNCS. – 2012. – Vol. 7237. – P. 9-26.
70. Joux A. Elliptic curve discrete logarithm problem over small degree extension fields / A. Joux, V. Vitse // Journal of Cryptology. – 2013. – January, Vol. 26, Issue 1. – P. 119-143.
71. Karaklajic D. Hardware Designer's Guide to Fault Attacks / D. Karaklajic, J. M. Schmidt, I. Verbauwhede // VLSI Systems, IEEE Transactions on. – 2013. – Vol. 21, Issue 12. – P. 2295-2306.
72. Karpiński M. Bezpieczeństwo przekazu informacji w sieciach oparte na metodach szyfrowania bazujących na krzywych eliptycznych / M. Karpiński, M. Aleksander, G. Litawa // III Międzynarodowa Konferencja Naukowa z cyklu "Informatyka w dobie XXI wieku". – 2009. – S. 27-30.

73. Karpinski M. Information Security / M. Karpinski. – Warsaw: Measurements, Automation and Monitoring, 2012. – 280 p. – ISBN 978-83-930505-3-6. (in Polish).
74. Karpinski M. The Security of Data Transmission over Telecommunication Networks Based on Advanced Data Encryption Methods / M. Karpinski, M. Aleksander, G. Litawa, V. Karpinskyi // Proceedings of the 9th International Workshop “Computational Problems of Electrical Engineering” (CPEE’08) (September 16-20, 2008, Alushta (Crimea), Ukraine). – P. 71-73.
75. Karpiński W. Rozwiązanie logarytmu dyskretnego metodą Rho-Pollarda w oparciu o biblioteki MPI2 i MIRACL / W. Karpiński, G. Litawa, I. Jakymenko // Materiały XXII Konferencji Naukowej Tarnopolskiego Państwowego Uniwersytetu Technicznego im. I. Puluja. – Tarnopol: Wyd. TPUT, 2008. – S. 93.
76. Kasper E. Fast Elliptic Curve Cryptography in OpenSSL / E. Käsper // Financial Cryptography and Data Security. LNCS. – 2012. – Vol. 7126. – P. 27-39.
77. Kharchenko V. S. Analysis of the problems of safeware engineering: the project TEMPUS-SAFEGUARD / V. S. Kharchenko // Radioelectronic and Computer Systems. – 2010. – No 48. – P. 297-300. (in Ukrainian).
78. Kim S. Fixed argument pairing inversion on elliptic curves [Electronic resource] / S. Kim, J. H. Cheon // Cryptology ePrint Archive, Report 2012/657. – 2012. – 10 p. – Режим доступу : http://eprint.iacr.org/.
79. Lagendijk R. Encrypted signal processing for privacy protection / R. (Inald) L. Lagendijk, Z. Erkin, M. Barni // IEEE Signal Processing Magazine. – 2013. – Vol. 30, No 1. – P. 82-105. – ISSN 1053-5888.
80. Lambda Coordinates for Binary Elliptic Curves / T. Oliveira, J. Lopez, D. F. Aranha, F. Rodriguez-Henriquez // Cryptographic Hardware and Embedded Systems - CHES 2013. LNCS. – 2013. – Vol. 8086. – P. 311-330.
81. Lashermes R. Inverting the Final Exponentiation of Tate Pairings on Ordinary Elliptic Curves Using Faults / R. Lashermes, J. Fournier, L. Goubin // CHES 2013. LNCS. –2013. – Vol. 8086. – P. 365–382.
82. Leveson N. G. Safeware: System Safety and Computers / N.G. Leveson. – Reading, Massachusetts: Addison-Wesley, 1995. – 680 p.

83. Litawa G. An Elliptic Curve Points Calculation Method with Rademacher–Krestenson's Bases / G. Litawa // Scientific Journal of the Ternopil National Technical University. – 2012. – Vol. 66, No 2. – P. 207-213.
84. Litawa G. Zaawansowana kryptoanaliza szyfrów opartych na krzywych eliptycznych z zastosowaniem układów programowalnych FPGA / G. Litawa, W. Karpiński // Prace naukowe Instytutu Technicznego Państwowej Wyższej Szkoły Zawodowej w Nowym Sączu. – Praca zbiorowa pod red. dr inż. Marka Aleksandra. – Nowy Sącz: Wydawca Państwowa Wyższa Szkoła Zawodowa w Nowym Sączu, 2007. – S. 175-185.
85. Lopez J. Fast Multiplication on elliptic curves over GF(2^m) without precomputation / J. Lopez, R. Dahab // Lecture Notes in Computer Science (LNCS). – 1999. – Vol. 1717. – P. 316-327.
86. Makoha A. H. The arithmetic of large integers in parallel computer systems [Electronic resource] / A. H. Makoha, B. U. Zuj/ – 2007. – Режим доступу : http://revolution.allbest.ru/mathematics/00011260_0.html (In Russian).
87. Menezes A. J. Handbook of Applied Cryptography / A. J. Menezes, P. C. Oorschot, S. A. Vanstone. – New York: CRC Press, 1996. – 816 p.
88. Menezes A. J. Reducing elliptic curve logarithms to finite field / A. J. Menezes, T. Okamoto, S.A. Vanstone // IEEE Trans, Info, Theory. – 1993. No 39. – P. 1639-1646.
89. Missala T. Bezpieczeństwo funkcjonalne – awers i rewers [Electronic resource] / T. Missala // Pomiary Automatyka Robotyka. – 2008. – Nr 1. – S. 12-17. – Режим доступу : http://ebookbrowse.com/piap-publikacja-5-pdf-d213304465.
90. Oorschot P. C. Parallel collision search with cryptanalytic applications / P. C. Oorschot, M. J. Wiener // Journal of Cryptology. – 1999. – No 12. – P. 1–28.
91. Oppliger R. Security and Privacy in an Online World / R. Oppliger // Computer. – 2011. – Vol. 44, No 9. – P. 21-22. – ISSN 0018-9162.
92. Polish LONUSERS® Group: Technologia SafetyLon w systemach związanych z bezpieczeństwem funkcjonalnym [Electronic resource]. – 2008. – Режим доступу:

http://www.plug.org.pl/pdf/SafetyLon/3%20Podstawy%20bezpieczenstwa%20funkcjonalnego.pdf.

93. Pollard J. M. Monte Carlo methods for index computation (mod p) / J. M. Pollard // Math. Comp. – 1978. – No 32. – P. 918-924.
94. Quisquater J. J. How easy is collision search? Application to DES / J. J. Quisquater, J. P. Delescaille // Advances in cryptology, EUROCRYPT 89. LNCS. – 1990. – Vol. 434. – P. 429-434.
95. Realizacja jednostki wspomagającej kryptoanalizę szyfrów opartych na krzywych eliptycznych w strukturach reprogramowalnych / P. Majkowski, T. Wojciechowski, M. Wojdyński, M. Rawski // Pomiary Automatyka Kontrola. – 2007. – Vol. 53, Nr 7. – S. 24-26.
96. Rodriguez-Henriquez F. On Fully Parallel Karatsuba Multipliers for GF(2^m) / F. Rodriguez-Henriquez, C. K. Koc // Computer Science and Technology. – 2003. – P. 405-410.
97. Romankevych V. O. Functional safety evaluation for the reconfigurable fault-tolerant multiprocessor control systems / V. O. Romankevych, M. S. Milad, S. O. Poleschuk // Applied Mathematics and Computing – AMC-2011: III Scientific Conference, April 13-15, 2011: Proceedings of the Conference. – Kiev: National Technical University of Ukraine „Kyiv Polytechnic Institute", 2011. – P. 157-161. (in Ukrainian).
98. SCADA Systems: Challenges for Forensic Investigators / I. Ahmed, S. Obermeier, M. Naedele, G. G. Richard // Computer. – 2012. – Vol. 45, No 12. – P. 73-78. – ISSN 0018-9162.
99. Semaev I. A. Evaluation of discrete logarithms on some alliptic curves / I. A. Semaev // Math. Comp. – 1998. – No 67. – P. 353-356.
100. Serial multiplier architectures over GF(2^n)for elliptic curve cryptosystems [Electronic resource] / L. Batina, N. Mentens, S. Ors, B. Preneel // Electrotechnical Conference MELECON 2004. – 2004. – P. 779-782. – Режим доступу : http://www.cosic.esat.kuleuven.be/ publications/article-30.pdf.

101. Silverman J. H. Advanced topics in the arithmetic of elliptic curves / J. H. Silverman. – NY: Springer-Verlag, 1994. – 528 p.

102. Solinas J. A. An improved algorithm for arithmetic on a family of elliptic curves / J. A. Solinas // Advances in cryptology, CRYPTO 97. LNCS. – 1997. – Vol. 1294. – P. 357-371.

103. Solving a 112–bit prime elliptic curve discrete logarithm problem on game consoles using sloppy reduction / J. W. Bos, M. E. Kaihara, T. Kleinjung [et al.] // International Journal of Applied Cryptography. – 2012. – Vol. 2, Issue 3. – P. 212-228.

104. Solving a Discrete logarithm problem with auxiliary input on a 160-bit elliptic curve / Y. Sakemi, G. Hanaoka, T. Izu [et al.] // Public Key Cryptography – PKC 2012. LNCS. – 2012. – Vol. 7293. – P. 595-608.

105. Solving a DLP with Auxiliary Input with the ρ-Algorithm / Y. Sakemi, T. Izu, M. Takenaka, M. Yasuda // In: Jung, S., Yung, M. (eds.); WISA 2011. LNCS. – 2012. – Vol. 7115. – P. 98-108.

106. Solving DLP with Auxiliary Input over an Elliptic Curve Used in TinyTate Library / Sakemi Y., Izu T., Takenaka M., Yasuda M // Ardagna, C.A., Zhou, J. (eds.); WISTP 2011. LNCS. – 2011. – Vol. 6633. – P. 116-127.

107. Survivable Network Systems: An Emerging Discipline [Electronic resource] / R. J. Ellison, D. A. Fisher, R. C. Linger [et al.] // Carnegie Mellon University, 1997. – 38 p. – Режим доступу : http://www.dtic.mil/ dtic/tr/fulltext/u2/a341963.pdf

108. Symmetrized Summation Polynomials: Using Small Order Torsion Points to Speed Up Elliptic Curve Index Calculus Advances in Cryptology/ J. C. Faugcre, L. Huot, A. Joux [et al.] // EUROCRYPT 2014. LNCS. – 2014. – Vol. 8441. – P. 40-57.

109. The security of data transmission over telecommunication networks based on advanced data encryption methods / M. Karpinski, M. Aleksander, G. Litawa, V. Karpinskyi // Electrical Review. – 2009. – No 4. – P. 19-21. – ISSN 0033-2097.

110. Three years of evolution cryptanalysis with copacobana [Electronic resource] / T. Guneysu, G. Pfeiffer, C. Paar, M. Schimmler // SHARCS '09. – 2009. – Режим доступу : http://www.hyperelliptic.org/tanja/SHARCS/record2.pdf

111. Ugus O. Performance of additive homomorphic ec-elgamal encryption for TinyPEDS [Electronic resource] / O. Ugus, A. Hessler, D. Westhoff // Technical Report, 6. Fachgespräch "Drahtlose Sensornetze", July 2007. – 2007. – Режим доступу : http://www.ist-ubisecsens.org/publications/EcElgamal-UgHesWest.pdf.

112. Unterluggauer T. Efficient Pairings and ECC for Embedded Systems / T. Unterluggauer, E. Wenger // Cryptographic Hardware and Embedded Systems – CHES 2014. LNCS. – 2014. – Vol. 8731. – P. 298-315.

113. Using symmetries in the index calculus for elliptic curves discrete logarithm / J. C. Faugere , P. Gaudry, L. Huot, G. Renault // Journal of Cryptology. – 2013. – P. 1-41.

114. Vickie R. A definition for information system survivability [Electronic resource] / R. Vickie, A. Westmark // IEEE Computer Society Washington. – 2004. – 10 p. – Режим доступу : http://dl.acm.org/citation.cfm?id=962757.963306.

115. Wenger E. A hardware processor supporting elliptic curve cryptography for less than 9 kGEs / E. Wenger, M. Hutter // Smart Card Research and Advanced Applications. LNCS. – 2011. – Vol. 7079. – P. 182-198.

116. What about Vulnerability to a Fault Attack of the Miller’s algorithm During an identity based protocol? / N. El Mrabet, J. H. Park, H. H. Chen [et al.] // ISA 2009. LNCS. – 2009. – Vol. 5576. – P. 122–134.

117. Yakymenko I. Matrix algorithms of processing of the information flow in computer systems based on theoretical and numerical Krestenson’s basis / I. Yakymenko, M. Kasyanchuk, Y. Nykolajchuk // TCSET'2010, February 23-27 2010, Lviv-Slavske, Ukraine. – 2010. – P. 241.

118. Zadiraka V. K. Using reserves for computation optimization to improve the integration of rapidly oscillating functions / V. K. Zadiraka, S. S. Melnikova, L.

V. Luts // Cybernetics and Systems Analysis. – 2011. – Vol. 47, No 4. – P. 613-630.

119. Zhang F. Speeding up elliptic curve discrete logarithm computations with point halving / F. Zhang, P. Wang // Designs, Codes and Cryptography. – 2013. – Vol. 67, Issue 2. – P. 197-208.

ДОДАТКИ

Додаток А.

Акти впровадження результатів дисертаційної роботи

Акт впровадження результатів дисертаційної роботи в навчальний процес кафедри КН ТНТУ ім. І. Пулюя.

„ЗАТВЕРДЖУЮ"

Проректор з науково-педагогічної роботи
Тернопільського національного технічного університету імені Івана Пулюя

_____ доц. С.Ф. Лячук

„11" 06 2013 р.

АКТ ВПРОВАДЖЕННЯ

1. ***Об'єкт впровадження:*** засоби функціональної безпеки та живучості інформаційно-управляючих систем на основі еліптичних кривих.

2. ***Ким запропоновано, виконавці, адреса:*** Гжегож Літаш, кафедра комп'ютерних наук (КП) Тернопільського національного технічного університету ім. І. Пулюя, 16001, м. Тернопіль, вул. Руська, 56.

3. ***Джерело інформації:*** матеріали дисертації Гжегожа Літаш, спрямованої на розроблення моделей та методів функціональної безпеки та живучості інформаційно-управляючих систем на основі еліптичних кривих і поданої на здобуття наукового ступеня кандидата технічних наук (спеціальність: 05.13.06 – інформаційні технології).

4. ***Де впроваджено:*** на кафедрі КН Тернопільського національного технічного університету ім. І. Пулюя, 46001, м. Тернопіль, вул. Руська, 56.

5. ***Термін впровадження:*** грудень 2012 р. — травень 2013 р.

6. ***Висновок по впровадженню:*** дані засоби призначені для забезпечення функціональної безпеки та живучості інформаційно-управляючих систем на основі еліптичних кривих. Запропоновані моделі та методи використано в навчальному процесі на кафедрі КП при проведенні лабораторних робіт з дисципліни „Технології захисту інформації", "Інформаційна безпека" в лабораторії комп'ютерних наук, курсовому, дипломному проектуванні та при написанні кваліфікаційних робіт бакалаврів, за навчальними планами з напрямів підготовки 6.050101 „Комп'ютерні науки" та 6.170101 "Безпека інформаційних та комунікаційних систем".

Завідувач кафедри КН,
д.т.н., професор М.В. Приймак

Акт впровадження результатів дисертаційної роботи в навчальний процес технічного інституту ДВТШ в Новому Сончі (Польща).

«ЗАТВЕРДЖУЮ»
Заступник директора Технічного інституту
Державної вищої технічної школи
у м. Новий Сонч (Республіка Польща)
канд. техн. наук Каріна Яніш
Z-ca DYREKTORA
INSTYTUTU TECHNICZNEGO
dr inż. Karina Janisz
"14" 06 2013 р.

АКТ
про впровадження результатів дисертаційної роботи
Гжегожа Літави
в галузі технічних наук за групою спеціальностей "Інформатика і кібернетика"

Основні результати наукового дослідження Гжегожа Літави на здобуття наукового ступеня кандидата технічних наук апробовані та впроваджені у наукових роботах зі статутних і власних досліджень закладу інформатики, а також використані у навчальному процесі при викладанні дисциплін "Безпека інформаційних технологій", "Криптографія і теорія кодів" та "Мережеві технології" для студентів напряму "Інформатика", згідно з Договором про партнерське співробітництво між Державною вищою технічною школою у м. Новий Сонч (Польща) і Тернопільським національним технічним університетом ім.. І.Пулюя (Україна) від 21.06.2011 р.

Використання матеріалів дисертаційної роботи Гжегожа Літави у викладенні вказаних дисциплін та напрямів наукових й методичних досліджень закладу інформатики сприяє підвищенню якості підготовки фахівців.

Старший викладач Технічного інституту
канд. техн. наук **Гжегож Сурувка**

PAŃSTWOWA
WYŻSZA SZKOŁA ZAWODOWA
Instytut Techniczny
ul. Zamenhofa 1a, 33-300 Nowy Sącz
tel. 18 547 29 08, tel./fax 18 547 32 36
(1)

Акт впровадження результатів дисертаційної роботи в ТОВ «Шредер»

Товариство з обмеженою відповідальністю "Шредер"
вул. Миколаївська, 46 Б - 49000 Дніпро, Україна
тел.: 0352 52 1611, факс: 0352 255953, 255610
e-mail: info@schreder.com.ua
www.schreder.com
Member of Schréder Group GIF

«ЗАТВЕРДЖУЮ»
Генеральний директор
ТОВ «Шредер»
Петренко К.Д.
«__» 06 2013 р.

АКТ

про впровадження результатів дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук Літави Гжегожа

Ми, що нижче підписалися, представники ТОВ «Шредер», генеральний директор Петренко К.Д., фінансовий директор Турій Л.Б. склали даний акт в тому, що наукові результати дисертаційної роботи Г. Літави, присвяченої методам і моделям функціональної безпеки та живучості інформаційно-управляючих систем (ІУС), є важливими для повноцінного функціонування та захисту ІУС нашого підприємства, зокрема:

1. Пропоновані у дослідженні підходи щодо удосконалення обчислювальних методів на еліптичних кривих дозволяють зберегти живучість ІУС за наявності потенційних загроз.
2. Застосування апарату Радемахера-Крестенсона для розв'язання дискретного логарифму в ГУС є оригінальним підходом і дає змогу запобігти несанкціонованому доступу до захищених ресурсів в цих системах.
3. Реалізація представлених в роботі методів сприяє підвищенню відмовостійкості та продуктивності ІУС, що функціонує на підприємстві.
4. Запропоновані методи використано для вдосконалення існуючої на підприємстві ІУС.

Генеральний директор
ТОВ «Шредер» Петренко К.Д.

Фінансовий директор Турій Л.Б.

Додаток Б

Опис паралельного алгоритму методу ро Поларда

Таблиця Б.1

Паралельний алгоритм ро Поларда на підставі [72]

Дані: $P \in E; n = ord(P), Q \in \langle P \rangle$

Результат: дискретний логарифм $k = \log_P Q$

1. Вибір відповідно s, яке служитиме для випадкового вибору точок із заданого діапазону
2. Вибір функції поділу H: $\langle P \rangle \to \{0,2,..., s-1\}$
3. Вибір множини W виділених точок z <P>
4. **for** i = 0 to s-1 **do**
5. Випадковий вибір коефіцієнта a_i, $b_i \in_R [1,\ldots, n\text{-}1]$
6. Обчислення i-тої випадкової точки $R_i \leftarrow a_iP + b_iQ$
7. **end for**
8. **for** кожний паралельний процесор **do**
9. Випадковий вибір початкового коефіцієнта $\alpha,\beta \in_R [1,\ldots, n\text{-}1]$
10. Обчислення стартової точки $X \leftarrow \alpha P + \beta Q$
11. **repeat**
12. **if** $X \in W$ це є виділена точка **then**
13. Висилання (X, α, β) до бази виділених точок
14. **end if**
15. Обчислення поточного поділу $i = g(X)$
16. Обчислення наступної точки $X \leftarrow X + R_i$; $\alpha \leftarrow \alpha + a_i \bmod n$; $\beta \leftarrow \beta + b_i \bmod n$
17. **until** якщо в базі виявлено колізію двох точок
18. **end for**
19. Перевірка суперечливих точок X (α_1, d_1, X) і (α_2, d_2, X)

Продовж. табл. Б.1

20. **if** $\alpha_1 = \alpha_2$ **then**
21. **return** помилка
22. **else**
23. Обчислення $k \leftarrow (\alpha_1 - \alpha_2)(\beta_1 - \beta_2)^{-1} \bmod n$
24. **return** l
25. **end if**