МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

Ковтун Марія Григорівна

УДК 004.056.55:003.26

ДИСЕРТАЦІЯ

МЕТОДИ УДОСКОНАЛЕННЯ АРИФМЕТИЧНИХ ОПЕРАЦІЙ У ПОЛЯХ, КІЛЬЦЯХ ТА АЛГЕБРАЇЧНИХ КРИВИХ ДЛЯ КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ

Спеціальність 05.13.21-«Системи захисту інформації»

Подається на здобуття наукового ступеня
кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело _____

Науковий керівник:
Гнатюк Сергій Олександрович
доктор технічних наук, доцент,
доцент кафедри безпеки
інформаційних технологій ННІІДС НАУ

Київ – 2018

АНОТАЦІЯ

Ковтун М.Г. Методи удосконалення арифметичних операцій у полях, кільцях та алгебраїчних кривих для криптографічних застосувань. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.21 «Системи захисту інформації». – Національний авіаційний університет, Київ, 2018.

Дисертаційна робота присвячена вирішенню актуальної науково-технічної задачі криптографічних перетворень для електронного цифрового підпису (ЕЦП) у інформаційно-телекомунікаційних системах центрів сертифікації ключів (ЦСК), шляхом зменшення обчислювальної складності алгоритмів криптографічних перетворень на основі розробки удосконалених методів та алгоритмів арифметичних операцій над числами, поліномами і точками еліптичних кривих (ЕК).

У роботі проведено аналіз функціонування складових Національної системи ЕЦП України та встановлено, що воно на пряму залежить від часу та кількості операцій формування та перевірки ЕЦП. Результати проведеного аналізу дали можливість визначити завдання дисертаційного дослідження щодо *розробки* та *удосконалення методів* для підвищення швидкодії інформаційно-телекомунікаційних систем ЦСК.

Розроблено метод автоматизації приведення довільного полінома за фіксованим модулем у полі $\boldsymbol{GF}\left(2^m\right)$, який враховує степені членів для заданого тричлена та п’ятичлена, що не приводиться, для різних цільових апаратних платформ: як для 8, 16, 32, так і 64-х розрядних, що дозволяє будувати алгоритми приведення за фіксованим модулем з меншою обчислювальною складністю, у відношенні до побітового методу, а також дозволив підвищити швидкодію при формуванні та перевірці ЕЦП, згідно ДСТУ 4145-2002 в 6-9.4 і 7-9.4 разів відповідно.

Удосконалено метод скалярного множення в групі точок ЕК над полем $\boldsymbol{GF}\left(2^m\right)$, який внаслідок проміжних обчислень на кривій Едвардса, за умови

$d_1 = d_2$, дозволяє підвищити стійкість до атак на реалізацію та підвищити швидкодію операції скалярного множення (СМ) на 6% при генерації ключів, накладанні та перевірці ЕЦП за алгоритмами ДСТУ 4145-2002 та ECDSA: формування ЕЦП на 5-7% та перевірки ЕЦП на 6-7% для поля $\boldsymbol{GF}\left(2^{257}\right)$.

Удосконалено метод здобуття n-вимірного кореня в полі $\boldsymbol{GF}\left(2^m\right)$, де m-непарне, на прикладі кубічного кореня, який внаслідок розкладу показника степеню, за допомогою адитивного ланцюга, на множники, дозволяє зменшити обчислювальну складність алгоритму пошуку біраціонально еквівалентних кривих Едвардса до кривих Вейерштрасса з ДСТУ 4145-2002 та рекомендованих NIST FIPS 186-4. Швидкодія відшукання біраціонально еквівалентних кривих Едвардса збільшилась в 1.3-1.8 разів.

Удосконалено метод ділення «в стовпчик» великих цілих чисел, який за рахунок спрощення операції порівняння великих чисел, враховуючи двійкову довжину чисел, проведення операцій зсуву, додавання і віднімання за значимими словами, дозволяє знизити обчислювальну складність звичайного та розширеного алгоритму Евкліда, під час генерації загальних параметрів криптосистеми RSA. Підвищення швидкодії генерації ключів RSA збільшилась на 7-14% зі збільшенням двійкової довжини.

Удосконалено метод мультиплікативного інвертування на основі розширеного алгоритму Евкліда у полі $\boldsymbol{GF}\left(2^m\right)$, який внаслідок використання інформації про двійкову довжину параметрів рівняння Безу: відмова від обчислення степеню полінома, а лише уточнення, зсуви і додавання лише за значимими словами, дозволяє знизити обчислювальну складність при генерації ключів, накладанні та перевірці ЕЦП за алгоритмами ДСТУ 4145-2002 та ECDSA: швидкодія при формуванні та перевірці ЕЦП для ДСТУ 4145-2002 збільшилась в 1.0011-1.0019 і 1.0027-1.0043 разів відповідно.

Методи реалізовані в бібліотеках криптографічних примітивів «Шифр+» v.2.1 системи криптографічного захисту інформації «Шифр-X.509».

Ключові слова: еліптична крива, двійкова крива Едвардса та Вейерштрасса, RSA, ECDSA, ДСТУ 4145-2002, ЕЦП, здобуття кубічного кореня,

скалярне множення, інвертування, Національна система ЕЦП, центр сертифікації ключів.

ABSTRACT

Kovtun M.G. Methods of implementation of high speed arithmetic operations in fields, rings and algebraic curves for cryptographic applications. – Qualifying scientific work as a manuscript.

The Thesis for the Candidate Degree of Technical Sciences, Specialty 05.13.21 «Information security systems». – National Aviation University, Kyiv, 2018.

Thesis is devoted to solving actual scientific and technical problem of cryptographic transformations in information and telecommunication systems of certification authority (CA) in National Electronic Digital Signature System of Ukraine by reducing the computational complexity of cryptographic transformation algorithms on the basis of developing methods and improving algorithms for arithmetic operations over numbers, polynomials and points on an elliptic curve (EC) with reduced computational complexity.

The analysis of the functioning of the components of the National DS system of Ukraine is carried out. It has been established that the functioning of this system depends of the time and number of operations of forming and checking DS. The results of the analysis made it possible to determine tasks of the dissertation research on the development and improvement of methods for increasing the speed of the information and telecommunication systems of the CA of the National EDS system.

Developed method for automating the reduction of an arbitrary polynomial by a fixed module in a field $\boldsymbol{GF}\left(2^m\right)$ that takes into account the degree of terms for a specified irreducible trinomial and pentanomial for different target hardware platforms: 8, 16, 32 and 64 bit. It allows to build modular reduction algorithms for the fixed module with less computational complexity, in relation to the bitwise method, and also to increase the speed during the formation and verification of DS, according to DSTU 4145-2002 in 6-9.4 and 7-9.4 times, respectively.

Developing improved method of scalar multiplication in a group of EC points over a field $\boldsymbol{GF}\left(2^m\right)$, which, due to intermediate computations on the Edwards curve

$d_1 = d_2$, allows to increase the resistance to Side-Channel Attacks and increase the speed of the scalar multiplication (SM) in keys generation, creation and verification of DS according to DSTU 4145 -2002 and ECDSA: signing by 5-7% and verification by 6-7% for the field.

Developing improved method of obtaining an n-dimensional root in a field $\boldsymbol{GF}(2^m)$, where m-odd, on an example of cube root, which allows to reduce the computational complexity of algorithm for searching birationally equivalent Edwards curves to Weierstrass curves from DSTU 4145-2002 and recommended by NIST FIPS 186-4, the speed-up of searching of birationally equivalent curves of Edwards is 1.3-1.8 times.

Developing improved method of dividing the large numbers "in a column", which allows to reduce the computational complexity of the ordinary and extended Euclidean algorithm, by simplifying the operation of comparing large numbers, taking into account the binary length of numbers, performing shift operations, adding and subtracting only meaningful words, in generating common RSA cryptosystem parameters: increasing the speed of RSA key generation on 7-14% with growing binary length.

Developing improved method of multiplicative inversion based on the extended Euclidean algorithm in the field $\boldsymbol{GF}(2^m)$, which, due to use of information about the binary length of the parameters of Bezu equation: the refusal to compute the power of the polynomial, but only clarification, shift and addition only with meaningful words, reduces the computational complexity in key generation, signing and verifigin according to DSTU 4145-2002 and ECDSA: the speed-up of signing and verifing of DS for DSTU 4145-2002 is 1.0011-1.0019 and 1.0027-1.0043 times, respectively.

All proposed methods in dissertation thesis are implemented in library of cryptographic primitives “Cipher+” v.2.1 of CA “Cipher-X.509”.

Keywords: elliptic curve, Edwards and Weierstrass binary curves, RSA, ECDSA, DSTU 4145-2002, DS, cubic roots, scalar multiplication, inverting, National DS system, key certification center.

Список публікацій здобувача:

1. М.Г. Ковтун, В.Ю. Ковтун. «Подходы к повышению производительности операции деления больших целых чисел, на основе расширенного алгоритма Евклида» в *Информационные технологии и защита информации в информационно-коммуникационных системах: раздел коллективной монографии*. В.С. Пономаренко, Харьков: ТОВ «Щедра садиба плюс», 2015, с. 208-219.

2. M. Kovtun, A. Okhrimenko, T. Gancarczyk, V. Karpinskyi, S. Gnatyuk. «Method of Algorithm Building for Modular Reducing by Irreducible Polynomial», *in Proc. of the 16th International Conference on Control, Automation and Systems*, Oct. 16-19, 2016, Gyeongju, Korea. pp.1476-1479. DOI: 10.1109/ICCAS.2016.7832498. *(Scopus)*

3. M. Kovtun, Z. Hu, S. Gnatyuk and N. Seilova, «Method of Searching Birationally Equivalent Edwards Curves Over Binary Fields», *Advances in Intelligent Systems and Computing*, pp. 309-319, 2018. DOI: 10.1007/978-3-319-91008-6_31. *(Scopus)*

4. M. Kovtun, V. Kovtun, A. Okrimenko. «Commands Integrity and Authority in Control Radio Link of UAV», *2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, 2015. DOI: 10.1109/APUAVD.2015.7346593. *(Scopus)*

5. M.G. Kovtun, V.Y. Kovtun, A.A. Okrimenko and S.A. Gnatyuk. «Search method de-velopment of birationally equivalent binary Edwards curves for binary Weierstrass curves from DSTU 4145-2002», in *Proc. PIC S&T*, Kharkov, Ukraine, Oct. 13-15, 2015. pp. 5-8. DOI: 10.1109/INFOCOMMST.2015.7357253. *(Scopus)*

6. М.Г. Булах, В.Ю. Ковтун, «Методы повышения производительности операции инвертирования в двоичном поле», *Безпека інформації*, том 20, № 1, с. 55-61, 2014.

7. М.Г. Ковтун, В.Ю. Ковтун, С.А. Гнатюк, О.М. Бердник, «Подходы к повы-шению производительности расширенного алгоритма Евклида для деления

больших чисел двойной точности на большие числа одинарной точности», *Безпека інформації*, том 21, № 1, с. 40-51, 2015.

8. М. Ковтун, «Применение кривых Эдвардса для защищенной реализации механизмов электронной цифровой подписи согласно ДСТУ 4145-2002», *Системи обробки інформації*, том. 5, №. 151, с. 130-137, 2017.

9. А.О. Охріменко, В.Ю. Ковтун, М.Г. Ковтун, С.П. Євсеєв, О.Г. Король та С.Ю. Ковтун. «Спосіб множення цілих чисел». Україна. Патент 111632, Бюл. 22. Листопад 25, 2016.

10. А.О. Охріменко, В.Ю. Ковтун, М.Г. Ковтун. «Спосіб приведення за модулем цілих чисел». Україна. Патент 118066, Бюл. 14. Липень 25, 2017.

11. А.О. Охріменко, В.Ю. Ковтун, М.Г. Ковтун, С.П. Євсеєв, О.Г. Король, Р.В. Грищук, Г.П. Коц. «Спосіб піднесення до квадрату цілих чисел». Україна. Патент 118065, Бюл.14. Липень 25, 2017.

12. М.Г. Ковтун, С.А. Гнатюк, В.И. Трофименко. «Ускоренное извлечение r-го корня в двоичном поле» в *Докл. Межд. науч.–практ. конф. Информационные и телекоммуникационные технологии: образование, наука, практика*, Алматы, Казахстан, Декабрь, 2-4, 2015, с. 547-551.

13. М.Г. Булах, В.Ю. Ковтун. «Модифицированный алгоритм мультипликативного инвертирования в двоичном поле», *Науч.-практ. Конф. «Проблемы эксплуатации и защиты информационно-коммуникационных систем»,* 2014, Киев, Украина, с. 11.

14. М.Г. Ковтун, В.Ю. Ковтун. «Подходы к повышению производительности операции деления больших целых чисел, на основе расширенного алгоритма Евклида», *18 Між. Наук.-прак. Конф. «Проблеми та перспективи розвитку IT-індустрії»,* 2015, Харків, Україна, с. 31.

15. М.Г. Ковтун, С.А. Гнатюк. «Модифицированный расширенный алгоритм Евклида для деления больших целых чисел двойной точности на числа одинарной точности», *15 Між. Наук.-практ. Конф.:Політ. Сучасні проблеми науки,* 2015, Київ, Україна, с. 121.

16. М.Г. Ковтун, С.А. Гнатюк. «Ускоренное мультипликативное инвертирование в двоичном поле для ДСТУ 4145-2002», *12 Між. Наук.-техн. Конф.:ABIA,* 2015, Київ, Україна, с. 2.62-2.65.

17. М.Г. Ковтун, С.А. Гнатюк. «Классификация алгоритмов деления и приведения по модулю для целых чисел в криптографических приложениях», *5th International Scientific Conference: ITSEC*, Киев-2015, с. 56-57.

18. М.Г. Ковтун, В.Ю. Ковтун, С.А. Гнатюк. «Быстрое деление целых чисел для криптографических приложений», *Безопасность информации в информационно-телекоммуникационных системах: 17 Межд. Конф.*, Киев-2015, с. 12-13.

19. М.Г. Ковтун. «Модифицированный алгоритм Евклида для деления больших целых чисел двойной и одинарной точности», *Інформаційна безпека держави, суспільства та особистості,* Кировоград, 2015, с. 55.

20. М.Г. Ковтун, А.А. Охрименко. «Методы построения алгоритма приведения по фиксированному модулю неприводимого полинома», *18 Межд. Научн.-практ. Конф.: Безопасность информации в информационно-телекоммуникационных системах,* Киев-2016, с. 21.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ЗО	Засвідчувальний орган
ЗЦ	Засвідчувальний центр
ЦЗО	Центральний засвідчувальний орган
АРМ	Автоматизоване робоче місце
АЦСК	Акредитований центр сертифікації ключів
КО	Контролюючий орган
ЕЦП	Електронний цифровий підпис
ПТК	Програмно-технічний комплекс
ЕК	Еліптична крива
СМ	Скалярне множення
БЕК	Біраціонально еквівалентна крива
ІВК	Інфраструктура відкритих ключів
ІТС	Інформаційно-телекомунікаційна система
КСЗІ	Криптографічна система захисту інформації
БД	База даних
ДКЕ	Довгострокові ключові елементи
МГК	Модуль генерації ключів
МУК	Модуль управління ключами
ОС	Операційна система
ПЗ	Програмне забезпечення
СВС	Список відкликаних сертифікатів
СУБД	Система управління базами даних
ВЦР	Віддалений центр реєстрації
ЦР	Центр реєстрації
ЦСК	Центр сертифікації ключів
ЦПД	Центр прийому дзвінків
HTTP	Hypertext Transfer Protocol
LDAP	Lightweight Directory Access Protocol
НСЕЦП	Національна система електронного цифрового підпису

OCSP	Online Certificate Status Protocol
PKCS#7	Cryptographic Message Syntax Standard
PKCS#10	Certification Request Standard
PKCS#11	Cryptographic Token Interface
TCP	Transmission Control Protocol
TSP	Time Stamping Protocol
CRL	Certificate Revocation List

ВСТУП

Актуальність. У новому тисячолітті суспільство переходить до інформаційної ери зі швидкоплинними інформаційними процесами, що зумовлюється постійним розвитком та удосконаленням інформаційних технологій. Для надання юридичної значущості даним процесам та електронним документам, що полягає у чіткій формалізації та їх автоматизації, в Україні були прийняті Закони України «Про електронний цифровий підпис», «Про електронні документи та електронний документообіг», «Про електронні довірчі послуги», які передбачають використання електронного цифрового підпису (ЕЦП) юридичними та фізичними особами, як аналог власного підпису. Процедура формування та перевірки ЕЦП виконується згідно державного стандарту ДСТУ 4145-2002. У напрямку інтеграції України у міжнародне суспільство, було надано нормативного статусу цілій низці міжнародних стандартів у галузі криптографії, а також розгорнуто гілки для RSA та ECDSA у Центральному засвідчувальному органі (ЦЗО) України.

Для забезпечення довірчого інформаційного простору, в Україні створено Національну систему ЕЦП, в межах якої сертифікуються ключі ЕЦП та ключі для вироблення спільного секрету. Під час розгортання та її експлуатації, виникає велика кількість юридичних, економічних та технічних питань. Серед технічних, слід виділити інформаційно-телекомунікаційні системи (ІТС), завдяки яким і можлива робота довірчого інформаційного простору України. Зараз ЕЦП використовується у багатьох державних і приватних установах у безперервному режимі для виконання електронних платежів у Національному банку України, для податкової звітності Державної фіскальної служби, різноманітні державні реєстри, державні закупівлі та торги тощо.

Досвід експлуатації таких систем за кордоном і в Україні, показує тенденцію зростання кількості звернень до складових частин – центрів сертифікації ключів (ЦСК) та періодичних сезонних імпульсів, пов’язаних зі здачею податкової звітності, формуванню різноманітних звітів, проведенням торгів та закупівель, подачею податкових декларацій державними службовцями.

З часом, такі навантаження можуть перевищити на які розраховані ЦСК та зменшити якість обслуговування користувачів відповідних ІТС, кількість яких постійно зростає, особливо, де використовується ЕЦП, що вимагає безперервного процесу модернізації Національної системи ЕЦП та її складових. Модифікація полягає, як у використанні апаратних засобів, з кращою обчислювальною потужністю, а також і телекомунікаційного обладнання, розрахованого на більшу пропускну здатність. Однак, заміна апаратного забезпечення буває фінансово невигідна або технічно неможлива. В таких випадках єдиним рішенням є удосконалення лише програмної частини ЦСК, тому виникає інтерес до підвищення швидкодії виконання операцій з ЕЦП, а також до пошуку математичного апарату для перспективних криптографічних перетворень. Зараз, активно використовуються схеми ЕЦП: на еліптичних кривих (ЕК) (ДСТУ 4145-2002, ECDSA, ECGDSA, ECKDSA, ГОСТ 34.10-2012, СТБ 34.101.45-2011); на перетвореннях у полях та кільцях (DSA); на перетвореннях у кільцях (RSA).

Алгоритми підпису, що базуються на арифметичних перетвореннях на ЕК використовують – операції над точками, які в свою чергу базуються на арифметичних операціях над координатами точок, що представлені як елементи базового поля та поля порядку ЕК. Елементи полів можуть бути представлені як великі поліноми чи великі цілі числа, які також використовуються в алгоритмах ЕЦП, що базуються на арифметичних перетвореннях у полях та кільцях.

Таким чином, *актуальною науково-технічною задачею* є підвищення швидкодії криптографічних операцій у ІТС ЦСК Національної системи ЕЦП, шляхом зменшення обчислювальної складності алгоритмів криптографічних перетворень на основі розробки та удосконалення методів арифметичних операцій над числами, поліномами і точками ЕК зі зменшеною обчислювальною складністю.

Зв’язок роботи з науковими програмами, планами, темами.

Тематика дисертаційної роботи та одержані результати безпосередньо пов’язані з «Основними науковими напрямами та найважливішими проблемами

фундаментальних досліджень у галузі природничих, технічних і гуманітарних наук НАН України на 2014-2018 роки», Стратегією кібербезпеки України від 15 березня 2016 року №96/2016 і Рамковою програмою ЄС з досліджень та інновацій «Горизонт 2020». Результати роботи відображені у звітах держбюджетних НДР Національного авіаційного університету «Квантово-криптографічні методи захисту критичної інформаційної інфраструктури держави» (д.р. № 0117U006770), «Методи забезпечення конфіденційності державних інформаційних ресурсів в інформаційно-комунікаційних системах» (№ 61/09.01.08), «Новітні технології криптографічного захисту інформації» (№ 100/14.01.06), у яких здобувач брав участь в якості виконавця.

Мета та завдання дослідження. Метою дисертаційної роботи є підвищення швидкодії інформаційно-телекомунікаційних систем центрів сертифікації ключів Національної системи ЕЦП за рахунок розробки та удосконалення методів арифметичних перетворень над числами, поліномами і точками еліптичної кривої зі зменшеною обчислювальною складністю і протидією до атак на їх реалізацію.

Для досягнення поставленої мети необхідно розв'язати такі **основні задачі**:

– проаналізувати методи постановки та перевірки ЕЦП, які використовуються у Національній системі ЕЦП Україні та шляхи, щодо підвищення їх швидкодії;

– розробити метод ділення великих цілих чисел у кільці цілих чисел зі зменшеною обчислювальною складністю;

– розробити метод мультиплікативного інвертування, приведення полінома за фіксованим модулем та здобуття кубічного кореня у полі $\boldsymbol{GF}\left(2^m\right)$ зі зменшеною обчислювальною складністю;

– розробити метод арифметичних перетворень в групі точок ЕК зі зменшеною обчислювальною, структурною складністю і протидією до атак на реалізацію;

– розробити програмні моделі криптографічних перетворень на ЕК, створити на їх основі бібліотеку криптографічних перетворень;

– експериментально дослідити розроблену бібліотеку на підтвердження наукових результатів.

Об'єктом дослідження є процес криптографічних перетворень з відкритим ключем у інформаційно-телекомунікаційних системах ЦСК Національної системи ЕЦП.

Предметом дослідження є методи та способи арифметичних перетворень над числами, поліномами і точками ЕК, що застосовуються у криптографічних перетвореннях з відкритим ключем.

Методи дослідження. Проведені дослідження базуються на сучасних методах оцінки складності алгоритмів та теорії алгоритмів (для аналізу складності алгоритму скалярного множення та арифметичних операцій у групі точок ЕК, полях та кільцях); теорії криптографії (для аналізу криптографічних перетворень, побудованих на ЕК, полях та кільцях); ймовірності та комбінаторики (для аналізу складності алгоритмів); теорії еліптичних кривих (для удосконалення арифметичних операцій на ЕК у формі Вейерштрасса та Едвардса, пошуку біраціонально еквівалентних відображень кривої Вейерштрасса до кривої Едвардса); теорія кілець, полів та ідеалів (для удосконалення методів мультиплікативного інвертування у полі $\boldsymbol{GF}\left(2^m\right)$, здобуття кубічного кореня у полі $\boldsymbol{GF}\left(2^m\right)$, приведенням полінома за фіксованим модулем у полі $\boldsymbol{GF}\left(2^m\right)$, ділення великих цілих чисел у полі $\boldsymbol{GF}\left(p\right)$ та кільці цілих чисел).

Наукова новизна отриманих результатів. У ході розв'язання поставлених задач *отримала подальший розвиток* теорія перетворень на ЕК, а також отримані такі результати:

– *вперше розроблено метод* автоматизації приведення довільного полінома за фіксованим модулем у полі $GF(2^m)$, який враховує степені членів для заданого тричлена та п'ятичленна, що не приводиться, для різних цільових апаратних платформ, що дозволяє будувати алгоритми приведення за фіксованим модулем з меншою обчислювальною складністю по відношенню з побітовим методом.

– *удосконалено метод* скалярного множення в групі точок ЕК над полем $\boldsymbol{GF}\left(2^m\right)$, який за рахунок проміжних обчислень на кривій Едвардса, при $d_1 = d_2$, дозволяє підвищити стійкість до атак на реалізацію та підвищити швидкодію операції СК при генерації ключів, накладанні та перевірці ЕЦП за алгоритмами ДСТУ 4145-2002 та ECDSA.

– *удосконалено метод* здобуття n – вимірного кореня в полі $\boldsymbol{GF}\left(2^m\right)$, де m – непарне, на прикладі кубічного кореня, який за рахунок розкладу показника степеню за допомогою адитивного ланцюга на множники, дозволяє зменшити обчислювальну складність алгоритму пошуку біраціонально еквівалентних кривих Едвардса до кривих Вейерштрасса з ДСТУ 4145-2002 та рекомендованих NIST FIPS 186-4.

– *удосконалено метод* ділення «в стовпчик» великих цілих чисел, який за рахунок спрощення операції порівняння великих чисел, враховуючи двійкову довжину чисел; проведення операцій зсуву, додавання і віднімання за значущими словами, дозволяє знизити обчислювальну складність звичайного та розширеного алгоритму Евкліда, під час генерації загальних параметрів криптосистеми RSA.

– *удосконалено метод* мультиплікативного інвертування на основі розширеного алгоритму Евкліда у полі $\boldsymbol{GF}\left(2^m\right)$, який за рахунок використання інформації про двійкову довжину параметрів рівняння Безу: відмова від обчислення степеню полінома, а лише уточнення, зсуви і додавання лише за значущими словами, дозволяє знизити обчислювальну складність при генерації ключів, накладанні та перевірці ЕЦП за алгоритмами ДСТУ 4145-2002 та ECDSA.

Практичне значення отриманих результатів полягає:

1. У розробці алгоритму ділення великих цілих чисел «в стовпчик», який дозволив підвищити швидкодію в 1,5-3 разів для чисел однакової довжини починаючи з довжини числа 512 біт, і з 128 біт для випадку, коли різниця в довжині між діленим та дільником складає 2 рази.

2. У розробці алгоритму мультиплікативного інвертування в полі $\boldsymbol{GF}\left(2^m\right)$ на основі розширеного алгоритму Евкліда, який дозволив підвищити швидкодію реалізації в 1.2-1.8 разів відносно алгоритму прототипу.

3. У розробці алгоритму побудови процедури приведення за фіксованим модулем у полі $\boldsymbol{GF}\left(2^m\right)$, який дозволяє будувати алгоритми для поліномів, що не приводяться, на різних цільових платформах, що дозволяють збільшити швидкодію операції приведення за модулем у 34-197 разів зі зростанням двійкової довжини відносно звичайного побітового алгоритму.

4. У розробці алгоритму здобуття n- вимірного кореня в полі $\boldsymbol{GF}\left(2^m\right)$, на прикладі здобуття кубічного кореня в полі $\boldsymbol{GF}\left(2^m\right)$, який дозволив зменшити обчислювальну складність в 4-4.9 разів і підвищити швидкодію в 2,8-3,7 разів зі зростанням двійкової довжини елемента поля.

5. У розробці алгоритму скалярного множення на основі удосконаленого методу з використанням проміжних обчислень на кривій Едвардса, при умові рівності параметрів $d_1 = d_2$, який дозволив підвищити швидкодію скалярного множення на 6%, ЕЦП на 5-7% та перевірки ЕЦП на 6-7% для поля $\boldsymbol{GF}\left(2^{257}\right)$.

6. Алгоритми реалізовано у бібліотеках криптографічних примітивів «Шифр+v.2.1» системи криптографічного захисту інформації «Шифр-X.509», що має дійсний позитивний експертний висновок Держспецзв'язку України від 16.05.2017 №04/03/02-1674 (Акт від 29.09.2017 р. №12/09-17). Результати дисертаційних досліджень впроваджено у навчальний процес кафедри безпеки інформаційних технологій НАУ (Акт від 18.01.2018 р.).

Особистий внесок здобувача. Основні положення і результати дисертаційної роботи, що виносяться до захисту, отримані автором самостійно. У роботах, написаних у співавторстві, автору належать: у публікації [6] досліджувались обчислювальна та просторова складності алгоритму мультиплікативного інвертування на основі розширеного алгоритму Евкліда у двійковому полі, та був запропонований удосконалений метод для підвищення швидкодії; у роботах [1, 7] досліджувалась та була доповнена класифікація

алгоритмів ділення та приведення за модулем великих цілих чисел, а також було проведено дослідження обчислювальної складності алгоритму ділення великих цілих чисел «в стовпчик» та удосконалених методів з ефективною програмною реалізацією; у роботі [2] досліджено приклади операції приведення за фіксованим модулем у двійковому полі та розроблено метод побудови алгоритму приведення за фіксованим модулем, який не залежить від характеристики полінома, що не приводиться; в роботі [12] виконано розклад показників степенів на множники для підвищення швидкодії операції здобуття кубічного кореня у двійковому полі; в роботі [3, 5] проведено дослідження обчислювальної складності алгоритму пошуку біраціонально еквівалентних кривих Едвардса до кривих Вейерштрасса та запропоновано удосконалені методи для програмної оптимізації; в роботі [8] розроблено модель операції СМ на основі проміжних обчислень на кривих Едвардса та досліджено швидкодію запропонованого і відомого методів; в патентах [9-11] досліджувалась швидкодія та обчислювальна складність запропонованих способів та прототипів.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на наступних конференціях: Науково-практична конференція «Проблеми експлуатації і захисту інформаційно-комунікаційних систем» (Київ, 2014); Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості» (Кіровоград, 2015); Міжнародна науково-практична конференція «Проблеми та перспективи розвитку ІТ-індустрії» (Харків, 2015); П’ятнадцята міжнародна науково-практична конференція молодих учених і студентів «ПОЛІТ»: Сучасні проблеми науки. Інформаційно-діагностичні системи: тези доповідей (Київ, 2015); Дванадцята міжнародна науково-технічна конференція «ABIA-2015» (Київ, 2015); П’ята міжнародна науково-технічна конференція «ITSEC» (Київ, 2015); Шістнадцята міжнародна науково-практична конференція «Безпека інформації у інформаційно-телекомунікаційних системах» (Київ, 2015); 3rd International Conference on the Actual Problems Of Unmanned Aerial Vehicles Developments «APUAVD 2015» (Ukraine, Kyiv, 2015); Сімнадцята міжнародна

науково-практична конференція «Безпека інформації у інформаційно-телекомунікаційних системах» (Київ, 2016); 16th International Conference on Control, Automation and Systems «ICCAS 2016» (Korea, Gyeongju, 2016), 1st International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2018) (Ukraine, Kyiv, 2018).

Публікації. Основні положення і результати дисертаційної роботи викладено в 20 наукових публікаціях: 7 наукових статей (4 – у міжнародних рецензованих виданнях, що входять до баз даних Scopus та 3 – у вітчизняних фахових наукових журналах), 1 розділ колективної монографії, 3 патенти України на корисну модель, 9 матеріалів та тез доповідей.

Структура та обсяг дисертації. Дисертація складається з анотації, змісту, переліку умовних позначень, вступу, п’ятьох розділів, висновку, додатків, списку використаних джерел (в кінці кожного розділу основної частини дисертації). Обсяг основного тексту дисертації складає 121 сторінку, 6 додатків на 37 сторінках, 26 рисунків, 35 таблиць. Перелік використаних джерел складається з 113 найменувань на 15 сторінках. Загальний обсяг дисертаційної роботи складає 158 сторінок.

РОЗДІЛ 1
ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО КОМПЛЕКСИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ ЦСК

1.1. Аналіз програмно-технічного комплексу ЦСК

Внаслідок активного розвитку інформаційних технологій, величезну цінність на сьогоднішній день представляє інформація, яка являється стратегічним ресурсом держави та бізнесу всіх рівнів, які зацікавлені в її створенні, передачі, обробці, збереженні та знищенні.

Для надання юридичної значимості інформаційним процесам та електронним документам, в Україні були прийняті Закони України «Про електронний цифровий підпис» [1], «Про електронні документи та електронний документообіг» [2], «Про електронні довірчі послуги» [3], які передбачають використання електронного цифрового підпису (ЕЦП) юридичними та фізичними особами, як аналог власного підпису. Застосування ЕЦП також регулюється за допомогою спільних наказів Держспецзв'язку та Мін'юсту України:

– Вимог до форматів криптографічних повідомлень;

– Переліку міжнародних та європейських стандартів, інших актів технічного регулювання для гармонізації з метою реформування, розвитку та забезпечення інтероперабельності системи електронного цифрового підпису.

– Вимоги до форматів, структури та протоколів, що реалізуються у надійних засобах електронного цифрового підпису.

– Переліків стандартів у сфері електронного цифрового підпису, перспективних для перегляду та гармонізації з європейськими та міжнародними стандартами відповідно до встановлених законодавством процедур.

– Вимоги до алгоритмів, форматів та інтерфейсів, що реалізуються у засобах шифрування та надійних засобах електронного цифрового підпису.

Процедура формування і перевірки ЕЦП виконується згідно державного стандарту ДСТУ 4145-2002.

Схема захищеної передачі документа через публічні мережі за допомогою ЕЦП, представлена на рис. 1.1.1.

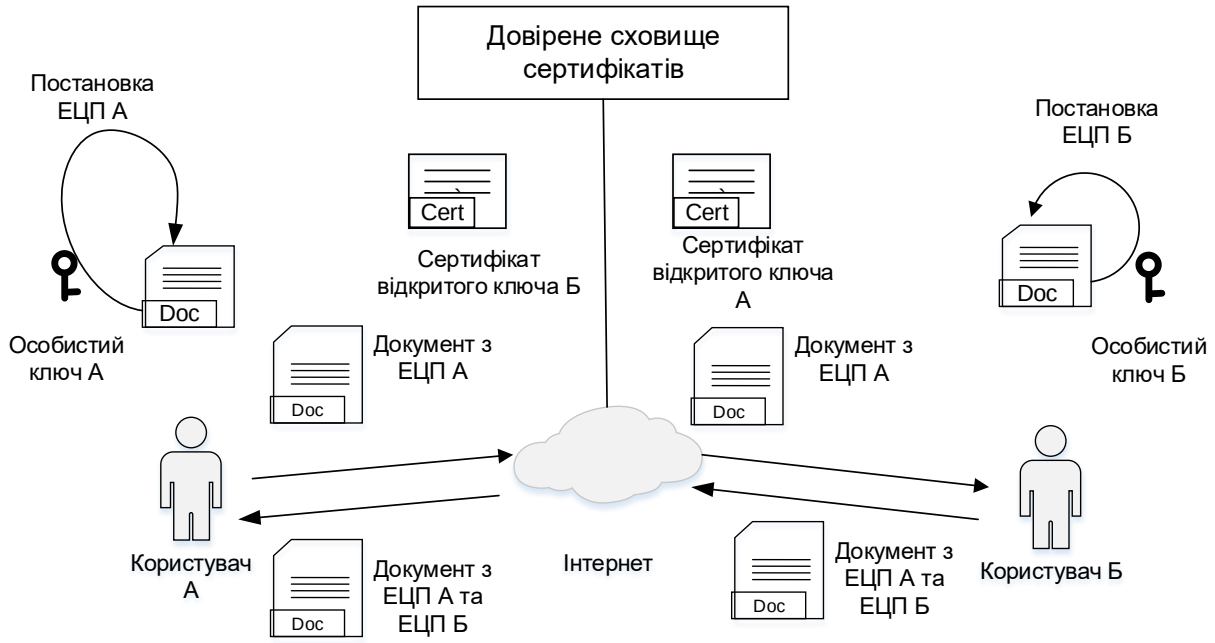

Рис.1.1.1. Схема захищеної передачі документів з використанням ЕЦП

Користувач А підписує (ставить ЕЦП) своїм особистим ключем документ, наприклад, для узгодження, і відправляє його користувачу Б. Користувач Б завантажує контейнер ЕЦП. В довіреному сховищі завантажує сертифікат відкритого ключа користувача А, перевіряє його на коректність, а також перевіряє на відповідність ЕЦП. Якщо все вірно, користувач Б підписує (завіряє своїм підписом документ) і відправляє його користувачу А. Після чого користувач А для перевірки ЕЦП користувача Б проводить ті самі дії, які описувались вище.

В Україні побудована і активно розвивається Національна інфраструктура відкритих ключів (ІВК) – Національна система ЕЦП (НС ЕЦП) – комплекс програмно-апаратних засобів та організаційно-технічних заходів, необхідних для використання асиметричних криптографічних схем в прикладних сферах, де використовуються механізми ЕЦП. Структура системи ЕЦП України складається з Центрально засвідчувального органу України (ЦЗО), який є ключовим елементом, засвідчувального органу (ЗО), акредитованих і звичайних центрів сертифікації ключів (АЦСК і ЦСК), контролюючого органу

(КО), який засвідчує центр Національного банку України (ЗЦ НБ України). Функціонування даних складових, показано на рис. 1.1.2.

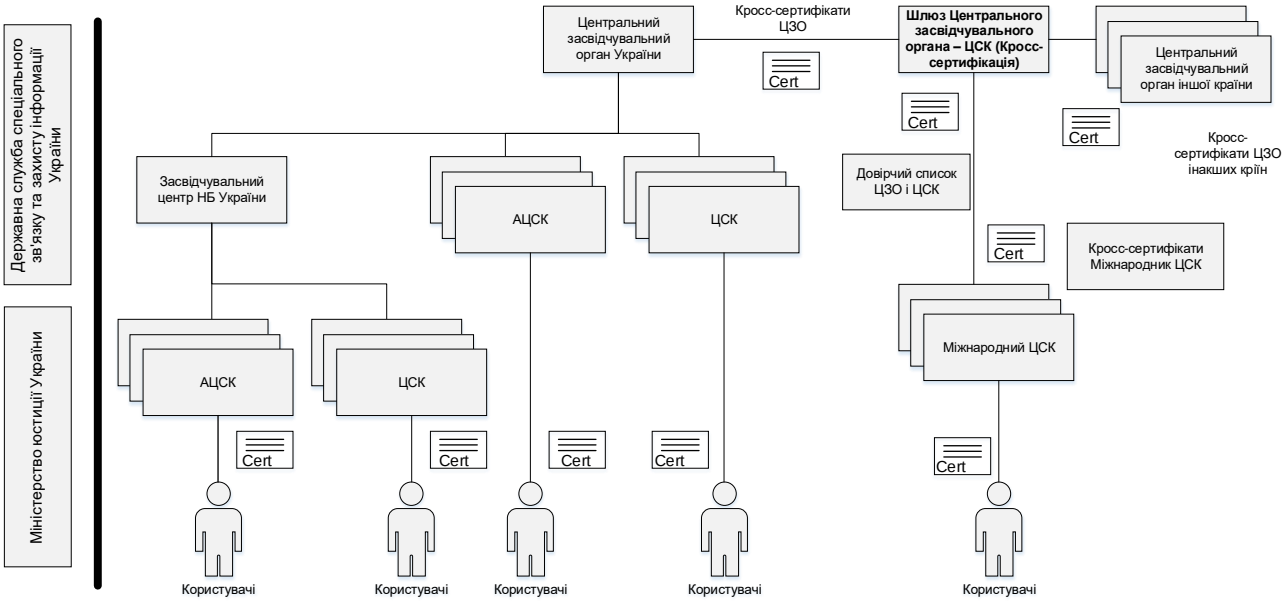

Рис.1.1.2. Структурна схема Національної системи ЕЦП

НС ЕЦП від імені держави гарантує якість послуг ЕЦП, а ЦЗО регулює (контролює) їх якість, будучи органом акредитації та державного нагляду за діяльністю ЦСК, акредитованих та зареєстрованих.

ЦЗО відповідає вимогам, встановленим законодавством для АЦСК. Положення про ЦЗО затверджується Кабінетом Міністрів України. Він видає, блокує, скасовує чи поновлює посилені сертифікати ключів ЗЦ та ЦСК, проводить акредитацію ЦСК, забезпечує цілодобовий доступ ЗЦ і ЦСК до посилених сертифікатів ключів та відповідних електронних реєстрів та інше.

ЗЦ виконує функції центру довіри всієї системи. Він випускає сертифікати ключів перевірки електронних підписів і видає їх користувачам, створює ключі ЕЦП і ключі перевірки ЕЦП, отримує і обробляє повідомлення про компрометацію ключів, анулює видані цим центром сертифікати, довіра до яких втрачена тощо.

З вище перерахованими функціями до НС ЕЦП, висувається важлива вимога – безперебійне функціонування в режимі реального часу, а також оперативне опрацювання визначеної кількості запитів на серверах з нерівномірним трафіком. Відповідно з напрямком дисертаційної роботи, інтерес

представляють питання програмно-технічного характеру пов'язані з функціонуванням системи.

Для взаємодії Національної ІВК з іншими ІВК (системами ЕЦП і міжнародними ЦСК) розробляється національний шлюз, який виконуватиме наступні завдання (рис.1.1.2):

– Формування списку довірених кореневих сертифікатів.

– Формування списку відкликаних сертифікатів (СВС).

– Формування крос-сертіфікатів.

– Функціонування OCSP сервісу.

– Ведення служби каталогу.

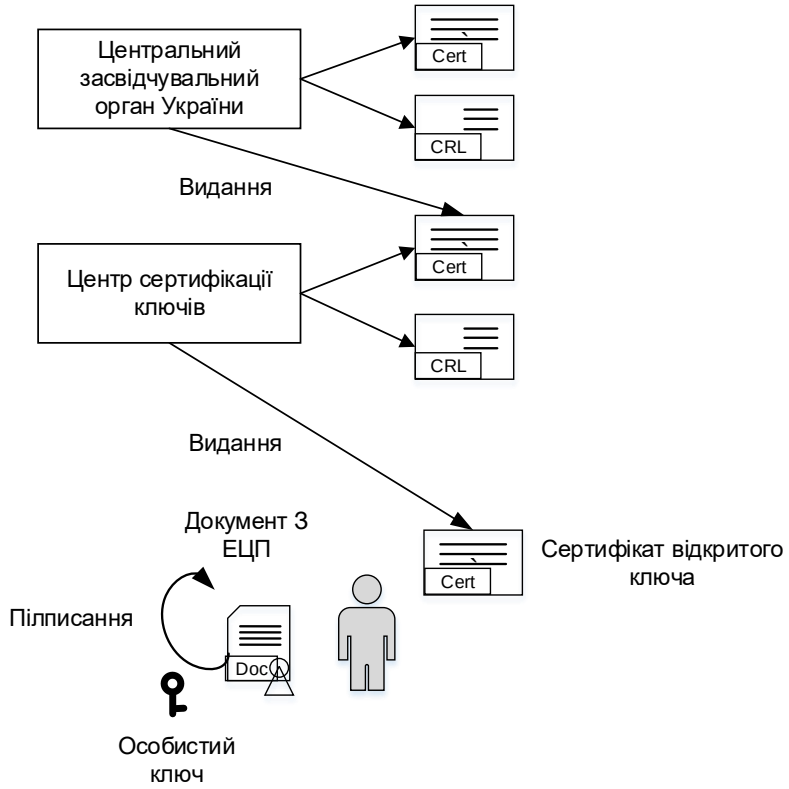

Рис.1.1.3. Шлях сертифікації для сертифікат користувача

На рис. 1.1.3 показана схема, за якою відбувається процес сертифікації для користувача. ЦЗО за допомогою свого особистого ключа видає власний самопідписаний сертифікат на основі алгоритму ЕЦП ДСТУ 4145-2002, а також видає СВС і дельту СВС. Далі ЦСК подають запит на сертифікат (реєстрація) в форматі PKCS#10, на основі якого ЦЗО видає сертифікат відкритого ключа. В свою чергу, ЦСК формує сертифікати відкритого ключа користувача і засвідчує

його своїм ЕЦП. Новий сертифікат переміщується в БД дійсних сертифікатів ЦСК (створюється СВС і дельта СВС), і стає доступним для всіх користувачів за загальнодоступними телекомунікаційними каналами. Потім користувач за допомогою свого особистого ключа створює ЕЦП для документа. Отримавши повідомлення, отримувач звертається до БД сертифікатів, за ідентифікаційним даним сертифікату підписанта та отримує його сертифікат, перевіряє його статус та коректність даних. Якщо сертифікат дійсний на момент перевірки ЕЦП, з отриманого сертифікату вилучається відкритий ключ відправника й виконується перевірка його підпису.

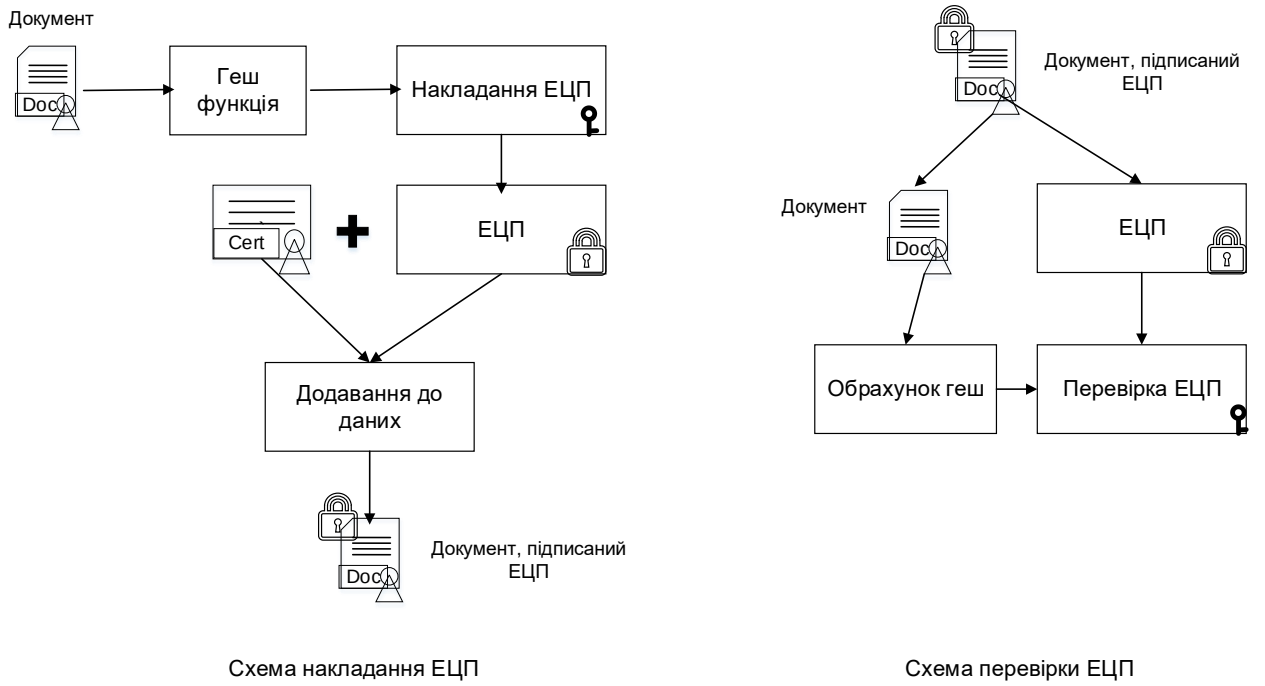

Схема накладання ЕЦП Схема перевірки ЕЦП

Рис.1.1.4. Схема накладання та перевірки ЕЦП

Процес підписання документа виглядає наступним чином (рис. 1.1.4). На першому кроці обраховується геш-функція, вона ідентифікує зміст документа. На другому кроці автор документу накладає ЕЦП на геш-функцію своїм особистим ключем. Створена ЕЦП для документу заноситься до PKCS#7 конверту, сам документ також може заноситись до цього конверту.

При отриманні підписаного документу, користувач може переконатися в його достовірності. На першому етапі одержувач повідомлення обчислює геш-функцію підписаного документа. На другому етапі відбувається перевірка ЕЦП, що дозволяє перевірити одночасно цілісність змісту документа та авторство.

Програмно-технічний комплекс (ПТК) ЦСК, на прикладі системи криптографічного захисту інформації «Шифр-Х.509» [5][6], призначений для реалізації регламентних процедур і механізмів роботи ЦСК, пов'язаних з обслуговуванням сертифікатів відкритих ключів, які включають [5]:

– управління ключами ЦСК;

– реєстрацію користувачів;

– сертифікацію відкритих ключів користувачів;

– поширення сертифікатів;

– управління статусом сертифікатів;

– поширення інформації про статус сертифікатів;

– надання послуг фіксування часу.

Коротко опишемо складові частини ПТК ЦСК [5] (рис. 1.1.5):

– LDAP-сервер (каталог) призначений для створення довідника сертифікатів ЦСК. Він забезпечує зберігання і надання доступу користувачам до опублікованих сертифікатів та списку відкликаних сертифікатів (СВС), які видаються ЦСК.

– OCSP-сервер призначений для передачі користувачам інформації про статус сертифіката в інтерактивному режимі протоколом TCP/IP або HTTP. Клієнт посилає серверу запит на отримання статусу сертифіката, а сервер повертає відповідь зі статусом сертифіката, за інформацією у СВС. СВС - спеціальний файл, який публікується в загальнодоступному місці в мережі ЗЦ. В файлі містяться ідентифікатори всіх відкликаних сертифікатів ЦСК та ЕЦП накладену відповідним ключем ЕЦП ЦСК. СВС список публікується з інтервалом визначеним регламентом ЦСК (наприклад, раз на тиждень, раз на добу чи раз на годину).

– TSP-сервер призначений для формування та передачі позначок часу для користувачів в інтерактивному режимі протоколом TCP/IP або HTTP.

– Сервер БД Центру реєстрації (ЦР) є виділеним сервером, на якому функціонує система управління БД (СУБД) і зберігаються дані всього ЦР.

– ПЗ АРМ віддаленого адміністратора реєстрації задовольняє заявки на видання сертифікатів, для цього йому необхідно надати доступ до БД ЦР.

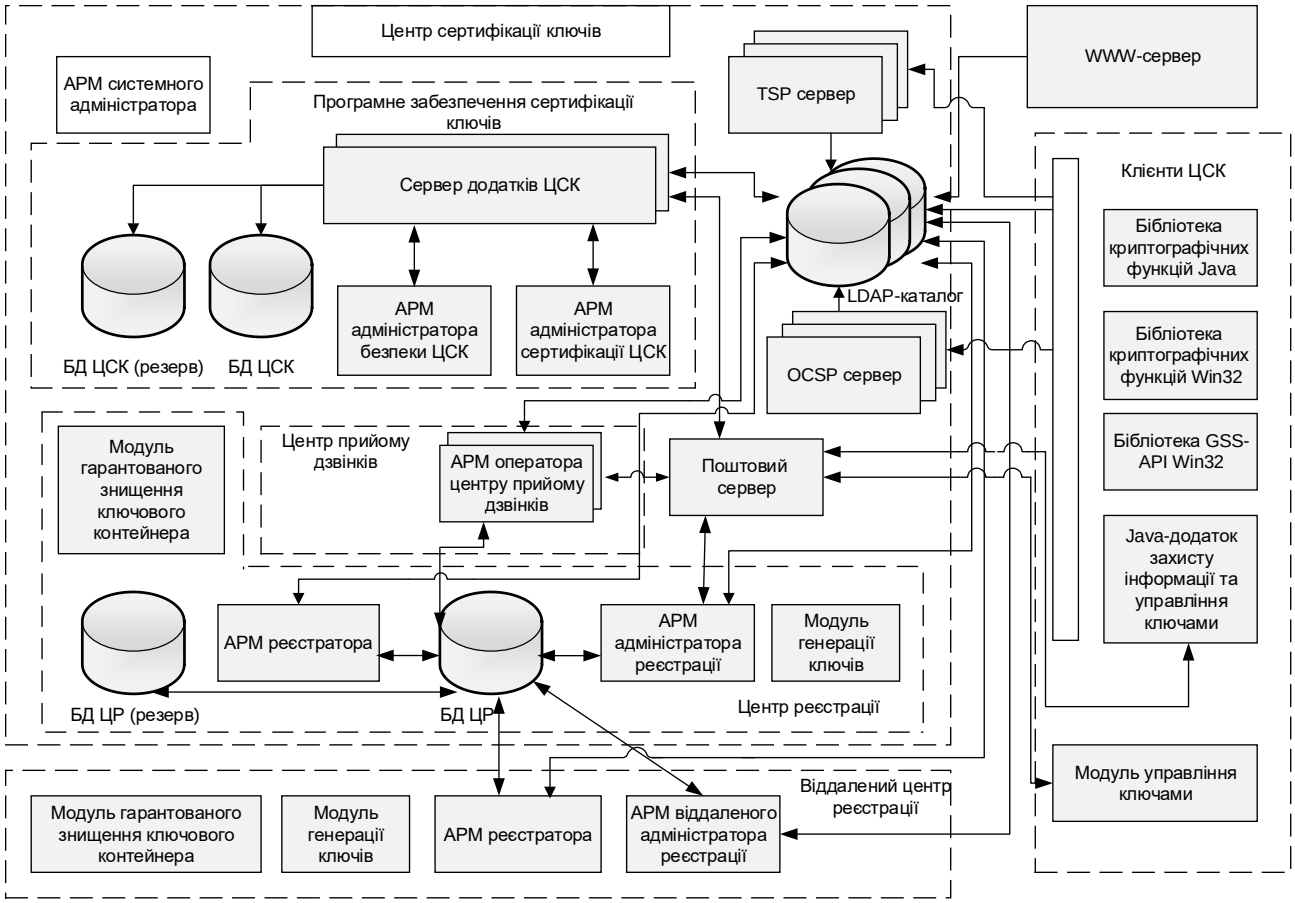

Рис.1.1.5. Структурна схема ПТК ЦСК

– ПЗ АРМ адміністратора сертифікації ЦСК - це клієнт сервера додатків ЦСК і призначений для управління сертифікатами виконавців.

– ПЗ АРМ адміністратора безпеки ЦСК - клієнт сервера додатків ЦСК і призначений для управління технологічними сертифікатами підсистеми управління ключами і аудиту роботи сервера ЦСК.

– ПЗ АРМ адміністратора реєстрації задовольняє заявки на видання сертифікатів, для цього йому необхідно надати доступ до БД, а також джерела запитів на видання сертифікатів в форматі PKCS#7.

– Сервер застосувань ЦСК призначений для видання сертифікатів відкритих ключів і СВС. Передбачається функціонування основного сервера і резервного сервера, кожен з яких функціонує зі своєю локальною СУБД.

– ПЗ АРМ оператора ЦПД призначене для виконання дій, пов'язаних з управлінням статусів сертифікатів на вимогу користувачів, яким ці сертифікати належать.

– ПЗ МУК призначене для виконання користувачами самостійних дій по генерації ключів та отримання відповідних сертифікатів.

– WWW-сервер ознайомитися з документами, що регламентують діяльність ЦСК, а також багато іншої корисної інформації. Через WWW-сервер, користувачі можуть переглядати вміст LDAP-каталогу.

– Бібліотеки криптографічних функцій базуються на наборі бібліотек «Шифр+» для 8, 16, 32 та 64-х розрядних процесорів з архітектурою x86, ARM, MIPS та різних ОС. Дані бібліотеки призначені для інтеграції до складу технологічного ПЗ, для виконання операцій середнього та високого рівня, пов'язаних з виробленням і перевіркою ЕЦП, шифруванням і управлінням ключами.

З огляду на специфіку побудови НС ЕЦП, нижче описується механізм перевірки ЕЦП поставленої під документом за допомогою ключів ЦСК (рис. 1.1.6):

– З репозитаріїв LDAP проводиться завантаження сертифікатів відкритих ключів підписувача, після чого за ланцюжком – завантаження сертифікату ЦСК (АЦСК) видавця і сертифіката ЦЗО.

– Для кожного сертифіката зі списку проводиться перевірка термінів дії і ЕЦП під ними. Таким чином формується довірений ланцюжок сертифікатів.

– Відбувається перевірка статусу сертифіката підписувача (звернення до сервера OCSP). Статус сертифіката ЦСК проводиться за СВС ЦЗО (або через звернення до сервера OCSP), під яким перевіряється ЕЦП. У разі компрометації сертифіката ЦЗО, в СВС додаються ідентифікатори всіх сертифікатів ЦСК, що обриває ланцюжок сертифікатів.

– Для підпису документа проводиться перевірка позначка часу документу/ЕЦП (створена за допомогою сервера TSP), яка дозволяє підтвердити факт існування документа/ЕЦП на момент часу.

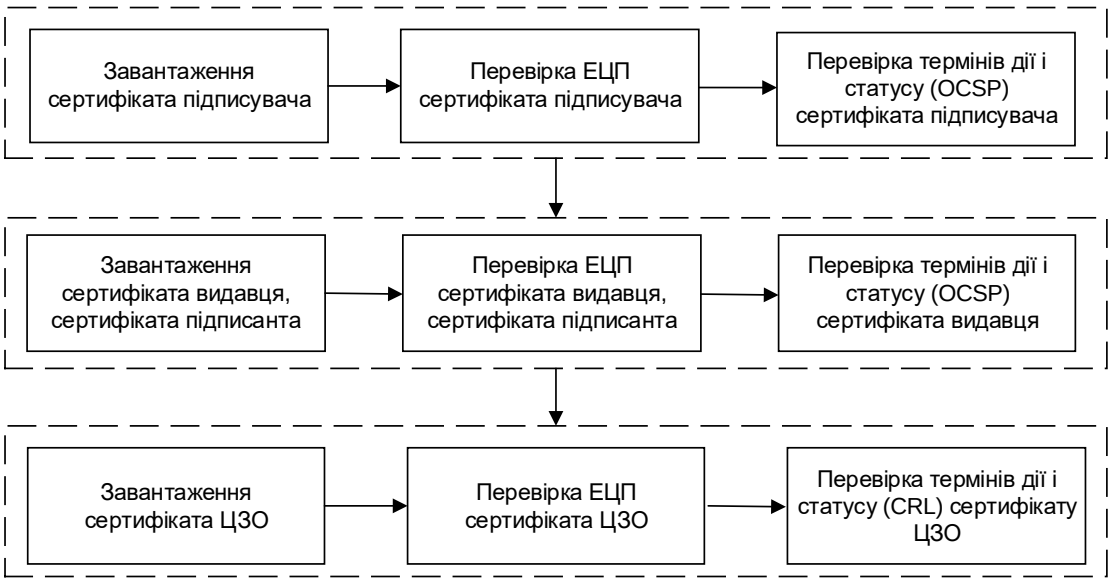

Рис.1.1.6. Процес перевірки ЕЦП документа

Іншими словами, процедура перевірки ЕЦП електронного документа в системі ІВК відбувається наступним чином. Спочатку перевіряється ЕЦП конкретного документа, а потім – ЕЦП сертифіката, за допомогою якого перевірявся попередній ЕЦП. Остання перевірка здійснюється доти, доки ланцюжок сертифікатів не призведе до кореневого.

При великій кількості перевірених ЕЦП на документах, з урахуванням повного ланцюжка, або якщо потрібно виконувати перевірку позначки часу, то час виконання операції ЕЦП стає критичним. А якщо відбуватиметься перевірка, наприклад, з ЦСК іншої країни через шлюз ЦЗО іншої країни, то цей ланцюжок подовжується. Для повноти критичності ситуації можна розглянути два приклади: закриття банківського дня або державних закупівель, тендерів і торгів.

Банки здійснюють перевірку даних, резервне копіювання і підготовку БД до наступного операційного дня. По-перше, відбувається перевірка великої кількості міжбанківських розрахункових документів: 300-500 тисяч з декількома ЕЦП та позначками часу (у разі необхідності). По-друге, при перевірці відбуваються запити на сервера ПТК OCSP: кількість в середньому дорівнює 1 млн. для кожного сервера. По-третє, відповіді на запити мають повертатися за короткий термін.

У зв’язку з цим, до НС ЕЦП висуваються наступні вимоги:

– Функціонування в масштабі часу близькому до реального (в режимі 24 години на добу), з урахуванням нерівномірного трафіку.

– Протидія від можливих атак «відмова в обслуговуванні» з боку хакерів.

– Та інші, які виходять за межі розгляду у роботі.

Завдання забезпечення оперативності обслуговування звернень до серверів ЦСК є актуальною як з апаратної, телекомунікаційної, так і з програмної точки зору.

Модель ІВК, яка розгорнута в Україні, виповідає ієрархії довірчих центрів сімейству стандартів X.509 [5-6], однак існує ще модель мережі довіри PGP [7]. Як подальший розвиток ІВК, слід розглядати технології, що будуються на деревах геш-функцій чи кодів автентифікації, які відомі під назвою ланцюжків блоків або блокчейн, що зберігаються у розподіленій БД [8-15].

Основна ідея полягає в зв’язуванні блоків геш-функцій у вигляді дерева. У якості блоків може виступати інформація про транзакції (запити на сертифікат, запити на відгук сертифікатів та самі сертифікати), представлені в деякій формі. Всі блоки організовані у ланцюжок, тобто пов'язані між собою. Для запису нового блоку чи заміні інформації в ньому, необхідно послідовне зчитування інформації про попередні блоки, тобто заміна лише даних одного блока веде за собою зміни інших блоків. Враховуючи розподіленість зберігання інформації у блокчейні та необхідність консенсусу, при внесення даних до блокчейну, то така модифікація стає неможливою.

Геш-функції або коди автентифікації в блокчейнах гарантують «незворотність» всього ланцюжка транзакцій. Справа в тому, що кожен новий блок транзакцій посилається на геш-образ попереднього блоку в реєстрі. Геш-образ самого блоку залежить від всіх транзакцій в блоці, але замість того, щоб послідовно передавати транзакції геш-функції, вони збираються в одне геш-значення за допомогою двійкового дерева з гешами (дерево Меркле) [16]. Тому не можливо змінити геш лише одного блоку, оскільки є результуючий геш, який складається з гешів блоків.

До ключових особливостей блокчейн можна віднести:

– Децентралізація - в ланцюжку немає єдиного сервера. Кожен учасник – це сервер.

– Прозорість – інформація про транзакції (запити на сертифікати, сертифікати та запити на відгук сертифікатів), зберігається у відкритому доступі. При цьому ці дані неможливо змінити.

– Теоретична необмеженість – теоретично блокчейн можна доповнювати записами до нескінченності. Однак з часом, при зростанні БД виявляються складності з експлуатацією такого сервера (вузла).

– Надійність – для запису нових даних необхідний консенсус вузлів блокчейна. Це дозволяє фільтрувати операції і записувати тільки легітимні транзакції. Здійснити підміну геша досить складно – необхідна можливість впливати на консенсус.

Розглянемо сучасні та перспективні алгоритми криптографічних перетворень з відкритим ключем у наступному підрозділі.

1.2. Огляд сучасних і перспективних криптографічних перетворень з відкритим ключем

Зараз активно використовуються схеми ЕЦП:

– на еліптичних кривих (ДСТУ 4145-2002, ECDSA, ECGDSA, ECKDSA, ГОСТ 34.10-2012, СТБ 34.101.45-2011) [4, 17, 18];

– на перетвореннях в полях та кільцях (DSA) [19];

– на перетвореннях в кільцях (RSA) [20].

Бурхливий розвиток математичного апарату та апаратних засобів, перед розробниками криптографічних алгоритмів ставить все нові і нові завдання (вимоги). Більшість поширених криптографічних алгоритмів з відкритим ключем, які використовуються при шифруванні та роботі з ЕЦП, засновані на проблемах розкладання великого числа на прості множники, дискретного логарифмування у полі чисел чи поліномів, дискретного логарифмування у групі точок ЕК і деякі інші математичні проблеми, які мають високу обчислювальну складність. Звичайні комп’ютери, з використанням найкращих відомих

алгоритмів, не здатні вирішити ці проблеми за розумний час. Тому більшість розробників схилилося до використання криптосистем заснованих на ЕК ще на початку 2-го тисячоліття.

Серед переваг криптосистем на ЕК над іншими криптосистемами [21]:

– генерація ключової пари виконується швидше;

– значно менша довжина ключів, при однаковій стійкості, займають менший обсяг пам’яті;

– обчислення виконуються швидше зі збереженням відповідного рівня стійкості;

– набагато більше джерело загальносистемних параметрів.

У табл. 1.2.1 наведено порівняння довжин ключів для криптосистем, які відповідають сучасним вимогам, згідно досліджень [22-25].

Таблиця 1.2.1

Порівняння довжин ключів для алгоритмів, що забезпечують однакову стійкість

Симетричний алгоритм AES	**RSA**	**DSA**	**ЕК**	**Геш-функції**
	Довжина відкритого, особистого ключа			
112	2048	224	224	SHA-224, SHA-512/224 SHA3-224
128	3072	256	256	SHA-256, SHA-512/256 SHA3-256
192	7680	384	384	SHA-384, SHA3-384
256	15360	521	521	SHA-512, SHA3-512

Криптосистеми на ЕК займають домінуюче місце в криптографії з відкритим ключем, що зумовлює подальші дослідження властивостей ЕК та криптографічних алгоритмів з їх використанням. Така активність може привести до появи нових, більш ефективних алгоритмів криптоаналізу. Слід зазначити, що спеціальний вибір типу ЕК дозволяє не тільки в багато разів ускладнити завдання криптоаналізу схеми ЕЦП, а й зменшити робочий розмір блоків перетворень, наприклад розмір елементів базового поля.

Так дослідження [26] показали, зростання використання криптографічних алгоритмів на ЕК та більш захищених кривих в мережі Інтернет, безпосередньо web-серверами та web-браузерами.

Деталізація значень таблиці: ecp224r1, secp256r1, secp384r1, secp521r1 – іменовані криві NIST для простого поля; x25519 – іменована крива у формі Монтгомері, яка використовується для узгодження ключів за Діффі-Хеллманом (ECDH), яка являється біраціонально еквівалентною до кривої у формі Едвардса Ed25519; Brainpool1256r1 – еліптична стандартизована крива Brainpool над 256 простим полем [27]; BASE – число серверів та робочих станцій (М - млн., К – тис.), які аналізувалися в експерименті; ECDHE – алгоритм узгодження ключів за Діффі-Хеллманом з динамічним узгодженням загально системних параметрів.

Таблиця 1.2.2

Підтримка web-браузерами та web-серверами сучасних протоколів з використанням різних елііптичних кривих

Кількість хостів, які підтримують зазначені криві										
Prot	**Port**	**Date**	**BASE**	**ECDHE**	**secp224r1**	**secp256r1**	**secp384r1**	**secp521r1**	**x25519**	**Brain-pool1256r1**
TSL	443	11/2016	38.6M	24.8M	643.4K (2.6%)	24.1M (97.0%)	5.7M (22.9%)	2.5M (10.2%)	0 (0.0%)	980.1K (3.9%)
	443	08/2017	41.0M	28.8M	811.6K (2.8%)	25.0M (86.9%)	9.1M (31.6%)	2.2M (7.7%)	740.7K (2.6%)	2.4M (8.4%)
SSH	22	11/2016	14.5M	7.9M	0 (0.0%)	7.7M (97.8%)	7.5M (95.6%)	7.5M (95.4%)	6.1M (77.2%)	0 (0.0%)
IKEv1	500	11/2016	1.1M	215.4K	143.8K (66.8%)	211.8K (98.3%)	206.8K (96.0%)	152.8K (771.0%)	0 (0.0%)	0 (0.0%)
IKEv2	500	11/2016	1.2M	101.1K	4.1K (4.1%)	98.2K (97.1%)	98.0K (96.9%)	240 (0.2%)	0 (0.0%)	0 (0.0%)

Використання застарілих реалізацій протоколу ранішніх версій, ніж TLS v1.2 та TLS v1.3, дозволяють використовувати вразливості протоколу TLS, під час утворення з’єднання, обираються застарілі алгоритми (DEA/TDEA) та загальносистемні параметри з слабкими властивостями (під час узгодження загальносистемних параметрів для ECDHE). Захиститися від таких вразливостей можливо за рахунок додаткових налаштувань web-серверів та використання протоколів TLS v1.2 та TLS v1.3, не нижче.

Таблиця 1.2.3

Квантові компп’ютери, моделі компютерів та їх характеристики

Рік	Розробник	Характеристики
2016	D-Wave Systems	2000-кубітний комп’ютер. Здатний вирішити лише вузького класу задачі. Кубіти згруповані між собою у невеликі групи, тому некоректно говорити про повноцінні 2000-кубітів.
2017	IBM	50-кубітний комп'ютер. Час когерентності системи досягло 90 мкс [30]. Всі кубіти пов’язані між собою у вигляді матриці 5x10.
2017	Михайло Лукин, (співзасновник Російського квантового центру і професор Гарварда)	51-кубітий комп’ютер побудований на холодних атомах рубідію [31]. Всі кубіти пов’язані між собою.
2018	Google	72-кубітний квантовий процесор Bristlecone [32]. Має низький рівень помилок під час обчислень. Всі кубіти пов’язані між собою у вигляді матриці 6×6, розташовані один над одним: дозволяє відстежувати і виправляти помилки, що виникають під час обчислень.

На даний момент не існує алгоритмів субекспоненційної складності для винайдення дискретного логарифму у групі точок ЕК над кінцевими полями, що дозволяє гарантувати стійкість криптосистеми на теперішній час. Як результат, пройшла стандартизація різноманіття криптосистем на ЕК: ISO/IEC CD 14883-3, ISO/IEC DIS 11770-3, ANSI ASC X.9.63, X.9.62, IEEE P1363-2000, ГОСТ 34.10-2012, СТБ 34.101.45-2011. Подальший розвиток теорії ЕК дозволив з’явитися таким криптосистемам на ЕК у формі Едвардса: ЕЦП EdDSA, PureEdDSA, HashDSA [28] та розподілу секрета: X25519, X448 [29]. Основна проблема при використанні таких криптосистем - необхідність підтримки ІВК, а також схильність до атак за сторонніми каналами [22], коли зловмисник, крім захищеного повідомлення або відкритого ключа, має додаткову інформацію, таку як: час виконання криптографічного перетворення; споживана потужність під час роботи; акустичні шуми, що виникають під час роботи; електромагнітне випромінювання, що генерується пристроєм і т.п. Оскільки основні операції в групах точок ЕК вимагають значних обчислювальних затрат, важливим для

подальшого розвитку даних криптосистем є зменшення обчислювальної складності арифметичних операцій в групах точок ЕК - модифікація існуючих криптосистем: зміна координат, використання інших кривих, для прикладу, Монтгомері, Едвардса.

Паралельно з класичними дослідженнями з криптоаналізу та удосконалення апаратних засобів, ведуться розробки квантових комп'ютерів. Вони відрізняються від класичних тим, що використовують для обчислень особливий тип бітів – кубітів. Кубіт – елемент пам'яті і одночасно примітивний обчислювальний модуль, який здатний зберігати в собі спектр значень між нулем та одиницею, тобто можуть одночасно знаходитися в декількох станах. Це дозволяє розробляти більш ефективні алгоритми вирішення складних задач (наприклад, набагато швидше розкладати числа на прості множники), причому ефективність квантового комп'ютера тим більше, чим більше кубітів в нього входить [30]. Потужність таких обчислювальних пристроїв зростає експоненціально від числа кубітів.

Проведемо аналіз існуючих квантових комп'ютерів та їх моделей, результати оформимо у вигляді табл. 1.2.3. Слід зазначити, що сучасні квантові комп'ютери не є універсальними і потребують певного налаштування зв'язків між кубітами для вирішення лише певної задачі.

При створені квантових комп'ютерів існує ряд перешкод, які на сьогоднішній день ще не вирішені:

– Приводити кубіти в певний вихідний стан.

– Об'єднувати їх в заплутані системи.

– Ізолювати кубіти від впливу зовнішніх завад.

– Зчитувати результати квантового розрахунку без зміни поточного стану кубіту.

– Забезпечити високу точність вимірювань.

– Забезпечити точність обчислень.

Головною перешкодою, що заважає побудувати комп'ютер з великим числом кубітів, являється помилки, які неминуче виникають при обчисленнях, зчитуванні й запису інформації в кубіти через руйнування їх квантового стану.

Чим більше кубітів, тим вище ймовірність, що кубіт стане взаємодіяти зі своїм «сусідом», і тим частіше виникають помилки.

На думку спеціалістів Google [32], щоб квантовий комп'ютер міг вирішити завдання, недоступні для «звичайних» комп'ютерів, потрібно дотримуватися таких умов: в його складі повинно бути не менше 49 кубів, глибина схеми повинна перевищувати 40 кубів, а ймовірність помилки в двокубітному логічному елементі повинно бути не вище 0,5%. Для побудованого комп’ютера ці вимоги виконуються, за виключенням долі помилок (вона становить 0,6%).

Тому квантова криптографія залишається на сьогоднішній день перспективою, бо не має поточного прикладного значення, однак при розробці перспективних криптографічних алгоритмів і стандартів слід враховувати можливість появи таких квантових комп’ютерів. В роботі [15] проведений аналіз теоретичної оцінки уразливості відомих криптосистем з відкритим ключем до квантового криптоаналізу (див. табл. 1.2.4).

Таблиця 1.2.4

Поточний стан безпеки класичних криптосистем стосовно квантових комп'ютерів

Криптосистеми	**Вразливість до квантового криптоаналізу**
Шифрування RSA з відкритим ключем	так
Обмін ключами Діффі-Хеллмана	так
Криптографія на еліптичних кривих	так
Обмін ключами Бухмана-Вільямса	так
Алгебраїчно гомоморфна	так
Шифрування відкритого ключа McEliece	На сьогодні - ні
Шифрування відкритого ключа NTRU	На сьогодні - ні
Шифрування відкритих ключів на основі решіток	На сьогодні - ні

Слід замітити, алгоритм Шора може бути використаний для злому криптографічного алгоритму на ЕК шляхом обчислення дискретних логарифмів на гіпотетичному квантовому комп’ютері (симуляторі). Останні оцінки квантових ресурсів для злому кривої з 256-бітовим модулем (128-бітний рівень безпеки) складають 2330 кубітів [33]. Для порівняння, використовуючи алгоритм

Шора для злому алгоритму 2048-бітної RSA, потрібно 4098 кубітів, що дозволяє вважати, що криптографія на ЕК є більш уразливою для квантових комп'ютерів, ніж RSA. До вказаної кількості кубітів не наблизився ні один квантовий комп'ютер і не зможе наблизились у найближчі 20 років. Справа в тім, що при збільшенні кількості кубітів складність підтримки у стабільному стані всієї системи зростає експоненційно.

В якості протидії квантовоим методам криптоаналізу, пропонується використання ізогенії груп на кривих. Цей підхід використовує значну частину вже існуючого математичного апарату ЕК та алгоритмів перетворень у полях і групі точок ЕК. В свою чергу, це суттєво спрощує використання ізогенії груп на кривих замість існуючих криптосистем на ЕК.

1.3. Формалізація постановки задачі досліджень

Розглянемо ефективність функціонування НС ЕЦП на основі реалізації криптографічних транзакцій.

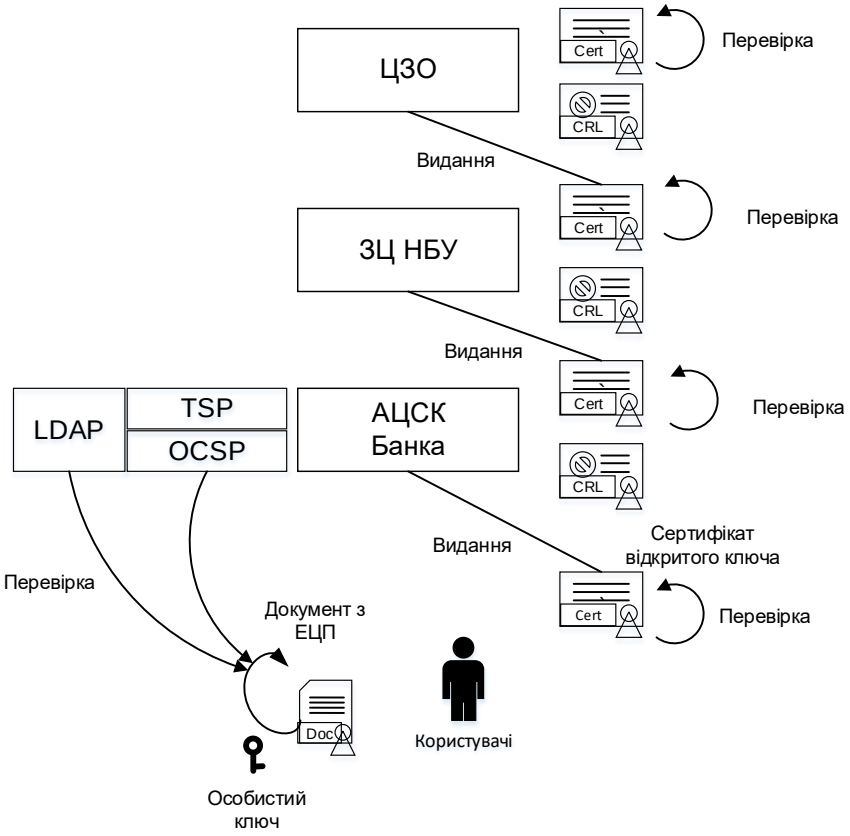

Рис.1.3.1. Сервіси і об'єкти АЦСК банків, які використовуються для перевірки ЕЦП

Для цього розглянемо приклад банківського дня на рис. 1.3.1. Операція перевірки ЕЦП складається з наступних операцій:

1. Отримання/завантаження сертифікату відкритого ключа підписувача, перевірки його цілісності (ЕЦП) та відповідності загальносистемним параметрам.

2. Перевірка статусу сертифікату підписувача за СВС чи за OCSP запитом, що включає:

2.1. За СВС: завантаження СВС за адресою, що вказана у сертифікаті підписувача; перевірка його цілісності (за допомогою ЕЦП); пошук за ідентифікатором у списку (може налічувати десятки або і сотні тисяч записів). Далі відбувається перевірка всього ланцюжка сертифікатів, аж до кореневого – самопідписаного (ЦЗО України).

2.2. За OCSP: відправка запиту до серверу OCSP, вказавши ідентифікатор сертифікату, що вказаний у сертифікаті підписувача; отримання відповіді від OCSP та перевірка ЕЦП на відповіді, що включає завантаження сертифікату OCSP з відповідного сховища, або вилучення цього з OCSP відповіді. Далі відбувається перевірка всього ланцюжка сертифікатів, аж до кореневого – самопідписаного (сертифікат ЦЗО України).

3. Обчислення геш-функції та самої математичної операції перевірки ЕЦП.

Таким чином час виконання транзакцій з ЕЦП можна записати наступним чином: $T = T_G + T_V$: де $T_G = \sum_{i=1}^{n} \left(T_i^{\text{EDS}} + T_i^T \right)$ – час транзакцій, пов’язаний з формуванням ЕЦП; $T_V = \sum_{j=1}^{m} \left(T_j^{\text{EDS}} + T_j^T \right)$ – час транзакцій, пов'язаний з перевіркою ЕЦП; m – кількість запитів для перевірки ЕЦП; n – кількість накладання ЕЦП; T_i^{EDS} – час криптографічних транзакцій, пов’язаних з ЕЦП; T_i^T – час на передачу інформації по каналам зв’язку.

Транзакції ЕЦП: формування ЕЦП $T_G^{EDS} = T_H + T_{SM} + T_F$, перевірка ЕЦП $T_V^{EDS} = T_H + 2T_{SM} + T_F$, де T_H, T_{SM}, T_F – час виконання операції гешування (ГОСТ 34.311-95) [4], виконання СМ та польових операцій в алгоритмах формування і перевірки ЕЦП згідно ДСТУ 4145-2002.

Таблиця 1.3.1

Час виконання складових операцій при формуванні та перевірці ЕЦП

Геш-функція	**Гешування, МБ/с**	**Скалярне множення, мс**		
		Поле, m	**ДСТУ 4145-2002**	**ECDSA**
ГОСТ 34.311-95	34.87	163	1.0624	1.1021
ДСТУ 7564:2014-256	101.79	167	1.4464	
ДСТУ 7564:2014-384	75.62	173	0.8057	
ДСТУ 7564:2014-512	75.59	179	1.5222	
SHA-1	422.30	191	1.1709	
SHA-224	148.81	233	2.3468	3.0743
SHA-256	148.81	257	3.1619	
SHA-384	244.14	283		3.7848
SHA-512	256.15	307	2.758	
SHA-512/224	252.01	367	5.2973	
SHA-512/256	260.42	409		8.5287
		431	6.5643	
		571		13.218

В табл. 1.3.1 представлений час реалізації стандартизованих функцій гешування з розміром даних 16 кБ та СМ, реалізованих за алгоритмами ДСТУ 4145-2002 та ECDSA (NIST криві [19]) в проективних координатах Лопеса-Дахаба, використовуючи метод Монтгомері та бінарний алгоритм приведення за модулем. Заміри часу проводилися для виконання 1 млн. операцій, за допомогою обчислювальної системи з процесором Intel Core i7-6700 2,60 GHz під управлінням ОС Windows 10 x86-64 на мові високого рівня C++(Visual C++2015).

Виходячи з даних, зрозуміло, що у більшості випадків практичного застосування ЕЦП час виконання СМ переважає над гешуванням. Тому підвищення швидкодії криптографічних перетворень на ЕК можливе за рахунок зменшення обчислювальної складності операції СМ.

Розглянемо проект EBATS (ECRYPT Benchmarking of Asymmetric Systems) створений в лабораторії VAMPIRE компанії ECRYPT для вимірювання швидкодії систем з відкритим ключем [34]. В табл. 1.3.3 [34] представлені результати тестів, зібрані в eBATS для систем підпису з відкритим ключем.

Введемо позначення: G – генерація ключової пари, S – формування ЕЦП, V– перевірка ЕЦП, відповідно. Вимірювання представлені в циклах для кожного з процесорів з характеристиками представленими в табл. 1.3.2.

Таблиця 1.3.2

Характеристика процесорів

Процесор	**Архітектура**	**Частота (GHz)**	**Реалізація**
Intel Core i5-6600	x86-64	3,31	Desktop
Intel Core i5-4210U	x86-64	1,7	Notebook
MediaTek MT8173	aarch64	2,1	Embedded

Таблиця 1.3. 3

Швидкодія реалізації генерації ключової пари, накладення та перевірки ЕЦП

	Intel Core i5-6600			**Intel Core i5-4210U**			**MediaTek MT8173**		
	G	**S**	**V**	**G**	**S**	**V**	**G**	**S**	**V**
ed25519	54780	49840	163206	479124	477792	1258434	160030	155806	415001
ecdonaldp256	90994	162978	308510	3958104	4318044	4875396	2323885	2525562	2874091
ecdonaldp224	138370	203656	407532	3437010	3746004	4256418	1781491	1966316	2210451
ed448goldilock	153060	160778	498662				497571	507838	1578376
ecdonaldp160	560146	603922	690516	2290308	2540382	2864274	1328804	1464517	1626392
ecdonaldp521	618168	993048	1770262	14596212	15277614	17799438	6540417	7654197	7892674
ecdonaldb163	644280	688828	1304988	2718930	2968968	5531340	1645386	1787152	3297614
ecdonaldp192	665062	711180	819746	2707236	2986716	3362994	1384057	1523095	1701407
ecdonaldb233	824940	897130	1681808	3529488	3873858	7266282	2361340	2569744	4734629
ecdonaldb283	1466316	1577018	2995814	6134418	6554634	12597930	5106845	5429921	10208910
ecdonaldp384	2045818	2186682	2527774	7512612	8066982	9266538	4646482	5149321	5668315
ecdonaldb409	2332984	2541942	4780078	10086852	10661970	20649624	10762611	11432266	21602541
ecdonaldb571	5522398	5938954	11256154	22194768	23014716	45192336	23167867	24413986	46626430
ronald2048	216077924	3473634	51562	684824682	15329400	256806	397719604	12089394	119500
ronald3072	666048194	8733058	84562	2295480732	41234502	387192	1616851613	34468532	218212
ronald4096	1203088092	17810266	135796	6047456094	85741968	567588	4518240685	70724855	326792

Криптографічні перетворення, які покладені в основі ДСТУ 4145-2002 та ECDSA, являються перетвореннями на ЕК. Вони базуються на операції СМ точок ЕК (рис. 1.3.2), до якої входять операції додавання і подвоєння точок ЕК, що в свою чергу виконуються над координатами точок – елементами двійкового поля. З точки зору процесора, операції над координатами точок – поліномами, представляються у вигляді машинних слів фіксованої довжини: додавання,

віднімання, множення, ділення, інвертування, піднесення до степеню, видобування квадратного кореня тощо.

Таблиця 1.3.4

Опис криптографічних примітивів для ЕЦП

Криптографічні примітиви	**Опис**
ed25519	ЕЦП згідно EdDSA з використання кривої Curve25519
ecdonaldb163	ЕЦП згідно ECDSA з використанням кривої NIST B-163: поліном, що не приводиться $x^{163}+x^7+x^6+x^3+1$.
ecdonaldb233	ЕЦП згідно ECDSA з використанням кривої NIST B-233: поліном, що не приводиться $x^{233}+x^{74}+1$.
ecdonaldb283	ЕЦП згідно ECDSA з використанням кривої NIST B-283: поліном, що не приводиться $x^{238}+x^{12}+x^7+x^5+1$.
ecdonaldb409	ЕЦП згідно ECDSA з використанням кривої NIST B-409: поліном, що не приводиться $x^{409}+x^{87}+1$.
ecdonaldb571	ЕЦП згідно ECDSA з використанням кривої NIST B-571: поліном, що не приводиться $x^{571}+x^{10}+x^5+x^2+1$.
ecdonaldp160	ЕЦП згідно ECDSA з використанням кривої RFC 5639 SECP160R1: поліном, що не приводиться $2^{160}+2^{31}+1$.
ecdonaldp192	ЕЦП згідно ECDSA з використанням кривої NIST P-192: поліном, що не приводиться $2^{192}-2^{64}-1$.
ecdonaldp224	ЕЦП згідно ECDSA з використанням кривої NIST P-224: поліном, що не приводиться $2^{224}-2^{96}+1$.
ecdonaldp256	ЕЦП згідно ECDSA з використанням кривої NIST P-256: поліном, що не приводиться $2^{256}-2^{224}+2^{192}+2^{63}-1$.
ecdonaldp384	ЕЦП згідно ECDSA з використанням кривої NIST P-384: поліном, що не приводиться $2^{384}-2^{128}-2^{96}+2^{32}-1$.
ecdonaldp521	ЕЦП згідно ECDSA з використанням кривої NIST P-521: поліном, що не приводиться $2^{521}-1$.
ed448goldilocks	ЕЦП згідно EdDSA з використанням Ed448-Goldilocks для підписання і вироблення загального секрету.
ronald2048	ЕЦП з відновленням повідомлення згідно RSA з довжиною ключа 2048-біт.
ronald3072	ЕЦП з відновленням повідомлення згідно RSA з довжиною ключа 3072-біт.
ronald4096	ЕЦП з відновленням повідомлення згідно RSA з довжиною ключа 4096-біт.

Тому щоб підвищити швидкість формування і перевірки ЕЦП, потрібно зменшити обчислювальну складність і підвищити швидкодію основної операції СМ за рахунок: перетворень в проміжних обчисленнях базової та довільної точки ЕК, а також обчислювальної складності арифметичних операцій над числами і поліномами. А саме ділення, приведення за модулем та інвертування, здобуття кубічного кореня (для переходу від кривих Вейерштрасса до кривих Едвардса).

Криптографічними перетвореннями на ЕК, а також підвищенням швидкодії СК на ЕК займаються такі вчені як Бернштейн (D. J. Bernstein), Ланге (T. Lange), Козєл (B. Koziel), Молоні (R. Moloney), Горбенко І.Д., Бессалов А.В. та інші.

Велике число публікацій [35-41], присвячених оптимізації операцій множення і приведення по модулю, не приділяють належної уваги операціям ділення. Дана операція досить часто використовується на другому плані – при генерації загальносистемних параметрів RSA, при переході до модуля Монтгомері тощо.

<table>
<tr><td>Криптографічні перетворення</td><td colspan="3">Шифрування/розшифрування</td><td colspan="3">Формування і перевірка цифрового підпису</td><td colspan="2">Обмін ключами</td></tr>
<tr><td rowspan="2">Арифметика в групі точок еліптичної кривої</td><td colspan="5">Скалярне множення точок еліптичної кривої</td><td colspan="3" rowspan="2">Генерація випадкової точки</td></tr>
<tr><td colspan="4">Додавання точок</td><td colspan="2">Подвоєння точок</td></tr>
<tr><td>Арифметика в полі GF(p), GF(2^m)</td><td>Множення</td><td>Складання</td><td>Ділення</td><td>Піднесення до квадрату</td><td>Приведе-ння по модулю</td><td>Інвертування</td><td colspan="2">Добування квадратного кореня</td></tr>
<tr><td>Операція над масивами</td><td colspan="2">Зсув</td><td colspan="2">Порівняння</td><td>Додавання</td><td>Віднімання</td><td colspan="2">Множення</td></tr>
<tr><td>Команди CPU</td><td colspan="8">mov, mul, shr, shl, add, sub</td></tr>
</table>

Рис.1.3.2. Ієрархія операцій при криптографічних перетворень з відкритим ключем

Аналіз існуючих робіт по операціям ділення та приведення по модулю, дозволив *доповнити класифікацію* [42] у напрямку алгоритмів ділення (рис. 1.3.3 і рис. 1.3.4) із залишком та без нього, на основі відомих алгоритмів: Баррета, Монтгомері, Джебеліна, Класичного, Ньютона, що враховує різні аспекти самих алгоритмів, спрямованих на їх оптимізацію.

1. Подання цілого числа (діленого та дільника):

– Двійкове (звичайне) [39].

– Четвертне [39].

– Двійкове скорочене (двійкове DRF) [38].

– Надмірне статичне [38, 43].

– Надмірне динамічне [43].

– Несуміжних (NAF) [44].

– Залишкових класів (RNS) [45].

– Двійкове, з відкладеним перенесенням (DCF) [42].

– Змішане.

– ZOT-представлення [46].

2. Дільник – довільні числа, не обов'язково прості. Алгоритми засновані на ідеї ділення «в стовпчик», Баррета і Монтгомері [47].

3. За одержуваних результатів. При ділені цілих чисел можуть представляти інтерес, як ціле, так і залишок. У зв'язку з цим можна виділити наступні алгоритми:

– Ділення цілих чисел для отримання тільки цілого.

– Ділення цілих чисел для отримання тільки залишку – приведення по модулю.

– Ділення цілих числі для отримання, як цілого, так і залишку.

4. Дільник – прості числа. З огляду на специфіку представлення дільника (модуля) в двійковому вигляді, можна оптимізувати алгоритм приведення, згідно з такими класами простих чисел:

– Прості числа загального вигляду. Алгоритми засновані на ідеї ділення «в стовпчик», Баррета і Монтгомері [47].

– Прості числа – узагальнені мерсенові і псевдо-мерсенові. Спеціалізовані алгоритми, що враховують властивості мерсенових і псевдо-мерсенових конкретних та в загальному вигляді чисел [48-51].

5. Напрямок аналізу діленого. Алгоритми, що враховують специфіку та напрямки аналізу діленого з початку або кінця:

– З початку (з молодших бітів) в алгоритмі Монтгомері [52-53].

– З кінця (зі старших бітів) в алгоритмі Баррета [52].

– Двосторонні та багатосторонні [52].

6. Модифікації алгоритму. З огляду на різні підходи до ділення і приведення, виділяють кілька основних алгоритмів та їх модифікацій [54]:

– Класичний («шкільний», ділення «в стовпчик») [35, 55-56].

– Монтгомері [37] та його модифікації [38].

– Баррета [36] та його модифікації [38, 56].

– Універсальний і жорстко запрограмований спеціальний дільник (модуль). Як правило, застосовується для простих модулів спеціального виду, наприклад, мерсенових, псевдо-мерсенових або узагальнених мерсенових чисел [48-49].

– Алгоритм ітерацій Ньютона [39, 57].

– Алгоритми розподілу Джебеліна [40-41] та його модифікації.

7. Точність отримання результату - наближення. Не завжди становить інтерес точний результат (ціле/залишок), досить отримати результат, що не сильно відрізняється від точного значення. Даний підхід може бути застосований до всіх відомих алгоритмів приведення по модулю. Можна виділити кілька основних алгоритмів:

– Часткове приведення для алгоритму Монтгомері [52].

– Часткове приведення для алгоритму Баррета [52].

– Часткове приведення для алгоритму ділення «в стовпчик».

– Часткове приведення для модуля спеціального виду.

8. Розпаралелювання. Більшість сучасних комп'ютерів володіють декількома процесорами або одним процесором з декількома ядрами. Відомі такі алгоритми з розпаралелюванням ділення (приведення по модулю):

– Двостороннє часткове приведення для алгоритму Монтгомері [52, 58].

– Багатостороннє часткове приведення для алгоритму Монтгомері [52, 58].

– Двостороннє часткове приведення для алгоритму Баррета [52, 58].

– Багатостороннє часткове приведення для алгоритму Баррета [52, 58].

– Приведення для алгоритму Баррета на основі DCF [42]. У даній роботі підлягає розпаралелюванню алгоритм Комба - часткового множення цілих чисел, що дозволяє його застосовувати не тільки до алгоритму Баррета, а й Монтгомері.

– Розпаралелювання алгоритму Карацуби, запропоноване Джебеліном [40-41, 59].

– Алгоритм розпаралелювання представлення багаторазової точності через одинарну точність цілого числа: паралельний циклічний метод приведення за модулем [57].

9. Попереднє обчислення. При виконанні операції ділення (приведення по модулю), використовується один і той же дільник (модуль), то становить інтерес перечислити заздалегідь деякі проміжні значення. Даний підхід може бути застосований до всіх відомих на сьогодні алгоритмів.

Механізм отримав розвиток в наступних алгоритмах:

– Одноразове перекриття алгоритму Баррета [38].

– Ітеративне перекриття алгоритму Баррета [50].

– Використання одного або декількох передобчислених значень (алгоритм Баррета, Монтгомері, Класичний) [60].

– Таблиця передобчислень для простого загального вигляду (велика) (алгоритм Баррета, Монтгомері, Класичний) [60].

– Таблиця передобчислень для простого спеціального виду DRF (велика) (алгоритм Баррета, Монтгомері, Класичний) [60].

– Таблиця передобчислень для модуля спеціального виду DRF (маленька) (алгоритм Баррета, Монтгомері, Класичний) [60].

10. Без передобчислення констант. З огляду на той факт, що в алгоритмах Монтгомері і Баррета, необхідно попереднє обчислення ряду констант, існує можливість піти від попереднього обчислення констант:

– Без обчислень констант для алгоритму Монтгомері (алгоритм масштабування Куіскуотера) [50].

– Без обчислень констант для алгоритму Баррета (алгоритм масштабування Куіскуотера) [50].

11. Алгоритм множення. Слід зауважити, що в алгоритмах ділення «в стовпчик» і ітерацій Ньютона, ділення і приведення по модулю Баррета і Монтгомері, використовується множення цілих чисел різної довжини, від реалізації яких, безпосередньо залежить швидкодія самого ділення і приведення. Відомо застосування алгоритмів множення:

– алгоритм Комба [61].

– алгоритм Карацуби [62].

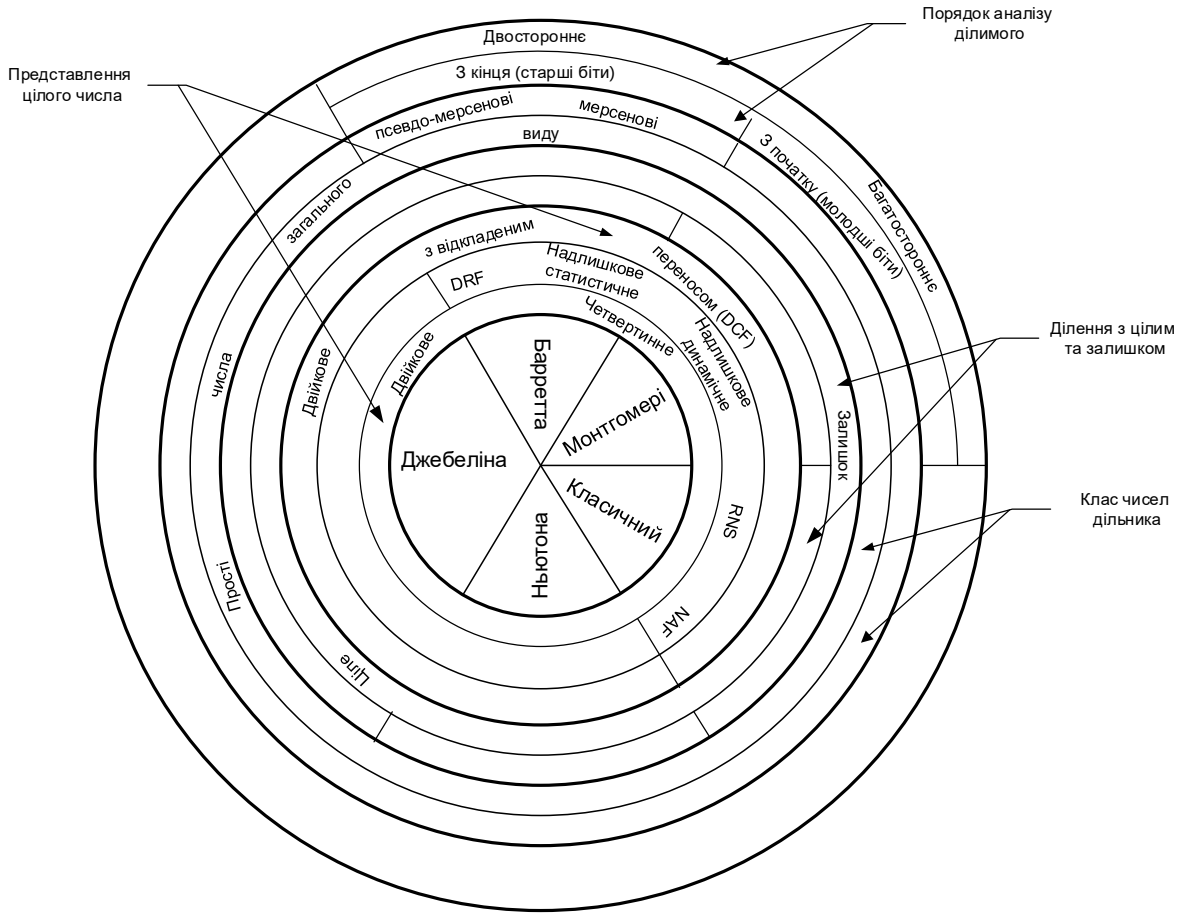

Рис.1.3. 3. Класифікація алгоритмів поділу

Класифікація алгоритмів ділення і приведення по модулю дає можливість виконувати передобчислення і використовувати наближені операції. А також вона може допомогти при виборі самого відповідного алгоритму при розробці криптосистеми, коли перед розробником ставиться певна задача і визначаються умови її реалізації.

Розвиток обчислювальної техніки дозволяє знаходити практичне застосування різних математичних теорій. До них можна сміливо віднести елементи абстрактної алгебри, до якої відносяться теорії полів, кілець, груп і ідеалів. Так поля виду $\boldsymbol{GF}\left(2^m\right)$ широко використовуються в різних областях сучасної обчислювальної техніки, пов'язаних з прийомом, передачею і обробкою цифрової інформації. Наприклад, завадостійке кодування (коди Ріда-Соломона, Гоппе і ін.), криптографія (криптосистеми засновані на вирішенні завдання дискретного логарифма в групах цілих чисел, полів і точок еліптичних кривих) тощо. З іншого боку, поля виду $\boldsymbol{GF}\left(2^m\right)$ зручні з точки зору програмної і

апаратної реалізації, яка має фіксоване число розрядів, необхідних для представлення даних і зберігання в пам'яті. Незважаючи на досить тривале використання двійкових полів в техніці, актуальність підвищення ефективності операцій над елементами поля $\boldsymbol{GF}\left(2^m\right)$ не викликає сумнівів. Це підтверджується появою спеціальних інструкцій PCLMULQDQ/CLMUL для множення елементів поля в сучасних процесорах сімейства x86/ARM.

Перейдемо до операції інвертування, яка являється не часто затребуваною, але трудомісткою і складною. Щоб зменшити кількість операцій інвертування при реалізації, пропонується перехід від афінних координат до проектних або використання операції множення (одне інвертування замінюється трьома операціями множення). Проведений огляд публікацій операції інвертування в двійковому полі дозволив виділити ряд наступних алгоритмів:

1. Алгоритм на основі Малої теореми Ферма (використовується операція exp)[65].

2. Алгоритм Ітo-Цуйі (суть алгоритму аналогічна попередньому, проте завдяки ряду модифікацій скоротилася кількість операцій множень [65-67].

3. Алгоритм матричних операцій над поліномами [68].

4. Існує ряд модифікацій алгоритму Ітo-Цуйі (при поданні показника степеню використовується не двійкова система числення, а використовуються змішані представлення показника степеню) [67, 69].

5. Алгоритм інвертування на основі розширеного алгоритму Евкліда [70].

6. Метод адитивних ланцюгів в операції інвертування в кінцевих полях:

– Адитивні ланцюжки та метод Брауера.

– Бінарний алгоритм та метод Ітo-Цуйі [66, 69].

Ще однією трудомісткою і затребуваною операцією являється приведення за модулем, яка відбувається після кожного піднесення полінома до степеню чи множення:

– Побітовий алгоритм приведення за довільним модулем [71].

– Послівний алгоритм приведення за фіксованим модулем [71-73]. В публікаціях викладається загальна ідея або алгоритми для конкретних поліномів

і відсутній чіткий метод побудови алгоритму приведення для довільного модуля (полінома, що не приводиться).

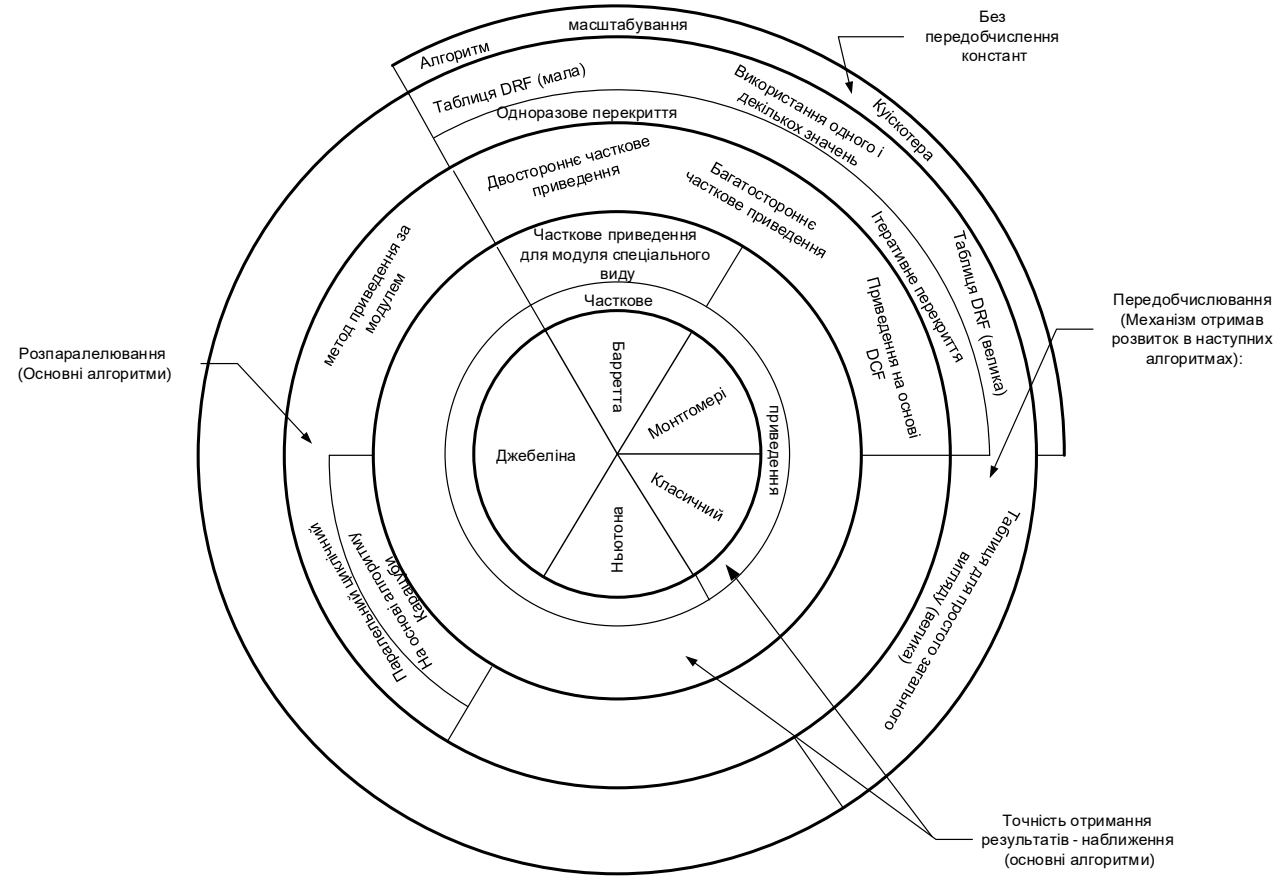

Рис.1.3. 4. Класифікація алгоритмів поділу

Що стосується трудомісткої операції СМ, то існує безліч алгоритмів її реалізації:

– Бінарне СМ (скануючи біти скаляра в режимі зліва направо (left-to-right) або справа наліво (right -to-left) [75].

– Віконний метод (windowed method) і його модифікації [76-78].

– Метод ковзного вікна (sliding-window method) і його модифікації [78-79].

– w-ray метод несуміжних форм (wNAF) і його модифікації [76-78].

– Метод Монтгомері (Montgomery ladder) і його модифікації [74, 76, 80-84].

Але найефективнішим являється метод Монтгомері, оскільки його реалізація стійка до атак по стороннім каналам (клас атак, спрямований на уразливості в практичній реалізації криптосистеми) [85-86].

Опір до побічного каналу витоків пов'язаний з тим, що на кожному кроці методу Монтгомері, незалежно від поточного біта скаляра k, виконується

операція подвоєння однієї точки, а також операція додавання точок і змінюється тільки порядок виконання даних дій, що не дозволяє відновити весь показник (зокрема секретний ключ), оскільки неможливо визначити порядок операцій. Існують криптографічні реалізації, запущені на серверах, де можуть виконуватися віддалені синхронізуючі атаки - тимчасові. Ця проблема знову може бути вирішена за допомогою метода Монтгомері, оскільки час реалізації є постійним для фіксованих скалярних довжин.

Також вчені провели аналіз стандартизованих кривих NIST Вейерштрасса [88-89], де задали багато питань з приводу вибору стандартизованих кривих та висунули певні умови:

– захист ECDLP не гарантує безпеку ECC;

– легкість в програмній реалізації;

– вимагають сумісність з кривими Монтгомері;

– вимагають сумісність з кривими Едвардса;

– вимагають повноту;

– безпека від скручування;

– запропонували стійкі еліптичні криві Едвардса для простого та двійкового полів.

Програмна реалізація операції СМ на стандартизованих кривих Вейерштрасса охоплює ряд ситуацій: наприклад, при додаванні точок $P+Q$, якщо точка P або Q належить нескінченності, або якщо сам результат $P+Q$ являється точкою на нескінченності, або $P=Q$. Кожна з цих можливостей повинна бути перевірена і аналізуватися окремо, на що витрачається додатковий час. Алгоритм повного складання виходить шляхом склеювання декількох неповних формул складання [74].

В роботі [74] автори представили перехід до більш захищених кривих Едвардса, які володіють бажаними криптографічними властивостями і перевагами, в порівнянні з типовою формою Вейерштрасса:

– груповий закон для кривих Едвардса - повний і уніфікований;

– більш безпечні реалізації до атак по стороннім каналам;

– мають ефективне складання та подвоєння точок;

– закон складання точок для кривих Едварда дозволяє виключити різні перевірки, властиві при додаванні точок ЕК в формі Вейерштрасса (наприклад формула складання ідентична для P і P, а також P і $-P$);

– кожна ЕК в формі Вейерштрасса над двійковим полем має біраціонально еквівалентну повну криву Едвардса $E_{B,\,d_1,\,d_2}$ для $m \geq 3$.

Все вище перераховане дозволяє кривим Едвардса забезпечувати кращу платформу для побудови криптографічних примітивів.

Нижче представлена двійкова крива Вейерштрасса [74, 81]:

$$v^2 + uv = u^3 + a u^2 + b \tag{1.3.1}$$

Двійкова крива Едвардса, при $d_1 \neq d_2$ має вигляд [74, 81]:

$$d_1(x + y) + d_2\left(x^2 + y^2\right) = xy + xy(x + y) + x^2 y^2, \tag{1.3.2}$$

де d_1 і d_2 - коефіцієнти кривої, які задовольняють умови $d_1 \neq 0$ і $d_2 \neq d_1^2 + d_1$.

Крива (1.3.2) біраціонально еквівалентна до кривої Вейерштрасса (1.3.1):

$$v^2 + uv = u^3 + \left(d_1^2 + d_2\right)u^2 + d_1^4\left(d_1^4 + d_1^2 + d_2^2\right), \tag{1.3.3}$$

де умова $d_1^4\left(d_1^4 + d_1^2 + d_2^2\right) \neq 0$ вказує на не суперсингулярність кривої.

Для перетворення точки Вейерштрасса в точку Едвардса використовують наступну маску [74, 81]:

$$x = \frac{d_1\left(u + d_1^2 + d_1 + d_2\right)}{u + v + \left(d_1^2 + d_1\right)\left(d_1^2 + d_1 + d_2\right)},\ y = \frac{d_1\left(u + d_1^2 + d_1 + d_2\right)}{v + \left(d_1^2 + d_1\right)\left(d_1^2 + d_1 + d_2\right)} \tag{1.3.4}$$

Зворотнє відображення має вигляд: $(x, y) \mapsto (u, v)$:

$$u = d_1\left(d_1^2 + d_1 + d_2\right)\frac{x + y}{xy + d_1(x + y)},\ v = d_1\left(d_1^2 + d_1 + d_2\right)\left(\frac{x}{xy + d_1(x + y)} + d_1 + 1\right) \tag{1.3.5}$$

При умові $d_1 = d_2$, в роботі [90] описані умови представлення повної кривої Едвардса, як біраціонально еквівалентній кривій Вейерштрасса (1.3.1) $E_{B,\,d_1,\,d_2}$, при умові, що $\mathrm{Tr}(b) = 1 = \mathrm{Tr}(d_1)$ з ізоморфізмом $v \to v + d_1 u$:

$$d_1\left(x + y + x^2 + y^2\right) = xy + xy(x + y) + x^2 y^2 \tag{1.3.6}$$

де коефіцієнти ізоморфної кривої Вейерштрасса $a = d_1 + d_1^2$ та $b = d_1^8$.

Перетворення точки кривої $E_{W,\,d_1+d_1^2,\,d_1^8}$ в $E_{B,\,d_1,\,d_1}$ і навпаки відбувається за допомогою масок [90]:

$$(x,y)\to(u,v)=\left(\frac{d_1^3(x+y)}{xy+d_1(x+y)};\frac{d_1^3 x}{(xy+d_1(x+y))+d_1+1}\right) \tag{1.3.7}$$

$$(u,v)\to(x,y)=\left(\frac{d_1\left(u+d_1^2\right)}{u+v+\left(d_1^2+d_1\right)d_1^2};\frac{d_1\left(u+d_1^2\right)}{v+\left(d_1^2+d_1\right)d_1^2}\right) \tag{1.3.8}$$

Двійкова крива Едвардса з одним параметром цікава тим, що дозволяє значно скоротити кількість арифметичних операцій при додаванні та подвоєнні точок ЕК, тим самим збільшуючи швидкодію СМ. Це пов'язано з тим, що у даного виду кривої існує лише один нейтральний елемент – точка $(0,0)$- яку приймають як точку на нескінченності при реалізації СМ (у кривої відсутні точки на нескінченності). Оскільки всі точки кривої Едвардса з одним коефіцієнтом – унікальні, результат при СМ $(0,0)$ ніколи не виходить, тому це дозволяє не проводити витратні перевірки, що дозволяє значно економить час реалізації.

Алгоритми пошуку біраціонально еквівалентних кривих, представлені в роботі [74, 93]. Обидва алгоритми – ймовірні, однак на відміну від алгоритму [74], який використовує квадратний корінь, для отримання параметрів шуканої кривої Едвардса необхідно зробити значне число ітерацій: оскільки $\boldsymbol{P}(\mathrm{Tr}(\mathrm{d}_1)=\mathrm{Tr}(a)+1)=1/2$ п. 1.1 повинен виконуватися в середньому 2 рази; а також слід зауважити, що якщо $\mathrm{Tr}\left(\sqrt{b}/d_1^2\right)\neq 1$, то потрібно повернутися до п. 1.1 Алгоритм [93] позбавлений даних складнощів і дозволяє майже завжди на першій ітерації отримати параметри шуканої кривої Едвардса, причому на кожній ітерації алгоритму слід виконувати менш складні математичні операції. Даний алгоритм використовує операцію здобуття кубічного кореня елемента в двійковому полі [94], яка значно впливає на час реалізації алгоритму. Тому в дисертаційній роботі пропонується *удосконалення* методу здобуття кубічного кореня для пришвидшення реалізації пошуку біраціонально еквівалентних кривих.

Для побудови високопродуктивного ПТК криптографічного захисту інформації системи ЕЦП, актуальним завданням являється підвищення швидкодії ІТС ЦСК Національної системи ЕЦП за рахунок *удосконалення методів та алгоритмів арифметичних* перетворень над числами, поліномами і точками ЕК зі зменшеною обчислювальною складністю і протидією до атак на їх реалізацію.

1.4. Висновки до першого розділу

На основі аналізу проведеного в першому розділі, можна зробити наступні висновки:

1. Сервера ЦСК (АЦСК) та ЦЗО НС ЕЦП обробляють велику кількість запитів (криптографічних транзакцій) по формуванню і перевірці ЕЦП, через що висуваються до них такі вимоги: робота в режимі реального часу з нерівномірним трафіком - вимоги до продуктивності системи ЕЦП.

2. Швидкодія формування та перевірки ЕЦП напряму залежить від часу реалізації СМ. Тому для пришвидшення необхідно удосконалити арифметичні операції в полі.

3. Підвищити швидкодію і рівень стійкості криптографічних перетворень з відкритим ключем в ПТК криптографічного захисту інформації можливо за допомогою переходу від криптоперетворень на ЕК в формі Вейерштрасса до криптоперетворень на кривій Едвардса. Для цього потрібно удосконалити алгоритм пошуку бірaціонально еквівалентних кривих.

Список використаних джерел у першому розділі

[1] *Закон України «Про електронний цифровий підпис»*, № 852-4. Відомості Верховної Ради України (ВВР), 2003.

[2] *Закон України «Про електронні документи та електронний документообіг»*, № 851-4. Відомості Верховної Ради України (ВВР), 2003.

[3] *Закон України «Про електронні довірчі послуги»,* № 2155-19. Відомості Верховної ради України (ВВР), 2017.

[4] Національні стандарти України, «ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптічніх кривих. Формування та перевірка», Київ, 2002.

[5] *Система криптографической защиты информации Шифр- X.509*, №9. Киев: Сайфер БИС, ООО, 2016. URL: https://cipher.kiev.ua/ru/x509

[6] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List. [Online]. Available: https://tools.ietf.org/html/rfc5280

[7] «OpenPGP», *OpenPGP*. [Online]. Available: https://www.openpgp.org. [Accessed: 15- Aug- 2015].

[8] F. Corella, «Implementing a PKI on a Blockchain – Pomcor», *Pomcor.com*, 2016. [Online]. Available: https://pomcor.com/2016/10/25/implementing-a-pki-on-a-blockchain/. [Accessed: 25- Oct- 2016].

[9] C. Allen, A. Brock, V. Buterin and J. Callas, «Decentralized Public Key Infrastructure», *Weboftrust.info*, 2015. [Online]. Available: http://www.weboftrust.info/downloads/dpki.pdf. [Accessed: 23- Dec- 2015].

[10] K. Lewison and F. Corella, «Backing Rich Credentials with a Blockchain PKI», *Pomcor.com*, 2016. [Online]. Available: https://pomcor.com/techreports/BlockchainPKI.pdf.

[11] C. Fromknecht, D. Velicanu and S. Yakoubov, «A Decentralized Public Key Infrastructure with Identity Retention», *IACR Cryptology ePrint Archive*, no. 803, 2014.

[12] S. Matsumoto and R. Reischuk, «IKP: Turning a PKI Around with Decentralized Automated Incentives», *IEEE Symposium on Security and Privacy (SP)*, 2017.

[13] A. Clerc, «About & Beyond PKI: blockchain and PKI», www.temet.ch, 2017.

[14] «Public Key Infrastructure and Blockchain», *Computersecuritypgp.blogspot.com*, 2016. [Online]. Available: https://computersecuritypgp.blogspot.com/pki-and-blockchain.html. [Accessed: 17- May- 2016].

[15] D. Bernstein, J. Buchmann and E. Dahmen, *Post-quantum cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[16] F. Corella, «Implementing a PKI on a Blockchain – Pomcor», *Pomcor.com*, 2016. [Online]. Available: https://pomcor.com/2016/10/25/implementing-a-pki-on-a-blockchain/. [Accessed: 25- Oct- 2016].

[17] IEEE P1363: Standard specification for public key cryptography, 2000.

[18] ISO/IEC 14888-3: *Information technology - Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms*, 2006.

[19] Digital Signature Standard (DSS), FIPS Publication 186-4, National Institute of Standards and Technology, July, 2013.

[20] PKCS #1 v2.2: RSA Cryptographic Standard, 2012.

[21] D. Giry, «Keylength - Cryptographic Key Length Recommendation», *Keylength.com*. [Online]. Available: https://www.keylength.com.

[22] І. Горбенко and Ю. Горбенко, *Прикладна криптологія*, 2 вид. Харків, Україна: Форт, 2012.

[23] Kryptographische Verfahren: Empfehlungen und Schlüssellängen, TR-02102-1 v2017-01, BSI, 02/2017.

[24] Commercial National Security Algorithm, Information Assurance Directorate at the NSA, 01/2016.

[25] Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 4, NIST, 01/2016.

[26] L. Valenta, N. Sullivan, A. Sanso and N. Heninger, «In search of CurveSwap: Measuring elliptic curve implementations in the wild», *Eprint.iacr.org*. [Online]. Available: https://eprint.iacr.org/2018/298.pdf.

[27] M. Lochter, «RFC 5639 - Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation», *Tools.ietf.org*, 2010. [Online]. Available: https://tools.ietf.org/html/rfc5639. [Accessed: Mar- 2010].

[28] S. Josefsson, «RFC 8032 - Edwards-Curve Digital Signature Algorithm (EdDSA)», *Tools.ietf.org*, 2017. [Online]. Available: https://tools.ietf.org/html/rfc8032. [Accessed: Jan- 2017].

[29] A. Langley, M. Hamburg, S. Turner, «RFC 7748 - Elliptic Curves for Security», *Tools.ietf.org*, 2016. [Online]. Available: https://tools.ietf.org/html/rfc7748. [Accessed: Jan- 2013].

[30] Д. Трунин, «IBM построила 50-кубитный квантовый компьютер», *N+1*, 2017. [Online]. Available: https://nplus1.ru/news/2017/11/13/IBM-50-qubit. [Accessed: 13- Nov- 2017].

[31] В. Королев, «Пятьдесят кубитов и еще один», *N+1*, 2017. [Online]. Available: https://nplus1.ru/material/2017/07/18/51-qubit-text. [Accessed: 18- Jul- 2017].

[32] Д. Трунин, «Google построил 72-кубитный квантовый компьютер», *N+1*, 2018. [Online]. Available: https://nplus1.ru/news/2018/03/06/google-72-qubit. [Accessed: 06- Mar- 2018].

[33] M. Roetteler, «Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms». *Cryptology ePrint Archive, Report 2017/598*. [Online]. Available: https://eprint.iacr.org/2017/598.pdf.

[34] *eBACS: ECRYPT Benchmarking of Cryptographic Systems*. Virtual Applications and Implementations Research Lab URL: https://bench.cr.yp.to/results-sign.html.

[35] J. Bhattacharya, *Rudiments of computer science*. Kolkata: B.K. Dhur of Academic Publishers, 2010, pp. 50-57.ISBN: 978-93-80599-02-1.

[36] P. Barrett, «Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor», in *Conference on the Theory and Application of Cryptographic Techniques*, 2000, pp. 311-323.

[37] P. Montgomery, «Modular multiplication without trial division», *Mathematics of Computation*, vol. 44, no. 170, pp. 519-519, 1985.

[38] W. Hasenplaugh, G. Gaubatz and V. Gopal, «Fast Modular Reduction», in *18th IEEE Symposium on Computer Arithmetic*, IEEE Computer Society Washington, DC, USA, 2007, pp. 223-229.

[39] D. Knuth and M. Ruckert, *The art of computer programming*. Upper Saddle River, NJ: Addison-Wesley, 2015.

[40] «GNU MP 6.1.2: Exact Division», *Gmplib.org*, 2018. [Online]. Available: https://gmplib.org/manual/Exact-Division.html.

[41] «GNU MP 6.1.2: References», *Gmplib.org*, 2018. [Online]. Available: https://gmplib.org/manual/References.html#References.

[42] В. Ковтун і А. Охрименко, «Метод підвищення продуктивності операції приведення по простому модулю», 2 вид., За ред. В. Пономаренко. Харків, Україна: ТОВ «Щедра садиба плюс», 2014, с. 204-219.

[43] V.Dupaquis, A. Venelli, «Redundant modular reduction algorithms», in: Prouff, E. (ed.) *CARDIS 2011*. LNCS vol. 7079, 2011, pp 102-114.

[44] D. Hankerson, S. Vanstone and A. Menezes, *Guide to elliptic curve cryptography*. New York: Springer, 2011, p. 98.

[45] N. Guillermin. A compressor for secure and high speed modular arithetic. Technical Report 354, Cryptology ePrint Archive, 2011. URL:https://eprint.iacr.org/2015/193.pdf.

[46] S. Jahani and A. Samsudin, «Zot-binary: a new numbering system with an application on big-integer multiplication», *Journal of Theoretical and Applied Information Technology (JATIT)*, vol. 48, no. 1, pp. 29-40, 2013.

[47] M. Johnson, B. Phung, T. Shackelford, S. Rueangvivatanakij. *Modular Reduction of Large Integers Using Classical, Barrett, Montgomery Algorithms*. URL:http://teal.gmu.edu/courses/ECE646/project/reports_2002/IP-1_report.pdf

[48] J. Chung and A. Hasan, «More Generalized Mersenne Numbers», in *10th Annual International Workshop on Selected Areas in Cryptography (SAC 2003)*, Ottawa, Canada, 2003, pp. 335-347.

[49] H. Wu, «On Computation of Polynomial Modular Reduction», Centre of Applied Cryptographic Research, University of Waterloo, 2000.

[50] L. Hars, «Long Modular Multiplication for Cryptographic Applications», in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004)*, MA, USA, 2004, pp. 45-61.

[51] M. Taschwer, «Modular Multiplication Using Special Prime Moduli», in *Kommunikationssicherheit im Zeichen des Internet*, P. Horster, Ed. Wiesbaden: Springer Vieweg, 2001, pp. 346-371.

[52] P. Giorgi, L. Imbert, T. Izard. «Multipartite Modular Multiplication». RR-11024, 2011, pp.25.

[53] S. Gueron, «Enhanced Montgomery Multiplication», in *Cryptographic Hardware and Embedded Systems - CHES 2002*, CA, USA, 2002, pp. 46-56.

[54] A. Menezes, P. Oorschot and S. Vanstone, *Handbook of applied cryptography*. [S.l.]: CRC Press, 2002.

[55] "The GNU MP Bignum Library", *Gmplib.org*. [Online]. Available: https://gmplib.org.

[56] K. Hasselström. «Fast Division of Large Integers A Comparison of Algorithm», Master's Thesis in Computer Science at the School of Computer Science and Engineering, Royal Institute of Technology, 2003. URL: http://www.treskal.com/kalle/exjobb/original-report.pdf.

[57] D. Takahashi, «A Parallel Algorithm for Multiple-Precision Division by a Single-Precision Integer», *Large-Scale Scientific Computing*, pp. 729-736, 2008.

[58] H. Lo, T. Chang and M. Lee, «Parallel unidirectional division algorithms and implementations parallel unidirectional division algorithms and implementations», *Journal of the Chinese Institute of Engineers*, vol. 24, no. 4, pp. 487-496, 2001.

[59] N. Emmart and C. Weems, «Parallel multiple precision division by a single precision divisor», in *18th International Conference on High Performance Computing*, Bangalore, India, 2011.

[60] Chae Ho on Lim, Hyo Sun Hwang. «Fast Modular Reduction With Precomputation», in Korea-Japan Joint Workshop on Information Security and Cryptology, Lecture.

[61] P. Comba, «Exponentiation cryptosystems on the IBM PC», *IBM Systems Journal*, vol. 29, no. 4, pp. 526-538, 1990. URL: http://eprint.iacr.org/2012/482.pdf, http://eprint.iacr.org/2012/170.pdf.

[62] А. Карацуба и Ю. Офман, «Умножение многозначных чисел на автоматах», АН СССР, 1962.

[63] D. Stehlé and P. Zimmermann, «A Binary Recursive Gcd Algorithm», *Lecture Notes in Computer Science*, pp. 411-425, 2004.

[64] L. Lhote and B. Vallée, «Sharp Estimates for the Main Parameters of the Euclid Algorithm», *LATIN 2006: Theoretical Informatics*, pp. 689-702, 2006.

[65] W. Mahmoud, «Reduced-Latency Algorithm for Finite Field Inversion in GF(2^m)», *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 9, pp. 7857-7858, 2011.

[66] M. Bluhm and S. Gueron, «Fast software implementation of binary elliptic curve cryptography», *Journal of Cryptographic Engineering*, vol. 5, no. 3, pp. 215-226, 2015.

[67] T. Itoh and S. Tsujii, «A fast algorithm for computing multiplicative inverses in GF(2m) using normal bases», *Information and Computation*, vol. 78, no. 3, pp. 171-177, 1988.

[68] L. Lei and T. Nakamura, «A fast algorithm for evaluating the matrix polynomial I+A+. . .+A/sup N-1/», *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 4, pp. 299-300, 1992.

[69] J. Maitin-Shepard, «Optimal software-implemented Itoh–Tsujii inversion for $GF(2^m)$», *Designs, Codes and Cryptography*, vol. 82, no. 1-2, pp. 301-318, 2016.

[70] D. Hankerson, J. López Hernandez and A. Menezes, «Software Implementation of Elliptic Curve Cryptography over Binary Fields», *Cryptographic Hardware and Embedded Systems — CHES 2000*, pp. 1-24, 2000.

[71] F. Rodríguez-Henríquez, G. Morales-Luna, N. Saqib and N. Cruz-Cortés, «Parallel Itoh–Tsujii multiplicative inversion algorithm for a special class of trinomials», *Designs, Codes and Cryptography*, vol. 45, no. 1, pp. 19-37, 2007.

[72] M. Scott. «Optimal Irreducible Polynomials for GF(2^m) Arithmetic». Cryptology ePrint Archive, Report 2007/192, 2007.

[73] J. Guajardo, S. Kumar, C. Paar and J. Pelzl, «Efficient Software-Implementation of Finite Fields with Applications to Cryptography», *Acta Applicandae Mathematicae*, vol. 93, no. 1-3, pp. 3-32, 2006.

[74] D. Bernstein, T. Lange and R. Rezaeian Farashahi, «Binary Edwards Curves», *Cryptographic Hardware and Embedded Systems – CHES 2008*, pp. 244-265.

[75] M. Rivain, «Fast and regular algorithms for scalar multiplication over elliptic curves», *IACR Cryptology ePrint Archive*, p. 388, 2011.

[76] M. Joye, «Highly Regular Right-to-Left Algorithms for Scalar Multiplication», *Cryptographic Hardware and Embedded Systems - CHES 2007*, pp. 135-147.

[77] B. Möller, «Securing Elliptic Curve Point Multiplication against Side-Channel Attacks», *Lecture Notes in Computer Science*, pp. 324-334, 2001.

[78] P. Birkner, «Scalar Multiplication and Addition Chains», Summer School on Elliptic and Hyperelliptic Curve Cryptography, Toronto, 2006.

[79] P. Shah, Xu Huang and D. Sharma, «Sliding window method with flexible window size for scalar multiplication on wireless sensor network nodes», *2010 International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, 2010..

[80] P. Montgomery, «Speeding the Pollard and elliptic curve methods of factorization», *Mathematics of Computation*, vol. 48, no. 177, pp. 243-243, 1987.

[81] B. Koziel, R. Azarderakhsh and M. Mozaffari-Kermani, «Low-Resource and Fast Binary Edwards Curves Cryptography», *Progress in Cryptology -- INDOCRYPT 2015*, pp. 347-369, 2015.

[82] É. Brier and M. Joye, «Weierstraß Elliptic Curves and Side-Channel Attacks», *Public Key Cryptography*, pp. 335-345, 2002.

[83] T. Izu and T. Takagi, «A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks», *Public Key Cryptography*, pp. 280-296, 2002.

[84] M. Ciet and M. Joye, «(Virtually) Free Randomization Techniques for Elliptic Curve Cryptography», *Information and Communications Security*, pp. 348-359, 2003.

[85] Жуков А.Е. Криптоанализ по побочным каналам. Информационно-методический журнал «Защита информации. Инсайд», №5, 2010 г.

[86] Y. Zhou and D. Feng, «Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing», *{IACR} Cryptology ePrint Archive*, p. 388, 2005.

[87] R. Rezaeian Farashahi and S. Hosseini, «Differential Addition on Binary Elliptic Curves», *Arithmetic of Finite Fields*, pp. 21-35, 2016.

[88] D. Bernstein and T. Lange, *Failures in NIST's ECC standards*. 2016. [Online]. Available: https://cr.yp.to/newelliptic/nistecc-20160106.pdf.

[89] «SafeCurves: choosing safe curves for elliptic-curve cryptography», *Safecurves.cr.yp.to*. [Online]. Available: http://safecurves.cr.yp.to.

[90] Q. Lin and F. Zhang, «Halving on Binary Edwards Curves», *Guofang Keji Daxue Xuebao/Journal of National University of Defense Technology*, vol. 2, no. 4, 2010.

[91] «Hyperelliptic org», Hyperelliptic.org, 2018. [Online]. Available: http://www.hyperelliptic.org.

[92] K. Kim, C. Lee and C. Negre, «Binary Edwards Curves Revisited», *Progress in Cryptology -- INDOCRYPT 2014*, pp. 393-408, 2014.

[93] M. Li, A. Miri and D. Zhu, «Fast Algorithm for Converting Ordinary Elliptic Curves into Binary Edward Form», *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 1, pp. 405-412, 2012.

[94] P. Barreto and J. Voloch, «Efficient Computation of Roots in Finite Fields», *Designs, Codes and Cryptography*, vol. 39, no. 2, pp. 275-280, 2006.

[95] IEEE P1363-2000: Standard Specifications for Public Key Cryptography. -2000. - 206p. URL: http://www.ieee.org

[96] Г. Уоррен, Алгоритмические трюки для программистов. Россия, Москва: Пер. с англ. Издательский дом «Вильяме», 2004.

[97] L. Axon and M. Goldsmith, «PB-PKI: A Privacy-aware Blockchain-based PKI», *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, vol. 6, pp. 311-318, 2017.

[98] Louise Axon, University of Oxford. «Privacy-awareness in Blockchain-based PKI», 2015 . URL: https://ora.ox.ac.uk/objects/ uuid:f8377b69-599b-4cae-8df0-f0cded53e63b

[99] W. Al-Saqaf and N. Seidler, "Blockchain technology for social impact: opportunities and challenges ahead", *Journal of Cyber Policy*, vol. 2, no. 3, pp. 338-354, 2017.

[100] A. Chatterjee and I. Sengupta, «Design of a high performance Binary Edwards Curve based processor secured against side channel analysis», *Integration, the VLSI Journal*, vol. 45, no. 3, pp. 331-340, 2012.

[101] N.Sklavos, A.Fournaris, «Binary Edwards Curve Design Strategy for Efficient and Power Attack Resistant Architectures», *Proceedings of the 4th Workshop on Secure Hardware & Security Evaluation, Cryptographic Hardware and Embedded Systems (CHES'15)*, 2015.

РОЗДІЛ 2
РОЗРОБКА МЕТОДУ ПІДВИЩЕННЯ ШВИДКОДІЇ АРИФМЕТИЧНИХ ОПЕРАЦІЙ НАД ЦІЛИМИ ЧИСЛАМИ

2.1. Дослідження методу ділення «в стовпчик» великих цілих чисел

В алгоритмі RSA при генерації загально системних параметрів, потрібно проводити тестування великих чисел на взаємну простоту, а також шукати обернений елемент – особистий ключ. Для цього використовують звичайний та розширений алгоритми Евкліда [1-2], який є основним інструментом сучасної теорії чисел, і в основу якого входить ділення.

Суть звичайного алгоритму Евкліда полягає в наступному: для будь-яких цілих чисел a та b, $b \neq 0$, $a,b \in \boldsymbol{Z}$ існують певні цілі числа q, r, які задовольняють умову $a = b \cdot q + r, 0 \le r < b$, де r - залишок від ділення, q - неповне ціле при діленні a на b.

Нижче подається детальніший опис алгоритму. Від самого початку, $r_1 = b$ і $r_2, \ldots, r_n$ - наступні дільники в прототипі, тоді послідовно обчислюються наступні рівності:

$a = r_0 = b \cdot q_1 + r_2,\ 0 \le r_2 < b,$

$b = r_1 = r_2 \cdot q_2 + r_3,\ 0 \le r_3 < r_2,$

$r_2 = r_3 \cdot q_3 + r_4, 0 \le r_4 < r_3,$

. . . . . .

$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n,\ 0 \le r_n < r_{n-1},$

$r_{n-1} = r_n \cdot q_n.$

Обчислення припиняються, коли залишок від ділення r_i буде меншим за b .

Алгоритм ділення великих цілих чисел «в стовпчик» вибраний за простоту в розумінні та реалізації, використовуються операції складання, віднімання і зсуву, універсальний, оскільки дозволяє проводити ділення із залишком і без нього, а також відсутні обмеження на дільник.

Властивості реалізації алгоритму ділення в стовпчик аналізувалися в двох випадках:

– однакової двійкової довжини – кількість машинних слів діленого та дільника однакові (1);

– двійкова довжина діленого в 2 рази перевищує довжину дільника (результат отриманий після операції множення або піднесення до квадрату) (2).

Алгоритм 2.1. Прототип ділення великих цілих чисел в стовпчик.

Вхід: $a,b \in \boldsymbol{Z}$, $a,b \neq 0$, $n_a \leftarrow \log_2 \left\lceil \frac{a}{w} \right\rceil$, де n_a - кількість машинних слів, яке займає ділене; $n_b \leftarrow \log_2 \left\lceil \frac{b}{w} \right\rceil$ (В разі (2) $n_b \leftarrow \lceil n_a/2 \rceil$) - кількість машинних слів, яке займає дільник; w - ширина машинного слова, зазвичай $w = 32$.

Вихід: $q,r \in \boldsymbol{Z}$.

1. $r \leftarrow a$, $m \leftarrow b$, $s \leftarrow 1$, $q \leftarrow 0$.

2. While $r > m$ do

2.1. $m \leftarrow m << 1$, $s \leftarrow s << 1$.

3. While $r > b$ do

3.1. While $r > m$ do

3.1.1. $m \leftarrow m >> 1$, $s \leftarrow s >> 1$.

3.2. $r \leftarrow r - m$, $q \leftarrow q + s$.

4. Return (q,r).

Рис.2.1. 1. Псевдокод операції ділення в стовпчик

Проаналізувавши алгоритм 2.1, можна виділити ряд напрямків для подальшого удосконалення.

У п.1 відбувається ініціалізація параметрів рівняння: залишку від ділення $r \leftarrow a$; вирівняного по старшому біту діленого s; проміжного дільника $m \leftarrow b$, для подальшого віднімання; проміжного цілого $s \leftarrow 1$, шуканого цілого $q \leftarrow 0$.

У циклі п. 2 відбувається перевірка умови $r > m$ до тих пір, доки не відбудеться вирівнювання номерів старших бітів проміжного дільника m і діленого a. Поки умова п.2 - істинна, відбувається зсув вліво проміжного дільника m і проміжного цілого s. При порівнянні великих чисел, виконується

обчислювально складна перевірка $r_i > m_i$, $i = \overline{n-1, 0}$ за всіма машинними словами і на кожній ітерації. Число даних перевірок можна значно скоротити, використовуючи підхід наближеного порівняння великих цілих чисел, за допомогою порівняння номерів старших бітів r та m, а за допомогою знайденої різниці k (між номерами старших бітів r та m), виконати зсув вліво на k біт п. 2.1 за одну ітерацію. Повністю від перевірки умови п. 2 позбутися неможливо, тому що в разі, коли номери старших бітів r та m - рівні між собою, може виникнути невизначеність. Щоб скоротити обчислювальну складність при порівнянні $r_i > m_i$, $\overline{i = n-1, 0}$, слід оперувати лише значущими машинними словами (не порожніми).

У циклі ділення п. 3, для перевірки умови $r > b$, відбувається, як і в попередньому пункті, складна перевірка умови $r_i > b_i$, $i = \overline{n-1, 0}$ для всіх машинних слів. Спочатку порівняння цілих чисел необхідно виконувати, використовуючи підхід порівняння номерів старших бітів, і в разі рівності старших бітів - виконувати порівняння за значущими словами.

У п. 3.1 відбувається проміжне ділення, яке вимагає циклічної перевірки умови $r_i < m_i$, $i = \overline{n-1, 0}$ для всіх слів. Зсув вправо всіх машинних слів на 1 вирівняного дільника m та проміжного цілого s виконується в п. 3.1.1, доки виконується умова п. 3.1, в іншому випадку виконується додавання $q \leftarrow q + s$ і віднімання $r \leftarrow r - m$ за всіма машинним словами в п. 3.2. За аналогією з попередніми зауваженнями, кількість перевірок п. 3.1 можна скоротити, перейшовши до порівняння номерів старших бітів r і m. Знайдену різницю k (різниця між номерами старших бітів r і m), використовувати для зсуву вправо за 1 ітерацію в п. 3.1.1.

У разі невиконання умови п. 3, відбувається перехід до п. 4 і повернення шуканої неповної частки q і залишку від ділення r.

2.2. Розробка удосконаленого методу ділення «в стовпчик» великих цілих чисел

Запропоновані вище підходи, реалізовані у вигляді удосконаленого алгоритму 2.2 [3-4]. Позначимо через R, M, S, T – номера старших бітів залишку від ділення r, вирівняного проміжного дільника m, проміжного цілого s, дільника t; $\tilde{r}$, $\tilde{m}$, $\tilde{s}$, $\tilde{t}$ – кількість значущих слів r, m, s, t, відповідно; $sgf(\cdot)$ – функція, яка обчислює кількість значущих слів великого числа; $msb(\cdot)$ – функція, яка обчислює номер старшого біта великого числа; $msb(\cdot,\tilde{\cdot})$ – функція, яка одночасно обчислює кількість значущих слів і номер старшого біта великого числа; $r_j \underset{\tilde{t}}{\geq} t_j$ – порівняння великих цілих чисел по значущим словам $\tilde{t}$.

Алгоритм 2.2. Удосконалений алгоритм ділення великих цілих чисел в стовпчик.

Вхід: $a,b \in \boldsymbol{Z}$, $a,b \neq 0$, $n_a \leftarrow \log_2\left\lceil \frac{a}{w} \right\rceil$,

$n_b \leftarrow \log_2\left\lceil \frac{b}{w} \right\rceil$ ($n_b \leftarrow \lceil n_a/2 \rceil$ - в разі (2)),

$w = 32$.

Вихід: $q,r \in \boldsymbol{Z}$.

1. $r \leftarrow a$, $m \leftarrow b$, $s \leftarrow 1$, $q \leftarrow 0$, $t \leftarrow b$, $S \leftarrow 1$, $\tilde{s} \leftarrow 1$.
2. $M \leftarrow msb(m,\tilde{m})$, $R \leftarrow msb(r,\tilde{r})$, $T \leftarrow M$, $\tilde{t} \leftarrow \tilde{m}$.
3. $k \leftarrow R - M$.
4. if $k > 0$ then $m \leftarrow m \underset{\tilde{m}}{<<} k$, $s \leftarrow s \underset{\tilde{s}}{<<} k$, $M \leftarrow M + k$, $S \leftarrow S + k$.
5. $\tilde{m} \leftarrow sgf(m)$.
6. While $(R > M) or \left(r \underset{\tilde{m}}{\geq} m\right)$ do
   6.1. $\tilde{s} \leftarrow sgf(s)$.
   6.2. $m \leftarrow m \underset{\tilde{m}}{<<} 1$, $s \leftarrow s \underset{\tilde{s}}{<<} 1$, $M \leftarrow M + 1$, $S \leftarrow S + 1$, $\tilde{m} \leftarrow sgf(m)$.
7. While $(R > T) or \left(r \underset{\tilde{t}}{\geq} t\right)$ do
   7.1. $k \leftarrow M - R$.
   7.2. if $k > 0$ then $\tilde{s} \leftarrow sgf(s)$, $\tilde{m} \leftarrow sgf(m)$.
   7.3. $m \leftarrow m \underset{\tilde{m}}{>>} k$, $s \leftarrow s \underset{\tilde{s}}{>>} k$.
   7.4. $S \leftarrow S - k$, $M \leftarrow M - k$.
   7.5. $\tilde{m} \leftarrow sgf(m)$.
   7.6. While $(R < M) or \left(r \underset{\max(\tilde{m},\tilde{r})}{<} m\right)$ do
      7.6.1. $\tilde{s} \leftarrow sgf(s)$.
      7.6.2. $m \leftarrow m \underset{\tilde{m}}{>>} 1$, $s \leftarrow s \underset{\tilde{s}}{>>} 1$.
      7.6.3. $M \leftarrow M - 1$, $S \leftarrow S - 1$.
      7.6.4. $\tilde{m} \leftarrow sgf(m)$.
   7.7. $\tilde{r} \leftarrow sgf(r)$.
   7.8. $r \leftarrow r \underset{\tilde{r}}{-} m$.
   7.9. $R \leftarrow msb(r,\tilde{r})$.
   7.10. $q \leftarrow q + s$.
8. Return (q,r).

Рис.2.2. 1. Псевдокод удосконаленої операції ділення в стовпчик

У п.1 відбувається ініціалізація параметрів: залишок від ділення $r \leftarrow a$, вирівняного проміжного дільника $m \leftarrow b$ по старшому біту діленого для подальшого віднімання, проміжного цілого $s \leftarrow 1$, шуканого цілого $q \leftarrow 0$,

додаткова змінна $t \leftarrow b$, номер старшого біта проміжного цілого $S \leftarrow 1$, кількість значущих слів $\tilde{s} \leftarrow 1$.

Для скорочення кількості обчислювально складних перевірок п.6, в п.2 відбуваються обрахунки номерів старших бітів і кількості значущих слів: M, $\tilde{m}$, R, $\tilde{r}$, а також присвоєння: $T \leftarrow M$, $\tilde{t} \leftarrow \tilde{m}$. Надалі обчислюється різниця k між номерами старших бітів r і m в п. 3.

У п. 4 здійснюється перевірка умови $k > 0$, якщо умова - істинна, за 1 крок виконується зсув вліво тільки значущих слів m і s на дану різницю k біт. Після зазначених операцій номера старших бітів r і m – рівні. Надалі обчислюються номера старших бітів модифікованих m і s, а також кількість значущих слів $sgf(m)$, п. 5.

Додаткова умова в п. 6, дозволяє виконувати обчислювально складну перевірку (по значущим словам) $r \underset{\tilde{m}}{\geq} m$, лише коли $R == M$.

Якщо умова п. 6 – істинна, то в п. 6.1 обчислюється кількість значущих слів $\tilde{s} \leftarrow sgf(s)$ і здійснюються зсуви вліво на 1 біт значущих слів m_j, $j \leftarrow \overline{\tilde{m}-1, 0}$ і s_j, $j \leftarrow \overline{\tilde{s}-1, 0}$. Після зсувів, номера старших бітів m і s приймають нові значення, і надалі обчислюється $sgf(m)$. Якщо умова п. 6 – хиба, то здійснюється перехід до п. 7.

Перевірка умови (по значущим словам) $r \underset{\tilde{t}}{\geq} t$ в п. 7 (проміжне ділення) проводиться лише, коли $msb(r) == msb(t)$. Якщо умова п. 7 – істинна, то виконуються дії п.п. 7.1–7.10, в іншому випадку – перехід до п. 8.

П.п. 7.1-7.5 дозволяють звести кількість обчислювально складних порівнянь великих чисел в п. 7.6 до мінімуму. У п. 7.1 обчислюється різниця k між номерами старших бітів r і m. Якщо різниця більша за 0, то обчислюється кількість значущих слів $\tilde{s}$ і $\tilde{m}$, п. 7.2. Надалі виконується зсув вправо по значущим словам m і s на k біт, після чого $R == M$, а також обчислюються номера старших бітів (див. п. 5.4) і кількість значущих слів $sgf(m)$, див. п. 7.5.

У циклі п.7.6 проводиться порівняння $R < M$: якщо $msb(r) == msb(m)$, тоді виконується додаткове порівняння $r \underset{\max(\tilde{r},\tilde{m})}{<} m$ по значущим словам. Якщо умова п. 7.6 – істинна, то виконуються дії п.п. 7.6.1 - 7.6.4, в іншому випадку перехід до п. 7.7.

На кроці п. 7.6.1 обчислюється кількість значущих слів $\tilde{s} \leftarrow sgf(s)$, після чого виконуються зсуви вправо на 1 біт значущих слів і обчислюються нові номери старших бітів m і s (п. 7.6.2 - п. 7.6.3), в подальшому визначаються значущі слова $\tilde{m} \leftarrow sgf(m)$, п.7.6.4.

У п. 7.7 обчислюється кількість значущих слів $\tilde{r} \leftarrow sgf(r)$, яке потрібно для обчислення $r \leftarrow r \underset{\tilde{r}}{-} m$ п.7.8.

Після вирахування в п. 7.9, обчислюється номер старшого біта і кількість значущих слів $\tilde{r}$, в п. 7.10 виконується накопичення цілого $q \leftarrow q + s$.

На кроці п.8 відбувається повернення значень залишку від ділення r і шуканого цілого q.

2.3. Оцінка обчислювальної складності

Для оцінки ефективності алгоритмів, необхідно оцінити їх теоретичні складності. Позначимо через L – арифметичні операції (не розрізняються арифметичні операції, операції присвоєння, зсуву); C – операції порівняння; n_a та n_b – номери старших бітів діленого та дільника, відповідно; n – кількість біт, зарезервованих в пам'яті для зберігання максимально можливого числа, k – різниця між номером старшого біта діленого і дільника; w – довжина машинного слова.

Обчислювальна складність алгоритму 2.1 для випадку (1): операції присвоєння п.1 мають складність $\left\lceil \frac{n}{w} \right\rceil L$, порівняння і зсуву п.п. 2 - 2.1 - $C\left\lceil \frac{n+n_a}{2 \cdot w} \right\rceil + 2\left\lceil \frac{n}{w} \right\rceil L$, порівняння п. 3 - $k \cdot C$, порівняння п. 3.1 - $\frac{3}{2}kC$, зсуви п. 3.1.1 - $\frac{2}{3}\left\lceil \frac{n}{w} \right\rceil L$, додавання і віднімання (п. 3.2) - $2k\left\lceil \frac{n}{w} \right\rceil L$.

$$I\left(A_{2.1}^{(1)}\right)=\left\lceil\frac{n}{w}\right\rceil L\cdot\left(6+\frac{8}{3}k\right)+\frac{5}{2}kC+\frac{1}{2}\left\lceil\frac{n-n_a}{w}\right\rceil kC.$$

Обчислювальна складність алгоритму 2.2 для випадку (1): операції присвоєння п. 1 володіють обчислювальною складністю $4\left\lceil\frac{n}{w}\right\rceil L+2L$; визначення номерів старших бітів п. 2 можна представити $52L+2C$; зсуви п. 4. і присвоєння номерів старших бітів – $\frac{1}{3}\left(\left\lceil\frac{n_a}{w}\right\rceil L+\left\lceil\frac{k}{w}\right\rceil L+2L\right)$, де ймовірність $\boldsymbol{P}(k>0)=\frac{1}{3}$; обчислення значущих слів – L; порівняння п. 6 – $k\left(C+\frac{1}{3}C\right)$, де ймовірність $\boldsymbol{P}(M==R)=\frac{1}{3}$; зсуви п. 6.2 – $\left\lceil\frac{n_a+n_b}{2\cdot w}\right\rceil kL$; порівняння п.7 – $\frac{4}{3}kC$, де ймовірність $\mathbf{P}(R==T)=\frac{1}{3}$; різниця номерів старших бітів п. 7.1 – kL; обчислення значущих слів п. 7.2 – $\frac{2}{3}kL$, де ймовірність $\boldsymbol{P}(k>0)=\frac{1}{3}$; зсув вправо п. 7.3 – $\left(\frac{1}{2}\left(\left\lceil\frac{n_a}{w}\right\rceil+\left\lceil\frac{n_b}{w}\right\rceil\right)+\left\lceil\frac{n_a-n_b}{w}\right\rceil\right)kL$; присвоєння нових номерів старших біт п. 7.4 та обчислення значущих слів п. 7.5 $-3kL$; операція порівняння п. 7.6 – $\frac{4}{3}kC$, де $\boldsymbol{P}(R==M)=\frac{1}{3}$; обчислення значущих слів і зсувів вправо п.п.7.6.1 - 7.6.4 – $\frac{1}{3}\left(2+\frac{1}{2}\left(\left\lceil\frac{n_a}{w}\right\rceil+\left\lceil\frac{n_b}{w}\right\rceil\right)+\left\lceil\frac{n_a-n_b}{w}\right\rceil\right)kL$, де $\boldsymbol{P}(R==M)=\frac{1}{3}$; обчислення значущих слів п. 7.7 – kL; віднімання п.7.8 – $\frac{1}{2}kL\left(\left\lceil\frac{n_a}{w}\right\rceil+\left\lceil\frac{n_b}{w}\right\rceil\right)$; обчислення номера старшого біта п. 7.9 – $(26L+C)k$; операція додавання – $\left\lceil\frac{n_a}{w}\right\rceil k$. Оцінку обчислювальної складності алгоритму можна представити таким чином:

$$I\left(A_{2.2}^{(1)}\right)=L\cdot\left(55\frac{2}{3}+36\frac{1}{3}k+\frac{1}{3}\left\lceil\frac{n_a}{w}\right\rceil\cdot(8k+1)+\frac{5}{3}k\cdot\left\lceil\frac{n_b}{w}\right\rceil+\left\lceil\frac{k}{w}\right\rceil\cdot\left(\frac{1}{3}+k\right)\right)+C\cdot(2+5k).$$

Обчислювальна складність алгоритму 2.1 для випадку (2): складність операції порівняння і зсувів в п.п. 2 – 2.1 - $k\left\lceil \frac{n-n_a}{w} \right\rceil C + 2Lk\left\lceil \frac{n}{w} \right\rceil$, порівняння п.3 – $\frac{k}{3}\left\lceil \frac{n}{w} \right\rceil C$, де $\boldsymbol{P}\left(\underset{z}{r > b},\ z = \left\lceil \frac{n}{w} \right\rceil \right) = \frac{2}{3}$, порівняння п. 3.1 – $\frac{2}{3} \cdot k\left\lceil \frac{n-n_b}{w} \right\rceil C$, де $\boldsymbol{P}\left(\underset{z}{r < m},\ z = \left\lceil \frac{n-n_b}{w} \right\rceil \right) = \frac{2}{3}$, зсувів п. 3.1.1 – $2k\left\lceil \frac{n}{w} \right\rceil L$, додавання і віднімання (п. 3.2) – $\left\lceil \frac{n}{w} \right\rceil k\,L$. Оцінка обчислювальної складності має вигляд:

$$I\left(A_{2.1}^{(2)}\right) = 5 \cdot k\left\lceil \frac{n}{w} \right\rceil L + C \cdot k\left(\frac{1}{3}\left\lceil \frac{n}{w} \right\rceil + \left\lceil \frac{n-n_a}{w} \right\rceil + \left\lceil \frac{n-n_b}{w} \right\rceil \right).$$

Обчислювальна складність алгоритму 2.2 для випадку (2): обчислювальну складність операції обчислення номерів старших бітів і значущих слів п. 2 – $54L$; операції перевірки $k > 0$, зсувів п. 4. і присвоєнь номерів старших бітів – $C + \left\lceil \frac{n_a}{w} \right\rceil L + \left\lceil \frac{k}{w} \right\rceil L + 2L$; обчислення значущих слів п. 5 і порівняння п. 6 складає – $2L + C + C\left\lceil \frac{n_a}{w} \right\rceil \left\lceil \frac{n_a}{w} \right\rceil^{-1}$, тобто операція порівняння великих чисел виконується лише в разі, коли $R == M$ і ймовірність того, що порівнюватися будуть числа по всім словам $\boldsymbol{P}\left(\underset{z}{r > m},\ z = \left\lceil \frac{n_a}{w} \right\rceil \right) = \left\lceil \frac{n_a}{w} \right\rceil^{-1}$; обчислення значущих слів п. 6.1 та операції зсуву в п. 6.2 - $\frac{1}{3}L\left(2 + \left\lceil \frac{n_a}{w} \right\rceil + \left\lceil \frac{k}{w} \right\rceil \right)$; порівняння п. 7 – $\left(\frac{k}{2} + 1 \right) C$, де $\boldsymbol{P}\left(\underset{z}{r > t},\ z = \left\lceil \frac{n_b}{w} \right\rceil \right) = \left\lceil \frac{n_b}{w} \right\rceil^{-1}$ – ймовірність того, що порівняння чисел проводиться за всіма словами; $\frac{k}{2}C$ – складність операції перевірки $k > 0$, обчислення значущих слів – $\frac{4}{3}k\,L$ в п. 7.2, де $\boldsymbol{P}(k > 0) = \frac{1}{3}$; зсув вправо п. 7.3 – $\frac{k}{3}L\left(\left\lceil \frac{n_a}{w} \right\rceil + \left\lceil \frac{k}{w} \right\rceil \right)$; присвоєння нових номерів старших бітів п. 7.4 та

обчислення значущих слів п. 7.5 – $\frac{4}{3}kL$; операція порівняння п. 7.6 – $kC+\frac{k}{2}C\left\lceil\frac{n_a}{w}\right\rceil\left\lceil\frac{n_a}{w}\right\rceil^{-1}$, де $\boldsymbol{P}\left(r<_z m,\ z=\left\lceil\frac{n_a}{w}\right\rceil\right)=\left\lceil\frac{n_a}{w}\right\rceil^{-1}$ – ймовірність порівняння всіх машинних слів представлення чисел за умови рівності номерів старших бітів; обчислення значущих слів і зсувів вправо п.п. 7.6.1-7.6.4 – $kL\left(1+\frac{1}{3}\left(\left\lceil\frac{n_a}{w}\right\rceil+\left\lceil\frac{k}{w}\right\rceil\right)\right)$, де $\boldsymbol{P}(r<m)=\frac{1}{3}$ – ймовірність попадання в цикл п. 7.6; обчислення значущих слів п. 7.7 $-kL$; віднімання п. 7.8 – $\frac{k}{4}\left(1+\left\lceil\frac{n_a}{w}\right\rceil\right)$; обчислення номера старшого біта і кількості значущих слів п. 7.9 – $\frac{27\cdot k}{2}$; складання в п. 7.10 – $\frac{k}{2}\left\lceil\frac{n}{w}\right\rceil L$.

$$I\left(A_{2.2}^{(2)}\right)=L\left(63+\left\lceil\frac{n_a}{w}\right\rceil\left(\frac{5}{3}+\frac{7}{12}k\right)+\left\lceil\frac{k}{w}\right\rceil\left(\frac{k}{3}+\frac{5}{3}\right)+\left\lceil\frac{n}{w}\right\rceil\frac{k}{2}+\frac{207}{12}k\right)+C\left(4+\frac{5}{2}k\right).$$

Для об'єктивності порівняння, розглядаються теоретичні оцінки обчислювальної складності алгоритму 2.1 та алгоритму 2.2.

У табл. 2.3.1 представлені теоретичні оцінки обчислювальних складностей алгоритму прототипу ділення «в стовпчик» та удосконаленого, в залежності від відношень двійкових довжин діленого і дільника. Так, під $\rho\in\{0.1,\ 0.2,\ldots,\ 0..9\}$ - відношення числа використовуваних біт до числа зарезервованих у великому числі (дільнику). Двійкова довжина діленого змінюється в діапазоні від 64 до 16384 біт. Ширина машинного слова $w=32$. Таблиця з оцінками складностей розбита на блоки, кожен з яких описує відповідну частину, наприклад, $64\,біта\,|\,n_a=2,\ 32\,біта\,|\,n_b=1$ вказує, що двійкова довжина діленого становить 64 біта (2 машинних слова), а дільника – 32 (одне машинне слово). Для зручності порівняння, кількість арифметичних операцій, відповідних алгоритмів $I_L\left(A_{2.1}^{(2)}\right)$ і $I_L\left(A_{2.2}^{(2)}\right)$, представлені один над одним. Аналогічним чином представлені і кількість операцій порівняння – $I_C\left(A_{2.1}^{(2)}\right)$ і

$I_C\left(A_{2.2}^{(2)}\right)$. Також наводяться співвідношення кількості відповідних операцій $I_L\left(A_{2.1}^{(2)}\right)/I_L\left(A_{2.2}^{(2)}\right)$ і $I_C\left(A_{2.1}^{(2)}\right)/I_C\left(A_{2.2}^{(2)}\right)$, для наочності.

Таблиця 2.3.1

Теоретичні оцінки складності алгоритмів РАЕ і МРАЕ, виражені в арифметичних операціях та операціях порівняння для випадку (2)

ρ	1	0,9	0,8	0,7	0,6	0,5	0,4	0,3	0,2	0,1
64 *біта*$\|n_a = 2$, 32 *біта*$\|n_b = 1$										
k	29	32	35	39	42	45	48	51	55	58
$I_L\left(A_{2.1}^{(2)}\right)$	320	352	384	416	448	480	512	544	576	608
$I_L\left(A_{2.2}^{(2)}\right)$	701	788	860	932	1003	1075	1147	1219	1290	1362
$I_C\left(A_{2.1}^{(2)}\right)$	74	105	114	124	134	143	153	162	172	181
$I_C\left(A_{2.2}^{(2)}\right)$	77	85	93	101	109	117	125	133	141	149
$I_L\left(A_{2.1}^{(2)}\right)/I_L\left(A_{2.2}^{(2)}\right)$	0,46	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45
$I_C\left(A_{2.1}^{(2)}\right)/I_C\left(A_{2.2}^{(2)}\right)$	0,97	1,24	1,24	1,23	1,23	1,23	1,23	1,22	1,22	1,22
128 *біт*$\|n_a = 4$, 64 *біта*$\|n_b = 2$										
k	61	67	74	80	87	93	99	106	112	119
$I_L\left(A_{2.1}^{(2)}\right)$	1280	1408	1536	1664	1792	1920	2048	2176	2304	2432
$I_L\left(A_{2.2}^{(2)}\right)$	1589	1783	1 946	2109	2271	2434	2648	2814	2980	3145
$I_C\left(A_{2.1}^{(2)}\right)$	233	303	330	358	385	413	508	540	571	603
$I_C\left(A_{2.2}^{(2)}\right)$	157	173	189	205	221	237	253	269	285	301
$I_L\left(A_{2.1}^{(2)}\right)/I_L\left(A_{2.2}^{(2)}\right)$	0,81	0,79	0,79	0,79	0,79	0,79	0,77	0,77	0,77	0,77
$I_C\left(A_{2.1}^{(2)}\right)/I_C\left(A_{2.2}^{(2)}\right)$	1,49	1,75	1,75	1,75	1,75	1,75	2,01	2,01	2,01	2,01
512 *біт*$\|n_a = 16$, 256 *біт*$\|n_b = 8$										
k	242	268	293	319	344	370	396	421	447	472
$I_L\left(A_{2.1}^{(2)}\right)$	20480	22528	24576	26624	28672	30720	32768	34816	36864	38912
$I_L\left(A_{2.2}^{(2)}\right)$	10479	11713	12972	14096	15394	16718	18067	19441	20617	22030
$I_C\left(A_{2.1}^{(2)}\right)$	2959	3441	3348	3957	4616	4946	5681	6467	7304	8191
$I_C\left(A_{2.2}^{(2)}\right)$	609	673	737	801	865	929	993	1057	1121	1185
$I_L\left(A_{2.1}^{(2)}\right)/I_L\left(A_{2.2}^{(2)}\right)$	1,95	1,92	1,89	1,89	1,86	1,84	1,81	1,79	1,79	1,77
$I_C\left(A_{2.1}^{(2)}\right)/I_C\left(A_{2.2}^{(2)}\right)$	4,86	5,11	4,54	4,94	5,34	5,32	5,72	6,12	6,52	6,91
4096 *біт*$\|n_a = 128$, 2048 *біт*$\|n_b = 64$										
k	1 950	2 155	2 360	2 564	2 769	2 974	3 179	3 384	3 588	3 793
$I_L\left(A_{2.1}^{(2)}\right)$	1310720	1441792	1572864	1703936	1835008	1966080	2097152	2228224	2359296	2490368
$I_L\left(A_{2.2}^{(2)}\right)$	404284	454263	504494	557238	610029	664049	720889	777469	837074	896214
$I_C\left(A_{2.1}^{(2)}\right)$	137953	167361	197173	232055	266936	304251	347242	389626	438092	485544

Продовження таблиці 2.3.1

$I_C\left(A_{2.2}^{(2)}\right)$	4879	5391	5903	6415	6927	7439	7951	8463	8975	9487
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	3,24	3,17	3,12	3,06	3,01	2,96	2,91	2,87	2,82	2,78
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	28,27	31,04	33,40	36,17	38,54	40,90	43,67	46,04	48,81	51,18
16384 *біт*$\|n_a = 512,$ 8132 *біт*$\|n_b = 256$										
k	8 162	8 981	9 800	10620	11439	12258	13077	13896	14716	15535
$I_L\left(A_{2.1}^{(2)}\right)$	20971520	23068672	25165824	27262976	29360128	31457280	33554432	35651584	37748736	39845888
$I_L\left(A_{2.2}^{(2)}\right)$	6423899	7180839	7963161	8761471	9585982	10431792	11292361	12180360	13082299	14012487
$I_C\left(A_{2.1}^{(2)}\right)$	2084372	2524758	3007316	3521503	4087595	4683694	5333320	6025118	6744490	7519822
$I_C\left(A_{2.2}^{(2)}\right)$	20409	22457	24505	26553	28601	30649	32697	34745	36793	38841
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	3,26	3,21	3,16	3,11	3,06	3,02	2,97	2,93	2,89	2,84
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	102,13	112,43	122,72	132,62	142,92	152,82	163,11	173,41	183,31	193,61

Таблиця 2.3. 2

Теоретичні оцінки складнощів алгоритмів PAE і MPAE, виражені в арифметичних операціях та операціях порівняння для випадку (1)

ρ	1	0,9	0,8	0,7	0,6	0,5	0,4	0,3	0,2	0,1
64 *біта*$\|n_a = 2,$ 64 *біта*$\|n_b = 2$										
k	1	7	13	19	25	31	37	43	49	55
$I_L\left(A_{2.1}^{(2)}\right)$	38	99	159	219	279	339	399	460	520	580
$I_L\left(A_{2.2}^{(2)}\right)$	83	215	347	479	611	743	894	1029	+1164	1299
$I_C\left(A_{2.1}^{(2)}\right)$	9	23	37	51	65	101	119	137	155	173
$I_C\left(A_{2.2}^{(2)}\right)$	6	21	36	51	66	81	96	111	126	141
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	0,46	0,46	0,46	0,46	0,46	0,46	0,45	0,45	0,45	0,45
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	1,46	1,08	1,02	0,99	0,98	1,24	1,24	1,23	1,23	1,22
128 *біт*$\|n_a = 4,$ 128 *біта*$\|n_b = 4$										
k	5	17	29	41	53	65	77	89	101	113
$I_L\left(A_{2.1}^{(2)}\right)$	154	394	635	876	1116	1357	1597	1838	2079	2319
$I_L\left(A_{2.2}^{(2)}\right)$	182	476	769	1085	1385	1718	2024	2330	2688	2999
$I_C\left(A_{2.1}^{(2)}\right)$	23	59	95	159	203	292	343	395	516	575
$I_C\left(A_{2.2}^{(2)}\right)$	16	46	76	106	136	166	196	226	256	286
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	0,84	0,83	0,83	0,81	0,81	0,79	0,79	0,79	0,77	0,77
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	1,46	1,28	1,25	1,50	1,49	1,76	1,75	1,75	2,01	2,01
512 *біт*$\|n_a = 16,$ 512 *біт*$\|n_b = 16$										
k	8	32	56	81	105	129	153	177	201	225
$I_L\left(A_{2.1}^{(2)}\right)$	614	1577	2540	3502	4465	5427	6390	7352	8315	9277

Продовження таблиці 2.3.2

$I_L\left(A_{2.2}^{(2)}\right)$	319	1 044	1764	2526	3311	4121	4877	5723	6593	7486
$I_C\left(A_{2.1}^{(2)}\right)$	66	196	315	492	701	942	1109	1397	1717	2069
$I_C\left(A_{2.2}^{(2)}\right)$	25	85	145	205	266	326	386	446	506	566
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	1,93	1,51	1,44	1,39	1,35	1,32	1,31	1,28	1,26	1,24
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	2,65	2,30	2,17	2,40	2,64	2,89	2,87	3,13	3,39	3,65
$4096\ бiт \mid n_a = 128,\ 4096\ бiт \mid n_b = 128$										
k	448	533	918	1303	1688	2073	2458	2843	3228	3613
$I_L\left(A_{2.1}^{(2)}\right)$	157286	403702	650117	896532	1142948	1389363	1635779	1882194	2128609	2375025
$I_L\left(A_{2.2}^{(2)}\right)$	26708	98904	175721	257157	343214	433892	529189	629107	733645	842804
$I_C\left(A_{2.1}^{(2)}\right)$	12662	37494	68425	105455	148583	197811	253137	314562	382085	455708
$I_C\left(A_{2.2}^{(2)}\right)$	373	1336	2299	3261	4224	5186	6149	7111	8074	9036
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	5,89	4,08	3,70	3,49	3,33	3,20	3,09	2,99	2,90	2,82
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	33,91	28,07	29,77	32,34	35,18	38,14	41,17	44,23	47,32	50,43
$16384\ б1та \mid n_a = 512,\ 16384\ бiта \mid n_b = 512$										
k	953	2493	4033	5573	7113	8654	10194	11734	13274	14814
$I_L\left(A_{2.1}^{(2)}\right)$	2516582	6459228	10401874	14344520	18287165	22229811	26172457	30115103	34057748	38000394
$I_L\left(A_{2.2}^{(2)}\right)$	643053	1740849	2914587	4161002	5481341	6875606	8343794	9885908	11501945	13191908
$I_C\left(A_{2.1}^{(2)}\right)$	187190	560389	1031167	1603224	2270180	3034717	3896833	4856531	5913809	7068667
$I_C\left(A_{2.2}^{(2)}\right)$	2387	6237	10087	13937	17788	21638	25488	29338	33189	37039
$I_L\left(A_{2.1}^{(2)}\right)/ I_L\left(A_{2.2}^{(2)}\right)$	3,91	3,71	3,57	3,45	3,34	3,23	3,14	3,05	2,96	2,88
$I_C\left(A_{2.1}^{(2)}\right)/ I_C\left(A_{2.2}^{(2)}\right)$	78,43	89,85	102,23	115,03	127,63	140,25	152,89	165,54	178,19	190,85

Порівнюючи теоретичну оцінку обчислювальної складності двох алгоритмів, варто зауважити: кількість операцій порівнянь було компенсовано за рахунок збільшення числа арифметичних операцій.

2.4.Результати експериментальних оцінок швидкодії розробленого методу

В табл. 2.4.1-2.4.2 представлено порівняння швидкодії методу прототипу і удосконаленого, на основі програмної реалізації за допомогою Visual C++2010. Заміри часу проводилися для 100 000 операцій, за допомогою обчислювальних систем Intel Core i3 M350 (PC1) і Intel Xeon E5 - 2640 (PC2), під управлінням ОС Windows 7 SP1 x86-64. Двійкова довжина діленого змінюється в діапазоні від 64

біт до 16 384 біт. Ширина машинного слова $w = 32$. Таблиця з оцінками розбита на блоки, кожен з яких описує відповідну частину результатів. ρ - відношення числа використовуваних біт до числа зарезервованих у великому числі (дільник); div – реалізація алгоритму прототипу; div*- реалізація алгоритму, удосконаленого методу.

Таблиця 2.4.1

Експериментальні оцінки часу виконання операції ділення для випадку (1)

	PC1	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**
ρ	0,1		0,2		0,5		0,6		0,8		0,9	
	Час, мкс											
Кількість машинних слів: діленого - 8 (256 біт), дільника - 8 (256 біт)												
div*	17,60	10,55	14,40	8,75	8,95	6,74	7,21	3,40	3,89	2,79	1,36	1,28
div	30,19	17,89	25,65	15,07	15,01	9,95	11,11	6,49	5,54	3,70	2,26	1,65
	1,72	**1,70**	**1,78**	**1,72**	**1,68**	**1,48**	**1,54**	**1,91**	**1,42**	**1,33**	**1,66**	**1,29**
	Час, мс											
Кількість машинних слів: діленого - 16 (512 біт), дільника - 16 (512 біт)												
div*	54,54	32,40	51,39	31,26	32,70	18,72	27,33	14,52	14,20	8,00	6,88	4,01
div	106,0	65,93	94,49	59,81	57,16	34,62	53,77	26,96	24,87	13,31	11,79	6,52
	1,94	**2,03**	**1,84**	**1,91**	**1,75**	**1,85**	**1,97**	**1,86**	**1,75**	**1,66**	**1,71**	**1,63**
Кількість машинних слів: діленого - 32 (1024 біт), дільника - 32 (1024 біт)												
div*	0,188	0,111	0,178	0,102	0,101	0,068	0,089	0,057	0,047	0,026	0,021	0,014
div	0,396	0,260	0,383	0,231	0,207	0,139	0,182	0,111	0,093	0,051	0,046	0,027
	2,11	**2,34**	**2,15**	**2,26**	**2,05**	**2,04**	**2,04**	**1,95**	**1,98**	**1,96**	**2,19**	**1,93**
Кількість машинних слів: діленого - 128 (4096 біт), дільника - 128 (4096 біт)												
div*	2,059	1,606	1,996	1,388	1,294	0,936	1,029	0,765	0,546	0,39	0,296	0,187
div	6,24	4,274	5,538	3,729	3,354	2,262	2,621	1,794	1,295	0,874	0,655	0,421
	3,03	**2,66**	**2,77**	**2,69**	**2,59**	**2,42**	**2,55**	**2,35**	**2,37**	**2,24**	**2,21**	**2,25**
Кількість машинних слів: діленого - 512 (16184 біт), дільника - 512 (16184 біт)												
div*	32,42	23,38	30,46	21,33	19,67	14,9	16,05	11,79	8,424	5,959	4,306	3,057
div	97,5	66,35	63,57	59,69	52,21	35,4	41,28	28,07	20,28	13,56	10,08	6,71
	3,01	**2,84**	**2,09**	**2,80**	**2,65**	**2,38**	**2,57**	**2,38**	**2,41**	**2,28**	**2,34**	**2,19**

За допомогою запропонованого методу ділення великих цілих чисел, вдалося підвищити швидкодію програмної реалізації в 1,5-3 рази порівнюючи з прототипом з ростом двійкової довжини цілого числа: у випадку (1) для чисел довжиною від 512 біт і у випадку (2) - від 128 біт.

Таблиця 2.4. 2

Експериментальні оцінки часу виконання операції ділення для випадку (2)

	PC1	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**	**PC1**	**PC2**
ρ	0,1		0,2		0,4		0,5		0,8		0,9	
t	**Час, мкс**											
Кількість машинних слів: діленого - 4 (128 біт), дільника - 2 (64 біта)												
div*	5,241	3,634	4,836	3,026	4,914	3,026	4,571	2,668	3,058	2,637	2,964	2,043
div	8,112	5,772	7,707	5,211	7,036	4,899	6,52	4,43	4,898	3,728	4,399	3,089

Продовження таблиці 2.4.2

	1,55	**1,59**	**1,59**	**1,72**	**1,43**	**1,62**	**1,43**	**1,66**	**1,60**	**1,41**	**1,48**	**1,51**
	Час, мс											
Кількість машинних слів: діленого - 16 (512 біт), дільника - 8 (256 біт)												
div*	0,049	0,039	0,047	0,033	0,041	0,030	0,040	0,030	0,030	0,025	0,030	0,022
div	0,098	0,072	0,092	0,062	0,080	0,057	0,075	0,055	0,058	0,043	0,053	0,038
	2,00	**1,85**	**1,96**	**1,88**	**1,95**	**1,90**	**1,88**	**1,83**	**1,93**	**1,72**	**1,77**	**1,73**
Кількість машинних слів: діленого - 32 (1024 біта), дільника - 16 (512 біт)												
div*	0,156	0,129	0,160	0,120	0,139	0,117	0,131	0,108	0,113	0,096	0,108	0,083
div	0,380	0,277	0,360	0,256	0,309	0,235	0,289	0,210	0,229	0,172	0,209	0,152
	2,44	**2,15**	**2,25**	**2,13**	**2,22**	**2,01**	**2,21**	**1,94**	**2,03**	**1,79**	**1,94**	**1,83**
Кількість машинних слів: діленого - 128 (4096 біт), дільника - 64 (2048 біт)												
div*	2,371	1,732	2,169	1,653	1,996	1,514	1,919	1,419	1,575	1,139	1,31	1,129
div	6,224	4,305	5,803	4,072	5,133	3,572	4,805	3,276	3,791	2,590	3,37	2,32
	2,63	**2,49**	**2,68**	**2,46**	**2,57**	**2,36**	**2,50**	**2,31**	**2,41**	**2,27**	**2,57**	**2,05**
Кількість машинних слів: діленого - 512 (16184 біт), дільника - 256 (8132 біт)												
div*	35,01	25,05	32,65	24,62	30,47	21,72	28,88	21,08	23,74	17,38	21,56	15,89
div	96,3	63,94	90,68	60,67	79,69	53,17	74,47	49,45	58,36	39,48	53,03	36,1
	2,75	**2,55**	**2,78**	**2,46**	**2,62**	**2,45**	**2,58**	**2,35**	**2,46**	**2,27**	**2,46**	**2,27**

Час виконання відрізняється через архітектурних особливостей сучасних процесорів з суперскалярною архітектурою. На цей час впливає: знаходження даних і коду програми в кеші першого і другого рівня, а також механізм передбачення переходів і цілий ряд інших параметрів. Відповідно до теорії ймовірності математичній статистиці, такі процеси зазвичай описуються нормальним законом. Перевірка статистичної гіпотези щодо однорідності двох незалежних вибірок – часу реалізації методу прототип та удосконаленого – виконувалась за допомогою прогнозованого аналітичного програмного забезпечення IBM SPSS. Виконувалось 114 повторень експерименту з 100 тис. операцій (див. додаток В). Точність виміру часу становить 100 нс – для

процесорної архітектури x86 Intel Core i7-6700 HQ. Значимість перевірялася за допомогою t-теста Стьюдента (див. табл. В.2, додаток В) і теста Манна-Уітні (див. табл. 2.4.3) [5].

Таблиця 2.4. 3

Статистичний критерій Манна-Уітні для випадку, коли кількість машинних слів діленого та дільника - 1024 (32368)

ρ	**0,1**	**0,2**	**0,3**	**0,4**	**0,5**	**0,6**	**0,7**	**0,8**	**0,9**
U Манна-Уітні	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
W Вілкоксона	6555	6555	6555	6555	6555	6555	6555	6555	6555
Z	-13,048	-13,048	-13,048	-13,048	-13,048	-13,048	-13,048	-13,048	-13,048
Асимпт. зн.(2-сторн.)	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**

Оскільки p-рівень критерію Лівіня ≤0,05, то дисперсії порівнюваних розподілів значень статистично достовірно різняться, і тест Стьюдента застосовувати не можна (див. табл. В.2, додаток В). Тому проводився U-тест Манна-Уітні (див. табл.2.4.3), на основі якого зроблено висновок (асимптотична значимість <0,05), що відмінності між вибірками є статистично значущими – швидкодія запропонованого та відомого методів значно відрізняється. Запропонований метод має кращу швидкодію.

2.5. Продуктивність удосконаленого методу для Національної системи ЕЦП України

В табл. 2.5.1 представлено час реалізації генерації ключів для RSA з використанням алгоритму прототипу і удосконаленого. Використовувався генератор псевдовипадкових послідовностей з ДСТУ 4145-2002 [6] на основі ГОСТ 28147-89. Для генерації простих чисел p і q застосовувався тест Рабіна-Міллера [7], де число випробувань дорівнює 50. Для наочності, наводяться значення для відкритої експоненти $e = 65537$. Програмна реалізація відтворена за допомогою Visual C++2015 на Intel Core i7-6700HQ (4 cores, 8 threads, 2.6 GHz, 6 MB Smart Cache, RAM 16GB) під управлінням Windows 10 x86-64.

Удосконалений метод дозволив підвищити швидкодію генерації ключів RSA на 7-14% (див. табл. 2.5.1).

Таблиця 2.5. 1

Результати експериментальної оцінки генерації ключів RSA

Довжина ключа	Час, с		Виграш	Довжина ключа	Час, с		Виграш
	Відомий метод	Удосконалений метод			Відомий метод	Удосконалений метод	
512	0.014	0.013	1.077	4096	3.153	2.803	1.125
1024	0.0762	0.07	1.089	8192	147.107	129.166	1.139
2048	0.3799	0.345	1.101				

2.6. Висновки до другого розділу

За результатами виконання досліджень, були сформульовані наступні висновки:

1. Проаналізовано метод прототип ділення в стовпчик та сформульовані підходи до його удосконалення: використовувати порівняння номерів старших бітів для зменшення трудомістких операцій порівняння великих цілих чисел; виконувати зсуви, додавання і віднімання та порівняння великих чисел лише по значущим словам.

2. Проведена оцінка обчислювальної складності методу прототипу та удосконаленого методу, показала виграш:

– в 1,46-190,85 раз при порівнянні кількості операцій порівняння і в 1.93-2,88 раз (починаючи з числа 256 біт) при порівнянні арифметичних операцій з ростом двійкової довжини великого цілого числа для випадку (1);

– в 1,24-196,91 раз при порівнянні кількості операцій порівняння і виграш в 1,34-3,26 разів (починаючи з числа 256 біт) при порівнянні арифметичних операцій з ростом двійкової довжини великого цілого числа для випадку (2).

Слід зауважити, що аналітична оцінка складності збіглася з результатами моделювання. Вдалося скоротити кількість обчислювально складних операцій порівняння за рахунок арифметичних операцій, які швидше виконуються сучасними процесорами і їх можна розпаралелювати, на відміну від операцій порівняння.

3. Проведена експериментальна оцінка швидкодії програмної реалізації методу прототипу та удосконаленого, показала лінійну залежність часу виконання від співвідношення одиничних бітів і щільності дільника в діапазоні $\rho \in \{0.1,\ 0.2,\ \ldots,\ 0.9\}$.

4. Удосконалений метод ділення великих цілих чисел дозволив підвищити швидкодію програмної реалізації в 1,5-3 рази порівнюючи прототип з ростом двійкової довжини цілого числа: в разі (1) для чисел довжиною від 512 біт і в разі (2) – від 128 біт.

5. Проведені статистичні тести показали, що удосконалений метод є більш швидкодіючим.

6. Удосконалений метод ділення великих цілих чисел в стовпчик, не орієнтований на багатопоточне виконання, що не дозволило повністю реалізувати потенціал сучасних багатоядерних процесорів.

Список використаних джерел у другому розділі

[1] D. Stehlé and P. Zimmermann. «A Binary Recursive Gcd Algorithm», *Lecture Notes in Computer Science*, pp. 411-425, 2004.

[2] L. Lhote and B. Vallée. «Sharp Estimates for the Main Parameters of the Euclid Algorithm», *LATIN 2006: Theoretical Informatics*, pp. 689-702, 2006.

[3] М.Г. Ковтун, В.Ю. Ковтун. «Подходы к повышению производительности операции деления больших целых чисел, на основе расширенного алгоритма Евклида» в *Информационные технологии и защита информации в информационно-коммуникационных системах: раздел коллективной монографии*. В.С. Пономаренко, Харьков: ТОВ “Щедра садиба плюс”, 2015, с. 208-219.

[4] М.Г. Ковтун, В.Ю. Ковтун, С.А. Гнатюк, О.М. Бердник, «Подходы к повышению производительности расширенного алгоритма Евклида для деления больших чисел двойной точности на большие числа одинарной точности», *Безпека інформації*, том 21, № 1, с. 40-51, 2015.

[5] П. Приставка, *Статистичний аналіз даних*. Україна, Київ: Національний авіаційний університет, Кафедра прикладної математики, 2017, с. 48-64.

[6] Національні стандарти України, «ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка», Київ, 2002.

[7] «Тест Миллера — Рабина», *Ru.wikipedia.org*, 2018. [Online]. Available: Wikipedia.

РОЗДІЛ 3
РОЗРОБКА МЕТОДІВ ПІДВИЩЕННЯ ШВИДКОДІЇ АРИФМЕТИЧНИХ ОПЕРАЦІЙ У ДВІЙКОВОМУ ПОЛІ

3.1. Мультиплікативне інвертування на основі розширеного алгоритма Евкліда

Розглянемо алгоритм мультиплікативного інвертування, який дозволяє обчислювати обернене для не нульового елемента $a \in \boldsymbol{GF}(2^m)$, використовуючи розширений алгоритму Евкліда (РАЕ) для поліномів [1]. Алгоритм [1] оснований на двох інваріантах $ba + df = u$ та $ca + ef = v$ для деяких d і e, які обчислюються неявно. У кожній ітерації, якщо виконується умова $\deg(u) \geq \deg(v)$, частковий розподіл u через v, виконуючи зсуви $x^j v$ для u, де $j = \deg(u) - \deg(v)$. Отже, степінь полінома u залишатиметься сталою, або зменшиться хоча б на 1. Складання $x^j c$ з b дозволяє зберегти інваріанти. Алгоритм виконує в середньому $\deg(a) = k$ ітерацій і зупиняється, коли виконується умова $\deg(u) = 0$: $u = 1$ і $ba + df = 1$, $b = a^{-1} \bmod f(x)$.

Алгоритм 3.1. Розширений алгоритм Евкліда для мультиплікативного інвертування в полі $\boldsymbol{GF}(2^m)$.

Вхід: елемент $a \in \boldsymbol{GF}(2^m)$, $a \neq 0$.

Вихід: $a^{-1} \bmod f(x)$.

1. $b \leftarrow 1$, $c \leftarrow 0$, $u \leftarrow a$, $v \leftarrow f$.

2. While $\deg(u) \neq 0$ do

2.1. $j \leftarrow \deg(u) - \deg(v)$.

2.2. if $j < 0$ then $u \leftrightarrow v$, $b \leftrightarrow c$, $j \leftarrow -j$.

2.3. $u \leftarrow u + x^j v$, $b \leftarrow b + x^j c$.

3. Return (b).

Рис.3.1. 1. Псевдокод реалізації мультиплікативного інвертування на основі розширеного алгоритму Евкліда

Проведений аналіз алгоритму 3.1, дозволяє виділити ряд аспектів для подальшого вдосконалення РАЕ:

– На кроці п. 2.2 та п. 2.3 відбувається модифікація полінома u, в той час як поліном v містить попереднє значення полінома u. Це дозволяє відмовитися від обчислення степеня полінома v на кроці п. 2.1. Що стосується первинного обчислення $\deg(v)$ на кроці п. 2.1, то степінь відома наперед і є константою $\deg(v) = \deg(f) = m$.

– На кроці п. 2.1 відбувається обчислення степеню полінома u: він постійно зменшується, хоча б на 1. Це дозволяє відмовитися від обчислення степеня $\deg(u)$ на кожній ітерації в загальному вигляді, а лише займатися його уточненням.

– На кроці п. 2.3 проводиться зсув полінома v і подальше, його складання з u, Проте слід зауважити, що в процесі виконання циклу п. 2 степені поліномів v і u постійно зменшуються, а степені поліномів b і c постійно росте. Це дозволяє виконувати зсув і складання не всіх елементів масиву, в якому представлені елементи поля, а лише значущих – відмінні від нуля.

3.2. Удосконалений метод мультиплікативного інвертування на основі розширеного алгоритму Евкліда

Детального розгляду потребує безпосередньо сам алгоритм обчислення степеня довільного полінома a: для представлення елементів поля $\boldsymbol{GF}(2^m)$, використовується поліноміальний базис.

Елемент поля $b \in \boldsymbol{GF}(2^m)$ в поліноміальному базисі представляється двійковим вектором $b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \cdots + b_1x^1 + x_0$, таким, що можна представити у вигляді $a_{n-1}^{(w)}2^{(n-1)w} + a_{n-2}^{(w)}2^{(n-2)w} + \cdots + a_1^{(w)}2^w + a_0^{(w)}$ масиву машинних слів з двійковою довжиною w, де $n = \lceil \frac{m}{w} \rceil$ – число машинних слів, необхідних для подання полінома двійкової довжини m; b_i – двійкові коефіцієнти; $a_j^{(w)}$ – машинні слова двійкової довжини w.

Ідея алгоритму обчислення степеню полінома полягає в пошуку номера найстаршого елемента масиву $a_j^{(w)}$, відмінного від нуля, і знаходження номера найстаршого одиничного біта в знайденому елементі масиву $a_j^{(w)}$, який полягає в послідовному переборі елементів, починаючи з кінця, до тих пір, поки не зустрінеться шуканий. Що стосується визначення номера старшого одиничного біта в елементі масиву (машинному слові), то відомо досить багато алгоритмів [2], які вимагають послідовного перебору, тобто значних обчислювальних ресурсів. Підвищити швидкодію обчислення номера старшого одиничного біта в слові, авторами, пропонується скористатися цілою низкою відомих «трюків» [2], заснованих на бітових операціях машинних слів. В алгоритмі 3.2 викладено метод обчислення номера старшого біта полінома. Проведемо його більш детальний розгляд. На кроці п. 2, в циклі відбувається пошук найстаршого елемента масиву - машинне слово, відмінне від нуля. Пошук проводиться з самого старшого машинного слова до молодшого. На кроці п. 3 фіксується не нульовий елемент масиву і його номер. Номер елемента, в подальшому, буде використано для обчислення степеню полінома.

На кроці п. 4 застосовується «трюк» [2] - формування «маски», який дозволяє заповнити одиничними бітами всі молодші біти від найстаршого. Далі необхідно підрахувати кількість одиничних бітів в машинному слові, щоб визначити номер старшого біта.

Для підрахунку всіх одиничних бітів в машинному слові, на кроці п. 5, використовується «трюк» [2], причому їх число буде на одиницю більше ніж, номер найстаршого відмінного від нуля біта.

Степінь полінома, обчислюється на кроці п. 6, де враховується загальна кількість бітів в кожному машинному слові, номер найстаршого відмінного від нуля машинного слова, а також номер старшого біта в старшому слові.

Врахувавши в алгоритмі 3.1, результати аналізу, а також сам алгоритм обчислення степеня полінома (алгоритм 3.2) [3], був удосконалений РАЕ (МРАЕ) для мультиплікативного інвертування в полі $\boldsymbol{GF}\left(2^m\right)$, який

представлений у вигляді алгоритму 3.3. Через $msb(\cdot)$ позначимо функцію обчислення степеня полінома, U, V - степені поліномів u та v, відповідно.

Алгоритм 3.2. Алгоритм обчислення степеня полінома в полі $\boldsymbol{GF}(2^m)$.

Вхід: $a \in \boldsymbol{GF}(2^m)$; $n = \lfloor \frac{m}{w} \rfloor$, де n - число машинних слів, яке займає поліном; w - ширина машинного слова, зазвичай $w = 32$.

Вихід: $\deg(a)$.

1. $i \leftarrow n-1$.

2. While $\left(a_i^{(32)} \neq 0\right) \& \& (i > 0)$

2.1. $i \leftarrow i-1$.

3. $t \leftarrow a_i^{(32)}$.

4. $t \leftarrow t \mid (t >> 1)$, $t \leftarrow t \mid (t >> 2)$, $t \leftarrow t \mid (t >> 4)$, $t \leftarrow t \mid (t >> 8)$, $t \leftarrow t \mid (t >> 16)$.

5. $t \leftarrow t - ((t >> 1) \& \text{ 0x55555555 })$, $t \leftarrow (t \& \text{ 0x33333333 }) + ((t >> 2) \& \text{ 0x33333333 })$,
$t \leftarrow ((t + (t >> 4) \& \text{ 0xf0f0f0f }) \cdot \text{0x1010101 }) >> 24$.

6. Return $((i << 5) + t - 1)$.

Рис. 3.2. 1. Псевдокод операції обчислення степеню полінома

На кроці п. 1 проводиться ініціалізація поліномів b, c, u і v, які будуть модифікуватися протягом реалізації алгоритму. Далі на кроці п. 2 визначається степінь полінома u, а v завідома відома: $v = m$.

Цикл п. 3 виконується до тих пір, поки U відмінна від 0, в середньому число ітерацій $\deg(a) = k$. У циклі, на кроці п. 3.1 проводиться перевірка степенів U і V поліномів u і v, відповідно. В разі $(U < V)$, необхідно зробити обмін вмісту поліномів u, v і b, c, відповідно. В середньому, число перевірок $(U < V)$ становить $k/3$, обчислюється різниця $(U - V)$, яка в подальшому використовується для зсуву поліномів v і c. На кроках п. 3.3 та п. 3.4 проводиться очищення старшого біта полінома u і установка відповідних бітів полінома b. Відзначимо, що на кроках п. 3.3 та п. 3.4 слід оперувати лише значущими машинними словами, тобто тими, які містять біти менші $\deg(b)$ і

$\deg(v)$. Врахуємо, що степінь полінома b росте, а степінь u - зменшується, причому $\deg(b)+\deg(u)=m$.

Алгоритм 3.3. Удосконалений розширений алгоритм Евкліда для мультиплікативного інвертування в полі $GF(2^m)$.

Вхід: $a \in GF(2^m)$, $a \neq 0$, $n = \left\lfloor \frac{m}{w} \right\rfloor$, де n -число машинних слів, яке займає поліном; w - ширина машинного слова, зазвичай $w = 32$.

Вихід: $a^{-1} \bmod f(x)$.

1. $b \leftarrow 1$, $c \leftarrow 0$, $u \leftarrow a, v \leftarrow f$.

2. $U = msb(u)$, $V = m$.

3. While $(U > 0)$ do

3.1. if $(U < V)$ then $k \leftarrow V - U$, $U \leftrightarrow V$, $u \leftrightarrow v$, $b \leftrightarrow c$.

3.2. else $k \leftarrow U - V$.

3.3. if $(k > 0)$ then $u = u + x^k \cdot v$, $b = b + x^k \cdot c$.

3.4. else $u = u + v$, $b = b + c$.

3.5. $U = msb(u)$.

4. Return (b).

Рис. 3.2. 2. Псевдокод удосконаленого методу мультиплікативного інвертування в двійковому полі

Обчислення степеня полінома u здійснюється на кроці п.3.5. Після завершення циклу п. 3, алгоритм повертає поліном b, такий що $b = a^{-1} \bmod f(x)$.

3.3. Розробка методу автоматизації приведення довільного полінома за фіксованим модулем у двійковому полі

Даний метод можна застосувати для побудови алгоритму на основі довільного тричлена і п'ятичлена, який не приводиться. Коротко зупинимося на загальній ідеї алгоритму приведення за фіксованим модулем [1, 4-6]. В результаті множення або піднесення до квадрату у двійковому полі, довжина полінома, який потрібно привести за модулем, становить подвійну довжину полінома, що не приводиться з максимально можливим степенем $2(k-1)$. Розглянемо дану

ситуацію на прикладі п’ятичлена $f(x)=x^k+x^l+x^g+x^e+1$, де $k>l>g>e$. Виконання приведення (додавання за модулем 2) полінома за модулем відбувається послівно, тобто розглядаються всі біти слова за один раз, що дозволяє в рази зменшити число ітерацій. Для цього на рис. 3.3.1 продемонстровано ідею алгоритму:

$$\begin{array}{cccc} x^{k+r} & x^{k+r-1} & \cdots & x^{k+r-(w-1)} \\ x^{l+r} & x^{l+r-1} & \cdots & x^{l+r-(w-1)} \\ x^{g+r} & x^{g+r-1} & \cdots & x^{g+r-(w-1)} \\ x^{e+r} & x^{e+r-1} & \cdots & x^{e+r-(w-1)} \\ x^{r} & x^{r-1} & \cdots & x^{r-(w-1)} \end{array}$$

Рис.3.3. 1. Представлення модуля після вирівнювання

де горизонтально записані біти, що утворюють одне слово, довжиною w-біт, а r ($r=\lceil 2(k-1)/w\rceil\cdot w-k$) - це різниця вирівнювання полінома, що не приводиться, під максимально можливу степінь результуючого полінома.

Описане представлення, дозволяє формувати слово з бітів $\left(x_{k+r},x_{k+r-1},\ldots,x_{k+r-(w-1)}\right)$ і складати його за модулем з бітами $\left(x_{l+r},x_{l+r-1},\ldots,x_{l+r-(w-1)}\right)$, $\left(x_{g+r},x_{g+r-1},\ldots,x_{g+r-(w-1)}\right)$, $\left(x_{e+r},x_{e+r-1},\ldots,x_{e+r-(w-1)}\right)$, $\left(x_{r},x_{r-1},\ldots,x_{r-(w-1)}\right)$. Дану операцію необхідно повторити зменшуючи на кожній ітерації значення r на величину w, поки $r>0$. На останній ітерації слід розглянути вже біти не всього слова, а лише його частини $r \bmod w$.

Запропонований метод побудови, дозволяє записати в загальному вигляді алгоритм приведення по модулю для п’ятичлена.

Оскільки деякі значення z_1, z_2, z_3 і z_4 можуть бути рівними, то можливе об'єднання кроків 4.1-4.3, а також кроків 6-9 (рис. 3.3.2).

Алгоритм 3.4. Розроблений метод приведення по фіксованому п'ятичленну, що не приводиться.

Вхід: поліном $c(x)$ степеню не більше $2(k-1)$.

Вихід: поліном $d(x) \equiv c(x) (\mathrm{mod} f(x))$.

1. $m \leftarrow \lceil 2(k-1)/w \rceil - 1$, $n \leftarrow \lceil k/w \rceil$, $r \leftarrow \lceil 2(k-1)/w \rceil \cdot w - k$, $s_1 \leftarrow \lceil (l+r)/w \rceil \cdot w - (l+r)$, $z_1 \leftarrow m - (\lceil (l+r)/w \rceil - 1)$
2. $s_2 \leftarrow \lceil (g+r)/w \rceil \cdot w - (g+r)$, $z_2 \leftarrow m - (\lceil (g+r)/w \rceil - 1)$, $s_3 \leftarrow \lceil (e+r)/w \rceil \cdot w - (e+r)$, $z_3 \leftarrow m - (\lceil (e+r)/w \rceil - 1)$
3. $s_4 \leftarrow \lceil r/w \rceil \cdot w - r$, $z_4 \leftarrow m - (\lceil r/w \rceil - 1)$
4. for $i \leftarrow m; i \geq n; i--$
4.1. $t \leftarrow c_i$, $d_{i-z_1} \leftarrow c_{i-z_1} \oplus (t >> s_1)$, $d_{i-z_1-1} \leftarrow c_{i-z_1-1} \oplus (t << (w-s_1))$, $d_{i-z_2} \leftarrow c_{i-z_2} \oplus (t >> s_2)$
4.2. $d_{i-z_2-1} \leftarrow c_{i-z_2-1} \oplus (t << (w-s_2))$, $d_{i-z_3} \leftarrow c_{i-z_3} \oplus (t >> s_3)$, $d_{i-z_3-1} \leftarrow c_{i-z_3-1} \oplus (t << (w-s_3))$
4.3. $d_{i-z_4} \leftarrow c_{i-z_4} \oplus (t >> s_4)$, $d_{i-z_4-1} \leftarrow c_{i-z_4-1} \oplus (t << (w-s_4))$.
5. $t \leftarrow c_{n-1}(x_{w-1}, x_{w-2}, \ldots, x_{(k \bmod w)}, 0_{(k \bmod w)-1}, \ldots, 0_0)$ // розглядаються старші біти, від $(w-1)$ до $(k \bmod w)$, решта бітів ігноруються.
6. if $(n-z_1) \geq 0$ then $d_{n-z_1} \leftarrow c_{n-z_1} \oplus (t >> s_1)$, if $(n-z_1-1) \geq 0$ then $d_{n-z_1-1} \leftarrow c_{n-z_1-1} \oplus (t << (w-s_1))$
7. if $(n-z_2) \geq 0$ then $d_{n-z_2} \leftarrow c_{n-z_2} \oplus (t >> s_2)$, if $(n-z_2-1) \geq 0$ then $d_{n-z_2-1} \leftarrow c_{n-z_2-1} \oplus (t << (w-s_2))$
8.if $(n-z_3) \geq 0$ then $d_{n-z_3} \leftarrow c_{n-z_3} \oplus (t >> s_3)$, if $(n-z_3-1) \geq 0$ then $d_{n-z_3-1} \leftarrow c_{n-z_3-1} \oplus (t << (w-s_3))$
9. if $(n-z_4) \geq 0$ then $d_{n-z_4} \leftarrow c_{n-z_4} \oplus (t >> s_4)$, if $(n-z_4-1) \geq 0$ then $d_{n-z_4-1} \leftarrow c_{n-z_4-1} \oplus (t << (w-s_4))$
10. $d_{n-1} \leftarrow c_{n-1}(0_{w-1}, 0_{w-2}, \ldots 0_{(k \bmod w)}, x_{(k \bmod w)-1}, x_{(k \bmod w)-2}, \ldots, x_0)$. // розглядаються молодші біти від $(k \bmod w)-1$ до 0, решта бітів ігноруються.
11. Return $(d(x))$.

Рис.3.3. 2. Псевдокод побудови алгоритмів приведення за фіксованим модулем на прикладі п'ятичленна

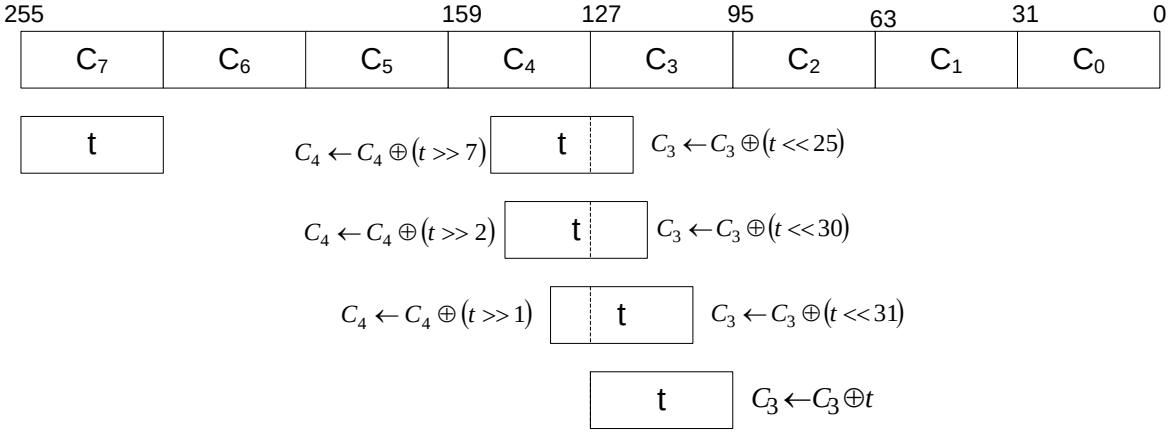

Рис.3.3.3. Додавання слова t $(t = C_7)$ з діапазонами бітів відповідними позиціями полінома, що приводиться: словами C_4 і C_3.

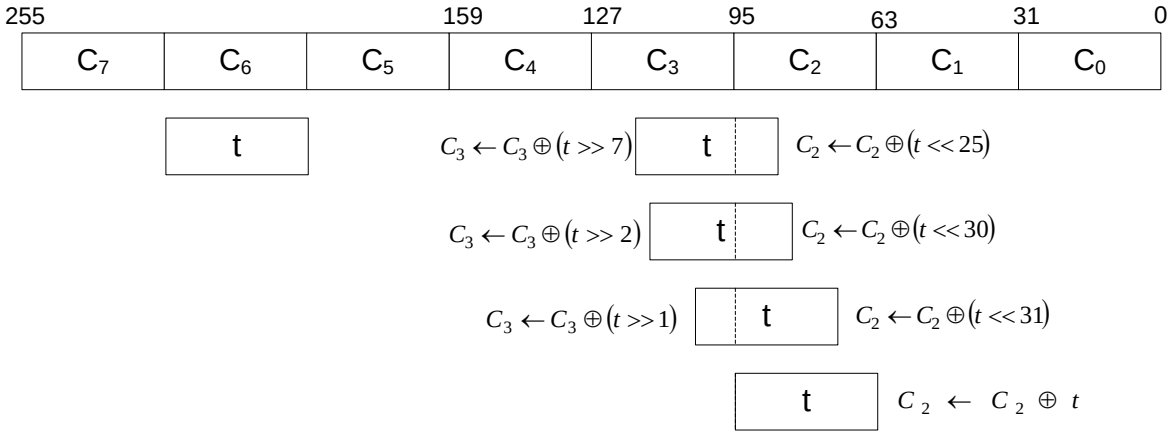

Рис.3.3.4. Додавання слова t $(t = C_6)$ з діапазонами бітів відповідними позиціями полінома, що приводиться: словами C_3 і C_2

На Рис.3.3.3-Рис.3.3.6 приведена графічна інтерпретація роботи алгоритму на прикладі полінома, що не приводиться $f_{128}(x) = x^{128} + x^7 + x^2 + x^1 + 1$, який використовується в блоковому симетричному шифрі ДСТУ 7624:2014 для 32-х розрядної цільової платформи. Машинні слова полінома, що приводяться, позначаються через C_0, C_1, …, C_7. Через t позначимо старше машинне слово полінома, який потрібно привести (рис.3.3.3). Після чого його потрібно скласти з відповідними діапазонами бітів цього ж полінома, на відповідних позиціях (розділивши, за допомогою зсувів, на старші та молодші біти), які визначаються степенями полінома, що не приводиться 7, 2, 1.

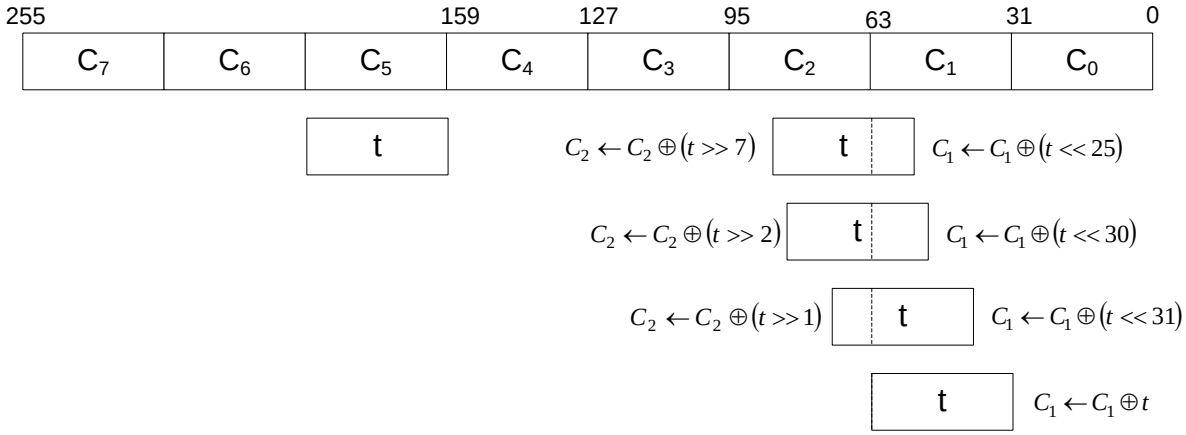

Рис.3.3.5. Додавання слова t $(t = C_5)$ з діапазонами бітів відповідними позиціями полінома, що приводиться: словами C_2 і C_1

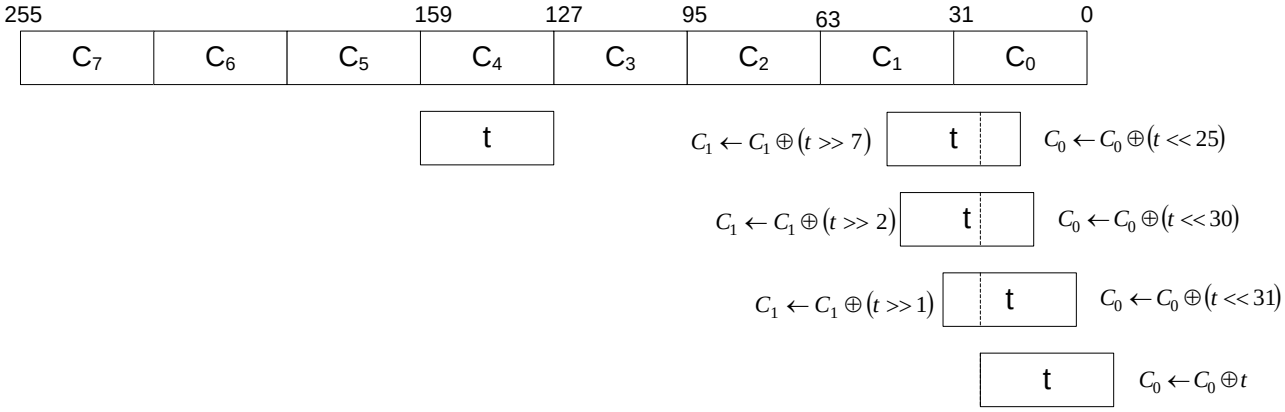

Рис.3.3.6. Додавання слова t $(t = C_4)$з діапазонами бітів відповідними позиціями полінома, що приводиться: словами C_1 і C_0.

На рис. 3.3.7 представлений псевдокод побудови алгоритму приведення полінома за модулем $f_{128}(x) = x^{128} + x^7 + x^2 + x^1 + 1$.

Алгоритм 3.5. Приведення по фіксованому п'ятичлену, що не приводиться $f_{128}(x) = x^{128} + x^7 + x^2 + x^1 + 1$ для 32-розрядних платформ.

Вхід: поліном $c(x)$ з максимальним степенем 254, $w \leftarrow 32$.

Вихід: поліном $d(x) \equiv c(x) \bmod f(x)$.

1. $m \leftarrow 7$, $n \leftarrow 4$.
2. $s_1 \leftarrow 25$, $z_1 \leftarrow 3$.
3. $s_2 \leftarrow 30$, $z_2 \leftarrow 3$.
4. $s_3 \leftarrow 31$, $z_3 \leftarrow 3$.
5. $s_4 \leftarrow 0$, $z_4 \leftarrow 3$.
6. for $i \leftarrow m; i \geq n; i--$
6.1. $t \leftarrow c_i$.
6.2. $d_{i-z_1} \leftarrow c_{i-z_1} \oplus (t >> s_1)$, $d_{i-z_1-1} \leftarrow c_{i-z_1-1} \oplus (t << (w - s_1))$.
6.3. $d_{i-z_2} \leftarrow c_{i-z_2} \oplus (t >> s_2)$, $d_{i-z_2-1} \leftarrow c_{i-z_2-1} \oplus (t << (w - s_2))$.
6.4. $d_{i-z_3} \leftarrow c_{i-z_3} \oplus (t >> s_3)$, $d_{i-z_3-1} \leftarrow c_{i-z_3-1} \oplus (t << (w - s_3))$.
6.5. $d_{i-z_4} \leftarrow c_{i-z_4} \oplus (t >> s_4)$, $d_{i-z_4-1} \leftarrow c_{i-z_4-1} \oplus (t << (w - s_4))$.
7. $d_{n-1} \leftarrow 0$.
8. Return $(d(x))$.

Рис.3.3. 7. Псевдокод реалізації приведення по фіксованому модулю для 32-розрядної платформи

Оскільки значення z_1, z_2, z_3 і z_4 - рівні, тоді алгоритм 3.5 може бути спрощений до алгоритму 3.6 (рис.3.3. 8).

Алгоритм 3.6. Приведення по фіксованому п’ятичлену, що не приводиться $f_{128}(x)=x^{128}+x^7+x^2+x^1+1$ для 32-розрядних платформ.
Вхід: поліном $c(x)$ максимальний степінь 254, $w \leftarrow 32$.
Вихід: поліном $d(x) \equiv c(x) \bmod f(x)$.
1.for $i \leftarrow 7; i \geq 4; i--$
1.1. $t \leftarrow c_i$.
1.2. $d_{i-3} \leftarrow c_{i-3} \oplus (t >> 25) \oplus (t >> 30) \oplus (t >> 31)$.
1.3. $d_{i-4} \leftarrow c_{i-4} \oplus (t << 7) \oplus (t << 2) \oplus (t << 1) \oplus t$.
2. $d_4 \leftarrow 0$.
3.Return $(d(x))$.

Рис.3.3. 9. Псевдокод реалізації приведення по фіксованому модулю для 32-розрядної платформи

Розглянемо алгоритм приведення по фіксованому п’ятичлену для 64-розрядних платформ. Даний алгоритм буде виглядати дещо простіше, оскільки оперує меншим числом машинних слів (алгоритм 3.7).

Алгоритм 3.7. Приведення по фіксованому п’ятичлену, що не приводиться $f_{128}(x)=x^{128}+x^7+x^2+x^1+1$ для 64-розрядних платформ.
Вхід: поліном $c(x)$ з максимальним степенем 254, $w \leftarrow 64$.
Вихід: поліном $d(x) \equiv c(x) \bmod f(x)$.
1.for $i \leftarrow 3; i \geq 2; i--$
1.1. $t \leftarrow c_i$.
1.2. $d_{i-1} \leftarrow c_{i-1} \oplus (t >> 57) \oplus (t >> 62) \oplus (t >> 63)$.
1.3. $d_{i-2} \leftarrow c_{i-2} \oplus (t << 7) \oplus (t << 2) \oplus (t << 1) \oplus t$.
2. $d_2 \leftarrow 0$.
3. Return $(d(x))$.

Рис.3.3. 10. Псевдокод реалізації приведення по фіксованому модулю для 64-розрядної платформи

3.4. Оцінка обчислювальної складності

Для порівняння обчислювальної складності алгоритму 3.1 та алгоритму 3.3 для мультиплікативного інвертування введемо позначення:

– $I_{\deg}$ – складність алгоритму обчислення степеня полінома;

– I_{add} – складність алгоритму складання двох поліномів;

– I_{shl} – складність алгоритму зсуву на довільне число біт (може перевищувати довжину машинного слова);

– I_{swp} – складність алгоритму обміну двох поліномів.

Оцінка складності алгоритму 3.1 Нехай степінь полінома $\deg(a)=k$ і вага по Хеммінгу полінома $h=\text{weight}(a)=k/2$. Тоді кількість ітерацій циклу п. 2 складе k, і число істинних умов $j\neq 0$, складе $2k/3$. В останніх $k/3$ випадках зсуви не виконуються. Складність Алгоритму 3.1 матиме вигляд:

$$I_{avr}(A_{3.1})=k\left(2I_{\deg}+2I_{add}+2I_{shl}\right)+2k/3\left(2I_{swp}\right) \quad (3.4.1)$$

Складністю I_{swp} можна знехтувати в силу використання покажчиків, що не потребує виконання значного числа переприсвоєнь. У спрощеному вигляді (3.4.1) має вигляд:

$$I_{avr}(A_{3.1})=k\left(2I_{\deg}+2I_{add}+2I_{shl}\right) \quad (3.4.2)$$

Оцінка складності алгоритму 3.3. Нехай степінь $\deg(a)=k$ полінома a і вага по Хеммінгу $h=\text{weight}(a)=k/2$ полінома a. Тоді кількість ітерацій циклу п. 3 складе k, і число істинних умов $j\neq 0$ - складе $2k/3$. Складність Алгоритму 3.3 представлено нижче:

$$I_{avr}(A_{3.3})=k\left(I_{\deg}+\tfrac{1}{2}\left(2I_{add}+2I_{shl}\right)\right)+2k/3\left(2I_{swp}\right) \quad (3.4.3)$$

У спрощеному вигляді (3.4.3) матиме вигляд:

$$I_{avr}(A_{3.3})=k\left(I_{\deg}+I_{add}+I_{shl}\right) \quad (3.4.4)$$

Порівнюючи складність (3.4.2) і (3.4.4) видно, що її вдалося зменшити в два рази.

3.5. Результати експериментальних оцінок швидкодії розробленого методу

3.5.1. Експериментальна оцінка швидкодії реалізацій операції мультиплікативного інвертування на основі розширеного алгоритму Евкліда

Для оцінки ефективності запропонованих підходів по модифікації алгоритму мультиплікативного інвертування в полі $\boldsymbol{GF}\left(2^m\right)$, була виконана програмна реалізація за допомогою Visual C++ 2015 на мобільних процесорах Intel Core i3 M350 2,26 GHz і настільних процесорах Intel Core i5-3570 3,80 GHz

і Intel Core i5-4670 3,80 GHz під управлінням ОС Windows 7 SP1 x86-64. Експерименти проводилися для різних показників розширення двійкових полів m. За основу бралися двійкові поля з ДСТУ 4145-2002 [7] і NIST FIPS 186-4 [8]. В таблиці 3.5.1 наведено результати експериментальних досліджень програмної реалізації удосконаленого методу та прототипу, з використання компіляторів Intel (ICC) та Microsoft (MCC).

Результати експериментів показують, що степінь інвертуючого полінома не впливає на час обчислення оберненого елемента за допомогою PAE, в той час як MPAE показує лінійне зменшення часу обчислення оберненого елемента зі зменшенням його степеня.

Для процесора Intel Core i3-350M, MPAE показав виграш на 18-21% (ICC) і 16-18% (MCC), в той же час MCC реалізація виявилася ефективнішою на 2,2% ніж на ICC. Для процесора Intel Core i5-3570, MPAE показав виграш на 19-22% (ICC) і 17-22% (MCC), в той же час ICC реалізація виявилася ефективнішою на 4,9% ніж на MCC. Для процесора Intel Core i5-4670, MPAE показав виграш на 18-25% (ICC) і 14-22% (MCC), в той же час ICC реалізація виявилася ефективнішою до 14,7% ніж MCC.

Таблиця 3.5.1

Результати експериментальної оцінки часу виконання операції інвертування

m	**Час, мкс**											
	Intel Core i3-350M				**Intel Core i5-3570**				**Intel Core i5-4670**			
	ICC XE2013		**MCC2015**		**ICC XE2013**		**MCC2015**		**ICC XE2013**		**MCC2015**	
	Inv	**Inv***	**Inv**	**Inv***	**Inv**	**Inv***	**Inv**	**Inv***	**Inv**	**Inv***	**Inv**	**Inv***
89	6.71	5.44	6.27	5.1	2.53	1.95	2.76	1.84	2.53	1.95	2.37	1.84
163	16.08	11.34	14.38	11.96	6.85	4.05	6.33	4.65	6.85	3.95	6.13	4.05
191	18.17	14.52	17.38	14.71	7.73	5.46	7.85	5.48	7.73	5.26	7.65	5.48
233	27.13	22.12	24.57	19.39	11.8	7.04	11.59	7.65	11.8	7.02	11.02	6.9
257	31.56	25.18	27.93	24.54	13.21	7.99	13.33	8.56	12.17	7.81	12.33	7.6
307	37.72	30.45	34.24	26.11	17.98	11.11	17.64	12.41	17.68	9.58	17.54	11.41
367	53.18	42.17	46.05	33.81	23.35	14.78	21.84	16.63	22.35	13.18	21.64	14.63
409	61.31	40.18	54.48	47.76	26.41	16.97	26.81	18.51	25.97	14.93	26.41	17.51
431	67.75	54.24	59.1	44.03	28.99	17.86	29.29	19.25	28.27	17	28.99	18.25
571	103.4	64.86	94.42	70.26	46.46	25.47	44.87	26.98	43.39	24.64	44.67	26.83

- [*] - удосконалений метод;
- степінь полінома близька до m.

Отримані оцінки повністю відповідають теоретичним оцінками обчислювальної складності. Причому програмна реалізація запропонованого МРАЕ показала кращу швидкодію у 1,25-1,75 рази, ніж в роботі [1], для відповідних полів.

Проводилася статистична оцінка реалізації удосконаленого методу та прототипу для знаходження мультиплікативного оберненого. Досліджувалося 120 повторів експериментів з 100 тис. операцій (див.табл. В.3, Додаток В) для полів з ДСТУ 4145-2002. Точність виміру часу становить 100 нс – для процесорної архітектури x86 Intel Core i7-6700 HQ. Значимість перевірялася за допомогою t-теста Стьюдента (див. табл. В.3, додаток В) з 99,5% довірчим інтервалом, використовуючи систему для аналізу даних IBM SPSS. Результати тесту показали, що p – рівень критерію Лівіня ≤0.05 (див. табл. В.4, Додаток В), тому дисперсії порівнюваних розподілів значень статистично достовірно різняться. Тому надалі застосовувався критерій для незалежних вибірок Манна-Уітні (див. табл. 3.5.2), який показав значення асимптотичної значимості <0.05 – порівнювані значення з двох вибірок дійсно статистично достовірно різняться.

Таблиця 3.5.2

Статистичний критерій Манна-Уітні для мультиплікативного інвертування

Поле	**431**	**367**	**307**	**257**	**233**
U Манна-Уітні	0,000	0,000	0,000	0,000	0,000
W Вілкоксона	7260,000	7260,000	7260,000	7260,000	7260,000
Z	-13,399	-13,399	-13,398	-13,401	-13,409
Асимптотична значимість (2-сторонн)	**0,000**	**0,000**	**0,000**	**0,000**	**0,000**

3.5.2. Експериментальна оцінка швидкодії реалізацій операції приведення по модулю

У табл. 3.5.3Таблиця 3.5.3 наведено результати практичної реалізації запропонованого методу та побітового методу приведення за модулем на обчислювальній системі з процесором Intel Core i5-3570 під управлінням ОС Windows 7 SP1 x86-64. Програмна реалізація виконана за допомогою Visual C++ 2015.

Таблиця 3.5.3

Час реалізації побітового та послівного методів для 32-розрядних платформ

Поліном, що не приводиться	Побітовий метод, мс	Послівний метод, мс	Виграш
$f_{128}(x) = x^{128} + x^7 + x^2 + x^1 + 1$	0,324494	0,009327	34,8
$f_{256}(x) = x^{256} + x^{10} + x^5 + x^2 + 1$	1,151691	0,015098	76,28
$f_{512}(x) = x^{512} + x^8 + x^5 + x^2 + 1$	5,420564	0,027486	197,21

Виграш в швидкодії при використанні послівного методу в 34.8-197.21 разів у порівнянні з побітовим методом для поліномів приведених у ДСТУ 7624:2014.

3.6. Продуктивність удосконалених методів для Національної системи ЕЦП України

В табл. 3.6.1 представлено вплив удосконаленого методу інвертування при формуванні та перевірці ЕЦП згідно ДСТУ 4145-2002 [7]: СМ відбувалося у 2 потоки (метод Монтгомері) в проективних координатах Лопеса-Дахаба. Програмна реалізація відтворена за допомогою Visual C++ 2015 на Intel Core i7-6700HQ під управлінням Windows 10 x86-64.

Таблиця 3.6. 1

Результати експериментальної оцінки виконання формування і перевірки ЕЦП

m	Формування ЕЦП			Перевірка ЕЦП		
	Inv, мс	Inv*, мс	Виграш	Inv, мс	Inv*, мс	Виграш
163	1.693	1.69	1.0018	1.853	1.845	1.0042
167	1.747	1.744	1.0019	1.879	1.871	1.0043
173	1.871	1.868	1.0018	1.993	1.985	1.0040
179	2.002	1.999	1.0016	2.256	2.249	1.0033
191	2.324	2.32	1.0016	2.488	2.479	1.0035
233	3.522	3.516	1.0017	3.767	3.752	1.0041
257	4.541	4.534	1.0015	4.979	4.962	1.0034
307	6.506	6.497	1.0014	7.045	7.023	1.0032
367	9.559	9.547	1.0012	10.389	10.36	1.0028
431	13.615	13.6	1.0011	14.445	14.406	1.0027

- Inv – метод прототип мультиплікативного інвертування;
- Inv* - удосконалений метод мультиплікативного інвертування.

Вплив швидкодії операції інвертування розглянемо на прикладі середнього банку, як складової Національної системи ЕЦП, у продовж банківського дня при передачі 450 тис. документів (див. табл. 3.6.2). При передачі документів розглядається лише 1 постановка і 1 перевірка ЕЦП, насправді їх більше.

Таблиця 3.6. 2

Продуктивність удосконаленого методу інвертування для 1 банківського дня, як складової Національної системи ЕЦП

Поле, m	**Постановка ЕЦП, с**			**Перевірка ЕЦП, с**		
	Побітовий метод	**Послівний метод**	**Різниця**	**Побітовий метод**	**Послівний метод**	**Різниця**
163	762.39	760.55	1.85	834.15	830.25	3.90
167	786.74	784.80	1.94	846.05	841.95	4.10
173	842.58	840.60	1.98	897.36	893.25	4.11
179	901.53	899.55	1.98	1015.94	1012.05	3.89
191	1046.24	1044.00	2.24	1120.05	1115.55	4.50
233	1585.71	1582.20	3.51	1696.14	1688.40	7.74
257	2044.16	2040.30	3.86	2241.39	2232.90	8.49
307	2928.96	2923.65	5.31	3171.65	3160.35	11.30
367	4303.11	4296.15	6.96	4676.70	4662.00	14.70
431	6128.95	6120.00	8.95	6502.15	6482.70	19.45

Як показують дані табл. 3.6.2 виграш при використанні удосконаленого методу мультиплікативного інвертування, складає при постановці 1-9 с, при перевірці 3-20 с. В табл. 3.6.3 показано результати впливу використання побітового приведення за модулем та послівного при формуванні та перевірці ЕЦП згідно ДСТУ 4145-2002, використовуючи удосконалений метод мультиплікативного інвертування.

Таблиця 3.6.3

Результати експериментальної оцінки виконання формування і перевірки ЕЦП

Поле, *m*	**Час постановки ЕЦП, мс**			**Час перевірки ЕЦП, мс**		
	Побітовий метод	**Послівний метод**	**Виграш**	**Побітовий метод**	**Послівний метод**	**Виграш**
163	1.69	0.228	7.41	1.845	0.237	7.78
167	1.744	0.258	6.76	1.871	0.248	7.54
173	1.868	0.252	7.41	1.985	0.259	7.66
179	1.999	0.259	7.72	2.249	0.259	8.68

Продовження таблиця 3.6.3

191	2.32	0.267	8.69	2.479	0.285	8.70
233	3.516	0.58	6.06	3.752	0.591	6.35
257	4.534	0.688	6.59	4.962	0.693	7.16
307	6.497	0.938	6.93	7.023	0.912	7.70
367	9.547	1.366	6.99	10.36	1.457	7.11
431	13.6	1.967	6.91	14.406	2.088	6.90

Послівний метод дозволив підвищити швидкодію при формуванні ЕЦП в 6-8.7 разів, при перевірці ЕЦП в 7-8.7 разів.

Таблиця 3.6. 4

Продуктивність операції приведення за модулем для 1 банківського дня, як складової НС ЕЦП

Поле, *m*	**Постановка ЕЦП, хв**			**Перевірка ЕЦП, хв**		
	Побітовий метод	**Послівний метод**	**Різниця**	**Побітовий метод**	**Послівний метод**	**Різниця**
163	12,68	1,71	**10,97**	13,84	1,78	**12,06**
167	13,08	1,94	**11,15**	14,03	1,86	**12,17**
173	14,01	1,89	**12,12**	14,89	1,94	**12,95**
179	14,99	1,94	**13,05**	16,87	1,94	**14,93**
191	17,40	2,00	**15,40**	18,59	2,14	**16,46**
233	26,37	4,35	**22,02**	28,14	4,02	**24,12**
257	34,01	4,98	**29,03**	37,22	5,14	**32,07**
307	48,73	5,73	**43,00**	52,67	5,89	**46,79**
367	71,60	7,46	**64,14**	77,70	7,81	**69,89**
431	102,00	10,80	**91,20**	108,05	11,45	**96,59**

Як показують дані табл. 3.6.4 виграш при використанні послівного методу приведення за фіксованим модулем, складає при постановці 10-91 хв., при перевірці 12-97 хв.

Для статистичної оцінки реалізації обох методів (часу) мультиплікативного інвертування досліджувалося повторення 120 разів експериментів з 100 тис. операцій (див.табл. В.3, Додаток В) для полів з ДСТУ 4145-2002. Точність виміру часу становить 100 нс – для процесорної архітектури x86 Intel Core i7-6700 HQ. Значимість перевірялася за допомогою t-теста Стьюдента (див. табл. В.3, додаток В) і тест Манна-Уітні, використовуючи прогнозне аналітичне програмне забезпечення IBM SPSS.

Оскільки p-рівень критерію Лівіня ≤0,05 (див. табл. В.4, додаток В), то дисперсії порівнюваних розподілів значень статистично достовірно різняться, і тест Стьюдента застосовувати не можна. Тому проводився U-тест Манна-Уітні (див. табл.2.4.3). Критерій Манна-Уітні (асимптотична значимість <0,05) показав, що відмінності між вибірками є статистично значущими

3.7. Висновки до третього розділу

1. Запропоновані підходи до удосконалення алгоритму мультиплікативного інвертування в двійковому полі, дозволили зменшити обчислювальну складність МРАЕ в 2 рази, що підтверджується експериментальними оцінками.

2. Для процесора Intel Core i3-350M, МРАЕ показав виграш на 18-21% (ІСС) і 16-18% (МСС), в той же час МСС реалізація виявилася ефективнішою на 2,2% ніж на ІСС. Для процесора Intel Core i5-3570, МРАЕ показав виграш на 19-22% (ІСС) і 17-22% (МСС), в той же час ІСС реалізація виявилася ефективнішою на 4,9% ніж на МСС. Для процесора Intel Core i5-4670, МРАЕ показав виграш на 18-25% (ІСС) і 14-22% (МСС), в той же час ІСС реалізація виявилася ефективнішою до 14,7% ніж МСС.

3. Проведені статистичні тести, щодо часу реалізації обох методів мультиплікативного інвертування показали значущу різницю між замірами (вибірками часу).

4. Виграш при використанні удосконаленого методу мультиплікативного інвертування, складає при постановці 1-9 с, при перевірці 3-20 с для 1 робочого дня середнього банку, як складової Національної системи ЕЦП.

5. Запропонована реалізація МРАЕ інвертування не орієнтована на багатопоточне виконання, що не дозволило повністю реалізувати потенціал сучасних багатоядерних процесорів.

6. Розроблено метод автоматизації приведення довільного полінома за фіксованим модулем у двійковому полі (тричлена, п’ятичлена).

7. Побудовані алгоритми можуть бути ефективно реалізовані для різних цільових платформ, як для 8, 16, 32, так і 64-х розрядних.

8. Виграш при використанні послівного методу приведення за фіксованим модулем, складає при постановці 10-91 хв., при перевірці 12-97 хв. для 1 робочого дня середнього банку, як складової Національної системи ЕЦП.

Список використаних джерел у третьому розділі

[1] D. Hankerson, J. López Hernandez and A. Menezes, "Software Implementation of Elliptic Curve Cryptography over Binary Fields", *Cryptographic Hardware and Embedded Systems — CHES 2000*, pp. 1-24, 2000.

[2] "Bit Twiddling Hacks", *Graphics.stanford.edu,* 2018. [Online]. Available: http://graphics.stanford.edu/~seander/bithacks.html.

[3] М.Г. Булах, В.Ю. Ковтун, «Методы повышения производительности операции инвертирования в двоичном поле», *Безпека інформації*, том 20, № 1, с. 55-61, 2014.

[4] M. Scott. «Optimal Irreducible Polynomials for GF(2^m) Arithmetic». Cryptology ePrint Archive, Report 2007/192, 2007.

[5] J. Guajardo, S. Kumar, C. Paar and J. Pelzl, «Efficient Software-Implementation of Finite Fields with Applications to Cryptography», *Acta Applicandae Mathematicae*, vol. 93, no. 1-3, pp. 3-32, 2006.

[6] M. Kovtun, A. Okhrimenko, T. Gancarczyk, V. Karpinskyi and S. Gnatyuk, «Method of algorithm building for modular reducing by irreducible polynomial», *16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1476-1479, 2016., Gyeongju, Korea. DOI: 10.1109/ICCAS.2016.7832498.

[7] Національні стандарти України, «ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка», Київ, 2002.

[8] «FIPS 186-4, Digital Signature Standard (DSS) | CSRC», *Csrc.nist.gov*. [Online]. Available: https://csrc.nist.gov/publications/detail/fips/186/4/final.

РОЗДІЛ 4
РОЗРОБКА МЕТОДУ ВИЛУЧЕННЯ КОРЕНІВ В ДВІЙКОВОМУ ПОЛІ

4.1. Дослідження методів операцій здобуття кореня в двійковому полі

Для обчислення $\sqrt[r]{x}$, $x \in \boldsymbol{GF}(q^m)$ при $q=2$, m і r б відповідала умовам $(q(q-1),r)=1$ [1]. Ідея алгоритму полягає в пошуку деякого $v = r^{-1}(\bmod q^m - 1)$, що дозволяє звести обчислення $\sqrt[r]{x}$ до x^v в полі $\boldsymbol{GF}(2)$, а також скористатися перевагами розкладання v в адитивний ланцюг $v = a + b\sum_{j=0}^{n-1} q^{j\cdot k}$.

Лема 4.1 [1] визначає умови вилучення r-кореня в $\boldsymbol{GF}(q^m)$.

Лема 4.1. Задано q і r такі, що $(q(q-1),r)=1$, нехай $k>1$ - порядок q по модулю r. Для будь-якого $m>0$, $(m,k)=1$, нехай u, $1 \le u < r$, яке задовольняє умови $u(q^m-1) \equiv -1 (\bmod r)$ і $v = \lfloor q^m u / r \rfloor$. Враховуючи, що $r v \equiv 1 (\bmod q^m - 1)$, тоді v може бути представлено як $v = a + b\sum_{j=0}^{n-1} q^{j\cdot k}$, де $a,b < q^{2k}$, $n = \lfloor m/k \rfloor$.

Іншими словами, для обчислення $\sqrt[r]{x}$ в $\boldsymbol{GF}(2^m)$ необхідно обчислити x^v для $v^{-1} \equiv r (\bmod 2^m - 1)$, причому v, згідно алгоритму Іто-Цуйі, може бути представлено у вигляді адитивної ланцюга $v = a + b(1 + 2^2 + 2^4 + \ldots 2^{\lfloor m/k \rfloor})$.

Алгоритм 4.1. Здобуття кубічного кореня в полі $\boldsymbol{GF}(2^m)$.

Вхід: $\alpha \in \boldsymbol{GF}(2^m)$.

Вихід: $\beta \leftarrow \sqrt[3]{\alpha}$, $\beta \in \boldsymbol{GF}(2^m)$.

1. $\beta \leftarrow b$, $t_0 \leftarrow (b^2)^2$.
2. For $i \leftarrow 0$, $i \le ((m-1)/2)$, $i{+}{+}$ do
2.1. $t_i \leftarrow (t_{i-1}^2)^2$.
2.2. $\beta \leftarrow \beta \cdot t_i$.
3. Return β.

Рис.4.1. 1. Псевдокод операції здобуття кубічного кореня

Як приклад, розглянемо здобуття кубічного кореня. Згідно леми 4.1, для $q = 2$ і непарного m, показник степеня $1/r$ можна представити як $\frac{1}{3} = \sum_{j=0}^{(m-1)/2} 2^{2 \cdot j} \left(\mathrm{mod}\, 2^m - 1\right)$. Отже кубічний корінь може бути представлений наступним чином: $\sqrt[3]{b} = b^{1+2^2+2^4+\ldots+2^{m-1}}$.

На основі, описаного адитивного ланцюга, наведено алгоритм 4.1. здобуття кубічного кореня [1] (рис.4.1.1).

4.2. Розробка удосконаленого методу операції здобуття кубічного кореня

Як видно з алгоритму 4.1, найбільшу трудомісткість являє цикл п. 2.2, оскільки виконується ітераційне множення. У зв'язку з цим, становить інтерес до пошуку розкладання на множники ланцюга $1/3 = 1 + 2^2 + 2^4 + \ldots + 2^{m-1}$, що дозволить зменшити кількість операцій множення в циклі п. 2.2, використовуючи схему Горнера.

Проведений аналіз публікацій, показав існування в роботі [2] ітераційне розкладання адитивного ланцюга на множники:

$$1 + 2^n + 2^{2n} + \ldots + 2^{(k-2)n} = \begin{cases} \left(1 + 2^n\right) \times \left(1 + 2^{2n} + \ldots + 2^{(k-3)n}\right), & \text{якщо } k - 1 \equiv 0 \bmod 2 \\ 1 + 2^n \left(1 + 2^n\right) \times \left(1 + 2^{2n} + \ldots + 2^{(k-4)n}\right), & \text{якщо } k - 1 \equiv 1 \bmod 2 \end{cases} \quad (4.2.1)$$

при початковому значенні $n = 1$ і $(k-2)n$ - непарне.

Пропонується адаптувати відоме розкладання (4.2.1) до специфіки v [3]: де початкове $n = 2$ і $(k-2)n$ - парне.

Таблиця 4.2. 1

Розкладання адитивної ланцюга на множники для $\boldsymbol{GF}\left(2^{163}\right)$

Покрокове розкладання	Параметри	Умова
$1 + 2^2 + 2^4 + \ldots + 2^{162}$	$k = 83, n = 2$	$k - 1 \equiv 0 \bmod 2$
$\left(1 + 2^2\right)\left(1 + 2^4 + \ldots + 2^{(83-3)\cdot 2}\right)$	$k = 42,\ n = 4$	$k - 1 \equiv 1 \bmod 2$
$\ldots\left(1 + 2^4\left(1 + 2^4\right)\left(1 + 2^8 + \ldots + 2^{(42-4)\cdot 4}\right)\right)$	$k = 21,\ n = 8$	$k - 1 \equiv 0 \bmod 2$
$\ldots\left(1 + 2^8\right)\left(1 + 2^{16} + \ldots + 2^{(21-3)\cdot 8}\right)$	$k = 11,\ n = 16$	$k - 1 \equiv 0 \bmod 2$

Продовження таблиці 4.2.1

$\ldots\left(1+2^{16}\right)\left(1+2^{32}+\ldots 2^{(11-3)16}\right)$	$k=6,\ n=32$	$k-1\equiv 1 \bmod 2$
$\ldots\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\right)$		

Для більшої наочності, в таблиці 4.2.1 наведено приклад розкладання на множники v для поля $\boldsymbol{GF}\left(2^{163}\right)$ [3].

Таким чином, розкладання на множники ланцюга для $\boldsymbol{GF}\left(2^{163}\right)$ може бути представлене в такому вигляді: $1+2^2+2^4+\ldots+2^{162}=$ $\left(1+2^2\right)\left(1+2^4\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\right)\right)$.

У табл. 4.2.2 [3] представленні розклади показника степеню в адитивний ланцюг в залежності від характеристики полів $\boldsymbol{GF}\left(2^m\right)$, представлених у ДСТУ 4145-2002.

Таблиця 4.2.2

Декомпозиція ланцюга для $\boldsymbol{GF}\left(2^m\right)$, m - непарне

Поле	**Розкладання на множники**
$\boldsymbol{GF}\left(2^{163}\right)$	$\left(1+2^2\right)\left(1+2^4\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\right)\right)$
$\boldsymbol{GF}\left(2^{167}\right)$	$\left(1+2^2\right)\left(1+2^4\right)\left(1+2^8\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\right)\right)$
$\boldsymbol{GF}\left(2^{173}\right)$	$1+2^2\left(1+2^2\right)\left(1+2^4\left(1+2^4\right)\left(1+2^8\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\right)\right)\right)$
$\boldsymbol{GF}\left(2^{179}\right)$	$\left(1+2^2\right)\left(1+2^4\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\right)\left(1+2^{64}\right)\right)\right)$
$\boldsymbol{GF}\left(2^{191}\right)$	$\left(1+2^2\right)\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\right)\left(1+2^{64}\left(1+2^{64}\right)\right)$
$\boldsymbol{GF}\left(2^{233}\right)$	$1+2^2\left(1+2^2\right)\left(1+2^4\right)\left(1+2^8\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\left(1+2^{64}\right)\right)\right)\right)$
$\boldsymbol{GF}\left(2^{257}\right)$	$1+2^2\left(1+2^2\right)\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\right)\left(1+2^{32}\right)\left(1+2^{64}\right)\left(1+2^{128}\right)$
$\boldsymbol{GF}\left(2^{307}\right)$	$\left(1+2^2\right)\left(1+2^4\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\left(1+2^{128}\right)\right)\right)\right)$
$\boldsymbol{GF}\left(2^{367}\right)$	$\left(1+2^2\right)\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\left(1+2^{64}\right)\left(1+2^{128}\right)\right)\right)\right)$
$\boldsymbol{GF}\left(2^{431}\right)$	$\left(1+2^2\right)\left(1+2^4\right)\left(1+2^8\right)\left(1+2^{16}\left(1+2^{16}\right)\left(1+2^{32}\left(1+2^{32}\right)\left(1+2^{64}\right)\left(1+2^{128}\left(1+2^{128}\right)\right)\right)\right)$

Здобуття кубічного кореня в полі $\boldsymbol{GF}\left(2^{163}\right)$, наведено у вигляді алгоритму 4.2 (рис.4.2.1), решта алгоритмів отримуються за аналогією (див. Додаток Б).

Алгоритм 4.2. Здобуття кубічного кореня в полі $GF(2^{163})$.

Вхід: $b(x) \in GF(2^{163})$.

Вихід: $\sqrt[3]{b(x)}$

1. $t_0 \leftarrow b$, $t_0 \leftarrow (t_0^2)^2 \cdot t_0$, $t_1 \leftarrow t_0$.

2. $t_0 \leftarrow (t_0^2)^4$, $t_0 \leftarrow (t_0^2)^4 \cdot t_0$, $t_0 \leftarrow (t_0^2)^8 \cdot t_0$, $t_0 \leftarrow (t_0^2)^{16} \cdot t_0$.

3. $t_2 \leftarrow t_0$, $t_0 \leftarrow (t_0^2)^{32}$, $t_0 \leftarrow (t_0^2)^{32} \cdot t_0$, $t_0 \leftarrow (t_0^2)^{64} \cdot t_0$, $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2$.

4. Return t_0.

Рис.4.2. 1. Псевдокод удосконаленого методу здобуття кубічного кореня для $GF(2^{163})$

Удосконалений метод здобуття кубічного кореня використовується при відшуканні біраціонально еквівалентних повних кривих Едвардса до кривих Вейерштрасса у двійковому полі (див. Розділ 5) [4-7].

4.3. Розробка удосконаленого методу здобуття r-вимірного кореня

Для здобуття кореня 9-го степеня, потрібно всього лише двічі здобути кубічний корінь (виключний випадок).

Для загальних випадків, коли m - непарне, r - корінь, з інвертованим показником (див. Лему 4.1) нижче представлена схема здобуття:

– Обчислити порядок (кратність) k: $2^k \equiv 1 (\mathrm{mod}\, r)$.

– Обчислити верхню межу суми $n-1$: $n = \lfloor m/k \rfloor$.

– Представити суму у вигляді адитивного ланцюга: $1 + (2^h)^2 + (2^h)^4 + \ldots + (2^h)^s$, де $h = k/2$, $s = 2(n-1)$, $n = \lfloor m/k \rfloor$.

– Представити (розкласти на множники) адитивний ланцюг в декомпозицію за формулою (4.2.1), де початкове $n = 2$ і $(k-2)n$ - парне.

4.4. Оцінка обчислювальної складності та результати експериментальних оцінок швидкодії

Експериментальна оцінка швидкодії здобуття кубічного кореня та обчислювальна складність для різних показників розширення m полів $GF(2^m)$,

використовуючи процесор Intel Core i7-2600 під управлінням ОС Windows 7 SP1 x86-64, приведена у табл 4.4.1. Програмна реалізація виконана за допомогою Visual C++ 2015. Число випробувань складає 1 млн. операцій. Кубічний корінь здобувався з параметра b еліптичної кривої заданої над відповідним полем, представленого в ДСТУ 4145-2002.

Для порівняння обчислювальної складності, приймається відношення операцій: 1**S**=0.1**M**.

Таблиця 4.4. 1

Порівняння обчислювальної складності та швидкодії

m	**Відомий метод**		**Удосконалений метод**		**Виграш**	
	Кількість операцій	**Час реалізації, мс**	**Кількість операцій**	**Час реалізації, мс**	**Складність**	**Швидко-дія**
163	81**M**+162**S**	0.0233	8**M**+162**S**	0.0093	4.02	2.51
167	83**M**+166**S**	0.024	8**M**+166**S**	0.0095	4.05	2.53
173	86**M**+172**S**	0.0261	10**M**+172**S**	0.0104	3.79	2.51
179	89**M**+178**S**	0.0277	9**M**+178**S**	0.0114	3.99	2.43
191	95**M**+190**S**	0.0292	7**M**+190**S**	0.0099	4.38	2.95
233	116**M**+232**S**	0.0481	10**M**+232**S**	0.0173	4.19	2.78
257	128**M**+256**S**	0.0629	8**M**+256**S**	0.0201	4.57	3.13
307	153**M**+306**S**	0.086	10**M**+306**S**	0.0284	4.52	3.03
367	183**M**+366**S**	0.1315	10**M**+366**S**	0.0361	4.71	3.64
431	215**M**+430**S**	0.1953	10**M**+430**S**	0.0523	4.87	3.73

За результатами експериментальних оцінок видно, що теоретична оцінка виграшу у складності складає 4-5 разів та виграш у швидкодії складає 2.4-3.7 рази.

Проводилася статистична оцінка реалізації удосконаленого методу та прототипу здобуття кубічного кореня для полів представлених в ДСТУ 4145-2002.

Досліджувалося 120 повторів експериментів з 100 тис. операцій (див.табл. В.3, додаток В) для полів з ДСТУ 4145-2002. Точність виміру часу становить 100 нс – для процесорної архітектури x86 Intel Core i7-6700 HQ. Значимість перевірялася за допомогою t-теста Стьюдента (див. табл. В.3, Додаток В) з 99,5%

довірчим інтервалом та критерія Мана-Уітні (див. табл. 3.5.2), використовуючи систему для аналізу даних IBM SPSS.

Оскільки p-рівень критерію Лівіня ≤0.05 (див. табл. В.4, Додаток В): дисперсії порівнюваних розподілів значень статистично достовірно різняться, надалі застосовувався критерій для незалежних вибірок Манна-Уітні, який показав значення асимптотичної значимості <0.05 - порівнювані значення з двох вибірок дійсно статистично достовірно різняться.

Таблиця 4.4. 2

Порівняння обчислювальної складності та швидкодії

Поле	**431**	**367**	**307**	**257**	**233**	**191**	**179**	**173**	**167**	**163**
U Манна-Уітні	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
W Вілкоксона	7260	7260	7260	7260	7260	7260	7260	7260	7260	7260
Z	-13,40	-13,404	-13,407	-13,403	-13,405	-13,402	-13,403	-13,404	-13,402	-13,403
Асимп. значм. (2-сторонн)	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**	**,000**

4.5. Висновки до четвертого розділу

1. Виграш обчислювальної складності удосконаленого алгоритму (на прикладі, здобуття кубічного кореня) відносно відомого, складає 4-5 разів в залежності від характеристики заданого поля.

2. Виграш швидкодії удосконаленого алгоритму (на прикладі, здобуття кубічного кореня) показує виграш в 2.4-3.7 разів залежно від характеристики заданого поля.

3. Проведені статистичні тести, щодо часу реалізації обох методів здобуття кубічного кореня у двійковому полі показали, що удосконалений метод є більш швидкодіючий, на відміну від прототипу.

Список використаних джерел у четвертому розділі

[1] P. Barreto and J. Voloch, «Efficient Computation of Roots in Finite Fields», *Designs, Codes and Cryptography*, vol. 39, no. 2, pp. 275-280, 2006.

[2] M. Bluhm, «Software optimization of binary elliptic curves arithmetic using modern processor architectures», Master's Thesis, Ruhr-Universität-Bochum, 2014.

[3] М.Г. Ковтун, С.А. Гнатюк, В.И. Трофименко. «Ускоренное извлечение r -го корня в двоичном поле» в *Докл. Межд. науч.–практ. Конф. Информационные и телекоммуникационные технологии: образование, наука, практика*, Алматы, Казахстан , Декабрь, 2-4, 2015, с. 547-551.

[4] D. Bernstein, T. Lange and R. Rezaeian Farashahi, «Binary Edwards Curves», *Cryptographic Hardware and Embedded Systems – CHES 2008*, pp. 244-265.

[5] M. Li, A. Miri and D. Zhu, «Fast Algorithm for Converting Ordinary Elliptic Curves into Binary Edward Form», *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 1, pp. 405-412, 2012.

[6] M. Kovtun, Z.B. Hu, S. Gnatyuk, N. Seilova. «Method of Searching Birationally Equivalent Edwards Curves over Binary Fields», *in Proc. of the 1st International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2018)*, Jan 18-20, 2018, Kiev, Ukraine.

[7] M.G. Kovtun, V.Y. Kovtun, A.A. Okrimenko and S.A. Gnatyuk. «Search method development of birationally equivalent binary Edwards curves for binary Weierstrass curves from DSTU 4145-2002», in *Proc. PIC S&T*, Kharkov, Ukraine, Oct. 13-15, 2015. pp. 5-8. DOI: 10.1109/INFOCOMMST.2015.7357253.

РОЗДІЛ 5
РОЗРОБКА МЕТОДУ ПІДВИЩЕННЯ ШВИДКОДІЇ ТА ЗАХИЩЕНОСТІ КРИПТОГРАФІЧНИХ ПЕРЕТВОРЕНЬ НА ЕЛІПТИЧНИХ КРИВИХ У ДВІЙКОВОМУ ПОЛІ

5.1. Удосконалений метод скалярного множення точок еліптичної кривої з проміжними обчисленнями на кривій Едвардса

Пропонується при СМ для формування і перевірки ЕЦП на основі базової точки згідно ДСТУ 4145-2002 [1], замість кривих Вейерштрасса в двійковому полі використовувати біраціонально еквівалентні повні криві (БЕК) Едвардса. Це можна записати у наступному вигляді:

1. Обчислення перетворень:

– Пошук біраціонально еквівалентної повної кривої Едвардса (1.3.2) та (1.3.6).

– Перетворення базової точки P на кривій Вейерштрасса, в біраціонально еквівалентну точку P' на кривій Едвардса (1.3.4) та (1.3.8).

2. Операція СМ:

– $Q' = k \cdot P'$ з використанням алгоритму Монтгомері (w-координатне диференціальне складання та подвоєння точок) [2-5].

– Перетворення результуючої точки Q' на кривій Едвардса, в точку Q на кривій Вейерштрасса (1.3.5) та (1.3.7).

3. Return Q.

5.1.1. Пошук біраціонально еквівалентних повних кривих Едвардса до кривих Вейерштрасса у двійковому полі

Аналіз відомих методів пошуку біраціонально еквівалентної кривої та базової точки при умові $d_1 \neq d_2$ [4-5] показав, що найбільш ефективним є метод Li-Miri-Zhu, який використовує здобуття кубічного кореня.

Алгоритм Li-Miri-Zhu можна значно прискорити, використовуючи передобчислювання, а також при використанні операції здобуття кубічного кореня скористатися алгоритмами розкладання показника степеня на множники

(див. Розділ 4). Дані пропозиції, представлені в алгоритмі 5.1. Удосконалений метод пошуку біраціонально еквівалентних кривих наведено в роботах [6-7], а знайдені відповідні криві при умовах $d_1 \neq d_2$ та $d_1 = d_2$ наведені в Додатку А.

Алгоритм 5.1. (Удосконалений) Пошуку біраціонально еквівалентної кривої Едвардса з використанням кубічного кореня.

Вхід: $a, b \in \boldsymbol{GF}(2^m)$, $X_W = (u, v) \in \boldsymbol{GF}(2^m)$.

Вихід: $d_1, d_2 \in \boldsymbol{GF}(2^m)$, $X_E = (x, y) \in \boldsymbol{GF}(2^m)$.

передобчислювання

1. $SR_b \leftarrow \sqrt{b}$, $\gamma \leftarrow 1/b$.

Основний алгоритм.

2. $a_2 \leftarrow a$, $\lambda \leftarrow 0$, $U \leftarrow a_2^6 \cdot \gamma$.

3. While $(\mathrm{Tr}(U) \neq 0)$ do

3.1. Random (λ), $\lambda \in \boldsymbol{GF}(2^m)$.

3.2. $a_2 \leftarrow a_2 + \lambda^2 + \lambda$, $U \leftarrow a_2^6 \cdot \gamma$.

3.3. Вирішення квадратного рівняння, відносно r: $r^2 + r + U = 0$.

Обчислення коефіцієнтів кривої Едвардса:

4. $g_1 \leftarrow SR_b \cdot r$, $g_2 \leftarrow SR_b \cdot (r+1)$, $SR_1 \leftarrow \sqrt[3]{g_1}$, $SR_2 \leftarrow \sqrt[3]{g_2}$, $d_1 \leftarrow SR_1 + SR_2 + a_2$, $\delta \leftarrow d_1^2$, $d_2 \leftarrow \delta + a_2$.

Обчислення біраціональних еквівалентної точки Едвардса, якщо задана точка на кривій Вейерштрасса:

5. $\varphi \leftarrow \delta + d_1$, $C_1 \leftarrow \varphi + d_2$, $C_2 \leftarrow (C_1 + u) \cdot d_1$, $C_3 \leftarrow \varphi \cdot C_1 + v$, $x \leftarrow C_2/(u + C_3)$, $y \leftarrow C_2/C_3$.

6. if $\lambda > 0$ then

6.1. Вирішення квадратного рівняння відносно g: $g^2 + g + a + a_2 = 0$.

6.2. $y \leftarrow y + g \cdot x$.

7. Return $(d_1, d_2, \{x, y\})$.

Рис.5.1. 1. Псевдокод пошуку біраціонально еквівалентної кривої та базової точки Едвардса

Нижче представлена арифметика на двійкових кривих Едвардса, використовуючи змішане w-координатне диференціальне складання та подвоєння [2, 5, 8]. Під диференціальним складанням мається на увазі обчислення $Q + P$ з урахуванням Q, P, $Q - P$. Наприклад, обчислення $(2m+1)P$, за умови $(m+1)P$, mP, P або обчислення $2mP$, за умови mP, P, $0P$. Зокрема,

«w-координатне диференціальне складання» означає обчислення $w(Q+P)$ при $w(Q), w(P),\ w(Q-P)$.

$W_i = X_i + Y_i\,(1 \le i \le 4), 2(X_1 : Y_1 : Z_1) = (X_4 : Y_4 : Z_4)$,

$(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$ знаючи w_0, де

$(x_0 : y_0 : 1) = (X_2 : Y_2 : Z_2) - (X_1 : Y_1 : Z_1)$. Дана формула використовує 6M+D+ 5S або 5M+3D+4S при $d_1 \neq d_2$ [5, 10].

$$C = W_1 \cdot (W_1 + Z_1), \quad D = W_2 \cdot (W_2 + Z_2), \quad E = (W_1 + W_2)(W_1 + W_2 + Z_1 + Z_2) + C + D,$$

$$V = C \cdot D, \quad W_3 = d_1 \cdot E^2, \quad Z_3 = V + \frac{1}{w_0} \cdot W_3, \quad W_4 = C^2,$$

$$Z_4 = W_4 + \left(\sqrt[4]{d_1} \cdot Z_1 + \sqrt[4]{d_2/d_1 + 1} \cdot W_1\right)^4.$$

для випадку $d_1 = d_2$ - 5M + D + 4S [5].

$$C = W_1 \cdot (W_1 + Z_1), \quad D = W_2 \cdot (W_2 + Z_2), \quad E = (W_1 + W_2)(W_1 + W_2 + Z_1 + Z_2) + C + D,$$

$$V = C \cdot D, \quad W_3 = d_1 \cdot E^2, \quad Z_3 = V + \frac{1}{w_0} \cdot W_3, \quad Z_3 = V + \frac{1}{w_0} \cdot W_3, \quad W_4 = C^2,$$

$$Z_4 = W_4 + d_1 \cdot Z_1^4.$$

Алгоритм 5.2 – алгоритм Монтгомері для СМ на кривих Едвардса, використовуючи ω-координати при додаванні та подвоєнні точок. Де $\mathrm{DiffDBL}(\cdot,\cdot)$ – диференціальне подвоєння, а $\mathrm{MDiffDBL}(\cdot,\cdot,\cdot,\cdot,\cdot)$ – диференціальне складання [11].

Після реалізації СМ $w_2 = w(kP)$ і $w_3 = w((k+1)P)$, отримані результати потрібно представити в явному вигляді $Q = (x_2, y_2) = kP$. Для цього необхідно скористатися формулою (5.1.1) [2]:

$$x_2^2 + x_2 = \frac{w_3\left(d_1 + w_0 w_2 (1 + w_0 + w_2) + {}^{d_2}\!/_{d_1}\, w_0^2 w_2^2\right) + d_1(w_0 + w_2) + \left(y_1^2 + y_1\right)\left(w_2^2 + w_2\right)}{w_0^2 + w_0} \quad (5.1.1)$$

Вирішивши квадратне рівняння (5.1.1), знаходиться x_2. Для відшукання y_2 при умові, що $d_1 \neq d_2$, потрібно підставити значення x_2 в рівняння кривої Едвардса (1.3.2), після чого отримаємо вираз:

$$y_2^2 + y_2 \cdot \left(\frac{d_1 + x_2 + x_2^2}{d_2 + x_2 + x_2^2}\right) + \frac{d_1 \cdot x_2 + d_2 \cdot x_2^2}{d_2 + x_2 + x_2^2} = 0.$$

Алгоритм 5.2. Алгоритм Монтгомері для СМ використовуючи w-координати.

Вхід: $P=(x_0,y_0)\in \boldsymbol{GF}(2^m)$, $k=(k_{l-1},\cdots,k_1,k_0)_2$.

Вихід: $w(Q)=w(kP)\in \boldsymbol{GF}(2^m)$.

1. $w_0 \leftarrow x_0 + y_0$

1.1. $W_1 \leftarrow w_0$, $Z_1 \leftarrow 1$.

1.2. $(W_2,Z_2)=\text{DiffDBL}(W_1,Z_1)$.

2. for $l-2$ down to 0 do

2.1. if $k_i=1$ then

2.1.1. $(W_1,Z_1)=\text{MDiffDBL}(W_1,Z_1,W_2,Z_2,w_0)$.

2.1.2. $(W_2,Z_2)=\text{DiffDBL}(W_2,Z_2)$.

2.2. else

2.2.1. $(W_1,Z_1)=\text{DiffDBL}(W_1,Z_1)$.

2.2.2. $(W_2,Z_2)=\text{MDiffDBL}(W_1,Z_1,W_2,Z_2,w_0)$.

3. Return $w(kP)\leftarrow(W_1,Z_1)$ і $w((k+1)P)\leftarrow(W_2,Z_2)$.

Рис.5.1. 2. Псевдокод СМ Монтгомері, використовуючи w-координати

Зробивши заміну $z=y_2\cdot\left(\frac{d_2+x_2+x_2^2}{d_1+x_2+x_2^2}\right)$, вирішуємо квадратне рівняння (5.1.2) відносно z:

$$z^2+z=\frac{\left(d_1\cdot x_2+d_2\cdot x_2^2\right)\left(d_2+x_2+x_2^2\right)}{\left(d_1+x_2+x_2^2\right)^2} \qquad (5.1.2)$$

Після знаходження z, результуючий $y_2=z\cdot\frac{\left(d_1+x_2+x_2^2\right)}{\left(d_2+x_2+x_2^2\right)}$.

Для випадку кривої Едвардса $d_1=d_2$ (1.3.6), результуючий y_2 має вигляд:

$$y_2^2+y_2=\frac{d_1\left(x_2+x_2^2\right)}{d_1+x_2+x_2^2}.$$

5.2. Оцінка обчислювальної складності

В табл. 5.2.1 для порівняння, показана обчислювальна складність арифметичних операцій додавання і подвоєння на двійковій кривій Вейерштрасса і БЕК Едвардса при різних початкових умовах [9-10].

Таблиця 5.2.1

Обчислювальна складність арифметичних операцій на кривих Вейерштрасса і Едвардса

Метод	**Складність кривої Вейерштрасса**	**Складність для БЕК**	
		$d_1 \neq d_2$	$d_1 = d_2$
Монтгомері (проектні w-координати)		5M+3D+4S	5M+D+4S
Монтгомері	6M+5S		

5.3. Результати експериментальних оцінок швидкодії розробленого методу

Для порівняння швидкодії пошуку біраціонально еквівалентних кривих та базових точок Едвардса до кривих Вейерштрасса для ДСТУ 4145-2002 була виконана реалізація за допомогою Visual C++2015 з урахуванням приведення за фіксованим модулем і удосконаленого методу інвертування на основі розширеного алгоритму Евкліда. Заміри часу проводилися для 1 тис. операцій, за допомогою обчислювальної системи з процесорами Intel Core i7-6700 2,60 GHz (Microsoft Windows 10 x86-64). Удосконалений метод пошуку біраціонально еквівалентних кривих і точок показав виграш у швидкодії в 1.3-1.8 разів.

Таблиця 5.3. 1

Час пошуку біраціонально еквівалентної кривої і точки Едвардса до кривої Вейерштрасса для ДСТУ 4145-2002

m	**Час реалізації, мс**		**Виграш**	m	**Час реалізації, мс**		**Виграш**
	LMZ	**LMZ***			**LMZ**	**LMZ***	
163	0.1202	0.0805	1.49	233	0.2741	0.2054	1.33
167	0.1507	0.1054	1.43	257	0.2676	0.1854	1.44
173	0.1252	0.0872	1.44	307	0.3883	0.2408	1.61
179	0.1412	0.0967	1.46	367	0.6256	0.3807	1.64
191	0.1739	0.123	1.41	431	0.8382	0.4762	1.76

- LMZ* - метод Li-Miri-Zhu з використанням удосконаленого методу здобуття кубічного кореня [6-7].

Для порівняння швидкодії СМ була виконана реалізація алгоритму СМ методом Монтгомері за допомогою Visual C++2015 та gcc 5.4.0 з урахуванням приведення за фіксованим модулем і удосконаленого методу інвертування на основі розширеного алгоритму Евкліда. Заміри часу проводилися для 10 тис. операцій, за допомогою обчислювальних системи з процесорами Intel Core i7-6700 2,60 GHz (Microsoft Windows 10 x86-64), Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2), Intel Core i7-4702MQ 2,2GHz (Microsoft Windows 8.1 Pro x86-64), Intel Core i5-4670 (Microsoft Windows 7 x86-64) 3,40 GHz (AVX2 version), Intel Xeon E3-1270v5 3.6GHz (Microsoft Windows Server 2012 R2 x86-64) (див.табл 5.3.3), а також Intel Xeon E5-2695v3 2.3 GHz (CentOS Linux v7.0 x86-64) (див.табл.5.3.4), Intel Xeon E5-2640 2.5 GHz (CentOS Linux v7.0 x86-64), Intel Xeon X5670 2.93 GHz (CentOS Linux v7.0 x86-64), Intel Core i7-4702MQ 3.20 GHz (CentOS Linux Ubuntu 16.04) (див. Додаток Д).

Таблиця 5.3. 2

Порівняння швидкодії реалізації операції СМ в проективних координатах для ДСТУ 4145-2002

m	**Час реалізації СМ, мс**					
	Біраціонально еквівалентна крива Вейерштрасса до БЕК 251	**Криві Вейерштрасса з ДСТУ4145-2002**		**Біраціональна крива Едвардса для кривих з ДСТУ4145-2002**		**БЕК 251**
				$d_1 \neq d_2$	$d_1 = d_2$	$d_1 = d_2$
	Алгоритм Монтгомері (проективні координати Лопеса-Дахаба)		**Бінарний алгоритм (змішані координати Лопеса-Дахаба)**	**Алгоритм Монтгомері (проективні w-координати)**		
163		0.2065	0.2228	0.3279		
167		0.209	0.3796	0.3335		
173		0.2293	0.2414	0.3733	0.2452	
173***		0.228	0.2472	0.3705	0.2436	
173#		0.2751	0.2906	0.4417	0.3007	
179		0.2426	0.4177	0.3892		
191		0.2572	0.2816	0.4363		
233		0.4255	00.4487	0.6651		
251	0.5496					0.4903
251*						0.4600
251**						0.5986
251***						0.6143
257		0.6927	1.1483	1.0424	0.6485	

Продовження таблиці 5.3.2

257*		0.6488	1.076	0.9771	0,613505	
257**		0.8117	1.3829	1.2552	0,759427	
257***		0.8115	1.3825	1.2589	0,775469	
307		0.7742	1.3205	1.2023		
367		1.1507	1.1668	1.7775		
431		1.7486	2.8938	2.6802		

* - Intel Xeon E3-1270v5 3,60 GHz, (Microsoft Windows Server 2012 R2);
** - Intel Core i7-4702MQ 2,2 GHz, (Microsoft Windows 8.1 Pro x64);
*** - Intel Core i5-3570 3,40 GHz, (AVX version) (Microsoft Windows 7 x86-64);
# - Intel Core i5-4670 3,40 GHz, (AVX2 version). (Microsoft Windows 7 x86-64).

Результати вимірів швидкодії реалізації операції СМ виконувались для двійкових кривих Вейерштрасса і відповідних біраціонально еквівалентних кривих Едвардса в проективних w-координатах.

Крім біраціонально еквівалентних кривих Едвардса до кривих Вейерштрасса з ДСТУ 4145-2002, експериментальні оцінки проводилися і для кривої БЕК251 [12] –крива запропонована Бернштейном при умові $d_1 = d_2$, де розмір d_1 не перевищує 64 біта. Дана крива відповідає всім криптографічним властивостям та рівню стійкості відповідним для NIST.

Таблиця 5.3. 3

Порівняння швидкодії реалізації операції СМ в проективних координатах для ДСТУ 4145-2002 (Intel Xeon E3-1270v5 3.6GHz , Windows) використовуючи спеціалізований набір інструкцій

Поле, m	**Час реалізації СМ, мс**				
	Біраціонально еквівалентна крива Вейерштрасса до БЕК 251	**Криві Вейерштрасса з ДСТУ4145-2002**	**Біраціональна крива Едвардса для кривих з ДСТУ4145-2002**		**БЕК 251**
			$d_1 \neq d_2$	$d_1 = d_2$	$d_1 = d_2$
	Алгоритм Монтгомері (проективні координати Лопеса-Дахаба)		**Алгоритм Монтгомері (проективні w-координати)**		
163		0.183	0.354		
167		0.186	0.353		
173		0.203	0.378	0.214	
179		0.211	0.400		
191		0.226	0.427		
233		0.399	0.759		
251	0.527				0.489
257		0.531	0.981	0.505	
307		0.719	1.323		

Продовження таблиці 5.3.3

367		1.062	1.984		
431		1.602	2.943		

Таблиця 5.3. 4

Порівняння швидкодії реалізації операції СМ в проективних координатах для ДСТУ 4145-2002 (Intel Xeon E5-2695v3 2.3 GHz , CentOS Linux v7.0 x86-64)

Поле, m	**Час реалізації СМ, мс**				
	Біраціонально еквівалентна крива Вейерштрасса до ВЕС 251	**Криві Вейерштрасса з ДСТУ4145-2002**	**Біраціональна крива Едвардса для кривих з ДСТУ4145-2002**		**ВЕС 251**
			$d_1 \neq d_2$	$d_1 = d_2$	$d_1 = d_2$
	Алгоритм Монтгомері (проективні координати Лопеса-Дахаба)		**Алгоритм Монтгомері (проективні w-координати)**		
163		0.197	0.401		
163#		0.254	0.360		
167		0.188	0.397		
167#		0.177	0.394		
173		0.209	0.433	0.282	
173#		0.287	0.397	0.282	
179		0.214	0.443		
179#		0.214	0.670		
191		0.219	0.439		
191#		0.215	0.455		
233		0.368	0.679		
233#		0.361	0.668		
251	0.519				0.466
251#	0.512				0.472
257		0.525	1.325	0.495	
257#		0.518	0.980	0.487	
307		0.723	1.572		
307#		0.701	1.341		
367		1.285	2.394		
367#		1.071	2.013		
431		1.809	3.170		
431#		1.588	3.072		

- # - спеціалізований набір інструкцій процесора.

Результати показують, що швидкодія операції СМ на кривих Вейерштрасса переважає над біраціонально еквівалентними кривими Едвардса з двома параметрами. Однак за умови $d_1 = d_2$, швидкодія реалізації СМ для кривих Едвардса зростає, починаючи з поля, розміром 257 біт:

– для процесора Intel Core i7-4702MQ 2,2 GHz, (Microsoft Windows 8.1 Pro x64) та Intel Core i7-6700 2,60 GHz (Microsoft Windows 10 x86-64) виграш в швидкодії склав 7%;

– для процесора Intel Core i5-4670 3,40 GHz, (AVX2). (Microsoft Windows 7 x86-64) та Intel Xeon E3-1270v5 3,60 GHz, (Microsoft Windows Server 2012 R2) виграш склав 5% та 6% відповідно;

– для процесора Intel Xeon E3-1270v5 3.6GHz (Microsoft Windows Server 2012 R2 x86-64) та Intel Xeon E5-2695v3 2.3 GHz (CentOS Linux v7.0 x86-64) виграш в швидкодії склав 5% і 6%, відповідно.

Для кривої БЕК251, швидкодія реалізації СМ склала:

– для процесора Intel Core i7-4702MQ 2,2 GHz, (Microsoft Windows 8.1 Pro x64) - 12%;

– для процесора Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2) - 7%;

Intel Xeon E5-2695v3 2.3 GHz (CentOS Linux v7.0 x86-64) - 11%.

Таблиця 5.3. 5

Час перетворення результату, отриманої точки, на кривій Едвардса в точку на кривій Вейерштрасса

Поле, m	**Час перетворення, мс**		**Поле,** m	**Час перетворення, мс**	
	$d_1 \neq d_2$	$d_1 = d_2$		$d_1 \neq d_2$	$d_1 = d_2$
163	0.0045		233	0.0067	
167	0.0048		257	0.0087	0.0083
173	0.0049	0.0047	307	0.0096	
179	0.005		367	0.0124	
191	0.0053		431	0.0164	

Для порівняння швидкодії формування та перевірки ЕЦП була виконана реалізація Visual C++2015 та gcc 5.4.0 з урахуванням приведення за фіксованим модулем і удосконаленого методу інвертування на основі розширеного алгоритму Евкліда. Заміри часу проводилися для 10 тис. операцій, за допомогою обчислювальних системи з процесорами Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2 x86-64), а також Intel Xeon E5-2695v3 2,3 GHz (CentOS Linux v7.0 x86-64) (див.табл. 5.3.6), Intel Xeon E5-2640 2,50 GHz (CentOS

Linux v7.0 x86-64), Intel Xeon X5670 2,93 GHz (CentOS Linux v7.0 x86-64), Intel Core i7-4702MQ 3,20 GHz (CentOS Linux Ubuntu 16.04) (див. Додаток Д).

Таблиця 5.3. 6

Час реалізації формування і перевірки ЕЦП згідно ДСТУ 4145-2002

Поле, m	**Час, мс**							
	Формування ЕЦП				**Перевірка ЕЦП**			
	W	**E** $d_1 \neq d_2$	**E** $d_1 = d_2$	**Виграш**	**W**	**E** $d_1 \neq d_2$	**E** $d_1 = d_2$	**Виграш**
163*	0,201	0,446			0,467	0,984		
163**	0,192	0,426			0,443	1,016		
163#	0,186	0,355			0,195	0,368		
167*	0,187	0,413			0,506	0,913		
167**	0,194	0,382			0,529	0,843		
167#	0,187	0,359			0,189	0,361		
173*	0,203	0,446	0,228		0,472	1,119	0,492	
173**	0,211	0,406	0,234		0,477	1,106	0,487	
173#	0,205	0,394	0,220		0,209	0,395		
179*	0,270	0,407			0,515	1,007		
179**	0,221	0,494			0,536	1,026		
179#	0,213	0,410			0,216	0,413		
191*	0,229	0,447			0,540	1,215		
191**	0,227	0,451			0,515	0,980		
191#	0,227	0,439			0,230	0,446		
233*	0,365	0,768			0,821	1,507		
233**	0,369	1,088			0,807	1,977		
233#	0,404	0,764			0,410	0,776		
257*	0,539	1,185	0,510	1,06	1,251	2,345	1,207	1,04
257**	0,535	1,599	0,507	1,06	1,164	2,309	1,111	1,05
257#	0,535	0,984	0,506	1,06	0,556	0,982	0,525	1,06
307*	0,753	1,518			1,645	3,281		
307**	0,740	1,463,42			1,506	3,593		
307#	0,715	1,320			0,721	1,356		
367*	1,098	2,146			2,504	4,719		
367**	1,180	2,148			2,59	5,319		
367#	1,062	1,988			1,081	2,014		
431*	1,716	3,194			3,769	7,052		
431**	1,626	3,124			4,038	7,533		
431#	1,574	2,937			1,575	2,955		

* - Intel Xeon E5-2695v3 2,30 GHz (CentOS Linux v7.0 x86-64) зі стандартним набором інструкцій;

** - Intel Xeon E5-2695v3 2,30 GHz (CentOS Linux CentOS v7.0 x86-64) з спеціалізованим набором інструкцій;
# - Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2 x86-64) з спеціалізованим набором інструкцій і СМ у два потоки при перевірці ЕЦП;
W – двійкова крива Вейерштрасса;
E - біраціонально еквівалентна крива Едвардса.

Виграш при формуванні та перевірці ЕЦП для ДСТУ 4145-2002 для кривих Едвардса при $d_1 = d_2$ для поля 257, показують: для процесора Intel Xeon E5-2695v3 2,30 GHz (CentOS Linux v7.0 x86-64) та Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2 x86-64) виграш 6% при формуванні та 4% і 6% при перевірці, відповідно.

5.4. Продуктивність використання СК на двійкових кривих Едвардса для Національної системи ЕЦП України

Таблиця 5.4. 1

Результати експериментальної оцінки виконання формування і перевірки ЕЦП для ДСТУ 4145-2002

Поле, m	Час, с					
	Постановка ЕЦП			Перевірка ЕЦП		
	Крива Вейерштрасса	Крива Едвардса	Виграш	Крива Вейерштрасса	Крива Едвардса	Виграш
257*	242,55	229,5	13,05	562,95	543,15	19,8
257**	240,75	228,15	12,6	523,8	499,95	23,85
257#	240,75	227,7	13,05	250,2	236,25	13,95

* - Intel Xeon E5-2695v3 2,30 GHz (CentOS Linux v7.0 x86-64) зі стандартним набором інструкцій;
** - Intel Xeon E5-2695v3 2,30 GHz (CentOS Linux v7.0 x86-64) з спеціалізованим набором інструкцій;
# - Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2 x86-64) з спеціалізованим набором інструкцій процесора.

В табл. 5.4.1 показана швидкодія операції формування та перевірки ЕЦП, використовуючи біраціонально еквівалентні двійкові криві Едвардса при $d_1 = d_2$ для поля 257 до кривих Вейерштрасса з ДСТУ 4145-2002 на прикладі роботи середнього банку, як складової НС ЕЦП, у продовж банківського дня при передачі 450 тис. документів. При передачі документів розглядається лише 1 постановка і 1 перевірка ЕЦП, насправді їх більше.

5.5. Висновки до п’ятого розділу

1. Удосконалений метод здобуття кубічного кореня дозволяє прискорити відшукання біраціонально еквівалентних кривих Едвардса до кривих Вейерштрасса з ДСТУ 4145-2002 в двійковому полі, а також підвищити швидкодію пошуку в 1,3-1,8 разів.

2. Швидкодія операції СМ на кривих Вейерштрасса переважає над біраціонально еквівалентними кривими Едвардса з двома параметрами. При умові $d_1 = d_2$, швидкодія реалізації СМ для кривих Едвардса зростає, починаючи з поля, розміром 257 біт:

– для процесора Intel Core i7-4702MQ 2,2 GHz, (Microsoft Windows 8.1 Pro x64) та Intel Core i7-6700 2,60 GHz (Microsoft Windows 10 x86-64) виграш в швидкодії склав 7%;

– для процесора Intel Core i5-4670 3,40 GHz, (AVX2 version). (Microsoft Windows 7 x86-64) та Intel Xeon E3-1270v5 3,60 GHz, (Microsoft Windows Server 2012 R2) виграш склав 5% та 6% відповідно;

– для процесора Intel Xeon E3-1270v5 3.6GHz (Microsoft Windows Server 2012 R2 x86-64) та Intel Xeon E5-2695v3 2.3 GHz (CentOS Linux v7.0 x86-64) виграш в швидкодії склав 5% і 6%, відповідно.

3. Виграш при формуванні та перевірці ЕЦП для ДСТУ 4145-2002 для кривих Едвардса при $d_1 = d_2$ для поля 257, склав: для процесора Intel Xeon E5-2695v3 2,30 GHz (CentOS Linux v7.0 x86-64) та Intel Xeon E3-1270v5 3,60 GHz (Microsoft Windows Server 2012 R2 x86-64) - 6% при формуванні та 4% і 6% при перевірці, відповідно.

4. Виграш при використанні біраціонально еквівалентних кривих Едвардса для поля 257 згідно ДСТУ 4145-2002 складає при постановці 12-13 с, при перевірці 13-23 с для 1 робочого дня середнього банку, як складової НС ЕЦП.

Список використаних джерел у п’ятому розділі

[1] Національні стандарти України, «ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка», Київ, 2002.

[2] D. Bernstein, T. Lange and R. Rezaeian Farashahi, «Binary Edwards Curves», *Cryptographic Hardware and Embedded Systems – CHES 2008*, pp. 244-265.

[3] M. Rivain, «Fast and regular algorithms for scalar multiplication over elliptic curves», *IACR Cryptology ePrint Archive*, p. 388, 2011.

[4] M. Li, A. Miri and D. Zhu, «Fast Algorithm for Converting Ordinary Elliptic Curves into Binary Edward Form», *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 1, pp. 405-412, 2012.

[5] М. Ковтун, «Применение кривых Эдвардса для защищенной реализации механизмов электронной цифровой подписи согласно ДСТУ 4145-2002», *Системи обробки інформації*, том. 5, №. 151, с. 130-137, 2017.

[6] M. Kovtun, Z.B. Hu, S. Gnatyuk, N. Seilova. «Method of Searching Birationally Equivalent Edwards Curves over Binary Fields», *in Proc. of the 1st International Conference on Computer Science, Engineering and Education Applications (ICCSEEA2018)*, Jan 18-20, 2018, Kiev, Ukraine.

[7] M.G. Kovtun, V.Y. Kovtun, A.A. Okrimenko and S.A. Gnatyuk. «Search method development of birationally equivalent binary Edwards curves for binary Weierstrass curves from DSTU 4145-2002», in *Proc. PIC S&T*, Kharkov, Ukraine, Oct. 13-15, 2015. pp. 5-8. DOI: 10.1109/INFOCOMMST.2015.7357253.

[8] B. Koziel, R. Azarderakhsh and M. Mozaffari-Kermani, «Low-Resource and Fast Binary Edwards Curves Cryptography», *Progress in Cryptology -- INDOCRYPT 2015*, pp. 347-369, 2015.

[9] «Hyperelliptic org», Hyperelliptic.org, 2018. [Online]. Available: http://www.hyperelliptic.org.

[10] K. Kim, C. Lee and C. Negre, «Binary Edwards Curves Revisited», *Progress in Cryptology -- INDOCRYPT 2014*, pp. 393-408, 2014.

[11] P. Montgomery, «Speeding the Pollard and elliptic curve methods of factorization», *Mathematics of Computation*, vol. 48, no. 177, pp. 243-243, 1987.

[12] D. Bernstein, «Batch Binary Edwards», *Advances in Cryptology - CRYPTO 2009*, pp. 317-336, 2009.

[13] N.Sklavos, A.Fournaris, «Binary Edwards Curve Design Strategy for Efficient and Power Attack Resistant Architectures», *Proceedings of the 4th Workshop on Secure Hardware & Security Evaluation, Cryptographic Hardware and Embedded Systems (CHES'15)*, 2015.

ВИСНОВКИ

В дисертаційній роботі, у відповідності до поставленої мети, вирішена науково-технічна задача підвищення швидкодії інформаційно-телекомунікаційних системах ЦСК Національної системи ЕЦП шляхом зменшення обчислювальної складності криптографічних алгоритмів на основі розробки нових методів та удосконалення алгоритмів арифметичних операцій над числами, поліномами і точками ЕК.

У процесі виконання дисертаційної роботи отримані такі основні результати:

1. Проведено аналіз функціонування складових НС ЕЦП України. Встановлено, що функціонування системи на пряму залежить від часу та кількості операцій формування та перевірки ЕЦП. Результати проведеного аналізу дали можливість визначити завдання дисертаційного дослідження щодо розробки та удосконалення методів для підвищення швидкодії інформаційно-телекомунікаційних системах ЦСК НС ЕЦП.

2. Удосконалений метод ділення великих цілих чисел дозволив збільшити швидкодію операції генерації ключів RSA на 7-14% зі збільшенням двійкової довжини, використовуючи компілятор Visual C ++ 2015 на Intel Core i7-6700HQ 2,60 GHz під управлінням Microsoft Windows 10 x86-64.

3. Удосконалений метод інвертування на основі розширеного алгоритму Евкліда в полі $\boldsymbol{GF}\left(2^m\right)$ дозволив підвищити швидкодію при формуванні та перевірці ЕЦП для ДСТУ 4145-2002 в 1.0011-0.0019 і 1.0027-0.0043, використовуючи компілятор Visual C ++ 2015 на Intel Core i7-6700HQ 2,60 GHz під управлінням Windows 10 x86-64. Виграш при постановці та перевірці ЕЦП для 1 робочого дня середнього банку склав 1-9 с та 3-20 с відповідно.

4. Удосконалений метод здобуття n- мірного кореня в полі $\boldsymbol{GF}\left(2^m\right)$, на прикладі здобуття кубічного кореня в полі $\boldsymbol{GF}\left(2^m\right)$ дозволив збільшити швидкодію відшукання біраціонально еквівалентних кривих Едвардса в 1.3-1.8 разів, використовуючи компілятор Visual C ++ 2015 на Intel Core i7-6700HQ 2,60 GHz під управлінням Microsoft Windows 10 x86-64.

5. Розроблений метод автоматизації приведення довільного полінома за фіксованим модулем у полі $\boldsymbol{GF}\left(2^m\right)$, дозволив в незалежності від полінома (тричлена, п'ятичленна), що не приводиться, згенерувати алгоритми приведення за модулем для різних цільових платформ, а також дозволив підвищити швидкодію при формуванні та перевірці ЕЦП, гідно ДСТУ 4145-2002 в 6.-9.4 і 7-9.4 разів відповідно.

6. Використання біраціонально еквівалентних кривих Едвардса з одним параметром для поля 257 при виконанні операції скалярного множення дозволили підвищити швидкодію формування ЕЦП на 5-7% та перевірки ЕЦП на 6-7% для кривих з ДСТУ 4145-2002, в залежності від процесора.

7. Виграш при використанні біраціонально еквівалентних кривих Едвардса для поля 257 згідно ДСТУ 4145-2002 складає при постановці 12-13 с, при перевірці 13-23 с для 1 робочого дня середнього банку, як складової НС ЕЦП.

8. Для методів ділення в стовпчик в кільці цілих чисел, мультиплікативного інвертування та здобуття кубічного кореня в двійковому полі були проведені статистичні тести, щодо часу реалізації з довірчим інтервалом 0.995, показали, що вміст двох вибірок дійсно статистично достовірно різняться, тому удосконалені методи являються більш швидкодіючі, на відміну від прототипів.

9. На основі запропонованих удосконалених методів було розроблено бібліотеку криптографічних примітивів, яка використовується у діяльності «Сайфер ЛТД» (акт від 28.09.2017 року № 12/09-17). Результати дисертаційних досліджень впроваджено у навчальний процес кафедри безпеки інформаційних технологій НАУ (акт впровадження від 18.01.2018 р.).

ДОДАТКИ

ДОДАТОК А. Біраціонально еквівалентні повні криві Едвардса до кривих Вейерштрасса у двійковому полі

Таблиця А. 1

Біраціонально еквівалентні криві та базові точки Едвардса до кривих Вейерштрасса, при умові $d_1 \neq d_2$ з ДСТУ 4145-2002

m=163	
d_1=2cfd2b5ee202a45ddd703a5613477654ce1856a21; d_2=619b17743b13c8c2bf09172fd478da4f5f68ba16c;	X_E=6b57cccdeecfc90122f4d9a06c8a286a9d759ff9; Y_E=67168b7dec6c2cd9c0b876c9b36af475c1cdcfc3a;
m=167	
d_1=2fb7423b788289c550638ee191d5c0e235cc1d5fa4; d_2=514d1e7e3a90d4414c5179bf7c17f4b038f46c046; lambda=1f6dc01e64a349a7af364a3442df553d02acfc4ba7;	X_E=e1d3bc148fbcfd86145914fcca40ba314f2238684; Y_E=353371ebbd5e354e551aeebb347c89a5d55a055c31;
m=173	
d_1=1270c0a05cefab3de162c3b30cdddedaff4004562137; d_2=1f46525dd9dfb37dee43908b5e72ff2553afeee83ed8	X_E=bf95ca3e0df1a8ffa6a12a0936b7a78f324268df4a0; Y_E=19e7c368876e79c903ad174f5b6b7abaf61eb197dbe0;
m=179	
d_1=227e1b7649d10d207a03739a67521c21e3155f47fbbc7; d_2=2cfb3e893554615aabb20e7e185fad537c6d54c0239db;	X_E=2891a5f263a7f8efa5f01a7215ec0c207e8d43f5d5f72; Y_E=7e9c20eeb83176d7b73b638a15e8c7f495b0bbb680dd5;
m=191	
d_1=5df048b880a782a33fb8e44dba4420efb88465b51ca10842; d_2=282f5629d94825bdc6858637222d5e8d3804260f4f57125;	X_E=1b6faa1378083dc132b6ac1096554a7fdc371223f8dd2618; Y_E=4dda88cd114bdb29c58ac21df3debb00c5d23179a7b8d85f;
lambda=4167760cd239092bbb208ecab3b9457953b481e0893ca101;	
m=233	
d_1=19abd1a2690c3facf68d6c27246f3e439c0bb1fddbcb3d62eae0165dac3; d_2=5610cf9259018e18936b27df17a056629cfbfe4617a0028b1038ff7e6c; lambda=1df956ba7bacb7f10622c20f9667674714faf2ac69cd17195e37f9aec10;	X_E=106205c7a18a3ef7042ca2ba2f58b5c48aa6801e5ba933c3a2033b50517; Y_E=18412186d03710d86f492155aeb9d552f65a06e723fab7550b593b90df2;
m=257	
d_1=651a38e75ae7b53003eed5d73b31f23f184f8b110b129d92bd8a3f1c546e05fb;	X_E=1f40e77c3aec15fc99ccc4f5e8e87da77c1962973f18d64f1b9298daa46d8b9b3;

Продовження таблиці А.1

d_2=18342b0dd40458381aa4781266bf1c3846ff148e6a75501f0bb9a9c7c8abcf7bf;	Y_E=19ab0c6021428d9eff6a0368946041f3e0123f80ff8d86b5c4e0bb35b8b0b794;
m=307	
d_1=5793a5de88e69b4b020ffc634d7b34ab416a70abc280aed0f17dd3fbf9d75796672667563c5c3; d_2=66ffa0440824ef471a25f2917f3530575cfdf6151b5b13f41b63e358e163e971b340df0133cf9;	X_E=5d910361c2486822bfea48b0af814c53212caf8801f3a7e215c029e6a031df755a882ed3c7bff; Y_E=64de21ef9c219bc1ebf144fced52d57592819a95f4b2d5380e664d6c18633a78c71c1869a8654;
m=367	
d_1=7d7aa892b7c426c2352bc3a588b6fb156b4289db38748000ea3941980e11bcee1dfc13c4f752b2238df8f68baa4e; d_2=1f8221261dab9dea9610ff878a5baedb764679850fc12e011ebcdf29c16984ae5de0f92644feb6d80845fc960a29; lambda=413600d6d16feddefb3d221c7ad4a97ad7af3ab8a3c6e483b4b0b275a334fd40871000d4daa08f82ca8a779c8a45;	X_E=68f159b03b9d2f99238e1fd50f5821eb9ccac12137f81e93c9cef2a977872a2d7f549794e8bad83040bb98e3074d; Y_E=ce4390e941c2ffe359c9bd3d048dd073d8aac2c07294608436ac3cea0ed017d56bae0cb641ea56d74b2667dd84;
m=431	
d_1=2a53805cbfd2e7d05914d55f429f789329775a8f2280b6b6957eb7df33096026f3387cbcac81c82a1e5c2f4a914de97ebc1cac5f4413;	X_E=3c693638e7029b2785c20ef3bcb16e19343213ef13dabd2e079cd53963dbfc2990b9a1f146552c6daa875d608d144a3533476efab89;
d_2=1d88321e804432f49beae24cb93ea35566861720f33277ab6509c6be6ac5d35a5d962b6a329cc5ae4c19c5448b7cce7dc2676fbc8b3c;	Y_E=202c5bc44f62069d76d32307220e85865609d6b5476cc7fe6eacacedbffcd8839f7ad6fbd7c74b084391af1b21baf7d37ed30fdfd613;

Таблиця А. 2

Бірaціонально еквівалентні криві та базові точки Едвардса до кривих Вейерштрасса, при умові $d_1 = d_2$ з ДСТУ 4145-2002

m=173	
d_1=e92f31c2f97583b5b079a66498651cff964d4edc1;	X_E=9c31a259ac014b272a80452b4c50ab08afd833f48f7; Y_E=18a8bce3975a6c58bc03279c87f7902cbd44b8d6728;
m=257	
d_1=1a561c01c65fda18821a29f0299c88c26c8a85c4f5f326bf152b115950e5ab7a6;	X_E=14f0139b391a9f8dd68a0160b2e1a0e08dd2dfa731e8ce7ffadf5fda4102705d6; Y_E=5cde2ec44b73aee5413d6aa26219ff673ff206f082b073b6ad3b47d2e7f873f5;

Таблиця А. 3

Біраціонально еквівалентні криві та базові точки Едвардса до кривих Вейерштрасса, при умові $d_1 \neq d_2$ для NIST

m=163	
d_1=466149982fc574d67e0ec4955924dbcdf109197fa; d_2=4e2dc6c944d8067c89f9be7066ddd9dfca4a849a; lambda=7e2b8fcd6b686ac9b49c2f80fac4358e6f08ad2a7;	X_E=3e9a08d9ea1c852df5acac86ca5db81bb20b49f3b; Y_E=609e6acab2dcc1c6d6b03bade86339991143c57b2;
m=233	
d_1=7c75c2e99bce88eb638d1ed7c592e46238f9c2c9b5dc44da03ea8c84f9; d_2=107ed38a60f37292ad27e1871db54c36f8db963dbc191964b7e774c6c8b; lambda=18b8d75697defae0ecc2cf91943f6c5e273ec8f4657d060079f75f85b;	X_E=107d082f1447da4a3bc84dc38d2a3359ca9d3135f66f821ae2dc8d3269c; Y_E=211ae7e72b2422d434156e8a36776ce25c0ca9fd29cb44da0bf47e9684;
m=283	
d_1=491957d66d7d2fdda1c7345211e140027ad1b76f4cf61e079d4a0aa1218ae2ea3ed02aa; d_2=45ba60b0890874e20f4d4f6cb6909e33e300548c4c55f32d2cc3ce126055c2a0b1b855d;	X_E=218e55d2a9addd435d5fb4a03f867fdfa1b1beea2be5e4cb5dc1ab09b00e5448c889dd0; Y_E=5e92005556de8bef7c884e495664ef73a85126f257d40c66b83b7c780ee03771c61514a;
m=409	
d_1=1da0f30f93359d4869e22bce6fc0b0fa001b575431c39232fa105d3a0f6e86b064efd5acf9932b4d62e9a8ac734de80a0961c40; d_2=1d3127d261cc98edc94a2b1664d01af3c63112fd302b8844c6417b0dea42b3ccaf54b98c5e4635afac264809a7601a118f9f95d; lambda=29c48015be08f5a5a48b69f28e552c2eb0a309e3ae3451c07ad1ac61f6689de5969f094aee9f2cfa9a573abb1462d4b80910d2;	X_E=1929dbf81beef48ea94048cc2e6041262c0623fa8fbf42f658db0cd745434a4b76e1eef306522fccadc5196c4d65f22eda2a89d; Y_E=683b649d46fc5043533120510939aa3237a67b5f411faa5b0efa441f47ecc6da375bbf94a8c58b574235581d05e5831725fd86;
m=571	
d_1=38d270e03d9b00691eab8e024a0c804297190a2c2f0df8fc0ef47b1ea4b0335fc6f5009a5e20c0b941780a0c45ab1a3072441753f5b6c7a43f2cc6984a689e3e26581ee9ef2ae46; d_2=330b0bc5c5b70c7fef72e0b600a3d2c5fe3885b6671c61fe92fad5338095d22232994499768aea429715cc92573c9036b4b3874bfdfdbf6e6edaa33d4bf9129092729eae474ce53;	X_E=74cc633aae8948ecb289596390f1fdcdf44de2e075823444a032bf5319719bbcddaa575c8758be14cae2bddd0aa5b72598b70c1980571ba31d5583d7d3463de47ae5c16ff736f28; Y_E=7e4c211288e34594667efdaf80a77014615a101183820f9d6462348b2cd0f3438ba853cc1a467edf1efaace8fd6d001c37f29df11637a4efcd6347b1cac822f50fe2e44ec718db4;

ДОДАТОК Б. Розроблені алгоритми здобуття кубічного кореня для двійкових полів з ДСТУ 4145-2002 та ECDSA.

Алгоритм Б.1. Здобуття кубічного кореня в полі $GF\left(2^{163}\right)$

Вхід: $b \in GF\left(2^{163}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{163}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_2}$.

3. $t_0 \leftarrow t_0 \cdot t_0^{s_2}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$.

4. $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$ 5. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2$.

6. Return t_0.

Алгоритм Б.2. Здобуття кубічного кореня в полі $GF\left(2^{167}\right)$

Вхід: $b \in GF\left(2^{167}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{167}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$, $t_1 \leftarrow t_0$.

3. $t_0 \leftarrow t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_2 \leftarrow t_0$

4. $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$.

5. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2$

6. Return t_0.

Алгоритм Б.3. Здобуття кубічного кореня в полі $GF\left(2^{173}\right)$

Вхід: $b \in GF\left(2^{173}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{173}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_1 \leftarrow t_0$.

3. $t_0 \leftarrow t_0^{s_2}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$, $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_3}$.

4. $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_3 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$.

5. $t_0 \leftarrow t_0 \cdot t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$.

6. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3 \cdot b$.

7. Return t_0.

Алгоритм Б.4. Здобуття кубічного кореня в полі $GF\left(2^{179}\right)$

Вхід: $b \in GF\left(2^{179}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{179}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_2}$.

3. $t_0 \leftarrow t_0 \cdot t_0^{s_2}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_4}$

4. $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_3 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$

5. $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3$.

6. Return t_0.

Алгоритм Б.5. Здобуття кубічного кореня в полі $GF\left(2^{191}\right)$

Вхід: $b \in GF\left(2^{191}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{191}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$.

3. $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$.

4. $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_1$

5. Return t_0.

Алгоритм Б.6. Здобуття кубічного кореня в полі $GF\left(2^{233}\right)$

Вхід: $b \in GF\left(2^{233}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{233}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$

3. $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$

4. $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$, $t_3 \leftarrow t_0$.

5. $t_0 \leftarrow t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3 \cdot b$

6. Return t_0.

Алгоритм Б.7. Здобуття кубічного кореня в полі $GF\left(2^{257}\right)$

Вхід: $b \in GF\left(2^{257}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{257}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$, $s_7 = s_6^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$.

3. $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$.

4. $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_7}$, $t_0 \leftarrow t_0 \cdot b$.

5. Return t_0.

Алгоритм Б.8. Здобуття кубічного кореня в полі $GF\left(2^{283}\right)$

Вхід: $b \in GF\left(2^{283}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{283}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$, $s_7 = s_6^2$.

2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_2}$.

3. $t_0 \leftarrow t_0 \cdot t_0^{s_2}$, $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$.

4. $t_3 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$.

5. $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_7}$, $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3$.

6. Return t_0.

Алгоритм Б.9. Здобуття кубічного кореня в полі $GF\left(2^{307}\right)$

Вхід: $b \in GF\left(2^{307}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{307}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$, $s_7 = s_6^2$.
2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_2}$.
3. $t_0 \leftarrow t_0 \cdot t_0^{s_2}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_4}$
4. $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_3 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$
5. $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_7}$, $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3$
6. Return t_0.

Алгоритм Б.10. Здобуття кубічного кореня в полі $GF\left(2^{367}\right)$

Вхід: $b \in GF\left(2^{367}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{367}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$, $s_7 = s_6^2$.
2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$.
3. $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$
4. $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$, $t_3 \leftarrow t_0$.
5. $t_0 \leftarrow t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_0 \leftarrow t_0 \cdot t_0^{s_7}$.
6. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3$.
7. Return t_0.

Алгоритм Б.11. Здобуття кубічного кореня в полі $GF\left(2^{409}\right)$

Вхід: $b \in GF\left(2^{409}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{409}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$, $s_7 = s_6^2$.
2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$
3. $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_3}$, $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_2 \leftarrow t_0$.
4. $t_0 \leftarrow t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$.
5. $t_0 \leftarrow t_0 \cdot t_0^{s_6}$, $t_3 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_7}$, $t_0 \leftarrow t_0 \cdot t_0^{s_7}$
6. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3 \cdot b$.
7. Return t_0.

Алгоритм Б.12. Здобуття кубічного кореня в полі $GF\left(2^{431}\right)$

Вхід: $b \in GF\left(2^{431}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{431}\right)$.

1. $s_1 = 2^2$, $s_2 = s_1^2$, $s_3 = s_2^2$, $s_4 = s_3^2$, $s_5 = s_4^2$, $s_6 = s_5^2$, $s_7 = s_6^2$.
2. $t_0 \leftarrow b$, $t_0 \leftarrow t_0 \cdot t_0^{s_1}$, $t_0 \leftarrow t_0 \cdot t_0^{s_2}$.
3. $t_0 \leftarrow t_0 \cdot t_0^{s_3}$, $t_1 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_4}$, $t_0 \leftarrow t_0 \cdot t_0^{s_4}$
4. $t_2 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_5}$, $t_0 \leftarrow t_0 \cdot t_0^{s_6}$
5. $t_3 \leftarrow t_0$, $t_0 \leftarrow t_0^{s_7}$, $t_0 \leftarrow t_0 \cdot t_0^{s_7}$.
6. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3$
7. Return t_0.

Алгоритм Б.13. Здобуття кубічного кореня в полі $GF\left(2^{571}\right)$

Вхід: $b \in GF\left(2^{571}\right)$.

Вихід: $\sqrt[3]{b} \in GF\left(2^{571}\right)$.

1. $s_1 = 2^2,\ s_2 = s_1^2,\ s_3 = s_2^2,\ s_4 = s_3^2,\ s_5 = s_4^2,\ s_6 = s_5^2,\ s_7 = s_6^2,\ s_8 = s_7^2$.

2. $t_0 \leftarrow b,\ t_0 \leftarrow t_0 \cdot t_0^{s_1},\ t_1 \leftarrow t_0,\ t_0 \leftarrow t_0^{s_2},\ t_0 \leftarrow t_0 \cdot t_0^{s_2},\ t_2 \leftarrow t_0,\ t_0 \leftarrow t_0^{s_3},\ t_0 \leftarrow t_0 \cdot t_0^{s_3},\ t_3 \leftarrow t_0$.

3. $t_0 \leftarrow t_0^{s_4},\ t_0 \leftarrow t_0 \cdot t_0^{s_4},\ t_4 \leftarrow t_0,\ t_0 \leftarrow t_0^{s_5},\ t_0 \leftarrow t_0 \cdot t_0^{s_5},\ t_0 \leftarrow t_0 \cdot t_0^{s_6},\ t_0 \leftarrow t_0 \cdot t_0^{s_7},\ t_0 \leftarrow t_0 \cdot t_0^{s_8}$

4. $t_0 \leftarrow t_0 \cdot t_1 \cdot t_2 \cdot t_3 \cdot t_4$

5. Return t_0.

ДОДАТОК В. Статистичний аналіз даних

Таблиця В.1

Вибірка середнього часу для удосконаленого методу ділення та прототипу в мс

div_10	div_p_10	div_20	div_p_20	div_30	div_p_30	div_40	div_p_40	div_50	div_p_50	div_60	div_p_60	div_70	div_p_70	div_80	div_p_80	div_90	div_p_90
144,1581	449,2142	132,2049	389,5593	110,0798	342,5704	95,38025	294,303	87,33205	228,6611	68,28742	175,3506	45,39418	121,1638	32,77667	84,64118	15,07501	39,56416
144,0921	449,1409	132,0492	389,7551	110,29	342,7465	95,18847	294,145	87,51809	228,7104	68,14463	175,4492	45,44775	121,2872	32,79766	84,56039	15,20698	39,64017
144,1665	449,1865	132,2404	389,5483	110,0762	342,7737	95,38253	294,1662	87,52117	228,6727	68,07514	175,3761	45,56553	121,1888	32,88533	84,57071	15,24638	39,74583
144,0159	449,1982	132,1631	389,7094	110,2265	342,7394	95,17527	294,2047	87,27892	228,5654	68,26374	175,2754	45,57225	121,2682	33,00603	84,69619	15,3105	39,70882
144,0612	449,1876	132,3344	389,6804	110,3279	342,5946	95,23564	294,3161	87,49814	228,7225	68,28434	175,3835	45,60077	121,2865	32,84426	84,79364	15,14545	39,66072
144,1037	449,2354	132,2885	389,7685	110,3464	342,7344	95,40594	294,1308	87,34134	228,6074	68,35274	175,3242	45,36787	121,238	32,81349	84,5935	15,27004	39,79725
144,1703	449,1834	132,1458	389,622	110,1081	342,791	95,22934	294,3498	87,30457	228,6652	68,24418	175,447	45,33741	121,2187	32,9116	84,68016	15,13694	39,56589
144,0123	449,1849	132,2908	389,6889	110,1579	342,5944	95,29067	294,3222	87,57346	228,7309	68,04866	175,3238	45,59726	121,1239	33,02764	84,71727	15,22338	39,61383
144,0786	449,3209	132,2518	389,6645	110,0694	342,6943	95,22227	294,2997	87,3244	228,5195	68,23093	175,4281	45,44199	121,3193	32,93537	84,78387	15,14822	39,57076
144,0765	449,2555	132,1314	389,5837	110,3383	342,7806	95,44439	294,1249	87,41466	228,6622	68,16502	175,3165	45,62708	121,2427	32,97632	84,58916	15,17585	39,65356
144,1073	449,2544	132,1579	389,6072	110,3028	342,7767	95,16667	294,1494	87,3396	228,5383	68,22663	175,3003	45,50908	121,1877	32,85144	84,64656	15,12781	39,67811
144,1887	449,1649	132,0375	389,7391	110,1744	342,5738	95,39044	294,1203	87,26286	228,7131	68,05358	175,3716	45,48465	121,1944	32,98704	84,61432	15,13789	39,78623
144,0537	449,265	132,219	389,7195	110,1344	342,6134	95,19143	294,3595	87,42279	228,6058	68,21873	175,3568	45,3901	121,3056	32,85209	84,67966	15,31436	39,62385
144,1007	449,283	132,0766	389,5941	110,1726	342,5544	95,246	294,2909	87,26866	228,6065	68,20899	175,2771	45,60726	121,1773	32,94475	84,68932	15,07439	39,63369
144,1988	449,3139	132,1091	389,5607	110,3117	342,6631	95,32775	294,1194	87,44616	228,6503	68,25834	175,4241	45,58534	121,1052	32,95522	84,628	15,23733	39,63512
144,1232	449,1545	132,3435	389,5734	110,3239	342,6911	95,30318	294,1474	87,49574	228,5147	68,12629	175,3733	45,64441	121,1529	32,97085	84,65082	15,05337	39,77151
144,1819	449,165	132,0619	389,7085	110,0578	342,5849	95,25812	294,1257	87,26629	228,7395	68,31648	175,3052	45,47672	121,255	32,76576	84,61543	15,09671	39,67317

Продовження таблиці В.1

144,1757	449,2665	132,0876	389,6227	110,3242	342,6403	95,45348	294,1513	87,52387	228,5754	68,04999	175,3212	45,65308	121,3282	32,99464	84,68965	15,32713	39,56337
144,089	449,1934	132,2802	389,6927	110,2691	342,7287	95,28685	294,2484	87,56414	228,5273	68,20516	175,3208	45,36523	121,3335	32,76579	84,66554	15,27505	39,77071
144,1565	449,2079	132,0558	389,6107	110,0579	342,7509	95,44485	294,1186	87,43772	228,5501	68,27005	175,4476	45,60945	121,2753	32,74416	84,60096	15,04124	39,70593
144,1954	449,3178	132,1903	389,7672	110,0395	342,6063	95,24227	294,3191	87,32065	228,6908	68,31474	175,4789	45,65251	121,145	32,85462	84,71497	15,07048	39,59095
144,1385	449,2736	132,0857	389,7647	110,1307	342,6692	95,14214	294,3382	87,42728	228,6494	68,14978	175,4148	45,64371	121,2773	32,8893	84,74004	15,33547	39,65394
144,2012	449,2442	132,1414	389,6102	110,0933	342,7963	95,42306	294,3114	87,4794	228,685	68,03083	175,4004	45,36626	121,3326	32,84547	84,65829	15,14234	39,63016
144,1673	449,2975	132,1908	389,5798	110,3359	342,7338	95,31532	294,2876	87,2876	228,6421	68,13953	175,3192	45,36396	121,2228	32,72646	84,54864	15,31353	39,69586
144,0853	449,2839	132,1347	389,726	110,104	342,6894	95,39462	294,1964	87,47686	228,5571	68,18834	175,4159	45,41239	121,339	32,80199	84,75832	15,22529	39,5756
144,0126	449,2401	132,3	389,7131	110,2406	342,6115	95,29476	294,3261	87,48851	228,6985	68,29637	175,4151	45,53359	121,284	32,8592	84,57412	15,24561	39,66027
144,0586	449,1412	132,2458	389,7499	110,3078	342,5843	95,28691	294,1978	87,36888	228,6035	68,10005	175,2671	45,47806	121,2762	32,70996	84,7697	15,05019	39,76449
144,0424	449,1365	132,3312	389,7908	110,2663	342,6404	95,25898	294,271	87,2654	228,6349	68,33732	175,4587	45,59668	121,2493	32,94426	84,7558	15,31912	39,63909
144,0364	449,2106	132,3306	389,5848	110,138	342,6524	95,29495	294,3074	87,39289	228,6524	68,22252	175,3115	45,43269	121,2094	32,96566	84,63352	15,33131	39,73937
144,0282	449,1943	132,0522	389,6558	110,1736	342,6598	95,30966	294,2681	87,29419	228,7264	68,17015	175,4825	45,43021	121,2077	32,94857	84,61069	15,15374	39,56169
144,0082	449,1458	132,0567	389,7422	110,2953	342,769	95,25103	294,1146	87,58029	228,6451	68,17917	175,3822	45,60418	121,1462	32,97622	84,61936	15,21594	39,55216
144,1707	449,2907	132,2366	389,6073	110,3498	342,6182	95,34493	294,1849	87,29895	228,6457	68,05411	175,3168	45,36631	121,1776	32,73942	84,79004	15,29037	39,61746
144,0126	449,2152	132,2767	389,7849	110,2345	342,7768	95,4317	294,1398	87,38808	228,7305	68,34411	175,3782	45,50799	121,3361	32,75959	84,77593	15,28525	39,64796
144,023	449,2604	132,281	389,756	110,2668	342,7761	95,27	294,3526	87,41447	228,6128	68,29582	175,4533	45,44841	121,2629	32,85952	84,65824	15,31037	39,75874
144,057	449,2045	132,1932	389,6593	110,2494	342,7554	95,32646	294,1344	87,45323	228,5468	68,08182	175,3182	45,62655	121,1886	32,9593	84,61796	15,14538	39,69617
144,1043	449,2021	132,1888	389,633	110,0643	342,6119	95,38637	294,1517	87,41253	228,6932	68,17542	175,4428	45,55757	121,1881	32,90815	84,55581	15,08248	39,70388

Продовження таблиці В.1

144,0887	449,126	132,328	389,7826	110,3324	342,582	95,40068	294,1393	87,43631	228,5766	68,24524	175,4568	45,37634	121,3108	32,77099	84,55591	15,18887	39,62136
144,0311	449,1439	132,2827	389,7396	110,2227	342,7327	95,34925	294,3568	87,43127	228,7168	68,22172	175,2858	45,36121	121,0983	32,92181	84,75501	15,3211	39,71527
144,0939	449,1465	132,0986	389,7892	110,3095	342,7735	95,13907	294,1686	87,27931	228,757	68,04247	175,2679	45,49504	121,166	32,71239	84,61535	15,32115	39,64287
144,0954	449,3085	132,2815	389,6457	110,0517	342,5954	95,21513	294,1339	87,39098	228,6784	68,18139	175,3727	45,60385	121,1538	32,73393	84,6451	15,11037	39,58405
144,0154	449,1544	132,2519	389,7847	110,0581	342,6602	95,44117	294,1426	87,26984	228,6991	68,30898	175,4379	45,34685	121,1251	32,85226	84,69648	15,26544	39,71465
144,167	449,2178	132,3224	389,6754	110,2059	342,7032	95,4294	294,2292	87,50475	228,7471	68,31231	175,3506	45,4136	121,1892	32,70364	84,72925	15,3059	39,72782
144,0751	449,2696	132,1617	389,6876	110,0922	342,6771	95,3605	294,1873	87,54615	228,7639	68,14632	175,3693	45,49931	121,1006	32,95126	84,55806	15,05337	39,64418
144,084	449,2546	132,3558	389,6792	110,2978	342,7781	95,15808	294,2867	87,46252	228,7385	68,33207	175,503	45,60018	121,2985	33,01211	84,64841	15,2515	39,61061
144,1567	449,2973	132,2896	389,7342	110,07	342,7887	95,21457	294,1239	87,45201	228,7463	68,34536	175,479	45,41826	121,3008	32,93487	84,55687	15,11554	39,71879
144,0252	449,1564	132,0776	389,7025	110,1326	342,6211	95,29717	294,1638	87,43444	228,6499	68,03904	175,3312	45,55315	121,1703	32,97152	84,67926	15,14719	39,79245
144,1511	449,13	132,2607	389,73	110,0584	342,766	95,33137	294,2737	87,283	228,5456	68,05967	175,3974	45,5494	121,1869	32,8572	84,70433	15,33915	39,76472
144,1996	449,2785	132,3459	389,7777	110,0834	342,577	95,44291	294,1145	87,27262	228,6524	68,28762	175,456	45,62103	121,2248	32,77354	84,69678	15,28918	39,70833
144,2	449,2348	132,3092	389,6355	110,2211	342,6187	95,14752	294,1958	87,44808	228,5969	68,33072	175,34	45,50928	121,1896	32,99555	84,65181	15,08193	39,66057
144,1534	449,1819	132,3394	389,7138	110,026	342,7303	95,38005	294,2848	87,42157	228,5512	68,25891	175,2938	45,4018	121,0947	32,893	84,76437	15,17729	39,77505
144,0954	449,1894	132,2747	389,7665	110,1515	342,7099	95,34776	294,2785	87,42499	228,598	68,12462	175,2631	45,43237	121,1389	32,94193	84,75053	15,31114	39,71545
144,0896	449,1427	132,1334	389,7952	110,0593	342,7432	95,33313	294,3195	87,34483	228,5905	68,15287	175,2657	45,57426	121,1506	32,98022	84,77238	15,03877	39,68714
144,1545	449,276	132,2474	389,5843	110,336	342,7778	95,16847	294,1544	87,26854	228,7477	68,30104	175,3903	45,59661	121,1348	32,90939	84,59199	15,35471	39,62015
144,1095	449,2004	132,1448	389,6591	110,093	342,7065	95,43055	294,1898	87,44787	228,7364	68,18481	175,2587	45,42562	121,1477	32,89898	84,59745	15,18804	39,6082
144,1772	449,2074	132,3105	389,6205	110,2575	342,6117	95,35019	294,2424	87,37859	228,6158	68,30211	175,494	45,34097	121,2164	32,88209	84,70354	15,10377	39,73002

Продовження таблиці В.1

144,0089	449,2379	132,0468	389,7174	110,2025	342,6621	95,34819	294,3597	87,54702	228,5782	68,1675	175,3665	45,607	121,1052	32,75305	84,71842	15,29999	39,63512
144,1675	449,1818	132,218	389,7904	110,2047	342,5991	95,33165	294,2086	87,29195	228,5324	68,34038	175,4502	45,61936	121,1635	32,90326	84,59645	15,28589	39,6602
144,005	449,2165	132,3268	389,7444	110,3024	342,5715	95,19186	294,1968	87,58042	228,752	68,2218	175,3663	45,32372	121,1468	32,72629	84,69775	15,0798	39,686
144,028	449,3079	132,2828	389,675	110,0585	342,5755	95,42131	294,3001	87,32199	228,5997	68,05875	175,44	45,40099	121,3116	32,74462	84,63339	15,17891	39,73839
144,1141	449,145	132,2812	389,5956	110,1674	342,7659	95,44443	294,2288	87,35739	228,5832	68,10097	175,4824	45,53117	121,2798	32,86513	84,6754	15,25754	39,66904
144,0425	449,2587	132,3145	389,7835	110,2467	342,5831	95,31905	294,1124	87,41602	228,7563	68,0565	175,3274	45,616	121,2899	33,01412	84,60073	15,09495	39,63283
144,0707	449,177	132,236	389,7762	110,213	342,5895	95,31582	294,2508	87,35249	228,6504	68,13285	175,445	45,32476	121,2355	32,81602	84,56013	15,24718	39,73933
144,1699	449,1619	132,1415	389,5695	110,151	342,5793	95,18141	294,2656	87,5012	228,6394	68,05724	175,3239	45,56528	121,3233	32,83386	84,60122	15,28858	39,60185
144,1126	449,2008	132,0813	389,7919	110,353	342,689	95,19351	294,1354	87,57473	228,5514	68,27532	175,3356	45,51626	121,2806	32,74337	84,75805	15,33774	39,72556
144,0764	449,124	132,2762	389,7121	110,2032	342,7973	95,17153	294,2091	87,41128	228,5402	68,20364	175,4887	45,52734	121,34	33,00248	84,65932	15,35588	39,63577
144,0482	449,2343	132,0578	389,5953	110,1098	342,5599	95,13657	294,1453	87,48155	228,7404	68,12694	175,3534	45,43207	121,1124	32,93275	84,71273	15,18342	39,77552
144,1477	449,1434	132,0555	389,7654	110,1662	342,7258	95,25564	294,3301	87,45874	228,6072	68,32045	175,3661	45,54379	121,2085	32,84079	84,66891	15,1496	39,69103
144,1842	449,1337	132,1617	389,5911	110,3399	342,7794	95,2696	294,2908	87,38967	228,6028	68,03399	175,2835	45,54896	121,2802	32,85173	84,77757	15,26739	39,73389
144,1809	449,2809	132,1662	389,6582	110,107	342,7149	95,1525	294,3066	87,56672	228,6908	68,17114	175,4218	45,48962	121,2421	33,00955	84,62944	15,2048	39,57223
144,0713	449,2583	132,304	389,6631	110,1501	342,628	95,39949	294,3403	87,41161	228,6492	68,21297	175,2969	45,43406	121,3388	32,72363	84,69697	15,25771	39,64708
144,0417	449,1613	132,3127	389,6187	110,3241	342,7974	95,19496	294,3404	87,56317	228,5423	68,29814	175,489	45,54366	121,1119	32,93182	84,5563	15,34853	39,63462
144,0171	449,1273	132,3475	389,7451	110,063	342,6151	95,36617	294,314	87,43013	228,7111	68,18703	175,3582	45,58728	121,1149	32,8108	84,63551	15,13554	39,55005
144,0753	449,1381	132,2437	389,5927	110,1103	342,6971	95,42826	294,2504	87,29308	228,6475	68,26682	175,3683	45,36153	121,2338	32,964	84,55393	15,31334	39,79579
144,102	449,2226	132,0566	389,7146	110,0652	342,7936	95,2271	294,2108	87,35887	228,7201	68,21856	175,3951	45,42612	121,1193	32,74031	84,73885	15,28676	39,58608

Продовження таблиці В.1

144,0253	449,1478	132,2231	389,7843	110,2719	342,7717	95,20078	294,1613	87,30789	228,6632	68,16483	175,3838	45,38853	121,1901	32,86797	84,67064	15,18232	39,65617
144,0131	449,1295	132,1504	389,7976	110,2641	342,7246	95,24924	294,2659	87,43117	228,6663	68,33633	175,476	45,62126	121,1994	32,94059	84,6054	15,09868	39,60741
144,151	449,1877	132,3031	389,5703	110,1512	342,6963	95,3849	294,3509	87,49284	228,6175	68,08783	175,3435	45,62086	121,3329	32,7339	84,77108	15,19148	39,65617
144,0622	449,152	132,1971	389,6114	110,0425	342,7503	95,42735	294,3106	87,29168	228,6814	68,03543	175,457	45,44197	121,3243	32,9839	84,60925	15,27209	39,68288
144,1289	449,1337	132,3378	389,6263	110,2548	342,5564	95,30589	294,1893	87,54537	228,712	68,25469	175,4568	45,60967	121,2848	32,73316	84,78283	15,04691	39,61121
144,1453	449,1596	132,2623	389,702	110,0824	342,7145	95,35986	294,3444	87,544	228,6763	68,04304	175,497	45,4791	121,1953	32,72402	84,63669	15,20798	39,74232
144,0249	449,1891	132,3249	389,5546	110,1854	342,5481	95,38453	294,1353	87,33095	228,6767	68,33571	175,2715	45,3733	121,2344	32,7235	84,6509	15,02309	39,72519
144,1302	449,1739	132,0303	389,7737	110,1443	342,6643	95,42211	294,1412	87,53596	228,6134	68,21699	175,3772	45,51058	121,281	33,0105	84,59277	15,3472	39,56245
144,0921	449,268	132,258	389,7609	110,1929	342,6436	95,23991	294,2284	87,56222	228,7006	68,24487	175,4972	45,41963	121,289	32,78705	84,69542	15,2353	39,72964
144,0039	449,1824	132,2258	389,6417	110,2221	342,6662	95,21209	294,1983	87,38359	228,5296	68,29779	175,3376	45,46633	121,0917	32,88888	84,54932	15,31366	39,61869
144,1647	449,1807	132,1271	389,6739	110,0248	342,767	95,34459	294,2113	87,36541	228,7356	68,13212	175,4745	45,43211	121,3382	32,71953	84,77556	15,24266	39,75652
144,0911	449,1702	132,1498	389,5633	110,3192	342,6616	95,34855	294,2507	87,50655	228,5898	68,11174	175,2844	45,40073	121,1414	32,9507	84,77568	15,20211	39,70039
144,0264	449,3218	132,0385	389,6074	110,3179	342,7348	95,39436	294,2431	87,45246	228,6056	68,09162	175,4327	45,36147	121,127	32,77613	84,6993	15,13452	39,72236
144,1415	449,2699	132,0885	389,549	110,0886	342,7776	95,33509	294,2944	87,40156	228,544	68,07747	175,4485	45,52034	121,2208	32,91425	84,61263	15,03409	39,61077
144,1849	449,1932	132,1803	389,703	110,1734	342,6169	95,21545	294,3379	87,53573	228,6885	68,09263	175,4781	45,4651	121,1658	32,74792	84,67936	15,07745	39,76788
144,1167	449,2343	132,0495	389,6753	110,2211	342,6595	95,17487	294,3071	87,42545	228,7581	68,28626	175,4934	45,33983	121,2474	32,80032	84,75628	15,0858	39,74213
144,0077	449,2216	132,1054	389,7721	110,2039	342,6748	95,2325	294,1225	87,31945	228,557	68,27015	175,3061	45,38654	121,0938	32,8268	84,74541	15,20384	39,71109
144,1686	449,1304	132,2182	389,603	110,2904	342,5566	95,35085	294,1659	87,49359	228,6004	68,16695	175,4419	45,351	121,3083	32,72569	84,73914	15,04941	39,67156
144,1243	449,259	132,3455	389,7856	110,0522	342,6628	95,21448	294,3057	87,26026	228,5246	68,28473	175,2976	45,5874	121,2655	33,00551	84,77933	15,20038	39,67021

Продовження таблиці В.1

144,1101	449,2031	132,1206	389,6267	110,1002	342,7904	95,32718	294,1859	87,46109	228,6714	68,21782	175,3348	45,36352	121,1011	32,86824	84,73809	15,0691	39,57377
144,077	449,2638	132,1724	389,7852	110,282	342,7324	95,40721	294,1728	87,41443	228,566	68,21805	175,4138	45,63937	121,1202	32,74695	84,6623	15,04492	39,6637
144,1265	449,2982	132,1544	389,6935	110,1276	342,7398	95,2086	294,3032	87,54276	228,6071	68,04501	175,481	45,60668	121,2995	32,80814	84,73417	15,2208	39,59165
144,1729	449,2269	132,1191	389,6893	110,2913	342,7137	95,31228	294,2429	87,453	228,7199	68,26929	175,4953	45,59565	121,1075	32,93224	84,66786	15,23893	39,69865
144,0548	449,1394	132,3432	389,5589	110,2233	342,5874	95,39097	294,2988	87,54887	228,7628	68,31107	175,315	45,51607	121,2583	32,79408	84,78829	15,06885	39,5559
144,1177	449,247	132,2414	389,7377	110,1147	342,6784	95,40329	294,1587	87,30644	228,566	68,2996	175,2637	45,51267	121,2894	32,7087	84,74218	15,14307	39,67036
144,0256	449,3022	132,0307	389,5601	110,297	342,7862	95,14418	294,1348	87,5815	228,7557	68,32492	175,4823	45,61815	121,2306	32,97357	84,55637	15,2811	39,61577
144,1453	449,2905	132,2488	389,6569	110,0454	342,7329	95,22995	294,3506	87,44879	228,6075	68,08526	175,4772	45,38349	121,1568	33,02369	84,65605	15,22615	39,62951
144,0296	449,2149	132,0374	389,7908	110,0984	342,7583	95,33642	294,1934	87,27349	228,7573	68,34223	175,401	45,40795	121,2217	32,75901	84,55018	15,27692	39,73147
144,034	449,1411	132,1681	389,7064	110,3321	342,7514	95,35275	294,2654	87,50564	228,7629	68,2123	175,388	45,53998	121,3369	32,99095	84,56723	15,29496	39,6729
144,1333	449,2834	132,0322	389,6279	110,332	342,7118	95,1359	294,1807	87,4181	228,7283	68,30636	175,2961	45,45906	121,3115	33,02474	84,65015	15,31742	39,71104
144,0712	449,133	132,2214	389,7833	110,3523	342,5573	95,21682	294,336	87,57382	228,6423	68,07667	175,387	45,62058	121,1424	32,76992	84,7089	15,04349	39,69774
144,0413	449,1662	132,1979	389,7065	110,3132	342,7434	95,23793	294,2747	87,45041	228,6767	68,11609	175,3681	45,55531	121,2724	32,945	84,62361	15,13059	39,70557
144,0923	449,3208	132,2922	389,6079	110,3221	342,7487	95,14658	294,145	87,49656	228,6141	68,33519	175,4614	45,62521	121,2775	32,75046	84,63348	15,23512	39,66727
144,0341	449,193	132,0741	389,6328	110,2541	342,6122	95,12855	294,2613	87,46225	228,7229	68,04522	175,3709	45,37678	121,2043	32,9321	84,74075	15,32407	39,79718
144,1038	449,3114	132,1798	389,7253	110,2833	342,767	95,42299	294,2203	87,49073	228,6351	68,14739	175,465	45,43472	121,2096	32,87059	84,78059	15,23598	39,59363
144,052	449,3103	132,135	389,7854	110,2256	342,5657	95,16885	294,3565	87,3717	228,5902	68,13582	175,4048	45,41147	121,1827	32,85544	84,63655	15,10803	39,7455
144,1637	449,2775	132,1938	389,5804	110,0339	342,5605	95,15347	294,1482	87,33986	228,5617	68,32205	175,4126	45,37794	121,1214	32,97195	84,69845	15,2255	39,55675
144,0487	449,1389	132,1722	389,738	110,2708	342,7042	95,42525	294,2288	87,37549	228,6369	68,20829	175,2828	45,47333	121,1953	32,9437	84,55148	15,15046	39,629

Продовження таблиці В.1

144,0931	449,3209	132,2853	389,5585	110,3064	342,5991	95,22516	294,3105	87,48095	228,6759	68,08198	175,4238	45,35662	121,1513	32,89894	84,71073	15,0704	39,65568
144,0172	449,1286	132,1975	389,6184	110,3433	342,5711	95,20035	294,1845	87,46978	228,6237	68,3228	175,3048	45,4098	121,2772	32,91858	84,65064	15,22102	39,60406

- div –удосконалений метод ділення;
- div_p – прототип.

Таблиця В.2

t-критерій Стьюдента для випадку, коли кількість машинних слів: діленого та дільника - 1024 (32368) з 99.5% довірчим інтервалом

ρ		**Критерій рівності дисперсій Лівіня**		**t-критерій для рівності середніх**						
		F	Знач.	т	ст.св.	Знач. (2-х стор.)	Різниця середніх	Серед. кв. похиб. різн.	Нижня	Верхня
10	D	3,370	,068	-40947,368	226	0,000	-305,124465	,007452	-305,124933	-305,123997
	ND			-40947,368	223,132	0,000	-305,124465	,007452	-305,124933	-305,123997
20	D	3,617	,058	-23020,538	226	0,000	-257,48727	,01119	-257,48797	-257,48657
	ND			-23020,538	220,730	0,000	-257,48727	,01119	-257,48797	-257,48657
30	D	38,943	,000	-20133,434	226	0,000	-232,470904	,011547	-232,471628	-232,470179
	ND			-20133,434	194,987	0,000	-232,470904	,011547	-232,471628	-232,470179
40	D	15,456	,000	-17385,812	226	0,000	-198,953588	,011443	-198,954306	-198,952869
	ND			-17385,812	209,683	0,000	-198,953588	,011443	-198,954306	-198,952869
50	D	10,518	,001	-12767,371	226	0,000	-141,201342	,011060	-141,202036	-141,200648
	ND			-12767,371	211,231	0,000	-141,201342	,011060	-141,202036	-141,200648
60	D	12,416	,001	-9866,642	226	0,000	-107,204974	,010865	-107,205656	-107,204292
	ND			-9866,642	210,732	0,000	-107,204974	,010865	-107,205656	-107,204292

Продовження таблиці В.2

70	D	21,320	,000	-6662,023	226	0,000	-75,725132	,011367	-75,725845	-75,724418
	ND			-6662,023	204,676	0,000	-75,725132	,011367	-75,725845	-75,724418
80	D	18,962	,000	-4693,737	226	0,000	-51,796658	,011035	-51,797351	-51,795965
	ND			-4693,737	207,206	0,000	-51,796658	,011035	-51,797351	-51,795965
90	D	10,784	,001	-2147	226	0,000	-24,496	,0114	-24,49	-24,49
	ND			-2147	209	0,000	-24,49	,0114	-24,494032	-24,49

- D - передбачається рівність дисперсій;
- ND - не передбачається рівність дисперсій.

Таблиця В.3

Вибірка середнього часу для удосконаленого методу мультиплікативного інвертування та прототип в мкс

431		**367**		**307**		**257**		**233**	
Inv	**Inv_p**	**Inv**	**Inv_p**	**Inv**	**Inv_p**	**Inv**	**Inv_p**	**Inv**	**Inv_p**
0,017053	0,028049	0,013998	0,021529	0,011877	0,01824	0,008509	0,013475	0,007509	0,01188
0,017059	0,028074	0,013993	0,021549	0,011879	0,018217	0,008508	0,013508	0,007503	0,011874
0,017053	0,028007	0,013997	0,021554	0,011875	0,018186	0,00851	0,013509	0,007509	0,011874
0,01705	0,027992	0,013992	0,021557	0,011877	0,018214	0,008507	0,013529	0,007506	0,011879
0,017057	0,028042	0,013999	0,021557	0,011873	0,01821	0,008507	0,013565	0,007502	0,011871
0,017051	0,028002	0,014	0,021527	0,01188	0,01822	0,008505	0,013509	0,007505	0,011878
0,017055	0,028052	0,013997	0,021558	0,011876	0,01818	0,008505	0,013504	0,007506	0,011877
0,01706	0,028068	0,013998	0,021492	0,011874	0,018207	0,008507	0,013556	0,007507	0,01187
0,017056	0,028004	0,013993	0,021519	0,011877	0,018217	0,008508	0,013544	0,007507	0,011871
0,017056	0,028049	0,013996	0,02148	0,011874	0,018184	0,008508	0,013523	0,007504	0,011871
0,017054	0,028001	0,013999	0,021569	0,011875	0,018174	0,008504	0,013499	0,007502	0,011873
0,017051	0,028038	0,013995	0,021563	0,011875	0,018175	0,0085	0,013489	0,0075	0,011879
0,017055	0,027992	0,013994	0,021551	0,011877	0,018262	0,008508	0,013518	0,007507	0,011871
0,017056	0,028	0,013996	0,021551	0,011877	0,018172	0,008505	0,013518	0,00751	0,011873
0,017056	0,028033	0,013993	0,021536	0,011876	0,018186	0,008503	0,013491	0,007507	0,011879
0,017051	0,028001	0,013996	0,02149	0,01187	0,018239	0,008502	0,013565	0,007503	0,011872
0,017058	0,027999	0,013997	0,021537	0,01187	0,018194	0,008504	0,013505	0,007503	0,011877
0,017053	0,028045	0,013999	0,02152	0,011875	0,018197	0,008504	0,013553	0,007506	0,011877
0,017052	0,028077	0,013991	0,021505	0,01188	0,018269	0,008505	0,013542	0,007506	0,011877
0,017051	0,028055	0,013991	0,021526	0,011875	0,018175	0,008501	0,013512	0,007503	0,011871
0,017055	0,028042	0,013998	0,021557	0,011878	0,018192	0,008506	0,013569	0,007506	0,011871
0,017057	0,027985	0,013997	0,02156	0,011875	0,018211	0,008505	0,013526	0,007504	0,011875
0,017058	0,028064	0,013998	0,021502	0,011874	0,018225	0,008508	0,013535	0,007503	0,011875
0,017051	0,027995	0,013991	0,021552	0,011875	0,018207	0,008509	0,013486	0,007506	0,011877

Продовження таблиці В.3

0,017053	0,028034	0,013993	0,021564	0,011879	0,01825	0,008504	0,013484	0,007504	0,011879
0,017052	0,027992	0,013992	0,021495	0,011873	0,018212	0,008509	0,013499	0,007504	0,011875
0,017053	0,028046	0,013991	0,021494	0,011871	0,018235	0,008501	0,013564	0,007506	0,011871
0,017052	0,02802	0,013993	0,021535	0,011879	0,018187	0,008502	0,013556	0,007504	0,011876
0,017053	0,028019	0,013996	0,021563	0,011879	0,018267	0,008503	0,01356	0,007509	0,011879
0,017058	0,028038	0,013993	0,021495	0,011872	0,018251	0,008508	0,013503	0,007503	0,011872
0,017051	0,028003	0,013991	0,021505	0,011873	0,018267	0,008507	0,013528	0,007506	0,011879
0,017057	0,027996	0,013992	0,021523	0,011876	0,018224	0,008503	0,013475	0,00751	0,011876
0,017051	0,028032	0,013995	0,021475	0,011874	0,018254	0,00851	0,013551	0,007502	0,01187
0,017057	0,028009	0,013996	0,021552	0,011878	0,018183	0,008507	0,013514	0,007504	0,01187
0,017056	0,028053	0,013992	0,021564	0,011877	0,018254	0,008507	0,01349	0,007507	0,011873
0,017054	0,028066	0,013998	0,021511	0,011872	0,018265	0,008501	0,013485	0,007507	0,011872
0,017056	0,028079	0,013993	0,021518	0,011877	0,018215	0,008501	0,01349	0,007501	0,011875
0,017059	0,028014	0,013998	0,021541	0,011875	0,018182	0,008507	0,013546	0,007504	0,01188
0,017051	0,028047	0,013998	0,021569	0,011878	0,018172	0,008509	0,01354	0,007509	0,011871
0,01705	0,02802	0,013991	0,021521	0,011873	0,018205	0,008501	0,01353	0,00751	0,011871
0,017056	0,027999	0,013991	0,021543	0,011873	0,018188	0,008507	0,013563	0,007501	0,01188
0,017053	0,028075	0,013996	0,02149	0,011871	0,018221	0,008502	0,013478	0,007501	0,011877
0,017058	0,028039	0,013995	0,021476	0,011875	0,018242	0,008501	0,013515	0,00751	0,011871
0,01706	0,028039	0,013991	0,021505	0,01187	0,018244	0,008501	0,013492	0,007504	0,011879
0,017056	0,028046	0,013994	0,021549	0,011874	0,018187	0,008502	0,013554	0,007503	0,011875
0,017051	0,028032	0,013996	0,021474	0,011871	0,018244	0,008503	0,013483	0,007504	0,01188
0,017052	0,02806	0,01399	0,021505	0,011873	0,018229	0,008501	0,013529	0,007506	0,011872
0,017059	0,028078	0,013998	0,021545	0,011879	0,0182	0,00851	0,013495	0,007507	0,011875
0,01706	0,028047	0,013999	0,021508	0,011873	0,018259	0,008501	0,013565	0,007503	0,011875
0,017052	0,027982	0,013996	0,021518	0,011871	0,018252	0,008506	0,013563	0,007502	0,01188
0,017051	0,028015	0,013996	0,021533	0,011878	0,018182	0,008503	0,013541	0,007506	0,011877
0,017057	0,028047	0,013998	0,021532	0,011878	0,018229	0,008503	0,013505	0,007504	0,011871
0,017057	0,027998	0,014	0,021555	0,011879	0,018212	0,008509	0,013479	0,0075	0,011871
0,01705	0,028044	0,013995	0,021491	0,011878	0,018182	0,008502	0,013481	0,007502	0,011879
0,017052	0,028041	0,013998	0,021519	0,011879	0,018202	0,008502	0,013474	0,007505	0,011878
0,017053	0,028046	0,013992	0,021492	0,011874	0,01824	0,008502	0,013544	0,007509	0,01188
0,017058	0,02805	0,013993	0,021475	0,011872	0,018258	0,008507	0,013473	0,007509	0,011872
0,017055	0,028069	0,013999	0,021479	0,011875	0,018184	0,008505	0,01354	0,007507	0,01188
0,017052	0,028045	0,013995	0,021568	0,011877	0,018228	0,008506	0,013552	0,00751	0,011872
0,017056	0,028073	0,01399	0,02155	0,011872	0,018212	0,008507	0,013488	0,007509	0,011876
0,017053	0,028072	0,013995	0,021521	0,011878	0,018181	0,008501	0,013482	0,007508	0,011878
0,017055	0,028046	0,013998	0,021507	0,011874	0,018263	0,008505	0,01349	0,007506	0,011876
0,017059	0,028073	0,013999	0,021476	0,011872	0,018203	0,008509	0,013475	0,007506	0,011874
0,017057	0,02804	0,013992	0,021498	0,011878	0,018246	0,008504	0,013554	0,007507	0,011874
0,017055	0,02806	0,013993	0,021518	0,011872	0,018192	0,008501	0,013562	0,007503	0,011872
0,017053	0,028007	0,013997	0,021559	0,011871	0,018213	0,008508	0,013533	0,007509	0,011878
0,017056	0,02807	0,013996	0,021525	0,011879	0,018258	0,008508	0,0135	0,007502	0,011877
0,017052	0,02807	0,013992	0,021547	0,011879	0,018193	0,008502	0,013542	0,007502	0,01187
0,017051	0,027982	0,013992	0,021534	0,011875	0,018262	0,008506	0,013507	0,007507	0,011872
0,01706	0,028076	0,013991	0,021508	0,011879	0,018191	0,008506	0,013569	0,007508	0,011872

Продовження таблиці В.3

0,017057	0,028012	0,013998	0,021518	0,011878	0,018238	0,008508	0,01352	0,007506	0,011875
0,017051	0,028039	0,014	0,021506	0,01187	0,01818	0,008504	0,013481	0,007504	0,011873
0,017057	0,028008	0,013998	0,021474	0,011871	0,018269	0,008508	0,013558	0,007504	0,011875
0,017053	0,028048	0,013997	0,021478	0,011876	0,018256	0,008502	0,013495	0,007505	0,011878
0,017053	0,028079	0,013995	0,021474	0,011871	0,018263	0,008505	0,013479	0,007502	0,011872
0,017059	0,028001	0,013996	0,021483	0,011878	0,018179	0,008503	0,013485	0,007506	0,011877
0,017055	0,028012	0,01399	0,0215	0,011871	0,01824	0,008509	0,013528	0,00751	0,011878
0,017056	0,028062	0,013996	0,021535	0,011874	0,018193	0,008506	0,013499	0,007508	0,011875
0,017057	0,028032	0,013993	0,021491	0,011873	0,018254	0,008507	0,013506	0,007501	0,011875
0,017059	0,028061	0,013994	0,021503	0,011878	0,018193	0,00851	0,013536	0,007506	0,011871
0,017059	0,028006	0,013993	0,021504	0,011876	0,018192	0,008507	0,013515	0,007508	0,011872
0,017051	0,028062	0,013993	0,021501	0,011873	0,018188	0,008509	0,013546	0,007503	0,011873
0,017053	0,028069	0,013994	0,021509	0,011876	0,018193	0,008508	0,013488	0,007507	0,011874
0,017058	0,028045	0,013992	0,021492	0,011876	0,018172	0,008509	0,013536	0,007503	0,011879
0,017055	0,027988	0,013993	0,021486	0,011873	0,01826	0,008508	0,013483	0,00751	0,011873
0,017052	0,027993	0,013996	0,021531	0,011872	0,018185	0,008509	0,013488	0,007501	0,011873
0,017054	0,028075	0,013996	0,021528	0,011879	0,018213	0,008506	0,013557	0,0075	0,011874
0,017055	0,027981	0,013993	0,021561	0,011877	0,018212	0,008507	0,013569	0,007505	0,011879
0,017052	0,027993	0,013995	0,021472	0,011875	0,018269	0,008501	0,013562	0,007506	0,011876
0,017051	0,028009	0,013991	0,02157	0,011873	0,018259	0,008504	0,013569	0,007502	0,011879
0,017059	0,028048	0,013999	0,021556	0,011878	0,018259	0,008502	0,013545	0,007507	0,011876
0,017059	0,028018	0,013991	0,021509	0,011878	0,018214	0,008505	0,013551	0,007507	0,011878
0,017057	0,028029	0,013993	0,021529	0,011879	0,01826	0,008509	0,013479	0,007505	0,01187
0,017059	0,028047	0,013996	0,021518	0,011878	0,018246	0,008501	0,013563	0,007508	0,011873
0,017057	0,028062	0,013998	0,021532	0,01188	0,018233	0,008501	0,013557	0,007504	0,011875
0,017051	0,027999	0,013992	0,02148	0,011878	0,01823	0,0085	0,013476	0,007502	0,011879
0,017059	0,028062	0,013998	0,021544	0,011875	0,018244	0,008503	0,01351	0,007501	0,011874
0,017056	0,028063	0,01399	0,021481	0,011875	0,018197	0,008503	0,013567	0,007504	0,011873
0,017058	0,028026	0,013992	0,021486	0,011872	0,01821	0,008501	0,013486	0,007508	0,011874
0,017059	0,028035	0,013998	0,021526	0,011871	0,018177	0,008508	0,013482	0,007504	0,011872
0,017058	0,028074	0,013993	0,021489	0,011872	0,018246	0,008508	0,013478	0,007508	0,01188
0,017055	0,028075	0,013999	0,021486	0,011873	0,018208	0,008507	0,013505	0,007507	0,011877
0,017057	0,028062	0,013995	0,021512	0,011874	0,018249	0,008508	0,013562	0,007502	0,011878
0,017052	0,027994	0,013992	0,021495	0,011876	0,018234	0,0085	0,013533	0,007506	0,011873
0,017057	0,028078	0,013992	0,021537	0,011878	0,018186	0,008509	0,013538	0,007506	0,011879
0,017057	0,028035	0,013992	0,021493	0,011876	0,018177	0,008504	0,013557	0,007506	0,011877
0,017051	0,028079	0,013998	0,021555	0,011871	0,018231	0,008503	0,013479	0,0075	0,011874
0,017055	0,02804	0,013995	0,021525	0,011874	0,018204	0,008508	0,013477	0,007507	0,011874
0,017053	0,028068	0,013992	0,021507	0,011877	0,018191	0,00851	0,013512	0,007501	0,011871
0,017054	0,02801	0,014	0,021545	0,011878	0,018248	0,008507	0,013476	0,007508	0,011874
0,017058	0,028067	0,013999	0,021508	0,01188	0,018237	0,008503	0,013504	0,007501	0,011871
0,017058	0,028018	0,013994	0,021552	0,011877	0,018203	0,008502	0,013551	0,007501	0,011879
0,017057	0,028017	0,014	0,021541	0,011875	0,018189	0,00851	0,013495	0,007506	0,011872
0,017055	0,02799	0,013999	0,021533	0,011878	0,018245	0,008505	0,013493	0,007507	0,011871
0,017052	0,028021	0,013993	0,021561	0,011877	0,018182	0,008508	0,0135	0,007509	0,011876
0,01705	0,028056	0,013992	0,021488	0,011876	0,018233	0,008505	0,013523	0,0075	0,01187

Продовження таблиці В.3

0,017056	0,027995	0,013994	0,021498	0,011879	0,018267	0,008506	0,013495	0,007503	0,011878
0,01705	0,027999	0,013992	0,021508	0,011875	0,018193	0,008509	0,01352	0,007508	0,011872
0,017059	0,028025	0,013994	0,021472	0,011872	0,01824	0,008501	0,013498	0,007501	0,011879
0,01706	0,02804	0,01399	0,021558	0,011876	0,018183	0,008509	0,013501	0,007505	0,011879

- Inv – удосконалений метод мультиплікативного інвертування;
- Inv_p - прототип.

Таблиця В.4

t-критерій Стьюдента для мультиплікативного інвертування з 99.5% довірчим інтервалом

Поле		**Критерій рівності дисперсій Лівіня**		**t-критерій для рівності середніх**						
		F	**Знач.**	**т**	**ст.св.**	**Знач. (2-х сторн.)**	**Різниця середніх**	**Серед. кв. похиб. різн.**	**Нижн.**	**Верх.**
431	D	284,675	,000	-4162,584	238	0,000	-,010973750	,000002636	-,010978943	-,010968557
	ND			-4162,584	121,277	0,000	-,010973750	,000002636	-,010978969	-,010968531
367	D	270,032	,000	-2760,584	238	0,000	-,007526267	,000002726	-,007531637	-,007520896
	ND			-2760,584	121,257	,000	-,007526267	,000002726	-,007531664	-,007520869
307	D	276,581	,000	-2335,182	238	0,000	-,006343783	,000002717	-,006349135	-,006338432
	ND			-2335,182	121,245	,000	-,006343783	,000002717	-,006349161	-,006338405
257	D	265,809	,000	-2769,586	238	0,000	-,007015458	,000002533	-,007020448	-,007010468
	ND			-2769,586	121,374	,000	-,007015458	,000002533	-,007020473	-,007010444
233	D	,255	,614	-11480,764	238	0,000	-,004370900	,000000381	-,004371650	-,004370150
	ND			-11480,764	237,601	0,000	-,004370900	,000000381	-,004371650	-,004370150

- D - передбачається рівність дисперсій;
- ND - не передбачається рівність дисперсій.

Таблиця В.5

Вибірка середнього часу для удосконаленого методу здобуття кубічного кореня та прототип в мс для полів з ДСТУ 4145-2002:431, 367, 307, 257, 233

431		367		307		257		233	
cub	**cub_p**	**cub**	**cub_p**	**cub**	**cub_p**	**cub**	**cub_p**	**cub**	**cub_p**
0,053	0,1962	0,0368	0,1302	0,0284	0,0803	0,0203	0,0613	0,0183	0,0479
0,0524	0,1955	0,0363	0,1304	0,0288	0,0808	0,0202	0,0608	0,0176	0,0479
0,0525	0,1956	0,0366	0,1307	0,0285	0,0813	0,0205	0,0605	0,0176	0,0482
0,0529	0,1961	0,0363	0,1304	0,0286	0,081	0,0206	0,0614	0,0179	0,0482
0,0526	0,1952	0,0363	0,1305	0,0284	0,081	0,0203	0,0606	0,0177	0,0492
0,0525	0,1962	0,0364	0,131	0,0284	0,0809	0,0202	0,0606	0,0178	0,048
0,0531	0,1964	0,0369	0,1309	0,028	0,0805	0,0201	0,0605	0,0176	0,0492
0,0523	0,1966	0,0365	0,1308	0,0288	0,0811	0,0205	0,0612	0,0179	0,048
0,0525	0,1962	0,0368	0,1311	0,0289	0,0802	0,0208	0,0607	0,0182	0,0482
0,0532	0,1964	0,036	0,131	0,0283	0,08	0,02	0,0602	0,0182	0,0482
0,0529	0,1958	0,0362	0,1304	0,0283	0,0812	0,0201	0,0601	0,018	0,0478
0,0522	0,196	0,0368	0,1312	0,0281	0,0813	0,0204	0,0605	0,0184	0,0478
0,0526	0,196	0,0362	0,1305	0,0282	0,0803	0,0202	0,06	0,0175	0,0481
0,0526	0,1956	0,036	0,1305	0,0283	0,0802	0,0204	0,0612	0,0182	0,0488
0,0531	0,1956	0,0365	0,1303	0,0288	0,0811	0,0201	0,0601	0,0176	0,0487
0,0527	0,1957	0,0365	0,1311	0,0281	0,0805	0,0201	0,0606	0,0182	0,0489
0,0525	0,1964	0,0364	0,1301	0,0283	0,0812	0,0199	0,0611	0,0177	0,0485
0,053	0,1953	0,0361	0,13	0,0283	0,0801	0,0199	0,0602	0,0176	0,0491
0,0531	0,1954	0,036	0,1306	0,0282	0,0804	0,0203	0,061	0,0181	0,0479
0,0526	0,1966	0,0367	0,1302	0,0283	0,081	0,0201	0,06	0,0181	0,0488
0,0531	0,1962	0,036	0,1305	0,028	0,0803	0,0208	0,0602	0,0182	0,049
0,053	0,1953	0,0364	0,1301	0,0282	0,0814	0,0201	0,0609	0,0178	0,0482
0,0524	0,1953	0,0362	0,1308	0,0285	0,0808	0,0208	0,0612	0,0177	0,0491
0,0531	0,1953	0,0363	0,1309	0,0283	0,08	0,0209	0,0605	0,0179	0,0482
0,0526	0,1956	0,0363	0,1309	0,0283	0,0807	0,0203	0,0603	0,0177	0,0482
0,053	0,1963	0,0368	0,1313	0,028	0,0812	0,0202	0,0608	0,0185	0,0488
0,0528	0,1955	0,0362	0,1313	0,0285	0,0805	0,0205	0,0607	0,0178	0,0488
0,0524	0,1953	0,0366	0,1304	0,0289	0,081	0,0203	0,0614	0,0182	0,0478
0,0524	0,1965	0,0367	0,1309	0,0283	0,0806	0,0207	0,06	0,0178	0,0483
0,0523	0,1955	0,0365	0,1305	0,0279	0,0802	0,0199	0,0601	0,0182	0,0484
0,053	0,1966	0,0366	0,131	0,028	0,0804	0,0205	0,0603	0,0179	0,049
0,053	0,1955	0,0363	0,1306	0,0288	0,0812	0,0202	0,0605	0,0177	0,0491
0,0529	0,1955	0,0368	0,1314	0,0286	0,0803	0,0199	0,0606	0,0177	0,0482
0,0522	0,1953	0,0362	0,1308	0,0289	0,0802	0,0204	0,0605	0,0177	0,0486
0,0525	0,1961	0,0367	0,1306	0,0282	0,0808	0,0208	0,0612	0,0177	0,048
0,0532	0,1965	0,036	0,1308	0,0283	0,0804	0,0202	0,0608	0,0181	0,0483
0,0522	0,1963	0,0364	0,1301	0,0285	0,0806	0,0204	0,0603	0,0178	0,0483
0,0526	0,1965	0,0361	0,1311	0,0283	0,0807	0,0201	0,061	0,018	0,049
0,0524	0,1964	0,0369	0,1306	0,0281	0,0804	0,0206	0,06	0,0183	0,0487
0,0529	0,1954	0,0367	0,1303	0,028	0,0803	0,0201	0,0612	0,018	0,049
0,0531	0,1964	0,0367	0,1303	0,0286	0,0801	0,0202	0,0608	0,0177	0,048

Продовження таблиці В.5

0,0524	0,1956	0,0359	0,1307	0,028	0,0806	0,0203	0,0603	0,0177	0,0482
0,0523	0,1961	0,0363	0,1306	0,0287	0,0811	0,0205	0,0603	0,0177	0,0481
0,0523	0,1965	0,036	0,1309	0,0281	0,0811	0,0203	0,0602	0,0175	0,0486
0,0527	0,1965	0,0368	0,1311	0,028	0,0801	0,0208	0,0608	0,0185	0,0489
0,0523	0,196	0,0362	0,1304	0,0284	0,0808	0,0206	0,0606	0,0177	0,0484
0,0531	0,1954	0,0361	0,1313	0,0285	0,0809	0,0201	0,0609	0,0183	0,0478
0,0527	0,1957	0,0367	0,1306	0,028	0,0806	0,0204	0,0607	0,0179	0,0482
0,0526	0,1959	0,0362	0,1307	0,0287	0,0805	0,0206	0,0609	0,0184	0,048
0,0532	0,1961	0,036	0,1309	0,0283	0,08	0,0204	0,0614	0,018	0,0488
0,0524	0,1964	0,0368	0,1313	0,0281	0,0803	0,0208	0,0604	0,0184	0,0483
0,0523	0,1963	0,0363	0,1311	0,0288	0,0809	0,0203	0,0609	0,0175	0,0487
0,0529	0,1964	0,0363	0,1308	0,0283	0,0802	0,02	0,0611	0,0179	0,0485
0,0522	0,1961	0,0367	0,1304	0,0287	0,0807	0,0203	0,0601	0,0179	0,0483
0,0525	0,1965	0,036	0,1304	0,0286	0,0808	0,0208	0,0612	0,0177	0,0487
0,0529	0,1953	0,0363	0,1305	0,0284	0,08	0,0208	0,0608	0,0183	0,049
0,0525	0,1961	0,0359	0,1301	0,0287	0,0805	0,0207	0,06	0,0184	0,0487
0,0523	0,1965	0,0369	0,1313	0,0287	0,0807	0,0202	0,0601	0,018	0,0488
0,0528	0,1955	0,0362	0,1311	0,0286	0,081	0,0207	0,0606	0,0181	0,0491
0,0531	0,1956	0,0363	0,1309	0,0284	0,08	0,0207	0,0605	0,018	0,049
0,0529	0,1957	0,0364	0,1314	0,0285	0,0814	0,0209	0,06	0,0181	0,0486
0,0523	0,1962	0,0368	0,1313	0,0287	0,0801	0,0206	0,0609	0,0176	0,0485
0,0526	0,1955	0,0367	0,13	0,0283	0,0802	0,0207	0,0601	0,0177	0,0478
0,0525	0,196	0,0366	0,1311	0,0286	0,08	0,0208	0,0602	0,0182	0,0479
0,0528	0,1952	0,0363	0,1301	0,0287	0,0813	0,0202	0,061	0,0181	0,0482
0,0525	0,1957	0,0367	0,1305	0,0288	0,0804	0,0208	0,0611	0,0176	0,0487
0,0528	0,1961	0,0362	0,1313	0,0281	0,0805	0,02	0,0608	0,0179	0,0483
0,0523	0,1954	0,0364	0,1309	0,0285	0,081	0,0202	0,0614	0,0176	0,0486
0,0528	0,1952	0,036	0,1311	0,0281	0,0812	0,0205	0,0608	0,0178	0,0483
0,0525	0,196	0,0363	0,1306	0,0284	0,0803	0,0201	0,061	0,0184	0,0492
0,0522	0,1965	0,0364	0,1304	0,0283	0,0801	0,0203	0,0604	0,0175	0,0485
0,0526	0,1966	0,0368	0,1306	0,028	0,0811	0,0208	0,0605	0,0182	0,049
0,0523	0,1961	0,0365	0,1306	0,0282	0,0811	0,0199	0,0613	0,0177	0,048
0,0526	0,1958	0,0361	0,131	0,0281	0,081	0,0201	0,0602	0,0182	0,0491
0,053	0,1957	0,0365	0,1304	0,028	0,0811	0,0205	0,0611	0,0181	0,0479
0,0526	0,1961	0,0361	0,1301	0,0281	0,0808	0,0207	0,0603	0,0177	0,0489
0,0524	0,1954	0,0369	0,1308	0,0281	0,0811	0,0207	0,061	0,0182	0,049
0,0523	0,1959	0,0363	0,1311	0,0285	0,081	0,0208	0,0609	0,0178	0,0489
0,0529	0,1965	0,0359	0,1305	0,0285	0,0811	0,0206	0,0614	0,0179	0,0481
0,0526	0,1959	0,0359	0,1301	0,0282	0,0801	0,0207	0,0612	0,018	0,0487
0,0531	0,1966	0,0361	0,1308	0,0288	0,08	0,0201	0,0609	0,0178	0,0484
0,0526	0,1953	0,0367	0,1314	0,0281	0,0811	0,0207	0,0602	0,0184	0,0481
0,0522	0,1956	0,0363	0,1307	0,0283	0,0811	0,0201	0,0601	0,0181	0,0481
0,0525	0,1955	0,0361	0,1309	0,0283	0,0808	0,0205	0,0612	0,018	0,0486
0,0525	0,1957	0,0363	0,1305	0,0289	0,0812	0,0203	0,0611	0,0184	0,0486
0,053	0,1955	0,0368	0,1308	0,0281	0,0801	0,0201	0,0609	0,0176	0,0491

Продовження таблиці В.5

0,0525	0,1955	0,0368	0,1314	0,0283	0,0804	0,0202	0,0609	0,0181	0,0489
0,0523	0,1954	0,0364	0,1309	0,0288	0,081	0,0204	0,0601	0,0182	0,049
0,0528	0,196	0,036	0,1309	0,0285	0,0807	0,0202	0,0603	0,0185	0,0485
0,0527	0,1965	0,036	0,1307	0,0288	0,0812	0,0209	0,0607	0,0182	0,0485
0,0528	0,1956	0,0363	0,1301	0,0283	0,0812	0,0207	0,06	0,0178	0,0485
0,0526	0,1966	0,0362	0,1311	0,0282	0,0802	0,0201	0,061	0,0176	0,0489
0,0529	0,1956	0,0365	0,1302	0,0281	0,0807	0,0207	0,061	0,0179	0,0489
0,0527	0,1954	0,0362	0,1313	0,028	0,08	0,02	0,0608	0,0177	0,049
0,0531	0,1964	0,0367	0,1314	0,0288	0,0812	0,0208	0,0607	0,0181	0,0479
0,0527	0,1961	0,0365	0,13	0,0284	0,0803	0,0206	0,0602	0,018	0,0486
0,0528	0,1956	0,0359	0,1311	0,0284	0,0802	0,0206	0,0613	0,0184	0,048
0,0522	0,1954	0,0366	0,1304	0,0288	0,0807	0,0203	0,0606	0,0184	0,049
0,0531	0,1955	0,0362	0,1306	0,028	0,081	0,0201	0,0605	0,0184	0,0479
0,053	0,1955	0,0361	0,1304	0,0287	0,0805	0,0205	0,0601	0,0179	0,049
0,0527	0,1961	0,0367	0,1306	0,0289	0,0812	0,0205	0,0613	0,0179	0,0486
0,0526	0,1962	0,0364	0,1302	0,0288	0,081	0,0204	0,0602	0,0179	0,0483
0,0526	0,1956	0,0366	0,1311	0,0284	0,0811	0,0206	0,0602	0,0175	0,0487
0,0523	0,1953	0,0368	0,1312	0,0287	0,0803	0,0201	0,0613	0,0178	0,0488
0,0528	0,196	0,0362	0,1312	0,0283	0,0806	0,02	0,0606	0,0178	0,0488
0,0526	0,1958	0,0359	0,1311	0,0284	0,0812	0,0199	0,0608	0,0176	0,049
0,0525	0,1954	0,0362	0,1313	0,028	0,0814	0,0205	0,0614	0,0177	0,049
0,0522	0,1961	0,0365	0,1305	0,028	0,0813	0,0199	0,061	0,0181	0,0482
0,0529	0,1954	0,0363	0,1313	0,0284	0,0803	0,0202	0,0607	0,0175	0,0485
0,0531	0,1953	0,0364	0,131	0,0283	0,0804	0,0203	0,0602	0,0181	0,0487
0,0523	0,1954	0,0361	0,13	0,0287	0,0803	0,0204	0,0604	0,0183	0,049
0,0528	0,196	0,0361	0,131	0,0283	0,0811	0,02	0,0609	0,0179	0,0482
0,0526	0,1963	0,0361	0,1307	0,0284	0,0805	0,0208	0,061	0,0179	0,0491
0,0532	0,1955	0,0365	0,1301	0,0284	0,0809	0,0209	0,0607	0,0182	0,0486
0,0525	0,1955	0,0365	0,1311	0,0283	0,0803	0,0207	0,0609	0,0179	0,0491
0,0528	0,1953	0,036	0,1313	0,0288	0,0803	0,0206	0,0611	0,0182	0,0482
0,0523	0,1963	0,0363	0,1307	0,0285	0,0813	0,0206	0,0608	0,0175	0,048
0,0532	0,1955	0,036	0,1307	0,0285	0,0804	0,0201	0,0612	0,0183	0,0482
0,0525	0,1958	0,0366	0,1311	0,0285	0,0807	0,0202	0,0613	0,0177	0,0486
0,0523	0,1955	0,0363	0,1301	0,0281	0,08	0,0207	0,0601	0,0184	0,0482

- cub –удосконалений метод здобуття кубічного кореня;
- cub_p – прототип.

Таблиця В.6

Вибірка середнього часу для удосконаленого методу здобуття кубічного кореня та прототип в мс для полів з ДСТУ 4145-2002: 191, 179, 173, 167, 163

191		179		173		167		163	
cub	**cub_p**	**cub**	**cub_p**	**cub**	**cub_p**	**cub**	**cub_p**	**cub**	**cub_p**
0,0112	0,0303	0,0124	0,0284	0,0112	0,0271	0,0105	0,0212	0,0092	0,0207
0,0117	0,0304	0,0122	0,0289	0,0119	0,0283	0,0104	0,0213	0,0094	0,0201
0,0116	0,0297	0,0126	0,0287	0,0115	0,0282	0,0102	0,0223	0,0095	0,0202
0,0118	0,0297	0,0127	0,0283	0,0114	0,0281	0,0106	0,0219	0,0099	0,0208
0,0118	0,0308	0,012	0,0292	0,0118	0,0277	0,0105	0,0211	0,009	0,0212
0,0117	0,0303	0,012	0,0291	0,0113	0,0277	0,01	0,0219	0,0097	0,0208
0,0113	0,03	0,0129	0,0281	0,0115	0,0275	0,0109	0,0221	0,0097	0,0208
0,0113	0,0305	0,0121	0,0284	0,0119	0,0276	0,0102	0,0219	0,0093	0,0208
0,0114	0,0295	0,0122	0,0286	0,0112	0,0282	0,0106	0,021	0,0099	0,0205
0,0116	0,0297	0,0127	0,0288	0,011	0,0271	0,0108	0,022	0,0095	0,0209
0,0116	0,0309	0,0123	0,0287	0,0115	0,028	0,01	0,0221	0,0091	0,0205
0,0112	0,0301	0,0125	0,0289	0,012	0,028	0,0103	0,0216	0,009	0,0207
0,0111	0,0305	0,0128	0,0288	0,0117	0,0271	0,0101	0,0214	0,0091	0,0203
0,0116	0,0299	0,0124	0,029	0,0119	0,0284	0,0101	0,021	0,0095	0,0203
0,0118	0,0309	0,0124	0,0281	0,012	0,0278	0,0103	0,0216	0,0092	0,0214
0,0118	0,0299	0,0125	0,0284	0,0113	0,0271	0,0108	0,0219	0,009	0,0201
0,0114	0,0304	0,012	0,0291	0,0114	0,0279	0,0106	0,0214	0,0099	0,0214
0,0118	0,0305	0,0121	0,0293	0,0111	0,0278	0,01	0,0214	0,0095	0,02
0,0109	0,0303	0,0126	0,0294	0,0113	0,0278	0,01	0,0214	0,0098	0,0207
0,0109	0,0298	0,0119	0,0289	0,0115	0,0284	0,0104	0,0221	0,0099	0,0212
0,0116	0,0308	0,0124	0,0288	0,012	0,0278	0,0102	0,0216	0,0094	0,0207
0,0117	0,0301	0,0121	0,0284	0,0115	0,028	0,0105	0,0217	0,0097	0,0206
0,0114	0,0302	0,0128	0,0281	0,0112	0,0276	0,0099	0,0214	0,0099	0,0204
0,011	0,0298	0,0128	0,0284	0,0114	0,027	0,0101	0,0219	0,0098	0,0208
0,0118	0,0308	0,0126	0,0283	0,0111	0,0284	0,0106	0,022	0,0096	0,0206
0,0111	0,0302	0,0126	0,0283	0,0116	0,0275	0,0103	0,021	0,0095	0,0203
0,0117	0,0299	0,012	0,0291	0,0115	0,0273	0,0103	0,0219	0,01	0,0208
0,0118	0,0303	0,0129	0,0285	0,0115	0,0283	0,0099	0,0215	0,0098	0,0211
0,0112	0,03	0,0125	0,029	0,0115	0,0273	0,0105	0,0219	0,0096	0,0206
0,0114	0,0305	0,0126	0,029	0,0114	0,0278	0,0103	0,0214	0,0091	0,0206
0,0118	0,0302	0,0127	0,029	0,0112	0,0276	0,0106	0,0219	0,0097	0,0207
0,011	0,0297	0,0122	0,0283	0,0117	0,0278	0,0104	0,0224	0,0099	0,0201
0,0112	0,0309	0,0119	0,0286	0,0114	0,0278	0,0101	0,0215	0,0098	0,0209
0,0109	0,0299	0,0128	0,0283	0,0113	0,0274	0,0102	0,0222	0,0095	0,0214
0,0115	0,0299	0,0122	0,0294	0,0115	0,028	0,0105	0,0216	0,0093	0,0211
0,011	0,0309	0,0126	0,0283	0,0114	0,0279	0,0099	0,0222	0,0095	0,0202
0,0115	0,0305	0,0126	0,0293	0,0112	0,0276	0,0104	0,0211	0,0092	0,0206
0,0111	0,0301	0,0126	0,0287	0,0114	0,0277	0,0102	0,0214	0,0095	0,0204
0,0117	0,0308	0,0126	0,0289	0,0114	0,0275	0,0103	0,0212	0,0099	0,0213
0,0116	0,0296	0,0125	0,0284	0,0119	0,0281	0,0107	0,0224	0,009	0,0206

Продовження таблиці В.6

0,011	0,0297	0,0124	0,0283	0,0118	0,0283	0,0105	0,0212	0,0096	0,0202
0,0109	0,0307	0,0123	0,0287	0,0112	0,0279	0,0107	0,0211	0,0094	0,0214
0,011	0,0307	0,012	0,0286	0,0114	0,0275	0,0106	0,0217	0,0098	0,0213
0,011	0,0297	0,0122	0,0292	0,0112	0,0272	0,0105	0,0224	0,0094	0,0205
0,0112	0,0306	0,0128	0,0288	0,0117	0,0277	0,01	0,0217	0,0099	0,0203
0,0111	0,0307	0,0129	0,028	0,0117	0,0271	0,0109	0,0215	0,0094	0,0201
0,0116	0,0297	0,012	0,0289	0,0111	0,0283	0,0107	0,0219	0,0097	0,0207
0,0115	0,0303	0,0125	0,0287	0,0115	0,0281	0,0104	0,0222	0,0094	0,0211
0,0111	0,0308	0,012	0,0288	0,0119	0,0277	0,0109	0,0213	0,0096	0,021
0,011	0,0304	0,0127	0,0293	0,0119	0,0281	0,0104	0,0214	0,0099	0,0207
0,0116	0,0298	0,0122	0,0283	0,0117	0,0284	0,0105	0,0218	0,0096	0,0203
0,0113	0,0298	0,0121	0,0289	0,0117	0,0276	0,0099	0,0215	0,009	0,0201
0,0114	0,0297	0,0126	0,0282	0,0116	0,0278	0,0104	0,0217	0,0092	0,0212
0,0115	0,0307	0,0121	0,0287	0,012	0,0272	0,0108	0,0213	0,0099	0,0211
0,011	0,0309	0,0123	0,0283	0,0115	0,027	0,01	0,0219	0,0098	0,0213
0,0116	0,0296	0,0123	0,0292	0,0119	0,0278	0,0101	0,0213	0,0095	0,0211
0,0119	0,0296	0,0119	0,0289	0,0113	0,0276	0,0099	0,022	0,0092	0,0203
0,0112	0,0303	0,0122	0,0281	0,0113	0,0278	0,0099	0,0212	0,0093	0,0202
0,0117	0,0308	0,0126	0,0288	0,0116	0,027	0,0099	0,0217	0,0098	0,0206
0,0119	0,0302	0,0124	0,0288	0,0119	0,0275	0,0099	0,0224	0,009	0,0202
0,011	0,0308	0,0123	0,0286	0,0111	0,0274	0,01	0,0217	0,0095	0,0212
0,0116	0,0307	0,0122	0,0291	0,0115	0,0283	0,0109	0,0214	0,0091	0,0205
0,0111	0,0307	0,0125	0,0284	0,0116	0,0279	0,0106	0,0219	0,0096	0,0207
0,0112	0,0306	0,0121	0,0287	0,0114	0,0273	0,0102	0,022	0,0095	0,0203
0,0115	0,0297	0,0123	0,0285	0,0114	0,0279	0,0106	0,0217	0,0099	0,0214
0,0118	0,03	0,0122	0,0283	0,0117	0,0272	0,0102	0,022	0,0092	0,0209
0,011	0,0305	0,0126	0,0289	0,011	0,028	0,0107	0,0218	0,0097	0,0202
0,0114	0,0303	0,0119	0,0281	0,0116	0,0272	0,0102	0,022	0,0096	0,021
0,011	0,0302	0,012	0,0282	0,0117	0,0274	0,01	0,0211	0,0096	0,0203
0,0114	0,0298	0,0122	0,0284	0,0111	0,0273	0,0102	0,0217	0,009	0,0203
0,0114	0,0305	0,0124	0,0284	0,012	0,0281	0,0108	0,0223	0,0097	0,0212
0,0112	0,0302	0,0126	0,0285	0,0112	0,0282	0,0107	0,0219	0,0091	0,0209
0,0116	0,0298	0,0127	0,0284	0,0119	0,0278	0,0106	0,0222	0,009	0,0201
0,0109	0,0305	0,0128	0,0288	0,011	0,027	0,0103	0,0212	0,0095	0,02
0,011	0,0304	0,0126	0,0285	0,0117	0,028	0,0106	0,0212	0,0092	0,0201
0,0117	0,0307	0,0122	0,0281	0,0118	0,0282	0,0102	0,0218	0,0095	0,02
0,0113	0,0306	0,0121	0,0289	0,0117	0,0278	0,0107	0,0215	0,0091	0,0201
0,0111	0,0296	0,0124	0,0294	0,0112	0,0284	0,01	0,0224	0,009	0,0214
0,0119	0,0308	0,0123	0,0288	0,011	0,028	0,0103	0,0213	0,0092	0,0205
0,0109	0,0296	0,0123	0,0287	0,0116	0,027	0,0102	0,0212	0,0093	0,0201
0,0116	0,0305	0,0123	0,0294	0,0112	0,0274	0,0109	0,0211	0,0091	0,0213
0,011	0,0309	0,0124	0,0293	0,0116	0,028	0,01	0,0222	0,0099	0,0201
0,0118	0,0301	0,0126	0,0293	0,0117	0,0278	0,0103	0,0219	0,01	0,0203
0,0117	0,03	0,012	0,0293	0,0111	0,0278	0,0108	0,0216	0,0094	0,021

Продовження таблиці В.6

0,0114	0,03	0,0128	0,0287	0,0115	0,0277	0,0107	0,0224	0,0092	0,0202
0,0117	0,0309	0,0121	0,029	0,0117	0,0273	0,0101	0,0214	0,0091	0,0203
0,0112	0,0307	0,0127	0,0282	0,011	0,0276	0,0102	0,0219	0,0091	0,0201
0,0112	0,0307	0,0129	0,0294	0,0116	0,0275	0,0108	0,022	0,0095	0,0214
0,0117	0,03	0,0124	0,0284	0,0113	0,0277	0,0106	0,0218	0,0094	0,021
0,0116	0,0303	0,0121	0,028	0,0117	0,0272	0,0103	0,0223	0,0092	0,0214
0,011	0,0296	0,0127	0,0281	0,0115	0,0284	0,0101	0,0222	0,0094	0,0205
0,0111	0,0305	0,0128	0,0293	0,0118	0,0277	0,01	0,022	0,0094	0,0213
0,0115	0,0297	0,0123	0,0284	0,0118	0,0274	0,0108	0,0222	0,0095	0,0201
0,0111	0,0309	0,0125	0,0292	0,0116	0,0283	0,0105	0,0212	0,0097	0,0207
0,0111	0,0307	0,0125	0,029	0,0115	0,0273	0,0108	0,0218	0,009	0,0201
0,0115	0,0299	0,0123	0,0287	0,0117	0,0273	0,0101	0,0212	0,0097	0,0202
0,011	0,0309	0,0121	0,0288	0,0119	0,0271	0,0108	0,0215	0,0091	0,0206
0,0113	0,0304	0,0123	0,0281	0,0113	0,0272	0,0101	0,0213	0,0092	0,0202
0,0113	0,0304	0,0126	0,0282	0,0114	0,0282	0,0106	0,0217	0,0092	0,0211
0,0111	0,0308	0,0127	0,0286	0,0118	0,0279	0,0103	0,0217	0,0091	0,0207
0,0111	0,0301	0,0124	0,0286	0,0117	0,0274	0,0102	0,0221	0,0093	0,02
0,0109	0,0304	0,0119	0,0283	0,0115	0,0271	0,0105	0,0213	0,0095	0,0205
0,0118	0,0302	0,0128	0,0288	0,0113	0,027	0,0101	0,0218	0,0097	0,0204
0,0114	0,0301	0,0123	0,0286	0,0113	0,0271	0,0106	0,0223	0,0098	0,02
0,0114	0,0295	0,0129	0,0281	0,0117	0,0281	0,0108	0,0224	0,0098	0,0203
0,0115	0,0309	0,0126	0,0292	0,0112	0,0284	0,0109	0,0224	0,0096	0,0203
0,0116	0,03	0,0125	0,0291	0,0116	0,028	0,0106	0,0221	0,0092	0,0206
0,0113	0,0308	0,012	0,0292	0,011	0,0284	0,0105	0,0222	0,0098	0,021
0,0113	0,0304	0,0128	0,0291	0,0112	0,0278	0,0101	0,022	0,0098	0,021
0,0117	0,03	0,0121	0,0282	0,012	0,0278	0,0101	0,0224	0,0095	0,0204
0,0109	0,0295	0,0122	0,0293	0,0118	0,0283	0,01	0,0216	0,0099	0,0213
0,011	0,0303	0,0126	0,0294	0,0115	0,0282	0,0105	0,021	0,01	0,0203
0,0115	0,0296	0,0123	0,0293	0,0117	0,0273	0,0102	0,0212	0,0092	0,0202
0,0116	0,0308	0,0121	0,0286	0,0117	0,0275	0,0102	0,0215	0,0097	0,0214
0,0115	0,0309	0,0126	0,0281	0,0118	0,0278	0,0106	0,0224	0,0091	0,0209
0,0118	0,0307	0,0127	0,0291	0,0113	0,0283	0,01	0,021	0,0099	0,0202
0,0118	0,03	0,0125	0,0287	0,0115	0,0278	0,0104	0,021	0,0093	0,0202
0,0115	0,0307	0,0125	0,0281	0,0114	0,0272	0,0103	0,0216	0,0093	0,0205
0,0112	0,0306	0,0122	0,0292	0,0115	0,0271	0,0102	0,0223	0,0092	0,0212
0,0118	0,0299	0,0121	0,0292	0,0111	0,0274	0,0101	0,0221	0,0098	0,0201

- cub –удосконалений метод здобуття кубічного кореня;
- cub_p – прототип.

Таблиця В.7

t-критерій Стьюдента для мультиплікативного інвертування з 99.5% довірчим інтервалом

Поле		Критерій рівності дисперсій Лівіня		t-критерій для рівності середніх						
		F	Знач.	т	ст.св.	Знач. (2-х сторонняя)	Різниця середніх	Серед. кв. похибка різниці	Нижня	Верхня
431	D	37,255	,000	-2992,479	238	0,000	-,1432033	,0000479	-,1432976	-,1431091
	ND			-2992,479	210,211	0,000	-,1432033	,0000479	-,1432977	-,1431090
367	D	18,129	,000	-2106,662	238	0,000	-,0943617	,0000448	-,0944499	-,0942734
	ND			-2106,662	214,857	0,000	-,0943617	,0000448	-,0944500	-,0942734
307	D	43,301	,000	-1137,387	238	0,000	-,0522767	,0000460	-,0523672	-,0521861
	ND			-1137,387	203,327	0,000	-,0522767	,0000460	-,0523673	-,0521860
257	D	22,659	,000	-861,592	238	0,000	-,0402883	,0000468	-,0403805	-,0401962
	ND			-861,592	210,569	0,000	-,0402883	,0000468	-,0403805	-,0401962
233	D	27,151	,000	-675,121	238	0,000	-,0305625	,0000453	-,0306517	-,0304733
	ND			-675,121	210,448	0,000	-,0305625	,0000453	-,0306517	-,0304733
191	D	22,003	,000	-393,749	238	0,000	-,0188975	,0000480	-,0189920	-,0188030
	ND			-393,749	212,889	,000	-,0188975	,0000480	-,0189921	-,0188029
179	D	20,617	,000	-363,319	238	0,000	-,0163108	,0000449	-,0163993	-,0162224
	ND			-363,319	210,688	,000	-,0163108	,0000449	-,0163993	-,0162223
173	D	25,178	,000	-355,975	238	0,000	-,0161992	,0000455	-,0162888	-,0161095
	ND			-355,975	205,444	,000	-,0161992	,0000455	-,0162889	-,0161094

Продовження таблиці В.7

167	D	18,019	,000	-243,846	238	,000	-,0113550	,0000466	-,0114467	-,0112633
	ND			-243,846	214,128	,000	-,0113550	,0000466	-,0114468	-,0112632
163	D	23,461	,000	-230,592	238	,000	-,0111533	,0000484	-,0112486	-,0110580
	ND			-230,592	210,609	,000	-,0111533	,0000484	-,0112487	-,0110580

- D - передбачається рівність дисперсій;

- ND - не передбачається рівність дисперсій.

ДОДАТОК Д. Результати експерементальних оцінок

Таблиця Д.1

Порівняння швидкодії реалізації операції СМ в проективних координатах для ДСТУ 4145-2002 (Intel Xeon X5670 2,93 GHz, Linux CentOS v7.0 x86-64) використовуючи спеціалізований набір інструкцій

Поле, m	**Час реалізації СМ, мс**				
	Біраціонально еквівалентна крива Вейерштрасса до БЕК 251	**Криві Вейерштрасса з ДСТУ4145-2002**	**Біраціональна крива Едвардса для кривих з ДСТУ4145-2002**		**БЕК 251**
			$d_1 \neq d_2$	$d_1 = d_2$	$d_1 = d_2$
	Алгоритм Монтгомері (проективні координати Лопеса-Дахаба)		**Алгоритм Монтгомері (проективні w-координати)**		
163		0.262	0.486		
167		0.267	0.501		
173		0.284	0.526	0.302	
179		0.292	0.549		
191		0.312	0.591		
233		0.539	0.972		
251	0.699				0.670
257		0.718	1.183	0.681	
307		1.001	1.638		
367		1.353	2.500		
431		2.156	3.676		

Таблиця Д.2

Порівняння швидкодії реалізації операції СМ в проективних координатах для ДСТУ 4145-2002 (Intel Xeon E5-2640 2,50 GHz, Linux CentOS v7.0 x86-64)

Поле, m	**Час реалізації СМ, мс**				
	Біраціонально еквівалентна крива Вейерштрасса до БЕК 251	**Криві Вейерштрасса з ДСТУ4145-2002**	**Біраціональна крива Едвардса для кривих з ДСТУ4145-2002**		**БЕК 251**
			$d_1 \neq d_2$	$d_1 = d_2$	$d_1 = d_2$
	Алгоритм Монтгомері (проективні координати Лопеса-Дахаба)		**Алгоритм Монтгомері (проективні w-координати)**		
163		0.285	0.549		
163#		0.290	0.552		
167		0.286	0.553		
167#		0.290	0.554		
173		0.309	0.599	0.329	
173#		0.300	0.609	0.331	
179		0.326	0.631		
179#		0.323	0.627		

Продовження таблиці Д.2

191		0.340	0.659		
191#		0.348	0.680		
233		0.574	1.058		
233#		0.573	1.191		
251	0.815				0.745
251#	0.801				0.741
257		0.825	1.520	0.785	
257#		1.194	1.711	1.095	
307		1.107	2.040		
307#		1.101	2.116		
367		1.699	3.126		
367#		1.709	3.418		
431		3.096	4.772		
431#		2.567	4.839		

- # - спеціалізований набір інструкцій процесора.

Таблиця Д.3

Порівняння швидкодії реалізації операції СМ в проективних координатах для ДСТУ 4145-2002 (Intel Core i7-4702MQ 3,20 GHz, SentOS Linux Ubuntu 16.04)

Поле, m	**Час реалізації СМ, мс**				
	Біраціонально еквівалентна крива Вейерштрасса до БЕК 251	**Криві Вейерштрасса з ДСТУ4145-2002**	**Біраціональна крива Едвардса для кривих з ДСТУ4145-2002**		**БЕК 251**
			$d_1 \neq d_2$	$d_1 = d_2$	$d_1 = d_2$
	Алгоритм Монтгомері (проективні координати Лопеса-Дахаба)		**Алгоритм Монтгомері (проективні w-координати)**		
163		0.203	0.391		
163#		0.203	0.391		
167		0.203	0.391		
167#		0.203	0.406		
173		0.219	0.438	0.234	
173#		0.219	0.422	0.224	
179		0.219	0.453		
179#		0.219	0.438		
191		0.234	0.500		
191#		0.234	0.469		
233		0.375	0.766		

Продовження таблиці Д.3

233#		0.375	0.766		
251	0.536				0.516
251#	0.545				0.500
257		0.547	1.063	0.525	
257#		0.578	1.094	0.541	
307		0.750	1.422		
307#		0.781	1.438		
367		1.156	2.203		
367#		1.188	2.234		
431		1.719	3.344		
431#		1.719	3.281		

- # - спеціалізований набір інструкцій процесора.

Таблиця Д.4

Час реалізації формування і перевірки ЕЦП згідно ДСТУ 4145-2002 (Broadcom BCM2835 SoC, ARM11 (ARM v6), 800 MHz, Rasbian Linux)

Поле, m	**Час, мс**							
	Формування ЕЦП				**Перевірка ЕЦП**			
	W	**E** $d_1 \neq d_2$	**E** $d_1 = d_2$	**Виграш**	**W**	**E** $d_1 \neq d_2$	**E** $d_1 = d_2$	**Виграш**
163	3.928	7.725			8.088	15.47		
167	3.996	8.032			8.176	16.102		
173	4.332	8.652	4.698		8.866	17.45	9.202	
179	4.583	9.201			9.349	18.502		
191	4.978	9.986			10.13	19.987		
233	9.652	19.01			19.527	38.036		
257	12.392	24.785	11.896	1.04	25.302	49.65	24.543	1.03
307	18.293	36.258			37.144	72.758		
367	27.187	54.102			54.846	107.986		
431	41.702	83.01			83.708	166.02		

Таблиця Д.5

Час реалізації формування і перевірки ЕЦП згідно ДСТУ 4145-2002

Поле, m	**Час, мс**							
	Формування ЕЦП				**Перевірка ЕЦП**			
	W	**E** $d_1 \neq d_2$	**E** $d_1 = d_2$	**Виграш**	**W**	**E** $d_1 \neq d_2$	**E** $d_1 = d_2$	**Виграш**
163	0.234	0.469			0.493	0.919		
163*	0.303	0.617			0.600	1.166		

Продовження таблиці Д.5

163**	0.298	0.602			0.617	1.288		
163#	0.209	0.407			0.421	0.865		
163##	0.209	0.406			0.453	0.828		
167	0.255	0.451			0.576	0.913		
167*	0.293	0.568			0.616	1.277		
167**	0.323	0.608			0.622	1.273		
167#	0.203	0.409			0.440	0.859		
167##	0.207	0.404			0.412	0.865		
173	0.253	0.486	0.278		0.529	1.016	0.543	
173*	0.320	0.632	0.342		0.789	1.368	0.801	
173**	0.322	0.652	0.344		0.655	1.359	0.689	
173#	0.225	0.443	0.241		0.462	0.940	0.489	
173##	0.223	0.431	0.239		0.459	0.893	0.472	
179	0.263	0.569			0.545	1.042		
179*	0.329	0.676			0.908	1.346		
179**	0.421	0.681			0.844	1.434		
179#	0.234	0.464			0.484	0.984		
179##	0.234	0.457			0.478	0.937		
191	0.326	0.540			0.326	1.110		
191*	0.347	0.668			0.804	1.430		
191**	0.360	0.720			0.776	1.433		
191#	0.246	0.485			0.537	1.012		
191##	0.240	0.476			0.512	1.006		
233	0.491	0.898			1.028	1.844		
233*	0.585	1.155			1.345	2.363		
233**	0.595	1.105			1.373	2.364		
233#	0.401	0.750			0.805	1.559		
233##	0.395	0.753			0.903	1.629		
257	0.645	1.203	0.616		1.358	2.472	1.254	
257*	0.859	1.579	0.823		1.970	3.417	1.880	
257**	0.840	1.665	0.797		1.863	3.352	1.778	
257#	0.567	1.103	0.544		1.262	2.318	1.204	
257##	0.559	1.067	0.534		1.200	2.259	1.145	
307	0.906	1.667			1.883	3.388		
307*	1.141	2.187			2.624	4.420		

Продовження таблиці Д.5

307**	1.185	2.233			2.295	4.431		
307#	0.778	1.446			1.618	3.148		
307##	0.757	1.456			1.634	3.221		
367	1.372	2.598			2.982	5.242		
367*	1.796	3.309			3.716	3.795		
367**	1.824	3.391			3.608	6.961		
367#	1.173	2.215			2.565	4.623		
367##	1.176	2.203			2.475	4.640		
431	2.134	3.945			4.239	7.681		
431*	2.563	4.657			5.337	9.749		
431**	2.566	4.779			5.178	10.257		
431#	1.735	3.362			3.632	7.093		
431##	1.737	3.385			3.650	7.554		

- Intel Xeon X5670 2.93 GHz, SentOS Linux v7.0 x86-64) використовуючи спеціалізований набір інструкцій;

* – Intel Xeon E5-2640 2.5 GHz (SentOS Linux v7.0 x86-64) зі стандартним набором інструкцій;

** – Intel Xeon E5-2640 2.5 GHz (SentOS Linux v7.0 x86-64) зі спеціалізованим набором інструкцій;

– Intel Core i7-4702MQ 3.20 GHz (SentOS Linux Ubuntu 16.04) зі стандартним набором інструкцій;

– Intel Core i7-4702MQ 3.20 GHz (SentOS Linux Ubuntu 16.04) зі спеціалізованим набором інструкцій;

W – двійкова крива Вейерштрасса;

E – бірaціонально еквівалентна крива Едвардса.

ДОДАТОК Е. Документи, що підтверджують впровадження результатів дисертації

ЗАТВЕРДЖУЮ

Проректор з наукової роботи
Національного авіаційного
університету

_____ В. Харченко
«__» _____ 2018 р.

АКТ

впровадження у навчальний процес результатів дисертаційної роботи Ковтун Марії Григорівни «Методи удосконалення арифметичних операцій у полях, кільцях та алгебраїчних кривих для криптографічних застосувань» на здобуття кандидата технічних наук.

Комісія у складі: голова – завідувач кафедри безпеки інформаційних технологій (БІТ) Корченко О.Г., доцент кафедри БІТ Гнатюк С.О., доцент кафедри БІТ Жмурко Т.О. склали даний акт про те, що результати дисертаційної роботи Ковтун Марії Григорівни впровадженні у навчальний процес та використовуються на кафедрі БІТ у 2016-2017 навчальному році при викладанні дисциплін «Новітні технології захисту інформації», «Безпека інформаційних і комунікаційних систем» що входить до навчальних планів підготовки фахівців у галузі знань «Інформаційна безпека».

№ з/п	Назва роботи, що впроваджується	Форма впровадження	Ефективність від впровадження
	1	2	3
1	Арифметичні операції над великими числами у кільцях, полях та їх представлення для ефективної програмної реалізації	Лабораторна робота	Студентами під час лабораторної роботи, програмно реалізуються арифметичні операції над великими числами (множення, додавання, віднімання, ділення, приведення за модулем та інвертування у полі). Для програмної реалізації слід використовувати мову програмування високого рівня.
2	Арифметичні операції над великими поліномами у кільцях, полях та їх представлення для ефективної програмної реалізації	Лабораторна робота	Студентами під час лабораторної роботи, програмно реалізовуються арифметичні операції над поліномами у поліноміальному базисі (множення, додавання, віднімання, ділення, приведення за модулем та інвертування). Для програмної реалізації слід використовувати мову програмування високого рівня.
3	Арифметичні перетворення над еліптичними кривими у формі Вейерштрасса та Едвардса, біраціональні відображення між ними.	Лекція	Ознайомлення студентів з теоретичними основами теорії еліптичних кривих, де розглядаються еліптичні криві у формах: Вейерштрасса, Хасе, Едвардса, Монтгомері, Кобліца, та можливі біраціональні відображення між ними. Розглядаються різні подання точок еліптичних кривих у афінному та проективному базисах. Приводяться аналітичні вирази для додавання та подвоєння точок еліптичних кривих у різних формах та і різних базисах подання точок.

Голова комісії,

завідувач кафедри безпеки інформаційних технологій О. Корченко

Члени комісії:

доцент кафедри безпеки інформаційних технологій С. Гнатюк

доцент кафедри безпеки інформаційних технологій Т. Жмурко

САЙФЕР

Системи захисту інформації

ТОВ «Сайфер ЛТД»
Адреса: 04107, Київ, вул. Нагірна, 25
Тел./Факс: (044) 484-46-17, 484-46-12, 483-03-22
E-mail: sales@cipher.kiev.ua
http://www.elpay.com; http://www.cipher.kiev.ua

Вих. № 12/09-17
28.09.2017 р.

АКТ

впровадження результатів дисертаційної роботи
Ковтун Марії Григорівни
у діяльності ТОВ «Сайфер ЛТД»

Комісія у складі голови – директора товариства з обмеженою відповідальністю «Сайфер ЛТД», кандидата технічних наук Боровікова О.М., членів комісії – провідного розробника Бойко С.Т., провідного розробника Прокоповича Л.Ю. склала цей акт про те, що при розробці бібліотек криптографічних примітивів «Шифр+», використано результати дисертаційної роботи Ковтун Марії Григорівни.

У криптопримітивах «Шифр+» використовуються арифметичні перетворення над цілими числами та поліномами – елементами поля $GF(2^m)$ для реалізації алгоритмів електронного цифрового підпису, де застосовуються такі результати дисертаційної роботи Ковтун М.Г:

1. **Удосконаленний метод** ділення великих цілих чисел на основі розширеного алгоритму Евкліда одинарної та подвійної точності, який відрізняється від відомого урахуванням двійкової довжини для виконання операцій порівняння, додавання та зсуву значимих (не порожніх) машинних слів, що дозволяє знизити обчислювальну складність операцій з цифровим підписом.

2. **Удосконаленний метод** інвертування поліномів – елементів поля $GF(2^m)$ на основі розширеного алгоритму Евкліда, який відрізняється від відомого урахуванням двійкової довжини поліномів для виконання операцій зсуву, додаванням лише значимих (не порожніх) машинних слів, що дозволило зменшити обчислювальну складність з цифровим підписом.

3. **Розроблено метод** побудови алгоритму приведення за фіксованим модулем у полі для заданого трьох- та п'ятичлену, який вперше формалізовано, що дозволяє в незалежності від полінома, що не приводиться згенерувати алгоритм приведення за модулем, що дозволяє зменшити обчислювальну складність з цифровим підписом.

4. **Удосконалено метод** операції здобуття n - вимірного кореня в полі $GF(2^m)$, де n - непарне, на прикладі кубічного кореня, який відрізняється від відомого процедурою піденесення до степеню, а саме розкладом показника степеню у адитивний ланцюг, що дозволяє зменшити обчислювальну складність генерації загальносистемних параметрів криптосистем на еліптичних кривих.

5. **Удосконалено метод** пошуку біраціонально еквівалентних кривих Едвардса в полі $GF(2^m)$, який відрізняється від відомого введенням процедури передобчислень елементів групи і компоновки результатів передобчислень. При здобутті кубічного кореня, використовувати розклад показника степеню у адитивний ланцюг, що дозволяє зменшити обчислювальну складність.

6. **Удосконалено метод** операції скалярного множення у групі точок еліптичної кривої над полем $GF(2^m)$, який відрізняється від відомого використанням проміжних

обчислень на кривій Едвардса за рахунок відмови від процесорних інструкцій галуження, що дозволяє підвищити стійкість до атак на реалізацію та підвищення продуктивності операцій з цифровим підписом.

Бібліотеки криптографічних примітивів «Шифр+» і перевірено їх якість та придатність для використання в реальних умовах.

Отже, результати, отримані Ковтун М.Г. під час написання дисертаційної роботи, дозволили підвищити швидкість криптографічних перетворень та стійкість до атак на реалізацію алгоритмів цифрового підпису за схемами ДСТУ 4145-2002, ECDSA, ECGDSA, EdDSA у бібліотеках криптографічних примітивів «Шифр+», які мають позитивний експертний висновок Держспецзв'язку № 04/03/02-1674 від 16.05.2017 р.

Голова комісії
Директор ТОВ «Сайфер ЛТД» О.М. БОРОВІКОВ

Члени комісії:
Провідний розробник: С.Т. БОЙКО

Провідний розробник: Л.Ю.ПРОКОПОВИЧ

ДОДАТОК Ж. Патенти України на корисну модель

УКРАЇНА

ПАТЕНТ

НА КОРИСНУ МОДЕЛЬ

№ 111632

СПОСІБ МНОЖЕННЯ ЦІЛИХ ЧИСЕЛ

Видано відповідно до Закону України "Про охорону прав на винаходи і корисні моделі".

Зареєстровано в Державному реєстрі патентів України на корисні моделі **25.11.2016.**

В.о. Голови Державної служби
інтелектуальної власності України А.А.Малиш

Державна служба інтелектуальної власності України
Ідентифікаційний код 37552556
№1

УКРАЇНА

ПАТЕНТ

НА КОРИСНУ МОДЕЛЬ

№ 118065

СПОСІБ ПІДНЕСЕННЯ ДО КВАДРАТА ЦІЛИХ ЧИСЕЛ

Видано відповідно до Закону України "Про охорону прав на винаходи і корисні моделі".

Зареєстровано в Державному реєстрі патентів України на корисні моделі **25.07.2017.**

Директор департаменту інтелектуальної власності Міністерства економічного розвитку і торгівлі України В.О. Жалдак

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37508598

УКРАЇНА

ПАТЕНТ

НА КОРИСНУ МОДЕЛЬ

№ 118066

СПОСІБ ПРИВЕДЕННЯ ЗА МОДУЛЕМ ЦІЛИХ ЧИСЕЛ

Видано відповідно до Закону України "Про охорону прав на винаходи і корисні моделі".

Зареєстровано в Державному реєстрі патентів України на корисні моделі **25.07.2017.**

Директор департаменту інтелектуальної власності Міністерства економічного розвитку і торгівлі України В.О. Жалдак

МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ ІДЕНТИФІКАЦІЙНИЙ КОД 37508596