МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ОХРІМЕНКО Андрій Олександрович

УДК 004.056.55:003.26

МЕТОДИ АРИФМЕТИЧНИХ ПЕРЕТВОРЕНЬ В ПОЛЯХ І КІЛЬЦЯХ ДЛЯ КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ

Спеціальність 05.13.21 – «Системи захисту інформації»

Автореферат

дисертації на здобуття наукового ступеня
кандидата технічних наук

Київ – 2020

Дисертацією є рукопис.

Робота виконана на кафедрі безпеки інформаційних технологій Національного авіаційного університету Міністерства освіти і науки України.

Науковий керівник: кандидат технічних наук
Ковтун Владислав Юрійович,
ТОВ «САЙФЕР ІТ», директор.

Офіційні опоненти: доктор технічних наук, професор
Кузнецов Олександр Олександрович,
Харківський національний університет імені В. Н. Каразіна, професор кафедри безпеки інформаційних систем і технологій;

доктор технічних наук, професор
Смірнов Олексій Анатолійович,
Центральноукраїнський національний технічний університет, завідувач кафедри кібербезпеки та програмного забезпечення.

Захист відбудеться «26» листопада 2020 р. о 13^{00} на засіданні спеціалізованої вченої ради Д 26.062.17 при Національному авіаційному університеті за адресою: 03680, м. Київ, пр. Любомира Гузара, 1, корпус 11, аудиторія 111.

З дисертацією можна ознайомитись у Науково-технічній бібліотеці Національного авіаційного університету за адресою: 03680, м. Київ, пр. Любомира Гузара, 1.

Автореферат розісланий «26» жовтня 2020 р.

Учений секретар
спеціалізованої вченої ради
д.т.н., доцент С.О. Гнатюк

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність. Глобалізаційні процеси втягують держави в безперервні перегони «на виживання», в яких критичним є оперативна реакція на різні події, що відбуваються в світі. Оперативність досягається за допомогою високої інформатизації суспільства і активного розвитку інформаційних і телекомунікаційних технологій в багатьох галузях економіки і сферах життя людини. Держава виступає в ролі основної рушійної сили та регулятора цих процесів. Таким чином, автоматизація більшості процесів в державі дозволяє значно збільшити масштаби та темпи розвитку усіх сфер діяльності, за рахунок більш якісної та ефективної обробки інформації, що надходить, встановлення нових зв'язків між різними ланками, а також можливості поступової відмови від використання паперового документообігу.

В України з метою автоматизації управлінських процесів створено не лише Міністерство цифрової трансформації та державне агентство з питань електронного урядування, але й прийнято та впроваджено ряд законів та нормативних актів. Проте при переході на безпаперове управління, виникає проблема та необхідність регулювання цих процесів і забезпечення їх захисту від несанкціонованого доступу, для цього в Україні реалізована та продовжує розвиватись інфраструктура відкритих ключів (ІВК) у вигляді національної ІВК. Для створення електронного підпису (ЕП), учасники електронного документообігу повинні мати особисті ключі та сертифікати відкритих ключів, отримані в центрі сертифікації ключів (ЦСК). ЦСК відносяться до інформаційно-телекомунікаційних систем (ІТС) загального користування з територіально розподіленою інфраструктурою, що забезпечують безперервне функціонування ЕП в масштабі часу близькому до реального. Це стало можливим лише після появи високопродуктивних обчислювальних систем і високошвидкісних мереж доступу до Інтернету, що дозволяють працювати в умовах нерівномірного навантаження звернень до ЦСК. Прикладом таких нерівномірних навантажень може служити подача звітності в кінці звітного періоду.

Досвід експлуатації подібних систем в державних органах влади показує тенденцію до постійного зростання кількості обслуговуваних сертифікатів відкритих ключів, а також зростання звернень до сервісів ЦСК (OCSP, TSP, HTTP). В результаті такого зростання, кількість звернень може перевищити розрахункове обчислювальне навантаження серверів ІТС ЦСК, що призведе до погіршення якості обслуговування користувачів або навіть до повної відмови в обслуговуванні. Крім того, сервіси ЦСК (OCSP, TSP, HTTP) протягом дня можуть обробляти десятки і навіть сотні мільйонів запитів, які нерівномірно розподілені в часі. Слід зазначити, що при кожному формуванні відповіді на запит OCSP або TSP, сервіс накладає ЕП на цю відповідь, відповідно одержувач зобов'язаний перевірити статус ЕП отриманої відповіді від сервісу. У сучасній реалізації Національної системи ЕП, під час роботи сервісів OCSP та TSP в ЦСК, основний час займає виконання криптографічних перетворень для постановки та перевірки ЕП, прийом і обробка запитів, передача відповідей. Таким чином, підвищення якості обслуговування клієнтів ЦСК, а саме зменшення часу обробки запитів сервісами OCSP та TSP можливо за допомогою зменшення часу виконання криптографічних перетворень ЕП і є *актуальною науково-практичною задачею,* що має теоретичне і практичне значення.

Зв’язок роботи з науковими програмами, планами, темами. Тематика дисертаційної роботи та одержані результати безпосередньо пов’язані з Постановою Президії Національної академії наук України №30 затвердженої 30.01.2019 р. «Про Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних, суспільних і гуманітарних наук Національної академії наук України на 2019-2023 роки» та відповідають науковим напрямам в області «1.2. Інформатика». Дисертаційне дослідження пов’язане зі Стратегією національної

безпеки України від 26 травня 2015 року №287/2015 у контексті п.4.12 «Забезпечення кібербезпеки і безпеки інформаційних ресурсів, зокрема реформування системи технічного і криптографічного захисту інформації з урахуванням практики держав-членів НАТО та ЄС», зі Стратегією кібербезпеки України від 15 березня 2016 року №96/2016 і Рамковою програмою ЄС з досліджень та інновацій «Horizon Europe». Результати роботи відображені у звітах держбюджетних науково-дослідних робіт Національного авіаційного університету «Квантово-криптографічні методи захисту критичної інформаційної інфраструктури держави» (0117U006770) та «Система забезпечення конфіденційності критичної інформаційної інфраструктури держави на базі квантових детерміністичних протоколів» (д.р. № 0120U101400), у яких здобувач брав участь у якості виконавця.

Мета і задачі дослідження. Метою роботи є підвищення швидкодії інформаційно-телекомунікаційних систем центрів сертифікації ключів національної інфраструктури відкритих ключів за рахунок розробки методів арифметичних перетворень над великими цілими числами з відкладеним переносом.

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

– проаналізувати існуючі підходи до представлення цілих чисел, а також методи арифметичних перетворень над числами в цих представленнях;

– розробити метод представлення цілих чисел з відкладеним переносом;

– удосконалити методи основних арифметичних перетворень (додавання, віднімання, множення, піднесення до квадрату, приведення за модулем, ділення, порівняння та зсувів) з відкладеним переносом;

– удосконалити методи арифметичних перетворень множення, піднесення до квадрату та приведення за модулем з відкладеним переносом та паралельним виконанням двох циклів множення в двох окремих потоках;

– удосконалити методи арифметичних перетворень множення, піднесення до квадрату та приведення за модулем з відкладеним переносом та паралельним виконанням ітерацій двох циклів множення в декілька потоків;

– програмно реалізувати розроблені методи арифметичних перетворень та експериментально дослідити їх для підтвердження ефективності в криптографічних перетвореннях з відкритим ключем для роботи з ЕП.

Об'єктом дослідження є процес криптографічних перетворень з відкритим ключем в інформаційно-телекомунікаційних системах ЦСК національної ІВК.

Предметом дослідження є методи арифметичних перетворень над великими цілими числами, що застосовуються у криптографічних перетвореннях з відкритим ключем.

Методи дослідження. Вибрані методи дослідження базуються на теорії ймовірності та комбінаторики (для аналізу складності алгоритмів); теорії полів, кілець та ідеалів (для удосконалення методів арифметичних перетворень над цілими числами); складності алгоритмів (для аналізу складності методів арифметичних перетворень); теорії криптографії (для аналізу криптографічних перетворень, що використовуються в різних схемах ЕП).

Наукова новизна одержаних результатів. У ході вирішення поставлених задач були отримані наступні результати.

– *вперше розроблено* метод представлення цілих чисел з відкладеним переносом, який за рахунок можливості відкласти операцію переносу зі старших розрядів в молодші та операцію займу з молодших розрядів у старші, дозволяє виключити взаємозалежність між машинними операціями при виконанні арифметичних перетворень та в свою чергу підвищити швидкодію криптографічних перетворень з відкритим ключем.

– *удосконалено* методи арифметичних перетворень додавання, віднімання, зсуву вліво, зсуву вправо, множення, піднесення до квадрату, приведення за модулем, ділення

та порівняння, які за рахунок використання цілих чисел в представленні з відкладеним переносом дозволяють підвищити швидкодію перетворень в полях та кільцях цілих чисел, що в свою чергу призводить до підвищення швидкодії криптографічних перетворень з відкритим ключем.

– *удосконалено* методи арифметичних перетворень множення, піднесення до квадрату та приведення за модулем великих цілих чисел з відкладеним переносом та паралельним виконанням двох циклів множення в двох окремих потоках, що дозволяє підвищити швидкодію криптографічних перетворень з відкритим ключем.

– *удосконалено* методи арифметичних перетворень множення, піднесення до квадрату та приведення за модулем великих цілих чисел з відкладеним переносом та паралельним виконанням ітерацій двох циклів множення в декілька потоків, що дозволяє підвищити швидкодію криптографічних перетворень з відкритим ключем.

Практичне значення одержаних результатів полягає в наступному:

1. Удосконалено методи арифметичних перетворень множення (дозволив підвищити швидкодію реалізації в 1,05-1,7 разів для w=32 біт, та 1,02-2,28 разів для w=64 біт відносно прототипу), піднесення до квадрату (дозволив підвищити швидкодію реалізації в 1,25-3,19 разів для w=32 біт, та 1,01-4,12 разів для w=64 біт відносно прототипу), приведення за модулем (дозволив підвищити швидкодію реалізації в 1,02-1,8 разів для w=32 біт, та 1,01-4,12 разів для w=64 біт відносно прототипу) великих цілих чисел з відкладеним переносом.
2. Удосконалено методи арифметичних перетворень множення (дозволив підвищити швидкодію реалізації в 1,01-3,24 разів для w=32 біт, та 1,07-2,29 разів для w=64 біт відносно прототипу), піднесення до квадрату (дозволив підвищити швидкодію реалізації в 1,01-5,75 разів для w=32 біт, та 1,18-2,46 разів для w=64 біт відносно прототипу), приведення за модулем (дозволив підвищити швидкодію реалізації в 1,22-2,10 разів для w=32 біт, та 1,04-2,46 разів для w=64 біт відносно прототипу) великих цілих чисел з використанням відкладеного переносу та паралельним виконанням двох циклів множення в двох окремих потоках.
3. Удосконалено методи арифметичних перетворень множення (дозволив підвищити швидкодію реалізації в 1,34-8,29 разів для w=32 біт, та 1,05-5,1 разів для w=64 біт відносно прототипу), піднесення до квадрату (дозволив підвищити швидкодію реалізації в 1,31-17,3 разів для w=32 біт, та 1,09-4,6 разів для w=64 біт відносно прототипу), приведення за модулем (дозволив підвищити швидкодію реалізації в 1,22-5,14 разів для w=32 біт, та 1,31-4,6 разів для w=64 біт відносно прототипу) великих цілих чисел з використанням відкладеного переносу та паралельним виконанням ітерацій двох циклів множення в декілька потоків.
4. Удосконалено методи арифметичних перетворень, що дозволяють підвищити швидкодію перетворень у кільці цілих чисел (операція додавання за модулем – в 1,01-1,97 разів для w=32 біт, та 1,03-3,28 разів для w=64 біт; операція віднімання за модулем – в 1,01-1,34 разів для w=32 біт, та 1,03-2,41 разів для w=64 біт; операція множення за модулем – в 1,01-1,34 разів для w=32 біт, та 1,03-2,41 разів для w=64 біт; операція піднесення до квадрату за модулем – в 1,02-1,46 разів для w=32 біт, та 1,02-1,73 разів для w=64 біт; операція піднесення до степеню за модулем – в 1,38-1,75 разів для w=32 біт, та 1,01-1,76 разів для w=64 біт).
5. Удосконалено методи арифметичних перетворень, що дозволяють підвищити швидкодію перетворень у простому полі цілих чисел (операція додавання (модуль загального вигляду) – в 1,02-1,09 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція додавання (модуль спеціального вигляду) – в 1,01-1,08 разів для w=32 біт, та 1,01-1,04 разів для w=64 біт; операція віднімання (модуль загального вигляду) – в 1,01-1,08 разів для w=32 біт, та 1,02-1,05 разів для w=64 біт; операція віднімання (модуль спеціального вигляду) – в 1,01-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція множення (модуль загального вигляду) – в 1,02-1,09 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція множення (модуль спеціального вигляду) – в 1,02-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція

піднесення до квадрату (модуль загального вигляду) – в 1,02-1,08 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція піднесення до квадрату (модуль спеціального вигляду) – в 1,02-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція піднесення до степеню (модуль загального вигляду) – в 1,03-1,12 разів для w=32 біт, та 1,02-1,14 разів для w=64 біт; операція піднесення до степеню (модуль спеціального вигляду) – в 1,03-1,09 разів для w=32 біт, та 1,02-1,13 разів для w=64 біт.).

6. Удосконалено методи арифметичних перетворень, що дозволяють підвищити швидкодію перетворень у групі точок ЕК над простим полем (додавання точок ЕК – в 1,02-1,08 разів для w=32 біт, та 1,01-1,09 разів для w=64 біт; подвоєння точок ЕК – в 1,01-1,06 разів для w=32 біт, та 1,02-1,08 разів для w=64 біт; додавання точок ЕК в змішаних координатах – в 1,03-1,06 разів для w=32 біт, та 1,02-1,07 разів для w=64 біт; подвоєння точок ЕК в проективних координатах – в 1,02-1,08 разів для w=32 біт, та 1,03-1,08 разів для w=64 біт; скалярне множення випадкової точки ЕК з використанням проективних координат – в 1,02-1,12 разів для w=32 біт, та 1,02-1,09 разів для w=64 біт; скалярне множення фіксованої точки ЕК з використанням проективних координат – в 1,02-1,12 разів для w=32 біт, та 1,02-1,10 разів для w=64 біт).

7. Удосконалено методи арифметичних перетворень, що дозволяють підвищити швидкодію криптографічних перетворень у криптосистемі ECDSA (генерування особистого ключа – в 1,02-1,18 разів для w=32 біт, та 1,02-1,09 разів для w=64 біт; генерування відкритого ключа – в 1,01-1,06 разів для w=32 біт, та 1,01-1,05 разів для w=64 біт; створення підпису – в 1,01-1,10 разів для w=32 біт, та 1,01-1,05 разів для w=64 біт; перевірка підпису – в 1,01-1,03 разів для w=32 біт, та 1,01-1,29 разів для w=64 біт).

8. Удосконалено методи арифметичних перетворень, що дозволяють підвищити швидкодію криптографічних перетворень у криптосистемі RSA (генерування особистого ключа – в 1,01-5,94 разів для w=32 біт, та 1,01-2,64 разів для w=64 біт; створення підпису – в 1,48-10,0 разів для w=32 біт, та 1,01-1,24 разів для w=64 біт; перевірка підпису – в 1,01-1,39 разів для w=32 біт, та 1,01-1,2 разів для w=64 біт).

9. Розроблено та отримано п’ять патентів України на корисну модель, а саме «Спосіб множення цілих чисел» (Патент 111632, опубліковано 26.11.2016, бюлетень № 22), «Спосіб піднесення до квадрату цілих чисел» (Патент 118065, опубліковано 25.07.2017, бюлетень № 14), «Спосіб приведення за модулем цілих чисел» (Патент 118065, опубліковано 25.07.2017, бюлетень № 14), «Спосіб криптографічного перетворення інформації з використанням подовжених кодів» (Патент 123375, опубліковано 26.02.2018, бюлетень №4), «Спосіб криптографічного перетворення інформації з використанням укорочених кодів» (Патент 123379, опубліковано 26.02.2018, бюлетень №4);

10. Методи арифметичних перетворень реалізовано у бібліотеках криптографічних примітивів «Шифр+ v.2.1» системи криптографічного захисту інформації «Шифр-Х.509» ТОВ «Сайфер ЛТД», що має дійсний позитивний експертний висновок Держспецзв'язку України від 16.05.2017 № 04/03/02-1674 (Акт № 22/17 від 04.08.2017 р.). Результати дисертаційних досліджень впроваджено у діяльність Кваліфікованого надавача електронних довірчих послуг Офісу Генерального прокурора України (Акт №18\10\2-8654-19 від 08.10.2020 р.) та Національного авіаційного університету (Акт від 25.09.2020 р.).

Особистий внесок здобувача. Основні положення і результати дисертаційної роботи, що виносяться до захисту, отримані автором самостійно. У роботах, написаних у співавторстві, автору належать: [1-4,12,15,17,20-22,25,31,33,34] – постановка завдання та розробка арифметичних перетворень з відкладеним переносом; [9,14,27] – розробка методів арифметичних перетворень з розпаралелюванням та відкладеним переносом; [5-8, 36] – розробка методики та обробка результатів експериментального дослідження; [7,29,35] – аналіз алгоритмів приведення цілих чисел за фіксованим модулем; [11,13,14] – дослідження та обґрунтування вимог до побудови сучасних криптосистем для захисту інформаційних

ресурсів держави; [23,24] – формування рекомендацій щодо реалізації криптографічних перетворень. З робіт, що опубліковані у співавторстві, у дисертаційній роботі використовуються виключно результати, отримані особисто здобувачем.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на таких наукових конференціях: НТК «Захист інформації з обмеженим доступом та автоматизація її обробки» (Київ, 2011 р., 2012 р.), МНПК молодих учених та студентів «Політ. Сучасні проблеми науки» (Київ, 2011 р., 2020 р.), МНПК «Інфокомунікації – сучасність та майбутнє» (Одеса, 2011 р., 2013 р.), НТК студентів та молодих учених «Наукоємні технології» (Київ, 2011 р.), НТК студентів та аспірантів «Захист інформації з обмеженим доступом та автоматизація її обробки» (Київ, 2012 р.), Всесвітній конгрес «Авіація у XXI столітті» – «Безпека в авіації та космічні технології» (Київ, 2012 р., 2014 р.), МНПК «Проблеми і перспективи розвитку IT-індустрії» (Харків, 2013 р., 2014 р., 2015 р.), МНПК «Інтегровані інтелектуальні робототехнічні комплекси (ІРТК)» (Київ, 2013 р.), МНПК «Інформаційні технології та комп'ютерна інженерія» (Вінниця, 2014 р.), НПК «Актуальні питання забезпечення кібербезпеки та захисту інформації» (Київ, 2015 р.), МНПК «Безпека інформації в інформаційно-телекомунікаційних системах» (Київ, 2013 р., 2015 р., 2016 р.), Міжнар. конф. «Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)» (Київ, 2015 р.), Міжнар. конф. «Control, Automation and Systems» (Кьонджу, 2016 р.) та ін.

Публікації. Основні положення дисертації опубліковано у 36 наукових працях, у тому числі – 3 колективних монографіях, 16 наукових статях (5 – у міжнародних рецензованих виданнях, що входять до бази даних SCOPUS, 9 – у вітчизняних фахових наукових журналах та 2 – у інших наукових виданнях), 5 патентів України на корисну модель, а також 12 матеріалів і тез доповідей на конференціях.

Структура роботи та її обсяг. Дисертація складається з анотації, змісту, переліку умовних позначень, вступу, чотирьох розділів, загальних висновків, додатків, списку використаних джерел (в кінці кожного розділу основної частини дисертації) і має 165 сторінок основного тексту, 65 рисунків, 13 таблиць, 127 сторінок додатків. Список використаних джерел містить 219 найменувань і займає 20 сторінок. Загальний обсяг дисертаційної роботи – 326 сторінок.

ОСНОВНА ЧАСТИНА

У **вступі** подано загальну характеристику роботи, обґрунтовано актуальність, сформульовано мету і задачі досліджень, відображено наукову новизну і практичну цінність отриманих результатів, наведено дані щодо їх апробації та впровадження.

У **першому розділі** проведено аналіз наукової літератури за темою дисертаційної роботи та нормативно-правових актів, що регулюють функціонування національної інфраструктури відкритих ключів та ЕП. Прийняття Закону України «Про електронні довірчі послуги» стало важливим кроком на шляху до розбудови єдиного простору довіри на основі системи електронних довірчих послуг (ЕДП). Цей Закон розширив перелік електронних послуг, що, в свою чергу, відкриває більше можливостей для держави, бізнесу та громадян. У зв'язку з цим спостерігається зростання кількості сервісів, що підтримують роботу з ЕП (з 10 в 2006 році, до більш ніж 150 в 2019 році), а також кількості активних власників сертифікатів ЕП (за останні 6 років більше ніж в 5 разів). В подальшому, все більше послуг буде надаватись в електронному вигляді, а кількість користувачів ЕП зростати. Багаторівнева структура інфраструктури відкритих ключів (ІВК), до якої входять користувачі ЕДП, надавачі ЕДП (зокрема ЦСК), центральний засвідчувальний орган, засвідчувальний центр НБУ та контролюючий орган (Держспецзв'язку), безпосередньо впливає на процес створення та перевірки ЕП на документах. Так для створення ЕП з повними даними для перевірки (в форматі

CAdES-X-Long) потребує формування двох запитів до сервера OCSP та одного запиту до сервера TSA, у відповідь на які формуються три відповіді, що містять ЕП. Для перевірки ж одного підпису в форматі CAdES-X-Long може знадобитись обчислення значень 5-9 геш-функцій та 5-9 ЕП. Як наслідок зростає навантаження на ІТС ЦСК (зокрема на сервіси OCSP та TSA), особливо в пікові періоди, що в свою чергу робить ЦСК елементами критичної інформаційної інфраструктури.

Стандарти ЕП, що можуть застосовуватись в національній ІВК, ґрунтуються на різних математичних апаратах: ДСТУ 4145-2002 на перетвореннях в групі точок еліптичної кривої над двійковим полем $GF(2^m)$ та операціях над елементами простого поля GF(p), ECDSA на перетвореннях в групі елементів поля $GF(2^m)$ або GF(p) та операціях над елементами простого поля GF(p), RSA на перетворення в кільці цілих чисел. Таким чином, арифметичні операції над цілими числами присутні в усіх вказаних схемах ЕП та застосовуються в різних криптографічних перетвореннях. В криптографічних перетвореннях можуть використовуватись цілі числа довжиною до 16384 біт.

При програмній чи апаратній реалізації перетворень над цілими числами суттєвим є форма представлення чисел. В сучасній криптографії застосовуються наступні представлення: двійкове представлення (binary), двійкове представлення зі знаком (signed-digit binary), несуміжне представлення (NAF), представлення в системі залишкових класів (RNS), представлення в частотній області (frequency domain). Найбільш поширеним та універсальним є двійкове представлення, решта мають обмежену специфічну сферу застосування в окремих криптосистемах. Проте двійкове представлення не завжди оптимальне та найбільш ефективне.

Також у розділі розглянуто арифметичні операції над цілими числами в двійковому представленні. Проаналізовано класичні алгоритми додавання, віднімання, зсуву вліво та право. Визначено, що операція множення займає провідне місце серед перетворень в кільцях та полях цілих чисел та є досить трудомісткою. Сьогодні відомі наступні методи множення цілих чисел, які використовуються в криптографії: в стовпчик, Карацуби-Офмана, Тоома-Кука, Шенхаге-Штрассена, Comba, Фюрера. Особливу увагу приділено широко відомому і часто використовуваному методу множення в стовпчик, та множенню методом Comba, який показує кращі результати швидкодії. Операція піднесення до квадрату цілих чисел є окремим випадком множення, при якому обидва множники рівні, тому розглядаються алгоритми на основі множення в стовпчик та методом Comba. Для операцій ділення та приведення за модулем проведено аналіз та класифікацію алгоритмів з врахуванням різних аспектів. Розглянуто класичний метод на основі ділення в стовпчик та часто застосовувані в криптографії методи Монтгомері та Барретта. Наприкінці розглянуто метод порівняння цілих чисел. Операції з цілими числами на основі класичних методів не враховують властивості сучасних процесорів, такі як суперскалярність та багатоядерність. У всіх основних арифметичних операціях присутні операції переносу чи займу, що негативно впливає на їх ефективність на сучасних процесорах.

Другий розділ присвячений розробці методу представлення цілих чисел з відкладеним переносом – Delayed Carry Form (DCF) та удосконаленню методів арифметичних перетворень для роботи з числами в DCF-представленні. Двійкове число $a = \{a_{n-1}, \ldots, a_1, a_0\}$ в DCF являє собою послідовність з машинних слів $d_{\text{DCF}} = \{d_{m-1}, \ldots, d_1, d_0\}_{\text{DCF}}$ розміром w –біт, при цьому, кожне машинне слово $d_i^{(w)} = a_i^{(r)} \,||\, a_i^{(v)}$ (Рис.1), де $a_i^{(r)}$ блок розміром r біт, що виділені для зберігання переносу, а $a_i^{(v)}$ блок розміром v біт, що заповнюються бітами з числа a (де $w = r + v$). В DCF-представленні, звичайне число a двійкової довжини l, в вигляді m блоків

розміром v біт, буде містити $m \cdot r$ біт надлишкової інформації, що відведена для зберігання переносів.

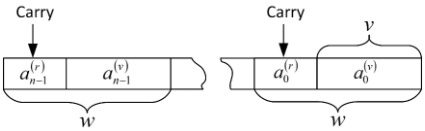

Рис. 1. Представлення цілих чисел в DCF

При роботі з числами в DCF представленні, процесор оперує машинними словами, в яких виділені біти для накопичення переносу та для зберігання безпосередньо самого числа.

Для перетворення двійкового числа $a_i^{(w)}$ в DCF представлення необхідно зарезервувати (обнулити) в машинному слові $d_i^{(w)}$ довжиною w –біт, r –біт під перенос $a_i^{(r)}$, а v – біт, що залишились, заповнюються бітами з числа $a_i^{(v)}$ в неперервній двійковій формі (Рис. 2).

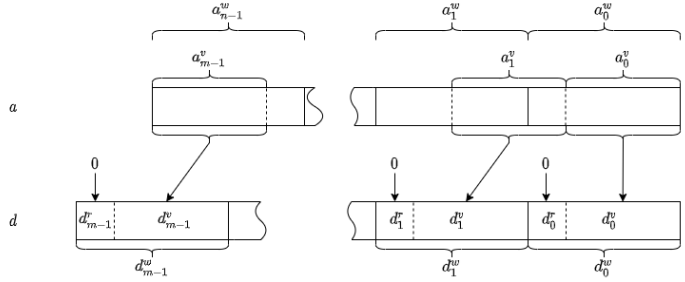

Рис. 2. Перетворення двійкового числа в DCF представлення

Для перетворення числа з DCF-представлення в двійкову неперервну форму, необхідно виконати коригування переносів – ітеративно врахувати перенос з молодшого машинного слова в старшому (Рис. 3).

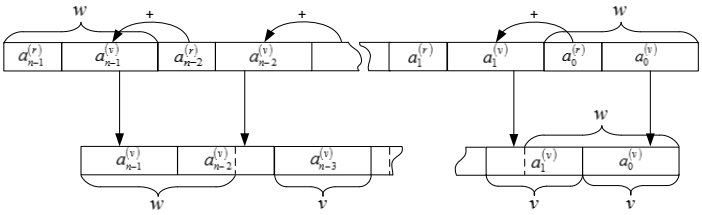

Рис. 3. Врахування переносу при перетворенні числа з DCF в двійкове представлення

Особливістю пропонованого DCF представлення цілих чисел є відсутність необхідності врахування переносів і займів, що дозволяє позбутися зайвих операцій присвоєння та перевірок при реалізації на мовах програмування високого рівня, а також від аналізу регістра прапорів на можливе перенесення. У свою чергу, це призводить до підвищення ефективності програмної реалізації на процесорах з суперскалярною архітектурою і можливостями сучасних компіляторів у передбаченні переходів, паралельного виконання команд, розгортання циклів і т.д. Виняток становить операція коригування переносів, яка виконується послідовно від молодших машинних слів до старших.

В розділі наводиться оцінка надлишковості представлення цілих чисел в DCF-представленні. Розмір числа в DCF-представленні можна обчислити за допомогою

виразу $m = \lceil n \cdot w / (w - r) \rceil$, де n та m – кількість машинних слів розміром w –біт, що необхідні для представлення цілого числа в двійковій неперервній формі та DCF-представлення відповідно, r – кількість біт, що виділені для накопичення переносу в машинному слові ($r < w$).

Аналітична оцінка надлишковості (в машинних словах) цілих чисел в DCF-представленні, може бути обчислена:

$$R(d_{DCF}) = m - n = \lceil n \cdot (\tfrac{w}{w-r} - 1) \rceil.$$

Відповідно, збільшення r – призводить до збільшення надлишковості числа в DCF-представленні, проте дозволяє відкласти операцію коригування переносів на більшу кількість арифметичних перетворень.

Для виконання перетворень з числами в DCF-представленні, необхідно удосконалити методи арифметичних перетворень з цілими числами.

В розділі запропоновані наступні методи арифметичних перетворень з числами в DCF-представленні:

- метод додавання (доданки та сума в DCF-представленні) та змішаного додавання: доданки в двійковій формі, сума в DCF-представленні; один доданок в DCF-представленні, а другий доданок та сума в двійковій формі.
- метод віднімання (зменшуване, від'ємник та різниця в DCF-представленні) та змішаного віднімання: зменшуване та від'ємник в двійковій формі, різниця в DCF-представленні; зменшуване в DCF-представленні, від'ємник в двійковій формі, а різниця в DCF-представленні; зменшуване в двійковій формі, а від'ємник та різниця в DCF-представленні.
- метод зсуву вліво (число та результат в DCF-представленні) та змішаного зсуву вліво (число в двійковій формі, а результат в DCF-представленні).
- метод зсуву вправо (число та результат в DCF-представленні) та змішаного зсуву вправо (число в двійковій формі, а результат в DCF-представленні).
- метод множення, що базується на основі методу Comba.
- метод піднесення до квадрату.
- метод приведення за модулем на основі методу Барретта.
- метод ділення на основі методу Барретта;
- метод порівняння.

Наприкінці розділу наводиться порівняння теоретичних оцінок обчислювальної складності існуючих методів для роботи з числами в двійковому представленні та запропонованих методів для роботи з числами в DCF представленні – кількість операцій та складність більшості запропонованих методів (за виключенням найбільш складних, таких як порівняння, ділення та приведення за модулем) нижче аналогічних існуючих методів.

У **третьому розділі** запропоновані підходи до розпаралелювання деяких методів арифметичних перетворень, а саме множення, піднесення до квадрату і приведення за модулем. Використання DCF-представлення цілих чисел дозволяє застосовувати паралельне використання операцій над відповідними машинними словами, за рахунок того, що порядок виконання операцій з машинними словами не є суттєвим. Для зручності розглядається частковий випадок DCF представлення, при якому розмір блоку r (для накопичення переносу), дорівнює розміру блоку v (що заповнюється бітами з числа в двійковому представленні) і відповідають довжині машинного слова w ($r = v = w$). Тобто, перенос накопичується в старших частинах $2w$-розрядних змінних. В якості прототипу методу множення обрано множення методом Comba, а в якості прототипу методу приведення за модулем – метод Барретта. Слід зазначити, що метод піднесення до квадрату є окремим

випадком методу множення, коли обидва множники однакові, а в методі приведення за модулем використовується часткове множення. Тобто у вказаних методах присутні два цикли множення, які можна виконати паралельно:

- паралельне виконання першого і другого циклу множення, з подальшим коригуванням результатів, використовуючи два паралельні потоки.
- паралельне виконання ітерацій першого і другого циклу множення, з подальшим злиттям проміжних результатів, з використанням множини паралельних потоків.

Відповідно до цього у розділі запропоновані методи множення, піднесення до квадрату та приведення за модулем без розпаралелювання ($r = v = w$), з паралельним виконанням двох циклів множення в два потоки та з паралельним виконанням ітерацій двох циклів множення в декілька потоків.

Розглянемо метод множення цілих чисел без розпаралелювання. На вхід подається два великих числа a і b, які представляються у вигляді машинних слів $a = (a_n, ..., a_i, ..., a_1, a_0)$ та $b = (b_n, ..., b_j, ..., b_1, b_0)$ розміром w-біт, n - кількість машинних слів необхідних для представлення чисел a і b. На виході отримуємо результат $c = a \cdot b,$ розміром $2n$ машинних слів.

Основу вказаного методу множення складають два цикли формування результату множення. Перший цикл формує результат в інтервалі та містить вкладений цикл множення в інтервалах $i = [\overline{0,k}]$ та $j = [\overline{k,0}]$. Другий цикл формує результат в інтервалі $k = [\overline{n,2n-1})$ з використанням допоміжного інтервалу $l = [\overline{1,n-1})$ та містить вкладений цикл множення в інтервалах $i = [\overline{l,n})$ та $j = [\overline{k-l,n-l})$. У вкладених циклах виконується множення $(uv)^{(2w)} \leftarrow a_i^{(w)} \cdot b_j^{(w)}$, результатом якого є ціле число розміром $2w$, яке потім розділяється на два w - розрядних $u^{(w)}$ и $v^{(w)}$. Накопичення відкладеного переносу відбувається в старших частинах $2w$-розрядних тимчасових змінних r_0 та r_1 (які заздалегідь проініціалізовані) на кожній ітерації:

$$r_0^{(2w)} \leftarrow r_0^{(2w)} + v^{(w)},\ r_1^{(2w)} \leftarrow r_1^{(2w)} + u^{(w)}.$$

На кожній ітерації циклу формування результату виконується корегування (врахування переносу) з використанням $2w$ - розрядних тимчасових змінних r_1 та r_2 (r_2 заздалегідь проініціалізована):

$$r_1^{(2w)} \leftarrow r_1^{(2w)} + \mathrm{hi}_{(w)}\left(r_0^{(2w)}\right),\ r_2^{(2w)} \leftarrow r_2^{(2w)} + \mathrm{hi}_{(w)}\left(r_1^{(2w)}\right).$$

та відбувається присвоєння кінцевого результату і зміна тимчасових змінних r_0, r_1 і r_2

$$c_k^{(w)} \leftarrow \mathrm{low}_{(w)}\left(r_0^{(2w)}\right),\ r_0^{(2w)} \leftarrow \mathrm{low}_{(w)}\left(r_1^{(2w)}\right),\ r_1^{(2w)} \leftarrow \mathrm{low}_{(w)}\left(r_2^{(2w)}\right),\ r_2^{(2w)} \leftarrow 0.$$

Наприкінці відбувається формування результату $c_{2n-1}^{(w)} \leftarrow low_{(w)}(r_0^{(2w)})$.

Результатом множення є ціле число $c = (c_{2n-1}, ..., c_k, ..., c_1, c_0)$.

Розглянемо метод множення цілих чисел з відкладеним переносом та паралельним виконанням двох циклів множення в двох окремих потоках. На вхід подаються два

великих числа a і b, які представлені у вигляді машинних слів $a=(a_n,...,a_i,...,a_1,a_0)$ та $b=(b_n,...,b_j,...,b_1,b_0)$ розміром w-біт, n - кількість машинних слів, що необхідні для представлення чисел a та b. На виході буде отримано результат $c=a\cdot b$, розміром $2n$ машинних слів.

Основу вказаного методу множення складають два цикли формування результату множення, які виконуються паралельно в двох окремих потоках. Перший цикл формує результат в інтервалі $k=[\overline{0,n})$ та містить вкладений цикл множення в інтервалах $i=[\overline{0,k}]$ та $j=[\overline{k,0}]$. Другий цикл формує результат в інтервалі $k=[\overline{n,2n-1})$ з використанням допоміжного інтервалу $l=[\overline{1,n-1})$ та містить вкладений цикл множення в інтервалах $i=[\overline{l,n})$ та $j=[\overline{k-l,n-l})$. Кожний потік використовує власні тимчасові змінні rl_0, rl_1 та rl_2, які заздалегідь проініціалізовані. У вкладених циклах виконується множення $(uv)^{(2w)}\leftarrow a_i^{(w)}\cdot b_j^{(w)}$ результатом якого є ціле число розміром $2w$, яке потім розділяється на два $u^{(w)}$ та $v^{(w)}$. Накопичення відкладеного переносу відбувається в старших частинах $2w$ - розрядних змінних rl_0 та rl_1 на кожній ітерації.

$$rl_0^{(2w)}\leftarrow rl_0^{(2w)}+v^{(w)},\ rl_1^{(2w)}\leftarrow rl_1^{(2w)}+u^{(w)}.$$

На кожній ітерації циклу формування результату виконується корегування (врахування переносу) з використанням $2w$ - розрядних тимчасових змінних rl_1 та rl_2: $rl_1^{(2w)}\leftarrow rl_1^{(2w)}+\mathrm{hi}_{(w)}\left(rl_0^{(2w)}\right)$, $rl_2^{(2w)}\leftarrow rl_2^{(2w)}+\mathrm{hi}_{(w)}\left(rl_1^{(2w)}\right)$ та відбувається присвоєння кінцевого результату і зміна тимчасових змінних rl_0, rl_1 і rl_2:

$$c_k^{(w)}\leftarrow \mathrm{low}_{(w)}\left(rl_0^{(2w)}\right),\ rl_0^{(2w)}\leftarrow \mathrm{low}_{(w)}\left(rl_1^{(2w)}\right),\ rl_1^{(2w)}\leftarrow \mathrm{low}_{(w)}\left(rl_2^{(2w)}\right),\ rl_2^{(2w)}\leftarrow 0.$$

Після завершення першого циклу формування результату множення для збереження значення можливого переносу використовується глобальна змінна r_0 : $r_0^{(2w)}\leftarrow rl_0^{(2w)}$.

Після завершення роботи двох паралельних потоків, необхідно виконати корекцію результатів роботи другого циклу формування результатів множення за рахунок переносу, отриманого в результаті роботи першого циклу. Для цього використовується цикл корекції результатів в інтервалі $k=[\overline{n,2n-1})$:

$$r_0^{(2w)}\leftarrow r_0^{(2w)}+c_k^{(w)},\ c_k^{(w)}\leftarrow \mathrm{low}_{(w)}\left(r_0^{(2w)}\right),\ \mathrm{low}_{(w)}\left(r_0^{(2w)}\right)\leftarrow hi_{(w)}\left(r_0^{(2w)}\right),\ hi_{(w)}\left(r_0^{(2w)}\right)\leftarrow 0.$$

Після завершення циклу корекції відбувається формування результату $c_{2n-1}^{(w)}$:

$$c_{2n-1}^{(w)}\leftarrow c_{2n-1}^{(w)}+low_{(w)}(r_0^{(2w)}).$$

Результатом множення є ціле число $c=(c_{2n-1},...,c_k,...,c_1,c_0)$.

Розглянемо метод множення цілих чисел з відкладеним переносом та паралельним виконанням ітерацій двох циклів множення в декілька потоків. На вхід подається два великих числа a і b, які представляються у вигляді машинних слів

$a=(a_n,...,\ a_i,...,a_1,a_0)$ та $b=(b_n,...,b_j,...,b_1,b_0)$ розміром w-біт, n - кількість машинних слів необхідних для представлення чисел a і b. На виході отримуємо результат $c=a\cdot b$, розміром $2n$ машинних слів.

Основу вказаного методу множення складають два цикли формування результату множення, ітерації яких не залежать одна від одної і можуть виконуватись в окремих потоках. Перший цикл формує результат в інтервалі $k=[\overline{0,n})$ та містить вкладений цикл множення в інтервалах $i=[\overline{0,k}]$ та $j=[\overline{k,0}]$. Другий цикл формує результат в інтервалі $k=[\overline{n,2n-1})$ з використанням допоміжного інтервалу $l=[\overline{1,n-1})$ та містить вкладений цикл множення в інтервалах $i=[\overline{l,n})$ та $j=[\overline{k-l,n-l})$. Кожний потік в рамках ітерації використовує заздалегідь проініціалізовані масиви $r0_i^{(2w)}$ та $r1_i^{(2w)}$), де $i=[\overline{0,2n-1})$. У вкладених циклах використовуються власні тимчасові змінні rl_0 та rl_1, які ініціалізуються на кожній ітерації циклів формування результату. У вкладених циклах виконується множення $(uv)^{(2w)}\leftarrow a_i^{(w)}\cdot b_j^{(w)}$, результатом якого є ціле число розміром $2w$, яке потім розділяється на два w- розрядних $u^{(w)}$ та $v^{(w)}$. Накопичення відкладеного переносу відбувається в старших частинах $2w$- розрядних змінних rl_0 та rl_1 на кожній ітерації:

$$rl_0^{(2w)}\leftarrow rl_0^{(2w)}+v^{(w)},\ rl_1^{(2w)}\leftarrow rl_1^{(2w)}+u^{(w)}.$$

На кожній ітерації циклу формування результату виконується збереження переносів: $r0_k^{(2w)}\leftarrow rl_0^{(2w)},\ r1_k^{(2w)}\leftarrow rl_1^{(2w)}$.

Після завершення циклів формування результатів, необхідно виконати цикл врахування збережених переносів в інтервалі $k=[\overline{0,2n-1})$, з використанням заздалегідь проініціалізованих тимчасових змінних $r^{(2w)}$, $rll_0^{(2w)}$, $rll_1^{(2w)}$:

$$rll_0^{(2w)}\leftarrow r0_k^{(2w)},\ rll_1^{(2w)}\leftarrow r1_k^{(2w)},\ rll_0^{(2w)}\leftarrow rll_0^{(2w)}+low_{(w)}(r^{(2w)}),$$

$$c_k^{(w)}\leftarrow low_{(w)}(rll_0^{(2w)}),\ rll_1^{(2w)}\leftarrow rll_1^{(2w)}+low_{(w)}(rll_0^{(2w)}),\ r^{(2w)}\leftarrow rll_1^{(2w)}.$$

Після завершення циклу врахування збережених переносів відбувається формування результату $c_{2n-1}^{(w)}:\ c_{2n-1}^{(w)}\leftarrow low_{(w)}(r^{(2w)})$.

Результатом множення є ціле число $c=(c_{2n-1},...,c_k,...,c_1,c_0)$.

Аналогічно у розділі наведено варіанти методів піднесення до квадрату та приведення за модулем.

В **четвертому розділі** виконується експериментальне дослідження запропонованих методів арифметичних перетворень над цілими числами в полях та кільцях цілих чисел, еліптичних кривих над полем цілих чисел, та в криптографічних перетвореннях схем ЕП, що застосовуються в національній ІВК України. Розроблено методику проведення експериментального дослідження, визначено мету та задачі експерименту, вхідні та вихідні параметри, послідовність дій та засоби проведення експерименту.

Для дослідження запропонованих у другому та третьому розділах методів та аналізу їх ефективності була використана програмна реалізація з використанням 32 та 64-бітних машинних слів для платформ Server, Desktop, Mobile та Embedded, що базуються на сучасному програмному та апаратному забезпеченні. Перетворення виконувалися над попередньо згенерованими випадковими та простими цілими числами. Порівняння отриманих результатів проводилось шляхом співставлення середнього часу виконання 1 мільйону ітерацій перетворень, що вимірювалися, в програмній реалізації. Ефективність запропонованих методів суттєво залежить від двійкової довжини цілого числа та розрядності машинних слів, а також має локальний екстремум, що дозволяє говорити про двійкові довжини цілих чисел, для яких запропоновані методи мають найбільшу ефективність.

Оцінка швидкодії арифметичних перетворень (множення, піднесення до квадрату та приведення за модулем) над цілими числами виконувалася без використання запропонованих методів арифметичних перетворень, з використанням запропонованих методів арифметичних перетворень без розпаралелювання, з використанням запропонованих методів арифметичних перетворень та розпаралелюванням в 2 потоки циклів множення, з використанням запропонованих методів арифметичних перетворень та розпаралелюванням ітерацій двох циклів в декілька потоків. Загалом, запропоновані методи арифметичних перетворень ефективніші: множення цілих чисел – в 1,05-1,7 разів для w=32 біт, та 1,02-2,28 разів для w=64 біт; множення цілих чисел з розпаралелюванням в 2 потоки – в 1,01-3,24 разів для w=32 біт, та 1,07-2,29 разів для w=64 біт; множення цілих чисел з розпаралелюванням в декілька потоків – в 1,34-8,29 разів для w=32 біт, та 1,05-5,1 разів для w=64 біт; піднесення до квадрату цілих чисел – в 1,25-3,19 разів для w=32 біт, та 1,01-4,12 разів для w=64 біт; піднесення до квадрату цілих чисел з розпаралелюванням в 2 потоки – в 1,01-5,75 разів для w=32 біт, та 1,18-2,46 разів для w=64 біт; піднесення до квадрату цілих чисел з розпаралелюванням в декілька потоків – в 1,31-17,3 разів для w=32 біт, та 1,09-4,6 разів для w=64 біт; приведення за модулем цілих чисел – в 1,02-1,8 разів для w=32 біт, та 1,01-4,12 разів для w=64 біт; приведення за модулем цілих чисел з розпаралелюванням в 2 потоки – в 1,22-2,10 разів для w=32 біт, та 1,04-2,46 разів для w=64 біт; приведення за модулем цілих чисел з розпаралелюванням в декілька потоків – в 1,22-5,14 разів для w=32 біт, та 1,31-4,6 разів для w=64 біт. Запропоновані методи арифметичних перетворень з розпаралелюванням ефективніші запропонованих методів без розпаралелювання: множення цілих чисел з розпаралелюванням в 2 потоки – в 1,17-2,71 разів для w=32 біт, та 1,02-2,03 разів для w=64 біт; множення цілих чисел з розпаралелюванням в декілька потоків – в 1,72-6,81 разів для w=32 біт, та 1,10-4,91 разів для w=64 біт; піднесення до квадрату цілих чисел з розпаралелюванням в 2 потоки – в 1,03-1,82 разів для w=32 біт, та 1,16-2,33 разів для w=64 біт; піднесення до квадрату цілих чисел з розпаралелюванням в декілька потоків – в 1,09-5,43 разів для w=32 біт, та 1,01-4,36 разів для w=64 біт; приведення за модулем цілих чисел з розпаралелюванням в 2 потоки – в 1,02-1,64 разів для w=32 біт, та 1,05-1,63 разів для w=64 біт; приведення за модулем цілих чисел з розпаралелюванням в декілька потоків – в 1,05-4,5 разів для w=32 біт, та 1,19-3,32 разів для w=64 біт.

Оцінка швидкодії арифметичних перетворень у кільці цілих чисел виконувалася без використання запропонованих методів арифметичних перетворень у кільці цілих чисел; з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел без розпаралелювання; з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел та розпаралелюванням в 2 потоки циклів множення; з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел та розпаралелюванням ітерацій двох циклів множення в декілька потоків. Загалом запропоновані методи арифметичних перетворень у кільці цілих чисел ефективніші:

операція додавання за модулем – в 1,01-1,97 разів для w=32 біт, та 1,03-3,28 разів для w=64 біт; операція віднімання за модулем – в 1,01-1,34 разів для w=32 біт, та 1,03-2,41 разів для w=64 біт; операція множення за модулем – в 1,01-1,34 разів для w=32 біт, та 1,03-2,41 разів для w=64 біт; операція піднесення до квадрату за модулем – в 1,02-1,46 разів для w=32 біт, та 1,02-1,73 разів для w=64 біт; операція піднесення до степеню за модулем – в 1,38-1,75 разів для w=32 біт, та 1,01-1,76 разів для w=64 біт. Запропоновані методи арифметичних перетворень в кільці цілих чисел з розпаралелюванням ефективніші запропонованих методів без розпаралелювання: операція додавання за модулем з розпаралелюванням в 2 потоки – в 1,01-2,07 разів для w=32 біт, та 1,01-1,74 разів для w=64 біт; операція додавання за модулем з розпаралелюванням в декілька потоків – в 1,01-1,97 разів для w=32 біт, та 1,01-1,97 разів для w=64 біт; операція віднімання за модулем з розпаралелюванням в 2 потоки – в 1,01-1,23 разів для w=32 біт, та 1,03-1,23 разів для w=64 біт; операція віднімання за модулем з розпаралелюванням в декілька потоків – в 1,02-1,44 разів для w=32 біт, та 1,01-1,44 разів для w=64 біт; операція множення за модулем з розпаралелюванням в 2 потоки – в 1,02-1,24 разів для w=32 біт, та 1,02-1,21 разів для w=64 біт; операція множення за модулем з розпаралелюванням в декілька потоків – в 1,02-1,2 разів для w=32 біт, та 1,02-1,14 разів для w=64 біт; операція піднесення до квадрату за модулем з розпаралелюванням в 2 потоки – в 1,03-1,15 разів для w=32 біт, та 1,01-1,15 разів для w=64 біт; операція піднесення до квадрату за модулем з розпаралелюванням в декілька потоків – в 1,01-1,12 разів для w=32 біт, та 1,01-1,09 разів для w=64 біт; операція піднесення до степеню за модулем з розпаралелюванням в 2 потоки – в 1,16-2,5 разів для w=32 біт, та 1,16-2,01 разів для w=64 біт; операція піднесення до степеню за модулем з розпаралелюванням в декілька потоків – в 1,11-4,62 разів для w=32 біт, та 1,23-4,69 разів для w=64 біт.

Оцінка швидкодії арифметичних перетворень у простому полі цілих чисел виконувалася без використання запропонованих методів і з використанням запропонованих методів арифметичних перетворень у простому полі цілих чисел. Розпаралелювання перетворень у полі не виконувалося, оскільки ефект від розпаралелювання досягається при значно більших значеннях розміру полів, ніж ті, що використовуються у криптосистемах ECDSA, ДСТУ 4145-2002 та інших. Розглядалися прості числа загального і спеціального вигляду (псевдо-мерсена) у якості модулей. Загалом запропоновані методи арифметичних перетворень в простому полі цілих чисел ефективніші: операція додавання (модуль загального вигляду) – в 1,02-1,09 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція додавання (модуль спеціального вигляду) – в 1,01-1,08 разів для w=32 біт, та 1,01-1,04 разів для w=64 біт; операція віднімання (модуль загального вигляду) – в 1,01-1,08 разів для w=32 біт, та 1,02-1,05 разів для w=64 біт; операція віднімання (модуль спеціального вигляду) – в 1,01-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція множення (модуль загального вигляду) – в 1,02-1,09 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція множення (модуль спеціального вигляду) – в 1,02-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція піднесення до квадрату (модуль загального вигляду) – в 1,02-1,08 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція піднесення до квадрату (модуль спеціального вигляду) – в 1,02-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція піднесення до степеню (модуль загального вигляду) – в 1,03-1,12 разів для w=32 біт, та 1,02-1,14 разів для w=64 біт; операція піднесення до степеню (модуль спеціального вигляду) – в 1,03-1,09 разів для w=32 біт, та 1,02-1,13 разів для w=64 біт.

Оцінка швидкодії перетворень у групі точок ЕК над простим полем (додавання, подвоєння та скалярне множення) виконувалася без використання запропонованих методів і з використанням запропонованих методів арифметичних перетворень у групі точок ЕК над простим полем цілих чисел. Розпаралелювання перетворень у базовому полі

не застосовувалось, оскільки ефект від розпаралелювання досягається при значно більших значеннях розміру полів, ніж ті, що використовуються у криптосистемах ECDSA, ДСТУ 4145-2002 та інших. Оцінка швидкодії арифметичних перетворень у групі точок ЕК над простим полем проводилась для наступних перетворень: додавання і подвоєння точок у афінних координатах, додавання і подвоєння точок у проективних координатах Чудновського, скалярне множення точок ЕК з довільною точкою на основі бінарного алгоритму зліва направо, скалярне множення з фіксованою точкою за алгоритмом Лім-Лі. Загалом запропоновані методи арифметичних перетворень у групі точок ЕК над простим полем ефективніші: додавання точок ЕК – в 1,02-1,08 разів для w=32 біт, та 1,01-1,09 разів для w=64 біт; подвоєння точок ЕК – в 1,01-1,06 разів для w=32 біт, та 1,02-1,08 разів для w=64 біт; додавання точок ЕК в змішаних координатах – в 1,03-1,06 разів для w=32 біт, та 1,02-1,07 разів для w=64 біт; подвоєння точок ЕК в проективних координатах – в 1,02-1,08 разів для w=32 біт, та 1,03-1,08 разів для w=64 біт; скалярне множення випадкової точки ЕК з використанням проективних координат – в 1,02-1,12 разів для w=32 біт, та 1,02-1,09 разів для w=64 біт; скалярне множення фіксованої точки ЕК з використанням проективних координат – в 1,02-1,12 разів для w=32 біт, та 1,02-1,10 разів для w=64 біт.

Оцінка швидкодії криптографічних перетворень у криптосистемі ECDSA, яка базується на перетвореннях групі точок ЕК над простим полем (додавання, подвоєння та скалярне множення) виконувалася без використання запропонованих методів і з використанням запропонованих методів арифметичних перетворень у простому полі цілих чисел. Розпаралелювання перетворень у базовому полі не застосовувалось, оскільки ефект від розпаралелювання досягається при значно більших значеннях розміру полів, ніж ті, що використовуються у криптосистемі ECDSA. Розглядалися прості числа спеціального вигляду (псевдо-мерсена) у якості модулей для базового поля та прості числа загального вигляду у якості модулей для поля порядку групи точок ЕК, оскільки вони передбачені NIST FIPS 186-3. Оцінка швидкодії проводилась для криптографічних перетворень (генерування особистого ключа; генерування відкритого ключа, як скалярне множення за фіксованою точкою ЕК; створення підпису, як скалярне множення за фіксованою точкою ЕК; перевірка підпису, як скалярне множення за довільною і фіксованою точкою ЕК) криптосистеми ECDSA. Загалом, криптографічні перетворень криптосистеми ECDSA з використанням запропонованих методів арифметичних перетворень у групі точок ЕК над простим полем ефективніші: генерування особистого ключа – в 1,02-1,18 разів для w=32 біт, та 1,02-1,09 разів для w=64 біт; генерування відкритого ключа – в 1,01-1,06 разів для w=32 біт, та 1,01-1,05 разів для w=64 біт; створення підпису – в 1,01-1,10 разів для w=32 біт, та 1,01-1,05 разів для w=64 біт; перевірка підпису – в 1,01-1,03 разів для w=32 біт, та 1,01-1,29 разів для w=64 біт.

Оцінка швидкодії криптографічних перетворень (генерація особистого ключа, створення і перевірка підпису) у криптосистемі ДСТУ 4145-2002, яка базується на перетвореннях групі точок ЕК над двійковим полем (додавання, подвоєння та скалярне множення) виконувалася з використанням запропонованих методів арифметичних перетворень. Результати швидкодії криптографічних перетворень національної криптосистеми ДСТУ 4145-2002 приводяться лише для порівняння з швидкодією криптосистеми ECDSA, оскільки криптосистема ДСТУ 4145-2002 базується на перетвореннях групі точок ЕК над двійковим полем та використовує незначну кількість перетворень у простому полі – поля порядку групи точок ЕК. Розпаралелювання перетворень у полі порядку не виконувалося, оскільки ефект від розпаралелювання досягається при значно більших значеннях розміру полів, ніж ті, що використовуються у криптосистемі ДСТУ 4145-2002 та незначної кількості таких операцій (у простому полі порядку групи точок ЕК).

Оцінка швидкодії криптографічних перетворень (генерування ключів, створення підпису та перевірка підпису) у криптосистемі RSA виконується без використання запропонованих методів арифметичних перетворень у кільці цілих чисел; з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел без розпаралелювання; з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел та розпаралелюванням в 2 потоки (генерація простих чисел p і q відбувається в окремих потоках); з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел та розпаралелюванням в 4 потоки (генерація простих чисел p і q відбувається в окремих потоках, з розпаралелюванням арифметичних перетворень у 2 потоки); з використанням запропонованих методів арифметичних перетворень у кільці цілих чисел та розпаралелюванням в декілька потоків (генерація простих чисел p і q відбувається в окремих потоках, з розпаралелюванням арифметичних перетворень у декілька потоків). Загалом, криптографічні операції криптосистеми RSA з використанням запропонованих методів арифметичних перетворень ефективніші: генерування особистого ключа – в 1,01-5,94 разів для w=32 біт, та 1,01-2,64 разів для w=64 біт; створення підпису – в 1,48-10,0 разів для w=32 біт, та 1,01-1,24 разів для w=64 біт; перевірка підпису – в 1,01-1,39 разів для w=32 біт, та 1,01-1,2 разів для w=64 біт. Загалом, криптографічні операції криптосистеми RSA з використанням запропонованих методів арифметичних перетворень з розпаралелюванням ефективніші, ніж з використанням запропонованих методів без розпаралелювання: генерування особистого ключа з застосуванням розпаралелювання у 2 потоки – в 1,01-6,33 разів для w=32 біт, та 1,01-4,57 разів для w=64 біт; створення підпису з застосуванням розпаралелювання у 2 потоки – в 1,01-1,13 разів для w=32 біт, та 1,01-2,67 разів для w=64 біт; перевірка підпису з застосуванням розпаралелювання у 2 потоки – в 1,01-1,25 разів для w=32 біт, та 1,01-2,54 разів для w=64 біт; генерування особистого ключа з застосуванням розпаралелювання у 4 потоки – в 1,01-13,18 разів для w=32 біт, та 1,01-8,29 разів для w=64 біт; створення підпису з застосуванням розпаралелювання у 4 потоки – в 1,05-2,42 разів для w=32 біт, та 1,19-2,04 разів для w=64 біт; перевірка підпису з застосуванням розпаралелювання у 4 потоки – в 1,01-1,20 разів для w=32 біт, та 1,01-1,17 разів для w=64 біт; генерування особистого ключа з застосуванням розпаралелювання у декілька потоків – в 1,02-12,74 разів для w=32 біт, та 1,09-8,02 разів для w=64 біт; створення підпису з застосуванням розпаралелювання у декілька потоків – в 1,17-5,24 разів для w=32 біт, та 1,17-5,25 разів для w=64 біт; перевірка підпису з застосуванням розпаралелювання у декілька потоків – в 1,01-1,20 разів для w=32 біт, та 1,01-1,14 разів для w=64 біт.

У **додатках** вміщено акти впровадження результатів дисертаційної роботи, а також результати експериментальних досліджень розроблених методів.

ВИСНОВКИ

Результатом виконаної роботи є розв’язання актуальної науково-практичної задачі підвищення швидкодії ІТС ЦСК національної ІВК за рахунок підвищення швидкодії реалізації криптографічних алгоритмів на основі розробки методів та алгоритмів арифметичних перетворень над великими цілими числами з відкладеним переносом.

У процесі виконання дисертаційної роботи отримані такі основні результати:

1. Проведено аналіз сучасних методів представлення цілих чисел, визначено їх переваги і недоліки, сфери застосування. Встановлено, що найбільш універсальним є двійкове представлення цілих чисел, проте воно є не завжди оптимальним, а решта представлень мають вузьку сферу використання.

2. Розроблено метод представлення цілих чисел з відкладеним переносом, який дозволяє відкласти перенос при виконанні арифметичних перетворень, що в свою чергу дозволяє підвищити швидкодію криптографічних перетворень з відкритим ключем.

3. Розроблено удосконалені методи арифметичних перетворень, які за рахунок використання цілих чисел в представленні з відкладеним переносом, дозволяють підвищити швидкодію перетворень в полях (операція додавання (модуль загального вигляду) – в 1,02-1,09 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція додавання (модуль спеціального вигляду) – в 1,01-1,08 разів для w=32 біт, та 1,01-1,04 разів для w=64 біт; операція віднімання (модуль загального вигляду) – в 1,01-1,08 разів для w=32 біт, та 1,02-1,05 разів для w=64 біт; операція віднімання (модуль спеціального вигляду) – в 1,01-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція множення (модуль загального вигляду) – в 1,02-1,09 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція множення (модуль спеціального вигляду) – в 1,02-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція піднесення до квадрату (модуль загального вигляду) – в 1,02-1,08 разів для w=32 біт, та 1,01-1,07 разів для w=64 біт; операція піднесення до квадрату (модуль спеціального вигляду) – в 1,02-1,07 разів для w=32 біт, та 1,01-1,08 разів для w=64 біт; операція піднесення до степеню (модуль загального вигляду) – в 1,03-1,12 разів для w=32 біт, та 1,02-1,14 разів для w=64 біт; операція піднесення до степеню (модуль спеціального вигляду) – в 1,03-1,09 разів для w=32 біт, та 1,02-1,13 разів для w=64 біт.) та кільцях цілих чисел (операція додавання за модулем – в 1,01-1,97 разів для w=32 біт, та 1,03-3,28 разів для w=64 біт; операція віднімання за модулем – в 1,01-1,34 разів для w=32 біт, та 1,03-2,41 разів для w=64 біт; операція множення за модулем – в 1,01-1,34 разів для w=32 біт, та 1,03-2,41 разів для w=64 біт; операція піднесення до квадрату за модулем – в 1,02-1,46 разів для w=32 біт, та 1,02-1,73 разів для w=64 біт; операція піднесення до степеню за модулем – в 1,38-1,75 разів для w=32 біт, та 1,01-1,76 разів для w=64 біт), що в свою чергу призводить до підвищення швидкодії криптографічних перетворень з відкритим ключем.

4. Розроблено удосконалені методи арифметичних перетворень множення (дозволив підвищити швидкодію реалізації в 1,01-3,24 разів для w=32 біт, та 1,07-2,29 разів для w=64 біт відносно прототипу), піднесення до квадрату (дозволив підвищити швидкодію реалізації в 1,01-5,75 разів для w=32 біт, та 1,18-2,46 разів для w=64 біт відносно прототипу) та приведення за модулем (дозволив підвищити швидкодію реалізації в 1,22-2,10 разів для w=32 біт, та 1,04-2,46 разів для w=64 біт відносно прототипу) великих цілих чисел з відкладеним переносом та паралельним виконанням двох циклів множення в двох окремих потоках, що дозволяє підвищити швидкодію криптографічних перетворень з відкритим ключем.

5. Розроблено удосконалені методи арифметичних перетворень множення (дозволив підвищити швидкодію реалізації в 1,34-8,29 разів для w=32 біт, та 1,05-5,1 разів для w=64 біт відносно прототипу), піднесення до квадрату (дозволив підвищити швидкодію реалізації в 1,31-17,3 разів для w=32 біт, та 1,09-4,6 разів для w=64 біт відносно прототипу) та приведення за модулем (дозволив підвищити швидкодію реалізації в 1,22-5,14 разів для w=32 біт, та 1,31-4,6 разів для w=64 біт відносно прототипу) великих цілих чисел з відкладеним переносом та паралельним виконанням ітерацій двох циклів множення в декілька потоків, що дозволяє підвищити швидкодію криптографічних перетворень з відкритим ключем.

6. В свою чергу запропоновані методи арифметичних перетворень дозволяють підвищити швидкодію криптографічних операцій у криптосистемі ECDSA (генерування особистого ключа – в 1,02-1,18 разів для w=32 біт, та 1,02-1,09 разів для w=64 біт; генерування відкритого ключа – в 1,01-1,06 разів для w=32 біт, та 1,01-1,05 разів для w=64 біт; створення підпису – в 1,01-1,10 разів для w=32 біт, та 1,01-1,05 разів для w=64 біт; перевірка підпису – в 1,01-1,03 разів для w=32 біт, та 1,01-1,29 разів для w=64 біт), та у криптосистемі RSA (генерування особистого ключа – в 1,01-5,94 разів для w=32 біт, та 1,01-2,64 разів для w=64 біт; створення підпису – в 1,48-10,0 разів для w=32 біт, та 1,01-1,24 разів для w=64 біт; перевірка підпису – в 1,01-1,39 разів для w=32 біт, та 1,01-1,2 разів для w=64 біт).

7. Ефективність запропонованих методів арифметичних перетворень підтверджується результатами експериментального дослідження за допомогою розробленого програмного забезпечення.

8. Зазначені результати роботи впроваджено у діяльність ТОВ «Сайфер ЛТД» (Акт № 22/17 від 04.08.2017 р.), Національного авіаційного університету (Акт від 25.09.2020 р.) та Кваліфікованого надавача електронних довірчих послуг Офісу Генерального прокурора (Акт №18\10\2-8654-19 від 08.10.2020 р.), що підтверджено відповідними актами впровадження, які містяться у додатках до дисертаційної роботи.

ПУБЛІКАЦІЇ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. А. Охрименко, В. Ковтун, «Умножения целых чисел с использованием отложенного переноса для криптосистем с открытым ключом», *Информационные технологи и системы в управлении, образовании, науке: Монография*, под ред. проф. В.С. Пономаренко, Харьков: Цифрова друкарня №1, 2013, С. 69-82.
2. А. Охрименко, В. Ковтун, «Метод повышения производительности операции приведения по простому модулю», *Информационные системы в управлении, образовании, промышленности: Монография*, под ред. В.С. Пономаренко, Харьков: Вид-во ТОВ «Щедра садиба плюс», 2014, С. 204-219.
3. А. Охрименко, В. Ковтун, «Арифметические операции с отложенным переносом над целыми числами», *Информационные технологии и защита информации в информационно-коммуникационных системах: Монография*, под ред. В.С. Пономаренко, Харьков: Вид-во ТОВ «Щедра садиба плюс», 2015, С. 193-207.
4. R. Brumnik, V. Kovtun, A. Okhrimenko, S. Kavun, «Techniques for Performance Improvement of Integer Multiplication in Cryptographic Applications», *Mathematical Problems in Engineering*, 2014, P. 1-7. (Scopus)
5. V.Yu. Kovtun, M.G. Kovtun, A.O. Okhrimenko, «Commands integrity and authority in control radio link of UAV», *Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2015 IEEE International Conference*, 2015, pp. 178-181. (Scopus)
6. A.O. Okhrimenko, M.G. Kovtun, S.O. Gnatyuk, V.Yu. Kovtun, «Development of a search method of birationally equivalent binary edwards curves for binary weierstrass curves from DSTU 4145-2002», *in Proc. of 2nd Intern. Scientific-Practical Conf. on the Problems of Infocommunications Science and Technology (PIC S&T)*, Kharkiv, Ukraine, October 13-15, 2015, pp. 5-8. (Scopus)
7. A. Okhrimenko, M. Kovtun, 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. (Scopus)
8. A. Okhrimenko, V. Kovtun «Experimental research of the developed methods of arithmetic operations in cryptographic transformations according to the ECDSA scheme» Proceedings 1st International Conference on Cyber Hygiene and Conflict Management in Global Information Networks (CyberConf 2019), Lviv, Ukraine, November 29, 2019. – CEUR Workshop Proceedings, p. 827-837. (Scopus)
9. O.G. Korchenko, V.Yu. Kovtun, A.O. Okhrimenko, «Parallelization of Integer Squaring Algorithms with Delayed Carry», *Journal of Computer Networks*, 2014, Vol. 2(2), pp 10-17.
10. О.Г. Корченко, С.О. Гнатюк, Ю.Є. Хохлачова, А.О. Охріменко, «Основні критерії та вимоги до побудови сучасних криптосистем», *Вісник Інженерної академії України*, №3-4, С. 77-83, 2011.
11. В.Ю. Ковтун, А.А. Охрименко, В.В. Нечипорук, «Подходы к повышению производительности программной реализации операции умножения в поле целых чисел», *Захист інформації*, Т. 14, № 1 (54), С. 68-75, 2012.
12. С.О. Гнатюк, В.М. Кінзерявий, А.О. Охріменко, «Особливості криптографічного захисту державних інформаційних ресурсів», *Безпека інформації*, Т. 17, № 1, С. 68-80, 2012.

13. В.Ю. Ковтун, А.А. Охрименко, «Подходы к распараллеливанию программной реализации операции умножения в поле целых чисел», *Радиотехника. Всеукраинский межведомственный научно-технический сборник,* № 171, С. 123-132, 2012.

14. V. Kovtun, A. Okhrimenko, «Integer multiplication algorithm with delayed carry mechanism for public key cryptosystems», *Безпека інформації,* Vol. 19, №1, pp. 45-50, 2013.

15. А.А. Охрименко, «Эффективная программная реализация алгоритмов умножения целых чисел для современных платформ», *Вісник Інженерної академії України*, № 2, С. 108-113, 2013.

16. В.Ю. Ковтун, А.А. Охрименко, «Алгоритм возведения в квадрат целых чисел с использованием отложенного переноса», *Безпека інформації*, Т. 19, №3, С. 188-192, 2013.

17. А.А. Охрименко, «Обобщенные алгоритмы возведения в квадрат целых чисел с использованием отложенного переноса и технологий распараллеливания», *Вісник Інженерної академії України*, № 1, С. 114-119, 2014.

18. А.А. Охрименко, «Арифметика с отложенным переносом», *Захист інформації*, Т. 16, № 2, С. 130-138, 2014.

19. А.О. Охріменко, «Оптимізація програмної реалізації криптографічних алгоритмів», *Защита информации: сб. науч. труд.,* № 18, С. 44-51, 2011.

20. А.О. Охріменко, В.Ю. Ковтун, М.Г. Ковтун, С.Ю. Ковтун, С.П. Євсеєв, О.Г. Король, «Спосіб множення цілих чисел», Пат. 111632 Україна, МПК G06F 7/253 (2006.01). Заявка № u 2015 11473; заявл. 23.11.2015; опублік. 25.11.2016, Бюл. № 22.

21. А.О. Охріменко, В.Ю. Ковтун, М.Г. Ковтун, С.П. Євсеєв, О.Г. Король, Р.В. Грищук, Г.П. Коц, «Спосіб піднесення до квадрата цілих чисел», Пат. 118065 Україна, МПК G06F 7/523 (Пат. 118066 Україна, МПК G06F 7/523 (2006.01)2006.01). Заявка № u 2016 13439; заявл. 27.12.2016; опублік. 25.07.2017, Бюл. № 14.

22. А.О. Охріменко, В.Ю. Ковтун, М.Г. Ковтун, «Спосіб приведення за модулем цілих чисел», Пат. 118066 Україна, МПК G06F 7/523 (2006.01). Заявка № u 2016 13441; заявл. 27.12.2016; опублік. 25.07.2017, Бюл. № 14.

23. А.О. Охріменко, С.П. Євсеєв, Р.В. Грищук, О.Г. Король, Г.П. Коц, Р.В. Корольов, В.Ю. Ковтун, М.Г. Ковтун, «Спосіб криптографічного перетворення інформації з використанням подовжених кодів», Пат. 123375 Україна, МПК (2006) G09C 1/00. Заявка № u 2017 08985; заявл. 11.09.2017; опублік. 26.02.2018, Бюл. № 4.

24. А.О. Охріменко, С.П. Євсеєв, Р.В. Грищук, О.Г. Король, Г.П. Коц, Р.В. Корольов, В.Ю. Ковтун, М.Г. Ковтун, «Спосіб криптографічного перетворення інформації з використанням укорочених кодів», Пат. 123379 Україна, МПК (2006) G09C 1/00, H04L 9/06 (2006.01), G06F 21/72 (2013.01), G06F 21/60 (2013.01). Заявка № u 2017 08995; заявл. 11.09.2017; опублік. 26.02.2018, Бюл. № 4.

25. А.А. Охрименко, В.Ю. Ковтун, «Подходы к повышению быстродействия криптосистем с открытым ключом», *Проблеми і перспективи розвитку ІТ-індустрії: V міжнар. наук.-практ. конф., 25-26 квітня 2013 р*., Харків, 2013, С. 202.

26. А.А. Охрименко, «Применение механизмов отложеного переноса и распараллеливания для повышения производительности операции возведения в квадрат целых чисел», *Безопасность информации в информационно-телекоммуникационных системах: XVI Междун. науч.-практ. конф., 21–24 мая 2013 г*., К., 2013, С. 28-29.

27. А.А. Охрименко, В.Ю. Ковтун, «Алгоритм умножения целых чисел с использова-нием технологий распараллеливания», *Питання оптимізації обчислень (ПОО-XL): праці міжн. наук. конф., 30 вересня – 4 жовтня 2013 р*., К., 2013, С.125-126.

28. A. Okhrimenko, «Squaring algorithms for public-key cryptosystems», *Інфокомунікації – сучасність та майбутнє: матеріали III міжнар. наук.-пр. конф. мол. вчених 17-18 жовтня 2013 р.*, Одеса, 2013, С. 178-182.

29. А.А. Охрименко, В.Ю. Ковтун, «Классификация методов повышения производительности операции приведения по большому простому модулю», *Проблеми і*

перспективи розвитку IT-індустрії: VI міжнар. наук.-практ. конф., 17-18 квітня 2014 р.: тези доп., Харків, 2014, С. 253.

30. А.А. Охрименко, «Модифицированный алгоритм Баррета приведения целых чисел по модулю», *Інформаційні технології та комп'ютерна інженерія: IV міжнар. наук.-практ. конф., 28-30 травня 2014 р.,* Вінниця, 2014, С. 180-181.

31. A. Okhrimenko, V.Yu. Kovtun, O.L. Stokipniy, «Integer representation with delayed carry», *AVIATION IN THE XXI-st CENTURY – Safety in Aviation and Space Technologies: VI World Congress, September 23-25, 2014*., К., 2014, P. 1.11.10-1.11.14.

32. A. Okhrimenko, «Arithmetic operations with delayed carry for public key transformations», *Актуальні питання забезпечення кібернетичної безпеки та захисту інформації Зб. наук. праць наук.-практ. конф., 25-28 лютого 2015 р.,* К., 2015, С. 83-84.

33. А.А. Охрименко, В.Ю. Ковтун, «Операции арифметического сдвига над целыми числами с отложенным переносом», *Проблеми та перспективи розвитку IT-індустрії: VII міжнар. наук.-практ. конф., 17-18 квітня 2015 р.,* Харків, 2015, С. 30.

34. А.А. Охрименко, В.Ю. Ковтун, О.Л. Стокипный «Использование представления целых чисел с отложенным переносом в криптографических преобразованиях», *Безпека інформації в інформаційно-телекомунікаційних системах: матеріали XVI міжнар. наук.-практ. конф. 26-28 травня 2015 р.:* К., 2015, С. 14-15.

35. А.О Охріменко, М.Г. Ковтун, «Метод побудови алгоритму приведення по фіксованому модулю незвідного поліному», *Безпека інформації в інформаційно-телекомунікаційних системах: матеріали XVI міжнар. наук.-практ. конф. 25-26 травня 2016 р.,* Київ, 2016, С. 21.

36. A. Okhrimenko, V. Kovtun «Experimental research of developed arithmetic transformations according to RSA», XX International conference of higher education students and young scientists «POLIT. Challenges of science today: Modern information and communication technologies in aviation», Kyiv, May 1-3, 2020., p. 30-31.

АНОТАЦІЯ

Охріменко А.О. Методи арифметичних перетворень в полях і кільцях для криптографічних застосувань. – Рукопис.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.21 – системи захисту інформації. – Національний авіаційний університет, Київ, 2020.

Дисертаційна робота присвячена розв'язанню актуальної науково-практичної задачі дослідження і розробки нових методів арифметичних перетворень над великими цілими числами з відкладеним переносом для підвищення швидкодії реалізації криптографічних перетворень, що мають місце в інформаційно-телекомунікаційних системах центрів сертифікації ключів національної інфраструктури відкритих ключів України. В роботі запропоновано метод представлення цілих чисел з відкладеним переносом, який за рахунок можливості відкласти операцію переносу зі старших розрядів в молодші та операцію займу з молодших розрядів у старші, дозволяє виключити взаємозалежність між машинними словами при виконанні арифметичних перетворень. Удосконалено методи арифметичних перетворень додавання, віднімання, зсуву вліво, зсуву вправо, множення, піднесення до квадрату, приведення за модулем, ділення та порівняння, які за рахунок використання цілих чисел в представленні з відкладеним переносом дозволяють підвищити швидкодію перетворень в полях та кільцях цілих чисел. Також в роботі запропоновано методи арифметичних перетворень множення, піднесення до квадрату та приведення за модулем великих цілих чисел з відкладеним переносом та розпаралелюванням в два та декілька потоків. Використання запропонованих методів дозволяє підвищити швидкодію перетворень в криптографічних системах електронного підпису, що використовуються в національній інфраструктурі відкритих ключів.

Ключові слова: електронний підпис, інфраструктура відкритих ключів, представлення цілих чисел, арифметичні операції, відкладений перенос підвищення швидкодії, розпаралелювання, просте поле, кільце цілих чисел, група точок еліптичної кривої, ECDSA, RSA, ДСТУ 4145.

ABSTRACT

Okhrimenko A.O. Methods of arithmetic operations in rings of integers and prime fields for cryptographic applications. – Manuscript.

Thesis for a Candidate of Technical Science degree in specialty 05.13.21 – information security systems. – National Aviation University, Kyiv, 2020.

Thesis is devoted to solving the scientific and practical task of research and development new methods of arithmetic transformations over large integers with delayed carry for increasing implementation of cryptographic transformations that take place in information and telecommunication systems of certification authorities in national public key infrastructure of Ukraine.

The national PKI regulates the use of a qualified electronic signature according to the algorithms of DSTU 4145-2002, ECDSA, DSA and RSA. Operations of creating and verifying electronic signature are based on various mathematical methods: transformation in a ring of integers, field of integers and polynomials, in a group of points of an elliptic curve. All these transformations are impossible without arithmetic operations on integers.

In this work proposed the method of integer representation with delayed carry, which due to the possibility of postponing carry operation from higher to lower words and the loan operation from lower to higher words, eliminates the interdependence between machine words when performing arithmetic operations. Performing operations with integers in the DCF representation, the processor operates with machine words in which two blocks are allocated to store the carry bits and to store the information bits. To convert a binary number to DCF it is necessary to reserve in the machine word r-bits for carry, and the remaining v-bits are filled with bits from the integer in a binary form. To convert a number from a DCF representation to a binary form, it is necessary to adjust carry (iteratively apply the carry from the lower machine word to the higher).

To perform operations with integers in the DCF representation, it is necessary to modify the algorithms of arithmetic operations. Improved methods of arithmetic operations – addition, subtraction, left shift, right shift, multiplication, squaring, modular reduction, division and comparison, which by using integers in the delayed carry representation can increase the speed of operations in fields and rings of integers.

The use of delayed carry allows to apply some approaches of parallelization for methods of arithmetic operations, for example, multiplication, squaring and modular reduction. In these operations, there is a multiplication operation, which consists of two multiplication cycles that can be parallelized. The first approach involves the parallel execution of the first and second multiplication cycles with subsequent adjustment of the results in two threads. The second approach involves the parallel execution of iterations of the first and the second multiplication cycles, followed by the merging of intermediate results, using multiple parallel threads. Proposed the methods of arithmetic operations of multiplication, squaring and modular reduction of large integers with delayed carry and parallelization in two or multiple threads.

This work contains results of experimental studies of the proposed methods of arithmetic operations in fields and rings of integers, elliptic curves over the prime field, and cryptographic transformations of electronic signature schemes used in the national public key infrastructure of Ukraine. The use of the proposed methods allows to increase the speed of operations in cryptographic electronic signature systems in the national public key infrastructure.

Keywords: electronic signature, public key infrastructure, integer representation, arithmetic operations, delayed carry, speed enhancement, parallelization, prime field, ring of integers, elliptic curve points group, ECDSA, RSA, DSTU 4145.