МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

На правах рукопису

УДК 004.056.53

Коркішко Леся Мирославівна

МЕТОДИ ТА ЗАСОБИ МАСКОВАНОЇ АРИФМЕТИКИ ДЛЯ ПРИСТРОЇВ ЗАХИСТУ ІНФОРМАЦІЇ

Спеціальність 05.13.21 – Системи захисту інформації

Дисертація на здобуття вченого ступеня

кандидата технічних наук

Науковий керівних:

доктор технічних наук, професор

Карпінський Микола Петрович

Університет у Бєльсько-Бялій (Польща),

завідувач кафедри інформатики та автоматики

Київ – 2017

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АСП	аналіз споживаної потужності
МП	масковане представлення
АМ	арифметичне маскування
ЛМ	логічне маскування
ОБ	операційний блок
СП	споживана потужність

ВСТУП

Актуальність. Зростання цінності інформації, яка зберігається, обробляється та передається у комп’ютерних системах зумовило зростання актуальності задач забезпечення конфіденційності, цілісності та автентичності інформації при зростанні ймовірності реалізації загроз несанкціонованого доступу до такої інформації. У комп’ютерних системах перелічені задачі часто вирішують шляхом криптографічних перетворень інформації. Сучасні криптографічні перетворення володіють належним рівнем стійкості до їх математичного аналізу з метою обчислення параметрів криптографічних перетворень. З іншого боку, набули широкого розповсюдження методи визначення параметрів криптографічних перетворень на основі аналізу залежності спостережуваних фізичних характеристик комп’ютерних пристроїв які їх реалізують (час обробки, споживаний струм, електромагнітне випромінювання, тощо) від даних, які обробляються – так звані "інженерно-криптографічні атаки". Одним із найбільш небезпечних напрямків атакування пристроїв є інженерно-криптографічні атаки на основі аналізу залежності споживаної потужності (АСП) пристрою від параметрів та даних криптографічного перетворення, де пристрій знаходиться під контролем порушника (введення даних, маніпулювання та встановлення сигналів синхронізації, під’єднання живлення). Відомі рішення для зменшення такої залежності і, як наслідок, значного ускладнення аналізу, передбачають рандомізоване виконання алгоритмів шифрування (перемішування порядку виконання елементарних операцій, випадкова зміна шляху виконання алгоритму), спотворення справжньої залежності характеристик споживаної потужності (введення шуму чи збільшення його рівня), вирівнювання споживаної потужності при обробці різних даних (фільтрування, спеціальні логічні елементи, тощо). Таким рішенням притаманні недоліки в частині їх високої вартості та низької технологічності виготовлення, підвищеного енергоспоживання, зменшеної продуктивності обробки даних, складності

програмної реалізації. Тому дослідниками, зокрема Кочером П., Мессергесом Т., був розроблений альтернативний шлях захисту від атак на основі АСП, який полягає у введенні невизначеності у рівень СП пристрою шляхом рандомізування проміжних значень, які виникають у процесі обчислень криптографічного перетворення. При цьому дані обробляються у так званому «маскованому представленні» (МП), яке містить хоча б одну випадкову маску та результат виконання деякої операції маскування над початковими даними та усіма масками. У ролі операції маскування використовують операцію додавання за модулем два, що, у деяких випадках, призводить до значного ускладнення методів виконання арифметичних операцій над даними. З огляду на те, що збільшення кількості масок у МП призводить до зменшення залежності СП від немаскованих даних та значного ускладнення атак на основі АСП, розробка методів та засобів виконання базових операцій, характерних для криптографічних перетворень шифрування, над даними у МП із довільною кількістю масок, є актуальним напрямком наукових досліджень.

Значний внесок у розвиток захисту інформації від АСП внесли такі вчені як Голік Д., Квісквотер Я., Корон Дж., Кочер П., Мессергес Т., Трічіна Е., Освальд Е., та ін.

Однак, у зазначеній галузі залишається низка завдань, вирішення яких має важливе наукове та практичне значення. З цих позицій, побудова і дослідження методів та засобів виконання складових операцій криптографічних перетворень над даними у МП, є актуальним науковим завданням.

Зв'язок роботи з науковими програмами, планами, темами. Одержані результати дисертаційної роботи відображені у звітах держбюджетних науково-дослідних робіт Тернопільського національного економічного університету «Методи та засоби реалізації алгоритмів захисту інформації стійких до атак на реалізацію» (№ 0105U008181, що виконувалась 07.2005-12.2010), та «Паралельні методи та засоби реалізації алгоритмів захисту інформації в комп’ютерних

мережах з використанням математичного апарату еліптичних кривих» (№ 0109U000035, що виконувалась 01.2009 -12.2013).

Мета і задачі дослідження. Метою дисертаційної роботи є підвищення ефективності захисту даних і ключів криптографічних алгоритмів від їх несанкціонованої реконструкції за допомогою інженерно-криптографічних атак на основі аналізу зміни споживаної потужності при реалізації цих алгоритмів у криптографічних операційних блоках термінальних обчислювальних пристроїв комп'ютерних систем за рахунок побудови їх структур на основі операцій над даними у маскованому представленні із довільною кількістю масок.

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

– проаналізувати можливості підвищення рівня захисту інформації від інженерно-криптографічних атак на основі аналізу споживаної потужності при її обробці компонентами шифрування з використанням маскованого представлення у сучасних системах захисту інформації та постановка задач дослідження;

– розробити методи виконання базових операцій для криптографічних операційних блоків, які оперують даними у маскованому представленні: логічних (кон'юнкції, диз'юнкції) із довільною кількістю масок, арифметичних (обчислення зворотнього елемента за модулем 2^N, додавання за модулем 2^N), табличних операцій, операцій перетворення маскованого представлення даних;

– на основі методів виконання базових операцій над даними у маскованому представленні, розробити структури криптографічних операційних блоків, масштабованих до кількості масок та дослідити їх характеристики складності при їх апаратній реалізації;

– на основі структур криптографічних операційних блоків розробити та експериментально дослідити програмні моделі ядер спеціалізованих апаратно-орієнтованих процесорів симетричного блокового шифрування даних у маскованому представленні.

Об’єктом дослідження є процеси виконання арифметичних та логічних операцій у апаратних криптографічних операційних блоках на термінальних обчислювальних пристроях комп’ютерних систем і мереж.

Предметом дослідження є алгоритми, методи, моделі, засоби виконання арифметичних та логічних операцій над даними у маскованому представленні для пристроїв захисту інформації.

Методи дослідження базуються на основі використання теорій множин, ймовірності та математичної статистики, математичної логіки, на основі методів математичного моделювання алгоритмів та експериментального дослідження прототипів апаратних засобів.

Наукова новизна одержаних результатів полягає у наступному:

– вперше запропоновано метод виконання операції диз’юнкції над даними у маскованому представленні, що, за рахунок обчислення функції корекції маски результату з використанням виключно даних у маскованому представленні та їх масок, дозволяє використати таку операцію для побудови структур криптографічних операційних блоків виконання операції диз’юнкції, масштабованих до кількості масок даних у їх маскованому представленні;

– вперше запропоновано метод перетворення маскованого представлення даних, що, за рахунок використання операції додавання за модулем 2^N над даними у маскованому представленні, побудованої на основі маскованих логічних операцій, дозволяє перетворювати масковане представлення даних із арифметичним маскуванням у дані із логічним маскуванням та навпаки, а також використати таке перетворення для створення структур криптографічних операційних блоків, які використовують масковане представлення даних як з логічною, так і з арифметичною маскою;

– отримав подальший розвиток метод виконання операції кон’юнкції над даними у маскованому представленні, що, за рахунок введення у функцію корекції маски результату обчислень з урахуванням усіх масок вхідних та вихідних даних, дозволяє використати таку операцію для побудови структур

криптографічних операційних блоків виконання операції кон'юнкції, масштабованих до кількості масок даних у їх маскованому представленні;

– отримав подальший розвиток метод інвертування даних у маскованому представленні у полях виду $GF(2^N)$, що, за рахунок введення у функцію корекції маски результату обчислень з урахуванням усіх масок вхідних та вихідних даних, дозволяє обробляти дані із довільною кількістю масок, а також використати таке перетворення для побудови структур криптографічних операційних блоків інвертування даних у полях виду $GF(2^N)$, які використовують табличні методи виконання операцій у цих полях;

– удосконалено метод табличних перетворень даних у маскованому представленні, що, за рахунок введення додаткового проміжного маскування із узгодженим типом маски вхідних даних, дозволяє виконувати табличні перетворення над вхідними даними як із логічною, так і з арифметичною масками та отримувати результат із заданим типом маскування, а також дозволяє використати таку операцію для побудови структур криптографічних операційних блоків табличної заміни засобів шифрування даних у маскованому представленні.

Практичне значення одержаних результатів

Отримані в дисертаційній роботі результати можуть бути використані для розширення варіантів побудови криптографічних ОБ апаратних або програмних систем захисту інформації, які, за рахунок обробки даних у МП, володіють підвищеною стійкістю до інженерно-криптографічних атак на основі аналізу споживаної потужності. Практична цінність роботи полягає у такому:

– створені програмні Verilog-моделі структур криптографічних ОБ виконання операцій кон'юнкції та диз'юнкції, додавання за модулем 2^N, пошуку інвертованого елемента у полі $GF(2^N)$ для даних у МП із довільною кількістю логічних масок, орієнтованих на подальше використання при створенні та дослідженні ядер спеціалізованих апаратно-орієнтованих криптографічних процесорів, що підтверджується актом про їх використання у науково-дослідних

роботах Тернопільського національного економічного університету (акт від 18.06.2015);

– створені програмні Verilog-моделі ядер спеціалізованих апаратно-орієнтованих процесорів симетричного блокового шифрування, які обробляють дані із одною логічною маскою та володіють підвищеною стійкістю до атак на основі аналізу споживаної потужності, що підтверджується актом про впровадження у діяльність Інституту передових технологій Самсунг Електронікс (Республіка Корея) (акт від 11.01.2011);

– розроблені алгоритми оцінки характеристик складності криптографічних блоків для виконання операцій кон'юнкції, диз'юнкції, додавання за модулем 2^N, табличних операцій, перетворення МП даних, пошуку інвертованого елемента у полі $GF(2^N)$ впроваджені у начальний процес підготовки фахівців у галузі інформаційної безпеки, що підтверджується актами про впровадження у навчальний процес Університету в Бєльсько-Бялій (Польща) (акт від 30.06.2015), Тернопільського національного економічного університету (акт від 18.06.2015); Тернопільського національного технічного університету імені І. Пулюя (акт від 21.06.2016).

Теоретичні та практичні результати дисертаційної роботи використані та впроваджені: при виконанні науково-дослідної роботи "Методи та засоби реалізації алгоритмів захисту інформації стійких до атак на реалізацію" та «Паралельні методи та засоби реалізації алгоритмів захисту інформації в комп'ютерних мережах з використанням математичного апарату еліптичних кривих», у роботах, які проводилися в Інституті передових технологій Самсунг (м. Сувон, Республіка Корея) в рамках виконання проекту "Засоби обробки даних у маскованому представленні", а також у навчальному процесі Тернопільського національного економічного університету (ТНЕУ) при викладанні дисциплін "Основи захисту інформації" та "Автоматизоване проектування комп'ютерних систем", Університету в Бєльсько-Бялій (УББ) (Польща) в курсі "Безпека інформаційних технологій", згідно з Угодою про

співробітництво між УББ та ТНЕУ, в Тернопільському національному технічному університете імені Пулюя при викладанні дисципліни «Комплексні системи захисту» курсовому, дипломному проектуванні та при написанні кваліфікаційних робіт за навчальними планами з напряму підготовки фахівців з інформаційної безпеки.

Особистий внесок здобувача. Усі положення, які становлять суть дисертаційної роботи, були сформульовані та вирішені автором самостійно.

У друкованих працях, опублікованих у співавторстві, автору дисертації належать: [1, 2] – розроблені статистичні моделі двомісних логічних операцій та операції додавання за модулем 2^N для атак за аналізом енергоспоживання; [3] – алгоритм атаки на основі аналізу енергоспоживання на програмно-апаратні реалізації криптографічного перетворення за чинним стандартом; [4] – методика рандомізованого виконання криптографічних перетворень з регулярною структурою; [5, 6] – узагальнена архітектура комп'ютерних компонентів для обробки даних у МП; [7] – теоретичні оцінки рівня безпеки виконання арифметичних, логічних операцій, операцій перетворення МП даних та табличних перетворень над даними у МП; [8] – методи перетворення МП даних; [9] – метод виконання операції підстановки над даними у МП; [10] – метод пошуку оберненого елемента у полях $GF(2^N)$ над даними у МП; [11] – метод виконання операції додавання за модулем 2^N над даними у МП; [12] – алгоритм криптографічного перетворення mCrypton над даними у МП, архітектура та Verilog модель процесора за цим алгоритмом; [13] – модифікований алгоритм криптографічного перетворення за ГОСТ 28147-89 над даними у МП, архітектура та Verilog модель процесора за цим алгоритмом, [120, 121] – аналіз стійкості реалізацій алгоритмів симетричного блокового шифрування до атак на основі АСП; [122] – огляд методів зворотнього трасування адрес; [126] – система тестування, програмні засоби обробки даних моделювання атаки.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідалися і обговорювалися на 9-ти науково-технічних

та міжнародних конференціях: International Workshop on Information Security Applications (WISA), Південна Корея, 2004, міжнародної конференція “Комп’ютерні науки та інформаційні технології” (CSIT), м. Львів, 2006, міжнародна конференція “Комп’ютерні науки та інженерія” (CSE), м. Львів, 2006, 2007, Науковій конференції Тернопільського державного технічного університету ім. І.Пулюя, м. Тернопіль, 2007, International Science Conference "Internet in the information society" (IIS), м. Домрова Гурніца, Польща, 2007, International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), м. Дортмунд, Німеччина, 2007, м. Варшава, Польща, 2015, International Conference "Advanced Computer Systems and Networks: Design and Application" (ACSN), м. Львів, International conference on modern problems of telecommunication, computer science and engineering training (TCSET), Львів-Славське, 2008.

Публікації. Основні положення дисертації опубліковано у 21 науковій праці, у тому числі 2 розділи у закордонних монографіях, 10 статей у наукових журналах та збірниках наукових праць, які входять до переліку фахових наукових видань МОН України (серед них 2 статті у виданнях, що входять до міжнародних наукометричних баз даних), а також 9 тез доповідей і матеріалів конференцій.

Структура роботи та її обсяг. Дисертація складається зі вступу, чотирьох розділів, загальних висновків, додатків, списку використаних джерел і має 144 сторінки основного тексту, 39 рисунків, 9 таблиць, 11 сторінок додатків. Список використаних джерел містить 126 найменувань і займає 16 сторінок. Загальний обсяг роботи 176 сторінок.

РОЗДІЛ 1
СУЧАСНІ МЕТОДИ ТА ЗАСОБИ ЗАХИСТУ КОМП'ЮТЕРНИХ КОМПОНЕНТІВ ШИФРУВАННЯ В УМОВАХ ПРОВЕДЕННЯ АТАК НА ОСНОВІ АНАЛІЗУ СПОЖИВАНОЇ ПОТУЖНОСТІ

1.1 Сучасні інженерно-криптографічні атаки на комп'ютерні засоби захисту даних

Сучасні системи захисту інформації використовують криптографічні перетворення для забезпечення конфіденційності, цілісності й автентифікації даних. Криптографічні перетворення є математичними функціями, які загалом мають два параметри: відкрите повідомлення та криптографічний ключ [14 – 26]. Алгоритм криптографічного перетворення відображає ці два параметри у результат, званий шифртекстом. Такий процес перетворення називають зашифруванням. У сучасній криптології припускається, що алгоритм криптографічного перетворення є відомим, а єдиною конфіденційною інформацією є ключ зашифрування, який не розголошується [27, 28, 29, 30]. Розрізняють алгоритми симетричних й асиметричних криптографічних перетворень. Для використання алгоритмів симетричних криптографічних перетворень два учасники сеансу зв'язку повинні володіти однаковим конфіденційним ключем. Прикладами таких алгоритмів криптографічних перетворень є ГОСТ 28147-89, AES, DES, mCrypton, Спектр, тощо.

У алгоритмах асиметричних криптографічних перетворень використовують пари ключів. Такі пари складаються з відомого відкритого ключа та конфіденційного ключа. Прикладами цих алгоритмів є RSA, Діффі-Хелмана, тощо.

Важливою властивістю сучасних алгоритмів криптографічних перетворень є їх ефективна реалізація на основі комп'ютерних платформ. Однак, така реалізація потребує зберігання ключів шифрування у пристроях на основі цих платформ, що, в свою чергу, вимагає спеціального налаштування цих пристроїв. Наприклад, персональні комп'ютери є найменш доцільною

платформою для зберігання ключів шифрування, оскільки віруси та інші зловмисні програми, які розповсюджуються через комп'ютерні мережі, можуть отримати доступ до цих ключів [31, 32, 33, 34, 35]. Тому, так звані "замкнені" комп'ютерні архітектури є доцільнішими засобами зберігання криптографічних ключів та виконання криптографічних перетворень. Прикладом пристрою на основі таких замкнених комп'ютерних архітектур є смарт-карта, яка не дозволяє встановлювати довільне програмне забезпечення, не під'єднана до комп'ютерних мереж. Сюди можна віднести також USB-токени, безконтактні теги ідентифікування (RFID), тощо. "Криптографічним пристроєм" будемо називати деякий електронний засіб, в якому зберігаються ключі шифрування та реалізуються алгоритми криптографічних перетворень з використанням цих ключів. Також криптографічні пристрої можуть приймати вхідні дані та видавати результати обчислень.

Той факт, що криптографічні пристрої використовуються для виконання алгоритмів криптографічних перетворень, зумовив появу нових задач у забезпеченні стійкості алгоритмів криптографічних перетворень в обчислювальному сенсі: необхідно враховувати безпеку усієї системи захисту інформації , а не лише безпеку її компонента у вигляді математичного опису алгоритму криптографічного перетворення. Злам криптографічного пристрою означає отримання ключа шифрування з цього пристрою. Особу, яка намагається отримати ключ шифрування з криптографічного пристрою недозволеним шляхом, будемо називається порушником. Довільну спробу отримання ключа шифрування з криптографічного пристрою техніко-аналітичним шляхом називатимемо "інженерно-криптографічною" атакою. З точки зору інформації, яка відома порушнику, приймемо, що він володіє відомостями про усі інженерні особливості реалізації алгоритму криптографічного перетворення.

Криптографічні пристрої будуються на основі комп'ютерних платформ та складаються з кількох компонентів. Кожний з цих компонентів виконує задану функцію: виконання алгоритму криптографічного перетворення, збереження

ключів шифрування, керування процесом обчислень, тощо. В свою чергу ці компоненти можуть реалізовуватись у вигляді довільних комбінацій апаратних та програмних засобів (рис. 1.1).

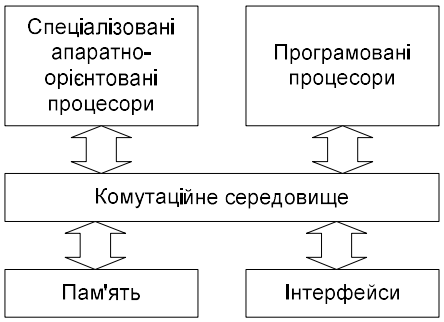


Рис. 1.1. Узагальнена архітектура криптографічного пристрою

Спеціалізовані апаратно-орієнтовані процесори забезпечують апаратну реалізацію алгоритмів криптографічних перетворень, наприклад, повну апаратну реалізацію алгоритму криптографічного перетворення згідно з ГОСТ 28147-89.

Програмовані процесори, з універсальною чи спеціалізованою системою команд та архітектурою, реалізують алгоритми криптографічних перетворень шляхом виконання набору інструкцій. Програмне забезпечення містить програмні реалізації необхідних алгоритмів криптографічних перетворень та протоколів. Як правило, програмне забезпечення виготовляється для конкретного програмованого процесора. Пам'ять криптографічного пристрою використовується для зберігання початкових, проміжних та остаточних результатів виконання алгоритмів криптографічних перетворень. Крім цього, пам'ять застосовується для зберігання програмного забезпечення та ключів шифрування. Інтерфейси криптографічного пристрою призначені для обміну даними між криптографічним та іншими пристроями. Крім цього, інтерфейси використовують для конфігурування режимів роботи пристрою. Обмін інформацією між компонентами здійснюється через комутаційне середовище.

Криптографічний пристрій можна реалізувати на одному кристалі, чи декількох кристалах інтегральних схем. Прикладами однокристальних криптографічних пристроїв є смарт-карти та USB-токени. Інтегральні схеми для

криптографічних пристроїв будують у вигляді напів- або замовлених інтегральних схем чи програмованих логічних пристроїв. У обох випадках найчастіше використовуваною технологією виготовлення цих інтегральних схем є комплементарна метал-оксидна напівпровідникова (КМОН) технологія.

Аналіз результатів досліджень у галузі інженерно-криптографічних атак свідчить, що ці атаки суттєво відрізняються за задіяними ресурсами часу, коштів, обладнання та рівнем досвідченості порушника. Для охарактеризування цієї області дисертаційної роботи скористаємося загальним підходом до класифікації інженерно-криптографічних атак на криптографічні пристрої (табл. 1.1) [36].

Інженерно-криптографічні атаки на криптографічні пристрої Таблиця 1.1

Міра доступу до внутрішніх компонент та інтерфейсів	**Метод впливу**	
	Активний	**Пасивний**
Необмежений прямий доступ	Модифікування структури пристрою	Пряме зчитування даних
Обмежений прямий доступ	Введення помилок (гамма-випромінювання, електромагнітні поля, світло)	Оптичне зчитування
Непрямий доступ	Введення помилок (маніпуляція синхронізацією, маніпуляція напругою живлення, зміна температури)	• АСП • Часовий аналіз • Аналіз електромагнітного випромінювання

Першим критерієм класифікації є метод впливу на криптографічний пристрій зі сторони порушника:

- пасивні атаки – при проведенні пасивних інженерно-криптографічних атак криптографічний пристрій функціонує, здебільшого, згідно з заданою

специфікацією. Конфіденційний ключ отримується шляхом аналізу результатів спостережень за роботою пристрою;

- активні атаки – при проведенні активних інженерно-криптографічних атак криптографічний пристрій, зокрема його вхідні дані, а також оточення, в якому працює цей пристрій, модифікуються з метою внесення порушень у його дію. Конфіденційний ключ отримується шляхом використання результатів обчислень пристрою, який працює з порушеннями.

Іншим критерієм класифікації інженерно-криптографічних атак є міра доступу до компонент та інтерфейсів криптографічних пристроїв:

- атаки з необмеженим доступом – це найнебезпечніші інженерно-криптографічні атаки на криптографічні пристрої. Для їх проведення у порушника є усі засоби та можливості маніпулювання криптографічним пристроєм, включаючи доступ до структури пристрою, в тому числі пряме зчитування даних, маніпулювання структурою, внесення нових компонент [37, 38];
- атаки з обмеженим доступом – на відміну від атак з необмеженим доступом, ці атаки не використовують прямого контакту з провідними елементами кристалу криптографічного пристрою. Метою таких пасивних атак є зчитування вмісту пам'яті пристрою, а активних – вплив на роботу пристрою шляхом введення помилок у його роботі за допомогою гамма-випромінювання, електромагнітного поля, світла [37, 39, 40, 41];
- непрямі атаки проводяться шляхом використання лише доступних інтерфейсів пристрою без внесення порушень у процес функціонування криптографічного пристрою, наприклад, часу виконання обчислень [42], електромагнітного випромінювання [43], тощо.

З точки зору низької вартості проведення та високої ефективності атакування, пасивні непрямі атаки на основі АСП пристрою є найпривабливішими для порушника. Тому найімовірнішою є здійснення загрози проведення цих атак на криптографічні пристрої [44].

1.2 Аналіз моделей атак на основі аналізу споживаної потужності

Основою атак на підставі АСП пристрою є факт, що СП криптографічного пристрою у деякий момент часу залежить від даних, які обробляються, та операцій, які проводяться тоді над цими даними. Така залежність характерна для пристроїв, виготовлених на основі технології КМОН та інших технологій [45]. Особливості КМОН технології зумовлюють незначні значення СП для логічних елементів у статичному режимі та суттєві значення СП у динамічному режимі, коли зміна вхідних даних призводить до зміни вихідних даних. Отже, при перемиканні стану логічного елемента $0 \to 0$ чи $1 \to 1$ домінуючою складовою його СП є СП у статичному режимі. А при перемиканні стану елемента зі станів $0 \to 1$ чи $1 \to 0$ СП елемента визначається як сума споживаних потужностей у статичному та динамічному режимах [46]. Оцінку СП криптографічних пристроїв проводять на етапі їх проектування. Для цього використовують аналоговий чи логічний (поведінковий) рівень оцінювання відповідної моделі пристрою. Оцінки на обох рівнях враховують "гонки" у комбінаційних схемах та відрізняються складністю проведення. Необхідно зазначити, що порушник, загалом, не має у розпорядженні точної моделі СП криптографічного пристрою на аналоговому чи логічному рівнях. Однак, для нього важливо знати не точне значення СП пристрою, а різницю між споживаними потужностями пристрою при обробці різних вхідних даних. Для цього порушник має у розпорядженні мінімум дві моделі для оцінки СП при обробці різних даних з подальшим визначенням різниці у цьому споживанні: модель на основі відстані Хемінга та модель на основі ваги Хемінга.

Модель на основі відстані Хемінга ґрунтується на залежності СП від кількості перемикань у комбінаційній схемі. Тому кількість перемикань за заданий інтервал часу використовується для оцінки СП протягом нього. Шляхом поділу усього часу роботи пристрою на малі інтервали можна отримати оцінку зміни СП у часі. Більш строго, така оцінка відображає зміну кількості перемикань у часі. При цьому припускається, що: а) логічний елемент споживає

однакову потужність при станових переходах $0 \to 1$ та $1 \to 0$; б) модель не враховує паразитних ємностей між провідниками та логічними елементами; в) усі логічні елементи вносять однаковий вклад у СП; г) ігнорується складова СП у статичному режимі. Хемінгова відстань між двома даними v_0 і v_1 зводиться до обчислення Хемінгової ваги їх суми за модулем два: $HD(v_0, v_1) = HW(v_o \oplus v_1)$. У роботі [47] показано, що модель на основі Хемінгової відстані адекватно описує зміну СП внутрішніх шин та регістрів криптографічного пристрою. При цьому необхідною умовою є володіння порушником відомостями про дані, які послідовно записувалися у регістр, чи передаються шинами обміну.

Зовсім інакше поступають, коли порушник володіє відомостями лише про поодинокі дані, які передаються шиною чи записуються у регістри і не володіє відомостями про попередні чи наступні дані. В цьому випадку модель на основі відстані Хемінга не застосовується, оскільки її результати моделювання неадекватно відображають СП пристрою у заданих проміжках часу. Тоді для оцінки СП використовується модель на основі ваги Хемінга. Згідно з цією моделлю порушник припускає, що СП пристрою пропорційна до кількості одиничних бітів, що містяться у даних, які обробляються. Однак, з іншого боку, така модель дещо наближено описує СП КМОН пристроїв. На практиці Хемінгова вага даних не повністю визначає споживану пристроєм потужність при обробці цих даних. Розглянемо три базових випадки для оцінки СП пристроєм, який послідовно обробляє дані v_0, v_1 і v_2. При цьому ставиться за мету отримати оцінки СП при обробці v_1 без відомостей про v_0 або v_2. Оскільки v_1 використовується у двох станових переходах, то $v_0 \to v_1$ і $v_1 \to v_2$. Розглянемо ці випадки лише для переходу $v_0 \to v_1$. Для переходу $v_1 \to v_2$ міркування будуть подібними.

Випадок 1. Біти v_0 є однаковими та незмінними в моменти виникнення переходу $v_0 \to v_1$. Таким прикладом є шина даних, якою передається завжди однакове число $v_0 = 0$ перед передаванням числа v_1. Тоді модель Хемінгової

ваги є еквівалентна до моделі Хемінгової відстані, тобто $HD(v_0, v_1) = HW(v_0 \oplus v_1) = HW(v_1)$. Якщо ж всі біти v_0 дорівнюють одиниці, то $HD(v_0, v_1) = HW(v_0 \oplus v_1) = n - HW(v_1)$. Таким чином, у цих випадках результати моделювання пропорційні до СП. Тому розглянуті моделі є еквівалентними щодо проведення атак за однакових бітів v_0 перед виникненням переходу $v_0 \to v_1$.

Випадок 2. Біти v_0 є незмінними, однак неоднаковими та невідомими для порушника. На відміну від першого випадку, тут можна розглядати лише один біт переходу $v_0 \to v_1$. Тоді розглянуті дві моделі СП є еквівалентними, виходячи з аналогічних міркувань, до першого випадку. Однак це справджується лише для одного біта. СП, викликана зміною деякого біта v_1 , прямо або обернено пропорційно залежить від значення цього біта за умови, що цей біт у v_0 завжди встановлюється у однакове значення перед виникненням переходу $v_0 \to v_1$. Із ростом кількості бітів у v_0, які залишаються незмінними, зростає залежність Хемінгової ваги v_1 від кількості бітових переходів, тобто порушник отримує повнішу інформацію.

Випадок 3. Біти v_0 є рівномірно розподіленими і незалежними від v_1 . Також біти v_0 не є сталими, а випадковими для кожного початку роботи криптографічного пристрою з метою виконання алгоритму криптографічного перетворення. Тому $HW(v_1)$ є незалежною від $HW(v_0 \oplus v_1)$, якщо v_0 є незалежним від v_1 та володіє рівномірним розподілом ймовірності. Відповідно, результати симулювання, основані на цих двох моделях, не можна використовувати для проведення атак порушником.

Розглянуті випадки не вичерпують повного переліку варіантів відношень між бітами v_0 і v_1. Разом з тим, відомо, що, на відміну від теоретичної моделі, при виконанні переходів $0 \to 1$ та $1 \to 0$ у пристроях СП дещо відрізняється. Тому СП при обробці даних із більшою Хемінговою вагою є вищою, ніж при обробці даних із меншою Хемінговою вагою, що зумовлює успішне застосування на практиці згаданих двох моделей. Оскільки модель на основі Хемінгової відстані,

на відміну від моделі на основі Хемінгової ваги, потребує більше відомостей про елементи структури криптографічного пристрою, то модель на базі Хемінгової ваги є прийнятнішою для порушника.

Розглянемо приклад атакування N-розрядної операції додавання за модулем два, вперше запропонований у [48]. Нехай СП у момент часу j представлено у вигляді $P[j]$. Для моделювання каналу витоку інформації у сигналі $P[j]$ скористаємося лінійною залежністю:

$$P[j] = \varepsilon \cdot d[j] + L + n \qquad (1.1)$$

де $d[j]$ репрезентує Хемінгову вагу результату, який отримується у момент часу j, ε – вклад у СП кожної одиниці Хемінгової ваги даних, L – споживана постійна загальна потужність, n – шум з нульовим середнім значенням.

Нехай j означає момент часу, коли виконується операція додавання за модулем 2. Тоді сума $S = K \oplus P$, причому K – N-бітовий невідомий доданок, P – це N-бітовий відкритий текст. Розглянемо атаку, запропоновану в [48], на N-бітовий суматор за модулем 2, метою якої є визначення бітів K без відомостей про значення бітів S. Припустимо, що залежність між СП у момент часу j і Хемінговою вагою результату, який отримується, описується виразом (1.1). Тоді узагальнений алгоритм атаки на реалізацію операції додавання за модулем 2 є таким:

```
Для і від 0 до N-1 {
Для b=0 до 1 {
      Обчислити усереднене значення сигналу споживаної потужності Ab[j] {
      Встановити і-й біт P рівним b;
      Встановити решту бітів P у випадкові значення;
      Зібрати дані про споживану потужність пристрою;
      }
   }
   Обчислити диференційний сигнал T[j] = A0[j] – A1[j];
   Якщо T[j] > 0, то і-й біт K є "1", якщо T[j] < 0 то і-й біт K є "0";
}
```

Результативність цієї атаки базується на незалежності очікуваного значення Хемінгової ваги результату додавання за модулем 2 від позиції біту, який піддається аналізу. Диференційний сигнал буде містити додатній пік за умови $k_i = 1$ і від'ємний пік за умови $k_i = 0$.

Аналогічно будуються атаки на основі статистичних моделей атакованих операцій, наприклад описані в [1, 3], які використовуються для подальшого атакування криптографічних пристроїв. У відкритих джерелах опубліковано декілька типів атак на пристрої на основі диференційного АСП – так званих DPA атак. Серед них: DPA атака на основі різниці середніх [48], DPA атака на основі відстані середніх [47], узагальнене тестування на найбільшу вірогідність [123], DPA атака на основі зразків [124]. Разом з тим, базовим методом атак залишається DPA атака на основі аналізу кореляційних коефіцієнтів [125] (додаток Б). Тому, для оцінки захищеності комп'ютерних компонент обробки даних у МП на етапі їх проектування було обрано саме цю атаку [126].

Розглянуті атаки, що ґрунтуються на витоку інформації із одного проміжного результату належать до атак на основі АСП першого порядку. Якщо у порушника є змога обрати та відслідкувати появу витоку інформації з кількох проміжних результатів, то порушник може покращити результати атакування за допомогою аналізу одразу кількох проміжних результатів. Якщо ж порушник одночасно аналізує n проміжних результатів, то така атака носить назву атаки на основі АСП n-го порядку.

1.3 Аналіз методів і засобів захисту від атак на основі аналізу споживаної потужності

Метою захисту від проведення атак на основі АСП є модифікування СП криптографічного пристрою таким чином, щоб вона не залежала від проміжних даних, які обробляються. Аналіз таких опублікованих методів захисту дозволяє виділити у них три основні групи для досягнення означеної мети (рис. 1.2).

Перша група методів полягає у використанні неоднакових ключів шифрування для різних сесій обробки даних обмеженого обсягу у криптографічному пристрої. Тоді порушник володіє обмеженим обсягом даних, які він може використати для проведення атаки. Обсяг необхідних даних обирається таким чином, щоб його не вистачало для успішного атакування криптографічного пристрою. Недоліком наведеної групи методів є обмеженість її застосування, оскільки лише невелика кількість криптографічних протоколів підтримує часту зміну ключів шифрування.

Друга група методів зводиться до усунення залежності значення СП від значень даних, які обробляються. Реалізація такого методу полягає у:

- рандомізуванні виконання алгоритму (часове вимірювання) шляхом виконання базових операцій алгоритму у різні (рандомізовані) моменти часу;
- такій зміні СП елементів при обробці різних даних (амплітудне вимірювання), при якій задача виявлення цих залежностей для порушника значно ускладнюється.

Рандомізування виконання алгоритму криптографічного перетворення досягається двома способами: випадковою зміною шляху виконання алгоритму та перемішуванням операцій. Для випадкової зміни шляху виконання алгоритму перед кожним його виконанням генерується випадкове число і на його основі обираються місця вставляння порожніх операцій та кількість порожніх операцій. Особливістю цього способу є зменшення продуктивності обробки даних внаслідок збільшення критичного шляху алгоритму криптографічного перетворення із-за додаткових порожніх операцій. Прикладами реалізації цього способу є [49, 50]: вставляння порожніх операцій між операціями обробки даних, вставляння порожніх циклів очікування, випадковий пропуск тактових імпульсів, випадкова зміна значення частоти тактових імпульсів, випадкове перемикання між декількома тактовими сигналами з різною частотою [51].

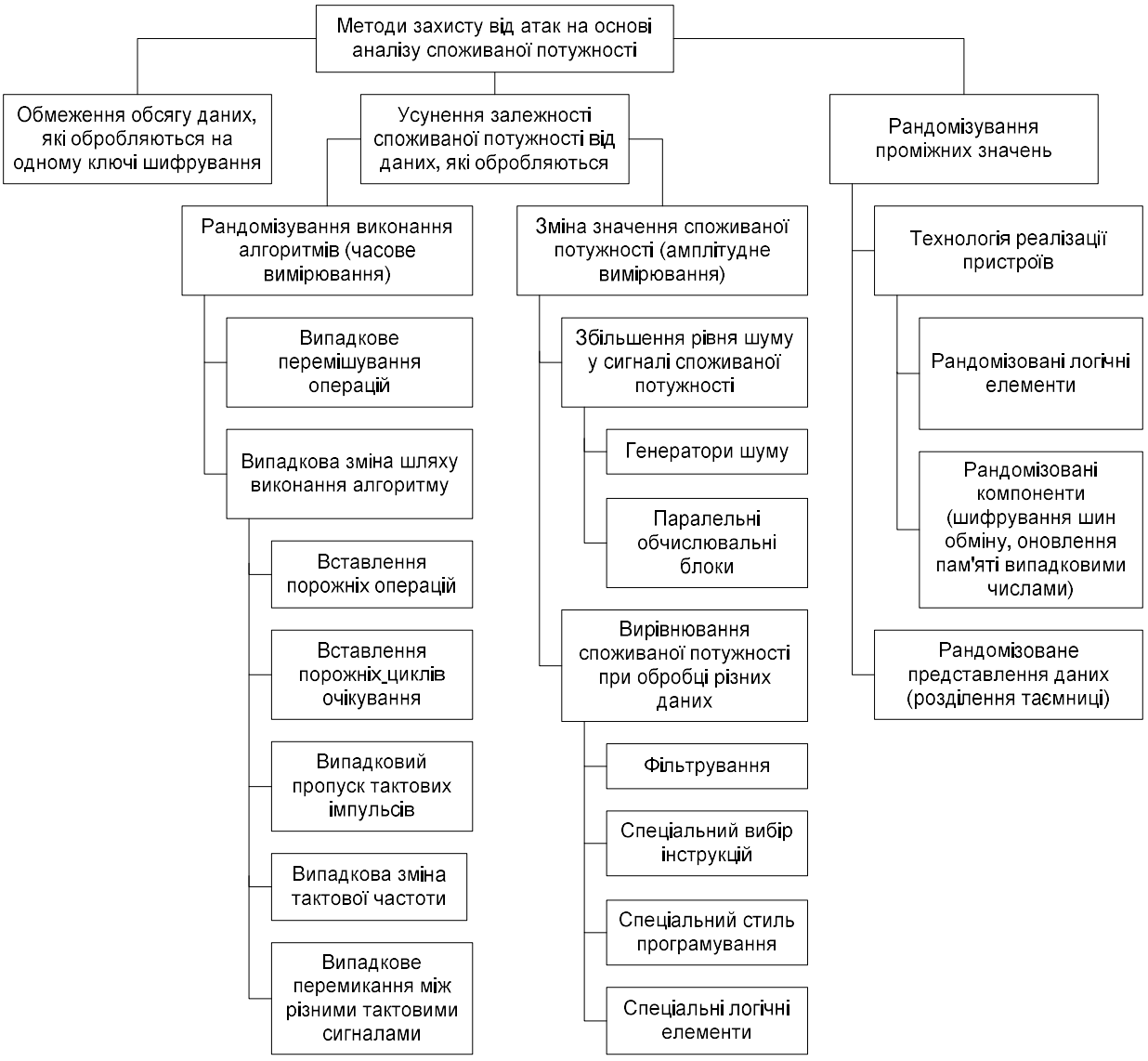


Рис. 1.2. Класифікація методів захисту від атак на основі АСП

Перемішування операцій алгоритму криптографічного перетворення полягає у випадковій зміні порядку виконання складових операцій алгоритму [4, 52]. При цьому критичний шлях алгоритму залишається незмінним. Особливістю застосування цього способу є залежність кількості варіантів перемішування від структури чи рівня паралелізму алгоритму криптографічного перетворення. На практиці використовується комбінування розглянутих двох способів.

У амплітудному вимірюванні зміна СП досягається такими способами [53, 54, 55]: збільшенням рівня шуму у сигналі про СП та вирівнюванням СП і

компонент при обробці різних даних. Рівень шуму у сигналі про СП збільшують за допомогою під'єднання генератора шуму до ліній живлення пристрою, використання додаткових паралельно працюючих обчислювальних блоків, які обробляють випадкові дані [56, 57].

Вирівнювання СП компонент криптографічного пристрою досягається шляхом: а) використання спеціально підібраного набору інструкцій процесора, у яких є збалансована вага Хемінга для виконання операцій над різними даними; б) застосування спеціального стилю програмування – уникнення умовних переходів, які використовують елементи ключа шифрування як умову переходу, уникнення генерування адрес пам'яті, які залежать від елементів ключа шифрування, тощо; в) побудови криптографічного пристрою на базі спеціальних DPR (Dual-Rail Pre-charge) логічних вентилів із вирівняним споживанням потужності, наприклад, елементів SABL (Sense Amplifier Based Logic) [58], WDDL (Wave Dynamic Differential Logic) [59], DSDR (Dual-Spacer Dual-Rail) [60, 61, 62] та їх варіантів TDPL (Three-Phase Dual-Rail Pre-charge Logic) [63], 3sDL (3-state Dynamic Logic) [64]. Також до цього способу відносять фільтрування ліній живлення пристрою за допомогою активних і пасивних фільтрів [65, 66, 67] та використання асинхронної логіки, наприклад [68, 69, 70, 71, 72].

Аналіз методів усунення залежності СП від даних, які обробляються у криптографічному пристрої, показав, що ці методи не дозволяють надійно захистити ці засоби від атак на основі АСП. Однак, зазначені методи доцільно використовувати сумісно з іншими методами захисту для збільшення складності атакування криптографічних пристроїв.

До третьої групи методів відноситься рандомізування проміжних значень, які обробляються криптографічним пристроєм. Цей метод можна застосовувати на кількох рівнях структури криптографічного пристрою: на рівні способу виготовлення пристрою, на рівні проміжних результатів які отримуються в процесі роботи пристрою. На рівні способу реалізації криптографічного пристрою застосовують рандомізовані логічні елементи та рандомізовані

компоненти криптографічного пристрою. На рівні проміжних результатів процесу виконання алгоритму криптографічного перетворення використовують МП даних на основі технології розділення таємниці [73, 74, 75].

Основою рандомізування є подання даних чи сигналів a у МП у вигляді пари $\{\breve{a}, x\}$, причому $\breve{a} = a \operatorname{op} x$, op – бінарна операція, яка належить скінченому полю чи кільцю, x – випадкове число з рівномірним розподілом ймовірності, незалежне від a. Дані a називають відкритими даними, $\breve{a}$ – замаскованими даними, а пара $\{\breve{a}, x\}$ – це дані у МП та x – маска даних, причому операція "op" – операція маскування. Маска x генерується всередині криптографічного пристрою, наприклад, за допомогою генератора випадкових чисел, та є невідомою порушнику. Вибір операції маскування "op" залежить від операцій, які використовуються в алгоритмі криптографічного перетворення. Типовими операціями маскування є додавання за модулем два, додавання за модулем 2^N, множення за модулем. Модуль операції маскування обирається залежно від заданого алгоритму криптографічного перетворення.

Беручи до уваги, що специфікації алгоритмів криптографічного перетворення створені для опису процесу обробки немаскованих даних, то для обробки даних у МП ці специфікації повинні бути дещо зміненими. Необхідні зміни у специфікацію алгоритму вносять для того, щоб врахувати факт подання оброблюваних даних у маскованому вигляді. З метою захисту від атак на основі АСП усі проміжні дані, які утворюються в процесі обробки даних, теж подають у МП, а процес обробки даних повинен бути організованим таким чином, щоб уникнути використання чи появи немаскованих даних на будь-якому етапі обчислень. Паралельно з обробкою даних у МП обробляють і саму маску, тобто здійснюючи так званий процес корекції маски. Після виконання обчислень з використанням конфіденційної інформації, дані переводяться із МП у немасковане шляхом виконання операції маскування над даними у МП та оберненим елементом маски, тобто $a = \breve{a} \operatorname{op} x^{-1}$, де x^{-1} є оберненим елементом

маски x . У загальному випадку, маскування даних може відбуватися з використанням будь-якої кількості масок на основі техніки розділення таємниці.

Рандомізовані логічні елементи використовують для створення компонентів криптографічних пристроїв, які обробляють дані з використанням конфіденційної інформації. Робота рандомізованих логічних елементів ґрунтується на застосуванні методу маскування до електричних сигналів, які обробляються. Прикладом таких логічних елементів, що отримали назву "масковані логічні елементи", є MDPL (Masked Dual-Rail Pre-charge Logic) [76, 77]. Елементи MDPL використовують одну маску для обробки усіх вхідних даних у МП. Альтернативні варіанти елементів MDPL були розроблені у [78]. Перевагою цих елементів є їх стійкість до гонок сигналів, що в свою чергу дозволяє уникнути залежності СП від оброблюваних даних [79]. Однак недоліком використання елементів MDPL є необхідність розробки криптографічного пристрою на основі напівзамовлених інтегральних схем із спеціальними бібліотеками елементів, що призводить до значного збільшення вартості цих пристроїв. Альтернативні підходи до побудови маскованих логічних елементів на базі стандартних бібліотечних компонентів були запропоновані у [80] для апаратної і програмної реалізації та розвинуті у [81] для апаратної реалізації. У [82] запропонований варіант маскованих логічних елементів, однак використана для їх побудови базова модель не враховує можливість виникнення гонок.

Рандомізовані апаратні компоненти криптографічних пристроїв використовують розглянутий вище принцип маскування. При цьому виділяються закінчені апаратні блоки, наприклад, маскований помножувач [83], масковані шини обміну даними [84, 85, 86], рандомізовані елементи пам’яті на основі записування випадкових значень на місця немаскованих даних [87]. Особливістю застосування рандомізованих апаратних компонентів є збільшення апаратної, часової та місткістної складності цих компонентів у порівнянні з їх немаскованими аналогами.

За критеріями *НВ* – низька вартість; *Т* – технологічність; *НЕ* – низьке енергоспоживання; *ВП* – висока продуктивність; *М* – масштабованість; *УЗ* – універсальність застосування (апаратна, програмна реалізації) відзначимо, що ці методи не є досконалими і мають певні обмеження щодо практичного застосування для розв'язання завдання побудови криптографічних пристроїв (табл. 1.2).

Таблиця. 1.2.

Результати порівння методів захисту від атак на основі АСП

Категорія	**Назва**	**Критерії**					
		НВ	Т	НЕ	ВП	М	УЗ
Протоколи обробки	Обмеження кількості вхідних даних для обробки з одним ключем	+	+	+	+	-	+
Рандомізоване у часі виконання алгоритмів	Перемішування операцій, зміна шляху виконання алгоритму, вставляння порожніх операцій/циклів очікування	+	+	-	-	-	+
	Маніпулювання сигналами синхронізації	-	+	+	+	-	-
Зміна значення споживаної потужності	Вирівнювання споживання: фільтрування, спеціальні інструкції процесора, спеціальна елементна база інтегральних схем (ІС)	-	-	-	+	-	-
	Внесення шуму: генерування шуму, паралельні обчислювальні блоки	-	+	-	+	-	-
	Ізольовані джерела живлення	-	-	-	+	+	-
Рандомізування проміжних даних	Рандомізовані компоненти системи на кристалі: шифровані шини обміну даних, запам'ятовуючі пристрої	-	+	-	-	-	+
	Спеціальна елементна база ІС	-	-	+	+	-	-
	Обробка даних у МП	+	+	+	-	-	+

Встановлено, що перспективний метод захисту від атак на основі аналізу залежності споживаної потужності пристрою повинен дозволяти будувати як програмні, так і апаратні засоби шифрування, не залежати від кількості даних, які обробляються, дозволяти реалізацію на існуючій технологічній базі із стандартними бібліотеками елементів ІС чи наборах команд процесора.

1.4 Аналіз методів обробки даних у маскованому представленні

Аналіз алгоритмів криптографічних перетворень, проведений у [88], показав, що структура та набір базових операцій алгоритмів криптографічних перетворень залежить від вибору рівня абстракції представлення цих алгоритмів. Враховуючи, що перелік алгоритмів криптографічних перетворень є доволі суттєвим за обсягом та включає в себе різнорідні алгоритми, оберемо для розгляду рівень елементарних базових операцій, які зустрічаються у переважній більшості цих алгоритмів. До складу цих операцій входять [88]:

- логічні операції Булевої алгебри логіки над двійковим представленням даних – логічне множення (кон'юнкція), логічне додавання (диз'юнкція), логічне заперечення та, додатково, операція еквівалентності (додавання за модулем два);
- операції маніпулювання бітами – перестановки бітів та циклічні зсуви;
- операції додавання у скінчених кільцях;
- операції додавання, множення та пошуку оберненого елемента у скінчених полях Галуа з характеристикою 2;
- операції заміни одного елемента даних на інший за допомогою таблиці.

Виконання перелічених операцій над даними у МП не є тривіальним та, загалом, потребує модифікування алгоритмів виконання цих базових операцій [5, 6]. При цьому, модифікування на алгоритмічному рівні визнано найдоцільнішим з точки зору вартості реалізації та стійкості проти інженерно-криптографічних атак. При цьому, виникає актуальна задача адаптування відомих алгоритмів криптографічних перетворень до обробки інформації у МП. Одним із шляхів розв'язання цієї задачі є побудова нових алгоритмів з врахуванням обробки даних у МП, які дозволяють отримувати аналогічні результати до початкових, в тому числі з використанням немаскованих даних. Однак, такий шлях пов'язаний з труднощами проведення математичного аналізу рівня безпеки нових алгоритмів криптографічного перетворення.

Альтернативний шлях полягає у розв'язанні цієї задачі шляхом адаптування початкових алгоритмів криптографічних перетворень до обробки даних у МП за допомогою заміни їх базових операцій на еквівалентні масковані базові операції. Для цього в алгоритмах виділяють базові та допоміжні операції та структуру зв'язків між цими операціями. Виділені операції замінюють їх маскованими еквівалентами, зберігаючи структуру зв'язків немаскованого алгоритму. Завдяки такому підходу уникають потребу в додатковому математичному аналізі рівня безпеки адаптованого алгоритму криптографічного перетворення, оскільки результати виконання маскованих еквівалентних операцій збігаються з результатами немаскованих операцій та порядок виконання операцій адаптованого алгоритму узгоджується з відповідним порядком початкового алгоритму. Адаптований алгоритм криптографічного перетворення до обробки даних у МП оптимізують з точки зору як досягнення заданих технічних характеристик, так і забезпечення заданого рівня безпеки обробки даних.

Нехай задано: 1) множину цілих чисел $Z_n = \{0,1,...,n-1\}$, $n = 2^l$, $l = 1,2,...$; 2) адитивну бінарну операцію "$+$" додавання за модулем n, яка разом з множиною Z_n утворює Абелеву групу G_+; 3) дві операції – адитивну "$\oplus$" і мультиплікативну "$\otimes$", які разом з Z_n утворюють поле $GF(2^l)$. Беручи до уваги введені в додатку А означення [7], які охоплюють типи використовуваних на практиці маскувань, нехай деяка базова операція алгоритму криптографічного перетворення обчислює результат z шляхом виконання операції op деякої скінченої адитивної Абелевої групи над двома аргументами a і b: $z = a \ \mathrm{op} \ b$. Відомо, що використання такої операції для маскування даних є достатнім для отримання рандомізованого представлення даних. Нехай для такого представлення використовуються дві маски x і y - незалежні випадкові числа з рівномірним розподілом ймовірності. Якщо op означує операцію маскування, то базова операція криптографічного алгоритму залишається незмінною:

$(\{\breve{a},x\})$ op $(\{\breve{b},y\}) = \{\breve{z},(x \text{ op } y)\}$. При цьому процедура корекції маски полягає у знаходженні x op y . Загалом, коли $z = f(a,b)$ і f є довільною функцією, результат обчислення повинен бути поданий у МП $\{\breve{z},u\}$, причому u – маска результату. Остання, в основному, дорівнює новому випадковому числу, незалежному від масок аргументів та самих аргументів, і володіє рівномірним розподілом ймовірності. Зауважимо, що у специфічних випадках маска результату може дорівнювати масці одного з аргументів. Тоді функцію f необхідно перетворити у деяку іншу функцію $\breve{f}$ з такими властивостями: $\{\breve{z},u\} = \{\breve{f}(\{\breve{a},x\},\{\breve{b},y\},u),u\} = \{f(\breve{a}\ op\ x,\breve{b}\ op\ y)\ op\ u,u\}$. Додатково до маски результату, під час обчислення функції $\breve{f}$ можуть використовуватись інші випадкові числа.

До оцінки характеристик ОБ дослідимо такі характеристики складності їх структур [113]: апаратну, часову і місткістну складність. Додатково оцінимо розмір вибірки випадкових чисел, необхідних для виконання обчислень, масштабованість структури пристрою до кількості масок, можливість адаптування структури до роботи із різними типами масок.

Апаратна складність оцінюється як кількість умовних (типових) елементів, необхідних для побудови ОБ. Як правило, для апаратної реалізації у інтегральному виконанні одиницею виміру апаратної складності є кількість умовних вентилів типу І-НЕ, необхідних для побудови ОБ заданої структури.

Часова складність оцінюється як довжина критичного шляху виконання того чи іншого процесу. При апаратній реалізації ОБ часова складність відображає максимальну затримку обробки даних при заданому порядку виконанні обчислень. При програмному виконанні алгоритмів обробки часова складність визначає загальний час, який необхідно витратити на обробку даних. Місткістна складність визначається як розмір пам’яті, необхідний для зберігання проміжних результатів обчислень.

Зазначимо, що складність виконання алгоритмів чи їх базових операцій суттєво змінюється при використанні маскування даних різного типу. Наприклад, застосування ЛМ виправдане при виконанні логічних операцій.

Разом з тим, використання АМ при здійсненні логічних операцій над даними у МП суттєво збільшує складність виконання таких операцій. Тому, якщо до набору базових операцій алгоритму криптографічного перетворення входять операції різних груп (наприклад, арифметичні, логічні, бітові перестановки), то при переході від операції з однієї групи до операції з іншої групи доцільно виконувати перетворення МП даних для спрощення подальших здійснюваних операцій. Такі перетворення отримали назву “перетворення маски” чи “перетворення МП” даних. На практиці найчастіше використовують перетворення логічної маски у арифметичну і навпаки. Практичне використання мультиплікативної маски є дещо обмеженим внаслідок можливості проведення атак спеціального виду – так званих “нуль-атак” [89, 90].

Розробка методів виконання операцій над даними у МП пов’язана з труднощами оцінки рівня безпеки процесу виконання отриманих операцій для заданого переліку атак. Один із шляхів оцінки цього рівня полягає у проведенні усіх заданих атак на комп’ютерну реалізацію алгоритму. Однак такому підходу притаманний недолік – якість та успішність інженерно-криптографічних атак часто залежить від характеристик використовуваної системи вимірювання параметрів пристрою, наприклад, СП. Із покращенням цих характеристик – швидкодії, роздільної здатності, динамічного діапазону вимірювань, чутливості тощо – зростає ймовірність успішного атакування. Тому все частіше для таких оцінок використовуються формальні методи, які дозволяють оцінити рівень безпеки процесу виконання тих чи інших операцій на основі їх математичного опису.

Нетривіальність знаходження функції $\breve{f}$ полягає у тому, що ця функція повинна обчислюватись без розголошень відомостей (витоку інформації) про a, b і z. Теоретичною умовою стійкості процесу виконання обчислень функції $\breve{f}$ до атакування на основі АСП є статистична незалежність кожного проміжного результату обчислень від вхідних даних, зокрема немаскованих.

Вибір кількості масок для МП залежить від умов, які висуваються для розробки алгоритмів виконання базових операцій. Як було доведено у [45], для побудови алгоритмів, процес виконання яких стійкий до проведення атак на основі АСП n-го порядку, необхідно використати не менш ніж n масок для представлення проміжних результатів.

Було розглянуто сучасні методи виконання операцій над даним у МП. Кожен метод було проаналізовано з позицій характеристик складності, кількості використаних масок у МП (масштабованості до кількості масок), оновлення маски результату та підтримки обробки даних у МП із різнотипними масками. Отримані дані представлені в табл. 1.3.

Серед відомих методів виконання логічних операцій над даними у МП слід відзначити [44, 91, 92]. Метод **SWITCH-MUX** [44] оснований на обчисленні результату виконання логічних операцій над даними у МП за допомогою деревоподібної структури, яка складається із двовходових мультиплексорів та суматорів за модулем два. Однак цей метод володіє значною апаратною і часовою складністю. Тому у [91] було запропоновано спрощені методи **MUX-TREE** виконання таких операцій. Спрощені методи основані на ідеї [44] та складаються також із набору двовходових мультиплексорів. Спрощення процесу обчислень досягається за рахунок повторного використання маски одного з вхідних даних. При цьому зменшується розмір вибірки випадкових чисел, які необхідні для роботи ОБ. Альтернативний напрямок побудови методів для виконання логічних операцій над даними у МП **XOR-AND** був запропонований у роботі [92] і розвинутий у [80]. Даний метод оснований на алгебраїчних властивостях операцій у полях $GF(2^N)$. Однак перелічені методи не дозволяють обробляти дані, у МП яких використано більш ніж одну маску. Крім цього, при використанні методів [44, 91] для створення комп'ютерних компонент на базі програмованих процесорів необхідно виконувати значну кількість команд процесора, що призводить до зниження продуктивності обробки даних. Наприклад, метод [91] вимагає 10 команд процесора для виконання операції

логічного множення двох даних у МП з використанням однієї маски. Тому більш перспективним є метод [80], який для отримання такого-ж результату потребує 8 команд процесора. Тому доцільним напрямком дослідження є розвиток методів обробки даних у МП в частині збільшення кількості масок, які можна використати при МП даних та зменшенні часової складності виконання операцій.

Табличні перетворення (підстановка) даних широко використовуються при побудові нелінійних операцій алгоритмів криптографічних перетворень. Так у [75] запропоновано програмну реалізацію цієї операції (так званий алгоритм "повного маскування" – **FULL-MASK**). Принцип виконання операції підстановки даних у МП полягає у маскуванні початкової таблиці T, за допомогою якої реалізовано перетворення заміни для деякого алгоритму. Маскування здійснюється за допомогою двох масок: вхідної маски R^{in} і вихідної маски R^{out} таким чином, що для модифікованої таблиці T' справджується рівність $T'[a_{i,j} \oplus R^{in}{}_{i,j}] = T[a_{i,j}] \oplus R^{out}{}_{i,j}$. Такий спосіб виконання операції передбачає, що маскована таблиця повинна обчислюватися для кожної пари R^{in} і R^{out}. Якщо зафіксувати пару масок для кожного раунду алгоритму та обчислити наперед таблицю, то для її збереження вимагається вдвічі більший обсяг пам'яті. Наприклад, для алгоритму AES необхідно використовувати 16 (чи 20) маскованих таблиць та додаткову коректувальну таблицю.

Альтернативний спосіб виконання табличних перетворень над даними у МП полягає у обчисленні таблиць "на льоту" (**ON-THE-FLY**), використовуючи, наприклад, алгоритм запропонований у [93, 94] і розвинутий у [92, 95]. Однак, такий підхід вимагає витрат часу, оскільки вимагається застосовувати біля 16 (чи 20, якщо врахувати алгоритм обчислення розпису ключа) разів за один раунд. Розвитком цієї ідеї є швидкий алгоритм для модифікування маскованих таблиць, запропонований у [92, 96] (**FAST-ON-THE-FLY**), який передбачає обчислення послідовностей розбиття таблиць на блоки деякого розміру і перестановку блоків залежно від значення маски вхідних даних (алгоритм "поділу-і-обміну"). Згідно з цим алгоритмом, для деякої вхідної маски M, маскована таблиця T'

поділяється на блоки, розмір яких залежить від позиції одиничних бітів у M. Після цього блоки таблиці попарно обмінюються місцями. Для отримання коректного результату з таблиці T' номер вузла заміни з неї обчислюється за допомогою коректувальної таблиці. Однак, внаслідок використання додаткової коректувальної таблиці алгоритму "поділу-і-обміну" притаманна збільшена місткістна складність. Крім того, врахувавши, що, в середньому, кількість одиниць в N -розрядній масці рівна $N/2$, часова складність виконання підготовчого етапу цього алгоритму складе $O(N \cdot 2^N / 2)$.

Спільним недоліком розглянутих методів є їх орієнтування лише на дані, подані у МП з використанням логічної маски. Якщо для маскування даних використано арифметичну маску, то згадані методи не можна використовувати. Тому доцільним напрямком дослідження є створення таких методів виконання табличних перетворень даних, які володіють малою місткістною складністю та дозволяють обробляти дані як із ЛМ, так і з арифметичним.

Якщо до переліку базових операцій алгоритму криптографічного перетворення входять як логічні, так і арифметичні операції, наприклад, [97, 98, 99, 100], то при адаптуванні цих алгоритмів до обробки даних у МП використовують перетворення типів масок та даних у МП: дані у МП та їх відповідна логічна маска перетворюються у дані у МП з відповідною логічною маскою і навпаки. Серед відомих автору, метод **XOR-MUX-MR** [89] передбачає використання таблиць, які необхідно оновлювати кожного разу, коли проходить нове перетворення маски. Даний метод може використовуватися для програмної реалізації перетворень масок на універсальних програмованих процесорах. Проте, при апаратній реалізації згаданий метод вимагає більше обладнання від запропонованого внаслідок його більшої місткістної складності.

Альтернативний метод обчислення довільних функцій над даними у МП **TABLE-MR** [9] володіє обмеженням на розрядність вхідних даних – при великій розрядності вхідних даних (більш ніж 8 чи 9 розрядів) сильно зростає місткістна складність методу.

Метод, описаний у [75], передбачає генерування випадкового переносу для перетворення масок. Однак, як було показано у [101], даний метод розголошує відомості про Хемінгову вагу відкритих даних. Тому на практиці його застосування не рекомендовано.

Традиційний спосіб виконання арифметичних операцій над даними у МП з логічною маскою полягає у послідовному виконанні таких перетворень [89, 93, 102 , 103]:

- перетворення МП даних із логічним маскуванням у представлення з АМ;
- виконання арифметичної операції над даними з арифметичною маскою;
- обчислення/генерування нової арифметичної маски;
- перетворення МП даних із АМ у представлення з ЛМ.

Однак, недоліком такого способу обробки даних є його висока часова складність, оскільки необхідно використовувати чотири послідовні перетворення, що призводить до зменшення продуктивності обробки даних. Крім цього, методи [89, 93, 102, 103] орієнтовані на обробку даних у МП з однією маскою і не є придатними для обробки даних з більшою кількістю масок.

Було розглянуто методи інвертування даних у МП у полях виду $GF(2^N)$. Відомі методи обробляють вхідні дані у МП з логічною маскою та дозволяють отримувати результат у МП з мультиплікативною маскою. Метод **MULT-MASK** [46] оснований алгебраїчних перетвореннях даних у МП та введенні нової мультиплікативної проміжної маски.

Метод **ADAPT-MULT-MASK** [93] є розвитком попереднього методу в частині зменшення кількості операцій, необхідних для обчислення результату. Як було показано у [104], обидва методи не забезпечують захист від АСП з використанням так званих «нуль атак», також вони не володіють масштабованістю до кількості масок у МП.

Порівняльна характеристика методів виконання операцій над даними у МП Таблиця 1.3.

Операція	**Метод виконання**	**Характеристики складності методів**		**Повне маскування результату**	**Нова маска результату**	**Різнотипне МП**	**Адаптування довільної кількості масок**
		Апаратна	**Часова**				
Кон'юнкція	SWITCH-MUX	$4A_{NOT} + 6A_{\wedge} + 2A_{\vee}$	$3t_{NOT} + 3t_{\wedge} + 2t_{\vee}$ 12 операцій	+	+	–	–
	MUX-TREE	$3A_{NOT} + 6A_{\wedge} + 2A_{XOR}$	$2t_{NOT} + 2t_{\wedge} + 2t_{\vee}$ 12 операцій	+	–	–	–
	XOR-AND	$3A_{NOT} + 6A_{\wedge} + 3A_{\vee}$	$1t_{NOT} + 2t_{\wedge} + 2t_{\vee}$ 12 операцій	+	+	–	–
Диз'юнкція	SWITCH-MUX	$4A_{NOT} + 6A_{\wedge} + 2A_{\vee}$	$3t_{NOT} + 3t_{\wedge} + 2t_{\vee}$ 12 операцій	+	+	–	–
	MUX-TREE	$3A_{NOT} + 6A_{\wedge} + 2A_{XOR}$	$2t_{NOT} + 2t_{\wedge} + 2t_{\vee}$ 12 операцій	+	–	–	–
Перетворення МП	XOR-MUX-MR	$3A_{NOT} + 6A_{\wedge} + 3A_{XOR}$	$t_{NOT} + 2t_{\wedge} + 2t_{\vee}$	+	–	–	–
	TABLE-MR	$O(N2^{N+1})A_{\wedge}$	$O(N2^{N})$ підготовка $O(1)$ виконання	+	–	–	–
Табличні перетворення	FULL-MASK ЛМ	$O(N2^{N+1})\ A_{\wedge}$	$O(N2^{N})$ підготовка $O(1)$ виконання	+	+	–	–
	ON-THE-FLY ЛМ	$O(N2^{N})\ A_{\wedge}$	$O(N2^{N+1})$	+	+	–	–
	FAST-ON-THE-FLY	$O(N2^{N})A_{\wedge}$	$O(N2^{N-1})$	+	+	–	–
Інвертування	MULT-MASK	$4A_{M}(N) + 2A_{\oplus}(N) + 2A_{I}(N)$	$3t_{M}(N) + 2t_{\oplus}(N) + 2t_{I}(N)$	–	+	–	–
	ADAPT-MULT-MASK	$3A_{M}(N) + 2A_{\oplus}(N) + A_{I}(N)$	$2t_{M}(N) + 2t_{\oplus}(N) + t_{I}(N)$	–	+	–	–

В результаті аналізу таблиці 1.3 встановлено, що розглянуті методи виконання операцій над даними у МП орієнтовані на МП лише з однією маскою, що дозволяє побудувати ОБ криптографічних пристроїв, стійких до атак на основі АСП першого порядку. Для створення ОБ, стійких до АСП вищих порядків, перспективним напрямком дослідження є розвиток таких методів обробки даних для уможливлення обчислень над даними у МП із довільною кількістю масок. Також, для спрощення погодження МП даних із різнотипними масками, необхідно дослідити методи обробки даних, які надають можливість використання різнотипних масок даних та дозволяють будувати ОБ з низькою місткістною складністю.

1.5 Висновки до першого розділу

Таким чином, проведено аналіз методів захисту від атак на основі АСП від обмеження обсягу даних, які обробляються на одному ключі шифрування до рандомізування проміжних значень, результатом якого стала їх класифікація за ознаковим принципом. Подальші дослідження можуть бути пов’язані з розробкою методів на базі технології представлення даних у МП для підвищення рівня захищеності криптографічних блоків пристроїв захисту інформації від атак на основі АСП.

Встановлено, що МП даних можна застосовувати як на рівні базової технології виготовлення криптографічних пристроїв, зокрема маскованих логічних елементів, так і на рівні високорівневої обробки даних. На основі аналізу властивостей використання МП даних виявлено, що при цьому необхідно модифікувати специфікації алгоритмів криптографічних перетворень та їх відповідні потокові графи на рівні базових операцій алгоритмів.

Також проведено огляд існуючих методів виконання операцій над даним у МП, а також проведено порівняльний аналіз методів виконання арифметичних та логічних операцій над даними у МП, що далі буде використано при розробці методів виконання таких операцій над даними у МП із довільною кількістю масок.

РОЗДІЛ 2
МЕТОДИ ВИКОНАННЯ ОПЕРАЦІЙ НАД ДАНИМИ У МАСКОВАНОМУ ПРЕДСТАВЛЕННІ

2.1 Логічні операції над даними у маскованому представленні

Для створення методу виконання двомісних логічних операцій Булевої алгебри над даними у МП, скористаємося алгебраїчним методом, запропонованим у [104]. Згаданий метод використовує дистрибутивні властивості операцій логічного множення та додавання за модулем два та дозволяє виконання операції логічного множення над даними у МП з використанням логічної маски. Для виконання операцій над даними з більшою кількістю масок (дві та більше) нами запропоновано параметризований алгебраїчний метод виконання двомісних логічних операцій (логічного множення і додавання), де параметром є кількість використаних логічних масок [105].

Припустимо, що a і b є немаскованими даними, над якими необхідно виконати двомісну логічну операцію у МП з використанням заданої кількості масок. Нехай дані a і b подані у МП з використанням ЛМ: $\tilde{a} = a \oplus x_1 \oplus \ldots \oplus x_n$, і $\tilde{b} = b \oplus y_1 \oplus \ldots \oplus y_n$, де $x = x_1,\ldots,x_n$, $y = y_1,\ldots,y_n$ – маски відповідно a і b, що є незалежними випадковими числами з рівномірним розподілом ймовірності. Задача обчислення логічної операції над даними у МП формулюється так: за заданими наборами $\{\tilde{a},x\}$, $\{\tilde{b},y\}$ обчислити без розголошення відомостей про a і b результат $\{\tilde{c},z\}$, де $\tilde{c} = (a * b) \oplus z_1 \oplus \ldots \oplus z_n$, "$*$" – позначення логічної операції, $z = z_1,\ldots,z_n$ – маски результату, що є незалежними випадковими числами з рівномірним розподілом ймовірності.

Використовуючи дистрибутивні властивості операцій логічного множення і додавання за модулем два, автором запропоновано вираз для виконання логічної операції множення над даними, поданими у МП з використанням n логічних масок, який має вид [105]:

$$\widetilde{c} = a \cdot b \oplus \bigoplus_{i=1}^{n} z_i = \widetilde{a} \cdot \widetilde{b} \oplus \bigoplus_{i=1}^{n} x_i \cdot \widetilde{b} \oplus \bigoplus_{j=1}^{n} y_j \cdot \widetilde{a} \oplus \bigoplus_{\substack{i=1 \\ j=1}}^{n} x_i \cdot y_j \oplus \bigoplus_{i=1}^{n} z_i, \qquad (2.1)$$

де "·" – позначення операції логічного множення.

На практиці найчастіше використовують одну, дві чи три маски.Вираз для обчислення логічного множення даних у МП із використанням однієї маски описано у [104] та є частковим випадком виразу (2.1) при $n = 1$.

Автором запропоновано вирази для виконання логічного множення двох даних із використанням двох та трьох логічних масок. Відповідні вирази мають вид [105]:

$$\begin{aligned} \widetilde{c} = (a \cdot b) \oplus z_1 \oplus z_2 = \widetilde{a} \cdot \widetilde{b} \oplus \widetilde{a} \cdot y_1 \oplus \widetilde{a} \cdot y_2 \oplus \widetilde{b} \cdot x_1 \oplus \widetilde{b} \cdot x_2 \oplus \\ \oplus x_1 \cdot y_1 \oplus x_1 \cdot y_2 \oplus x_2 \cdot y_1 \oplus x_2 \cdot y_2 \oplus z_1 \oplus z_2 \end{aligned}, \qquad (2.2)$$

де $\widetilde{a} = a \oplus x_1 \oplus x_2$, $\widetilde{b} = b \oplus y_1 \oplus y_2$, $z = z_1, z_2$ та

$$\begin{aligned} \widetilde{c} = (a \cdot b) \oplus z_1 \oplus z_2 \oplus z_3 = \widetilde{a} \cdot \widetilde{b} \oplus \widetilde{a} \cdot y_1 \oplus \widetilde{a} \cdot y_2 \oplus \widetilde{a} \cdot y_3 \oplus \\ \oplus \widetilde{b} \cdot x_1 \oplus \widetilde{b} \cdot x_2 \oplus \widetilde{b} \cdot x_3 \oplus \\ \oplus x_1 \cdot y_1 \oplus x_1 \cdot y_2 \oplus x_1 \cdot y_3 \oplus \\ \oplus x_2 \cdot y_1 \oplus x_2 \cdot y_2 \oplus x_2 \cdot y_3 \oplus \\ \oplus x_3 \cdot y_1 \oplus x_3 \cdot y_2 \oplus x_3 \cdot y_3 \oplus \\ \oplus z_1 \oplus z_2 \oplus z_3 \end{aligned}, \qquad (2.3)$$

де $\widetilde{a} = a \oplus x_1 \oplus x_2 \oplus x_3$, $\widetilde{b} = b \oplus y_1 \oplus y_2 \oplus y_3$, $z = z_1, z_2, z_3$.

Використовуючи дистрибутивні властивості операцій логічного множення і додавання за модулем два, автором запропоновано вираз для виконання операції логічного додавання над даними, поданими у МП з використанням n логічних масок, який має вид [105]:

$$\widetilde{c} = (a \vee b) \oplus \bigoplus_{i=1}^{n} z_i = \widetilde{a} \vee \widetilde{b} \oplus \bigoplus_{i=1}^{n} x_i \cdot \widetilde{b} \oplus \bigoplus_{j=1}^{n} y_j \cdot \widetilde{a} \oplus \bigoplus_{\substack{i=1 \\ j=1}}^{n} x_i \cdot y_j \oplus \bigoplus_{i=1}^{n} x_i \oplus \bigoplus_{j=1}^{n} y_j \oplus \bigoplus_{i=1}^{n} z_i, \qquad (2.4)$$

де "$\vee$ " – позначення операції логічного додавання.

Особливістю цього виразу, як і попереднього, є уникнення обчислень із відкритими даними a і b. При цьому можна запропонувати декілька варіантів

виконання виразу (2.4), які будуть відрізнятися між собою порядком виконання операцій. На практиці найчастіше використовують одну, дві чи три маски. Тому автором запропоновано вирази для виконання логічного множення двох даних із використанням однієї $(n=1)$, двох $(n=2)$ та трьох $(n=3)$ логічних масок. Відповідні вирази мають вид [105]:

$$\tilde{c} = (a \vee b) \oplus z = \tilde{a} \vee \tilde{b} \oplus \tilde{a} \cdot y \oplus \tilde{b} \cdot x \oplus x \cdot y \oplus x \oplus y \oplus z, \tag{2.5}$$

$$\begin{gathered}\tilde{c} = a \vee b \oplus z_1 \oplus z_2 = \tilde{a} \vee \tilde{b} \oplus \tilde{a} \cdot y_1 \oplus \tilde{a} \cdot y_2 \oplus \tilde{b} \cdot x_1 \oplus \tilde{b} \cdot x_2 \oplus x_1 \cdot y_1 \oplus x_1 \cdot y_2 \oplus \\ \oplus x_2 \cdot y_1 \oplus x_2 \cdot y_2 \oplus x_1 \oplus x_2 \oplus y_1 \oplus y_2 \oplus z_1 \oplus z_2\end{gathered}, \tag{2.6}$$

де $\tilde{a} = a \oplus x_1 \oplus x_2$, $\tilde{b} = b \oplus y_1 \oplus y_2$, $z = z_1, z_2$,

$$\begin{aligned}\tilde{c} = (a \vee b) \oplus z_1 \oplus z_2 \oplus z_3 = \tilde{a} \vee \tilde{b} \oplus \tilde{a} \cdot y_1 \oplus \tilde{a} \cdot y_2 \oplus \tilde{a} \cdot y_3 \oplus \\ \oplus \tilde{b} \cdot x_1 \oplus \tilde{b} \cdot x_2 \oplus \tilde{b} \cdot x_3 \oplus \\ \oplus x_1 \cdot y_1 \oplus x_1 \cdot y_2 \oplus x_1 \cdot y_3 \oplus \\ \oplus x_2 \cdot y_1 \oplus x_2 \cdot y_2 \oplus x_2 \cdot y_3 \oplus \\ \oplus x_3 \cdot y_1 \oplus x_3 \cdot y_2 \oplus x_3 \cdot y_3 \oplus \\ \oplus x_1 \oplus x_2 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus \\ \oplus z_1 \oplus z_2 \oplus z_3\end{aligned}, \tag{2.7}$$

де $\tilde{a} = a \oplus x_1 \oplus x_2 \oplus x_3$, $\tilde{b} = b \oplus y_1 \oplus y_2 \oplus y_3$, $z = z_1, z_2, z_3$.

Методи виконання операцій кон'юнкції чи диз'юнкції над даними у МП є подібними та полягають у наступному:

Етап 1. Для підвищення захищеності виконання обраної операції до атак на основі АСП заданого порядку na встановлюють кількість масок даних у МП рівну $n = na$ заданої розрядності.

Етап 2. Генерують n масок заданої розрядності для маскування вхідних даних – випадкові незалежні маски із рівномірним розподілом ймовірності. Аналогічно генерують необхідну кількість додаткових масок для внутрішніх операцій та масок результату.

Етап 3. Переводять дані у МП, виконуючи операцію маскування над даними із згенерованими на другому етапі масками. Встановлюють маски для внутрішніх обчислень та маски результату.

Етап 4. На основі виразу (2.1) для операції кон'юнкції чи виразу (2.4) для операції диз'юнкції проводять обчислення для обраної кількості масок. Особливістю цього етапу є уникнення обчислень із відкритими даними a і b. При цьому можна запропонувати декілька варіантів виконання згаданих виразів, які будуть відрізнятися між собою порядком виконання операцій.

Етап 5. Отриманий результат у МП та маску результату використовують для переведення даних із МП у немасковане. За необхідності, дані залишають у МП для виконання над ними подальших операцій.

Етап 6. Видаляють маски для внутрішніх обчислень, згенеровані на етапі 2 та встановлені на етапі 3.

Отже, використовуючи вирази (2.1) і (2.4), можна створити базові логічні ОБ для подальшої побудови на їх основі комп'ютерних компонентів для виконання криптографічних перетворень, які дозволяють обробляти дані, подані у МП з використанням n логічних масок. При цьому розроблені методи дозволяють будувати необхідні обчислення за заданим n, яке використовується для подання даних і результатів.

2.2 Табличні перетворення даних у МП

Для розробки методу табличних перетворень даних у МП приймемо, що задано:

- множину $Z_n = \{0,..., n-1\}$ цілих чисел, наділену бінарними операціями "$\circ$" та "$\bullet$", які утворюють групи $G_{\circ}$ та $G_{\bullet}$ відповідно;
- функцію $f(a)$, $a \in Z_n$, задану табличним способом для усіх $a \in Z_n$ за допомогою таблиці $f(a) = T[a]$. Функція $f(a)$ визначає вузли заміни у векторі заміни;

- числа x (маска аргументу), y (проміжна маска), z (маска результату), які є незалежними випадковими числами з рівномірними законами розподілу ймовірності, причому $x, y, z \in Z_n$.

Задача обчислення функції $f(a)$ з використанням маскованого аргументу, маски та отриманого маскованого результату формулюється так: обчислити $\tilde{f}(a) = f(a) \bullet z$ з використанням лише $\tilde{a} = a \circ x$, x, z, $T[a]$, таким чином, щоб надати порушнику мінімум інформації про відкритий аргумент a.

Для розв'язання сформульованої задачі автором запропоновано скористатися двома процедурами: підготовчою та основною [9]. Підготовча процедура виконується кожного разу при зміні y або z. Результатом її виконання є модифікована таблиця заміни T' з властивістю $T'[b \circ y] = T[b] \bullet z$.

Підготовча процедура 2.1 [9]:

Вхід: $T[a]$, y і z.

Вихід: таблиця T' з властивістю $T'[b \circ y] = T[b] \bullet z$.

Для усіх $i \in Z_n$, обраних випадково з Z_n, обчислити $T'[i] = T[i \circ y] \bullet z$.

Видати T'.

Основна процедура призначена для обчислення $\tilde{f}(a) = f(a) \bullet z$ з використанням результатів підготовчої процедури, вхідних даних у МП $\tilde{a} = a \circ x$ і масок x, y.

Основна процедура 2.2 [9]:

Вхід: таблиця T' з властивістю $T'[b \circ y] = T[b] \bullet z$, $\tilde{a} = a \circ x$, x, y.

Вихід: $\tilde{f}(a) = T[a] \bullet z$.

Обчислити $\tilde{b}_1 = \tilde{a} \circ y$.

Обчислити $\tilde{b}_2 = \tilde{b}_1 \circ x^{-1}$, де x^{-1} – обернений елемент до x.

Обчислити $\tilde{f}(a) = T'[\tilde{b}_2] = T[a] \bullet z$.

Повернути $\widetilde{f}(a)$.

Підкреслимо, що в основній процедурі є важливою послідовність виконання першого та другого кроків, тобто кроків модифікування маски вхідних даних. Якщо виконати послідовність цих кроків у зворотному порядку, то на першому кроці основної процедури $\widetilde{b}_1 = \widetilde{a} \circ x^{-1} = a \circ x \circ x^{-1} = a$, що створює передумови для успішної атаки на основі АСП.

При реалізації криптографічних перетворень для маскування даних використовують $n = p^l$, де p – просте число (загалом дорівнює двом), l – натуральне ціле число; операції побітового додавання двійкових подань аргументів за модулем 2 (так зване ЛМ) і додавання двійкових подань аргументів за модулем N (так зване "арифметичне маскування") чи віднімання за модулем N. Операції модульного множення використовують значно рідше внаслідок загрози проведення нуль-атак [93]. Наведені процедури дозволяють використовувати довільні комбінування перелічених операцій для маскування вхідних та вихідних даних.

Методи виконання операцій табличних перетворень над даними у МП є полягаєює у наступному:

Етап 1. Згенерувати випадкові маски даних та маски y та z заданої розрядності – випадкові незалежні маски із рівномірним розподілом ймовірності.

Етап 2. На основі таблиці $T[a]$ та масок y та z, заданих операцій маскування вхідних та вихідних даних виконати підготовчу процедуру 2.1 та отримати таблицю T'.

Етап 3. Переводять вхідні дані у МП, виконуючи операцію маскування над даними із згенерованими на першому етапі масками. Встановлюють маски для внутрішніх обчислень та маски результату.

Етап 4. Виконують основну процедуру 2.2 та отримують результат у МП $\widetilde{f}(a)$.

Етап 5. Отриманий результат у МП та маску результату z використовують для переведення даних із МП у немасковане. За необхідності, дані залишають у МП для виконання над ними подальших операцій.

Етап 6. Видаляють маски для внутрішніх обчислень, згенеровані на етапі 2 та встановлені на етапі 3.

2.3 Операція інвертування даних у маскованому представленні у скінчених полях Галуа з характеристикою 2

Для побудови методу інвертування даних у МП у скінчених полях Галуа виду $GF(2^N)$ скористаємось запропонованим у роботі [106] представленням елементів цього поля у вигляді степенів генератора поля. Таке представлення основане на факті, що ненульові елементи в скінченому полі $GF(2^N)$ можуть бути здобуті шляхом експоненціювання генератора в цьому полі. Тому, обравши базис у $GF(2^N)$, знаходимо генератор поля γ і обчислюємо усі пари (α, i), де $\alpha = \gamma^i$, $0 \le i \le 2^N - 1$, $\alpha \in GF(2^N) \setminus \{0\}$. Таке подання ненульових елементів в $GF(2^N)$ є унікальним для кожного обраного генератора γ. Також i називають дискретним логарифмом α відносно γ. Обчислені пари (α, i) зберігаються в двох таблицях: у $\log$ -таблиці, відсортованої в порядку зростання α, та у $a\log$ -таблиці, відсортованої в порядку зростання i. Кожна таблиця містить $2^N - 1$ слів, розміром N бітів. Базуючись на наведених в [106] алгоритмах обчислення добутку елементів α і β та інвертування у полі ненульового елемента α, отримуємо:

$$\alpha \cdot \beta = a\log[(\log[\alpha] + \log[\beta]) \bmod (2^N - 1)],$$

$$\alpha^{-1} = a\log[-\log[\alpha] \bmod (2^N - 1)].$$

З метою уникнення необхідності перевірки на рівність нулю елемента при виконанні інвертування, у роботі [107] запропоновано розширити таблиці за

допомогою додавання двох елементів $\log[0] = 2^N - 1$ і $a\log[2^N - 1] = 0$. Тоді алгоритми інвертування та множення переписуються у вигляді, відповідно:

$$\alpha^{-1} = \begin{cases} a\log[(2^N - 1) - \log(\alpha)], & \text{якщо} \quad 0 < (2^N - 1) - \alpha < 2^N - 1 \\ a\log[\alpha] \end{cases}, \tag{2.8}$$

$$\alpha \cdot \beta = \begin{cases} a\log[\beta], & \textit{якщо} \quad \alpha = (\log[\alpha] + \log[\beta]) \bmod (2^{\mathrm{N}} - 1) \\ a\log[\alpha], & \textit{якщо} \quad \beta = (\log[\alpha] + \log[\beta]) \bmod (2^{\mathrm{N}} - 1) . \\ a\log[(\log[\alpha] + \log[\beta]) \bmod (2^{\mathrm{N}} - 1)] \end{cases} \tag{2.9}$$

Задача виконання операції пошуку оберненого елемента над даними у МП формулюється так. Нехай задано $A \oplus R$, де A є немаскованими даними, R - маска у вигляді випадкового незалежного числа з рівномірним розподілом ймовірності. Тоді необхідно знайти ефективний алгоритм обчислення $A^{-1} \oplus \tilde{R}$ без виявлення інформації про A чи A^{-1} в процесі обробки. Тут маска $\tilde{R}$ може бути або рівною R або якійсь іншій (рівномірно розподіленій) випадковій величині. Далі подамо метод пошуку оберненого елемента до даних у МП з використанням n логічних масок.

Припустимо, що A є немаскованими даними, над якими необхідно виконати операцію інвертування у МП з використанням заданої кількості масок. Нехай дані A подані у МП з використанням ЛМ: $\tilde{A} = A \oplus R_1 \oplus \ldots \oplus R_n$, де $R = R_1, \ldots, R_n$ – маски відповідно A , що є незалежними випадковими числами з рівномірним розподілом ймовірності. Задача виконання операції інвертування над даними у МП формулюється так: за заданим набором $\{\tilde{A}, R\}$, обчислити без розголошення відомостей про A результат $\{\tilde{C}, Z\}$, де $\tilde{C} = A^{-1} \oplus z_1 \oplus \ldots \oplus z_n$, $z = z_1, \ldots, z_n$ – маски результату, що є незалежними випадковими числами з рівномірним розподілом ймовірності.

На базі таблично-алгоритмічного методу [10], де операції множення у полі та інвертування виконуються за допомогою виразів (2.8) і (2.9) з використанням $\log/a\log$ -таблиць, автором запропоновано [108] метод виконання операції

інвертування шляхом пошуку множника, на який необхідно помножити інвертовані дані у МП $(\widetilde{A})^{-1}$ щоб отримати A^{-1} . Для цього запишемо $\gamma^{y} = \gamma^{i} \oplus \gamma^{r_1} \oplus ... \oplus \gamma^{r_n}$, де $\widetilde{A} = \gamma^{y}$, $A = \gamma^{i}$, $R_1 = \gamma^{r_1}, ..., R_n = \gamma^{r_n}$ у вигляді:

$$\gamma^{-y} = (\gamma^{i} \oplus \gamma^{r_1} \oplus ... \oplus \gamma^{r_n})^{-1} = \gamma^{-i}(1 \oplus \bigoplus_{k=1}^{n} \gamma^{r_k - i})^{-1} = \gamma^{-i} MK , \qquad (2.10)$$

Підкреслимо, що у (2.10) не використовується мультиплікативне маскування, оскільки шуканий множник MK містить у собі й самі інвертовані дані у формі $\gamma^{r_k - i}$.

Обчислення множника $MK = (1 \oplus \bigoplus_{k=1}^{n} \gamma^{r_k - i})^{-1}$ необхідно проводити таким чином, щоб не розголошувати відомості про γ^{i}, тобто використовуючи лише γ^{y} та γ^{r_k} , де $k = 1,2,...,n$. Для цього знаходимо усі доданки $\gamma^{r_k - i}$ згідно з таким виразом:

$$\gamma^{r_k - i} = (\gamma^{y} \gamma^{-r_k} \oplus \bigoplus_{m=1}^{n} \gamma^{r_m - r_k})^{-1}, \qquad (2.11)$$

де $k = 1,2,...,n$.

Остаточний вираз для обчислення MK буде мати вид:

$$MK = (1 \oplus \bigoplus_{k=1}^{n} (\gamma^{y} \gamma^{-r_k} \oplus \bigoplus_{m=1}^{n} \gamma^{r_m - r_k})^{-1})^{-1} . \qquad (2.12)$$

Проведений автором аналіз [108] виразу (2.11) показав, що пряме обчислення доданків виду $\gamma^{r_k - i}$ уможливлює витік інформації про γ^{i} . Якщо порушник може виявити факт $\gamma^{r_k - i} = 0$, то це означає, що або $A = \gamma^{i} = 0$, або $\gamma^{r_k} = 0$. Тому порушник може систематично випробувати усі можливі значення A з метою знаходження такого значення, яке встановлює A у нуль. Зазначимо, у виразі (2.12) при $k = m$ доданок $\gamma^{r_m - r_k}$ перетворюється в одиницю. Найпростішим шляхом модифікування розробленого методу для уникнення нуль-атаки є здійснення інвертування таким чином, щоб замість виконання операції додавання за модулем два результату інверсії у гілці корекції маски з 1 і, таким

чином, виявлення γ^{i-r}, використаємо фіксовані скоректовані таблиці $\log'$ і $a\log'$ такі, що $\log'[\gamma^i \oplus 1] = i$ і $a\log'[i] = \gamma^i \oplus 1$. Це зумовлює необхідність незначного коректування процесу обробки даних. Однак загальніше рішення полягає у використанні підходу, подібного до запропонованого в [75]. Враховуючи це, автором було запропоновано [10] модифікований спосіб рандомізування операції табличного множення та інвертування у полі.

Приймемо, що перед виконанням інвертування обираються дві додаткові незалежні маски V і W. Маска V використовується для маскування кожного рядка початкової $\log$-таблиці, а маска W - для маскування вихідних значень таблиці $a\log$. Далі обчислимо нові таблиці на основі початкових $\log$- і $a\log$-таблиць. Для цього доцільно використати розроблений у цьому розділі метод виконання табличних перетворень над даними у МП (чи скористатися наведеним в [93] алгоритмом) в такий спосіб, що для кожного $\alpha = \gamma^i$, $\alpha \in GF(2^N)$, справджуються рівності: $\log'[\alpha \oplus V] = i$ і $a\log'[i] = \alpha \oplus W$. Враховуючи ці дві останні рівності, вираз (2.10) прийме вид:

$$\gamma^{-y} = (\text{alog}'(\text{inv}(\log'(\gamma^y \oplus V))) \oplus W)MK, \qquad (2.13)$$

а вираз (2.12) для обчислення MK можна записати у вигляді [108]:

$$\begin{aligned} MK = \text{alog}'(\text{inv}(\log'(1 \oplus \bigoplus_{k=1}^{m}(\text{alog}'(\text{inv}(\log'((\text{alog}'(\log'(\gamma^y \oplus V) + \\ + \text{inv}(\log'(\gamma^{r_k} \oplus V))) \oplus W \oplus \bigoplus_{m=1}^{n}(\text{alog}'(\log'(\gamma^{r_m} \oplus V) + \\ + \text{inv}(\log'(\gamma^{r_k} \oplus V))) \oplus W)) \oplus V)) \oplus V)) \oplus W \end{aligned}, \qquad (2.14)$$

де знаком "+" позначено операцію додавання за модулем $2^N - 1$.

З метою перетворення результату у представленні (2.10) у представлення з ЛМ, доцільно скористатися відповідним алгоритмом, розробленим у підрозділі 2.4.

У роботі [10] автором розглянуто приклади виконання операції інвертування даних у МП відповідно із однією та двома масками. При цьому в обох випадках інвертування даних у МП виконується згідно з виразом (2.13).

Для пошуку відповідного множника MK інвертованих даних у МП з однією логічною маскою γ^r необхідно обчислити вираз (2.12) із $n=1$:

$$MK = (1 \oplus (\gamma^y \gamma^{-r} \oplus 1)^{-1})^{-1}. \quad (2.15)$$

Скориставшись модифікованими таблицями для виконання множення, вираз (2.15) перепишемо у вигляді:

$$MK = \text{alog}'(\text{inv}(\log'(\text{alog}(\log'(\gamma^y \oplus V) + \text{inv}(\log'(\gamma^r \oplus V))) \oplus W \oplus V \oplus 1))) \oplus W, \quad (2.16)$$

де знаком "+" позначено операцію додавання за модулем $2^N - 1$.

Для пошуку відповідного множника MK інвертованих даних у МП з двома логічними масками γ^{r_1} і γ^{r_2} необхідно обчислити вираз (2.12) із $n=2$:

$$MK = (1 \oplus (\gamma^y \cdot \gamma^{-r_1} \oplus 1 \oplus \gamma^{r_2 - r_1})^{-1} \oplus (\gamma^y \cdot \gamma^{-r_2} \oplus 1 \oplus \gamma^{r_1 - r_2})^{-1})^{-1}, \quad (2.17)$$

Скориставшись модифікованими таблицями для виконання множення, вираз (2.17) можна переписати у вигляді, аналогічному до виразу (2.16).

Метод виконання операції інвертування даних у МП полягає у наступному:

Етап 1. Для підвищення захищеності виконання обраної операції до атак на основі АСП заданого порядку na встановлюють кількість масок даних у МП рівну $n = na$ заданої розрядності.

Етап 2. Генерують n масок заданої розрядності для маскування вхідних даних – випадкові незалежні маски із рівномірним розподілом ймовірності. Аналогічно генерують необхідну кількість додаткових масок для внутрішніх операцій та масок результату.

Етап 3. Переводять дані у МП, виконуючи операцію маскування над даними із згенерованими на другому етапі масками. Встановлюють маски для внутрішніх обчислень та маски результату.

Етап 4. На основі виразів (2.8) та (2.9) формують таблиці $log/alog$ та, на їх основі із використанням додаткових масок, обчислюють таблиці $log'/alog'$.

Етап 5. На основі виразу (2.11) та $log'/alog'$ таблиць обчислюють усі доданки γ^{r_k-i}, за допомогою яких обчислюють вираз (2.10) – інвертовані дані у МП.

Етап 6. На основі виразу (2.12) та $log'/alog'$ таблиць обчислюють результат корекції маски, необхідний для отримання немаскованих даних з результату етапу 5.

Етап 7. Отриманий результат у МП та корекцію маски результату використовують для переведення даних із МП у немасковане. За необхідності, дані залишають у МП для виконання над ними подальших операцій чи проводять перетворення МП даних у потрібну форму.

Етап 6. Видаляють маски для внутрішніх обчислень, згенеровані на етапі 2 та встановлені на етапі 3.

2.4 Перетворення маскованого представлення даних

В процесі виконання базових операцій криптографічних перетворень можна отримувати дані у МП з різними операціями маскування. Складність виконання одних і тих самих базових операцій над даними, поданими з використанням різних операцій маскування, буде суттєво відрізнятися. Оскільки ЛМ дозволяє обробляти окремо кожен біт замаскованих даних, то таке маскування доцільно використовувати при застосуванні базових операцій криптографічних перетворень, які допускають такий характер організації процесу обчислення. До таких базових операцій відносимо: логічні операції, операції бітових перестановок (включаючи циклічні зсуви, вибір бітів), операції підстановки з використанням таблиць. При цьому відбувається побітова обробка даних у МП та аналогічна обробка масок даних. АМ даних доцільно використовувати для виконання арифметичних операцій.

З іншого боку, в алгоритмах криптографічних перетворень часто використовуються операції різного типу, що призводить до необхідності

побудови ефективних методів перетворень МП даних. Тоді, перед виконанням операцій над даними, за необхідності, здійснюють перетворення МП даних. Далі здійснюють необхідні операції над даними в перетвореному МП. Такі перетворення доцільно застосовувати тоді, коли процес виконання операції над даними у МП значно спрощується при використанні альтернативного маскування.

Нехай задано множину $Z_n = \{0,...,n-1\}$ цілих чисел, наділену бінарними операціями “ $\circ$ ” та “ $\bullet$ ”, які утворюють адитивні Абелеві групи $G_{\circ}$ та $G_{\bullet}$ відповідно. Також множина Z_n наділена бінарною операцією "$\times$", яка дозволяє утворити скінчене поле $GF(2^N)$, де $N = \log n$.

Тоді операцію маскування даних $a \in Z_n$ за допомогою операції “$\circ$” можна подати у вигляді $\tilde{a} = a \circ x_1 \circ ... \circ x_k$, де $x_1,...,x_k \in Z_n$ – маски – незалежні випадкові числа з рівномірним законом розподілу ймовірності, а дані у МП можна подати у вигляді набору $\{\tilde{a},\{x\}\}$, де $\{x\} = \{x_1,...,x_k\}$. Виконання операції маскування полягає у генеруванні $\{x\} = \{x_1,...,x_k\}$ та застосуванні виразу $\tilde{a} = a \circ x_1 \circ ... \circ x_k$ до заданого a. У подальшому усі операції з перетворення a виконуються винятково з використанням представлення $\{\tilde{a},\{x\}\}$ без надання інформації про Хемінгову вагу a. Для перетворення представлення даних з маскованої форми у звичайну використовується операція зняття маски, яка застосовує обернені елементи маски: $a = \tilde{a} \circ x_1^{-1} \circ ... \circ x_k^{-1}$. Якщо для маскування даних використано операцію маскування “ $\bullet$ ”, то дані у МП подаються у вигляді $\{\hat{a},\{x\}\}$, де $\hat{a} = a \bullet x_1 \bullet ... \bullet x_k$, $\{x\} = \{x_1,...,x_k\}$, операція перетворення представлення даних з маскованого у звичайне має вигляд: $a = \hat{a} \bullet x_1^{-1} \bullet ... \bullet x_k^{-1}$. У випадку застосування операції "$\times$", дані у МП подаються у вигляді $\{\hat{a},\{x\}\}$, де $\hat{a} = a \times x_1 \times ... \times x_k$, операція перетворення представлених даних з маскованого у звичайне має вигляд: $a = \hat{a} \times x_1^{-1} \times ... \times x_k^{-1}$.

Як було зазначено у першому розділі, практика використання МП даних у криптографічних пристроях показує, що операціями "$\circ$" та "$\bullet$" є операції додавання за модулем два ("$\oplus$") та додавання за модулем 2^l ("+"). Відповідно, маскування за допомогою операції "$\oplus$" прийнято називати ЛМ, а маскування за допомогою операції "+" – АМ. Для простоти подальшого викладення приймемо, що для маскування використовується лише одна маска.

В основу запропонованих автором методів перетворення МП даних покладені функції виду [8]:

$$A_{\bullet}(\{\widetilde{a},x\},\{\widetilde{b},y\},u) = A_{\bullet}(\{a\circ x,x\},\{b\circ y,y\},u) = \{\{a\bullet b\}\circ u,u\} = \{\widetilde{s},u\}, \qquad (2.18)$$

$$A_{\circ}(\{\hat{a},x\},\{\hat{b},y\},u) = A_{\circ}(\{a\bullet x,x\},\{b\bullet y,y\},u) = \{\{a\circ b\}\bullet u,u\} = \{\hat{s},u\}. \qquad (2.19)$$

Особливістю цих функцій є обчислення результату без надання відомостей про Хемінгову вагу немаскованих аргументів. Для цього використовуються незалежні випадкові числа $u \in Z_n$ з рівномірним законом розподілу ймовірності. На підставі функцій (2.18) і (2.19), можна проводити перетворення МП даних для різних операцій маскування.

Розглянемо використання функції (2.18) для перетворення МП даних у МП, де операція маски може бути як "$\circ$", так і "$\bullet$".

Нехай задано $\{\widetilde{d},t\}$ і функцію (2.18). Тоді для перетворення МП $\{\widetilde{d},t\}$ у представлення $\{\hat{d},p\}$ необхідно виконати таку послідовність дій [8]:

Перетворення 2.3:

Вхід: $\{\widetilde{d},t\}$.

Вихід: $\{\hat{d},p\}$.

1) Згенерувати p,y,u.

2) Обчислити $A_{\bullet}(\{\widetilde{d},t\},\{\widetilde{p},y\},u) = \{\widetilde{\hat{d}},u\}$.

3) Зняти маскування з $\{\widetilde{\hat{d}},u\}$: $\{\hat{d},p\} = \{\widetilde{\hat{d}}\circ u,p\} = \{\hat{d},p\}$.

Повернути $\{\hat{d},p\}$.

Дійсно, виконання другої дії можна переписати у вигляді:

$$A_{\bullet}(\{\tilde{d},t\},\{\tilde{p},y\},u) = A_{\bullet}(\{d \circ t,t\},\{p \circ y,y\},u) = \\ = \{\{d \bullet p\} \circ u,u\} = \{\tilde{\tilde{d}},u\}$$

Тоді остання дія не призводить до витоку відомостей про Хемінгову вагу d, оскільки d подано у МП $\{\hat{d},p\}$, де p – випадкове число з рівномірним законом розподілу ймовірності.

Метод виконання перетворення МП даних із $\{\tilde{d},t\}$ у $\{\hat{d},p\}$ на основі функції (2.18) можна зформулювати так:

Етап 1. Згенерувати маски заданої розрядності: p – маску результату для маскування вихідних даних, y, u – проміжні маски – випадкові незалежні маски із рівномірним розподілом ймовірності. За необхідності генерують маску вхідних даних t.

Етап 2. Переводять дані у МП, виконуючи операцію маскування над даними із згенерованими на другому етапі масками. Встановлюють маски для внутрішніх обчислень та маски результату.

Етап 3.Виконують перетворення 2.3 та отримують $\{\hat{d},p\}$.

Етап 4. Отриманий результат у МП та маску результату використовують для переведення даних із МП у немасковане. За необхідності, дані залишають у МП для виконання над ними подальших операцій.

Етап 5. Видаляють маски для внутрішніх обчислень, згенеровані на етапі 1 та встановлені на етапі 3.

Аналогічно, для заданого $\{\hat{d},p\}$ і функції (2.18) автором побудовано перетворення з $\{\hat{d},p\}$ у $\{\tilde{d},t\}$ [8]:

Перетворення 2.4:

Вхід: $\{\hat{d},p\}$.

Вихід: $\{\tilde{d},t\}$.

1) Згенерувати випадкові числа x,y,t.
2) Обчислити $A_{\bullet}(\{\hat{d} \circ x,x\},\{p^{-1} \circ y,y\},t) = \{\tilde{d},t\}$.

Повернути $\{\tilde{d},t\}$.

Виконання останньої дії можна переписати у вигляді:

$$A_{\bullet}(\{\hat{d}\circ x,x\},\{p^{-1}\circ y,y\},t)=A_{\bullet}(\{(d\bullet p)\circ x,x\},\{p^{-1}\circ y,y\},t)=$$
$$=\{\{d\bullet p\bullet p^{-1}\}\circ t,t\}=\{\tilde{d},t\}.$$

Аналогічно до попереднього випадку, тут не відбувається витоку відомостей про Хемінгову вагу аргументу d, оскільки d подано у МП $\{\tilde{d},t\}$, де t – випадкове число з рівномірним законом розподілу ймовірності.

Метод виконання перетворення МП даних із $\{\hat{d},p\}$ у $\{\tilde{d},t\}$ на основі функції (2.18) можна зформулювати так:

Етап 1. Згенерувати маски заданої розрядності: t – маску результату для маскування вихідних даних, y, x – проміжні маски – випадкові незалежні маски із рівномірним розподілом ймовірності. За необхідності генерують маску вхідних даних t.

Етап 2. Переводять дані у МП, в иконуючи операцію маскування над даними із згенерованими на другому етапі масками. Встановлюють маски для внутрішніх обчислень та маски результату.

Етап 3.Виконують перетворення 2.4 та отримують $\{\tilde{d},t\}$.

Етап 4. Отриманий результат у МП та маску результату використовують для переведення даних із МП у немасковане. За необхідності, дані залишають у МП для виконання над ними подальших операцій.

Етап 5. Видаляють маски для внутрішніх обчислень, згенеровані на етапі 1 та встановлені на етапі 3.

Можна побудувати альтернативні перетворення з використанням функції (2.19). Нехай задано $\{\tilde{d},t\}$ і функцію (2.19). Тоді для перетворення МП з $\{\tilde{d},t\}$ у представлення $\{\hat{d},p\}$ автором запропоновано виконувати таку послідовність дій [8]:

Перетворення 2.5:

Вхід: $\{\tilde{d},t\}$

Вихід: $\{\hat{d}, p\}$.

1) Згенерувати випадкові числа p, x, y.

2) Обчислити $A_{\circ}(\{\widetilde{d} \bullet x, x\}, \{t^{-1} \bullet y, y\}, p) = \{\hat{d}, p\}$.

Повернути $\{\hat{d}, p\}$.

Справедливість останньої дії ґрунтується на властивості функції $A_{\circ}()$:

$$A_{\circ}(\{\widetilde{d} \bullet x, x\}, \{t^{-1} \bullet y, y\}, p) = A_{\circ}(\{(d \circ t) \bullet x, x\}, \{t^{-1} \bullet y, y\}, p) =$$
$$= \{\{d \circ t \circ t^{-1}\} \bullet p, p\} = \{d \bullet p, p\} = \{\hat{d}, p\}$$

Перетворення МП $\{\hat{d}, p\}$ у представлення $\{\widetilde{d}, t\}$ за заданими $\{\hat{d}, p\}$ і (2.19) автором запропоновано виконувати шляхом таких обчислень [8]:

Перетворення 2.6:

Вхід: $\{\hat{d}, p\}$.

Вихід: $\{\widetilde{d}, t\}$.

1) Згенерувати випадкові числа t, y, u;

2) Обчислити $A_{\circ}(\{\hat{d}, p\}, \{\hat{t}, y\}, u) = \{\hat{\widetilde{d}}, u\}$.

3) Зняти маскування з $\{\hat{\widetilde{d}}, u\}$: $\{\widetilde{d}, t\} = \{\hat{\widetilde{d}} \bullet u, t\} = \{\widetilde{d}, t\}$.

Повернути $\{\widetilde{d}, t\}$.

Справедливість останньої дії основана на властивості функції $A_{\circ}()$:

$$A_{\circ}(\{\hat{d}, p\}, \{\hat{t}, y\}, u) = A_{\bullet}(\{d \bullet p, p\}, \{t \bullet y, y\}, u) = \{\{d \circ t\} \bullet u, u\} = \{\hat{\widetilde{d}}, u\}.$$

Вибір функції (2.18) чи (2.19) для проведення перетворень залежить від набору операцій алгоритму криптографічного перетворення, особливостей реалізації засобів захисту інформації та зумовлюється вартістю їх реалізації та швидкодією. В загальному випадку, функцію (2.19) можна отримати з функції (2.18): $A_{\bullet}(\{\hat{a} \circ w_1, w_1\}, \{x^{-1} \circ v_1, v_1\}, t_1) = \{\widetilde{a}, t_1\}$. Аналогічно $A_{\bullet}(\{\hat{b} \circ w_2, w_2\}, \{y^{-1} \circ v_2, v_2\}, t_2) = \{\widetilde{b}, t_2\}$. Виконавши операцію "$\circ$" попарно над $\widetilde{a}$ і $\widetilde{b}$, t_1 і t_2, отримаємо $\{\widetilde{a} \circ \widetilde{b}, t_1 \circ t_2\} = \{(a \circ b) \circ (t_1 \circ t_2), t_1 \circ t_2\}$, що є МП результату виконання функції $A_{\circ}(\{\hat{a}, x\}, \{\hat{b}, y\}, u)$, де $u = t_1 \circ t_2$, однак з операцією маскування "$\circ$". Застосувавши перетворення 2.3 до отриманого результату отримаємо шукане

МП. Аналогічно можна отримати представлення функції (2.18) через функцію (2.19).

Методи виконання перетворення МП даних із використанням функції (2.19) аналогічні до описаних вище.

Альтернативним методом перетворення МП даних є використання розроблених автором методів табличних перетворень [9], що задається перетвореннями 2.1 і 2.2. Для цього задамо у перетворенні 2.1 функцію $f(a)$ у вигляді таблиці $f(a) = T[a] = a$ для усіх $a \in Z_n$. Якщо необхідно перетворити дані у МП з арифметичною маскою із даних, поданих у МП з логічною маскою, то у процедурах 2.1 і 2.2 в ролі операції "$\circ$" використовують операцію додавання за модулем два, а в ролі операції "$\bullet$" – операцію додавання за модулем 2^N. Якщо необхідно виконати перетворення даних у МП у зворотному напрямку, то у процедурах 2.1 і 2.2 в ролі операції "$\circ$" використовують операцію додавання за модулем 2^N, а в ролі операції "$\bullet$" – операцію додавання за модулем два. Формулу (2.18) реалізують за допомогою виразів, які описують суматор даних в МП за модулем 2^N (див. розділ 3).

Нехай задано дані $\{\hat{d}, x\}$, де у МП з використанням мультиплікативного маскування. Тоді для перетворення цих даних у представлення з адитивною маскою $\{\tilde{d}, y\}$ та операцією маскування “$\circ$” необхідно виконати згідно з запропонованим автором методом [10]:

$$\{\tilde{d}, y\} = \{(\hat{d} \circ y \times x) \times x^{-1}, y\}, \qquad (2.20)$$

де y - нова логічна маска, x^{-1} - зворотний до x елемент.

При сумісному використанні запропонованих у другому розділі методів пошуку оберненого елемента і перетворення маски, кількість операцій додавання за модулем два і замін по таблиці можна зменшити. У цьому випадку усі обчислення проводяться у логарифмічному представленні, уникаючи надлишкових перетворень у нормальне представлення на границі інвертування та перетворення маски.

2.5 Оцінка рівня безпеки обчислень згідно з розробленими методами

Для оцінки рівня безпеки процесу виконання алгоритмів операцій над даними у МП скористаємось методикою, вперше запропонованою у [45] і розвинутою у [109]. Розглянемо деяке криптографічне перетворення ENC, яке необхідно виконати без витоку відомостей через побічні канали. Аргументами перетворення ENC є деякий відкритий текст a і конфіденційні дані k. Проаналізуємо, згідно з ENC, процес обчислення послідовності проміжних результатів $I_1(a,k,r),...,I_t(a,k,r) = ENC(a,k)$. Кожен проміжний результат $I_i()$ залежить від відкритого тексту a, конфіденційних даних k і деякого $r \in \{0,1\}^s$. Елемент r використовується для внесення випадкового чинника в обчислення і володіє рівномірним розподілом ймовірності з $\{0,1\}^s$. Результат виконання $ENC(a,k)$ залежить лише від a і k, однак на нього не впливає r.

Розглянемо модель зловмисника, згідно з якою він володіє відомими парами відкритий-зашифрований текст $(a, ENC(a,k))$. Додатково, припустимо, що для кожної пари $(a, ENC(a,k))$ зловмисник отримав деякий набір проміжних результатів $I_1(a,k,r),...,I_\chi(a,k,r)$. Для різних пар $(a, ENC(a,k))$ зловмиснику може бути відомий різний набір проміжних результатів. Якщо зловмисник може отримати щонайбільше χ проміжних результатів для кожної пари $(a, ENC(a,k))$, то такий зловмисник називається зловмисником χ -го порядку. Метою зловмисника є обчислення конфіденційних даних k.

Зловмисник досягає успіху, якщо сумісний розподіл ймовірності отриманих ним проміжних результатів залежить від a і k. Зафіксуємо деякий набір $I_1,...,I_\chi$ проміжних результатів. Для кожної пари (a,k) позначимо $D_{a,k}(R)$ сумісний розподіл ймовірності проміжних результатів $I_1,...,I_\chi$, отриманих шляхом випадкового вибору r із $\{0,1\}^s$ з рівномірним розподілом ймовірності.

На основі отриманих автором і поданих у додатку А лем 1 – 4 [7] проведемо формальну оцінку рівня безпеки виконання операцій над даними у МП.

Розглянемо алгоритми виконання базових операцій алгоритмів криптографічних перетворень, зокрема – логічних операцій, бітових перестановок, операції заміни за таблицею, додавання, множення і пошуку оберненого елемента, перетворення МП даних.

Логічні операції над даними у МП містять операції побітового логічного додавання, логічного множення, додавання за модулем два, логічного заперечення. Розглянемо алгоритми виконання логічних операцій над даними у МП з використанням логічної маски.

Операція логічного множення двох однорозрядних операндів, поданих у МП $\{\tilde{a}, x\}$ і $\{\tilde{b}, y\}$ у полі $GF(2)$, виконується згідно з виразом [92]:

$$MAND(\{\tilde{a}, x\}, \{\tilde{b}, y\}, z\} = \{(a \otimes b) \oplus z, z\} = \{\tilde{a} \otimes \tilde{b} \oplus (\tilde{a} \otimes y \oplus (\tilde{b} \otimes x \oplus (x \otimes y \oplus z))), z\}, \quad (2.21)$$

де $x \in Z_2$ – елемент маски, випадково вибраний з рівномірним розподілом ймовірності із Z_2.

Автором отримано [7] оцінку рівня безпеки виконання (2.21) на базі аналізу властивостей проміжних змінних. Оскільки маски x і y вибираються незалежно і володіють рівномірним розподілом ймовірності, то результати виконання кожного з проміжних добутків $\tilde{a} \otimes \tilde{b}$, $\tilde{a} \otimes y$, $\tilde{b} \otimes x$ і $x \otimes y$ характеризуються розподілом ймовірності, визначеним згідно з першою частиною леми 3 (додаток А), тобто є незалежними від немаскованих даних. Обчислення проміжної суми шляхом додавання проміжного добутку $x \otimes y$ до нової маски z, згідно з лемою 1 додатку А, приводить до рівномірного розподілу ймовірності результатів проміжної суми. Продовжуючи застосування наведеної в додатку А леми 1 до наступних проміжних сум, отримуємо, що результату виконання (2.21) притаманний рівномірний розподіл ймовірності. При цьому цей результат є незалежним від даних a і b, а тому, згідно з означенням 5 додатку А, процес обчислень згідно з (2.21) характеризується властивістю повного маскування обчислень від зловмисника першого порядку.

Операція логічного додавання двох операндів, поданих у МП $\{\tilde{a},x\}$ і $\{\tilde{b},y\}$ у полі $GF(2)$, виконується згідно з наступним виразом [11]:

$$\begin{gathered} MOR(\{\tilde{a},x\},\{\tilde{b},y\},z)=\{(a\vee b)\oplus z,z\}= \\ =\{\tilde{a}\vee\tilde{b}\oplus(\tilde{a}\otimes y\oplus(\tilde{b}\otimes x\oplus(x\otimes y\oplus(x\oplus(y\oplus z))))),z\} \end{gathered}, \qquad (2.22)$$

де $x\in Z_2$ – випадковий елемент маски з рівномірним розподілом ймовірності, що належить множині Z_2.

Автором отримано [7] оцінку рівня безпеки виконання (2.22) на базі аналізу властивостей проміжних змінних. Оскільки маски x і y вибираються незалежно і володіють рівномірним розподілом ймовірності, то результати виконання кожного з проміжних добутків $\tilde{a}\otimes y$, $\tilde{b}\otimes x$ і $x\otimes y$ характеризуються розподілом ймовірності, визначеним згідно з першою частиною лемою 3 додатку А, тобто є незалежними від немаскованих даних. Проміжному результату $\tilde{a}\vee\tilde{b}$ притаманний розподіл ймовірності, визначений згідно з другою частиною леми 3 (додаток А). Обчислення проміжної суми шляхом додавання проміжного добутку $x\otimes y$ до нової маски z, відповідно до наведеної у додатку А леми 1, приводить до рівномірного розподілу ймовірності результатів проміжної суми. Продовжуючи застосування згаданої леми 1 до наступних проміжних сум, отримуємо, що результат виконання (2.22) володіє рівномірним розподілом ймовірності, є незалежним від даних a і b, а тому, згідно з означенням 5 додатку А, процес обчислень згідно з цим виразом характеризується властивістю повного маскування обчислень від зловмисника першого порядку.

Операція додавання за модулем два двох операндів, поданих у МП $\{\tilde{a},x\}$ і $\{\tilde{b},y\}$ у полі $GF(2)$, виконується згідно з виразом:

$$MXOR(\{\tilde{a},x\},\{\tilde{b},y\},z)=\{(a\oplus b)\oplus z,z\}=\{(\tilde{a}\oplus\tilde{b}\oplus z)\oplus x\oplus y,z\}. \qquad (2.23)$$

Альтернативне виконання (2.23) без використання додаткової маски z можна здійснити згідно з таким виразом: $MXOR(\{\tilde{a},x\},\{\tilde{b},y\})=\{(a\oplus b)\oplus(x\oplus y),x\oplus y\}=\{\tilde{a}\oplus\tilde{b},x\oplus y\}$.

Автором отримано оцінку рівня безпеки виконання (2.23) на базі аналізу властивостей проміжних змінних [7]. Проміжний результат $\widetilde{a} \oplus \widetilde{b} \oplus z$, обчислений у будь-якому порядку, згідно з наслідком леми 1 додатку А, володіє рівномірним розподілом ймовірності. Подальше виконання додавання x і y у довільній послідовності також приводить до рівномірного розподілу ймовірності результату. Застосувавши аналогічний підхід до аналізу спрощеного варіанту виразу (2.23), отримуємо, що, за означенням 5 додатку А, процес обчислення згідно з цим виразом та його спрощеним варіантом володіють властивістю повного маскування обчислень від зловмисника першого порядку.

Операція логічного заперечення операнду, поданого у МП $\{\widetilde{a}, x\}$ у полі $GF(2)$, виконується згідно з виразом:

$$MNOT(\{\widetilde{a}, x\}) = \{\overline{\widetilde{a}}, x\} = \{\widetilde{\overline{a}}, x\}. \qquad (2.24)$$

Автором отримано оцінку рівня безпеки виконання (2.24) на основі спостереження, що, згідно з лемою 4 додатку А, результат інвертування $\widetilde{a}$, який володіє рівномірним розподілом ймовірності, призводить до рівномірного розподілу ймовірності результату $\widetilde{\overline{a}}$ [7]. Тому, згідно з означенням 5 додатку А, процес виконання виразу (2.24) володіє властивістю повного маскування обчислень від зловмисника першого порядку.

Аналогічно можна побудувати алгоритми та оцінити рівень безпеки процесу їх виконання для обчислень інших операцій, наприклад обчислення штриху Шиффера та стрілки Пірса, тощо. Зауважимо, що виконання логічних операцій над даними у МП з використанням АМ пов’язане зі збільшенням складності виконання логічних операцій внаслідок залежності бітів результату маскування від попередніх бітів. Тому для спрощення виконання таких операцій використовують алгоритми перетворення МП даних, метою яких є заміна АМ на логічне.

Складність алгоритмів виконання арифметичних операцій над даними у МП відрізнятиметься для різних типів маскування – арифметичного чи

логічного. Розглянемо виконання арифметичних операцій додавання за модулем $n = 2^l$, $l > 1$ та пошук оберненого елемента у G_+.

Операція арифметичного додавання двох операндів $\{\hat{a}, x\}$ і $\{\hat{b}, y\}$, поданих у МП з АМ, виконується згідно з виразом:

$$MADD_+(\{\hat{a}, x\}, \{\hat{b}, y\}, z) = \{(a+b)+z, z\} = \{(\hat{a}+\hat{b}+z)-x-y, z\}, \qquad (2.25)$$

де для представлення суми використовується нова маска z. Альтернативним виразом для обчислення суми є:

$$MADD_+(\{\hat{a}, x\}, \{\hat{b}, y\}) = \{(a+b)+(x+y), x+y\} = \{\hat{a}+\hat{b}, x+y\}, \qquad (2.26)$$

де для представлення суми повторно застосовуються маски операндів.

Оцінка рівня безпеки виконання операції арифметичного додавання згідно з виразами (2.25), (2.26) проводиться аналогічно до оцінки рівня безпеки для виконання операції додавання за модулем 2 відповідно до виразу (2.23) та його спрощеного варіанту. Тому, згідно з означенням 5 додатку А, процес виконання виразів (2.25), (2.26) володіє властивістю повного маскування обчислень від зловмисника першого порядку.

У випадку застосування ЛМ, операція арифметичного додавання i-х бітів двох операндів $\{\widetilde{a}_i, x_i\}$ і $\{\widetilde{b}_i, y_i\}$ виконується згідно з виразами:

$$\{\widetilde{s}_i, u_i\} = MXOR(MXOR(\{\widetilde{a}_i, x_i\}, \{\widetilde{b}_i, x_i\}, z), \{\widetilde{p}_{i-1}, r_{i-1}\}, u_i), \qquad (2.27)$$

$$\{\widetilde{p}_i, r_i\} = MOR(MAND(\{\widetilde{a}_i, x_i\}, \{\widetilde{b}_i, y_i\}, z), MAND(\{\widetilde{p}_{i-1}, r_{i-1}\}, \{\widetilde{s}_i, u_i\}, v), r_i), \qquad (2.28)$$

де $i = 0,...,l-1$, $\{\widetilde{s}_i, u_i\}$ – МП суми, $\{\widetilde{p}_i, r_i\}$ – МП вихідного переносу, $p_{-1} = 0$, а z, u_i, r_i, v, r_{-1} – випадкові маски із рівномірним розподілом ймовірності, $MAND()$ - позначає операцію логічного множення даних у МП, $MOR()$ - позначає операцію логічного додавання даних у МП, $MXOR()$ - позначає операцію додавання за модулем два даних у МП.

Скориставшись виразами (2.27), (2.28), можна спростити запис операції додавання двох l-бітових даних за модулем $n = 2^l$:

$$MADD(\{\widetilde{a}, x\}, \{\widetilde{b}, y\}, \{z_s, \{q\}\}) = \{(a+b) \oplus z_s, z_s\} = \{\widetilde{s}, z_s\}, \qquad (2.29)$$

де $\{q\}$ – набір випадкових масок, які використовуються для проміжних обчислень згідно з виразами (2.27), (2.28).

Автором отримано [7] оцінку рівня безпеки виконання операції додавання у G_+ згідно з виразами (2.27), (2.28) шляхом аналізу статистичних властивостей проміжних змінних. Оскільки z, u_i, r_i, v і r_{-1} є незалежними масками з рівномірним розподілом ймовірності, то результат виконання усіх операцій *MXOR*, *MAND* і *MOR* буде володіти також рівномірним розподілом ймовірності. Тому, згідно з означенням 5 додатку А, процеси виконання обчислень відповідно до (2.27) – (2.29) характеризуються властивістю повного маскування обчислень від зловмисника першого порядку.

Обчислення оберненого елемента $-b$ до елемента $b \in G_+$ без розголошення його немаскованого представлення зручно виконувати за допомогою виразів (2.24) і (2.29) при використанні ЛМ. Для цього зауважимо, що обернений елемент легко знайти шляхом обчислення доповняльного коду до двійкового представлення елемента b. Відомий алгоритм обчислення доповняльного коду двійкового представлення числа передбачає побітове інвертування двійкового числа з наступним додаванням одиниці за модулем n:

$$\{-b \oplus z, z\} = MADD(\{x \oplus 1, x\}, MNOT(\{\widetilde{b}, y\}), \{z, \{q\}\}). \qquad (2.30)$$

Автором отримано оцінку рівня безпеки виконання (2.30) [7] на базі спостереження, що, згідно з лемою 4 додатку А, результат обчислення $MNOT(\{\widetilde{b}, y\})$ володіє рівномірним розподілом ймовірності. Відповідно до леми 1 додатку А результат виконання $x \oplus 1$ також характеризується рівномірним розподілом ймовірності. Як було показано вище, результату додавання двох проміжних результатів за допомогою функції $MADD()$ також притаманний

рівномірний розподіл ймовірності. Тому, за означенням 5 додатку А, процес виконання обчислень згідно з виразом (2.30) володіє властивістю повного маскування обчислень від зловмисника першого порядку.

Віднімання за модулем n можна виконати завдяки виразу (2.29): $\{(a-b)\oplus z, z\} = MADD(\{\widetilde{a}, x\}, \{-\widetilde{b}, y\}, \{z_s, \{q\}\})$, де $\{-\widetilde{b}, y\}$ – МП оберненого до b елемента, знайдене, наприклад, за допомогою обчислень згідно з виразом (2.30). Аналогічно виконується пошук різниці елементів, поданих у МП з використанням арифметичної маски: $\{(a-b)\oplus z, z\} = MADD(\{\hat{a}, x\}, \{-\hat{b}, y\})$. Оскільки, процеси обчислення згідно з функцією $MADD()$ володіють властивістю повного маскування обчислень від зловмисника першого порядку, то процеси обчислення відповідно до виразів для знаходження різниці також характеризуються аналогічною властивістю.

Вираз для виконання множення даних у полі $GF(2^l)$ у МП з використанням логічної маски аналогічний загалом до виразу (2.21), де "$\otimes$" означає операцію множення у полі $GF(2^l)$, а "$\oplus$" – додавання у полі $GF(2^l)$. Оцінка безпеки виконання множення даних у полі $GF(2^l)$ у МП з використанням логічної маски є аналогічною до оцінки безпеки виконання (2.21) з використанням леми 2 додатку А для оцінки розподілу ймовірності проміжних добутків. Тому, процес обчислення згідно з виразом (2.21) над даними із $GF(2^l)$ володіє властивістю повного маскування обчислень від зловмисника першого порядку.

Для виконання бітової перестановки π над даними у МП з використанням ЛМ необхідно виконати операцію перестановки вхідних даних у МП і маски згідно з заданою перестановкою:

$$\{\pi(a)\oplus x', x'\} = \{\pi(\widetilde{a}), \pi(x)\}. \qquad (2.31)$$

Справедливість виразу (2.31) зумовлюється тим, що при виконанні операції ЛМ кожен біт аргументу a маскується незалежно від інших бітів. Тому побітове переставляння a еквівалентне до побітового переставляння $\widetilde{a}$ з

відповідним представленням маски. Отже, маска результату дорівнюватиме результату виконання перестановки π над маскою x аргументу.

Автором отримано оцінку рівня безпеки виконання (2.31) [7] на базі спостереження, що виконання операції бітової перестановки приводить лише до зміни позицій бітів маскованого операнду без зміни значень цих бітів. Тоді для рівномірного розподілу ймовірності $\tilde{a}$ результат виконання $\pi(\tilde{a})$ також володіє рівномірним розподілом ймовірності. Тому, згідно з означенням 5 додатку А, процес обчислення за формулою (2.31) володіє властивістю повного маскування обчислень від зловмисника першого порядку.

Виконання операції бітової перестановки над даними, поданими у МП з використанням арифметичної маски, є складнішим в порівнянні з використанням логічної маски. Це зумовлено тим, що значення кожного біту у такому представленні залежить від значень попередніх бітів аргументу і маски. Тому перед виконанням бітової перестановки доцільно перетворити МП даних – замінити АМ на логічне. Далі операцію бітової перестановки можна виконати згідно з виразом (2.31). Обернене перетворення МП результату виконання (2.31) дозволяє отримати МП даних з використанням АМ.

Перетворення МП даних використовується для спрощення виконання подальших операцій над даними у МП. Як було зазначено вище, при використанні АМ складно проводити бітові маніпуляції над даними. Разом з тим, використання ЛМ ускладнює арифметичні операції у групі G_+. Розглянемо алгоритми перетворення МП даних, ґрунтуючись на виразі (2.29).

Нехай задано $\{\hat{d}, p\}$ з використанням арифметичної маски. Тоді задача перетворення $\hat{d}$ у $\{\tilde{d}, t\}$ зводиться до знаходження маски t та даних у МП $\tilde{d}$ без розкриття відомостей про Хемінгову вагу d. Скориставшись виразом (2.29) та ввівши заміну змінних $\{\tilde{a}, x\} = \{\hat{d} \oplus x, x\}$ і $\{\tilde{b}, y\} = \{(-p) \oplus y\}, y\}$, отримаємо:

$$MADD(\{\hat{d} \oplus x, x\}, \{(-p) \oplus y, y\}, z_s, \{q\}) = \{(\hat{d} - p) \oplus z_s, z_s\} = \{\tilde{d}, z_s\}. \qquad (2.32)$$

Автором отримано оцінку рівня безпеки виконання обчислень згідно з виразом (2.32) [7] на базі спостереження, що процеси обчислень за формулою (2.29) та обчислень оберненого елемента володіють властивістю повного маскування обчислень від зловмисника першого порядку, тому аналогічна властивість притаманна обчисленням відповідно до виразу (2.32).

Для перетворення МП даних з логічною маскою у представлення з арифметичною маскою приймемо, що задано МП $\{\tilde{d},t\}$ з використанням ЛМ. Тоді задача перетворення $\{\tilde{d},t\}$ у $\{\hat{d},p\}$ полягає у знаходженні маски p та $\hat{d}$ без розкриття відомостей про d. Скориставшись виразом (2.29) та ввівши заміну змінних $\{\tilde{a},x\}=\{\tilde{d},t\}$ і $\{\tilde{b},y\}=\{p\oplus y\},y\}$, де $p-$ випадкове число з рівномірним розподілом ймовірності, отримаємо:

$$MADD(\{\tilde{d},t\},\{p\oplus y,y\},z_s,\{q\})=\{(d+p)\oplus z_s,z_s\}=\{\tilde{\hat{d}},z_s\}. \qquad (2.33)$$

Оскільки у (2.33) p і z_s є незалежними випадковими числами, то $\tilde{\hat{d}}$ означає $\hat{d}$ у МП з логічною маскою z_s. Тоді $\hat{d}=\tilde{\hat{d}}\oplus z_s$. Виходячи з міркувань, аналогічних до оцінки рівня безпеки процесу виконання обчислень, згідно з виразом (2.31), автором показано [7], що процес виконання обчислень відповідно до (2.33) володіє властивістю повного маскування обчислень від зловмисника першого порядку.

Операції заміни даних чи їх частин широко використовуються у криптографічних перетвореннях і часто є основним засобом реалізації нелінійних операцій. Розглянемо процес виконання операцій заміни за таблицею над даними, поданими у МП з використанням арифметичного та ЛМ [9].

Автором отримано оцінку рівня безпеки виконання процедури заміни за таблицею шляхом окремої оцінки рівнів безпеки процесів виконання підготовчої і основної процедур [7]. При виконанні підготовчої процедури здійснюється вибір $i\in Z_n$ із рівномірним розподілом ймовірності. Результат виконання операції $i\circ y$, згідно з лемою 1 додатку А, володіє рівномірним розподілом

ймовірності. Згідно з наведеною в [94] лемою про виконання операції табличної підстановки над числом з рівномірним розподілом ймовірності та лемою 1 додатку А, результат $T'[i] = T[i \circ y] \bullet z$ також володіє рівномірним розподілом ймовірності. Таким чином, беручи до уваги означення 5 додатку А, процесу обчислення T' притаманна властивість повного маскування. Для оцінки рівня безпеки процесу виконання основної процедури проаналізуємо рівень безпеки процесу виконання окремих її кроків. Згідно з лемою 1 додатку А, результати виконання першого $\widetilde{b}_1 = \widetilde{a} \circ y$ та другого кроків $\widetilde{b}_2 = \widetilde{b}_1 \circ x^{-1}$ володіють рівномірними розподілами ймовірності. На підставі використання аналогічного підходу до оцінки рівня безпеки процесу виконання підготовчої процедури можна констатувати, що результат виконання третього кроку основної процедури $\widetilde{f}(a) = T'[\widetilde{b}_2] = T[a] \bullet z$ володіє рівномірним законом розподілу ймовірності. Тоді, за означенням 5 додатку А, процесу обчислень згідно з основною процедурою властиве повне маскування обчислень від зловмисника першого порядку.

2.6 Висновки до другого розділу

Таким чином, у другому розділі дисертації розроблено ряд методів виконання операцій над даними у МП. Зокрема, запропоновано метод виконання операції диз'юнкції, розвинуто методи виконання операції диз'юнкції та інвертування даних у МП у полях виду GF(2^N) над даними у МП із довільною кількістю масок, що, за рахунок обчислення функції корекції маски результату з використанням виключно даних у МП та їх масок. Це дозволить в подальшому використати такі операції для побудови структур криптографічних ОБ виконання операції диз'юнкції, масштабованих до кількості масок даних у їх МП та, на відміну від існуючих, підвищити стійкість таких блоків до атак на основі АСП вищих порядків.

Також запропоновано метод перетворення МП даних, що, за рахунок використання операції додавання за модулем 2^N над даними у МП, побудованої

на основі маскованих логічних операцій, дозволяє перетворювати МП даних із АМ у дані із ЛМ та навпаки. Це дозволить використати таке перетворення для створення структур криптографічних ОБ, які використовують МП даних як з логічною, так і з арифметичною маскою.

Удосконалено метод табличних перетворень даних у МП, що, за рахунок введення додаткового проміжного маскування із узгодженим типом маски вхідних даних, що дозволить виконувати табличні перетворення над вхідними даними як із логічною, так і з арифметичною масками та отримувати результат із заданим типом маскування, а також дозволить використати таку операцію для побудови структур криптографічних ОБ табличної заміни засобів шифрування даних у МП.

РОЗДІЛ 3
РОЗРОБКА І ДОСЛІДЖЕННЯ СТРУКТУР ОБ ВИКОНАННЯ ОПЕРАЦІЙ НАД ДАНИМИ У МАСКОВАНОМУ ПРЕДСТАВЛЕННІ

3.1 Структури операційних блоків логічних й арифметичних операції над даними у маскованому представленні

3.1.1 ОБ логічного множення

Для створення структури ОБ логічного множення двох даних a і b, поданих у МП виду $\{\tilde{a},x\}$, $\{\tilde{b},y\}$, де $\tilde{a}=a\oplus x_1\oplus\ldots\oplus x_n$, і $\tilde{b}=b\oplus y_1\oplus\ldots\oplus y_n$, де $x=x_1,\ldots,x_n$, $y=y_1,\ldots,y_n$ – маски відповідно a і b, що є незалежними випадковими числами з рівномірним розподілом ймовірності, скористаємось розробленим у другому розділі методом виконання логічного множення над даними у МП із довільною кількістю масок. Для цього перепишемо вираз (2.1) у більш зручній формі [105]:

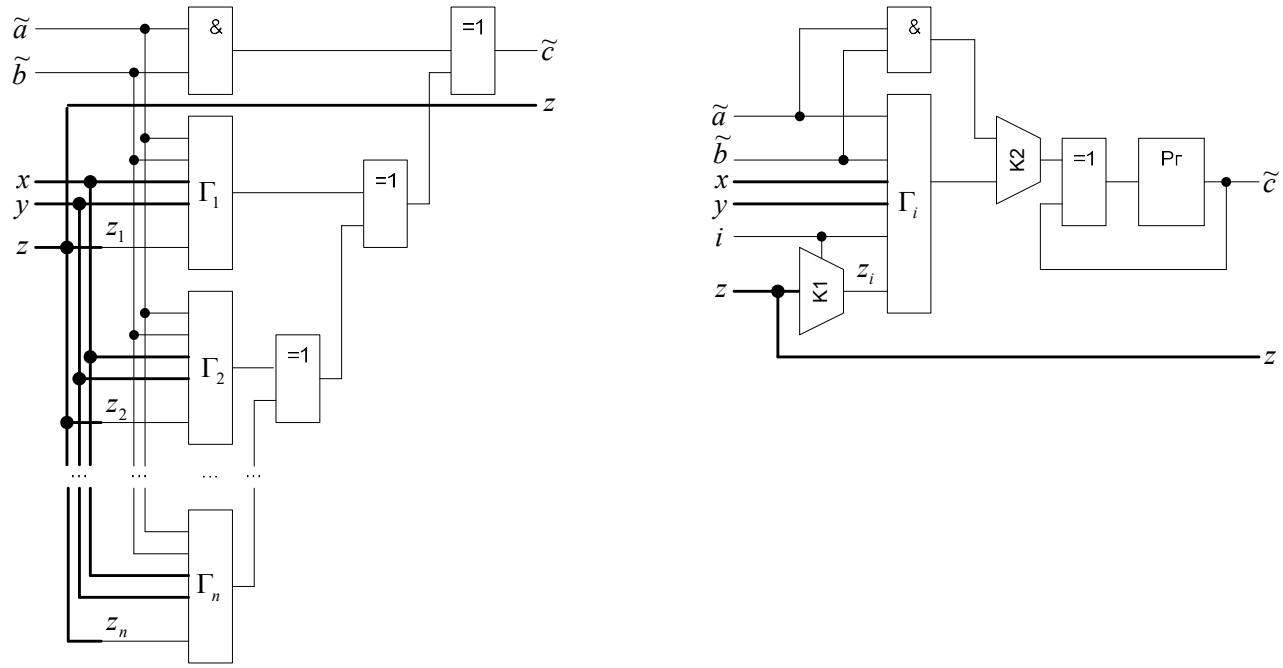
$$\tilde{c}=\tilde{a}\cdot\tilde{b}\oplus\bigoplus_{i=1}^{n}[(\tilde{a},\tilde{b})\cdot(y_i,x_i)^T\oplus(x_1,...,x_n)\cdot(ROT_{i-1}(y_1,...,y_n))^T\oplus z_i], \qquad (3.1)$$

де $ROT_{i-1}(y)$ - позначення функції циклічного зсуву вектора y на i позицій ліворуч, множення векторів відбувається з використанням операцій логічного множення, а додавання множників – за допомогою операції додавання за модулем два.

У виразі (3.1) виділимо "групові" доданки виду

$$\Gamma_i=(\tilde{a},\tilde{b})\cdot(y_i,x_i)^T\oplus(x_1,...,x_n)\cdot(ROT_{i-1}(y_1,...,y_n))^T\oplus z_i. \qquad (3.2)$$

Використовуючи метод прямого апаратного відображення потокового графу [110] виразу (3.1) з врахуванням (3.2), структура ОБ для виконання операцій логічного множення буде мати вид рис. 3.1а [105]:



а) б)

Рис. 3.1. Структури ОБ для виконання операцій логічного множення над даними у МП з використанням n масок: а) на базі прямого апаратного відображення потокового графу, б) на базі апаратного відображення згортки потокового графу

Структура ОБ (рис. 3.1а) містить n однакових блоків Γ_i, кожен з яких виконує операції згідно з виразом (3.2). Операції обчислення термів логічного множення виконуються паралельно, а операції додавання за модулем два термів – послідовно.

Зауважимо, що доданки виду (3.2) є базовими операціями для логічного множення даних у МП з використанням n масок, які відрізняються лише порядком використання вхідних даних та кількістю позицій циклічного зсуву ліворуч, заданих індексом i. Тому, можна запропонувати іншу структуру ОБ, базовану на апаратному відображенні згортки потокового графу [110], що заданий виразом (3.1) та подану на рис. 3.1б.

Розгортка потокового графу у часі здійснюється шляхом зміни індексу i, керування регістром Рг та керування адресними входами комутаторів К1 і К2.

Спочатку в регістр Рг записується нульове значення. Далі адресний вхід комутатора К2 встановлюється таким чином, щоб на вхід елемента додавання за модулем два подавався вихід блоку Γ_i. Подальше встановлення значень індексу i від 1 до n та записування даних у Рг дозволяє обчислити суму $\bigoplus_{i=1}^{n}\Gamma_i$. Для завершення обчислення виразу (3.1) комутатор К2 подає на вхід суматора результат логічного множення даних у МП. Після запису результату у Рг, регістр буде містити результат обчислення виразу (3.1).

Наприклад, для випадку представлення даних з двома логічними масками, ОБ для логічного множення таких даних доцільно будувати на базі структури, зображеної на рис. 3.1а. При $n=2$ елементи виразу (3.2) для $i=1,2$ будуть мати вид:

$$\begin{aligned}\Gamma_1 &= (\widetilde{a},\widetilde{b})\cdot(y_1,x_1)^T \oplus (x_1,x_2)\cdot(y_1,y_2)^T \oplus z_1 = \\ &= \widetilde{a}\cdot y_1 \oplus \widetilde{b}\cdot x_1 \oplus x_1\cdot y_1 \oplus x_2\cdot y_2 \oplus z_1\end{aligned}, \tag{3.3}$$

$$\begin{aligned}\Gamma_2 &= (\widetilde{a},\widetilde{b})\cdot(y_2,x_2)^T \oplus (x_1,x_2)\cdot(y_2,y_1)^T \oplus z_2 = \\ &= \widetilde{a}\cdot y_2 \oplus \widetilde{b}\cdot x_2 \oplus x_1\cdot y_2 \oplus x_2\cdot y_1 \oplus z_2\end{aligned}. \tag{3.4}$$

Тоді структура ОБ буде містити два блоки Γ_1, Γ_2 та елемент логічного множення [105]:

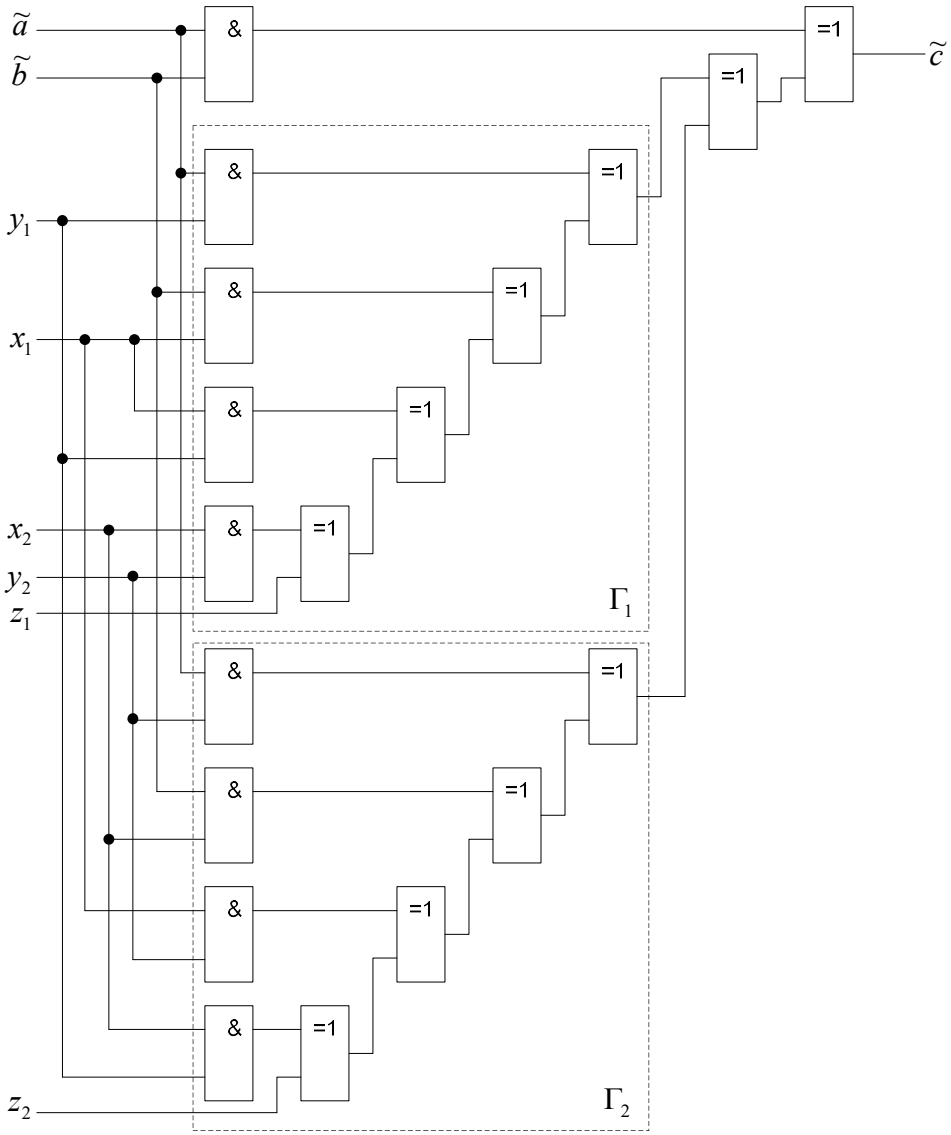


Рис. 3.2. Структура ОБ для виконання операції логічного множення над даними у МП з використанням двох масок на базі прямого апаратного відображення потокового графу

3.1.2 ОБ логічного додавання

Для створення структури ОБ логічного додавання двох даних a і b, поданих у МП виду $\{\widetilde{a}, x\}$, $\{\widetilde{b}, y\}$, де $\widetilde{a} = a \oplus x_1 \oplus \ldots \oplus x_n$, і $\widetilde{b} = b \oplus y_1 \oplus \ldots \oplus y_n$, де $x = x_1, \ldots, x_n$, $y = y_1, \ldots, y_n$ – маски відповідно a і b, що є незалежними випадковими числами з рівномірним розподілом ймовірності, скористаємось розробленим у другому розділі методом виконання логічного додавання даних у МП із

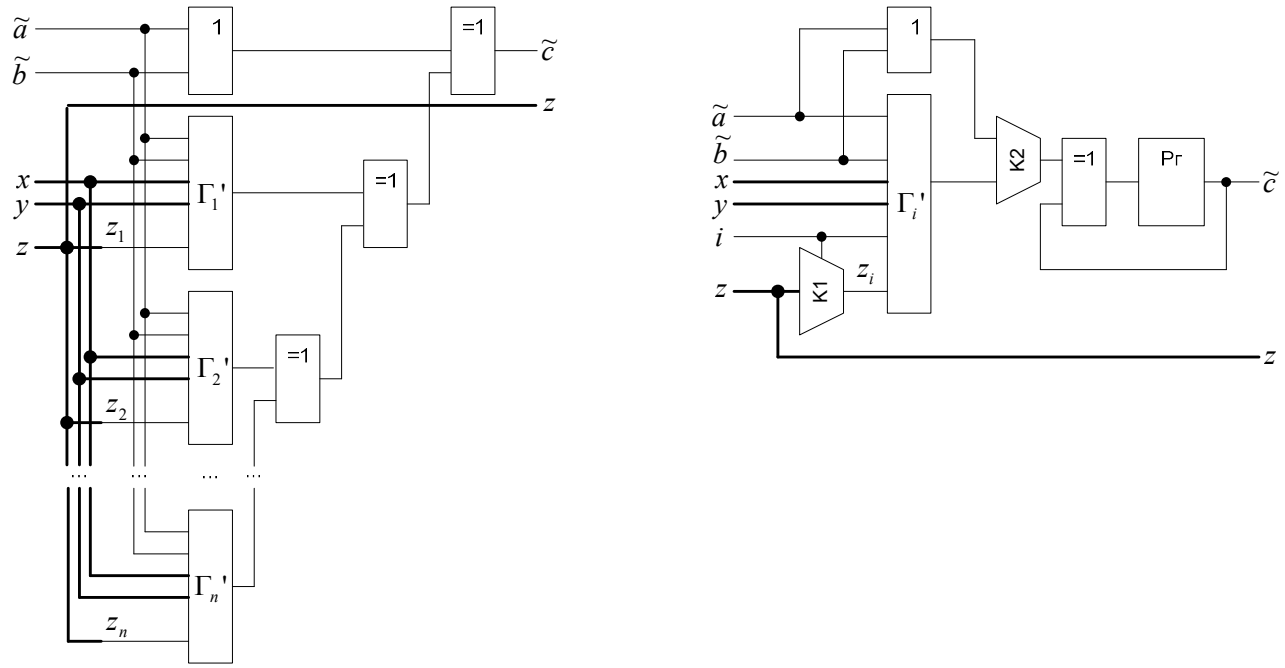
довільною кількістю масок. Для цього скористаємось виразом (2.4) і, врахувавши вираз (3.2) та увівши позначення

$$\Gamma_i' = \Gamma_i \oplus x_i \oplus y_i , \tag{3.5}$$

перепишемо вираз (2.4) у більш зручній формі:

$$\tilde{c} = \tilde{a} \vee \tilde{b} \oplus \bigoplus_{i=1}^{n} \Gamma_i' . \tag{3.6}$$

Використовуючи метод прямого апаратного відображення потокового графу [110], структура ОБ для виконання операцій логічного додавання даних у МП із довільною кількість масок на базі виразу (3.6) буде мати вид рис. 3.3а [105]:



а) б)

Рис. 3.3. Структури ОБ для виконання операцій логічного множення над даними у МП з використанням n масок: а) на базі прямого апаратного відображення потокового графу, б) на базі апаратного відображення згортки потокового графу

Структура ОБ рис. 3.3а містить n однакових блоків Γ_i', кожен з яких виконує операції згідно з виразом (3.5). Операції обчислення термів логічного

множення виконуються паралельно, а операції додавання за модулем два термів – послідовно.

Зауважимо, що доданки виду (3.5) є базовими операціями для логічного додавання даних у МП з використанням n масок, які відрізняються лише порядком використання вхідних даних та кількістю позицій циклічного зсуву ліворуч, заданих індексом i . Аналогічно до операції логічного множення даних тут можна запропонувати іншу структуру ОБ, базовану на апаратному відображенні згортки потокового графу [110], що заданий виразом (3.6) та подану на рис. 3.3б.

Розгортка потокового графу у часі здійснюється шляхом зміни індексу i, керування регістром Рг та керування адресними входами комутаторів К1 і К2. Спочатку в регістр Рг записується нульове значення. Далі адресний вхід комутатора К2 встановлюється таким чином, щоб на вхід елемента додавання за модулем два подавався вихід блоку Γ_i. Подальше встановлення значень індексу i від 1 до n та записування даних у Рг дозволяє обчислити суму $\bigoplus_{i=1}^{n}\Gamma_i'$. Для завершення обчислення виразу (3.6) комутатор К2 подає на вхід суматора результат логічного множення даних у МП. Після запису результату у Рг, регістр буде містити результат обчислення виразу (3.6).

Вибір структури ОБ залежить від заданих розробнику обмежень на обсяг використаного обладнання, час виконання операції та кількості масок у МП даних.

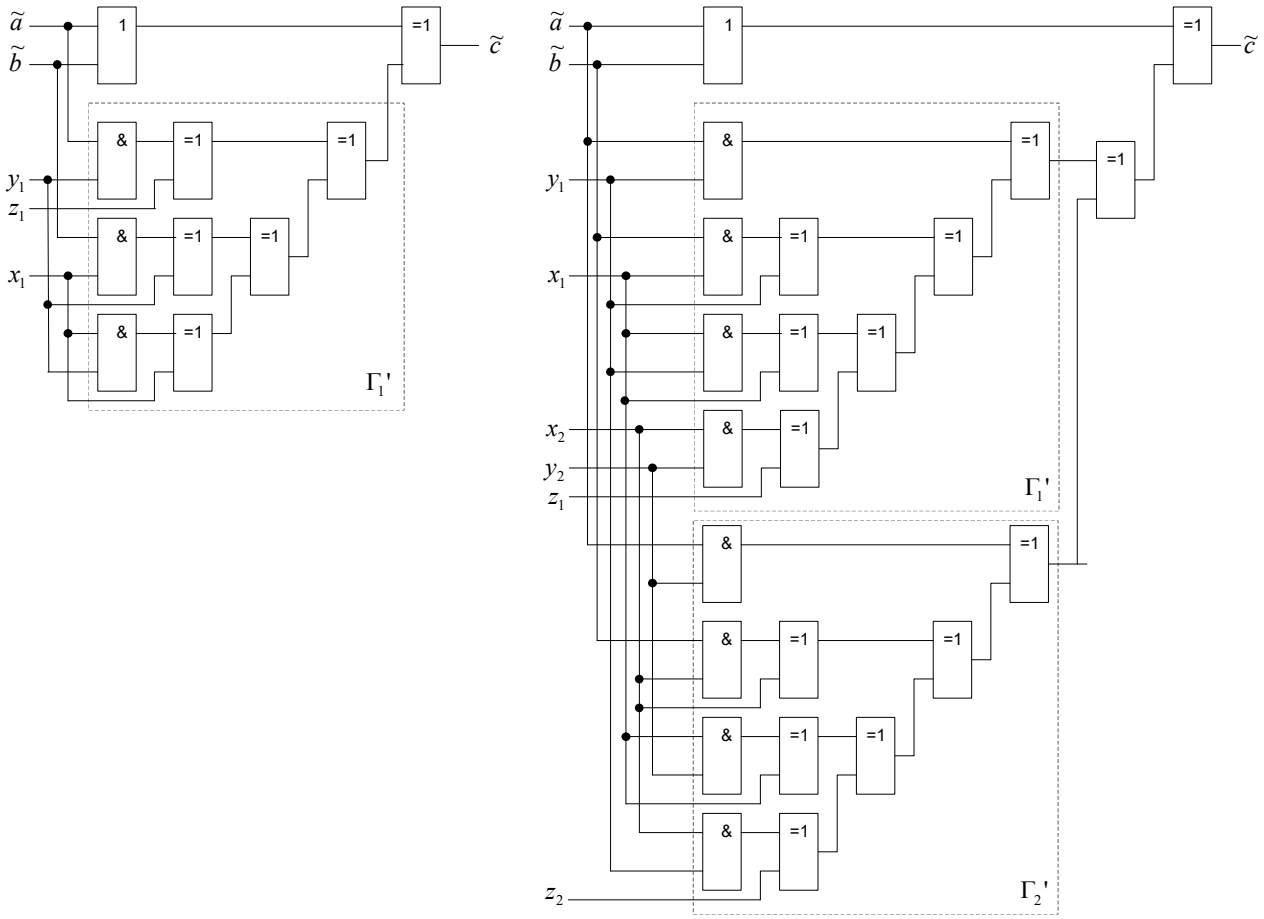
Наприклад, для випадку представлення даних з однією та двома логічними масками, ОБ для логічного додавання таких даних доцільно будувати на базі структури, зображеної на рис. 3.3а. При $n=1$ вираз (3.5) для $i=1$ буде мати вид:

$$\begin{aligned}\Gamma_1' &= \Gamma_1 \oplus x_1 \oplus y_1 = (\tilde{a},\tilde{b})\cdot(y_1,x_1)^T \oplus (x_1,x_2)\cdot(y_1,y_2)^T \oplus z_1 \oplus x_1 \oplus y_1 = \\ &= \tilde{a}\cdot y_1 \oplus \tilde{b}\cdot x_1 \oplus x_1\cdot y_1 \oplus x_2\cdot y_2 \oplus z_1 \oplus x_1 \oplus y_1\end{aligned}, \qquad (3.7)$$

а при $n=2$ елементи виразу (3.5) для $i=1,2$ будуть мати вид:

$$\Gamma_1' = \Gamma_1 \oplus x_1 \oplus y_1 = (\tilde{a},\tilde{b}) \cdot (y_1, x_1)^T \oplus (x_1, x_2) \cdot (y_1, y_2)^T \oplus z_1 \oplus x_1 \oplus y_1 = \\ = \tilde{a} \cdot y_1 \oplus \tilde{b} \cdot x_1 \oplus x_1 \cdot y_1 \oplus x_2 \cdot y_2 \oplus z_1 \oplus x_1 \oplus y_1 \quad , \tag{3.8}$$

$$\Gamma_2' = \Gamma_2 \oplus x_2 \oplus y_2 = (\tilde{a},\tilde{b}) \cdot (y_2, x_2)^T \oplus (x_1, x_2) \cdot (y_2, y_1)^T \oplus z_2 \oplus x_2 \oplus y_2 = \\ = \tilde{a} \cdot y_2 \oplus \tilde{b} \cdot x_2 \oplus x_1 \cdot y_2 \oplus x_2 \cdot y_1 \oplus z_2 \oplus x_2 \oplus y_2 \quad . \tag{3.9}$$



а) б)

Рис. 3.4. Структури ОБ для виконання операції логічного додавання над даними у МП на базі прямого апаратного відображення потокового графу з використанням: а) однієї маски, б) двох масок

Тоді структури ОБ для логічного додавання будуть подібні до відповідних структур ОБ для логічного множення [105]. Вибір структури ОБ залежить від заданих розробнику обмежень на обсяг використаного обладнання, час виконання операції та кількості масок у МП даних. Розроблені структури ОБ

доцільно використовувати при створенні процесорів криптографічних перетворень як для виконання логічних операцій, так і для створення структур інших ОБ.

3.1.3 ОБ додавання за модулем 2^N

Використовуючи розроблені у другому розділі методи логічного множення і додавання даних у МП, розробленими виразами (2.1), (2.4) для двох даних у МП з однією маскою, та відомими рівняннями для обчислення виходів однобітового напівсуматора, можна побудувати структуру напівсуматора MHS, який обробляє дані у МП з однією логічною маскою (рис. 3.5) [11]:

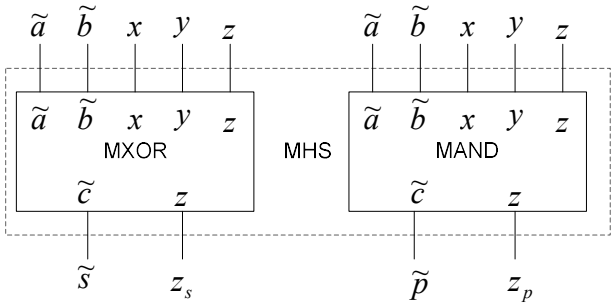


Рис. 3.5. Структура напівсуматора для даних у МП

У структурі напівсуматора для даних у МП (рис. 3.5) додатково використовуються маски суми z_s і переносу z_p. При використанні більшої кількості масок структура цього напівсуматора не зміниться.

Один із варіантів структури однорозрядного повного суматора для даних у МП заданий виразами для обчислення маскованої суми (2.27) та маскованого переносу (2.28) при виконанні арифметичного додавання двох i -х бітів операндів у МП з однією логічною маскою (рис. 3.6) [11].

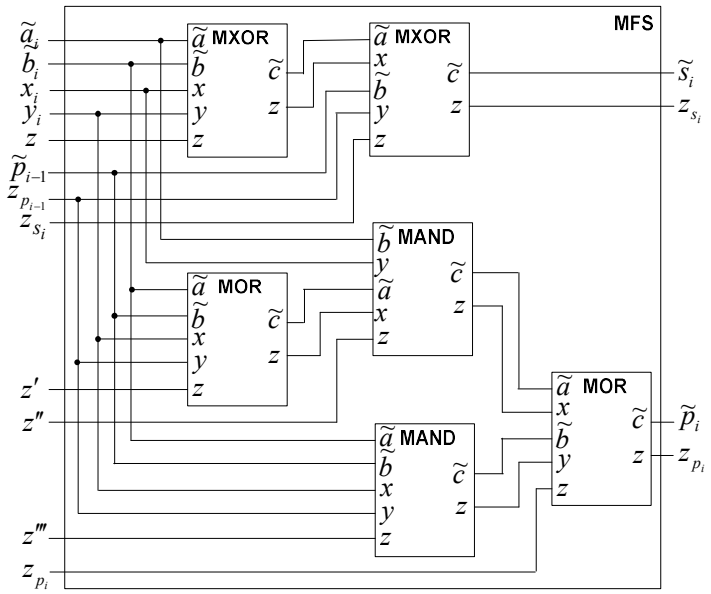


Рис. 3.6. Структура повного однорозрядного суматора для даних у МП з однією маскою

Структура такого повного однорозрядного суматора отримана шляхом прямої заміни логічних елементів на їх еквівалентні елементи, які обробляють дані у МП. Для організації обчислень у маскованому повному суматорі використовуються чотири додаткові маски $q_i = \{z_i, z_i', z_i'', z_i'''\}$ для проміжних результатів.

Використовуючи вирази (2.27), (2.28) та структури маскованого напівсуматора (рис. 3.5) і повного суматора (рис. 3.6), розроблено структуру суматора із послідовним переносом, який виконує додавання двох даних за модулем 2^N у МП з однією маскою (рис. 3.7) [11]:

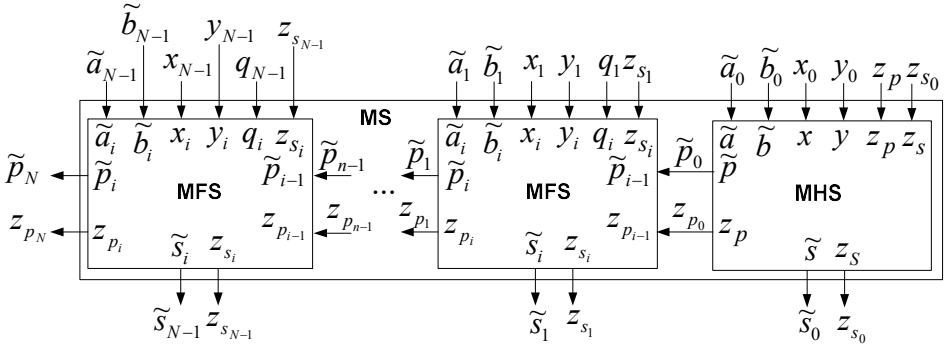


Рис. 3.7. Структура суматора за модулем 2^N для даних у МП з однією маскою

Особливістю цієї структури суматора є використання міжрозрядних переносів у МП, та ігнорування виходів переносу і маски найстаршого повного суматора, відповідно $\widetilde{p}_N$ і z_{p_N}. Як було показано у другому розділі, така структура суматора володіє властивістю маскування обчислень від порушника першого порядку. Розроблену структуру суматора доцільно використовувати для створення комп'ютерних компонент, які обробляють дані згідно з алгоритмами криптографічних перетворень, у яких використовуються операції додавання за модулем 2^N, наприклад згідно з алгоритмами криптографічних перетворень, визначеними у ГОСТ 28147-89, ДСТУ 7624:2014 тощо.

Процедура побудови інших типів суматорів аналогічна до описаного вище підходу. Спочатку будується схема суматора, а потім здійснюється заміна логічних елементів на їх еквіваленти, які обробляють дані у МП із заданою кількістю масок.

3.2 Структури операційних блоків табличного перетворення даних у маскованому представленні

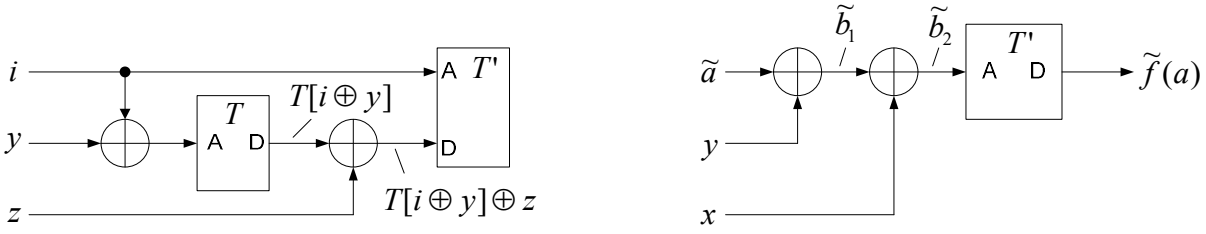
Для побудови ОБ табличного перетворення даних у МП скористаємось розробленими у підрозділі 2.2 методами виконання табличних перетворень над такими даними. Зокрема, розробимо структури ОБ, які обчислюють результат процедури 2.2, та структури ОБ, які обчислюють дані згідно з описом підготовчої процедури 2.1. Приймемо, що вхідні дані будуть подані у вигляді пари $\{\widetilde{a}, x\}$, а вихідні – у вигляді $\{\widetilde{f}(a), z\}$ з використанням однієї маски. Розглянемо два варіанти структур ОБ, які визначаються типом арифметично-логічних операцій маскування вхідних та вихідних даних.

3.2.1 Однотипні операції маскування вхідних і вихідних даних

При ЛМ вхідні та вихідні дані подаються парами $\{\widetilde{a} = a \oplus x, x\}$, $\{\widetilde{f}(a) = f(a) \oplus z, z\}$, відповідно. Тоді в описі процедур 2.1 і 2.2 операції додавання

“$\circ$” та “$\bullet$” є однаковими (додавання за модулем 2) та позначені символом “$\oplus$”. Процедура 2.1 виконується лише при підготовці до обчислень. Структура ОБ, який реалізує процедуру 2.1 (рис. 3.8а), містить такі елементи: запам’ятовуючий пристрій з таблицею T, два суматори за модулем два та запам’ятовуючий пристрій для запису таблиці T' [9].

Виконання підготовчої процедури 2.1 полягає у маскуванні даних таблиці T з подальшим впорядкуванням маскованої таблиці T' згідно з перестановкою, яка задається проміжною маскою y та операцією “$\oplus$”. Кількість перестановок, яка задається проміжною маскою y, дорівнює n. Для операції “$\oplus$” характер перевпорядкування маскованої таблиці визначається кількістю і розташуванням одиниць у масці. При цьому, одиниця у деякому розряді маски означає необхідність перестановки частин таблиці T' між собою. Наприклад, якщо старший розряд маски дорівнює одиниці, то таблицю необхідно розділити навпіл і переставити ці частини місцями.



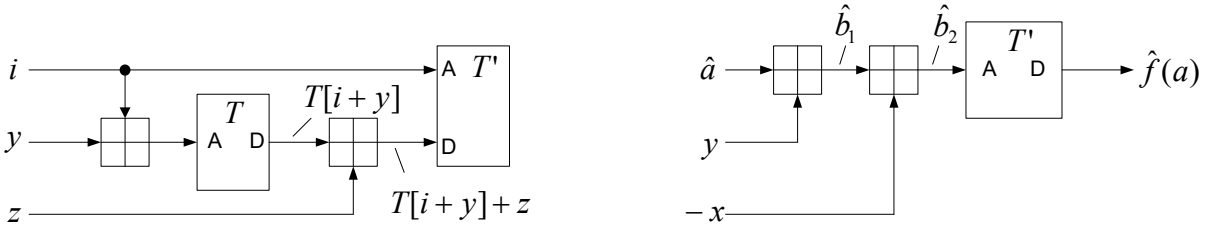
а) б)

Рис. 3.8. Структури ОБ табличних перетворень даних у МП з логічною маскою для виконання: а) підготовчої процедури 2.1, б) основної процедури 2.2

При виконанні основної процедури 2.2 (рис. 3.8б) на першому кроці здійснюється модифікування маски вхідних даних: $\tilde{b}_1 = \tilde{a} \oplus y = a \oplus x \oplus y$. Наступний крок, згідно з процедурою 2.2, потребує знаходження оберненого елемента x^{-1} для усунення маски x. Оскільки у групі $G_{\oplus}$ нейтральним елементом є 0 (нуль), то з рівності $x^{-1} \oplus x = 0$ і властивостей операції додавання за модулем 2 отримуємо, що $x^{-1} = x$. Таким чином, другий крок процедури 2.2 встановлює

таке значення маски вхідних даних y: $\tilde{b}_2 = \tilde{b}_1 \oplus x = a \oplus x \oplus y \oplus x = a \oplus y$. На третьому кроці основної процедури здійснюється підстановка $\tilde{b}_2$ на відповідний вузол заміни з таблиці T'. Оскільки $\tilde{b}_2$ є маскованим з маскою y і таблиця T' була відповідно сформована при виконанні підготовчої процедури пристроєм рис. 3.8а з використанням тієї ж маски, то $\tilde{f}(a) = T'[\tilde{b}_2] = T'[a \oplus y] = T[a] \oplus z$.

При АМ вхідні та вихідні дані подаються парами $\{\hat{a} = a + x, x\}$, $\{\hat{f}(a) = f(a) + z, z\}$, відповідно. Тоді в перетвореннях 2.1 і 2.2 операції додавання "$\circ$" і "$\bullet$" є однаковими (додавання за модулем 2^n) і позначені символом "+". Процедура 2.1 виконується лише при підготовці до обчислень. Структура ОБ, який реалізує процедуру 2.1 (рис. 3.9а), містить такі елементи: запам'ятовуючий пристрій з таблицею T, два суматори за модулем 2^n та запам'ятовуючий пристрій для запису таблиці T'.



а) б)

Рис. 3.9. Структури ОБ табличних перетворень даних у МП з арифметичною маскою для виконання: а) підготовчої процедури 2.1, б) основної процедури 2.2

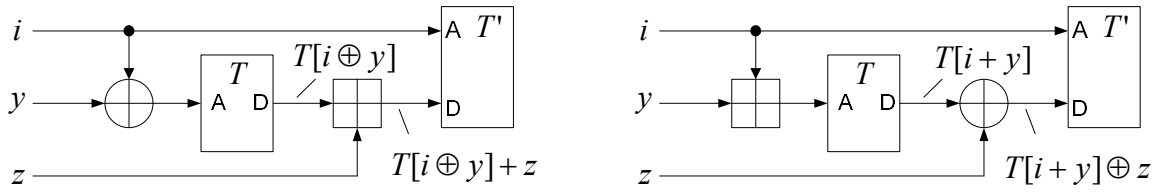
Виконання підготовчої процедури 2.1 полягає у маскуванні даних таблиці T з наступним впорядкуванням маскованої таблиці T' згідно з перестановкою, яка задається проміжною маскою y та операцією "+". Для операції "+" характер перевпорядкування маскованої таблиці визначається числовим значенням проміжної маски y: дані у таблиці циклічно зсуваються в бік старших адрес на числове значення проміжної маски y. Наприклад, для $y = 3$ усі дані у таблиці T' циклічно зсуваються в бік старших адрес на 3 позиції.

При виконанні основної процедури 2.2 (рис. 3.9б) на першому кроці основної процедури здійснюється модифікування маски вхідних даних: $\hat{b}_1 = \hat{a} + y = a + x + y$. Другий крок процедури вимагає оберненого елемента x^{-1} для усунення маски x. Оскільки у групі G_+ нейтральним елементом є 0 (нуль), то з рівності $x^{-1} + x = 0$ отримуємо, що $x^{-1} = -x$. Для обчислення x^{-1} можна скористатися операцією віднімання за модулем 2^n. Таким чином, другий крок основної процедури встановлює таке значення маски вхідних даних y : $\hat{b}_2 = \hat{b}_1 + x = a + x + y + (-x) = a + y$. На третьому кроці основного алгоритму здійснюється заміна $\hat{b}_2$ на відповідний елемент таблиці T' . Оскільки $\hat{b}_2$ є маскованим з маскою y і таблиця T' була відповідно сформована при виконанні підготовчої процедури з використанням тієї ж маски, то $\hat{f}(a) = T'[\hat{b}_2] = T'[a + y] = T[a] + z$.

Зауважимо, що для маскування даних можна використовувати операцію віднімання за модулем 2^n з відповідним модифікуванням процесу виконання підготовчої та основної процедур алгоритму.

3.2.2 Різнотипні операції маскування вхідних і вихідних даних

При використанні ЛМ вхідних даних та АМ вихідних даних, вхідні та вихідні дані подаються у вигляді пар $\{\tilde{a} = a \oplus x, x\}$, $\{\hat{f}(a) = f(a) + z, z\}$ відповідно. Тут процедури 2.1 і 2.2 містять різні операції додавання “$\circ$” і “$\bullet$”. У підготовчій процедурі 2.1 використовується операція “+” для маскування даних таблиці і операція “$\oplus$” для визначення закону перевпорядкування (рис. 3.10а). Основна процедура володіє тими ж особливостями, що й основна процедура для ЛМ вхідних і вихідних даних.



а) б)

Рис. 3.10. Структури ОБ табличних перетворень даних у МП з різними масками для виконання підготовчої процедури 2.1: а) АМ вихідних даних, б) ЛМ вихідних даних

При використанні АМ вхідних даних і ЛМ вихідних даних, вхідні та вихідні дані подаються парами $\{\hat{a} = a + x, x\}$, $\{\widetilde{f}(a) = f(a) \oplus z, z\}$ відповідно. Тут операції додавання “ $\circ$ ” і “ $\bullet$ ” є різними, а у підготовчій процедурі 2.2 використовується операція “$\oplus$” для маскування даних таблиці та операція “+” для визначення закону перевпорядкування (рис. 3.10б). Основна процедура має ті ж особливості, що й основна процедура для АМ вхідних і вихідних даних.

Розроблені структури ОБ табличних перетворень доцільно використовувати при проектуванні структур процесорів криптографічних перетворень, у яких використовуються операції підстановки (заміни за таблицею), наприклад ГОСТ 28147-89, AES, mCrypton, ДСТУ 7624:2014, тощо.

3.3 Структури операційних блоків обернення даних у маскованому представленні у полях Галуа з характеристикою 2

Для розробки структур ОБ обернення даних у МП з ЛМ у скінчених полях Галуа з характеристикою 2 скористаємося методами обробки таких даних, розроблених у підрозділі 2.3. Зокрема, скористаємося виразами (2.10) для обчислення даних у МП і (2.12) для обчислення корекції маски. Структура ОБ, побудованого на базі прямого відображення потокового графу перетворення для даних із ЛМ та n масками буде мати вид, поданий на рис. 3.11 [108].

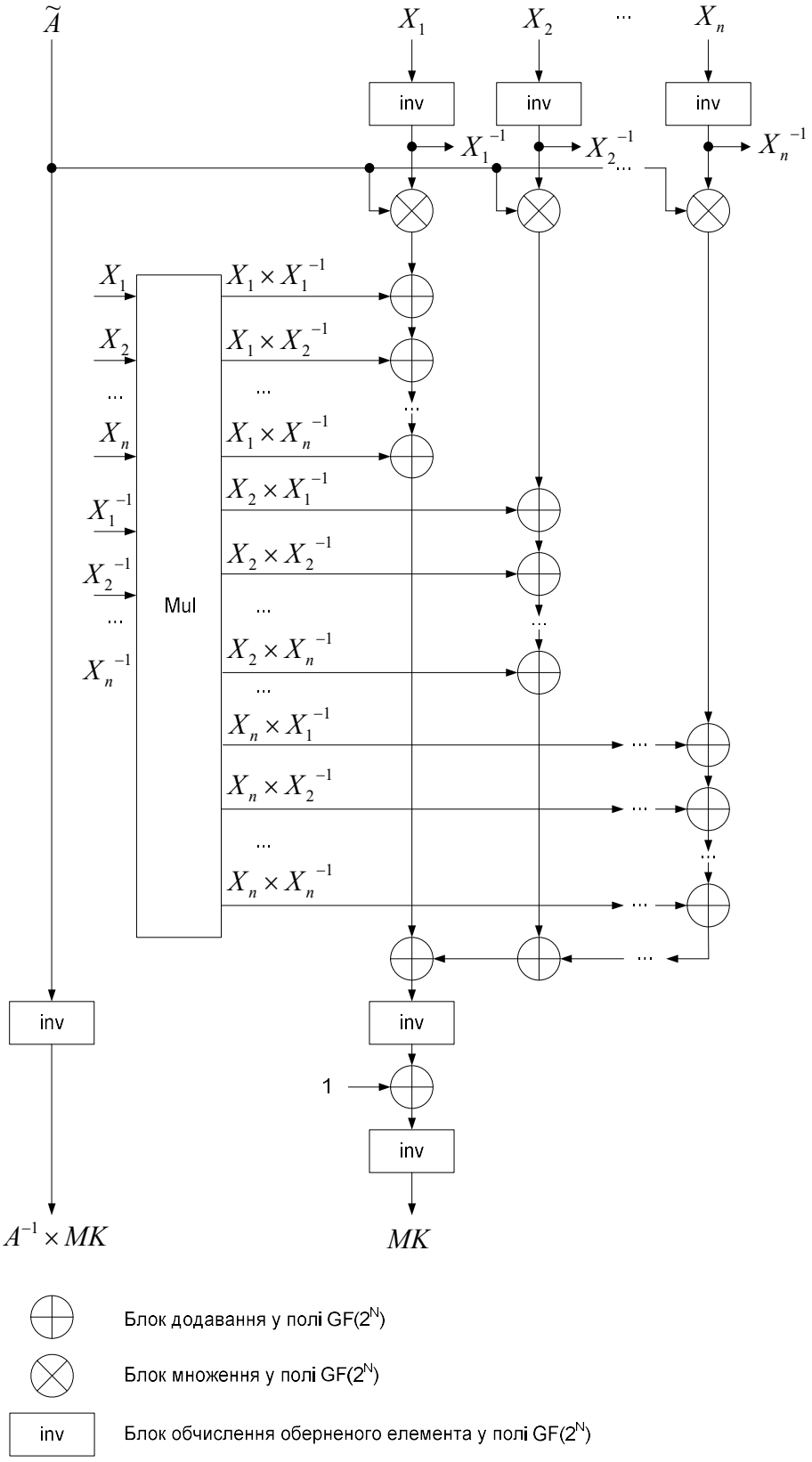


Рис. 3.11. Структура ОБ обернення даних у МП із ЛМ та n масками на базі апаратного відображення потокового графу

Згідно з виразом (2.10) вхідні дані у МП $\widetilde{A}$ подаються у блок обчислення оберненого елемента у полі $GF(2^N)$, на виході якого отримується шуканий результат у формі $A^{-1} \times MK$, де "$\times$" – позначення операції множення у $GF(2^N)$, MK – деякий множник. Паралельно вхідні дані подані на входи n блоків множення, на інші входи яких подано обернені маски даних. Маски даних та їх обернені значення подані на блок Mul, який обчислює попарні добутки виду $X_i \times X_j^{-1}$, де $i, j = 1,..., n$. Зазначимо, що за умови $i = j$ усі добутки виду $X_i \times X_j^{-1}$ рівні одиниці.

Далі для кожного індексу i здійснюється обчислення проміжного доданку шляхом додавання добутків $X_i \times X_j^{-1}$ до $X_i^{-1} \times \widetilde{A}$. Подальше додавання проміжних доданків, знаходження оберненого елемента до суми, додавання одиниці та знаходження оберненого елемента до результату дозволяє обчислити множник MK.

Якщо ОБ для обчислення оберненого елементу до даних у МП повинен забезпечувати максимальну продуктивність обробки поодиноких даних, то такий пристрій доцільно будувати у вигляді граф-алгоритмічного ОБ [110]. За умови невисокої продуктивності ОБ можна зменшити обсяг необхідного обладнання шляхом згортання структури потокового графу. Одним із варіантів структури цього ОБ є апаратна реалізація блоків множення, додавання та обчислення оберненого елемента у $GF(2^N)$. Для відтворення зв'язків між базовими операціями можна скористатися комутаційним середовищем, а для збереження початкових даних, проміжних та остаточних результатів – оперативним запам'ятовуючим пристроєм. У цьому випадку ОБ буде містити комутуюче середовище (КС), блок множення у $GF(2^N)$, блок обчислення оберненого елемента у $GF(2^N)$, блок додавання у $GF(2^N)$, блок оперативного запам'ятовуючого пристрою (ОЗП) та пристрій керування (ПК) (рис. 3.12).

Перед обробкою даних дані у МП та їх відповідні маски завантажуються у ОЗП. Далі пристрій керування формує адресні сигнали читання (АЧ) і запису

(АЗ) для ОЗП. Відповідно до цих сигналів ОЗП записує дані і читає дані, які надходять від (до) КС лініями ДЗ і ДЧ, відповідно. Паралельно ПК формує сигнали керування для КС, за допомогою яких перемикаються входи і виходи арифметичних блоків до ліній ДЧ і ДЗ. За допомогою генерування керуючих сигналів ПК розгортає потоковий граф алгоритму обчислення оберненого елементу до даних у МП.

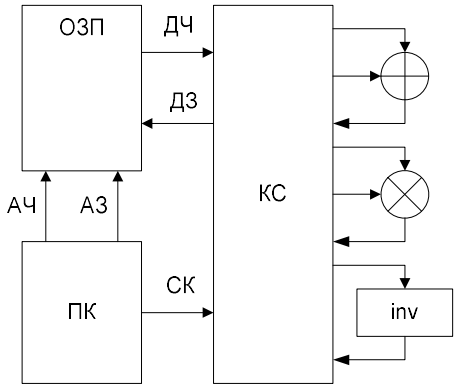


Рис. 3.12. Структура ОБ обернення даних у МП із ЛМ та n масками на базі апаратного відображення згортки потокового графу

Відзначимо, що подана на рис. 3.12 структура ОБ відображає структуру універсального комп'ютера, де роль ПК виконує універсальний програмований процесор, КС – програма обробки даних, а арифметичні блоки – внутрішні блоки процесора чи відповідні програмні процедури.

Як було зазначено у другому розділі, якщо в процесі обчислення оберненого елемента необхідно уникнути успішних нуль-атак, то складові операції множення та інвертування доцільно проводити за допомогою таблично-алгоритмічного методу, що описується виразами (2.8) і (2.9), модифікованими для використання двох масок V і W згідно з виразами (2.13) і (2.14). Тоді структура ОБ буде подібною до структури на рис. 3.11, де арифметичні блоки множення та обчислення оберненого є замінені табличними перетвореннями з використанням даних з ОЗП (рис. 3.13) [108]. Табличні перетворення здійснюються блоками log', alog' та inv згідно з виразами, поданими у п. 2.3.

Структури ОБ, побудованих на базі таблично-алгоритмічних способів виконання операцій множення і обчислення оберненого елемента, характеризуються спрощеним набором операцій, які виконуються над вхідними даними та проміжними результатами. До цього набору операцій входять: табличні перетворення, додавання у полі $GF(2^N)$ та додавання за модулем $2^N - 1$. Разом з тим ці операції об'єднані у подібні блоки, які можна використати для виконання інших операцій алгоритмів криптографічних перетворень. Наприклад, на базі блоку M можна побудувати ОБ множення даних у $GF(2^N)$. Для цього необхідно на вхід блоку подати дані у логарифмічному представлені, отримані, наприклад, за допомогою виконання перших двох операцій над $\tilde{A}$. Аналогічно, блок F2 можна використати для пошуку оберненого елемента у $GF(2^N)$ для немаскованих даних.

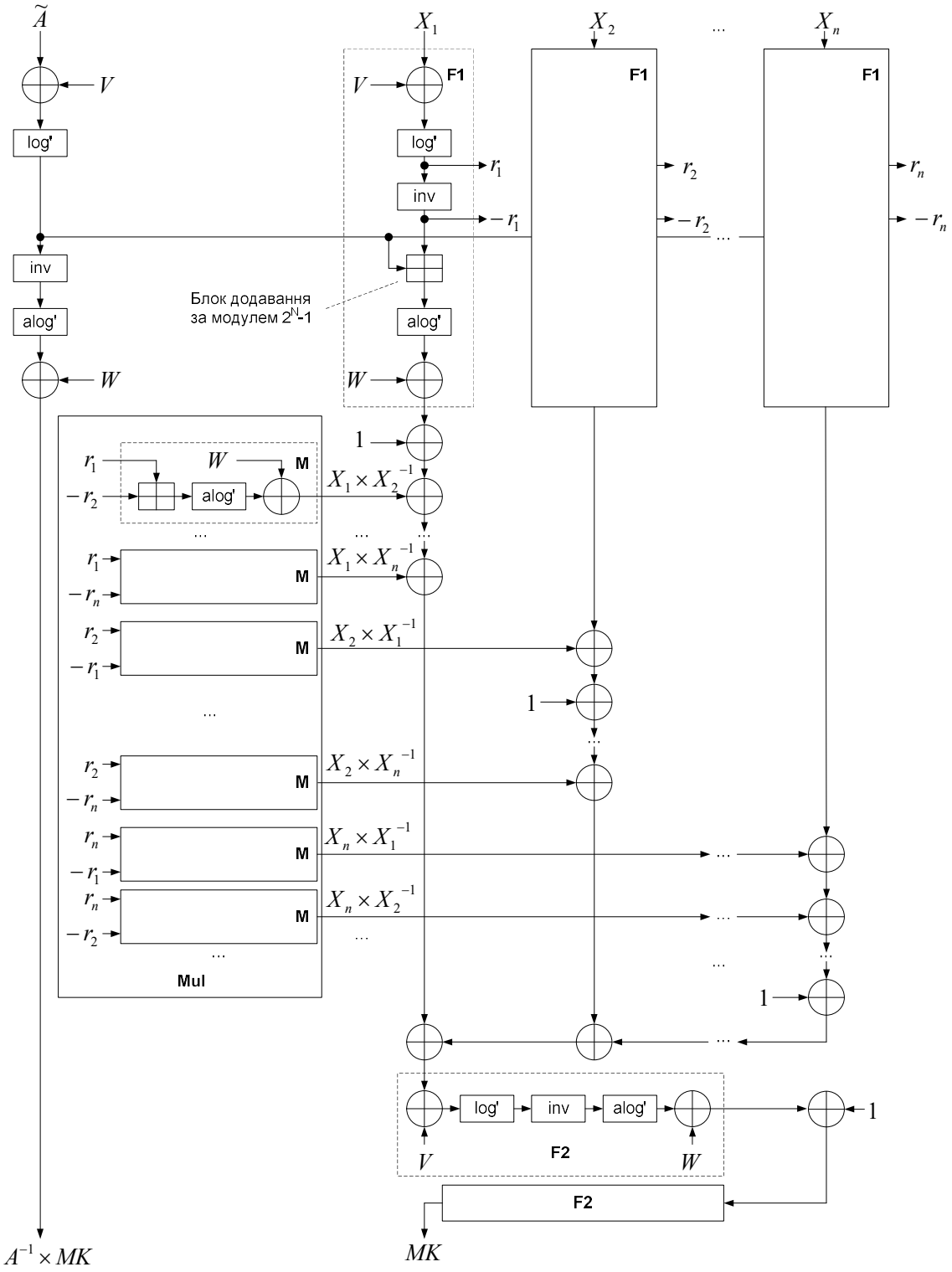


Рис. 3.13. Структура ОБ обернення даних у МП із ЛМ та n масками на базі табличних перетворень

При використанні однієї маски для представлення вхідних даних ($n=1$) та маскованих таблиць для виконання операцій множення та обчислення оберненого елемента, структура ОБ буде містити блоки додавання за модулем 2^N-1 (рис. 3.14).

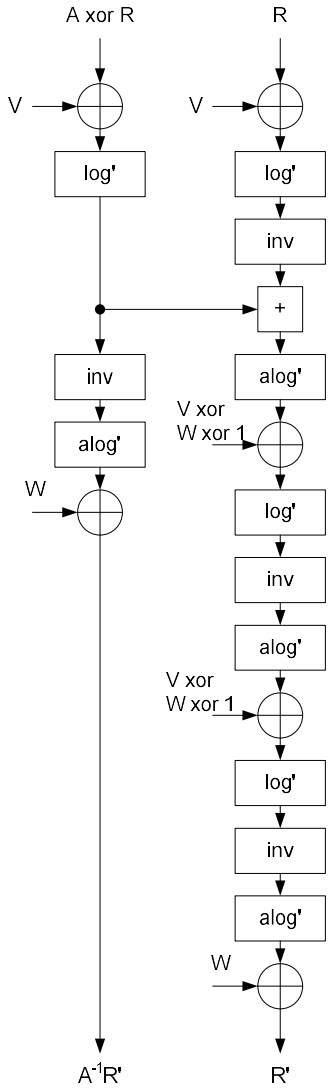


Рис. 3.14. Структура ОБ обернення даних у МП із ЛМ з однією маскою, безпечного до нуль-атак

У наведеній на рис. 3.14 структурі ОБ використано блоки додавання за модулем два та за модулем 2^N-1, блоки табличних перетворень. Виходячи з цього, безпечні до нуль-атак ОБ обчислення оберненого елемента до даних,

поданих у МП доцільно реалізовувати на базі універсальних процесорів, які володіють розвинутими засобами роботи з пам’яттю.

Розроблені структури ОБ обернення даних доцільно використовувати при проектуванні структур процесорів криптографічних перетворень, у яких використовуються такі операції чи можна виділити їх із складених операцій, що входять до складу перетворень, наприклад AES.

3.4 Структури операційних блоків перетворення маскованого представлення даних

На базі розроблених у підрозділі 2.4 методів перетворення МП даних розроблено структури ОБ перетворення МП даних із ЛМ у МП із АМ та навпаки.

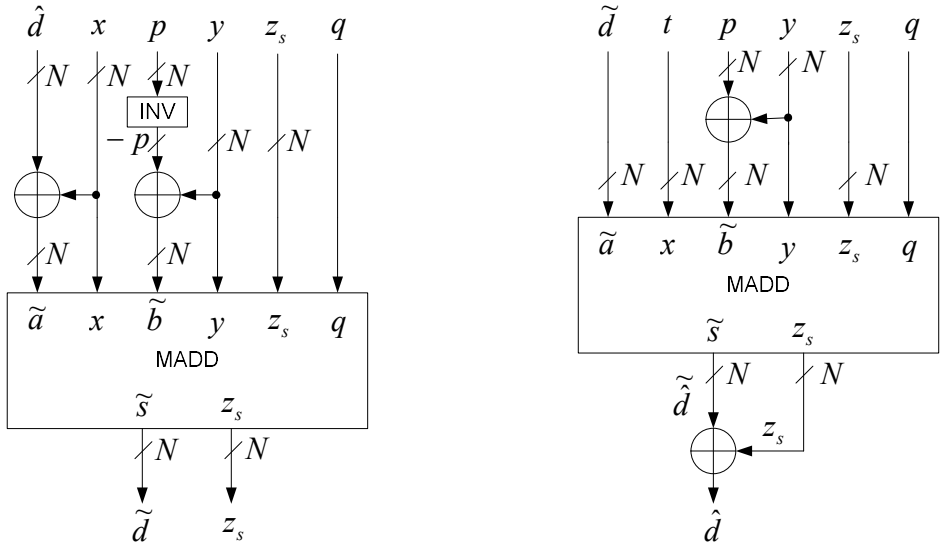
3.4.1 Структури ОБ на базі суматора

Як було зазначено у другому розділі, вираз (2.18) можна реалізувати за допомогою структури ОБ додавання за модулем 2^N даних у МП з логічними масками (рис. 3.7). Тоді для побудови ОБ перетворення МП даних схема суматора доповнюється додатковими елементами. Для перетворення арифметичної маски у логічну додаються два елементи додавання за модулем два та елемент INV для знаходження оберненого значення арифметичної маски (рис. 3.15а) та відповідає опису перетворення 2.3.

В найпростішому випадку для знаходження оберненого значення арифметичної маски доцільно інвертувати арифметичну маску та додати до неї одиницю за модулем 2^N. Для перетворення логічної маски в арифметичну, додаються два елементи додавання за модулем два (рис. 3.15б), що відповідає перетворенню 2.4.

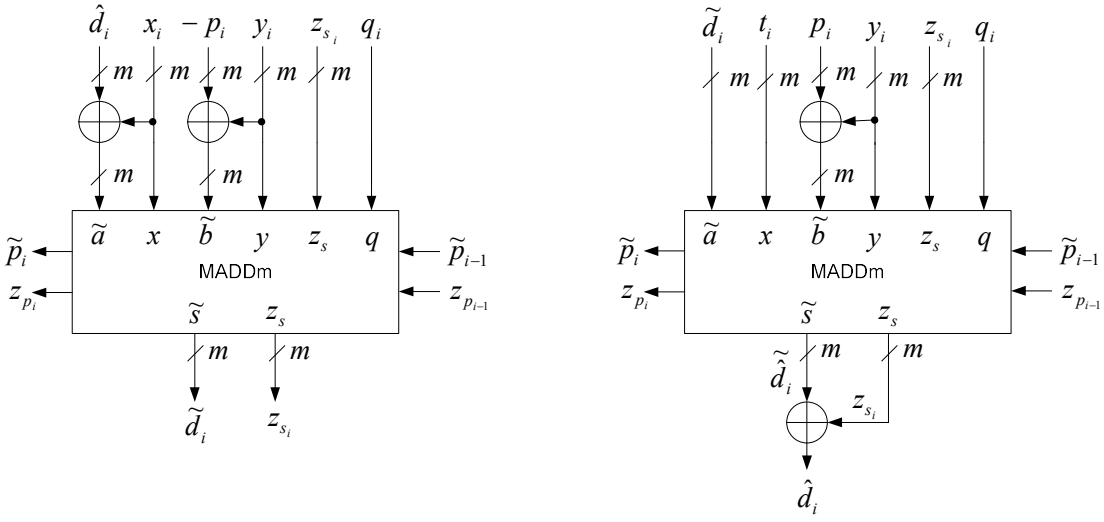
Враховуючи, що при побудові засобів виконання криптографічних перетворень висувається вимога щодо мінімального використання обладнання, то для таких випадків обчислення перетворення масок доцільно проводити з використанням секціонованої обробки даних у МП. Розмір блоку може становити від одного біта до N біт. При цьому, ОБ, подані на рис. 3.15, можна

подати каскадом послідовно ввімкнених пристроїв MADDm меншої розрядності m, які обробляють i-ті m-розрядні блоки даних (рис. 3.16а).



а) б)

Рис. 3.15. Структури ОБ перетворення МП даних на базі суматора: а) перетворення арифметичної маски у логічну, б) перетворення логічної маски у арифметичну



а) б)

Рис. 3.16. Структури каскадів ОБ перетворення МП даних: а) перетворення арифметичної маски у логічну, б) перетворення логічної маски у арифметичну

Для реалізації перетворень необхідно створити лише m-розрядний каскад та послідовно подавати на нього блоки вхідних даних та вихідний перенос з попереднього блоку $\{\widetilde{p}_{i-1}, z_{p_{i-1}}\}$. Вихідні переноси $\{\widetilde{p}_i, z_{p_i}\}$ у МП можна зберігати у пам'яті та подавати у наступний каскад при обробці чергових блоків. При цьому, апаратна складність одного m-розрядного каскаду є меншою за відповідну складність ОБ приблизно у N/m разів. При використанні послідовного перетворення масок за допомогою одного каскаду, продуктивність перетворення буде меншою в $\lceil N/m \rceil$ разів, де $\lceil\ \rceil$ – операція отримання більшого цілого числа.

3.4.2 Структури ОБ на базі запам'ятовуючого пристрою

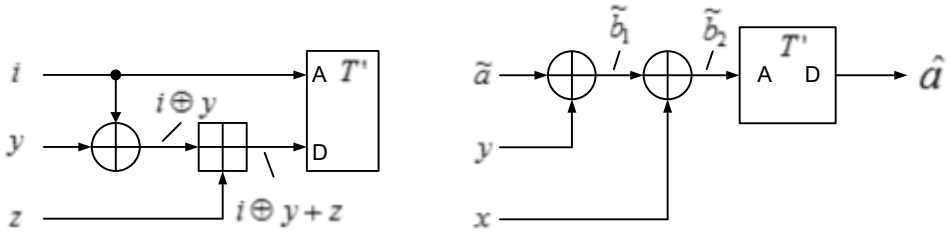
Альтернативні структури ОБ перетворення арифметичної маски у логічну та навпаки можна побудувати на базі запам'ятовуючого пристрою. Для цього використаємо метод табличного перетворення даних у МП, який розроблено у підрозділі 2.2. У перетвореннях 2.1 і 2.2 задамо функцію $f(a)$ у вигляді таблиці $f(a) = T[a] = a$ для усіх $a \in Z_n$. Тоді для перетворення даних із ЛМ у дані з АМ необхідно використати:

- для виконання підготовчої процедури – ОБ, поданий на рис. 3.10а;
- для виконання основної процедури – ОБ, поданий на рис. 3.8б.

Для перетворення даних із АМ у дані з ЛМ необхідно використати:

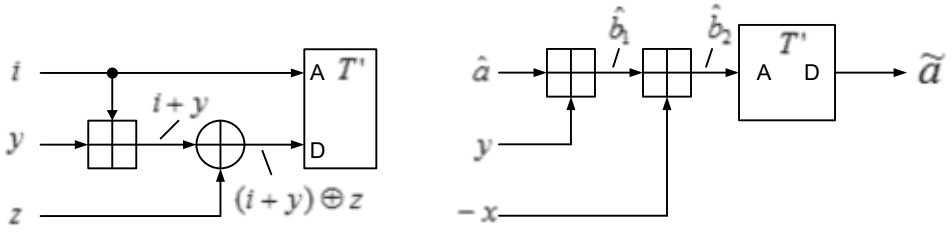
- для виконання підготовчої процедури – ОБ, поданий на рис. 3.10б;
- для виконання основної процедури – ОБ, поданий на рис. 3.9б.

Однак, враховуючи, що функція $f(a)$ задає тотожне перетворення, структури ОБ можна спростити (рис. 3.17, рис. 3.18).



а) б)

Рис. 3.17. Структури ОБ перетворення МП даних на базі запам’ятовуючого пристрою для даних у МП з логічною маскою: а) для виконання підготовчої процедури 2.1, б) для виконання основної процедури 2.2



а) б)

Рис. 3.18. Структури ОБ перетворення МП даних на базі запам’ятовуючого пристрою для даних у МП з арифметичною маскою: а) для виконання підготовчої процедури 2.1, б) для виконання основної процедури 2.2

Спрощення структур досягається за рахунок уникнення використання таблиці T у підготовчій процедурі 2.1. Даний факт зумовлений тим, що у процедурі 2.1 дані i приймають усі значення від 0 до $2^N - 1$, а функція $f(a)$ задає тотожне перетворення даних.

Структури ОБ на базі запам’ятовуючого пристрою володіють обмеженою областю застосування. Таке обмеження зумовлене експоненційною залежністю необхідного об’єму запам’ятовуючого пристрою від розрядності даних, які обробляються.

Розроблені структури ОБ на базі суматора доцільно використати при побудові процесорів криптографічних перетворень, до складу яких входять операції додавання за модулем 2^N із $2 \le N \le 16$. Для більших значень N доцільно використовувати структури на основі суматора, так як у цьому випадку будуть значними витрати на побудову запам'ятовуючого пристрою, що характерно для криптографічного перетворення, яке визначене у ГОСТ 28147-89.

3.5 Дослідження характеристик розроблених операційних блоків для даних у маскованому представленні

Для порівняння характеристик розроблених ОБ для даних у МП із існуючими, дослідимо залежність їх характеристик складності, введених у першому розділі, від кількісті масок у МП даних.

3.5.1 Дослідження характеристик ОБ логічних операцій над даними у МП

Для оцінки характеристик ОБ виконання логічних операцій над даними у МП скористаємося виразами (3.1) і (3.6). При цьому приймемо, що порядок виконання складових операцій цих виразів не впливає на характеристики апаратної складності.

Для виразів (3.1) і (3.6) критичний шлях визначається порядком виконання обчислень. Порядок виконання обчислень впливає на рівень витоку інформації із ОБ. Для цього у виразі (3.1) обчислення доданків $\tilde{a} \cdot \tilde{b}$, $x_i \cdot \tilde{b}$, $y_j \cdot \tilde{a}$ та $x_i \cdot y_j$ можна проводити паралельно, а маска результату вводиться поступово, починаючи з $x_i \cdot y_j$ і закінчуючи $\tilde{a} \cdot \tilde{b}$. Тому, часова складність виконання виразу (3.1) t_{MAND} для обробки даних у МП із використанням $n > 1$ масок буде залежати від тривалості виконання однієї операції логічного множення однобітових даних $t_\wedge$, додавання за модулем 2 однобітових даних $t_\oplus$ і становитиме

$$t_{MAND}(n) = 3nt_\oplus + n^2 t_\oplus, \qquad (3.10)$$

а часова складність виконання виразу (3.1) для обробки даних з однією маскою ($n=1$) дорівнюватиме $t_{MAND}=t_{\wedge}+4t_{\oplus}$.

Аналогічно, для виразу (3.6) обчислення доданків $\tilde{a}\vee\tilde{b}$, $x_i\cdot\tilde{b}$, $y_j\cdot\tilde{a}$, $x_i\cdot y_j$ можна проводити паралельно, а маску результату вводити поступово, починаючи з x_i, y_j і закінчуючи $\tilde{a}\vee\tilde{b}$. Часова складність виконання виразу (3.6) для обробки даних у МП із використанням n масок t_{MOR} буде залежати від тривалості виконання однієї операції логічного додавання однобітових даних $t_{\vee}$, логічного множення однобітових даних $t_{\wedge}$, додавання за модулем 2 однобітових даних $t_{\oplus}$ і складе:

$$t_{MOR}(n)=5nt_{\oplus}+n^2t_{\oplus}, \qquad (3.11)$$

а часова складність виконання виразу (3.6) для обробки даних з однією маскою ($n=1$) дорівнюватиме $t_{MOR}=t_{\wedge}+4t_{\oplus}$.

Паралельне виконання виразів (3.1) і (3.6) можна використати для апаратної реалізації ОБ логічних операцій над даними у МП. Для побудови графіків залежності часової складності виконання маскованих операцій від кількості масок припустимо, що $t_{\wedge}\approx t_{\oplus}$ (рис. 3.19а). З наведених на рис. 3.19а графіків випливає, що при використанні n масок час виконання (затримка виконання) маскованих операцій буде зростати пропорційно до n^2. При цьому, за однакових n часова складність виконання операції логічного множення над даними у МП є меншою за відповідну часову складність для виконання операції логічного додавання.

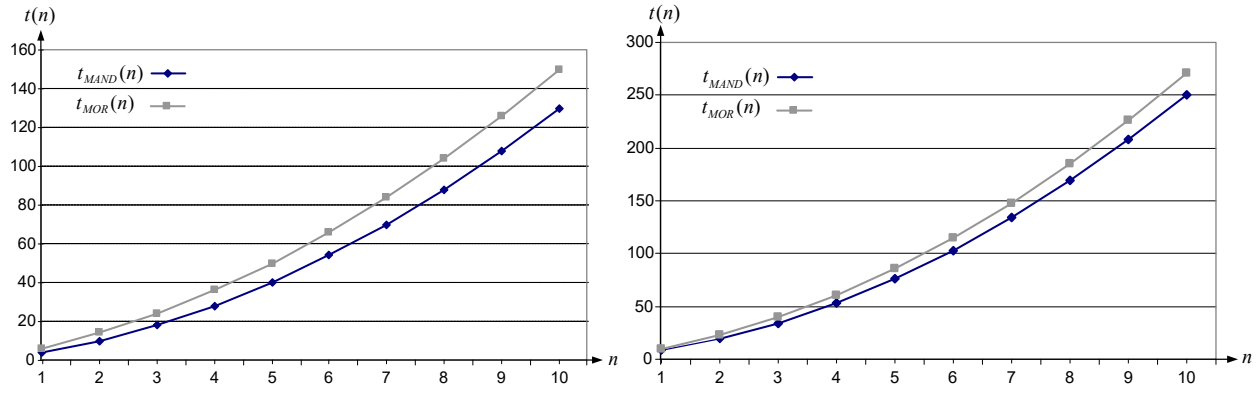


Рис. 3.19. Графік залежності часової складності виконання логічних операцій над даними у МП від кількості масок: а) паралельне виконання, б) послідовне виконання

При альтернативному виконанні усіх операцій послідовно (однак у заданому порядку) згідно з виразом (3.1), часова складність виконання такої операції логічного множення становитиме

$$t_{MAND}(n) = t_{\wedge} + nt_{\oplus} + 2n(t_{\oplus} + t_{\wedge}) + n^2(t_{\oplus} + t_{\wedge}). \qquad (3.12)$$

Аналогічно, для виразу (3.6) отримаємо:

$$t_{MOR}(n) = t_{\vee} + 3nt_{\oplus} + 2n(t_{\oplus} + t_{\wedge}) + n^2(t_{\oplus} + t_{\wedge}). \qquad (3.13)$$

Послідовне виконання виразів (3.1) і (3.6) можна використати при програмній реалізації обчислень над даними у МП на універсальних програмованих процесорах. Оскільки, у цих процесорах час виконання інструкцій логічних операцій є приблизно однаковий і майже не залежить від типу цих логічних операцій, то для побудови графіків залежності часової складності виконання операцій над даними у МП приймемо, що $t_{\wedge} \approx t_{\oplus} \approx t_{\vee}$. З наведених на рис. 3.19б графіків випливає, що при використанні наведених на рис. 3.1а і рис. 3.3а структур ОБ час виконання (затримка виконання) маскованих операцій буде змінюватися аналогічно до складності паралельного способу виконання – пропорційно до квадрату кількості використаних масок. При цьому,

за однакової кількості масок, часова складність виконання операції логічного множення над даними у МП є меншою за відповідну часову складність для виконання операції логічного додавання.

Для порівняння характеристик часової складності розроблених структур ОБ, скористаємося даними табл. 1.3. Для ОБ кон'юнкції та диз'юнкції даних у МП, побудованих на базі відомих методів SWITCH-MUX, MUX-TREE, XOR-AND, при послідовному способі виконання операцій необхідно виконати 12 логічних операцій типу AND, OR, NOT, XOR. Разом з тим, підставивши у (3.12) $n = 1$, отримаємо $t_{MAND}(1) = 4t_{\wedge} + 4t_{\oplus}$, що зумовлює необхідність виконання лише 8 елементарних операцій, та забезпечує виграш у часі на 33% для операції кон'юнкції даних у МП. Аналогічно, підставивши із (3.13) отримуємо $t_{MOR}(1) = 4t_{\vee} + 6t_{\oplus}$, що зумовлює необхідність виконання 10 елементарних операцій, та забезпечує виграш у часі на 20% для операції диз'юнкції. При паралельному виконанні операцій у апаратній реалізації ОБ, найменшої кількості операцій вимагає відомий метод XOR-AND. Прийнявши, що $1.2t_{\wedge} \approx t_{\vee}$, $1.1t_{\wedge} \approx t_{\oplus}$, $0.3t_{\wedge} \approx t_{NOT}$, отримаємо, що розроблена структура ОБ кон'юнкції даних у МП володіє часовою характеристикою складності $t_{MAND} = 5.4t_{\wedge}$, що у порівнянні із найшвидшою відомою структурою XOR-AND із $t_{MAND} = 4.7t_{\wedge}$ є на 15% повільнішою, у порівнянні із відомою структурою MUX-TREE із $t_{MAND} = 5t_{\wedge}$ є на 8% повільнішою, однак у порівнянні із структурою SWITCH-MUX із $t_{MAND} = 6.3t_{\wedge}$ на 16% швидшою для обробки даних у МП із однією маскою (табл. 3.1). Аналогічно отримаємо, що розроблена структура ОБ диз'юнкції даних у МП володіє часовою характеристикою складності $t_{MOR} = 5.4t_{\wedge}$, що у порівнянні із найшвидшою відомою структурою MUX-TREE із $t_{MOR} = 5t_{\wedge}$ є на 8% повільнішою, у порівнянні із структурою SWITCH-MUX із $t_{OR} = 6.3t_{\wedge}$ на 16% швидшою для обробки даних у МП із однією маскою.

Порівняльна характеристика розроблених структур ОБ виконання операцій над даними у МП Таблиця 3.1

Операція	**Метод виконання**	**Характеристики складності відомих структур ОБ**		**Повне маскування результату**	**Нова маска результату**	**Різнотипне МП даних**	**Адаптування до довільної кількості масок**
		Апаратна	**Часова**				
Кон'юнкція	SWITCH-MUX	$11.2A_{\wedge}$	$6.3t_{\wedge}$ / 12 операцій	+	+	–	–
	MUX-TREE	$14.7A_{\wedge}$	$5t_{\wedge}$ / 12 операцій	+	–	–	–
	XOR-AND	$11.7A_{\wedge}$	$4.7t_{\wedge}$ / 12 операцій	+	+	–	–
	Запропонований	$15A_{\wedge}$	$5.4t_{\wedge}$ / 8 операцій	+	+	–	+
Диз'юнкція	SWITCH-MUX	$11.2A_{\wedge}$	$6.3t_{\wedge}$ / 12 операцій	+	+	–	–
	MUX-TREE	$14.7A_{\wedge}$	$5t_{\wedge}$ / 12 операцій	+	–	–	–
	Запропонований	$15.2A_{\wedge}$	$5.4t_{\wedge}$ / 10 операцій	+	+	–	+
Перетворення МП даних	XOR-MUX-MR ЛА	$11.7A_{\wedge}$	$4.7t_{\wedge}$	+	–	–	–
	TABLE-MR	$O(N2^{N+1})A_{\wedge}$	$O(N2^{N})t_{\wedge}$	+	–	–	–
	Запропонований ЛА МС	$96A_{\wedge}$	$19.8t_{\wedge}$	+	+	–	–
	Запропонований ЛА Т	$O(N) + O(N2^{N})$	$O(1)t_{\wedge} + O(N2^{N})t_{\wedge}$	+	+	–	–
Табличні перетворення	FULL-MASK	$O(N2^{N+1})\ A_{\wedge}$	$O(N2^{N})\ t_{\wedge}$	+	+	–	–
	ON-THE-FLY	$O(N2^{N})\ A_{\wedge}$	$O(N2^{N+1})t_{\wedge}$	+	+	–	–
	FAST-ON-THE-FLY	$O(N2^{N})A_{\wedge}$	$O(N2^{N-1})\ t_{\wedge}$	+	+	–	–
	Запропоновані	$O(N)A_{\wedge} + O(N2^{N})A_{\wedge}$	$O(N2^{N})t_{\wedge}$	+	+	+	–
Інвертування	MULT-MASK	$4A_{M}(N) + 2A_{\oplus}(N) + 2A_{I}(N)$	$3t_{M}(N) + 2t_{\oplus}(N) + 2t_{I}(N)$	–	+	–	–
	ADAPT-MULT-MASK	$3A_{M}(N) + 2A_{\oplus}(N) + A_{I}(N)$	$2t_{M}(N) + 2t_{\oplus}(N) + t_{I}(N)$	–	+	–	–
	Запропонований	$A_{M}(N) + 2A_{\oplus}(N) + 4A_{I}(N)$	$t_{M}(N) + 2t_{\oplus}(N) + 3t_{I}(N)$	+	+	–	+

Оцінимо апаратну складність ОБ, структури яких наведено на рис. 3.1а і рис. 3.3а через кількість двовходових логічних елементів логічного множення $N_{\wedge}$, логічного додавання $N_{\vee}$, додавання за модулем 2 $N_{\oplus}$ (табл. 3.2).

Таблиця 3.2

Залежність апаратної складності структур ОБ кон'юнкції та диз'юнкції даних у МП від кількості масок n

Операція	$N_{\vee}$	$N_{\wedge}$	$N_{\oplus}$
кон'юнкція	0	$1+2n+n^2$	$3n+n^2$
диз'юнкція	1	$2n+n^2$	$5n+n^2$

Результат аналізу графіків залежності апаратної складності структур ОБ логічних операцій над даними у МП (рис. 3.20) свідчить, що при використанні n для МП даних апаратна складність ОБ буде зростати пропорційно до n^2.

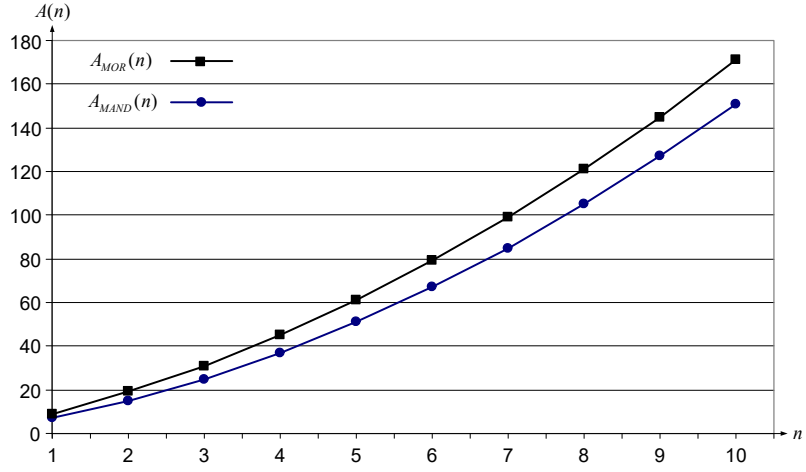


Рис. 3.20. Графік залежності апаратної складності ОБ виконання логічних операцій над даними у МП від кількості масок

Графіки рис. 3.20 отримано з врахуванням, що $A_{\oplus} = 2.2A_{\wedge}$, де $A_{\oplus}(n) = 2n+1$. При цьому, за однакових n апаратна складність ОБ логічного множення над

даними у МП є меншою відповідної апаратної складності ОБ операції логічного додавання. Розмір вибірки випадкових чисел для ОБ логічних операцій над даними у МП визначається є рівним кількості масок, які використано для представлення вхідних даних, тобто n. Для порівняння характеристик апаратної складності розроблених структур ОБ із існуючими, додатково приймемо, що $A_{\vee} = 1.2A_{\wedge}$, $A_{NOT} = 0.7A_{\wedge}$. З аналізу отриманих оцінок характеристик апаратної складності, вираженої у $A_{\wedge}$, випливає, що розроблена структура ОБ кон'юнкції вимагає збільшення затрат обладнання на 33% у порівнянні із структурою ОБ на основі методу SWITCH-MUX, на 28% у порівнянні із структурою ОБ на основі методу XOR-AND, на 2% у порівнянні із структурою ОБ на основі методу XOR-TREE для обробки даних у МП із одною маскою. Аналогічно, розроблена структура ОБ диз'юнкції вимагає збільшення затрат обладнання на 35% у порівнянні із структурою ОБ на основі методу SWITCH-MUX, на 3% у порівнянні із структурою ОБ на основі методу XOR-TREE для обробки даних у МП із одною маскою. Разом з тим, спираючись на розроблені у другому розділі методи, на відміну від існуючих, можна побудувати структури ОБ для обробки даних у МП із заданою кількістю масок та маскуванням даних новою маскою, що дозволяє створювати структури ОБ, стійкі до атак на основі АСП заданого порядку.

3.5.2 Дослідження характеристик ОБ додавання за модулем 2^N даних у МП

Структура напівсуматора даних у МП (рис. 3.5) містить два блоки: блок логічного множення та додавання за модулем два даних у МП. Оскільки ці блоки включені паралельно, то часова складність цього ОБ визначається критичним шляхом структури блока логічного множення даних у МП. Залежність довжини цього критичного шляху від кількості масок операндів описується виразами (3.10) чи (3.12): $t_{MHS}(N) = t_{MAND}(n) = 3nt_{\oplus} + n^2 t_{\oplus}$.

Апаратну складність цього ОБ можна обчислити як суму апаратної складності блока логічного множення та додавання за модулем два даних у МП:

$$A_{MHS}(n) = A_{\oplus}(n) + A_{\wedge}(n) = (4n^2 + 17n + 4)A_{\wedge}. \quad (3.14)$$

Розмір вибірки випадкових чисел для роботи маскованого напівсуматора обчислюється як сума розмірів вибірок випадкових чисел необхідних для роботи складових блоків:

$$R_{MHS}(n) = 2n. \quad (3.15)$$

Структура однобітового повного суматора даних у МП (рис. 3.6) містить три блоки логічного додавання, два блоки додавання даних за модулем два та два блоки логічного додавання даних у МП.

Часова складність цієї структури обумовлюється критичним шляхом, вздовж якого розташовані два блоки логічного додавання та одним блок логічного множення даних у МП. Тому, враховуючи вирази (3.10) і (3.11), вираз для оцінки часової складності повного однорозрядного суматора даних у МП буде мати вид:

$$t_{MFS}(n) = 2t_{MOR}(n) + t_{MAND}(n) = 13nt_{\oplus} + 3n^2 t_{\oplus}. \quad (3.16)$$

Апаратна складність повного однорозрядного суматора визначається як сума апаратної складності його складових блоків згідно з таким виразом:

$$A_{MFS}(n) = 2(A_{MXOR}(n) + A_{MOR}(n) + A_{MAND}(n)) = A_{\wedge}(10 + 68n + 16n^2). \quad (3.17)$$

Розмір вибірки випадкових чисел для однорозрядного повного суматора задається числом блоків, які використано для побудови його структури. При цьому вибірка випадкових чисел містить як маски вихідних даних, так і маски усіх проміжних даних, які утворюються у процесі обчислень. Оскільки наведена на рис. 3.6 структура містить шість блоків, то розмір вибірки випадкових чисел можна оцінити як:

$$R_{MFS}(n) = 6n. \quad (3.18)$$

Структура N-розрядного суматора з послідовним переносом (рис. 3.7) для додавання за модулем 2^N даних у МП складається з $N-1$ повного

однорозрядного суматора даних у МП та одного напівсуматора даних у МП. Тому характеристики структури суматора за модулем 2^N даних у МП можна отримати на основі відповідних виразів, що описують характеристики складових блоків.

Часова складність N-розрядного суматора з послідовним переносом для додавання за модулем 2^N даних у МП визначається згідно з таким виразом:

$$t_{MS}(N,n) = (N-1)t_{MFS}(n) + t_{MHS}(n). \quad (3.19)$$

Підставивши у (3.19) вирази (3.10) і (3.16), отримаємо:

$$t_{MS}(N,n) = (13N-10)nt_{\oplus} + (3N-3)n^2 t_{\oplus}. \quad (3.20)$$

Подальше зменшення часової складності ОБ можливе за рахунок використання інших структур суматорів.

Апаратна складність суматора з послідовним переносом для додавання за модулем 2^N ($N>1$) даних у МП визначається згідно з таким виразом:

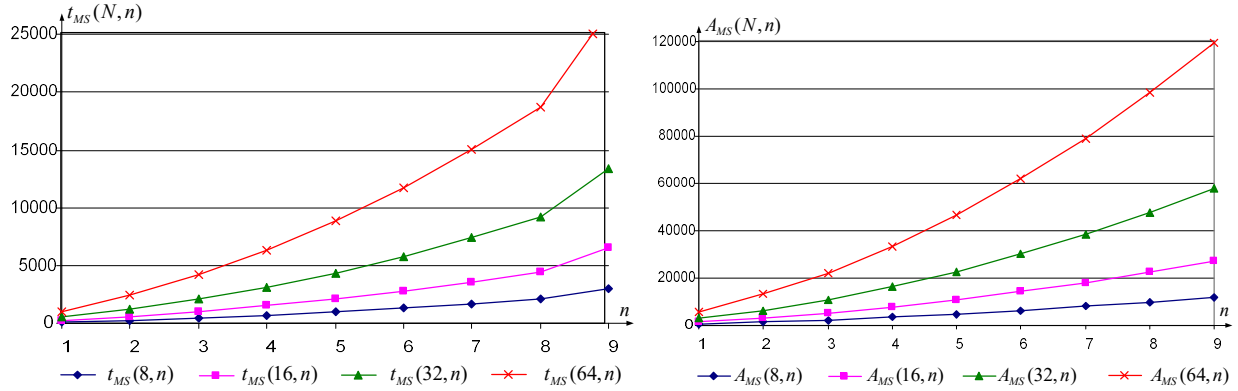
$$A_{MS}(N,n) = (N-2)A_{MFS}(n) + A_{MHS}(n) + 2A_{MXOR}(n). \quad (3.21)$$

Підставивши у (2.21) вирази (3.14) і (3.17), отримаємо:

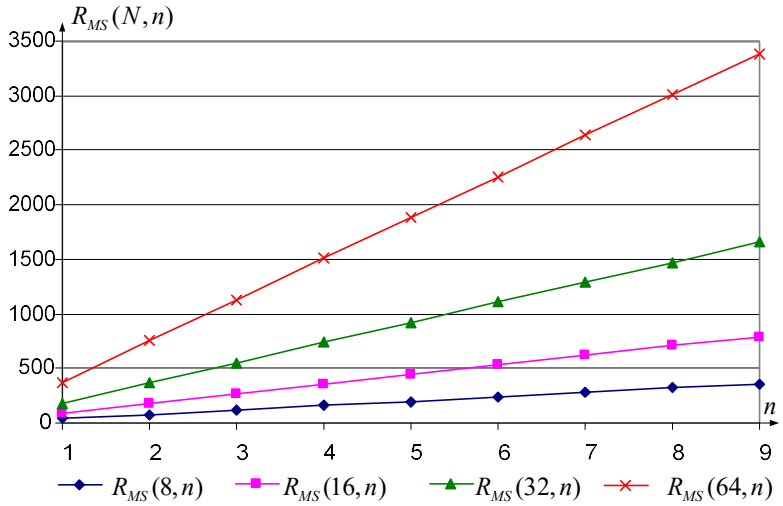
$$A_{MS}(N,n) = (10N-14+n(68N-115)+4n^2(4N-7))A_{\wedge}. \quad (3.22)$$

Розмір вибірки випадкових чисел, необхідний для структури такого суматора ($N>1$) можна знайти, згідно з виразом:

$$R_{MS}(N,n) = (N-2)R_{MFS}(n) + R_{MHS}(n) + 2n = n(6N-8). \quad (3.23)$$



а) б)



в)

Рис. 3.21. Графіки залежності характеристик складності структури суматора даних у МП (на базі структури суматора з послідовним переносом) за модулем 2^N від розрядності даних та кількості масок: а) часова складність, б) апаратна складність, в) розмір вибірки випадкових чисел

Результати аналізу графіків залежностей часової складності структури суматора даних у МП (на базі структури суматора з послідовним переносом) за модулем 2^N від розрядності даних та кількості масок (рис. 3.21а) показують, що для заданої розрядності даних часова складність ОБ збільшується згідно з квадратичним законом зміни кількості масок, використаних для МП. Аналогічно до часової складності, апаратна складність ОБ для додавання даних за модулем 2^N (рис. 3.21б) збільшується згідно з квадратичним законом зміни кількості масок, використаних для МП.

Аналіз виразу (3.23) та графіку залежності розміру вибірки випадкових чисел (рис. 3.21в) показує, що із збільшенням розрядності даних, які обробляються, розмір вибірки випадкових чисел для обробки одного біта даних збільшується лінійно від кількості використаних масок: $\lim\limits_{N\to\infty}\frac{n(6N-8)}{N}=6n$.

Отримані оцінки характеристик складності ОБ будуть використані для оцінки таких характеристик ОБ перетворення МП даних.

3.5.3 Дослідження характеристик ОБ табличних перетворення даних у МП

Структури ОБ табличних перетворень даних у МП (рис. 3.8 – рис. 3.10) характеризуються малою кількістю логічних операцій. До їх числа належать операції маскування даних – додавання за модулем два та додавання за модулем 2^N для логічного та АМ відповідно. Спільними елементами структур усіх ОБ для виконання підготовчої процедури є початкова таблиця T. На базі цієї таблиці формується таблиця T', яка використовується у основних процедурах перетворень.

Приймемо, що для побудови структур ОБ (рис. 3.9, рис. 3.10) використано суматори з послідовним переносом. Як було показано у [115], апаратна та часова характеристики складності такого суматора описуються виразами, відповідно:

$$A_{CRA}(N)=9NA_{\wedge} \tag{3.24}$$

$$t_{CRA}(N)=3Nt_{\oplus} \tag{3.25}$$

де N – розрядність суматора, $A_\wedge$ – умовні затрати на побудову одного логічного елемента І, $t_\oplus$ – затримка спрацювання одного елемента додавання за модулем два.

Апаратну складність запам'ятовуючого пристрою для зберігання модифікованої таблиці T' можна оцінити як пропорційну до $O(N \cdot 2^N)$, оскільки ці пристрої використовують N -розрядні шини адрес і даних. При цьому зауважимо, що відповідні пари структур для реалізації табличних перетворень над даними у МП з використанням логічної, арифметичної та змішаними масками використовують однаковий набір операцій. Для згаданих структур можна побудувати такий узагальнений вираз, який дозволяє оцінити апаратну складність:

$$A_T(N) = (A_1(N) + A_2(N) + 2A_3(N) + O(N \cdot 2^N))A_\wedge, \qquad (3.26)$$

де $A_1(N)$ – апаратна складність блоку модифікування адрес читання запам'ятовуючого пристрою T, $A_2(N)$ – апаратна складність блоку модифікування вихідних даних із запам'ятовуючого пристрою T, $A_3(N)$ – апаратна складність одного блоку модифікування вхідних даних при читанні з блоку T', $O(N \cdot 2^N)$ – оцінка апаратної складності запам'ятовуючого пристрою з організацією 2^N слів по N розрядів.

Характеристики часової складності структур ОБ оцінимо з таких міркувань. Для формування таблиці T' необхідно модифікувати адресу читання, прочитати усі дані з таблиці T згідно з модифікованою адресою та записати модифіковані дані з таблиці T у T'. Якщо здійснювати формування таблиці T' послідовно (без конвеєризування читання з T і запису у T'), то усі структури пристроїв для виконання підготовчої процедури будуть характеризуватися часовою складністю, що задана у вигляді такого виразу:

$$t_{TP}(N) = (t_1(N) + t_R(N) + t_2(N) + t_W(N)) \cdot 2^N, \qquad (3.27)$$

де $t_1(N)$ – часова складність модифікування адрес для запам'ятовуючого пристрою T, $t_R(N)$ – час читання з запам'ятовуючого пристрою T, $t_2(N)$ – часова складність модифікування даних з T, $t_W(N)$ – час запису у запам'ятовуючий пристрій T'.

Часова складність виконання операції табличного перетворення задається виразом виду:

$$t_{TM}(N) = 2t_3(N) + t_R(N), \tag{3.28}$$

де $t_3(N)$ – часова складність модифікування даних.

На базі виразів (3.26) і (3.27) можна отримати оцінки характеристик апаратної і часової складності для структур ОБ табличних перетворень даних у МП (табл. 3.3).

Таблиця 3.3

Характеристики складності структур ОБ табличних перетворень даних у МП

Тип маскування		**Характеристики складності**		
Вхідні дані	**Вихідні дані**	$t_{TP}(N)$	$t_{TM}(N)$	$A_T(N)$
Логічне (Л)	Логічне (Л)	$(2t_{\oplus} + t_R(N) + t_W(N)) \cdot 2^N$	$2t_{\oplus} + t_R(N)$	$(12N + O(N \cdot 2^N))A_{\wedge}$
Арифм. (А)	Арифм. (А)	$(6Nt_{\oplus} + t_R(N) + t_W(N)) \cdot 2^N$	$6Nt_{\oplus} + t_R(N)$	$(36N + O(N \cdot 2^N))A_{\wedge}$
Логічне (Л)	Арифм. (А)	$((6N+1)t_{\oplus} + t_R(N) + t_W(N)) \cdot 2^N$	$2t_{\oplus} + t_R(N)$	$(18N + O(N \cdot 2^N))A_{\wedge}$
Арифм. (А)	Логічне (Л)	$((6N+1)t_{\oplus} + t_R(N) + t_W(N)) \cdot 2^N$	$6Nt_{\oplus} + t_R(N)$	$(30N + O(N \cdot 2^N))A_{\wedge}$

Місткістна складність ОБ є однаковою для усіх типів маскування вхідних і вихідних даних та оцінюється згідно з виразом $V_T(N) = 2^N$.

Розмір вибірки випадкових чисел, які необхідні для організації обчислень у ОБ, залежить від розрядності даних, які необхідно обробляти, а тому й від об'єму пам'яті запам'ятовуючого пристрою T та оцінюється виразом $R(N) = 2^N$.

Результат аналізу графіків залежності часової складності виконання підготовчої процедури від розрядності вхідних даних (рис. 3.22), побудованих на основі виразів для $t_{TP}(N)$, поданих у табл. 3.3, свідчить про експоненційне зростання часової складності від розміру даних, які обробляються.

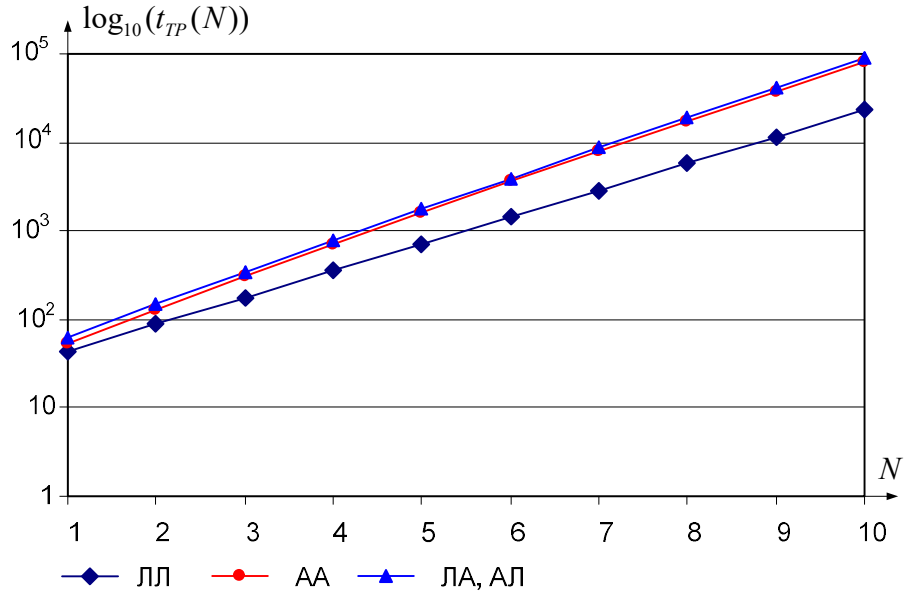


Рис. 3.22. Графіки залежності часової складності виконання підготовчої процедури від розрядності вхідних даних

Такий характер зростання часової складності зумовлений необхідністю читання усіх даних з таблиці T та подальшим їх записом у таблицю T'. При цьому структури ОБ табличних перетворень даних у МП з використанням ЛМ для вхідних і вихідних даних володіють найменшою часовою складністю. Структури ОБ із АМ вхідних та вихідних даних і структури із різнотипними операціями маскування володіють майже однаковою часовою складністю. Збільшення часової складності цих останніх структур пояснюється використанням суматорів за модулем 2^N, які володіють більшою часовою складністю ніж елементи побітового додавання даних за модулем два.

Основною складовою апаратної складності ОБ табличних перетворень є об’єм обладнання для побудови запам’ятовуючого пристрою. Тому із збільшенням розрядності даних, які обробляються, збільшення апаратної складності ОБ носить експоненційний характер (рис. 3.23).

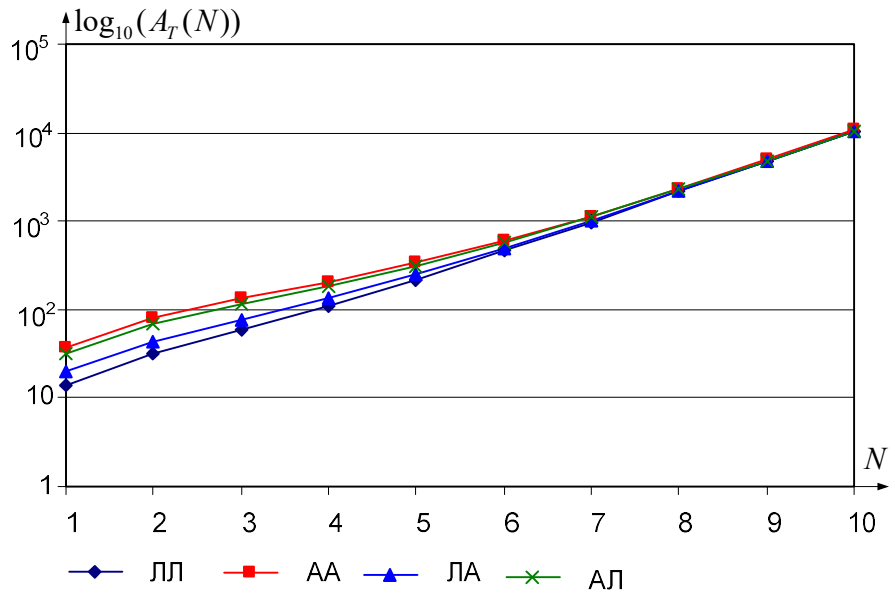


Рис. 3.23. Графік залежності апаратної складності ОБ табличних перетворень від розрядності вхідних даних

Зауважимо, що місткістна складність розробленого ОБ визначається лише об’ємом запам’ятовуючого пристрою для зберігання таблиці T'.

Для порівняння запропонованої структури і структур, побудованих на базі алгоритмів [75, 92], оцінимо місткістну складність розглянутих алгоритмів виконання операцій підстановки даних у МП для алгоритмів AES та ГОСТ 28147-89 (табл. 3.4).

Таблиця 3.4

Місткістна складність алгоритмів виконання операції підстановки даних у МП для алгоритмів AES та ГОСТ 28147-89

Алгоритм підстановки	**Алгоритм AES**	**Алгоритм ГОСТ 28147-89**	**Примітка**
FULL-MASK	40960 біт	512 біт	Маскована таблиця з повними вузлами замін AES
ON-THE-FLY	1344 біт	576 біт	Маскована таблиця з вузлами замін AES на основі операції інвертування у полі $\mathrm{GF}((2^4)^2)$
Запропонований	1280 біт	512 біт	

Отримані результати (табл. 3.1) дозволяють оцінити часову складність розробленої структури ОБ як $O(2^N)t_{\wedge}$, що, є співставима із часовою складністю ОБ побудованого на основі методу FUL-MASK, є у два рази вищою від часової складності ОБ побудованого на основі методу FAST-ON-THE-FLY, та є у два рази меншою часової складності ОБ, побудованого на основі методу ON-THE-FLY. З точки зору апаратної складності, у порівнянні із існуючими, розроблена структура ОБ вимагає додаткових затрат обладнання для проведення операцій накладання тимчасової маски, боєм якого оцінюється як $O(N)A_{\wedge}$. Місткістна складність для розробленої структури ОБ є у два рази меншою від структури ОБ на основі методу FULL-MASK та однаковою із ОБ на основі методів ON-THE-FLY і FAST-ON-THE-FLY. Додатковою перевагою розроблених структур ОБ табличних перетворень є їх адаптування для обробки даних у МП як із логічною, так і арифметичною масками. Це, в свою чергу, дозволяє використати розроблені структури ОБ для побудови криптографічних процесорів даних у МП, які містять у собі як логічні булеві операції, так і операції додавання за модулем 2^N, наприклад, криптографічний процесор на основі перетворення ГОСТ 28147-89.

3.5.4 Дослідження характеристик ОБ обернення даних у МП у полях Галуа з характеристикою 2

Як було зазначено у п. 3.4, ОБ обернення даних у МП у полях Галуа з характеристикою 2 можна будувати на базі апаратного відображення потокового графу або на базі універсального програмованого процесора.

При апаратній реалізації структур таких ОБ у вигляді компонент комп'ютерних пристроїв їх складності можна оцінити з використанням апаратної, часової та місткістної характеристик. Разом з тим при програмній реалізації – основними характеристиками буде час виконання операції та місткістна складність, оскільки комп'ютерна платформа для виконання цієї операції вважається заданою.

Апаратну та часову характеристики складності комп'ютерних компонент обернення даних у МП у полях Галуа з характеристикою 2, які побудовані на базі узагальненої структури ОБ (рис. 3.11), можна оцінити згідно з такими виразами, відповідно [108]:

$$A_{AI}(N,n) = (n+3)A_I(N) + n^2 A_M(N) + (n^2+n)A_{\oplus}(N), \tag{3.29}$$

$$t_{AI}(N,n) = 3t_I(N) + t_M(N) + 2nt_{\oplus}, \tag{3.30}$$

де $A_I(N)$, $t_I(N)$ - відповідно апаратна і часова складності блока обернення елемента у $GF(2^N)$, $A_M(N)$ і $t_M(N)$ - відповідно апаратна і часова складності помножувача у $GF(2^N)$, $A_{\oplus}(N)$ і $t_{\oplus}$ - відповідно апаратна і часова складності блока додавання даних у $GF(2^N)$.

Якщо комп'ютерні компоненти будувати на базі структури ОБ з використанням табличних перетворень (рис. 3.12), то характеристики апаратної, часової, місткістної складності таких компонент можна оцінити згідно з такими відповідними виразами [108]:

$$A_{AI}(N,n) = (2n^2+2n+5)A_{\oplus}(N) + n^2 A_{+}(N) + (n^2+2n+7)\cdot O(N\cdot 2^N), \tag{3.31}$$

$$t_{AI}(N,n) = (2n+6)t_{\oplus} + 9t_R(N) + t_+(N), \tag{3.32}$$

$$V_{AI}(N,n) = (n^2 + 2n + 7)N \cdot 2^N, \tag{3.33}$$

де $A_+(N)$, $t_+(N)$ - відповідно апаратна і часова складність ОБ додавання даних за модулем 2^N.

Основною складовою апаратної складності ОБ на базі табличних перетворень є об'єм обладнання, необхідний для побудови таблиць логарифмів та антилогарифмів (рис. 3.24).

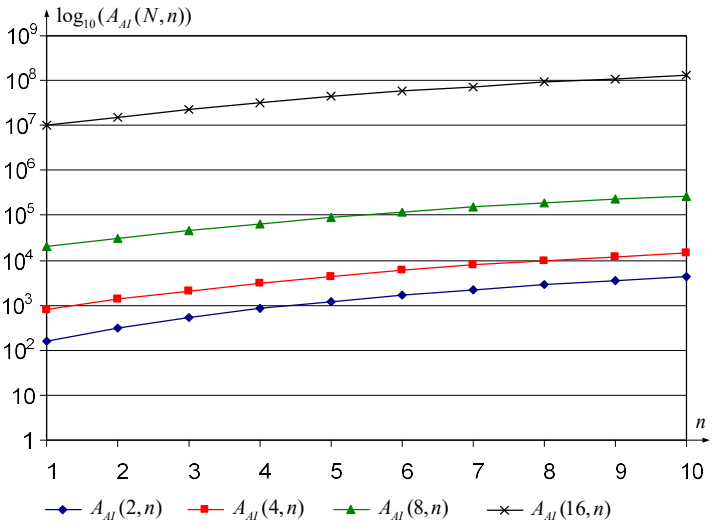


Рис. 3.24. Графік залежності апаратної складності ОБ обернення даних у МП від розрядності даних та кількості масок

При фіксованому розмірі вхідних даних апаратна складність ОБ зростає у квадратичній залежності від кількості масок, які використано для МП вхідних даних.

Часова складність ОБ росте лінійно із збільшенням кількості масок, які використано для представлення даних (рис. 3.25).

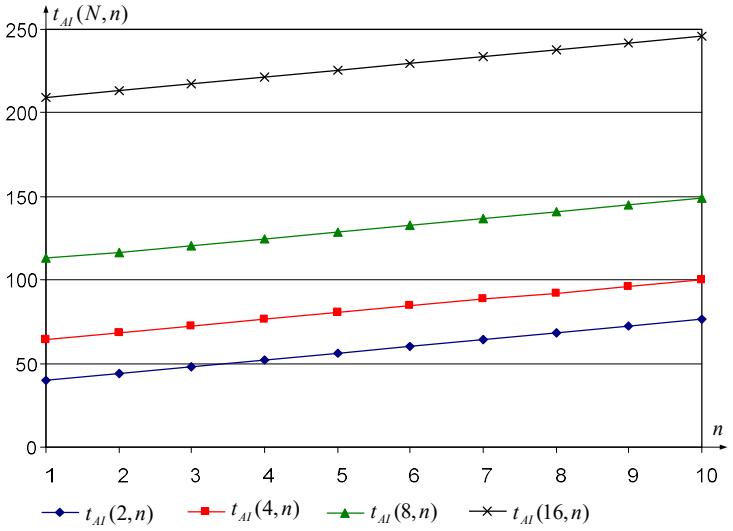


Рис. 3.25. Графік залежності часової складності ОБ обернення даних у МП від розрядності даних та кількості масок

Для заданого числа масок та змінної розрядності вхідних чисел часова складність ОБ визначається відповідними змінами часової складності суматора та запам'ятовуючого пристрою.

Розмір вибірки випадкових чисел, необхідний для роботи комп'ютерного компонента на базі запропонованої структури складає

$$R(N,n) = 2N\,, \tag{3.34}$$

і не залежить від кількості масок, які використано для подання вхідних даних у МП.

За умови програмної реалізації обчислень, послідовність яких повторює послідовність обчислень у структурі ОБ, поданого на рис. 3.12, час виконання обчислень та необхідний об'єм пам'яті для зберігання таблиць можна обчислити згідно з такими виразами, відповідно [108]:

$$t_{PI}(N,n) = (2n^2 + 2n + 5)t_{\oplus}(N) + n^2 t_{+}(N) + (n^2 + 2n + 7)t_R(N)\,, \tag{3.35}$$

$$V(N,n) = 3 \cdot 2^N\,. \tag{3.36}$$

При реалізації ОБ на базі універсального програмованого процесора час обробки даних буде залежати від співвідношення розрядності слова процесора і

розрядності даних у МП, кількості масок. Якщо розрядність даних у МП не перевищує розрядність слова процесора, то час обробки даних залежить від квадрату кількості масок (рис. 3.26).

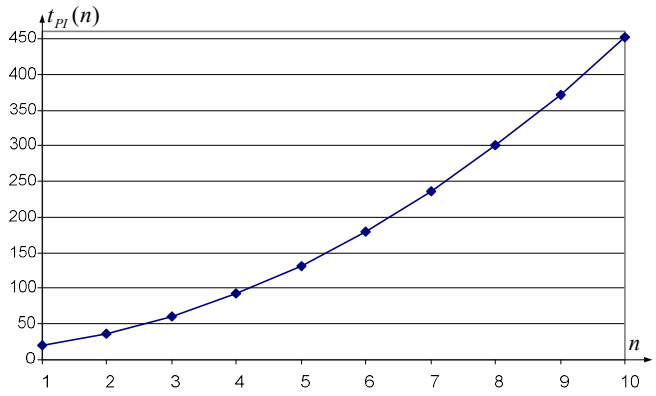


Рис. 3.26. Графік залежності часу виконання інвертування даних у МП від кількості масок при реалізації ОБ на базі універсального програмованого процесора

Результат аналізу залежності апаратної складності, поданої виразом (3.31), показує, що із збільшенням розрядності даних, які обробляються, ОБ на основі виразу (3.31) буде мати обмежене застосування. Причиною такого обмеження є експоненційне зростання апаратної складності запам'ятовуючих пристроїв. Тому областю доцільного використання такого пристрою є обробка даних розрядністю 2, 4, 8 бітів.

Якщо необхідно обробляти дані більшої розрядності, то доцільно будувати блоки множення у $GF(2^N)$ на базі структур апаратних помножувачів та інверторів, наприклад [116, 117]. Однак, при цьому необхідно враховувати, що програмна реалізація помножувачів і інверторів для даних великої розрядності володіє значною часовою складністю.

Розроблені структури ОБ інвертування даних у МП із однією маскою володіють характеристиками складності, наведеними у табл. 3.1. Для порівняння характеристик складності розроблених та відомих структур ОБ інвертування даних у МП із одною маскою, приймемо, що $t_I = 6t_M$, $A_I = 4A_M$, $t_I, t_M \gg t_\oplus$,

$A_M, A_I \gg A_\oplus$. Тоді отримаємо, що апаратна складність розробленої структури ОБ для даних у МП із однією маскою є 42% більшою від аратної складності ОБ на основі структури MULT-MASK та у 2.4 рази більшою від структури ADAPT-MULT-MASK. Відповідно часова складність розробленої структури є вищою, відповідно на 26% та у 2.4 рази для цих же структур.

Разом з тим, розроблена структура ОБ інвертування даних є вільна від недоліку відомих структур – вона забезпечує повне маскування даних. Додатково, за необхідності, можна побудувати ОБ, які будуть обробляти дані у МП із довільною кількістю масок, тим самим підвищуючи захищеність ОБ до атак на основі АСП вищих порядків. При обробці даних невеликої розрядності (4 – 8 біт) доцільно використати структури ОБ на основі таблиць логарифмів та антилогарифмів. Ці таблиці також можна використати для виконання операцій множення. Додатково, на відміну від ОБ інвертування даних у МП у $GF(2^N)$, запропонованого у [80], запропонований ОБ вимагає на 30% меншого розміру вибірки випадкових чисел за однакових розмірів даних, які обробляються. Таке зменшення розміру вибірки зумовлене використанням однакових таблиць для виконання операцій логарифмування та антилогарифмування.

3.5.5 Дослідження характеристик структур ОБ перетворення МП даних

Характеристики ОБ перетворення МП даних, в основному, повторюють відповідні характеристики ОБ додавання даних за модулем 2^N, який є їх основним елементом. Часову складність оцінимо як довжину критичного шляху N -розрядного ОБ. ОБ перетворення МП даних на базі суматора для перетворення арифметичної маски у логічну (рис. 3.15а) будуть володіти апаратної та часовою складністю, які описуються такими виразами, відповідно:

$$A_{CALS}(N,n) = A_{MS}(N,n) + 3NA_\oplus + 9NA_\wedge, \tag{3.37}$$

$$t_{CALS}(N,n) = t_{MS}(N,n) + 2t_\oplus + 3Nt_\oplus, \tag{3.38}$$

де $t_{MS}(N,n)$, $A_{MS}(N,n)$ - апаратна та часова складності суматора даних у МП, що визначені виразами (3.20) і (3.22) відповідно.

ОБ перетворення МП даних на базі суматора для перетворення логічної маски у арифметичну (рис. 3.15б) будуть володіти апаратною та часовою складністю, які описуються такими виразами, відповідно:

$$A_{CLAS}(N,n) = A_{MS}(N,n) + 2NA_{\oplus}, \tag{3.39}$$

$$t_{CALS}(N,n) = t_{MS}(N,n) + 2t_{\oplus}. \tag{3.40}$$

Результати оцінки складності структур ОБ перетворення МП даних на основі маскованого суматора, виражені у $A_{\wedge}$ і $t_{\wedge}$ згідно з виразами (3.37 – 3.40) наведено у табл. 3.1.

Якщо структури ОБ перетворення МП даних побудовані на базі ОБ табличних перетворень даних у МП з різними масками (рис. 3.17, рис. 3.18), то їх характеристики складності повторюють відповідні характеристики складності ОБ перетворення МП даних, поданих виразами у табл. 3.3 (третій і четвертий рядки).

Для порівняння характеристик складності розроблених структур ОБ перетворення МП даних на основі методів, розроблених у другому розділі, оберемо перетворення МП даних із ЛМ у АМ. Для обробки одного розряду даних апаратна складність розробленої структури ОБ на базі суматора даних у МП складе $96A_{\wedge}$, а часова - $19.8t_{\wedge}$, що у порівнянні структурою ОБ на базі методу XOR-MUX-MR вимагає у 8.2 рази більше обладнання та є у 4.9 рази повільнішою. Структура ОБ на основі табличних перетворень даних у МП володіє апаратною складністю $O(N)A_{\wedge} + O(N2^N)A_{\wedge}$ та часовою складністю $O(1)t_{\wedge} + O(N2^N)$, що, у порівнянні із структурою ОБ на основі табличних перетворень TABLE-MR вимагає на $O(N)A_{\wedge}$ більших затрат обладнання та є повільнішим на $O(1)t_{\wedge}$. Разом з тим, розроблені структури ОБ, забезпечують маскування вихідних даних новою випадковою маскою, що, на відміну від згаданих структури ОБ, зумовлює підвищення стійкості таких ОБ до атак на

основі АСП першого порядку і вимагає проведення атак АСП другого чи вищих порядків. Додатково, структури ОБ перетворення МП на основі маскованого суматора, на відміну від ОБ на основі табличних перетворень, легко адаптуються до довільної розрядності даних, дозволяє побудувати структуру ОБ, стійку до заданого порядку атаки на основі АСП.

3.6 Висновки до третього розділу

1. На основі розроблених у другому розділі методів виконання логічних операцій над даними у МП, розроблено та досліджено структури ОБ для виконання операції кон'юнкції та диз'юнкції, що дозволяє обробляти вхідні дані з ЛМ і формувати результат з ЛМ без розголошення відомостей про немасковані вхідні дані та результат. У результаті аналізу характеристик складності структур ОБ встановлено, що при послідовному виконанні операцій, розроблені структури забезпечують виграш у часі від 20% для операції диз'юнкції до 33% для операції кон'юнкції даних у МП із однією маскою. Аналіз характеристики апаратної складності розроблених структур ОБ вказує на збільшення затрат на їх реалізацію для обробки даних у МП із однією маскою. Разом з тим, спираючись на розроблені у другому розділі методи, на відміну від існуючих, можна побудувати структури ОБ для обробки даних у МП із заданою кількістю масок та маскуванням даних новою маскою, що дозволяє створювати структури ОБ, стійкі до атак на основі АСП заданого порядку.

2. На основі розроблених у другому розділі методів виконання операцій кон'юнкції та диз'юнкції над даними у МП, розроблено та досліджено структуру ОБ для виконання операції додавання за модулем 2^N на основі маскованих логічних елементів, що дозволяє обробляти вхідні дані з ЛМ і формує результат з ЛМ без розголошення відомостей про немасковані вхідні дані та результат. У порівнянні з потенційними аналогами на основі табличних перетворень, розроблена структура ОБ відрізняється простотою адаптування до обробки даних довільної розрядності за рахунок уникнення обмеження на розрядність вхідних даних для пам'яті. Додатково, можна побудувати структури ОБ для

додавання даних у МП за модулем 2^N із заданою кількістю масок та маскуванням даних новою маскою, що дозволяє створювати структури ОБ, стійкі до атак на основі АСП заданого порядку.

3. На підставі запропонованих у другому розділі методів виконання табличних перетворень над даними у МП, розроблено та досліджено структури ОБ для виконання цих операцій, що, на відміну від існуючих, уможливлюють використання довільних адитивних операцій маскування як вхідних, так і вихідних даних за рахунок введення випадкових проміжних масок. У порівнянні з відомими структурами ОБ, розроблені пристрої підтримують не лише ЛМ, а АМ як для вхідних так і вихідних даних, що дозволяє зменшити час виконання послідовності операцій табличного перетворення та перетворення типу МП даних за рахунок виконання одноразової підготовчої процедури обробки даних у МП.

4. На базі розробленого у другому розділі методу виконання інвертування даних у МП, розроблено та досліджено структуру ОБ для інвертування даних у МП з ЛМ, які, на відміну від існуючих, забезпечують повне маскування проміжних та вихідних даних. Розроблену структуру можна адаптувати для обробки даних із довільною кількістю масок, що дозволяє створити ОБ, із заданим рівнем захисту до атак основі АСП вищих порядків. При обробці даних невеликої розрядності (4 – 8 біт) доцільно використати структури ОБ на основі таблиць логарифмів та антилогарифмів із їх повторним використанням для виконання операцій множення. Додатково, запропонований ОБ вимагає на 30% меншого розміру вибірки випадкових чисел за однакових розмірів даних, які обробляються. Таке зменшення розміру вибірки зумовлене використанням однакових таблиць для виконання операцій логарифмування та антилогарифмування.

5. Ґрунтуючись на розробленому у другому розділі методі виконання перетворень МП даних та запропонованій у третьому розділі структурі маскованого суматора, розроблено та досліджено структури ОБ для

перетворення МП даних з використанням ЛМ у МП із АМ і навпаки, що, на відміну від структур, які можна побудувати на основі табличних перетворень, дозволило уникнути використання пам’яті для створення таких пристроїв та зняти обмеження на розмір даних за рахунок легкого адаптування структури суматора даних у МП до заданої розрядності даних.

6. Розроблені у цьому розділі структури ОБ на базі розроблених у другому розділі методів обробки даних у МП доцільно використати при реалізації комп’ютерних компонент із підвищеним рівнем захисту від атак на основі АСП. Для експериментальної перевірки розроблених рішень побудуємо процесори симетричного блокового шифрування згідно з алгоритмами криптографічних перетворень mCrypton та ГОСТ 28147-80, базові операції яких входять до переліку розроблених ОБ. Для цього адаптуємо ці алгоритми криптографічних перетворень до даних у МП за допомогою заміни арифметичних та логічних операцій криптографічних перетворень на еквівалентні операції, розроблені у другому розділі, пристосовані до обробки даних у МП.

РОЗДІЛ 4
РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ КОМП'ЮТЕРНИХ КОМПОНЕНТІВ ОБРОБКИ ДАНИХ У МАСКОВАНОМУ ПРЕДСТАВЛЕННІ

4.1 Архітектура комп'ютерних компонент обробки даних у маскованому представленні

Для використання розроблених у третьому розділі структур ОБ для даних у МП необхідно врахувати такі особливості архітектури комп'ютерних компонент обробки даних у МП є [5, 88, 109]:

- включення до складу компонентів генератора випадкових чисел (ГВЧ) для формування початкових масок та оновлення масок в процесі обчислень;
- розміщення деяких блоків у спеціальних областях компонентів, які є недоступними для маніпулювання (зона А) і недоступними для читання (зона Б);
- виконання алгоритмів криптографічних перетворень за допомогою процесора маскованих обчислень (ПМО) (рис. 4.1).

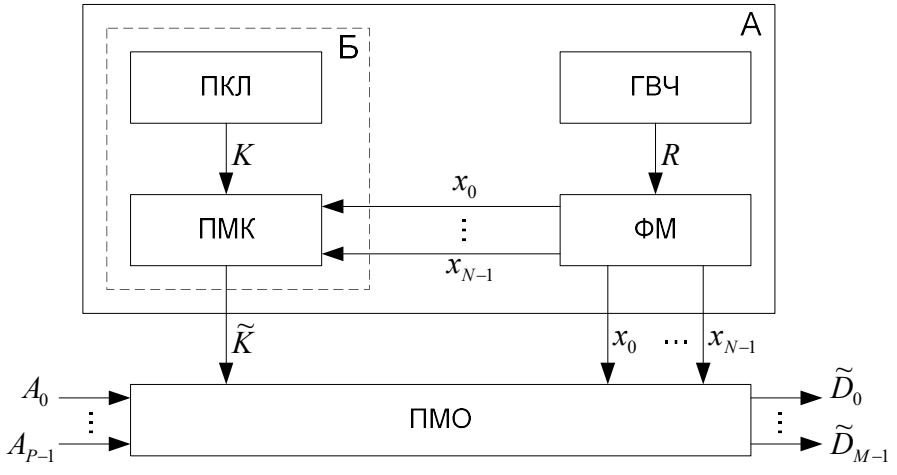


Рис. 4.1. Структура комп'ютерного компонента обробки даних у МП

До складу комп'ютерного компонента додатково входять формувач масок (ФМ), пам'ять ключів (ПКЛ), пристрій маскування ключа (ПМК) та процесор

маскованих обчислень (ПМО) (рис. 4.1). З міркувань безпеки виконання обчислень у МП блоки ГВЧ та ФМ розміщують у такій зоні компонента, яка недоступна для модифікування структури цих блоків (зона А), а блоки ПКЛ та ПМО додатково розміщують у зоні, недоступній для читання ззовні (зона Б).

Для організації маскованих обчислень ГВЧ генерує потік випадкових чисел R, з якого ФМ формує набір масок $x_0 ... x_{N-1}$ заданої розрядності, які використовуються у ПМК та ПМО. Ключ шифрування K зчитується з ПКЛ та подається на вхід ПМК. З виходу ПМК отримують ключ $\tilde{K}$ у МП, який використовується у ПМО.

ПМО виконує заданий алгоритм криптографічного перетворення над даними $A_0,..., A_{P-1}$, ключем $\tilde{K}$, масками $x_0 ... x_{N-1}$. При цьому усі операції над даними і ключем здійснюються у МП. З виходу ПМО отримують результат обчислень $\tilde{D}_0,...,\tilde{D}_{M-1}$ у МП. В свою чергу ПМО містить блоки пристрою оновлення масок (ПОМ), та ОБ, що складається з пам'яті масок і даних у МП (ПМ) та пристрою обробки даних у МП і ключів (ПОДК) (рис. 4.2).

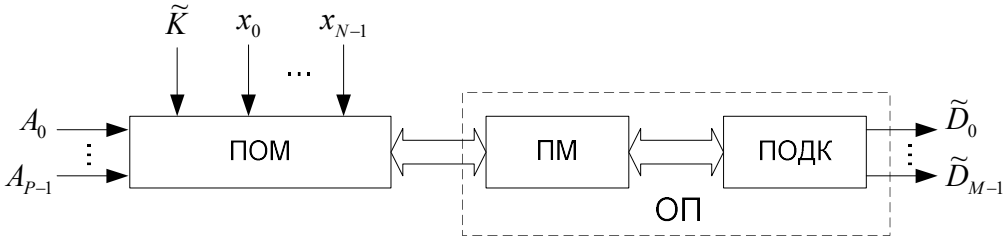


Рис. 4.2. Структура процесора маскованих обчислень

ПОМ забезпечує оновлення масок вхідних даних $A_0,..., A_{P-1}$ та ключа $\tilde{K}$ за допомогою отриманих масок $x_0 ... x_{N-1}$. Алгоритм оновлення і кількість використаних для цього масок залежать від особливостей алгоритму криптографічного перетворення та заданої наперед кількості масок для МП проміжних даних. Для оновлення масок даних у МП використовують такі базові стратегії [47]:

- фіксування масок даних і ключа перед початком виконання алгоритму криптографічного перетворення та оновлення цих масок лише перед наступним процесом обчислень;
- оновлення деякої частини масок проміжних результатів у процесі виконання алгоритму криптографічного перетворення;
- оновлення усіх масок проміжних результатів у процесі виконання алгоритму криптографічного перетворення.

Вибір стратегії організації оновлення МП даних залежить від рівня захисту, якого необхідно досягнути при реалізації комп'ютерного компонента та заданих обмежень на продуктивність ГВЧ. Із посиленням жорсткості вимог до рівня захисту компонента кількість масок для МП даних та кількість проміжних даних, які оновлюють збільшується. При цьому порядок атаки но основі АСП, від якої необхідно захистити пристрій, задає мінімальну кількість масок, яку необхідно використати для МП як вхідних, так і проміжних даних.

ПМ використовується для організації зберігання проміжних даних, які утворюються при обробці вхідних даних у ПОДК. У загальному випадку ця пам'ять є багато портовою та може входити до складу ПОДК. ПОДК забезпечує обробку даних і ключа у МП згідно з алгоритмом криптографічного перетворення. Структура ПОДК залежить від заданого алгоритму (чи набору алгоритмів). Наприклад, для алгоритмів симетричного блокового шифрування, до складу ПОДК будуть входити два взаємопов'язаних тракти – тракт обробки даних та тракт обробки ключів.

Результат обробки даних ПОДК подається на вихід комп'ютерного компонента у МП $\widetilde{D}_0,...,\widetilde{D}_{M-1}$. За необхідності можна отримати немасковане представлення результатів шляхом переходу від МП у немасковане. Такий перехід передбачає виконання операції зняття маски з результатів за допомогою їх маскування новими масками, оберненими до отриманих. Альтернативним способом переходу є модифікування маскованих результатів за допомогою отриманих масок та оберненої операції маскування.

4.2 Реалізація та експериментальне дослідження процесора mCrypton для даних у маскованому представленні

Процесор симетричного блокового шифрування mCrypton для даних у МП побудуємо з використанням розроблених у другому розділі методів обробки даних у МП та ОБ, розроблених у третьому розділі. Алгоритм симетричного блокового шифрування mCrypton використовується у комп'ютерних пристроях з жорсткими обмеженнями на апаратну і місткістну складність, СП [118]. Аналіз опису алгоритму mCrypton дозволив визначити його базові операції для їх заміни на еквівалентні операції для даних у МП, до переліку яких входять:

- заміна за таблицею базових елементів даних (нелінійне перетворення);
- бітові перестановки стовбців;
- транспозиції масиву A;
- додавання ключа.

Для побудови алгоритму обробки даних у МП замінимо базові операції алгоритму mCrypton на їх масковані еквіваленти. Також визначимо три додаткові масиви аналогічні до A, зокрема X, Y, Z, для вхідної, проміжної та вихідної масок, відповідно. Позначимо $\widetilde{A} = A \oplus X$ - МП A з використанням маски X. Застосовуючи ці позначення, побудуємо базові масковані операції для mCrypton [12].

Масковане нелінійне перетворення $\widetilde{\gamma}$ визначене наступним чином. Для 4-елементних слів $\widetilde{a} = (\widetilde{a}_0\widetilde{a}_1\widetilde{a}_2\widetilde{a}_3)$, $x = (x_0x_1x_2x_3)$, $y = (y_0y_1y_2y_3)$, $z = (z_0z_1z_2z_3)$ обробка маски визначається функцією $\mu_i(x, y, z) = z$. Перетворення даних у МП знаходиться як

$$\widetilde{\gamma}_i(\widetilde{a}, x, y, z) = (\widetilde{S}_i(\widetilde{a}_0, x_0, y_0, z_0), \widetilde{S}_{i+1}(\widetilde{a}_1, x_1, y_1, z_1), \widetilde{S}_{i+2}(\widetilde{a}_2, x_2, y_2, z_2), \widetilde{S}_{i+3}(\widetilde{a}_3, x_3, y_3, z_3)), \quad (4.1)$$

де індекси обчислюються за модулем 4 і $\widetilde{S}(\widetilde{a}, x, y, z) = S(a) \oplus z$.

Останні обчислення проводяться з використанням X, Y, Z і спеціальної процедури для уникнення витоку інформації про аргумент a. Тому, перетворення $\tilde{\gamma}$, μ і $\tilde{\gamma}^{-1}$ можна визначити для масивів $\tilde{A}$ і X, Y, Z так:

$$\tilde{\gamma}(\tilde{A}, X, Y, Z) = (\tilde{\gamma}_0(\tilde{A}_c[0], X_c[0], Y_c[0], Z_c[0]), ..., \tilde{\gamma}_3(\tilde{A}_c[3], X_c[3], Y_c[3], Z_c[3])), \quad (4.2)$$

$$\mu(X, Y, Z) = Z, \quad (4.3)$$

$$\begin{aligned} \tilde{\gamma}^{-1}(\tilde{A}, X, Y, Z) = (&\tilde{\gamma}_2(\tilde{A}_c[0], X_c[0], Y_c[0], Z_c[0]), \\ &\tilde{\gamma}_3(\tilde{A}_c[1], X_c[1], Y_c[1], Z_c[1]), \\ &\tilde{\gamma}_0(\tilde{A}_c[2], X_c[2], Y_c[2], Z_c[2]), \\ &\tilde{\gamma}_1(\tilde{A}_c[3], X_c[3], Y_c[3], Z_c[3])), \end{aligned} \quad (4.4)$$

$$\mu^{-1}(X, Y, Z) = Z. \quad (4.5)$$

Маскована бітова перестановка $\tilde{\pi}$ використовує перестановку стовпця $\tilde{\pi}_i$ для кожного стовпця $0 \le i \le 3$ масиву $\tilde{A}$ і двох масивів X, Z:

$$\tilde{\pi}(\tilde{A}, X, Z) = (\tilde{\pi}_0(\tilde{A}_c[0], X_c[0], Z_c[0]), ..., \tilde{\pi}_3(\tilde{A}_c[3], X_c[3], Z_c[3])), \quad (4.6)$$

де $\tilde{\pi}_i$ задане для $\tilde{a} = (\tilde{a}_0 \tilde{a}_1 \tilde{a}_2 \tilde{a}_3)^t$, $x = (x_0 x_1 x_2 x_3)^t$ і $z = (z_0 z_1 z_2 z_3)^t$ у вигляді $\tilde{b} = \tilde{\pi}_i(\tilde{a}, x, z) \Leftrightarrow \tilde{b}_j = \oplus_{k=0}^{3} (m_{i+j+k \bmod 4} \tilde{\bullet} (a_k, x_k, z_k))$.

Також визначимо відповідне перетворення маски для усіх i:

$$\eta_i(x, z) \Leftrightarrow z_j = \oplus_{k=0}^{3} x_k. \quad (4.7)$$

Тоді $\eta(X, Z) = (\eta_0(X_c[0]), \eta_1(X_c[1]), \eta_2(X_c[2]), \eta_3(X_c[3]))$, а маскована операція AND $m_{i+j+k \bmod 4} \tilde{\bullet} (\tilde{a}_k, x_k, z_k)$ виконується згідно з таким виразом:

$$m_{i+j+k \bmod 4} \tilde{\bullet} (\tilde{a}_k, x_k, z_k) = ((m_{i+j+k \bmod 4} \bullet x) \oplus z) \oplus (\tilde{a} \bullet m_{i+j+k \bmod 4}). \quad (4.8)$$

Маскована транспозиція $\tilde{\tau}$ є аналогічною до транспозиції τ і переміщає базовий елемент з (i, j)-ої позиції у (j, i)-ту позицію:

$$\tilde{B} = \tau(\tilde{A}) \Leftrightarrow \tilde{b}_{ij} = \tilde{a}_{ji}. \quad (4.9)$$

Зауважимо, що $\tau = \tilde{\tau} = \tilde{\tau}^{-1}$. Відповідна обробка маски визначена як перетворення τ над масивом маски: $Z = \tau(X) \Leftrightarrow z_{ij} = x_{ji}$.

Для маскованого раундового ключа $\tilde{K} = (\tilde{K}[0], \tilde{K}[1], \tilde{K}[2], \tilde{K}[3])$ і маски раундового ключа $Q = (Q[1], Q[2], Q[3], Q[4])$, операція маскованого додавання ключа $\tilde{\sigma}$ визначена у вигляді

$$\tilde{B} = \tilde{\sigma}_{\tilde{K}}(\tilde{A}) \Leftrightarrow \tilde{B}_r[i] = \tilde{A}_r[i] \oplus \tilde{K}[i], \tag{4.10}$$

де $0 \le i \le 3$, а відповідне перетворення маски знаходиться як

$$Z = \sigma_Q(X) \Leftrightarrow Z_r[i] = X_r[i] \oplus Q_r[i]. \tag{4.11}$$

Масковані раундові перетворення для зашифрування $\tilde{\rho}$ і розшифрування $\tilde{\rho}^{-1}$ визначені для маскованого ключа $\tilde{K}$ наступним чином:

$$\tilde{\rho}_{\tilde{K}} = \tilde{\sigma}_{\tilde{K}} \circ \tilde{\tau} \circ \tilde{\pi} \circ \tilde{\gamma}\,, \tag{4.12}$$

$$\tilde{\rho}^{-1}{}_{\tilde{K}} = \tilde{\gamma}^{-1} \circ \tilde{\pi} \circ \tilde{\tau} \circ \tilde{\sigma}_{\tilde{K}}\,. \tag{4.13}$$

Відповідні перетворення масок знаходяться з виразів:

$$\lambda = \sigma_Q \circ \tau \circ \eta \circ \mu\,, \tag{4.14}$$

$$\lambda^{-1} = \mu^{-1} \circ \eta \circ \tau \circ \sigma_Q\,. \tag{4.15}$$

Тоді, масковане перетворення зашифрування $\tilde{E}_{\tilde{K}}$ можна подати у вигляді

$$\tilde{E}_{\tilde{K}} = \tilde{\phi} \circ \tilde{\rho}_{\tilde{K}_e^{12}} \circ \tilde{\rho}_{\tilde{K}_e^{11}} \circ \cdots \circ \tilde{\rho}_{\tilde{K}_e^{2}} \circ \tilde{\rho}_{\tilde{K}_e^{1}} \circ \tilde{\sigma}_{\tilde{K}_e^{0}}\,, \tag{4.16}$$

а масковане перетворення розшифрування –

$$\tilde{D}_{\tilde{K}} = \tilde{\phi} \circ \tilde{\rho}'_{\tilde{K}_e^{12}} \circ \tilde{\rho}'_{\tilde{K}_e^{11}} \circ \cdots \circ \tilde{\rho}'_{\tilde{K}_e^{2}} \circ \tilde{\rho}'_{\tilde{K}_e^{1}} \circ \tilde{\sigma}_{\tilde{K}_e^{0}}\,, \tag{4.17}$$

де $\widetilde{\phi} = \tau \circ \widetilde{\pi} \circ \tau$, $\widetilde{\rho}'$ є таким самим як $\widetilde{\rho}$, однак із іншим впорядкуванням таблиць замін S, $\widetilde{K}_d^{r-i} = \widetilde{\phi}(\widetilde{K}_e^i)$ для $0 \le i \le 12$.

Немасковані дані отримуються шляхом виконання операції XOR над результуючими даними у МП та маскою.

Використаємо аналогічний підхід для побудови обчислювального процесу розпису маскованого ключа для різних розмірів ключа [12]. Розпис маскованого ключа дозволяє обчислити набір маскованих раундових ключів $\widetilde{K}$ і відповідних масок Q.

Розпис маскованого ключа для 64-бітових ключів.

Масковані раундові ключі для зашифрування обчислюються наступним чином. Спочатку регістр ключа $\widetilde{U}$ ініціалізується ключем $\widetilde{K}$, а регістр маски RQ – маскою Q. В подальшому для раунду $r = 0,1,...,12$ обчислюються масковані раундові ключі для зашифрування та тимчасові маски $Y = \{Y[0], Y[1], Y[2], Y[3]\}$ і $Z = \{Z[0], Z[1], Z[2], Z[3]\}$, де $Z[i] = (z_0^i, z_1^i, z_2^i, z_3^i)$:

$$\widetilde{T} \leftarrow \widetilde{S}(\widetilde{U}[0], RQ[0], Y, Z) \oplus C[r],\ \widetilde{T}_i \leftarrow \widetilde{T} \bullet M_i, \tag{4.18}$$

$$\widetilde{K}_e^r = (\widetilde{U}[1] \oplus \widetilde{T}_0, \widetilde{U}[2] \oplus \widetilde{T}_1, \widetilde{U}[3] \oplus \widetilde{T}_2, \widetilde{U}[0] \oplus \widetilde{T}_3), \tag{4.19}$$

$$\widetilde{U} \leftarrow (\widetilde{U}[1], \widetilde{U}[2], \widetilde{U}[3], \widetilde{U}[0]^{<<3}), \tag{4.20}$$

де $0 \le i \le 3$.

Відповідне оновлення RQ знайдене так:

$$Q_e^r = (RQ[1] \oplus Z[1], RQ[2] \oplus Z[2], RQ[3] \oplus Z[3], RQ[0] \oplus Z[0]), \tag{4.21}$$

$$RQ \leftarrow (RQ[1], RQ[2], RQ[3], RQ[0]^{<<3}). \tag{4.22}$$

Нижче описано знаходження маскованих раундових ключів для розшифрування. Для цього спочатку регістр ключа $\widetilde{V}$ ініціалізується елементами ключа: $\widetilde{V} \leftarrow (\widetilde{K}[0]^{<<9}, \widetilde{K}[1]^{<<9}, \widetilde{K}[2]^{<<9}, \widetilde{K}[3]^{<<9})$. Регістр маски

ініціалізується елементами маски: $RQ \leftarrow (Q[0]^{<<9}, Q[1]^{<<9}, Q[2]^{<<9}, Q[3]^{<<9})$. Потім для раунду $r = 0,1,...,12$ послідовно обчислюються масковані раундові ключі для розшифрування:

$$\widetilde{T} \leftarrow \widetilde{S}(\widetilde{V}[0], RQ[0], Y, Z) \oplus C[12-r],\ \widetilde{T}_i \leftarrow \widetilde{T} \bullet M_i, \tag{4.23}$$

$$\widetilde{K}_d^r = (\widetilde{\phi}_0(\widetilde{V}[1] \oplus \widetilde{T}_0, Z[1]), \widetilde{\phi}_1(\widetilde{V}[2] \oplus \widetilde{T}_1, Z[2]), \widetilde{\phi}_2(\widetilde{V}[3] \oplus \widetilde{T}_2, Z[3]), \widetilde{\phi}_3(\widetilde{V}[0] \oplus \widetilde{T}_3, Z[0])), \tag{4.24}$$

$$\widetilde{V} \leftarrow (\widetilde{V}[3]^{<<13}, \widetilde{V}[0], \widetilde{V}[1], \widetilde{V}[2]), \tag{4.25}$$

де $0 \le i \le 3$.

Відповідне оновлення RQ визначене наступним чином:

$$Q_d^r = (RQ[1] \oplus \bigoplus_{j=0}^{3} z_j^1, RQ[2] \oplus \bigoplus_{j=0}^{3} z_j^2, RQ[3] \oplus \bigoplus_{j=0}^{3} z_j^3, RQ[0] \oplus \bigoplus_{j=0}^{3} z_j^0), \tag{4.26}$$

$$RQ \leftarrow (RQ[3]^{<<13}, RQ[0], RQ[1], RQ[2]). \tag{4.27}$$

Розпис маскованого ключа для 96-бітових ключів.

Масковані раундові ключі для зашифрування обчислюються наступним чином. Спочатку регістр ключа $\widetilde{U}$ ініціалізується ключем $\widetilde{K}$. Регістр маски RQ ініціалізується маскою Q. Потім для раунду $r = 0,1,...,12$ обчислюються масковані раундові ключі розшифрування та тимчасові маски $Y = \{Y[0],..., Y[5]\}$, і $Z = \{Z[0],..., Z[5]\}$, де $Z[i] = (z_0^i, z_1^i, z_2^i, z_3^i)$:

$$\widetilde{T} \leftarrow \widetilde{S}(\widetilde{U}[0], RQ[0], Y, Z) \oplus C[r],\ \widetilde{T}_i \leftarrow \widetilde{T} \bullet M_i, \tag{4.28}$$

$$\widetilde{K}_e^r = (\widetilde{U}[1] \oplus \widetilde{T}_0, \widetilde{U}[2] \oplus \widetilde{T}_1, \widetilde{U}[3] \oplus \widetilde{T}_2, \widetilde{U}[4] \oplus \widetilde{T}_3), \tag{4.29}$$

$$\widetilde{U} \leftarrow (\widetilde{U}[5], \widetilde{U}[0]^{<<3}, \widetilde{U}[1], \widetilde{U}[2], \widetilde{U}[3]^{<<8}, \widetilde{U}[4]), \tag{4.30}$$

де $0 \le i \le 3$.

Відповідне оновлення RQ визначене знайдене так:

$$Q_e^r = (RQ[1] \oplus Z[1], RQ[2] \oplus Z[2], RQ[3] \oplus Z[3], RQ[4] \oplus Z[4]), \quad (4.31)$$

$$RQ \leftarrow (RQ[5], RQ[0]^{<<3}, RQ[1], RQ[2], RQ[3]^{<<8}, RQ[4]). \quad (4.32)$$

Масковані раундові ключі для розшифрування обчислюються згідно з нижче викладеним. Спочатку регістр ключа $\widetilde{V}$ ініціалізується елементами ключа: $\widetilde{V} \leftarrow (\widetilde{K}[0]^{<<6}, \widetilde{K}[1]^{<<6}, \widetilde{K}[2]^{<<6}, \widetilde{K}[3]^{<<6}, \widetilde{K}[4]^{<<6}, \widetilde{K}[5]^{<<6})$. Відповідний а регістр маски ініціалізується елементами маски виду: $RQ \leftarrow (Q[0]^{<<6}, Q[1]^{<<6}, Q[2]^{<<6}, Q[3]^{<<6}, Q[4]^{<<6}, Q[5]^{<<6})$. Далі для раунду $r = 0,1,...,12$ послідовно обчислюються масковані раундові ключі:

$$\widetilde{T} \leftarrow \widetilde{S}(\widetilde{V}[0], RQ[0], Y, Z) \oplus C[12-r],\ \widetilde{T}_i \leftarrow \widetilde{T} \bullet M_i, \quad (4.33)$$

$$\widetilde{K}_d^r = (\widetilde{\phi}_0(\widetilde{V}[1] \oplus \widetilde{T}_0, Z[1]), \widetilde{\phi}_1(\widetilde{V}[2] \oplus \widetilde{T}_1, Z[2]), \widetilde{\phi}_2(\widetilde{V}[3] \oplus \widetilde{T}_2, Z[3]), \widetilde{\phi}_3(\widetilde{V}[4] \oplus \widetilde{T}_3, Z[4])), \quad (4.34)$$

$$\widetilde{V} \leftarrow (\widetilde{V}[1]^{<<13}, \widetilde{V}[2], \widetilde{V}[3], \widetilde{V}[4]^{<<8}, \widetilde{V}[5], \widetilde{V}[0]). \quad (4.35)$$

де $0 \le i \le 3$.

Відповідне оновлення RQ визначене із співвідношень:

$$Q_d^r = (RQ[1] \oplus \bigoplus_{j=0}^{3} z_j^1, RQ[2] \oplus \bigoplus_{j=0}^{3} z_j^2, RQ[3] \oplus \bigoplus_{j=0}^{3} z_j^3, RQ[4] \oplus \bigoplus_{j=0}^{3} z_j^4), \quad (4.36)$$

$$RQ \leftarrow (RQ[1]^{<<13}, RQ[2], RQ[3], RQ[4]^{<<8}, RQ[5], RQ[0]). \quad (4.37)$$

Розпис маскованого ключа для 128-бітових ключів.

Масковані раундові ключі для зашифрування обчислюються відповідно до нижче наведеного. Спочатку регістр ключа $\widetilde{U}$ ініціалізуються ключем $\widetilde{K}$, а регістр маски RQ - маскою Q . Потім для раунду $r = 0,1,...,12$ обчислюються масковані раундові ключі для зашифрування та тимчасові маски $Y = \{Y[0],..., Y[7]\}$ і $Z = \{Z[0],..., Z[7]\}$, де $Z[i] = (z_0^i, z_1^i, z_2^i, z_3^i)$:

$$\widetilde{T} \leftarrow \widetilde{S}(\widetilde{U}[0], RQ[0], Y, Z) \oplus C[r],\ \widetilde{T}_i \leftarrow \widetilde{T} \bullet M_i, \quad (4.38)$$

$$\tilde{K}_e^r = (\tilde{U}[1] \oplus \tilde{T}_0, \tilde{U}[2] \oplus \tilde{T}_1, \tilde{U}[3] \oplus \tilde{T}_2, \tilde{U}[4] \oplus \tilde{T}_3), \quad (4.39)$$

$$\tilde{U} \leftarrow (\tilde{U}[5], \tilde{U}[6], \tilde{U}[7], \tilde{U}[0]^{<<3}, \tilde{U}[1], \tilde{U}[2], \tilde{U}[3], \tilde{U}[4])^{<<8}), \quad (4.40)$$

де $0 \le i \le 3$.

Відповідне оновлення RKM визначене знайдено з виразів:

$$Q_e^r = (RQ[1] \oplus Z[1], RQ[2] \oplus Z[2], RQ[3] \oplus Z[3], RQ[4] \oplus Z[4]), \quad (4.41)$$

$$RQ \leftarrow (RQ[5], RQ[6], RQ[7], RQ[0]^{<<3}, RQ[1], RQ[2], RQ[3], RQ[4])^{<<8}). \quad (4.42)$$

Масковані раундові ключі для розшифрування обчислюються згідно з нижче поданим. Спочатку регістр ключа $\tilde{V}$ ініціалізується елементами ключа: $\tilde{V} \leftarrow (\tilde{K}[4]^{<<3}, \tilde{K}[5]^{<<14}, \tilde{K}[6]^{<<3}, \tilde{K}[7]^{<<14}, \tilde{K}[0]^{<<14}, \tilde{K}[1]^{<<3}, \tilde{K}[2]^{<<14}, \tilde{K}[3]^{<<3})$. Регістр маски ініціалізується відповідними елементами маски $RQ \leftarrow (Q[4]^{<<3}, Q[5]^{<<14}, Q[6]^{<<3}, Q[7]^{<<14}, Q[0]^{<<14}, Q[1]^{<<3}, Q[2]^{<<14}, Q[3]^{<<3})$. Потім для раунду $r = 0,1,...,12$ послідовно обчислюються масковані раундові ключі:

$$\tilde{T} \leftarrow \tilde{S}(\tilde{V}[0], RQ[0], Y, Z) \oplus C[12 - r], \ \tilde{T}_i \leftarrow \tilde{T} \bullet M_i, \quad (4.43)$$

$$\tilde{K}_d^r = (\tilde{\phi}_0(\tilde{V}[1] \oplus \tilde{T}_0, Z[1]), \tilde{\phi}_1(\tilde{V}[2] \oplus \tilde{T}_1, Z[2]), \tilde{\phi}_2(\tilde{V}[3] \oplus \tilde{T}_2, Z[3]), \tilde{\phi}_3(\tilde{V}[4] \oplus \tilde{T}_3, Z[4])), \quad (4.44)$$

$$\tilde{V} \leftarrow (\tilde{V}[3]^{<<13}, \tilde{V}[4], \tilde{V}[5], \tilde{V}[6], \tilde{V}[7]^{<<8}, \tilde{V}[0], \tilde{V}[1], \tilde{V}[2]), \quad (4.45)$$

де $0 \le i \le 3$.

Відповідне оновлення RQ визначене наступним чином:

$$Q_d^r = (RQ[1] \oplus \bigoplus_{j=0}^{3} z_j^1, RQ[2] \oplus \bigoplus_{j=0}^{3} z_j^2, RQ[3] \oplus \bigoplus_{j=0}^{3} z_j^3, RQ[4] \oplus \bigoplus_{j=0}^{3} z_j^4), \quad (4.46)$$

$$RQ \leftarrow (RQ[3]^{<<13}, RQ[4], RQ[5], RQ[6], RQ[7]^{<<8}, RQ[0], RQ[1], RQ[2]). \quad (4.47)$$

Розроблені процедури обчислень розпису маскованого ключа та обробки маскованих даних було покладено нами в основу реалізації апаратного процесора за алгоритмом mCrypton для даних у МП.

4.2.1 Архітектура та експериментальне дослідження ядра процесора

Архітектура апаратної реалізації процесора за алгоритмом mCrypton для даних у МП ґрунтується на апаратному відображенні одного раунду алгоритму у тракт обчислення даних. Такий процесор перетворює вхідні дані у МП за 13 тактів. Для синхронного отримання маскованих раундових ключів використовується їх паралельне до даних обчислення у тракті обробки ключа (рис. 4.3) [12].

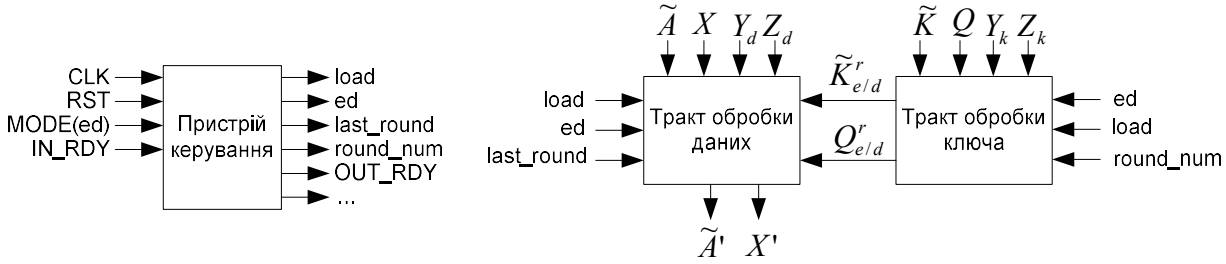


Рис. 4.3. Архітектура апаратної реалізації процесора за алгоритмом mCrypton для даних у МП

Обчислення супроводжуються використанням випадкових чисел для МП вхідних даних, ключа та проміжних масок Y_d, Z_d, Y_k і Z_k. Для їх отримання застосовується зовнішнє, по відношенню до процесора, джерело випадкових чисел.

Дані A та ключ зашифрування K подають на вхід процесора у маскованій формі: $\widetilde{A} = A \oplus X$ і $\widetilde{K} = K \oplus Q$ відповідно, причому X і Q – два незалежних випадкових числа з рівномірним законом розподілу. Тракт обробки ключа забезпечує тракт обробки даних набором раундових ключів $\widetilde{K}^r_{e/d} = K^r_{e/d} \oplus Q^r_{e/d}$ та їх відповідних масок $Q^r_{e/d}$ (рис. 4.4). Результуючі дані A' представлені в маскованій формі $\widetilde{A}' = A' \oplus X'$, де X' є результуючою маскою. Якщо необхідно отримати

вихідні немасковані дані, то їх нескладно обчислити шляхом виконання операції XOR над результуючими даними у МП та результуючою маскою: $A' = \widetilde{A}' \oplus X'$.

Тракт обробки даних процесора складається з двох каналів – каналу для обробки даних у МП $\widetilde{A}$ та каналу для обробки маски X (рис. 4.4) [12].

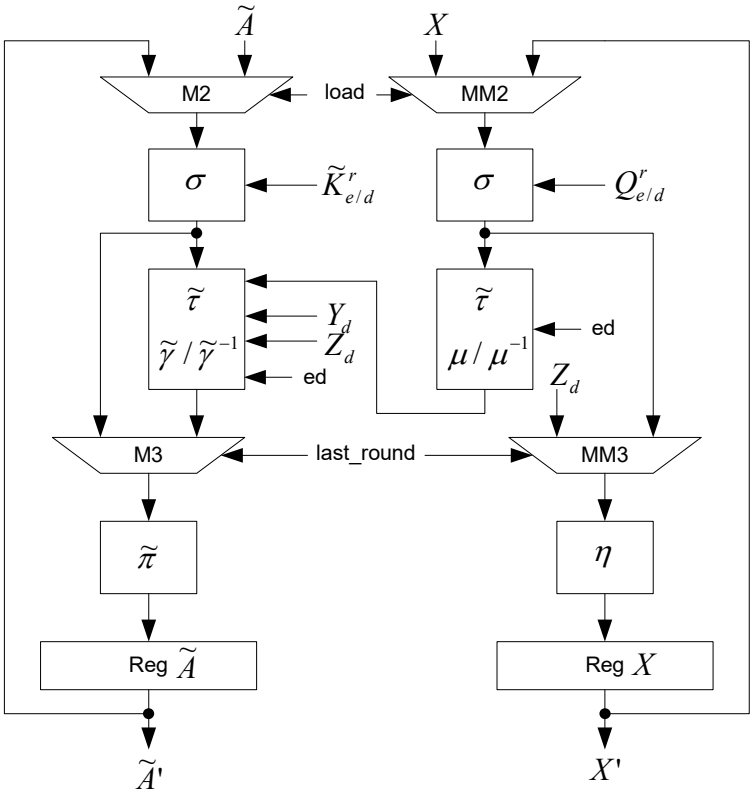


Рис. 4.4. Архітектура тракту обробки даних процесора за алгоритмом mCrypton для даних у МП

Канал для обробки даних у МП використовує додаткові маски Y_d і Z_d при виконанні маскованих операцій $\widetilde{\gamma}$ і $\widetilde{\gamma}^{-1}$. Та ж маска Z_d застосована в каналі обробки маски. Для виконання операції $\widetilde{\gamma} / \widetilde{\gamma}^{-1}$ і $\widetilde{\tau}$ було використано запропонований у [118] спосіб на основі використання одного набору із 16-ти таблиць замін та пари відповідно під'єднаних мультиплексорів. Відповідні перетворення μ / μ^{-1} і $\widetilde{\tau}$ здійснені за допомогою відповідної пари мультиплексорів, які використовують аналогічне до $\widetilde{\gamma} / \widetilde{\gamma}^{-1}$ і $\widetilde{\tau}$ комутування сигналів.

Тракт обробки ключа генерує масковані раундові ключі та відповідні до них маски з заданого ключа і випадкової маски різної довжини, зокрема 64, 96 і 128 бітів (рис. 4.5) [12]. Аналогічно до тракту обробки даних, тракт обробки ключа складається з двох каналів: канал обробки маскованого ключа та генерування маскованих раундових ключів $\widetilde{K}^r_{e/d}$, і канал для обробки маски та генерування масок раундових ключів $Q^r_{e/d}$.

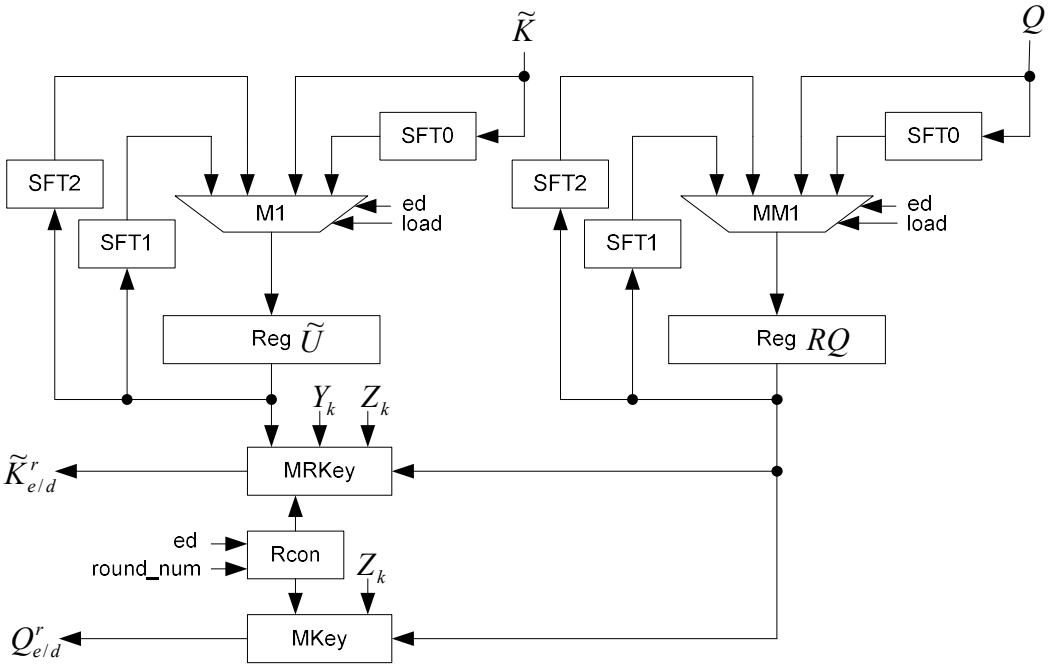


Рис. 4.5. Архітектура тракту обробки ключа процесора за алгоритмом mCrypton для даних у МП

Початкова пара з маскованого ключа та маски ключа завантажується у відповідні регістри Reg$\widetilde{U}$ і RegRQ. Тоді вміст регістрів використовується блоками MRKey і MKey (разом з раундовими константами $C[r]$ з блока Rcon) для генерування маскованого раундового ключа та маски раундового ключа, відповідно. Блок MRKey додатково використовує випадкові маски Y_k і Z_k для виконання маскованих операцій над таблицями заміни S. Ця ж маска Z_k використовується блоком MKey для обчислення відповідної маски маскованого

раундового ключа. Оновлення регістрів маскованого ключа та маски ключа відбувається через блоки циклічного зсуву SFT1 (SFT2 для розшифрування).

Використання маскованих таблиць замін може вимагати їх оновлення (переобчислення). З точки зору безпеки звичайний алгоритм mCrypton є вразливим для атакування при виконанні кількох перших та останніх раундів. Тому, пристрій керування переобчислює вміст таблиць замін для першого та останнього раундів. Інші раунди використовують однакові масковані таблиці. Однак, для уникнення можливості скидання у нуль деяких проміжних масок, раунд для оновлення вибирається випадково пристроєм керування (між другим і дванадцятим раундом). При цьому, таблиці замін оновлюються за допомогою нових масок Y_k , Z_k , Y_d , і Z_d . Такий спосіб використання випадкових масок дозволяє послабити вимоги до продуктивності генератора випадкових чисел.

Для оцінки апаратної часової складності процесора було створено його Verilog-модель [119] та прототип на основі 0,18 мкм КМОН HBIC бібліотеки для усіх розмірів ключа та режимів обробки даних [12]. Додатково було розроблено прототипи процесора для обробки даних лише у режимі зашифрування. Такий режим є доцільним для використання у пристроях з жорсткими вимогами до апаратної складності, наприклад, RFID. Результати оцінки апаратної складності (кількості логічних елементів) наведено у табл. 4.1 (1 логічний елемент еквівалентний 2-входовому логічному елементу NAND).

Критичний шлях створеного прототипу становить приблизно 10 нс. Однак, додатковий час використовується для пере обчислення 16-ти таблиць S. Переобчислення трьох таблиць S вимагає 768 тактів. Тому, найбільша продуктивність, якої можна досягти, дорівнює fsys/12 Мбіт/с, де fsys – тактова частота у МГц прототипу процесора. Більшість пристроїв з жорсткими обмеженнями на використовувані ресурси володіють відносно низькою fsys порядку від 100 КГц до десятків МГц. Отже, продуктивність обробки даних становитиме відповідно від 8.3 Кбіт/с до 0.83 Мбіт/с. Отримані результати, зокрема апаратна та часова складності, показують, що розроблений алгоритм

маскування обчислень для mCrypton доцільно застосовувати для пристроїв з жорсткими обмеженнями на використовувані ресурси, наприклад, смарт-карти, RFID, сенсори, тощо.

Апаратна складність процесора за алгоритмом mCrypton Таблиця 4.1

Режим	**Зашифрування та розшифрування**			**Тільки зашифрування**		
Розмір ключа	**64 біти**	**96 біт**	**128 біт**	**64 біти**	**96 біт**	**128 біт**
Маскований розпис ключа	2945	3566	4026	2213	2677	3093
Регістри	640	960	1280	640	960	1280
Блок Rcon	52	52	52	21	21	21
Функція ϕ	276	276	276	0	0	0
Таблиця S	1120	1120	1120	1120	1120	1120
Інші логічні блоки	857	1158	1298	432	576	672
Маскований тракт обробки даних	6590	6590	6590	4350	4350	4350
Регістри	640	640	640	640	640	640
Функція γ	4480	4480	4480	2240	2240	2240
Функція π	576	576	576	576	576	576
Додавання ключа	384	384	384	384	384	384
Інші логічні блоки	510	510	510	510	510	510
Пристрій керування	230	230	230	230	230	230
Розведення	203	284	315	136	189	210
Всього	9968	10670	11161	6929	7446	7883

Зменшення часової складності маскованих обчислень можливо досягти шляхом одноразового обчислення таблиць S для усіх можливих масок, приміром, використовуючи висвітлений в [10] метод переобчислень. Однак, останній вимагає збільшення місткістної складності.

Подальше зменшення апаратної складності можливе за рахунок використання методу об’єднання каналів обробки даних у МП і маски у трактах

обробки даних і ключа. Зазначимо, що канали обробки маски даних і даних у МП, маски ключа та маскованого ключа застосовують подібні блоки. Тоді, можна побудувати тракти процесора таким чином, щоб сумісно використовувати ці блоки. Внаслідок цього буде зменшена апаратна складність процесора, проте зменшиться продуктивність обробки даних.

4.3 Реалізація та експериментальне дослідження процесора виконання алгоритму за ГОСТ 28147-89 для даних у маскованому представленні

4.3.1 Алгоритм циклового перетворення для даних у МП

Для побудови алгоритму криптографічного перетворення за ГОСТ 28147-89 над даними у МП замінимо базові операції цього алгоритму на їх масковані еквіваленти, розглянуті у розділі 3 [13].

Операція підстановки MK – це підстановка вхідного маскованого 32-бітового блоку даних згідно з маскованою таблицею підстановки, що задана вузлами підстановки $MK_0,...,MK_7$ розміром 64 біти кожний. Відповідна таблиця підстановки подана у МП з використанням маски даних z'. Операція підстановки MK виконується над 32-розрядним вектором вхідних даних у МП $\tilde{a}$ з використанням відповідного вектора маски даних x, початкової маски таблиці z', проміжної y та вихідної масок z. Кожен з цих 32-розрядних векторів розбивається на вісім 4-розрядних векторів a_i, x_i, z'_i, y_i, z_i, $0 \le i \le 7$, а кожен набір a_i, x_i, z'_i, y_i, z_i перетворюється в 4-розрядний маскований вектор відповідним вузлом підстановки з відповідним 4-розрядним вектором маски. В подальшому 4-розрядні вихідні масковані вектори послідовно об’єднуються у 32-розрядний вектор. Аналогічно об’єднуються у 32-розрядний вектор маски маскованих векторів.

Також скористаємося маскованими еквівалентами решти операцій. Операцію додавання за модулем 2^{32} над даними у МП здійснюється за

допомогою функції *MADD*. Операція додавання даних у МП за модулем два виконується за допомогою функції *MXOR* або її спрощеним еквівалентом. Операція *MR* циклічного зсуву 32-бітового маскованого вектора даних в бік старших розрядів на 11 бітів над маскованими даними *mr* та їх маскою *m* виконується згідно з функцією зсуву початкового алгоритму, тобто

$$MR(mr_{32}, mr_{31}, mr_{30}, mr_{29}, mr_{28}, mr_{27}, mr_{26}, mr_{25}, mr_{24}, mr_{23}, mr_{22}, mr_{21}, mr_{20}, \ldots, mr_2, mr_1) = \\ = (mr_{21}, mr_{20}, \ldots, mr_2, mr_1, mr_{32}, mr_{31}, mr_{30}, mr_{29}, mr_{28}, mr_{27}, mr_{26}, mr_{25}, mr_{24}, mr_{23}, mr_{22}).$$

Відповідний зсув маски задано так:

$$M(m_{32}, m_{31}, m_{30}, m_{29}, m_{28}, m_{27}, m_{26}, m_{25}, m_{24}, m_{23}, m_{22}, m_{21}, m_{20}, \ldots, m_2, m_1) = \\ = (m_{21}, m_{20}, \ldots, m_2, m_1, m_{32}, m_{31}, m_{30}, m_{29}, m_{28}, m_{27}, m_{26}, m_{25}, m_{24}, m_{23}, m_{22}).$$

Маскований ключ шифрування $(\widetilde{W}_1, \widetilde{W}_2, ..., \widetilde{W}_{256})$ з маскою $(u_1, u_2, ..., u_{256})$, причому $\widetilde{W}_q = W_q \oplus u_q$, $W_q, u_q \in \{0,1\}$, $q = 1, \ldots, 256$ записують послідовно у 32-розрядні бітові вектори маскованого ключа $\widetilde{X}_0, ..., \widetilde{X}_7$. Аналогічно записують маску ключа у вісім 32-розрядних векторів $y_0, ..., y_7$. Вміст восьми 32-розрядних векторів $X_0, X_1, \ldots, X_7$ та відповідних векторів $y_0, ..., y_7$ має такий вигляд [13]:

$$\widetilde{X}_0 = (\widetilde{W}_{32}, \widetilde{W}_{31}, ..., \widetilde{W}_2, \widetilde{W}_1); \; y_0 = (u_{32}, u_{31}, ..., u_2, u_1);$$

$$\widetilde{X}_1 = (\widetilde{W}_{64}, \widetilde{W}_{63}, ..., \widetilde{W}_{34}, \widetilde{W}_{33}); \; y_1 = (u_{64}, u_{63}, ..., u_{34}, u_{33});$$

…

$$\widetilde{X}_7 = (\widetilde{W}_{256}, \widetilde{W}_{255}, ..., \widetilde{W}_{226}, \widetilde{W}_{225}); \; y_7 = (u_{256}, u_{255}, ..., u_{226}, u_{225}).$$

Відкриті дані у МП, що підлягають зашифруванню, та їх відповідні маски розбивають на блоки по 64 біти. Блок даних розбивається на дві частини і присвоюється векторам $\widetilde{N}_1$ (права частина) і $\widetilde{N}_2$ (ліва частина). Маска блоку даних також розбивається на дві частини і присвоюється векторам x_1 (права частина) і x_2 (ліва частина).

Алгоритм зашифрування 64-розрядного блоку відкритих даних у МП складається з 32 циклів [13]. У першому циклі початкове значення вектора $\widetilde{N}_1$ додається за модулем 2^{32} до значення вектора $\widetilde{X}_0$, використовуючи додаткові маски q та z, при цьому значення векторів $\widetilde{N}_1$ і x_1 передаються на вихід першого циклу. Результат додавання перетворюється за таблицею підстановки *MK* з використанням проміжних масок p, v і початкової маски таблиці z'.

Отриманий вектор даних у МП і відповідна маска циклічно зсуваються на одинадцять бітів у бік старших розрядів за допомогою операції *MR* та *R*, відповідно.

Результат зсуву додається за модулем 2 до 32-розрядного значення маскованого вектора $\widetilde{N}_2$ з використанням операції MXOR і додаткової маски w. Отриманий результат присвоюється $\widetilde{N}_1$, при цьому попереднє значення $\widetilde{N}_1$ присвоюється $\widetilde{N}_2$, новоутворена маска присвоюється x_1, попереднє значення x_1 присвоюється x_2. Перший цикл закінчується.

Наступні цикли виконуються аналогічно. У другому циклі використовується цикловий ключ $\widetilde{X}_1$ з відповідною маскою, в третьому циклі – цикловий ключ $\widetilde{X}_2$ з відповідною маскою і так далі, аж до восьмого циклу включно. В циклах з дев'ятого по шістнадцятий, а також з сімнадцятого по двадцять четвертий циклові ключі та їх маски використовуються у тому ж порядку: $\widetilde{X}_0, \widetilde{X}_1, \widetilde{X}_2, \widetilde{X}_3, \widetilde{X}_4, \widetilde{X}_5, \widetilde{X}_6, \widetilde{X}_7$. В останніх восьми циклах – з двадцять п'ятого по тридцять другий – порядок використання циклових ключів та їх масок зворотний: $\widetilde{X}_7, \widetilde{X}_6, \widetilde{X}_5, \widetilde{X}_4, \widetilde{X}_3, \widetilde{X}_2, \widetilde{X}_1, \widetilde{X}_0$. Отже, у випадку зашифрування в 32 циклах порядок вибору циклових ключів виконується так:

$$\{\widetilde{X}_0, y_0\}, \{\widetilde{X}_1, y_1\}, ..., \{\widetilde{X}_7, y_7\}, \{\widetilde{X}_0, y_0\}, \{\widetilde{X}_1, y_1\}, ..., \{\widetilde{X}_7, y_7\},$$

$$\{\widetilde{X}_0, y_0\}, \{\widetilde{X}_1, y_1\}, ..., \{\widetilde{X}_7, y_7\}, \{\widetilde{X}_7, y_7\}, \{\widetilde{X}_6, y_6\}, ..., \{\widetilde{X}_0, y_0\}.$$

У тридцять другому циклі результат додавання за модулем два присвоюється вектору $\widetilde{N}_2$, а вектор $\widetilde{N}_1$ зберігає попереднє значення. Отримані після тридцять другого циклу зашифрування значення векторів $\widetilde{N}_1$ і $\widetilde{N}_2$ є блоком зашифрованих даних у МП. Для розшифрування даних у МП використовується той самий алгоритм виконання циклу над даними у МП, що й для зашифрування. Однак циклові ключі використовують у зворотному до зашифрування порядку.

4.3.2 Архітектура ядра процесора виконання алгоритму за ГОСТ 28147-89 для даних у МП

Структурна схема спеціалізованого процесора за ГОСТ 28147-89 для даних у МП включає в себе такі функціональні блоки (рис. 4.6) [13]:

- блок інтерфейсу (БІ) до шини (шини адрес (ША), шини даних (ШД) і шини керування (ШК)) універсального програмованого процесора (УПП);
- блок пам’яті ключів (БПК);
- блок пам’яті таблиці заміни (БПТЗ);
- блок обробки даних (БОД)
- блок оновлення таблиці замін (БОТЗ);
- пристрій керування (ПК);
- генератор випадкових чисел (ГВЧ).

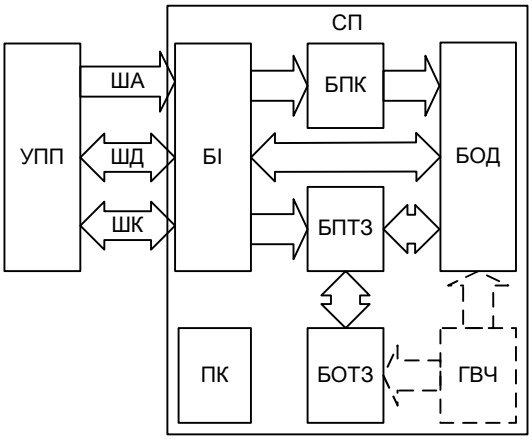
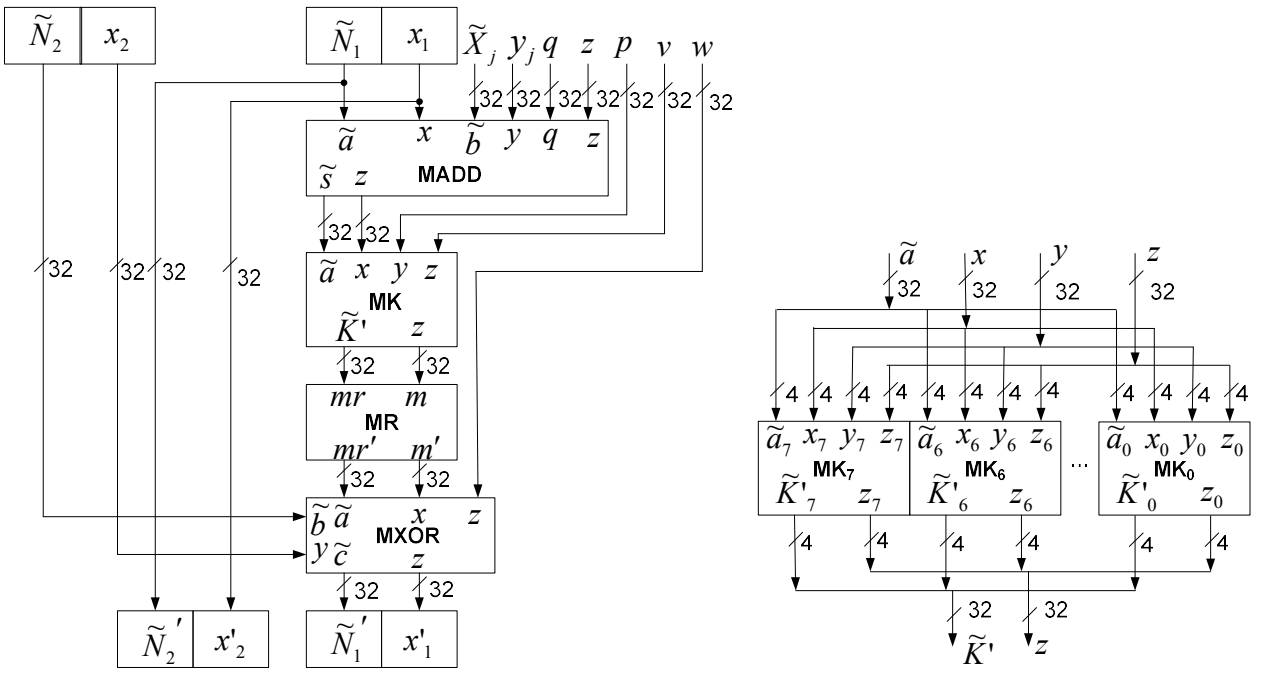


Рис. 4.6. Структурна схема спеціалізованого процесора за ГОСТ 28147-89 для даних у МП

УПП ініціалізує процесор шляхом запису у БПК маскованого ключа, маски ключа, у БПТЗ – маскованої таблиці заміни та її маски, у ПК – режиму обробки даних (зашифрування, розшифрування). Обробка даних здійснюється БОД під керуванням ПК з використанням маскованих циклових ключів та їх відповідних масок із БПК та маскованої таблиці заміни та її відповідної маски із БПТЗ. Для обробки даних і оновлення таблиці заміни використовуються випадкові числа з виходу ГВЧ. Оновлення таблиці замін із БПТЗ для різних значень маски здійснює БОТЗ.

Архітектура БОД основана на апаратному відображенні потокового графу одного циклу обробки даних у МП (рис. 4.7а) [13].



а) б)

Рис. 4.7. Структурні схеми блоків процесора: а) БОД, б) МК

БІ надсилає у БОД вхідні дані у вигляді блоків N_2 і N_1 та отримує від цього блока вихідні дані $\tilde{N}'_2$ і $\tilde{N}'_1$ та відповідні маски x'_2 і x'_1. Для обробки даних використовуються масковані циклові ключі $\tilde{X}_j$ та відповідні маски y_j з БПК, 32-розрядні випадкові числа q, z, p, v, w від ГВЧ. Права частина маскованого блоку з відповідною маскою посилаються на вхід маскованого суматора за

модулем 2^{32}. На другий вхід цього суматора поступає маскований цикловий ключ з відповідною маскою. На третій і четвертий входи суматора надано випадкові числа q і z. Результат додавання частини блоку даних у МП та маскованого ключа разом з відповідною маскою подаються на адресний вхід маскованої таблиці підстановки MK (реалізованій у БПТЗ). На додаткові входи MK подаються випадкові числа p і v. БПТЗ організований як набір із восьми 4-розрядних таблиць замін $\widetilde{K}'_i$, $0 \le i \le 7$ (рис. 4.7б).

Результат виконання таблиці підстановки із відповідною маскою подається на входи блоку циклічного зсуву MR. Результат циклічного зсуву даних у МП та маски на 11 розрядів ліворуч поступає на перший вхід суматора за модулем два даних у МП MXOR. На другий вхід цього суматора подаються ліва частина маскованого блоку даних $\widetilde{N}_2$ і відповідна маска x_2. На третій вхід суматора надано випадкове число w від ГВЧ. Результат виконання операції додавання за модулем два та маска результату присвоюються правій частині результату циклу $\widetilde{N}'_1$ та правій частині маски x'_1. Описаний процес повторюється в кожному із заданої кількості циклів.

Під час обробки даних БОТЗ оновлює вміст БПТЗ на основі проміжної маски p та вихідної масок v і початкової маски таблиці z'. Таблиця заміни оновлюється в процесі обробки даних чи в процесі ініціалізації обчислень. Час оновлення задається УПП.

4.3.3 Експериментальне дослідження ядра процесора виконання алгоритму за ГОСТ 28147-89 для даних у МП

Для експериментального дослідження ядра процесора було створено його модель з використанням мови опису апаратних засобів Verilog [119]. Результати синтезу прототипу процесора на базі цієї моделі та бібліотеки елементів КМОН НВІС 0,18 мкм наведено у табл. 4.2 [13]. Максимальна частота роботи процесора оцінена у 90 МГц, що дозволяє досягти продуктивності обробки даних у 180Мбіт/с для обробки даних у режимі простої заміни, де використовується 32

цикли шифрування. Нажаль, на час написання цієї роботи, автору були недоступні відомості про альтернативні реалізації таких процесорів для проведення порівняння розробленого алгоритму та процесора.

Характеристики складності синтезованого процесора Таблиця 4.2

Блок процесора	**Апаратна складність, логічних елементів**	**Місткістна складність, біт пам'яті**
БІ	511	
БОД	10365	
БОТЗ	1892	
ПК	1203	
БПК	453	256
БПТЗ	903	512
Всього	15327	778

4.4 Моделювання атак на основі аналізу споживаної потужності на програмні моделі процесорів

4.4.1 Система тестування моделей процесорів

Розробка напівзамовлених НВІС здійснюється, як правило, системою засобів автоматизованого проектування НВІС (рис. 4.8).

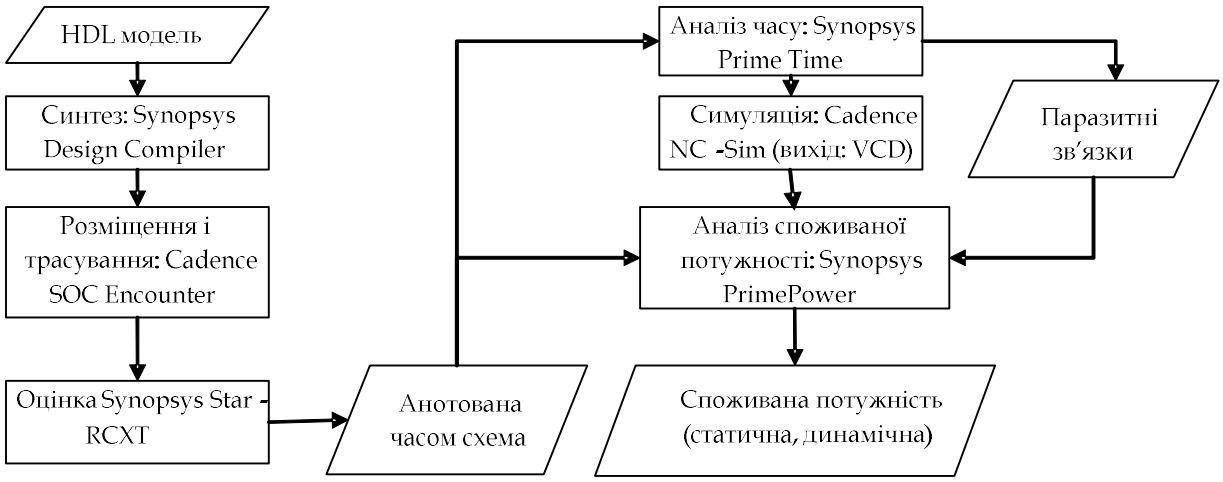


Рис. 4.8. Основні етапи автоматизованої розробки НВІС спеціалізованих процесорів

Спочатку створюється та відлагоджується програмна модель пристрою на одній із мов опису апаратних засобів (HDL – Hardware Description Language), наприклад Verilog, VHDL, SystemC, тощо. Далі з моделі синтезується схема пристрою, побудована на основі заданої бібліотеки елементів виробника. Після цього елементи отриманої схеми розміщуються на кристалі та проводяться з'єднання між цими елементами. Отримана інформація про розташування елементів використовується на наступному кроці для оцінки паразитних взаємовпливів елементів та з'єднань. Після цього визначаються часові затримки роботи елементів (при заданих зовнішніх факторах – температурі, напрузі живлення, тактових частотах) та отримують так звану часово-анотовану схему пристрою. Далі шляхом симулювання проводять перевірку роботи часово-анотованої схеми пристрою за допомогою системи тестів. В результаті такої симуляції отримують інформацію про внутрішню активність елементів схеми (їх перемикання, гонки) – файл VCD (Value-Change-Dump). Останнім етапом є оцінка СП пристрою у динаміці. Для цього аналізатором СП використовуються попередньо отримані дані про паразитні взаємовпливи елементів та з'єднань, часово-анотовану схему пристрою, файл VCD. У результаті отримують розгорнуті в часі траси СП. Оскільки було зафіксовано остаточне розміщення

елементів схеми пристрою на кристалі, для проведення DPA атак використано лише останні два етапи розробки – симуляцію роботи пристрою за допомогою тестів та отримання трас СП.

Система тестування пристрою написана на Verilog та складається з блоків інтерфейсу до моделі процесора, керування процесором та введення даних (рис. 4.9).

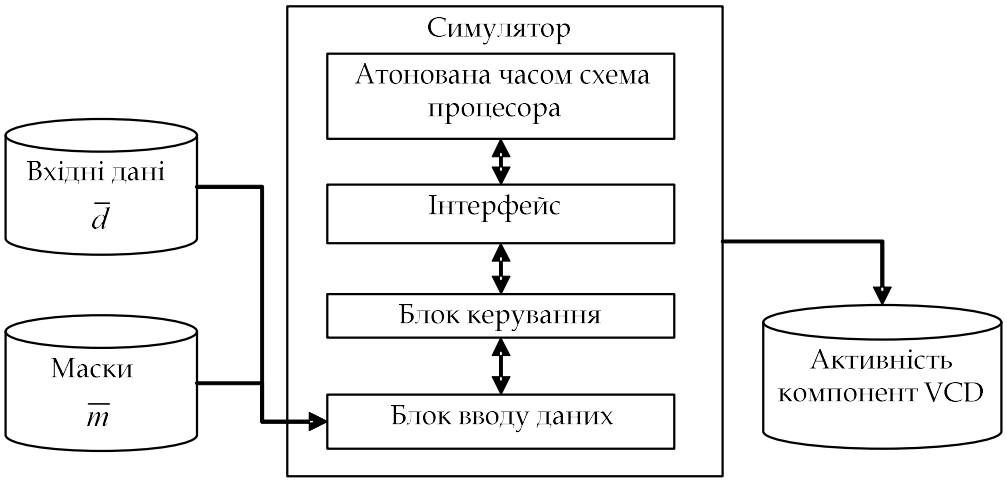


Рис. 4.9. Структура системи тестування процесорів шифрування даних у МП

Вхідними даними для тестування є файли з $\bar{d}$ та попередньо згенерованими випадковими числами $\bar{m}$. Останні використано для ініціалізації та оновлення масок та симулюють результати роботи внутрішнього генератора випадкових чисел. Вихідні дані, файл VCD, формується симулятором самостійно.

4.4.2 Моделювання DPA атаки на HDL моделі криптографічних процесорів даних у МП

Моделювання DPA атаки проведено для випадків обробки даних у немаскованому та МП. Хоча розроблені моделі процесорів обробляють дані лише у МП, їх можна використати для обох варіантів проведення атак. Якщо

встановити усі маски даних у нульове значення (усі елементи файлу $\overline{m}$ є нулями), то дані будуть оброблятися у немаскованому представленні [126].

При атакуванні процесора, який обробляє дані згідно з алгоритмом ГОСТ 28147-89, прийнято, що усі вузли заміни є відомими. Зауважимо, що при атакуванні реалізацій алгоритму на програмованих процесорах, можна відновити й невідомі вузли заміни [2]. У ролі проміжного значення $f_G(d,k)$ було обрано чотири молодших біти часткового результату обчислення першого циклу:

$$f_G(d,k) = S^0(k^0_{3...0} + d_{3...0}), \tag{4.48}$$

де S^0 – перший вузол заміни, $k^0_{3...0}$ – чотири молодших біти першого циклового ключа, встановлено у $78_{10} = 4E_{16}$, $d_{3...0}$ – відповідні біти даних, які подають на вхід, $\boxplus$ – операція додавання за модулем 16.

Аналогічно для процесора, який обробляє дані згідно з алгоритмом mCrypton, у ролі проміжного значення $f_C(d,k)$ було обрано чотири молодших біти проміжного результату обчислення першого циклу:

$$f_C(d,k) = S^0(K[0]_{3...0} \oplus d_{3...0}), \tag{4.49}$$

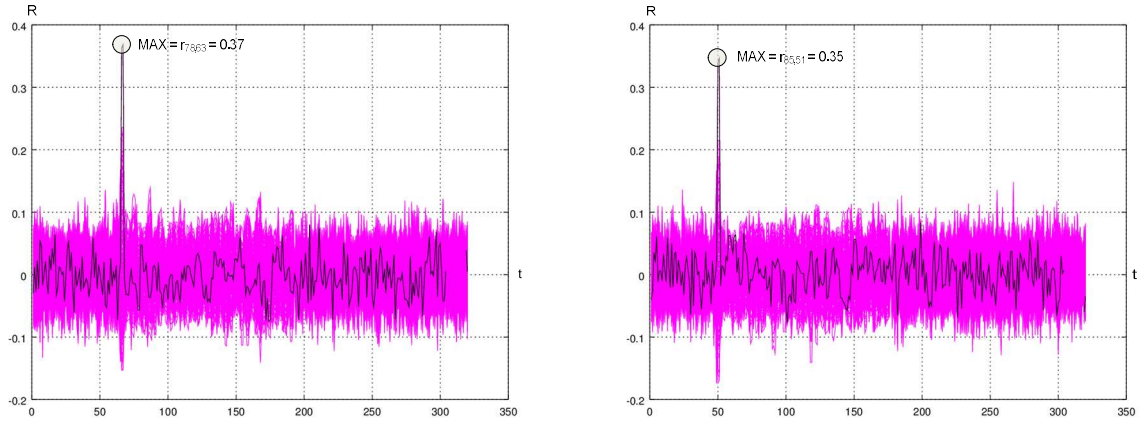
де S^0 – перший вузол заміни, $K[0]_{3...0}$ – чотири молодших біти першого циклового ключа, встановлено у $85_{10} = 55_{16}$, $d_{3...0}$ – відповідні біти даних, які подають на вхід, $\oplus$ – операція додавання за модулем два.

Для виявлення використаних частин циклових ключів було розроблено та реалізовано спеціальне програмне забезпечення, яким, на основі (4.49) і (**Error! Reference source not found.**), було обчислено відповідні матриці $|V_G|$, $|V_C|$ усіх можливих варіантів проміжних значень на заданому наборі вхідних даних $\overline{d}_G$, $\overline{d}_C$ та елементів циклового ключа для обох алгоритмів. Після цього проведено відображення матриць $|V|$ у відповідні матриці $|H_G|$, $|H_C|$ очікуваних значень СП із використанням спрощеної моделі на основі ваги Хемінга виду: $h_{i,j} = HW(v_{i,j}) * b$

, де $b = 0.013$ – масштабний коефіцієнт вкладу однієї одиниці Хемінгової ваги даних у СП пристрою.

Симульовані траси СП процесорів було отримано шляхом подавання заданих наборів вхідних даних $\bar{d}$ у симулятор, запису VCD файлів активності та подальшого їх аналізу аналізатором СП (рис. 4.9). Отримані траси $|W_G|$, $|W_C|$ використано для подальшого кореляційного аналізу з метою виявлення використаних частин циклових ключів. Для цього програмне забезпечення обчислює кореляційні коефіцієнти матриці $|R|$. Подальше графічне відображення залежності отриманих коефіцієнтів від індексу передбаченого елемента циклового ключа та часового відліку дозволяє візуально визначити шуканий індекс, при якому досягається найбільша кореляція (рис. 4.10). За незначного модифікування програмного забезпечення можна визначати максимальні коефіцієнти автоматично.

Як видно з рис. 4.10а, для процесора за ГОСТ 28147-89 найбільше значення кореляції фіксується у коефіцієнті $r_{78,63} = 0.37$, що відповідає встановленому елементу циклового ключа – 78_{10}. Аналогічно, на рис. 4.10б, для за mCrypton найбільше значення кореляції фіксується у коефіцієнті $r_{85,51} = 0.35$, що також відповідає встановленому елементу циклового ключа для цього процесора – 85_{10}. Залежність величини коефіцієнтів, які відповідають вірним елементів циклових ключів, подано темним кольором. Решта залежностей подано світлим кольором. Зауважимо, що величини кореляційних коефіцієнтів, які не відповідають встановленим елементів циклових ключів, лежать у діапазоні ± 0.15, тому можуть бути легко відфільтровані навіть за візуального перегляду.



а) б)

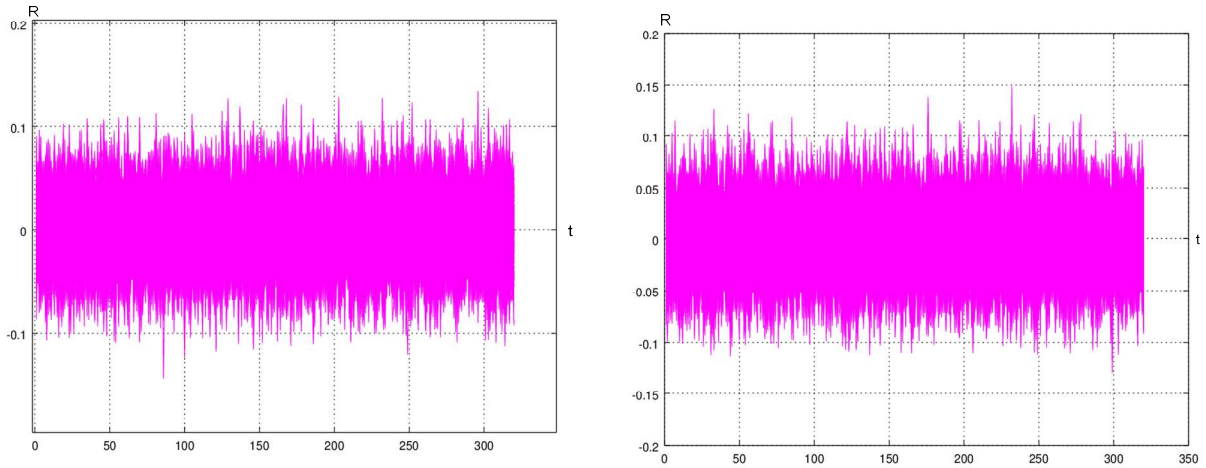
Рис. 4.10. Результати експерименту з визначення залежності коефіцієнтів кореляції від індексу передбаченого ключа та значення траси для процесорів без МП даних: а) для процесора за ГОСТ 28147-89, б) для процесора за mCrypton

Для визначення решти елементів циклового ключа у процесорі на основі mCrypton необхідно побудувати нову атаку, цього разу взявши нове відповідне проміжне значення $f_G(d,k)$. За необхідності продовжують атаку та визначають елементи другого циклового ключа і так далі.

На відміну від mCrypton, процес визначення наступних елементів циклового ключа за алгоритмом ГОСТ 28147-89 володіє такими особливостями: внаслідок використання операції додавання за модулем 2^{32}, успішність визначення кожних наступних чотирьохбітових елементів першого циклового ключа залежить від відомостей про попередні елементи. Така залежність обумовлена арифметичними переносами між розрядами при виконанні операції додавання за модулем 2^{32}. Для нівелювання впливу переносу із попередніх розрядів, можна використати відомості про вже визначені елементи циклового ключа та підібрати вхідні дані таким чином, щоб уникнути генерування такого переносу. Далі, після визначення кількох елементів циклового ключа та, за наявності обчислювальних ресурсів із достатнім об’ємом пам’яті, можна змінити

проміжне значення $f_G(d,k)$ так, щоб визначити решту елементів циклового ключа методом повного перебору.

При встановленні елементів масок у відмінне від нуля значення (усі елементи файлу $\overline{m}$ є незалежними випадковими числами із рівномірним розподілом ймовірності), процесори обробляють дані у МП із використанням цих масок. Для проведення моделювання атак було використано аналогічні функції проміжних значень. Визначення циклових ключів проводилося згідно з описаною вище методикою. У результаті обчислень матриць кореляційних коефіцієнтів $|R|$, виявлено, що значення кореляційних коефіцієнтів не виходить за межі інтервалу ± 0.15 та також відсутні їх різко виражені піки. Тому, спираючись на величини кореляційних коефіцієнтів, не вдається встановити елементи циклових ключів, оскільки ці величини для усіх варіантів елементів циклових ключів не дають однозначної відповіді про значення цих елементів ключів (рис. 4.11).



а) б)

Рис. 4.11. Результати експерименту з визначення залежності коефіцієнтів кореляції від індексу передбаченого ключа та значення траси для процесорів з МП даних: а) для процесора за ГОСТ 28147-89, б) для процесора за mCrypton

Таким чином, результати моделювання DPA атаки першого порядку з використанням кореляційного аналізу на моделі процесорів, дозволяють зробити

висновок про підвищену стійкість отриманих моделей процесорів до атаки такого виду.

4.5 Висновки до четвертого розділу

1. На основі розроблених у другому розділі методів та запропонованих у третьому розділі структур ОБ для виконання базових операцій алгоритмів криптографічних перетворень вперше адаптовано алгоритм mCrypton до обробки даних у МП, створено та досліджено процесор обробки даних у МП згідно з адаптованим алгоритмом.

2. На основі розроблених у другому розділі методів та запропонованих у третьому розділі структур ОБ для виконання базових операцій алгоритмів криптографічних перетворень вперше адаптовано алгоритм ГОСТ 28147-89 до обробки даних у МП, створено та досліджено процесор обробки даних у МП згідно з адаптованим алгоритмом.

3. Розроблено систему моделювання DPA атаки першого порядку на основі кореляційних коефіцієнтів на HDL моделі криптографічних процесорів за алгоритмами ГОСТ 28147-89 та mCrypton, які обробляють дані у МП. Розроблена система дозволяє прискорити оцінку стійкості HDL моделей процесорів до їх фактичного виготовлення, що значно скорочує час, необхідний для відлагодження роботи процесорів.

5. Виходячи з результатів моделювання DPA атаки на розроблені HDL моделі криптографічних процесорів показано, що ці процесори, на відміну від існуючих, володіють стійкістю до DPA атаки першого порядку на основі кореляційних коефіцієнтів за рахунок особливостей своєї архітектури, орієнтованої на обробку даних у МП із використанням одної маски. Характеристики розроблених ядер процесорів дозволяють рекомендувати їх до використання у пристроях з обмеженими ресурсами (смарт-карти, криптографічні токени, мобільні пристрої зв’язку тощо), які будуть мати підвищену стійкість до DPA атак першого порядку.

ВИСНОВКИ

Результатом виконаної роботи є розв’язання наукової задачі побудови і дослідження методів та засобів виконання маскованої арифметики, що можуть використовуватися для побудови криптографічних пристроїв із підвищеною стійкістю до інженерно-криптографічних атак на основі аналізу споживаної потужності. У процесі виконання дисертаційної роботи отримані такі результати:

1. Проведено аналіз існуючих методів та засобів захисту від інженерно-криптографічних атак на основі аналізу споживаної потужності, що дозволило виявити їх недоліки та сформулювати завдання щодо розробки нових та удосконалення існуючих методів та засобів виконання операцій маскованої арифметики для криптографічних пристроїв систем захисту інформації.

2. Запропоновано метод виконання операції диз’юнкції над даними у маскованому представленні із довільною кількістю масок, що, за рахунок обчислення функції корекції маски результату з використанням виключно даних у маскованому представленні та їх масок, дозволяє використати таку операцію для побудови структур криптографічних операційних блоків виконання операції диз’юнкції, рівень безпеки яких до атак на основі аналізу споживаної потужності визначається кількістю масок даних у маскованому представленні.

3. Запропоновано метод перетворення маскованого предствлення даних, що, за рахунок використання операції додавання за модулем 2^N над даними у маскованому представленні, побудованої на основі маскованих логічних операцій, на відміну від існуючих, дозволяє перетворювати масковане представлення даних довільної розрядності із арифметичним маскуванням у дані із логічним маскуванням та навпаки, а також використати таке перетворення для створення структур криптографічних операційних блоків, які використовують часту зміну типу маскування. Завдяки використанню нової маски результату, розроблені структури операційних блоків будуть володіти підвищеною стійкістю до атак на основі аналізу споживаної потужності першого порядку.

4. Розвинуто метод виконання операції кон’юнкції над даними у маскованому представленні, що, за рахунок введення у функцію корекції маски результату обчислень з урахуванням усіх масок вхідних та вихідних даних, дозволяє використати таку операцію для побудови структур криптографічних операційних блоків виконання операції кон’юнкції, рівень безпеки яких до атак на основі аналізу споживаної потужності визначається кількістю масок даних у маскованому представленні.

5. Розвинуто метод інвертування даних у маскованому представленні у полях виду $GF(2^N)$, що, за рахунок введення у функцію корекції маски результату обчислень з урахуванням усіх масок вхідних та вихідних даних, дозволяє обробляти дані із довільною кількістю масок, а також використати таке перетворення для побудови структур криптографічних операційних блоків інвертування даних у полях виду $GF(2^N)$, стійкість яких до атак на основі аналізу споживаної потужності задається кількістю використаних масок у маскованому представленні. На відміну від існуючих, розроблені методи не є вразливими до «нуль-атак».

6. Удосконалено метод табличних перетворень даних у маскованому представленні, що, за рахунок введення додаткового проміжного маскування із узгодженим типом маски вхідних даних, який, на відміну від існуючих, дозволяє виконувати табличні перетворення над вхідними даними як із логічною, так і з арифметичною масками та отримувати результат із заданим типом маскування, а також дозволяє використати таку операцію для побудови структур криптографічних операційних блоків табличної заміни засобів шифрування даних у маскованому представленні.

7. Розроблено структури криптографічних операційних блоків виконання зазначених вище операцій, придатні як для програмної, так і апаратної реалізації, що за рахунок орієнтування на використання існуючих стандартних бібліотек елементів напівзамовлених інтегральних схем, дозволяє досягнути низької вартості та енергоспоживання, високої технологічності реалізації. Додатково, на

відміну від існуючих, програмна реалізація методів виконання операцій кон'юнкції над даними у маскованому представленні володіє на 33% меншою часовою складністю, а операції диз'юнкції – на 20%.

8. Створено програмні Verilog-моделі структур криптографічних операційних блоків виконання зазначених вище операцій та, на їх основі, ядер спеціалізованих апаратно-орієнтованих процесорів симетричного блокового шифрування за алгоритмами mCrypton та ГОСТ 28147-89, які обробляють дані із одною логічною маскою та, за результатами моделювання атаки на основі аналізу споживаної потужності, дозволяють створити криптографічні процесори з стійкістю до атак на основі аналізу споживаної потужності першого порядку, порівняно з моделями без використання маскованого представлення для даних, впроваджені у діяльність Тернопільського національного економічного університету (акт від 18.06.2015) та Інституту передових технологій Самсунг Електронікс (Республіка Корея) (акт від 01.11.2011).

9. Розроблено алгоритми та результати оцінки характеристик складності криптографічних блоків для виконання перелічених вище операцій та впроваджено у начальний процес підготовки фахівців у галузі інформаційної безпеки, що підтверджується актами про впровадження у навчальний процес Університету в Бєльсько-Бялій (Польща) (акт від 30.06.2015), Тернопільського національного економічного університету (акт від 18.06.2015) та Тернопільського національного технічного університету ім. І. Пулюя (акт від 21.06.2016).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Карпінський М. П. Статистичні моделі двомісних логічних операцій для проведення інженерно-криптографічних атак за побічними каналами витоку інформації / М. П. Карпінський, Л. М. Коркішко // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2006. – Вип. 2(13). – С. 188-196.
2. Коркішко Л.М. Статистична модель операції додавання за модулем 2N для проведення інженерно-криптографічних атак за побічними каналами витоку інформації / Л.М. Коркішко, І.В. Васильцов // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2004. – №8. – С. 115- 121.
3. Карпінський М. Інженерно-криптографічна атака за аналізом споживаної потужності на програмно-апаратні реалізації криптографічного перетворення за чинним стандартом / М. Карпінський, Л. Коркішко, Т. Коркішко // Вісник Тернопільського державного технічного університету. – 2005. – №3. – С. 127-135.
4. Karpinskyy M. Randomized execution of regular cryptographic algorithms / M. Karpinskyy, L. Korkishko, T. Korkishko // Proc. of 3-rd International Conf. "Advanced Computer Systems and Networks: Design and Application" (ACSN’2007). – Lviv, 2007. – P. 114-117.
5. Karpinskyy M. Architecture of cryptographic devices resistant to side-channel attacks / M. Karpinskyy, L. Korkishko // Proc. of the International Conf. on Computer Science and Information Technologies. CSIT-2006. – Lviv: Lviv Polytechnic National University,2006. – P. 167-170.
6. Карпінський М.П. Адаптування алгоритмів криптографічних перетворень до обробки маскованих даних / М.П. Карпінський, Л.М. Коркішко, Т.А. Коркішко // Вісник хмельницького національного університету – 2007. – №3, Том 1 – С. 67-70.

7. Карпінський М. Оцінка рівня безпеки виконуваних засобами захисту інформації операцій / М. Карпінський, Л. Коркішко, Т. Коркішко // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – Випуск 1 (14), 2007. – С. 176-187.
8. Karpinski M. Masked arithmetic: transformations of masked data representation / M. Karpinski M., L. Korkishko // Internet in the Information Society / Sc. Ed. Tadeusz Wieczorek. – Dąbrowa Górnicza: Publisher Academy of Business in Dąbrowa Górnicza. – 2007. – Pp. 101–115. – ISBN 978-83-88936-38-8. – Розділ в монографії.
9. Карпінський М.П. Узагальнений алгоритм виконання операції підстановки над даними у маскованому представленні / М.П. Карпінський, Л.М. Коркішко // Вісник Хмельницького національного університету. – 2006. – №6 (87)– С. 100-106.
10. Korkishko L. Secure and efficient AES software implementation for smart cards / L. Korkishko, E. Trichina // Lecture Notes in Computer Science: Proc. of 5th International Workshop on Information Security Applications. WISA-2004. – Berlin: Springer, 2004. – Vol. 3325. – P. 779-792.
11. Карпінський М.П. Захист двійкових суматорів від інженерно-криптографічних атак за побічними каналами витоку інформації / М.П. Карпінський, Л.М. Коркішко // Матеріали 1-ї міжнародної конференції “Комп’ютерні науки та інженерія” (CSE`2006). – Львів, 2006. – С. 58-61.
12. Karpinskyy M. Masked Encryption Algorithm mCrypton for Resource-Constrained Devices / M. Karpinskyy, L. Korkishko, A. Furmanyuk // Proc. of 4th International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2007). – Dortmund, 2007. – P. 628-633.
13. Карпінський М.П. Процесор симетричного блокового шифрування за ГОСТ 28147-89 для даних у маскованому представленні / М.П. Карпінський, Л.М.

Коркішко // Матеріали 2-ї міжнародної конференції “Комп’ютерні науки та інженерія” (CSE`2007). – Львів, 2007. – С. 86-90.

14. Закон України “Про захист інформації в автоматизованих системах” від 05.07.1994
15. Згуровський М. Проблеми інформаційної безпеки в Україні, шляхи їх вирішення / М. Згуровський // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – Київ. – 2000. – С. 10 – 14.
16. Концепція технічного захисту інформації в Україні. Затверджена постановою Кабінету Міністрів України від 8 жовтня 1997 р., № 1126.
17. Молдовян А.А. Криптография / А.А. Молдовян, Н.А. Молдовян, Б.Я. Советов – СПб.: Лань, 2000. – 224 с.
18. Elizabeth D. Kryptografia i ochrona danyh / D. Elizabeth, R. Denning – Warszawa: Wydawnictwa Naukowo-Technichne, 1993. – 428 s.
19. Вербіцький О.В. Вступ до криптології / О.В. Вербіцький – Львів: ВНТЛ, 1998. – 248 с.
20. Молдовян Н.А. Введение в практическую криптографию / Н.А. Молдовян, В.М. Зима // Учебное пособие. – СПб.: ВИКУ им. А.Ф.Можайского, 2001. – 186 с.
21. Schneier B. Applied cryptography / B. Schneier – N.Y.: Wiley, 1996. –757 p.
22. Скоростные блочные шифры / Н.А. Молдовян – СПб.: СПбГУ, 1998. – 230 с.
23. Антонюк А. Загрози інформації і канали витоку / А. Антонюк, В. Жора // Правове нормативне та метрологічне забезпечення системи захисту інформації в Україні. – Київ, 2001. – №2. – С. 42 – 46.
24. Сачков В.Н. Современные проблемы криптографии / В.Н. Сачков // Труды 17 межрегиональной конференции “Информационная безопасность регионов Росии” (ИБРР-99). – СПб.: Политехника, 2000. – С. 85 – 88.

25. Герасименко В.А. Основы защиты информации / В.А. Герасименко, А.А. Малюк // Учебное пособие. – М.: МИФИ, 1997. – 540 с.
26. Зима В. Безопасность глобальных сетевых технологий / В. Зима, А. Молдовян, Н. Молдовян – СПб.: БХВ-Петербург, 2000. – 320 с.
27. Столлингс В. Криптография и защита сетей: принципы и практика / В. Столлингс // Пер. с англ. – 2-е изд. – М.: Вильямс, 2001. – 672 с.
28. Menezes A. Handbook of Applied Cryptography / A. Menezes, Van P. Oorschot, S. Vanstone // Filadelfia: CRC Press, 2001. – 816 p.
29. Горбенко Ю.І. Побудування та аналіз систем, протоколів і засобів криптографічного захисту інформації [Текст] : монографія / Горбенко Ю.І.; за заг. ред. Горбенка І.Д.; Харків. нац. ун-т ім. В. Н. Каразіна, Приват. т-во "Ін-т інформ. технологій". - Харків : Форт, 2015. Ч. 1 : Методи побудування та аналізу, стандартизація та застосування криптографічних систем. - 2015. - 959 с.
30. Горбенко І.Д. Прикладна криптологія. Теорія. Практика. Застосування: монографія / І.Д. Горбенко, Ю.І. Горбенко; Міністерство освіти і науки, молоді та спорту України, Харківський національний університет радіоелектроніки, Приватне акціонерне товариство "Інститут інформаційних технологій". - Харків : Форт, 2012. - 868 с.
31. Корченко А. Нейросетевые модели, методы и средства оценки параметров безопасности Интернет-ориентированных информационных систем: монографія / А. Корченко, И Терейковский, Н. Карпинский, С.Тынымбаев. К. : ТОВ «Наш Формат». 2016. – 275 с.
32. Корченко О. Г., Терейковський І. А., Дзюбаненко А. О. Сучаснінейромережеві методи та моделі оцінки параметрів безпеки ресурсівінформаційних систем, Захист інформації 16 (3) (2014). 223 – 232 с.
33. Терейковський І. А. Нейромережева методологія розпізнаванняінтернет-орієнтованого шкідливого програмного забезпечення /І. А. Терейковський // Безпека інформації. – 2013. – Т. 19, № 1. – С. 24-28.

34. Хорошко В.О. Анализ топологии сети передачи данных / Скоробогатько Е., Тимченко Н., Хорошко В., Хохлачова Ю. // Інформатика та математичні методи в моделюванні. – 2015. – Т.5. – №1. – С. 19-28.
35. Хорошко В.А. Управление информационными потоками в системе защиты / Хорошко В., Иванченко Е., Хохлачова Ю. // Системи обробки інформації. – 2016. – №5(142). – С. 99-102.
36. Zhou Y. B. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing / Y.B. Zhou, D.G. Feng // Proc. of National Institute of Standardization Physical Security Testing Workshop. – 2006. – [Цит. 2006, 10 січня] – Доступний з <http://csrc.nist.gov/cryptval/physec/papers/physecpaper19.pdf>.
37. Anderson R.J. Tamper Resistance — a Cautionary Note / R.J. Anderson, M.G. Kuhn // Proc. of The Second USENIX Workshop on Electronic Commerce. – Oakland, 1996. – P. 1-11.
38. Skorobogatov S. Semi-Invasive Attacks – A New Approach to Hardware Security Analysis / S. Skorobogatov // University of Cambridge. – Cambridge, 2004. – 196 p.
39. Скоробогатов С.П. Использование сфокусированного лазерного излучения для определения состояния ячеек памяти КМОП ОЗУ / С.П. Скоробогатов, П.К. Скоробогатов // Электроника, микро- и наноэлектроника. Сб. научных статей – М.: МИФИ, 2003. – С. 37 - 42.
40. Skorobogatov S. Optically Enhanced Position-Locked Power Analysis / S. Skorobogatov // Lecture notes in computer science: Proc. of Cryptographic Hardware and Embedded Systems Workshop. CHES-2006. – Berlin: Springer, 2006. – Vol. 4249. – P. 61-75.
41. Skorobogatov S. Optical Fault Induction Attacks / S. Skorobogatov, R. Anderson // Lecture notes in computer science: Proc. of Cryptographic Hardware and Embedded Systems Workshop. CHES-2002. – Berlin: Springer, 2002. – Vol. 2523. – P. 2-12.

42. Kocher P. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems / P. Kocher // Lecture Notes in Computer Science: Proc. of International Conf. Advances in Cryptology. CRYPTO-1996. – Berlin: Springer, 1996. – Vol. 1109. – P. 104-113.
43. Quisquater J. J. Electromagnetic analysis (EMA): measures and countermeasures for smart card/ J. J. Quisquater, D. Samide // Lecture Notes in Computer Science: Proc. of International Comf. Smartcard Programming and Security. – Berlin: Springer, 2001. – Vol. 2140. – P. 200-210.
44. Kocher P. Using unpredictable information to minimize leakage from smartcards and other cryptosystems / P. Kocher, J. Jaffe, B. Jun // USA Patent, International Publication. – 2001. – US 6327661.
45. Chari S. Towards sound approaches to counteract power analysis attacks / S. Chari, C.S. Jutla, J.R. Rao, P. Rohatgi // Lecture Notes in Computer Science: Proc. of International Conf. Advances in Cryptology. CRYPTO-1999. – Berlin: Springer, 1999. – Vol. 1666. – P. 398-412.
46. Akkar M. An implementation of DES and AES, Secure against some attacks / M. Akkar, C. Giraud // Lecture Notes in Computer Science: Proc. of Cryptographic Hardware and Embedded Systems Workshop. CHES-2001. – Berlin: Springer, 2001. – Vol. 2162. – P. 309-318.
47. Mangard S. Power Analysis Attacks: Revealing the Secrets of Smart Cards / S. Mangard, E. Oswald, T. Popp // Berlin: Springer, 2007. – 337 p.
48. Messerges T. Using second-order power analysis to attack DPA resistant software / T. Messerges // Lecture Notes in Computer Science: Proc. of Cryptographic Hardware and Embedded Systems Workshop. CHESS-2000. – Berlin: Springer, 2000. – Vol. 1956. – P. 238-251.
49. Kocher P. Differential power analysis / P. Kocher, J. Jaffe, B. Jun // Lecture Notes in Computer Science: Proc. of International Conf. Advances in Cryptology. CRYPTO-1999. – Berlin: Springer, 1999. – Vol. 1666. – P. 388-397.

50. Irwin J. Instruction Stream Mutation for Non-Deterministic Processors / J. Irwin, D. Page, N. Smart // IEEE Computer Society: Proc. of IEEE International Conference on Application-Specific Systems, Architectures and Processors. – 2002. – P. 286-295.
51. Yang S. Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach / S. Yang, W. Wolf, N. Vijaykrishnan, D. Serpanos, Y. Xie // IEEE Computer Society: Proc. of Design, Automation and Test in Europe Conference and Exposition. DATE-2005. – Munich, 2005. – P. 64-69.
52. May D. Non-deterministic Processors / D. May, H. L. Muller, N. P. Smart // Lecture Notes in Computer Science: Proc. of 6th Australasian Conference Information Security and Privacy. ACISP-2001. – Berlin: Springer, 2001. – Vol. 2119. – P. 115-129.
53. Ratanpal G.B. An On-Chip Signal Suppression Countermeasure to Power Analysis Attack / G.B. Ratanpal, R.D. Williams, T.N. Blalock // IEEE Transactions on Dependable and Secure Computing. – 2004. – Vol. 1(3). – P. 179-189.
54. Muresan R. Power-Smart System-On-Chip Architecture for Embedded Cryptosystems / R. Muresan, H. Vahedi, Y. Zhanrong, S. Gregori // Proc. of the 3rh IEEE/ACM/IFIP International Conf. on Hardware/Software Codesign and System Synthesis. – ACM Press, 2005. – P. 184-189.
55. Mesquita D. Current Mask Generation: A Tansistor Level Security Against DPA Attack / D. Mesquita, J-D. Techer, L. Torres, G. Sassatelli, G. Gambon, M. Robert, F. Moraes // Proc. of the 18th Annual Symposium on Integrated Circuits and System Design SBCCI'05. – ACM Press, 2005. – P. 115-120.
56. Benini L. Energe-Aware Design Techniques for Differential Power Analysis Protection / L. Benini, A. Macii, E. Macii, E. Omerbegovic, F. Pro, M. Poncino // Proc. of 40th Design, Automation Conf., DAC-2003. – ACM Press, 2003. – P. 36-41.

57. Benini L. A Novel Architecture for Power Maskable Arithmetic Units / L. Benini, A. Macii, E. Macii, E. Omerbegovic, M. Poncino, F. Pro // Proc. of 13th ACM Great Lakes Symposium on VLSI 2004. – Washington: ACM Press, 2003. – P. 136-140.
58. Tiri K. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards / K. Tiri, M. Akmal, I. Verbauwhere // Proc. of IEEE 28th Europpean Solid-State Circuits Conf. ESSCIRC-2002. – Florence, 2002. – P. 403-406.
59. Tiri K. A Logic Level Design Methodology for Secure DPA Resistant ASIC or FPGA Imlementation / K. Tiri, I. Verbauwhere // IEEE Computer Society: Proc. of 2004 Design, Automation and Test in Europe Conference and Exposition. DATE-2004. – Paris, 2004. – Vol. 1. – P. 246-251.
60. Bystrov A. Balancing Power Sinature in Secure Systems / A. Bystrov, D. Sokolov, A. Yakovlev, A. Koelmans // Proc. of 14th UK Asynchronous Forum. – Newcastle, 2003. – [Цит. 2003, 12 червня]. – Доступний з < http://www.staff.ncl.ac.uk/i.g.clark/async/ukasyncform14/forum14-papers/forum-bystrov.pdf>.
61. Sokolov D. A. Improving the Security of Dual-Rail Circuits / D. Sokolov, J. Murphy, A. Bystrov, A. Yakovlev // Lecture Notes in Computer Science: Proc. of 6th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2004. – Berlin:Springer, 2004. – Vol. 3156. – P. 282-297.
62. Sokolov D. Design and Analysis of Dual-Rail Circuits for Security Applications / D. Sokolov, J. Murphy, A. Bystrov, A. Yakovlev // IEEE Transactions on Computers, 2005. – Vol. 54(4). – P. 449-460.
63. Bucci M. Three-Rhase Dual-Rail Pre-Charge Logic / M. Bucci, L. Giancane, R. Luzzi, A. Trifiletti // Lecture Notes in Computer Science: Proc. of 8th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2006. – Berlin:Springer, 2006. – Vol. 4249. – P. 282-297.

64. Aigner M. A Novel CMOS Logic Style with Data Independent Power Consumotion / M. Aigner, S. Mangart, R. Menicocci, M. Olivieri, G. Scotti, A. Trifiletti // Proc. of IEEE Inernational Symposium on Circuits and Systems. ISCAS-2005. – 2005. – Vol. 2. – P. 1066-1069.

65. Coron J-S. Statistics and Secret Leakage / J-S. Coron, P.C. Kocher, D. Naccache // Lecture Notes in Computer Science: Proc. of 4th International Conference Financial Cryptography. FC-2000. – Berlin:Springer, 2001. – Vol. 1962. – P. 157-173.

66. Shamir A. Protection Smart Cards from Passiv Power Analysis with Detachted Power Supplies / A. Shamir // Lecture Notes in Computer Science: Proc. of Second International Workshop Cryptographic Hardware and Embedded Systems. CHES-2000. – Berlin:Springer, 2000. – Vol. 1956. – P. 71-77.

67. Corsonello P. A New Charge-Pump Based Countermeasure Against Differential Power Analysis / P. Corsonello, S. Perri, M. Margala // Proc. of the 6th International Conference on ASIC. ASICON-2005. – IEEE, 2005.– Vol. 1. – P. 66-69.

68. Moore S. Improving Smart Card Security using Self-timed Circuits / S. Moore, R.J. Anderson, P. Cunningham, R.D. Mullins, G.S. Taylor // Proc. of Eighth Inernational Symposium on Asynchronous Circuits and Systems. ASYNC-2002. – IEEE Computer Society, 2002. – P. 211-218.

69. Yu Z. C. An Investigation into the Security of Self-Timed Circuits / Z.C. Yu, S.B. Furber, L.A. Plana // Proc. of 9th Inernational Symposium on Advanced Research in Asynchronous Circuits and Systems. ASYNC-2003. – IEEE Computer Society, 2003. – P. 206-215.

70. Kulikowski K.J. Delay Insensitive Encoding and Power Analysis: A Balancing Act / K.J. Kulikowski, M. Su, A. B. Smirnov, A. Taubin, M.G. Karpovsky, D. MacDonald // In 11th Inernational Symposium on Advanced Research in Asynchronous Circuits and Systems. ASYNC 2005. – IEEE Computer Society, 2005. – P. 116-125.

71. Kulikowski K.J. Automated Design of Chryptographic Devices Resistant to Multiple Side-Channel Attacks / K.J. Kulikowski, A. B. Smirnov, A. Taubin // Lecture Notes in Computer Science: Proc. of 8th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2006. – Berlin: Springer, 2006. – Vol. 4249. – P. 399-413.
72. Yu A. A Clock-less Implementation of the AES Resists to Power and Timing Attacks / A. Yu, D.S. Bree // Proc. of International Conf. on Information Technology: Coding and Coputing. ITCC-2004. – IEEE Computer Society, 2004. –Vol. 2. – P. 525-532.
73. Goubin L. DES and Differential Power Analysis – The Duplication Method / L. Goubin, J. Patarin // Lecture Notes in Computer Science: Proc. of First International Workshop Cryptographic Hardware and Embedded Systems. CHES-1999. – Berlin:Springer, 1999.– Vol. 1717. – P. 158-172.
74. Chari S. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards / S. Chari, C.S. Jutla, J.R. Rao, P. Rohatgi // Proc. of Second Advanced Encryption Standart (AES) Candidate Conference. – Roma, 1999.
75. Messerges T. S. Securing the AES finalists against power analysis attacks / T. S. Messerges // Lecture Notes in Computer Science: Proc. of Workshop Fast Software Encryption. – Berlin: Springer, 2000. – Vol. 1978. – P. 150-165.
76. Popp T. Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints / T. Popp, S. Mangard // Lecture Notes in Computer Science: Proc. of 7th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2005. – Berlin:Springer, 2005. – Vol. 3659. – P. 172-186.
77. Popp T. Implementation Aspects of the DPA- Resistant Logic Style MDPL / T. Popp, S. Mangard // Proc. of Inernational Symposium on Circuits and Systems. ISCAS- 2006. – IEEE, 2006. – P. 2913-2916.
78. Suzuki D. Random Switching Logic: A Countermeasure against DPA based on Transition Probability / D. Suzuki, M. Saeki, T. Ichikawa // Cryptogy ePrint Archive (http://eprint.iacr.org/), Report 2004/346, 2004.

79. Chen Z. Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage / Z. Chen, Y. Zhou // Lecture Notes in Computer Science: Proc. of 8th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2006. – Berlin:Springer, 2006. – Vol. 4249. – P. 242-254.
80. Trichina E. Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results / E. Trichina, T. Korkishko, K-H. Lee // Lecture Notes in Computer Science: Proc. of 4th Conference Advanced Encryption Standart. AES-2004. – Berlin: Springer, 2005. – Vol. 3373. – P. 113-127.
81. Golic J.D. Universal Masking on Logic Gate Level / J.D. Golic, R. Menicocci // IEE Electronic Letters. – 2004. – Vol. 40(9). – P. 526-527.
82. Ishai Y. Private Circuits: Securing Hardware against Probing Attacks / Y. Ishai, A. Sahai, D. Wagner // Lecture Notes in Computer Science: Proc. of 23th Annual Inernational Cryptologe Conference Advances in Cryptology. CRYPTO-2003. – Berlin: Springer, 2003. – Vol. 2729. – P. 463-481.
83. Коркішко Л. Операція множення даних у маскованому представленні / Л. Коркішко // Матеріали XI наукової конференції Тернопільського державного технічного університету ім. І.Пулюя. – Тернопіль, 2007. – С. 83.
84. Benini L. Energi-Efficient Data Scrambling on Memory-Prosessor Interfaces / L. Benini, A. Galati, A. Macii, E. Macii, M. Poncino // Proc. of International Symposium on Low Power Electronics and Desin. – Berlin: Springer, 2003. – P. P. 26-29.
85. Golik J. D. DeKaRT: A New Paradigm for Key-Dependent Reversible Circuits / J. D. Golik // Lecture Notes in Computer Science: Proc. of 5th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2003. – Berlin: Springer, 2003. – Vol. 2779. – P. 98-112.
86. Elbaz R. Hardware Engines for Bus Encryption: A Survey of Existing Techniques / R. Elbaz, L. Torres, G. Sassatelli, P. Guillemin, C. Anguille, M. Bardouillet, C. Buatois, J-B. Rigaud // Proc. of Design, Automation and Test in Europe

Conference and Exposition. DATE-2005. – IEEE Computer Society, 2005. – P. 40-45.

87. Bucci M. A Power Consumption Randomization Countermeasure for DPA-Resistant Cryptographic Processors / M. Bucci, M. Gugieimo, R. Luzzi, A. Trifiletti // Lecture Notes in Computer Science: Proc. of 14th International Workshop on Integrated Circuit and System Design, Power and Timing Modeling, Optimization and Simulation. PATMOS 2004. – Berlin: Springer, 2004. – Vol. 3254. – P. 481-490.
88. Коркішко Т.А. Захист інформації в комп'ютерних і телекомунікаційних мережах: Алгоритми та процесори симетричного блокового шифрування / Т.А. Коркішко, А. О. Мельник, В.А. Мельник – Львів: БАК, 2003. – 168 с.
89. Coron J.S. A new algorithm for switching from arithmetic to boolean masking / J.S. Coron, A. Tchulkine, C. Walter, C.K. Koc, C. Paar // Lecture Notes in Computer Science: Proc. of International workshop Cryptographic hardware and embedded systems. CHES-2003. – Berlin: Springer, 2003. – Vol. 2779. – P. 89-97.
90. Golic J. Multiplicative masking and power analysis of for AES / J. Golic, Ch. Tymen // Lecture Notes in Computer Science: Proc. of International workshop Cryptographic Hardware and Embedded Systems. CHES 2002. – Berlin: Springer, 2002. – Vol. 2523. – P. 198-212.
91. Golic J.D. Techniques for random masking in hardware / J.D. Golic // IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2007. – Vol. 54 (2) – P. 291-300.
92. Trichina E. Combinatorial logic design for AES sybbyte transformation on masked data / E. Trichina // Cryptology eprint archive: Report 2003/236, IACR, November 11, 2003.
93. Trichina E. Simplified adaptive Multiplicative masking for AES and its secure implementation / E. Trichina, D. De Seta, L. Germani // Lecture Notes of

Computer Science: Proc. of Cryptographic Hardware and Embedded Systems. CHESS-2002. – Berlin: Springer, 2003. – Vol. 2532. – P. 187-197.

94. Oswald E. An efficient masking scheme for AES software implementations / E. Oswald, K. Schramm // Lecture notes in computer science: Proc. of 6th International Workshop on Information Security Applications. WISA-2005. – Berlin: Springer, 2005. – Vol. 3786. – P. 292-305.
95. Oswald E. Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers / E. Oswald, S. Mangard, C. Herbst, S. Tillich // Lecture Notes in Computer Science: Proc. of CT-RSA 2006. – Berlin: Springer, 2006. – Vol. 3860. – P. 192-207.
96. Schramm K. High Order Masking of the AES / K. Schramm, C. Paar // Lecture Notes in Computer Science: Proc. of CT-RSA 2006. – Berlin: Springer, 2006. – Vol. 3860. P. 208-225.
97. Daemen J. The design of Rijndael: AES – The Advanced Encryption Standard / J. Daemen, V. Rijmen – Berlin: Springer, 2002.
98. ГОСТ28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М.: Госстандарт СССР.
99. Rivest R. The RC6 Block Cipher / R. Rivest, M. Robshaw, R. Sidney, Y. Yin // First Advanced Encryption Standard(AES) Conference. – Ventura, 1998.
100. Lai X. A Proposal for a New Block Encryption Standard / X. Lai, J. Massey // Proc. of International Conference Advances in Cryptology. EUROCRYPT-1990. – Berlin: Springer, 1991. – P. 389- 404.
101. Coron J.S. On Boolean and arithmetic masking against differential power analysis / J.S. Coron, L. Goubin // Lecture Notes in Computer Science: Proc. of international Workshop Cryptographic Hardware and Embedded Systems. CHES-2000. – Berlin: Springer, 2000. – Vol. 1965. – P. 231-237.
102. Goubin L. A Sound Method for Switching between Boolean and Arithmetic Masking / L. Goubin // Lecture Notes In Computer Science: Proc. of Third

international Workshop on Cryptographic Hardware and Embedded Systems. CHES-2001. – Berlin: Springer, 2001. – Vol. 2162. – P. 3-15.

103. Baek Y.-J. Differential Power Attack and Masking / Y.-J. Baek, M.-J. Noh // Trends in Mathematics, 2005. – Vol. 8(1). – P. 53-67.
104. Trichina E. Secure AES Hardware Module for Resource Constrained Devices / E. Trichina, T. Korksihko // Lecture Notes in Computer Science. – Berlin: Springer, 2005. – Vol. 3313. – P. 215-229.
105. Коркішко Л.М. Базові логічні елементи для комп'ютерних пристроїв захисту інформації / Л.М. Коркішко // Вісник Національного університету "Львівська політехніка" "Комп'ютерні системи та мережі". – Львів, 2006. – №573 – С. 103- 113.
106. De Win E. A fast software implementation for arithmetic operations in $GF(2^n)$ / E. De Win, A. Bosselaers, S. Vadenberghe, P. De Gersem, J. Vandewalle // Lecture Notes in Computer Science: Proc. of International Conference Advances in Cryptology. ASIACRYPT-1996. – Berlin: Springer, 1996. – Vol. 1163. – P. 65-76.
107. Huang C. Fast software implementation of finite field operation / C. Huang, L. Xu // Technical Report. – Washington, Washington University in St. Louis, 2003. – [Цит. 2003, січень]. – Доступний з <http://www.nisl.wust1.edu/Papers/Tech/GF.pdf>.
108. Korkishko L. Inversion of masked data in GF(2^N) / L. Korkishko // Proc. of International Conf. TCSET-2008. – Lviv-Slavsko (Ukraine), 2008. – P. 573-576.
109. Blomer J. Provably secure masking of AES / J. Blomer, J.G. Merchant, V. Krummel // Lecture Notes in Computer Science: Proc. of 11th International Workshop Selected Areas in Cryptography. SAC 2004. – Berlin: Springer, 2004. – Vol. 3357. – P. 69-83.
110. Мельник А.О. Спеціалізовані комп'ютерні системи реального часу / А.О. Мельник– Львів: Державний університет "Львівська політехніка", 1996. – 54 с.

111. Shamir A. How to share a secret / A. Shamir // Communications of the ACM, 1979. – Vol. 22(1). – P. 612-613.
112. Messerges T. Eximining smart-card security under the threat of power analysis attack / T. Messerges, E. Dabbish, R. Sloan // IEEE Transactions on computers. – 2002. – Vol. 51 (5). – P. 541-552.
113. Черкаський М. Складність апаратно-програмних комп'ютерних засобів / М. Черкаський // Сучасні проблеми в комп'ютерних науках. Contemporary Computing in Ukraine CCU'2000. Збірник наукових праць. – Львів, 2000. – С. 58-67.
114. Коркішко Л. Операційні пристрої логічних операцій над даними у маскованому представлені / Л.М. Коркішко // Проблеми інформатизації та управління: збірник наукових праць. – К: НАУ, 2008. - № 1. – С.176-181.
115. Nuno Roma. Fast Adder Architectures: Modeling and Experimental Evaluation / Nuno Roma and Tiago Dias and Leonel Sousa.// Proc. of XVIII Conference on Design of Circuits and Integrated Systems. DCIS-2003. – 2003. – P. 367-372.
116. Rodriguez-Henriquez F. On fully parallel karatsuba multipliers for GF(2m) / F. Rodriguez-Henriquez, C. Koc // Proc. of International Conf. on Computer Science and Technology CST-2003. – 2003. – P. 405-410.
117. Itoh T. A Fast Algorithm for Computing Multiplicative Inverses in GF(2m) Using Normal Bases / T. Itoh, S. Tsujii // Information and Computation. – 1988. – Vol. 78. – P. 171-177.
118. Lim C.H. mCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors / C.H. Lim, T. Korkishko // Lecture Notes in Computer Science: Proc. of 6th International Workshop on Information Security Applications. WISA 2005. – Berlin: Springer, 2006. – Vol. 3786. – P. 243-258.
119. IEEE Standard. Verilog Hardware Description Language Reference Manual. Standard 1364-1995, New York: IEEE, 1995.
120. Karpinskyi V. Side-Channel Signal Processing and Modeling / V. Karpinskyi, L. Korkishko, M. Karpinski // Advanced Computer Systems and Networks: Design

and Application – ACSN-2009: 4th International Conference, November 9-11th, 2009: Proceedings of the Conference. – Lviv, Ukraine, 2009. – P. 199-202. – ISBN 978-966-345-190-9.

121. Korkiszko Ł. Odporność symetrycznych szyfrów blokowych na atak typu analizy mocy sensorowych / Ł. Korkiszko // Bezpieczeństwo informacji / M. Karpiński. – Warszawa: Wydawnictwo Pomiary Automatyka Kontrola. – 2012. – Rozd. 6. – S. 197-259. – ISBN 978-83-930505-3-6. [Information Security.– Warsaw: Measurements, Automation and Monitoring.– 280 p.] (in Polish) – Розділ в монографії.
122. Balyk A. A Survey of Modern IP Traceback Methodologies / A. Balyk, U. Iatsykovska, M. Karpinski, Y. Khokhlachova, A. Shaikhanova, L. Korkishko // Proceedings of the 2015 IEEE 8th International Conference on “Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications” (IDAACS’2015), Warsaw, Poland, September 24-26, 2015: Vol. 1. – P. 484-488.
123. Agrawal D. Multi-channel Attacks / D. Agrawal, J.R. Rao, P. Rohatgi // Lecture Notes in Computer Science: Proc. of 5th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2003. – Cologne, Germany: Springer, 2003. – Vol. 2779. – P. 2-16.
124. Chari S. Template Attacks / S. Chari, J.R. Rao, P. Rohatgi // Lecture Notes in Computer Science: Proc. of 4th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2002. – Redwood Shores, CA, USA: Springer, 2003. – Vol. 2523. – P. 13-28.
125. Brier E. Correlation Power Analysis with a Leakage Model / E. Brier, C. Clavier, F. Olivier // Lecture Notes in Computer Science: Proc. of 6th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2004. – Cambridge, MA, USA: Springer, 2004. – Vol. 3156. – P. 16-29.
126. Карпінський М.П. Моделювання DPA атаки першого порядку / М.П. Карпінський, Л.М. Коркішко // Безпека інформації. – 2016. – Т.22. – №2. – С. 184-195.

ДОДАТОК А
СТАТИСТИЧНІ ВЛАСТИВОСТІ ОПЕРАЦІЙ

Означення 1 (МП). МП даних $a \in Z_n$ називається його подання у вигляді пари $\{\breve{a}, x\}$, де $\breve{a} = a\, op\, x$, op – одна із бінарних операцій "$+$", "$\oplus$" або "$\otimes$", $x \in Z_n$ – елемент маски, випадково обраний з рівномірним розподілом ймовірності із Z_n .

Отримання немаскованого представлення даних згідно з означенням 1 означає виконання операції op над даним у МП і з елементом, оберненим до маски (інший варіант – використання оберненої операції з тією ж маскою).

Означення 2 (ЛМ). МП з використанням логічної маски називається представлення $\{\tilde{a}, x\} = \{a \oplus x, x\}$.

Означення 3 (арифметичне маскування). МП з використанням арифметичної маски називається представлення $\{\hat{a}, x\} = \{a + x, x\}$.

Означення 4 (мультиплікативне маскування). МП з використанням мультиплікативної маски називається представлення $\{\bar{a}, x\} = \{a \otimes x, x\}$.

Пряме виконання базових операцій алгоритмічних криптографічних перетворень над даними у МП не є тривіальним, оскільки у процесі отримання результату необхідно уникати розголошення відомостей про немасковані дані. Для цього необхідно використовувати спеціальні алгоритми виконання цих базових операцій, які забезпечують отримання необхідного результату у МП. При цьому практичні реалізації таких алгоритмів використовують джерело випадкових чисел з рівномірним розподілом ймовірності. Приклад архітектури пристрою для виконання операцій над даними у МП наведено у [5].

Означення 5 (повне маскування). Процес обчислення результатів перетворення ENC володіє властивістю повного маскування χ -го порядку, якщо для усіх наборів $I_1,...,I_\chi$ проміжних результатів виконується рівність $D_{a,k}(R) = D_{a',k'}(R)$ для усіх пар (a,k), (a',k'). Для $\chi = 1$ приймають, що процесу

обчислень притаманна властивість повного маскування від зловмисника першого порядку.

Частковий аналіз розподілу ймовірності проміжних результатів з використанням операції у полі $GF(256)$ здійснено у [109]. Подальший аналіз для операцій з поля $GF(2^l)$ наведено у [95]. Проведемо аналіз розподілу ймовірності проміжних даних для деякої скінченої адитивної Абелевої групи і деякого скінченого поля з врахуванням означення 5.

Нехай задано: а) множину Z_n, $|Z_n| = n$ наділену бінарною операцією "$\bullet$", утворюючу адитивну Абелеву групу $G_\bullet$; б) множину Z_n, $|Z_n| = n$ та дві бінарні операції – адитивну "$\bullet$" і мультиплікативну "$*$", такі, що разом з Z_n утворюють поле F. Z_n утворює Абелеву групу з нейтральним елементом 0 відносно операції "$\bullet$", тоді як $Z_n^* = \mathrm{Z_n} \setminus \{0\}$ – Абелеву групу з нейтральним елементом 1 відносно операції "$*$", причому операція "$*$" є дистрибутивна з операцією "$\bullet$".

Лема 1. Нехай a, $x \in Z_n$, де a – фіксований (заданий, відомий наперед) елемент, x – елемент, вибраний випадково з рівномірним розподілом ймовірності незалежно від a. Тоді $\hat{a} = a \bullet x$ і $\hat{a} \in Z_n$ є випадковим елементом з рівномірним розподілом ймовірності.

Доведення леми 1. Елемент $\hat{a} = a \bullet x$ належить множині Z_n внаслідок замкнутості групи $G_\bullet$. Тоді обчислення елемента $\hat{a} = a \bullet x$ еквівалентне до взаємно однозначного відображення (бієкції) множини Z_n самої у себе: $B : Z_n \xrightarrow{a \bullet x} Z_n$ для фіксованого (заданого) $a \in Z_n$ і випадкового x. Вибір певного нового x призводить до вибору деякого нового взаємно однозначного відображення B. При цьому, внаслідок ін'єктивності бієктивного відображення, для різних x обране нове відображення B також буде різним, а кількість таких відображень дорівнює кількості варіантів вибору x і становить $|Z_n| = n$. Беручи до уваги, що a є фіксованим (відомим, ймовірність його появи дорівнює одиниці), то ймовірність появи різних відображень B дорівнює ймовірності

появи значення x. Оскільки, за умовою, x володіє рівномірним розподілом ймовірності, то ймовірність появи довільного значення x дорівнює $1/n$. Тому, ймовірність появи різних відображень B характеризується значенням $1/n$.

Із ін'єктивності відображень B випливає, що кожен елемент із Z_n буде відображатися у інший елемент із цієї ж множини (в тому числі і сам у себе), а ймовірність такого відображення дорівнює ймовірності появи відображення B і становить $1/n$. Таким чином, $\hat{a} \in Z_n$ володіє рівномірним розподілом ймовірності.

Наслідок з леми 1. Беручи до уваги лему 1 та її доведення, отримуємо що для a, $x \in Z_n$, де a і x – вибрані випадково з Z_n із рівномірним розподілом ймовірності і незалежно один від одного, справджується $\hat{a} = a \bullet x$ і $\hat{a} \in Z_n$ є випадковим елементом з рівномірним розподілом ймовірності.

Лема 2. Нехай a, $a' \in F$ є заданими (відомими наперед), x, $x' \in F$ є незалежними і рівномірно розподіленими на множині Z_n. Встановимо $I_1 = a \bullet x$ і $I_2 = a' \bullet x'$. Тоді добуток $Z = I_1 * I_2$ володіє розподілом ймовірності:

$$P(Z = i) = \begin{cases} (2 \mid Z_n \mid -1) / \mid Z_n \mid^2, & якщо\ i = 0, \\ (\mid Z_n \mid -1) / \mid Z_n \mid^2, & якщо\ i \neq 0, \end{cases} \qquad (A.1)$$

де $i \in Z_n$.

Доведення леми 2. Згідно з лемою 1 $I_1 = a \bullet x$ і $I_2 = a' \bullet x'$ володіють рівномірним розподілом ймовірності. Кількість усіх можливих комбінацій I_1 і I_2 дорівнює $\mid Z_n \mid^2$, де $\mid Z_n \mid$ – потужність множини Z_n, а кількість різних варіантів добутків $I_1 * I_2$ становить $\mid Z_n \mid$ внаслідок замкнутості поля F та незалежності I_1 і I_2.

Подія $Z = I_1 * I_2 = 0$ можлива за таких умов: $I_1 = I_2 = 0$, $\{I_1 = 0, I_2 \neq 0\}$ чи $\{I_1 \neq 0, I_2 = 0\}$. Кількість випадків, коли справджується кожен вираз, є відповідно: 1, $\mid Z_n \mid -1$ і $\mid Z_n \mid -1$. Загальна кількість комбінацій змінних I_1 і I_2, що призводить до справдження виразів, становитиме $1 + \mid Z_n \mid -1 + \mid Z_n \mid -1 = 2 \mid Z_n \mid -1$.

Загальна кількість комбінацій змінних I_1 і I_2, що призводить до появи події $Z = I_1 * I_2 \neq 0$, дорівнюватиме $|Z_n|^2 - 2|Z_n| + 1 = (|Z_n| - 1)^2$. Однак, різних значень $Z = I_1 * I_2$ буде лише $|Z_n| - 1$ (внаслідок замкнутості поля). Тоді $P(Z = 0) = (2|Z_n| - 1)/|Z_n|^2$ і $P(Z \neq 0) = (|Z_n| - 1)/|Z_n|^2$, що й треба було довести.

Лема 3. Нехай a, $a' \in F$ є заданими (відомими наперед) і x, $x' \in F$ є незалежними та рівномірно розподіленими на множині $Z_n = \{0,1,...,2^l - 1\}$. Встановимо $I_1 = a \bullet x$ і $I_2 = a' \bullet x'$. Тоді результат виконання операції побітового логічного додавання $Z = I_1 \vee I_2$ володіє розподілом ймовірності: $P(Z = i) = \frac{3^{HW(i)}}{4^l}$, а результату здійснення операції побітового логічного множення $Z = I_1 \wedge I_2$ притаманний розподіл ймовірності: $P(Z = i) = \frac{3^{l - HW(i)}}{4^l}$, де $i \in Z_n$, $HW(i)$ – функція, яка повертає кількість одиниць (Хемінгову вагу) у двійковому представленні i.

Доведення леми 3. Згідно з лемою 1 $I_1 = a \bullet x$ і $I_2 = a' \bullet x'$ володіють рівномірним розподілом ймовірності. З іншого боку результат виконання операції логічного додавання над однорозрядними даними характеризується законом розподілу ймовірності, що описується виразом: $P(z = i) = \begin{cases} 1/4, & \text{якщо } i = 0, \\ 3/4, & \text{якщо } i = 1. \end{cases}$ Оскільки I_1 і I_2 незалежні і володіють рівномірним розподілом ймовірності, то значення результату $Z = I_1 \vee I_2$ при $l > 1$ буде залежати від кількості та позицій одиниць у двійковому представленні I_1 і I_2. Ймовірність появи деякого двійкового представлення результату $Z = \{z_{l-1}, z_{l-2}, ..., z_1, z_0\}$, де z_i – значення *i*-го розряду двійкового представлення Z, можна знайти з таких міркувань. Оберемо довільне $0 \leq k \leq l - 1$, яке позначає кількість одиниць у двійковому представленні Z. Тоді кількість нулів у цьому ж представленні дорівнюватиме $l - k$, а ймовірність появи деякої комбінації $i = \{z_{l-1}, z_{l-2}, ..., z_1, z_0\}$ із $l - k$ нулями та k одиницями становитиме $P(Z = i) = p(z_{l-1} = i_{i-1}) \cdot p(z_{l-2} = i_{i-2}) \cdot ... \cdot p(z_0 = i_0) = P(z = 0)^{l-k} P(z = 1)^k$. Підставивши у

останній вираз значення $P(z=i)$, маємо: $P(Z=i)=\left(\frac{1}{4}\right)^{l-k}\cdot\left(\frac{3}{4}\right)^{k}=\frac{3^k}{4^l}$. Враховуючи, що кількість одиниць k визначає Хемінгову вагу результату, отримаємо твердження першої частини леми.

Для доведення другої частини цієї леми зауважимо, що, згідно з лемою 2, результат виконання операції логічного множення над однорозрядними даними володіє законом розподілу ймовірності, який отримуємо з виразу (А.1): $P(z=i)=\begin{cases}3/4, & якщо\ i=0,\\ 1/4, & якщо\ i=1.\end{cases}$ Використовуючи аналогічний до першої частини підхід, знаходимо, що $P(Z=i)=\left(\frac{3}{4}\right)^{l-k}\cdot\left(\frac{1}{4}\right)^{k}=\frac{3^{l-k}}{4^l}$. Враховуючи, що кількість одиниць k визначає Хемінгову вагу результату, отримаємо твердження другої частини леми.

Лема 4. Нехай a, $x\in Z_n$, $Z_n=\{0,1,...,2^l-1\}$, де a – фіксований (заданий, відомий наперед) елемент, x – елемент, вибраний випадково з рівномірним розподілом ймовірності незалежно від a. Тоді $\overline{\widetilde{a}}=\widetilde{\overline{a}}$ і $\widetilde{\overline{a}}\in Z_n$ є випадковим елементом з рівномірним розподілом ймовірності, де "$\overline{\ }$" позначає операцію логічного заперечення.

Доведення леми 4. Зауважимо, що операцію логічного заперечення, виконану над двійковим представленням $\widetilde{a}$, можна подати у вигляді: $\overline{\widetilde{a}}=\widetilde{a}\oplus\{1\}^l=a\oplus x\oplus\{1\}^l=(a\oplus\{1\}^l)\oplus x=\widetilde{\overline{a}}$, а елемент $\{1\}^l=2^l-1$, $\{1\}^l\in Z_n$. Тоді, згідно з лемою 1, результат $\widetilde{\overline{a}}$ володіє рівномірним розподілом ймовірності.

ДОДАТОК Б
АТАКА ЗА СП НА ОСНОВІ КОРЕЛЯЦІЙНИХ КОЕФІЦІЄНТІВ

Стратегію DPA атаки на основі аналізу кореляційних коефіцієнтів можна подати у вигляді кількох етапів [47]. На першому етапі обирається деяке проміжне значення $f(d,k)$ у графі обчислень криптографічного алгоритму, яке буде обчислене даним пристроєм, де d – відомі дані (відкритий текст чи шифр текст), які можна довільно змінити, маніпулюючи входом пристрою, k - елемент ключа.

На другому етапі здійснюють вимірювання СП пристрою у процесі зашифрування чи розшифрування D різних блоків даних. Для кожного з цих зашифрувань чи розшифрувань аналітику необхідно знати відповідні значення d, які при цьому утворюються та беруть участь у обчисленні $f(d,k)$. Позначимо такі відомі значення як $\bar{d} = (d_1,\ldots,d_D)^T$, де d_i - відомі дані у i -му зашифруванні чи розшифруванні.

Протягом кожного з таких зашифрувань чи розшифрувань аналітик записує сигнал СП – так звану «трасу». Кожна траса відповідає певному блоку d_i та позначається як $\bar{t}_1 = (t_{i,1},\ldots,t_{i,T})$, де T - довжина траси у відліках. Тому набір трас можна подати у вигляді матриці $|W|$ розміру $D \times T$. Зауважимо, що для DPA атаки є суттєвим правильне вирівнювання трас у часі: виміряні значення СП у кожній колонці t_j матриці $|W|$ повинні спричинятися однією і тією ж операцією.

На третьому етапі обчислюють очікувані проміжні значення для кожного можливого елементу ключа k_j вектора $\bar{k} = (k_1,\ldots,k_K)$, де K – загальна кількість комбінацій варіантів вибору різних k_j. Такі k_j називають гіпотезами ключа. Тоді аналітик обчислює матрицю $|V|$ усіх можливих варіантів проміжних значень $v_{i,j} = f(d_i,k_j)$, де $i = 1,\ldots,D$, $j = 1,\ldots,K$. Так як $\bar{k}$ містить усі можливі варіанти елемента ключа, то метою аналітика є визначення індексу стовпця $|V|$, який відповідає елементу ключа, який був використаний при обробці $\bar{\mathbf{d}}$.

На четвертому етапі проводиться відображення $|V|$ у матрицю $|H|$ очікуваних значень СП: $v_{i,j} \to h_{i,j}$. Для цього аналітик використовує відомості про модель СП, яка характерна для досліджуваного пристрою. Найбільш розповсюдженими є моделі на основі ваги Хемінга, відстані Хемінга та, так звані, нуль-моделі.

На останньому етапі кожен стовпець h_i порівнюють із кожним стовпцем t_j, тобто порівнюють очікувані значення СП для кожної гіпотези елементу ключа із фактично отриманими трасами для деякого елементу ключа. Результатом такого порівняння є матриця $|R|$ кореляційних коефіцієнтів $r_{i,j}$, де $i = 1,\ldots,K$, $j = 1,\ldots,T$. Індекси найбільших коефіцієнтів матриці $|R|$ вказують на індекси елементу ключа, який був використаний пристроєм при обробці $\bar{d}$.

ДОДАТОК В

ВІДОМОСТІ ЩОДО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

«ЗАТВЕРДЖУЮ»
Проректор з наукової роботи
Тернопільського національного
економічного університету
проф. Задорожний З.-М. В.
«___» _____ 2015 р.

АКТ

Про використання результатів дисертаційної роботи аспіранта кафедри комп'ютерних наук Коркішко Лесі Мирославівни «Методи та засоби маскованої арифметики для пристроїв систем захисту інформації» у науково-дослідних роботах на теми: «Паралельні методи та засоби реалізації алгоритмів захисту інформації в комп'ютерних мережах з використанням математичного апарату еліптичних кривих» та «Методи та засоби реалізації алгоритмів захисту інформації стійких до атак на реалізацію»

Ми, комісія у складі зав. кафедри комп'ютерної інженерії д.т.н., професора Березького Олега Миколайовича, наукового керівника науково-дослідної роботи, д.т.н., проф. Карпінського Миколи Петровича та завідувача відділу організації науково-дослідних робіт та маркетингу Науково-дослідного інституту інноваційного розвитку та державотворення ТНЕУ Письменного В.І., створена для приймання робіт, виконаних на теми «Паралельні методи та засоби реалізації алгоритмів захисту інформації в комп'ютерних мережах з використанням математичного апарату еліптичних кривих» (Державний реєстраційний номер 0109U000035) та «Методи та засоби реалізації алгоритмів захисту інформації стійких до атак на реалізацію»(Державний реєстраційний номер 0105U008181), встановила:

1. Розроблені Коркішко Л.М. методи виконання операцій кон'юнкції та диз'юнкції над даними у маскованому представленні, які, на відміну від існуючих, є масштабованими до кількості використаних логічних масок та дозволяють підвищити захищеність комп'ютерних компонентів до атак на основі аналізу споживаної потужності.
2. Запропоновано Коркішко Л.М. метод перетворення маскованого представлення даних на базі операції додавання маскованих даних за модулем 2^N, який, на відміну від існуючих, володіє низькою місткісною складністю та дозволяє підвищити захищеність комп'ютерних компонентів до атак на основі аналізу споживаної потужності.
3. Удосконалено Коркішко Л.М. метод обернення даних у маскованому представленні у скінчених полях виду $GF(2^N)$, який, на відміну від існуючих, є масштабований до кількості використаних логічних масок.

Завідувач кафедри комп'ютерної інженерії, ТНЕУ
д.т.н., професор — Березький О.М.

Науковий керівник
науково-дослідних робіт,
д.т.н., професор — Карпінський М.П.

Завідувач відділу організації
науково-дослідних робіт та маркетингу
Науково-дослідного інституту
інноваційного розвитку
та державотворення ТНЕУ — Письменний В.І.

SAMSUNG ADVANCED INSTITUTE OF TECHNOLOGY
P.O.BOX 111 SUWON 440-600 GYEONGGI-DO KOREA TEL : 82-31-280-9114 FAX : 82-31-280-9099, 82-2- 44-6217

DOI: 092635673739826487592

Date: 2011-01-11

CERTIFICATE

given to Lesya Korkishko to confirm usage of her results obtained during execution of project "Methods and structures of masked arithmetic for hardware and software implementation safe to power analysis attacks" in Samsung Advanced Institute of Technology (SAIT).

We proud to confirm that Lesya Korkishko has completed execution of the project "Methods and structures of masked arithmetic for hardware and software implementation safe to power analysis attacks", where following results were achieved:

1. Verilog models of hardware specialized processors for symmetric block ciphers according to algorithm mCrypton.
2. Verilog models of experimental hardware specialized processors for symmetric block ciphers according to algorithm GOST 28147-89.
3. First order differential power analysis attack simulation testbench using post-place-and-route models of the processors, back-annotated with standard parasitic coupling parameters.

Achieved results were used in research and development works of SAIT. Developed Verilog models and power analysis testbench were used to estimate potential resistance of real chips based on the developed processors models to first order differential power analysis attack. For reference we have used Samsung STD library 0.18um with standard cells.

According to results of analysis and simulations using leading industry-standard tools, we can state that that developed models of the masked data processors would allow us with high probability to produce chips for resource-constrained environments with resistance to first order differential power-analysis.

Head of Security Technology Lab (signature) Dr. Kang Seo-Duk, Ph.D.

Head of Hardware Security Group (signature) Dr. Jung Tae-Chul, Ph.D.

Leading the Next

University
of Bielsko-Biala

AKADEMIA TECHNICZNO-HUMANISTYCZNA
KATEDRA INFORMATYKI
43-309 Bielsko-Biała, ul. Willowa 2
tel. 33 82 79 350

"ZATWIERDZAM"
D. Sc. **Stanisław Andrzej Rajba**
Vice Chairman of Department of Computer Science and Engineering

Wasze pismo z dnia:	Znak:	Nasz znak:	Data:
		K20/ 81 /2015	30.06.2015

Z A Ś W I A D C Z E N I E
o wdrożeniu wyników pracy dysertacyjnej
doktorantki **Lesi Korkishko**
w dziedzinie nauk technicznych w grupie specjalności
"Informatyka, Technika Obliczeniowa i Automatyka"

Główne wyniki badań naukowych doktorantki Lesi Korkishko będące podstawą do uzyskania stopnia naukowego doktora nauk technicznych aprobowano i wdrożono w pracach naukowych Katedry Informatyki, a także zastosowano w procesie dydaktycznym przy prowadzeniu przedmiotu „Bezpieczeństwo technologii informatycznych", zgodnie z Umową o współpracy pomiędzy Akademią Techniczno-Humanistyczną w Bielsku-Białej (Polska) a Tarnopolskim Narodowym Uniwersytetem Ekonomicznym (Ukraina) z dnia 30.04.2015 r.

Zastosowanie rezultatów badań zawartych w rozprawie dysertacyjnej doktorantki Lesi Korkishko w realizacji wyżej wymienionego przedmiotu oraz w realizacji badań naukowych i metodologicznych Katedry Informatyki przyczynia się do poprawy jakości kształcenia wysokowykwalifikowanej kadry.

Adiunkt Katedry Informatyki
dr **Nadiia Balyk**

AKADEMIA TECHNICZNO-HUMANISTYCZNA
KATEDRA INFORMATYKI
43-309 Bielsko-Biała, ul. Willowa 2
tel. 33 82 79 350

«ЗАТВЕРДЖУЮ»
Перший проректор
Тернопільського національного
економічного університету
Шинкарик М.І.
«13» 03 2015 р.

АКТ

Про використання результатів дисертаційної роботи аспіранта кафедри комп'ютерних наук Коркішко Лесі Мирославівни «Методи та засоби маскованої арифметики для пристроїв систем захисту інформації» у навчальному процесі кафедри комп'ютерних наук Тернопільського національного економічного університету

Комісія у складі декана факультету комп'ютерних інформаційних технологій, д.т.н., професора Дивака М.П., в.о. зав. кафедри комп'ютерних наук, к.т.н., доцента Пукаса Андрія Васильовича та доцента кафедри комп'ютерних наук, к.т.н., доцента Шевчука Руслана Петровича підтверджує, що результати кандидатської дисертації Коркішко Лесі Мирославівни впроваджені і використовуються в навчальному процесі при вивченні дисциплін: «Методи та засоби захисту програмного забезпечення» та «Безпека програм та даних» для студентів спеціальностей 6.050103 «Програмне забезпечення систем», 8.05010301 «Програмне забезпечення систем» та 8.05010302 «Інженерія програмного забезпечення», а саме:

1. Методи виконання арифметичних та логічних операцій для комп'ютерних компонентів з підвищеною стійкістю до атак на основі аналізу споживаної потужності.
2. Дослідження структур операційних пристроїв та їх характеристик складності при виконанні арифметичних та логічних операцій у маскованому представленні.
3. Експериментальні дослідження спеціалізованих апаратно-орієнтованих процесорів симетричного блокового шифрування даних у маскованому представленні.

Декан факультету комп'ютерних інформаційних технологій, д.т.н., професор — Дивак М.П.

В.о. зав. кафедри комп'ютерних наук, к.т.н., доцент — Пукас А.В.

Доцент кафедри комп'ютерних наук, к.т.н., доцент — Шевчук Р.П.

„ЗАТВЕРДЖУЮ”

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

_____ С.Ф. Дячук

„ 21 ” _____06_____ 2016 р.

АКТ ВПРОВАДЖЕННЯ

1. ***Об’єкт впровадження:*** методи та засоби маскованої арифметики для пристроїв систем захисту інформації.
2. ***Ким запропоновано, виконавці, адреса:*** Коркішко Леся Мирославівна, кафедра приладів та контрольно-вимірювальних систем Тернопільського національного технічного університету ім. І. Пулюя, 46001, м. Тернопіль, вул. Руська, 56.
3. ***Джерело інформації:*** матеріали дисертації Коркішко Лесі Мирославівни „Методи та засоби маскованої арифметики для пристроїв систем захисту інформації”, поданої на здобуття наукового ступеня кандидата технічних наук (спеціальність: 05.13.21 – системи захисту інформації).
4. ***Де впроваджено:*** на кафедрі кібербезпеки Тернопільського національного технічного університету ім. І. Пулюя, 46001, м. Тернопіль, вул. Руська, 56.
5. ***Термін впровадження:*** червень 2016 р.
6. ***Висновок по впровадженню:*** запропоновані методи та засоби маскованої арифметики для пристроїв систем захисту інформації використано в навчальному процесі на кафедрі кібербезпеки при проведенні лабораторних робіт з дисципліни «Захист інформації в інформаційно-комунікаційних системах» курсовому, дипломному проектуванні за навчальними планами з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем».

В.о. зав. кафедри кібербезпеки,
к.т.н., доцент Р.О. Козак