Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні»

КвРКІ. 2202139.22.02.38 ПЗ

Виконала: студентка 2 курсу, група КІ2м-22-2 _____ Олексій СМІРНОВ
Підпис Ім'я, прізвище

Керівник канд. техн. наук, доцент _____ Катерина БЕРЕЗЬКА
Науковий ступінь, вчене звання Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КІІС, д.т.н., проф.
Тетяна ГОВОРУЩЕНКО
29 05 2024 р.

Хмельницький, 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Комп'ютерної інженерії та інформаційних систем

Освітній рівень магістр

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

Освітня програма освітньо-наукова програма «Комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

" 01 " 09 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Смірнов Олексій Петрович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні

Керівник проекту (роботи) Березька К.М., к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.01.2024 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

аналіз відомих методів для криптографічного захисту від вразливостей в апаратному забезпеченні; моделювання процесу криптографічного захисту від вразливостей в апаратному забезпеченні; удосконалення методу криптографічного захисту від вразливостей в апаратному забезпеченні; засоби криптографічного захисту від вразливостей в апаратному забезпеченні.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІІС		

7. Дата видачі завдання «_01_»_____09_____2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2023	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2023	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2023	виконано
5	Робота над науковою статтею	01.02.204	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2024	виконано
7	Робота над розділом 4 – проектування засобів для вирішення поставленої задачі, експериментальна частина	01.04.204	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2024	виконано
9	Попередній захист КРМ	29.04.2024	виконано
10	Захист КРМ на засіданні ЕК	До 25.05.2024	

Студент _____ Олексій СМІРНОВ
Підпис Ініціали, прізвище

Керівник роботи _____ Катерина БЕРЕЗЬКА
Підпис Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: «Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні»

Автор роботи: Смірнов Олексій Петрович

Керівник роботи: Березька К. М.

Пояснювальна записка: 97 с., 5 рис., 1 табл., 2 дод., 81 джерело.

ШИФР, СТОРОННІ ЗНАКИ, ЗЛОВМИСНИК, КРИПТОГРАФІЯ, ВРАЗЛИВОСТІ, ГЕНЕТИЧНІ АЛГОРИТМИ.

Об’єктом дослідження є процес криптографічного захисту від вразливостей в апаратному забезпеченні.

Предметом дослідження є методи та засоби криптографічного захисту від вразливостей в апаратному забезпеченні.

Метою кваліфікаційної роботи магістра є покращення ефективності криптографічного захисту від вразливостей в апаратному забезпеченні.

Для розв’язання поставлених задач використовувалися методи криптографії, методи виявлення вразливостей, методи приховування сторонніх знаків.

Наукова новизна отриманих результатів:

- удосконалено метод криптографічного захисту від вразливостей в апаратному забезпеченні, в якому на відміну від відомих було розширено його межі застосування для внесених сторонніх знаків в текст.

На основі проведених досліджень розроблена архітектура і засоби виявлення та захисту від вразливостей в апаратному забезпеченні.

Практична значимість отриманих результатів полягає у розроблених засобах виявлення та захисту від вразливостей в апаратному забезпеченні.

У вступі подано об’єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо криптографічного захисту від вразливостей в апаратному забезпеченні.

У другому розділі здійснено дослідження предметної області та визначено

стратегію забезпечення криптографічного захисту від вразливостей в апаратному забезпеченні.

У третьому розділі розроблено спосіб криптографічного захисту від вразливостей в апаратному забезпеченні. Його реалізація базується на використанні апаратного пристрою. Також, розроблено метод криптографічного захисту від вразливостей в апаратному забезпеченні.

У четвертому розділі здійснено здійснено розроблення криптографічного захисту від вразливостей в апаратному забезпеченні.

У висновках підведено підсумки досягнення результатів з розв’язання завдань дослідження.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 5

ВСТУП 6

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ ВІД ВРАЗЛИВОСТЕЙ В АПАРАТНОМУ ЗАБЕЗПЕЧЕННІ 8

1.1 Огляд та поняття криптографічного захисту від вразливостей в апаратному забезпеченні 8

1.2 Відомі методи та засоби забезпечення захисту комп'ютерних пристроїв . 13

1.3 Постановка задачі 20

1.4 Висновки 21

2 МОДЕЛЬ ПРОЦЕСУ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ ВИЯВЛЕННЯ СТОРОННЬОГО КОДУ В ПРОГРАМНИХ МОДЕЛЯХ АПАРАТНИХ ПРИСТРОЇВ 22

2.1 Генетичні алгоритми як засіб виявлення стороннього коду в програмних моделях апаратних пристроїв 22

2.2 Модель процесу кодування станів на основі скінчених автоматів зі сторонніми знаками 28

2.3 Висновки 45

3 УДОСКОНАЛЕНИЙ МЕТОД КРИПТОГРАФІЧНОГО ЗАХИСТУ ВІД ВРАЗЛИВОСТЕЙ В АПАРАТНОМУ ЗАБЕЗПЕЧЕННІ 46

3.1 Основи методу криптографічного захисту від вразливостей в апаратному забезпеченні 46

3.2 Організація протидії текстовим атакам на шифри перестановок 53

3.3 Висновки 61

4 ПІДВИЩЕННЯ БЕЗПЕКИ ШИФРУ ЗА ДОПОМОГОЮ ДИНАМІЧНИХ МЕРЕЖ ПЕРЕСТАНОВОК З КЛЮЧЕМ 63

4.1 Вибір типу архітектури для підвищення безпеки шифру за допомогою динамічних мереж перестановок з ключем..... 63

4.2 Проєктування архітектури для мережевих шифрів підстановки-перестановки.. 70

4.3 Висновки..... 76

ВИСНОВКИ..... 77

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ..... 78

ДОДАТОК А ПРЕЗЕНТАЦІЯ РОБОТИ..... 87

ДОДАТОК Б НАУКОВА ПРАЦЯ ЗДОБУВАЧА……..94

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ГПС - граф переходу станів

АМ – апаратна модель

ІС - інтегральних схем

ОТ - обфускація тексту

ЗТ - заміщення в тексті

ГА - генетичний алгоритм

ОС - операційна система

ПЗ - програмне забезпечення

ВСТУП

Апаратне забезпечення піддається складним атакам. Не тільки програмна частина комп’ютерної системи зазнає впливів та інфікувань, але і певні складові частини апаратного забезпечення. Виробники апаратних засобів і пристроїв можуть виробляти їх з вкладеними в архітектуру частинами, які потім можуть використовувати із певною метою. Криптографічні алгоритми є безпечними в програмному забезпеченні і також захищені в апаратному забезпеченні. В апаратному забезпеченні можуть бути вставки, які можуть бути націлені на певні криптографічні алгоритми. Тому, потребують аналізу та розроблення методи протидії таким атакам із використанням нових стратегій.

Перспективним для розроблення є підхід з використанням генетичного алгоритму, який може бути застосовано для ефективного розв'язання складної задачі зіставлення підграфів, які можуть бути використані для подання архітектури апаратних засобів. Криптографічний алгоритм може бути зламаний за допомогою диференціальної атаки з відкритим текстом зі значно обмеженими ресурсами. Тому, щоб запобігти таким атакам, потрібно в розроблюваному методі враховувати контрзаходи не тільки для криптоалгоритму, але і для всіх мережевих шифрів підстановки-перестановки. Також, крім подання нових примітивів мереж взаємозв'язку, динамічних мереж маршрутизації, потрібно модифікуємо елементи, які вимагають прийняття рішень для зловмисника, щоб створити йому проблеми із виконанням зловмисних дій.

Актуальність роботи полягає в розробці методу і засобу криптографічного захисту від вразливостей в апаратному забезпеченні.

Метою кваліфікаційної роботи магістра є покращення ефективності криптографічного захисту від вразливостей в апаратному забезпеченні.

Поставлена мета досягається розв’язанням таких основних завдань:

- проаналізувати відомі методи криптографічного захисту від вразливостей в апаратному забезпеченні;

- розробити удосконалення методу криптографічного захисту від

вразливостей в апаратному забезпеченні;

- здійснити реалізацію розробленого методу криптографічного захисту від вразливостей в апаратному забезпеченні;

- здійснити есперименталні дослідження згідно розроблених рішень.

Об’єктом дослідження є процес криптографічного захисту від вразливостей в апаратному забезпеченні.

Предметом дослідження є методи та засоби криптографічного захисту від вразливостей в апаратному забезпеченні.

Наукова новизна отриманих результатів:

- удосконалено метод криптографічного захисту від вразливостей в апаратному забезпеченні, в якому на відміну від відомих було розширено його межі застосування для внесених сторонніх знаків в текст.

На основі проведених досліджень розроблена архітектура і засоби виявлення та захисту від вразливостей в апаратному забезпеченні.

Практична значимість отриманих результатів полягає у розроблених засобах виявлення та захисту від вразливостей в апаратному забезпеченні.

Для розв’язання поставлених задач використовувалися методи криптографії, методи виявлення вразливостей, методи приховування сторонніх знаків.

За темою кваліфікаційної роботи опубліковано одну публікацію [81] у Збірнику наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2023». (Хмельницький – 2023. – С. 278-279).

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ ТА ЗАСОБІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ ВІД ВРАЗЛИВОСТЕЙ В АПАРАТНОМУ ЗАБЕЗПЕЧЕННІ

1.1 Огляд та поняття криптографічного захисту від вразливостей в апаратному забезпеченні

Проектування апаратних засобів здійсеюється із застосуванням сучасних інформаційних технологій. По суті таке проектування розробляється програмно. І лише повністю відтестовані і промодельовані програмні моделі передаються на реалізацію з метою їх створення як апаратних пристроїв та засобів.

Автоматизація проектування до певної міри є самовизначальною. Вона включає інструменти та процеси, пов'язані з автоматизацією високорівневої архітектури аж до макета плати, готової до виготовлення. На найвищому рівні як правило розпочинають з графа переходу станів (ГПС) і за допомогою певного інструменту перевести його в апаратну модель (АМ). У зв'язку з цим, модель АМ може бути оброблена за допомогою таких інструментів, що дозволяє транслювати структури полів для перекладу в примітиви на рівні макета плати, де архітектура може бути спланована, маршрутизована і згенерована готовою до виготовлення як макет плати, враховуючи, що він формально перевірений за допомогою симуляції та тестування або інших програмних засобів.

Довіра до такої моделі апаратних пристроїв та засобів на сьогодні може бути нівельована через ризик внесення в неї програмним чином фрагменту, який в подальшому на апаратному рівні надаватиме доступ стороннім особам да цього пристрою чи засобу. До початку активного використання зловмисниками кіберсередовища замовники плат не дуже зосереджувались на такій проблемі. Але в поточний момент часу кіберризики зросли в усіх сферах використання інформаційних технологій і на усій етапах їх розроблення. Довіра протягом усього процесу створення апаратного пристрою до будь-якої конкретної системи або архітектури була високою, бо компанії припускали, що ліцензовані ядра інтелектуальної власності і готові компоненти за своєю суттю безпечні і не поставлять під загрозу довіру до даного архітектури. І навпаки, не тільки тому, що

процес проектування зазнає багато різних форматів і переходів з рук в руки для даної архітектури, але може включати зовнішні, здавалося б, надійні джерела, яких не вистачає загальній довірі.

Практика проектування інтегральних схем складається з безлічі різних етапів та тактик скорочення часу виходу на ринок і неповторюваних витрат на проектування при одночасному збереженні та виробництві високоякісних і надійних конструкцій чи схем. Таким чином, багато замовників можуть вирішити здійснити корпоративні придбання, щоб отримати перевірену інтелектуальну власність у існуючих розробників у сфері, на яку вони зараз не орієнтуються або не мають кваліфікованих інженерів. Аналогічним чином, часті випадки, коли компанії використовують готові компоненти від великих корпорацій або ліцензійні IP-ядра від проектних організацій третьої сторони.

Довіра до виробництва плат набуває актуальності. Особливо з розвитком систем з IoT, від яких залежатиме надійне функціонування, наприклад, систем в розумному будинку тощо. На сьогодні немає беззастережної довіри до виробничого процесу. Мається на увазі, що компанії, які вирішують перевести процес в офшор і передати процес на аутсорсинг, щоб належним чином обробляти макети архітектури або іншу конфіденційну інтелектуальну власність, не зловживаючи і не використовуючи їх поза узгодженими умовами виробництва. На основі таких проблем вже сформувалась стратегія «нульової довіри» [1, 2]. І навпаки, при розгляді цієї стратегії виявляється, що її метою є зміщення парадигми від одноразової перевірки до парадигми постійної [3]. Передумовою цієї ініціативи є як усунення довірених моделей, так і створення довіреного середовища. Тим не менш, за останні кілька місяців спостерігаємо перехід з метою кращого забезпечення виробництва та довіри в ланцюжку поставок мікросхем до забезпечення загальної зміни парадигми довіри до компонентів специфікацій плат.

В роботі не буде ставитись під сумнів етика або практика виробників. Це дійсно може бути ще однією точкою невдачі в забезпеченні довіри під час виробничого процесу. Наприклад, одноразове працевлаштування, яке передає розробки, може поставити під загрозу всю довіру і призвести до того, що проекти

стануть сумнівними з точки зору власності та порушення внутрішньої архітектури порівняно з еталонною. Концепція зрізу віддалено нагадує обчислювальні одиниці [4]. Можна побачити паралелі між різною архітектурою, які в кінцевому підсумку будемо використовувати в якості розгляду для експерименту. Оскільки кореляція не є причинно-наслідковим зв'язком, припустимо, що розробка з використанням ядер і додаткові продажі склали значну частину доходів, яку отримає компанія виробник. Але саме в частині ядра архітектури пристрою чи системи компанії виробники можуть також розпочати суперечку щодо інтелектуальної власності. Це буде друга причина через яку потрібно розробити метод, який би убезпечив витік інформації про внутрішню будову апаратного пристрою на основі його програмної моделі [5-9]. На сьогодні не існує жодного методу чи практики, для яких компанія могла б довести право власності. Таким чином, їй доведеться витратити як час, зусилля, так і ресурси на використання методів з реверс-інжинірингу архітектури апаратних пристроїв і визначення того, чи була включена унікальна частина її розробки в архітектуру пристроїв іншої компанії [10-12]. Не має методу або практики, за допомогою яких компанія могла б довести своє право власності, і при цьому не витративши значні витрати часу і ресурсів. Відсутність методів перевірки права власності, які були б включені в проекти, може бути пов'язана з різними причинами, наприклад, не існує простого методу, який би не впливав негативно на схему під час процесу проектування та життєвого циклу його розробки або метод не може бути легко інтегрований чи використаний з існуючими стандартними інструментами автоматизованого проектування.

Розглянемо опис моделі атак на апаратне забезпечення, Будемо класифікуючи ці атаки за трьома основними категоріями: час; впровадження несправностей; бічний канал. Також, описуємо стандартну модель атаки проти криптографічних алгоритмів. Для аналізу важливості і наслідків атак розглянемо ключові концепції інформаційної безпеки та криптографії [13-19]. Наявний стандарт визначає набір фундаментальних послуг, необхідних для інформаційної безпеки, які називаються тріадою CIA [13]: конфіденційність; цілісність; доступність. І навпаки, ця тріада полягала лише в тому, щоб вважати

фундаментальною структурою і для повної інкапсуляції інформаційної безпеки необхідні додаткові послуги: автентичність; незаперечення. В цілому ці сервіси складають основи інформаційної безпеки і є основними принципами, які повинні підтримуватися криптографічними алгоритмами. Однак це не єдині концепції, необхідні алгоритмам, що використовуються для криптографії. Так що основними методами, що використовуються для побудови будь-якого криптографічного алгоритму, є принципи Шеннона [15]: заплутуваня; змішування. Ці концепції ґрунтуються на оригінальному відкритому тексті секретного ключа і результуючого шифротексту. Плутанина полягає в тому, що текст повинен залежати від декількох частин, в той час як змішування використовується для забезпечення лавинного ефекту. Тобто, невеликі зміни в одному початковому тексті повинні істотно впливати на результуючий текст [16]. Найбільш поширеними способами виконання плутанини і змішування є використання транспозиції і заміщення. Тобто, обфускації тексту(ОТ) і заміщення в ньому (ЗТ). Ідея перестановки, або мережі, може полягати в тому, щоб просто зашифрувати початковий відкритий текст з полем підстановки, що замінює вихідні символи простого тексту деякими іншими значеннями [17].

Основна мета будь-якого криптографічного алгоритму полягає в тому, щоб використовувати певний задіяний процес, так що спроба атакувати або вплинути на цілісність даних була неможлива. У той час як ОТ і ЗТ є лише кількома способами досягти цього, інші алгоритми покладаються на математичні процеси для забезпечення інтенсивності обчислень і складності. Наприклад, криптосистема Рівеста-Шаміра-Адельмана [20] покладається на процес генерації двох різних простих чисел, для яких відкритий і закритий ключі обчислюються на основі цих простих чисел і арифметики за модулем. Існує багато різних архітектур, орієнтованих на ключові методи, які можуть реалізувати криптографічні алгоритми [21-26]. Щоб забезпечити краще розуміння пізніших концепцій і актуальності криптосистем, розглянемо основні поширені методи, що використовуються в сучасних криптографічних системах.

Спочатку розглянемо криптографічні методи на основі ключів [22]. Існує два

основних типи ключових методів: асиметричний; симетричний [22-24]. Асиметричний це секретний ключ для шифрування та для нього є окремий відкритий ключ для дешифрування. Симетричний ключ є єдиним секретним ключем для шифрування та дешифрування. Відомі приклади [25-32] криптографічних алгоритмів, що використовують ці ключові методи. Найбільш помітною відмінністю між цими двома ключовими методами є метод реалізації, такий, що асиметричні шифри представляють серію математичних операцій, що використовуються для виконання шифрування та дешифрування, тоді як шифри з симетричним ключем, як правило, можуть бути візуально представлені за допомогою раундів логічних операцій, що виконуються над даними для шифрування/дешифрування. Розглянемо цю різницю за допомогою алгоритму для RSA та архітектури для одного раунду DES [33-35]. При побудові шифру з симетричним ключем доступні різні методи реалізації, наприклад, одиночний раунд DES. Він використовує мережеву структуру. Таким чином, визначимо можливі структури шифрів із симетричним ключем: збалансований Фейстель; неврівноважений Фейстель; Лай-Массі [36]. Структура Фейстеля має дві основні відмінності між збалансованим і неврівноваженим варіантами, яка полягає в тому, що в ній параметри мають або рівні або різні довжини.

Однією з головних переваг використання структури Фейстеля, будь-яких варіантів, збалансованої, незбалансованої або файстелеподібної, є те, що функція не обов'язково повинна бути оберненою. І навпаки, інвертність залишається на розсуд розробника шифру, наприклад, AES використовує обернені лінійні перетворення, тоді як функція розширення DES не є інвертованою [37].

Структура мережі заміщення-перестановки на відміну від мережевої роботи Фейстела, спирається на інвертовність, тобто адитивні операції з вихідними/інвертованими мережами перестановки та заміщення, засновані на шифруванні/дешифруванні [38-41].

Мережева структура Лай-Массі є прикладом фейстелеподібної схеми, яка була розроблена спеціально для криптографічного алгоритму IDEA [42]. Великою відмінністю між схемою Лая-Мессі та структурою Фейстеля є використання

оберненої функції, але при цьому вона має перевагу перед мережею заміщення-перестановки, де функція не обов'язково має бути інвертованим.

На додаток до різних архітектурних реалізацій, існують також різні конфігурації [43-54], для яких можуть працювати криптографічні алгоритми, в яких в архітектурі використовують так звану блокову конфігурацію. При цьому блок даних фіксованої довжини для даного повідомлення або шифрується, або розшифровується. І навпаки, іншою конфігурацією [55], яку може реалізувати архітектура, є потік, де окремі цифри повідомлення шифруються цифрами з псевдовипадкового потоку ключів, які можуть бути згенеровані за допомогою таких компонентів, як регістр зсуву з лінійним зворотним зв'язком [56].

Таким чином, в архітектурі апаратних пристроїв і засобів може бути присутня зайва функція, яку було реалізовано виробниками навмисно з подальшим її використанням із зловмисною метою. Також, різні виробники можуть недоброчесно використовувати готові апаратні рішення, які отримують з програмних моделей. Їх перевірка забирає багато часу. Тому, ці дві проблеми вимагають розроблення методів для забезпечення швидкої перевірки програмних моделей апаратних засобів або розробки методів, які б базувались на шифруванні інтелектуальних частин програмних моделей, які б забезпечували захист від такий додаткових функцій або втрати інформації про архітектуру нових апаратних пристроїв. Розглянуті криптоалгоритми мають свої особливості і можуть бути використані для вирішення цієї проблеми.

1.2 Відомі методи та засоби забезпечення захисту комп’ютерних пристроїв

Розглянемо атаки, які можуть бути виконані неруйнівним способом [57]. Приймемо для розгляду цю модель з припущенням, що здійсненність атаки більша не тільки з точки зору економічної ефективності, але й завдяки кількості доступних добре задокументованих методів [58]. Ці типи атак можуть бути зосереджені на зборі інформації про систему шляхом спостереження за її виконанням або синхронізацією кешу, коли атака на кеш дозволяє зловмисникам розкрити рядки

даних кешу. Наприклад, AES використовує пошук таблиць, що зберігаються в кеші, які залежать від використовуваного ключа шифрування, таким чином, зловмисник може спробувати завантажити варіації ключів, щоб визначити, які набори бітів є в ключі, через час попадання кешу або промаху, спричиненого завантаженням відповідних таблиць. Цей тип атаки полягає в тому, що зловмисник намагається використовувати якісь засоби для активного порушення цілісності даних, які зараз обробляються в системі. При розгляді апаратної реалізації це може бути досягнуто за допомогою ряду способів, які в іншому випадку впливають на значення транзистора або шини. Одним із таких способів може бути зниження напруги в системі, оскільки транзистори не працюватимуть при заданих напругах і видаватимуть неправильні значення. Розглянемо попередні приклади інверторів, що використовують технологію, яка працює від джерела живлення. При подачі дуже малої напруги живлення інвертор не працює належним чином і, таким чином, призводить до несправності в системі [57].

Атака типу бічний канал [58] використовує методи зондування для вимірювання та збору інформації про систему за допомогою різних засобів та пристроїв. Наприклад, одним із методів, який зазвичай використовується, є аналіз диференціальної потужності, який досліджує потужність системи за різних вхідних умов. Якщо розглядати інвертор, то досліджуючи потужність при конкретному значенні, можемо отримати знання про те, що таке конкретний вхід у будь-який момент часу. При зміні вхідної напруги система буде видавати помітно різні значення. Тобто, суть атаки типу бічний канал в тому, що вплив здійснюють опосередковано [59].

Криптоаналітичні атаки згідно їх моделі спираються не на апаратні або фізичні методи бічного каналу для збору інформації, а на ретельний аналіз самого криптографічного алгоритму, шифротексту або інших методів з метою виявлення слабких місць системи та отримання використовуваного секретного ключа. Крім того, визначаємо також основні типи генералізованих криптоаналітичних атак [60]: тільки шифротекст (зловмисник має доступ лише до набору шифротекстів); відомий відкритий текст (зловмисник має доступ до наборів як відкритих текстів,

так і відповідних шифротекстів); вибраний відкритий текст (зловмисник вибирає випадкові відкриті тексти для шифрування, щоб отримати відповідні шифротексти); адаптивний обраний відкритий текст (зловмисник може вільно адаптувати введення відкритого тексту на основі раніше отриманих шифротекстів з вибраних відкритих текстів); обраний зашифрований текст (зловмисник збирає інформацію, вибираючи зашифрований текст, щоб отримати як ключ, так і відкритий текст); обраний ключ (зловмисник вибирає значення ключа, щоб отримати інформацію про процес перетворення відкритого тексту в зашифрований); метод грубої сили (зловмисник намагається отримати відкритий текст заданого зашифрованого тексту, вичерпно досліджуючи всі можливі ключі); диференціальний підхід (зловмисник намагається знайти статистичну кореляцію між значеннями ключів і перетвореннями шифру, використовуючи визначений відкритий текст для отримання ключа); лінійний метод (зловмисник намагається знайти лінійне наближення до шифру для пари відкритого тексту та шифротексту, зводячи наближення до простішого, де ключ можна легко отримати); протокол націлений на протоколи безпеки, такі як аутентифікація, при цьому атаки, такі як повторне відтворення, дозволяють зловмисникам повторно надсилати дані або з потенційною затримкою з метою маскування під справжнього користувача [60]; пов'язаний ключ (отримання ключової інформації шляхом спостереження за безліччю невідомих, але відомо, що математично пов'язаних, секретних ключів, які використовуються для процесу шифрування); інтегральний метод (метод, що використовує набори обраних відкритих текстів з деякою фіксованою різницею, операції XOR; якась частина відкритого тексту залишається незмінною, а інша частина змінюється); слайд (метод атаки, який зводить нанівець актуальність функцій числового раунду, що виконуються над даними, шляхом наближення до раундів як добутку однакових перестановок); об'єктний метод (мета полягає в тому, щоб використовувати один і той же шифр з двома відкритими текстами з різницею лише в один раунд між ними, націлюючись на шифри, відомі як слабкі шрифти проти атак з відомим відкритим текстом [61-65]); зустріч посередині (зловмисник шифрує деякий відкритий текст одним ключем, а отриманий

зашифрований текст знову вторинним ключем; розшифровуючи другий шифрований текст за допомогою другого ключа і шифруючи відкритий текст за допомогою першого ключа, зловмисник може ефективніше перебирати шифр і формувати пари ключів-кандидатів, які можуть бути додатково протестовані на додаткових парах відкритий текст/шифрований текст [66]).

Таким чином, криптоаналітичні атаки згідно їх моделі спираються не на апаратні або фізичні методи бічного каналу для збору інформації, а на ретельний аналіз самого криптографічного алгоритму, шифротексту або інших методів з метою виявлення слабких місць системи та отримання використовуваного секретного ключа. Для розвитку цих методів використовуються активно генетичні алгоритми. Розглянемо їх в контексті поставленої проблеми.

Зростання підробок інтелектуальної власності та крадіжки інтелектуальної власності інтуїтивно породило потребу в методах, які дозволяють розробникам апаратних засобів перевіряти та ідентифікувати свої ядра у випадку, якщо вони підозрюють неправильне використання або крадіжку інтелектуальної власності. Процес розмітки послідовних схем, кінцевих автоматів широко розглянуто в [4-45], де в цих роботах використовуються стратегії для використання властивих характеристик пристроїв (стан, введення-виведення, крайове кодування та кодування стану). Однак найновіші сучасні системи [46, 51, 55, 59] намагаються використовувати методи, які вирішують проблему ізоморфізму підграфів, що виникає під час вбудовування сторонніх знаків. Найбільш помітним недоліком є те, що не існує ефективного розв'язку задачі і не було показано, що ні евристичні, ні апроксимаційні алгоритми не дають оптимальних рішень [34, 59, 62]. Це пов'язано з відомою обчислювальною складністю, оскільки відомо, що це одна з найбільш ранніх NP-повних задач. Тому, розглянемо гібридизований генетичний алгоритм для ефективного розв'язання цієї проблеми як з точки зору часу, так і якості отриманого рішення. Запропонованого підхід походить від численних змін, внесених до традиційного підходу ГА, які базуються на твердженнях, з [67], та спостережень, які регулярно здійснюються в природі. Точніше, традиційні ГА нав'язують концепцію біологічного кросинговеру, тоді як тут можемо реалізувати

добре відомий біологічний метод квадрата для домінантних і рецесивних генів, щоб забезпечити більш природний механізм кросинговеру. Крім того, в ГА нехтують концепцією стійкості, рівня народжуваності дітей, спостережуваних альфа/омега-анімалістичних еліт, безлічі мутацій, зумовлених географічними міркуваннями, елітарністю щодо популяції, міграційними моделями споріднених видів одного роду, стохастичною мінливістю, багатоцільовою та пристосованістю, заснованою на мультископічному рівні, де поодинокі генетичні мутації можуть впливати на тривалість життя членів. Як це видно в медичній сфері, через поодинокі мутації ферментів в організмі людини [68-70].

Застосовуємо цей підхід до усталеної системи сторонніх знаків на основі кодування стану, яка демонструє значні покращення в порівнянні з відомими підходами. Із середньою економією або прийнятним допуском з точки зору площі, затримки та літералів, необхідних для реалізації сторонніх знаків. Ці результати показують, що цей підхід можна вважати ефективним рішенням проблеми, яка виникає в процесі вбудовування сторонніх знаків [71-73].

Найновішим методом для виявлення сторонніх знаків, який використовує підхід є метод з [74]. Цей метод використовує схему сторонніх знаків на основі країв. Техніка на основі країв використовує як невизначені, так і існуючі краї для виконання техніки сторонніх знаків на основі вводу/виводу. Крім того, показано, що він [74] перевершує як методи встановлення сторонніх знаків на основі стану, так і вводу-виводу [79].

Вперше представлений [79] метод сторонніх знаків на основі станів використовує додаткові стани для реалізації прихованої поведінки, доступ до якої можуть отримати лише ті, хто має доступ до секретного ключа. Додаткові стани, як правило, проявляються у вигляді декількох підсистем, таким чином, оригінальна частина може бути відтворена з модифікованою поведінкою і використовувати секретний ключ для свого ввімкнення.

Техніка вперше представлена в [75] використовує розширений шлях станів, що містять невизначені комбінації введення-виведення, в яких сигнатура переставляється. Використання цих вільних країв має на меті збереження

цілісності системи та секретності підписів, таким чином, незаперечення підтримується за допомогою ациклічного таємного шляху, який відповідає підпису автора. І навпаки, основним недоліком цієї системи є те, що коли система є повністю визначеною, то до системи необхідно додавати додаткові вхідні біти. Це дозволяє генерувати невизначені комбінації вводу-виводу з початковою кількістю вхідних бітів, що призводить до того, що система несе значні додаткові витрати від невизначених комбінацій вводу-виводу.

Модифікація сторонніх знаків на основі вводу-виводу, подана в [80] і ця техніка спрямована на зниження додаткових витрат, що генеруються повторним використанням існуючих країв з комбінаціями вводу-виводу, які можуть бути відображені на деяку підсерію сигнатури.

Сторонні знаки на основі кодування стану подано фреймворком в [79], де спочатку попередньо обробляється бажана сигнатура за допомогою алгоритму хешування, а постобробка отриманого хеш-рядка виконується з метою перетворення хешу в спрямований граф на основі станів, побудованих з довжини кодування і суміжностей, інкапсульованих в перетравленому хеші. Цей метод застосовує послідовності сторонніх знаків як значення кодування станів, таким чином, послідовність станів може бути обійдена, щоб відтворити сторонній знак через існуючі або додані краї.

Робота з використанням методу на основі ГА подана в [72]. Запропонований підхід істотно відрізняється від методу з [21]. Це пов'язано з тим, що в [72] представляється ГА, яка виконує додаткові завдання, засновані на синтезі, такі як злиття та скорочення, тоді як ця робота є узагальненим підходом до вирішення вищезгаданої проблеми та представляє її можливості через використання існуючого фреймворку та застосування.

Розглянемо рівень впливу синтезу на сторонні знаки скінченних автоматів на основі кодування станів. Оскільки проектні корпорації з розробки інтегральних схем продовжують шукати найбільш фінансово оптимальні моделі та методи для постачання великих обсягів схем споживчого класу, одночасно зменшуючи як вартість неповторюваних кроків, так і час виходу на ринок, то дана конструкція

може змінюватися як у форматі, так і багато разів. Цим змінам сприяє лише постійний розвиток інструментів автоматизованого проектування, які дозволяють швидко створювати прототипи систем з поведінковими описами за допомогою мов опису апаратного забезпечення і синтезувати їх від високорівневого опису до макета плати, готової до виготовлення, за лічені хвилини. Подібна практика серед корпорацій, що займаються електронікою споживчого класу, будь-то наймання архітекторських фірм третьої сторони, закупівля готових компонентів або аутсорсинг виробництва, і все це в рамках зусиль з постійного скорочення витрат для конструкцій, є поширеною.

В даний час не існує галузевого методу або практики, за допомогою яких компанія могла б легко довести своє право власності, витративши значні витрати часу і ресурсів. Відсутність методів перевірки права власності, які були б включені в проекти, може бути пов'язана з різними причинами, наприклад, не існує простого методу, який би не впливав негативно на схему під час процесу проектування та життєвого циклу його розробки; або просто метод не може бути легко інтегрований або використаний з існуючими галузевими стандартними інструментами.

Таким чином, метою роботи є подальше уточнення та вивчення існуючого методу на основі стороннього знаку з кодування та його структури. Продовжуючи експериментувати з використанням цієї техніки володіння інтелектуальною власністю, метою є врахування, як зусилля та стратегії синтезу, що використовуються інструментами протягом життєвого циклу розробки та нанесення сторонніх знаків архітектури, можуть зрештою вплинути на остаточну схему після синтезу з сторонніми знаками. Використання техніки сторонніх знаків на основі кодування стану дозволить ефективно збирати відповідні дані. Ці дані згодом забезпечать візуалізацію того, як зусилля з синтезу протягом життєвого циклу розробки архітектури впливають на синтезовану схему після стороннього знаку.

Метод сторонніх знаків на основі кодування стану, описаний в [74, 76, 79], є технікою, яка використовує процес присвоєння стану для введення блоків стороннього знака як примусових значень коду стану, таким чином, що за

допомогою деякої серії вхідних послідовностей, відомих лише власнику, і деякого методу спостереження реєстру можна реконструювати оригінальну послідовність сторонніх знаків.

Структура та інструменти, що використовуються для виконання всього процесу нанесення сторонніх знаків подані в [76], причому кожен крок може бути коротко узагальнений.

Генетичні алгоритми зазвичай використовуються в задачах оптимізації, оскільки вони здатні генерувати якісні рішення і в значній мірі досліджувати весь простір рішень для даної проблеми. Оскільки ГА є еволюційними алгоритмами, то вони досягають цього шляхом включення концепцій, зосереджених на природному відборі, та використанні біологічних операторів: мутація; кросовер; селекція.

Таким чином, генетичні алгоритми можуть бути ефективно використані при розв’язанні проблеми, в якій потрібно дослідити можливу наявність стороннього коду в програмній моделі при проектуванні апаратного пристрою чи наявність власного коду схеми в розробці схеми в конкуруючій компанії.

1.3 Постановка задачі

Поставлена мета досягається розв’язанням таких основних завдань:

- проаналізувати відомі методи криптографічного захисту від вразливостей в апаратному забезпеченні;
- розробити удосконалення методу криптографічного захисту від вразливостей в апаратному забезпеченні;
- здійснити реалізацію розробленого методу криптографічного захисту від вразливостей в апаратному забезпеченні;
- здійснити еспериментальні дослідження згідно розроблених рішень.

1.4 Висновки

Проаналізовано відомі методи та засоби криптографічного захисту від вразливостей в апаратному забезпеченні, а також визначено стратегію для покращення ефективності цього процесу. Запропоновано використати генетичні алгоритми, оскільки вони здатні генерувати якісні рішення і в значній мірі досліджувати весь простір рішень для даної проблеми. Оскільки ГА є еволюційними алгоритмами, то вони досягають цього шляхом включення концепцій, зосереджених на природному відборі, та використанні біологічних операторів: мутація; кросовер; селекція. Встановлено, що генетичні алгоритми можуть бути ефективно використані при розв’язанні проблеми, в якій потрібно дослідити можливу наявність стороннього коду в програмній моделі при проектуванні апаратного пристрою чи наявність власного коду схеми в розробці схеми в конкуруючій компанії.

2 МОДЕЛЬ ПРОЦЕСУ НА ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ ВИЯВЛЕННЯ СТОРОННЬОГО КОДУ В ПРОГРАМНИХ МОДЕЛЯХ АПАРАТНИХ ПРИСТРОЇВ

2.1 Генетичні алгоритми як засіб виявлення стороннього коду в програмних моделях апаратних пристроїв

Параметри, які зазвичай використовуються в ГА: розмір популяції (простір розв'язків ГА або кількість окремих розв'язків); генерація (єдина ітерація, в якій використовуються біологічні оператори для природного відокремлення та поповнення популяції); коефіцієнт елітарності (частка особин з популяції, яка гарантовано виживе і перейде в наступне покоління, тобто не замінюється потомством); частота мутацій (ймовірність появи потомства, яке міститиме гени, які не походять ні від батьків A, ні від B); частота кросинговеру (ймовірність того, що обрані батьки схрещують гени і виробляють потомство); придатність (якісне представлення рішення, засноване на кількісному значенні, що випливає з цільової функції); ціль (бажане значення придатності (відстані до цілі) або рішення відносно задачі). Після цього функція ініціалізації популяції стохастично генерує (розмір популяції) індивідуальні рішення, які складуть весь набір членів генеральної сукупності, що використовуються в алгоритмі. Пристосованість - це кількісна величина, отримана з математичної функції, що має відношення до простору задач ГА. Функція сполучення складається з трьох етапів: елітарність; розмноження; мутація. Під час виконання функції елітарності частина популяції, визначена коефіцієнтом елітарності, буде гарантовано виживати в наступному поколінні, а решта членів популяції замінюються народженим потомством. Дочірня функція, як правило, є процесом до двох перемог, в якому двоє батьків народжують двох дітей, де кожна дитина має потенціал замінити кожного з батьків у наступному поколінні. При застосуванні кросовера діти створюються за допомогою кросоверної функції, і в найпростішій формі можуть бути представлені або в одноточковій, або в двоточковій, або в якійсь її варіації, де кожна з них відноситься до числа точок, в яких батьківська хромосома зрощується і гени

отримані від кожного з батьків. У тому випадку, якщо батьки А і В мають ймовірність кросовера меншу, ніж ймовірність кросовера, то потомство не народжується, і батьки просто доживають до наступного покоління. Для правильного підбору батьків використовуються додаткові схеми, які будуть використовуватися в процесі спарювання, отримання потомства, де їх шанси на спаровування пропорційні пристосованості. Слідом за створенням потомства кожен піддається випадковим мутації, заснованої на частоті мутацій, де одним з простих прийомів, є метод виконання випадкового обміну. Один ген вибирається випадковим чином і замінюється випадковим геном, відсутнім в хромосомі.

У той час як ГА базується на теорієї еволюції Дарвіна, то традиційна структура ГА не враховує спостереження в природі. Таким чином, запропонований метод значною мірою ґрунтується на цьому традиційному підході, реалізуючи різноманітні нові дарвінівські концепції та параметри, які визначимо наступним чином: неострівна багатопопуляція (множинні незалежні популяції споріднених видів отримують ту саму проблему; кросовер популяції (швидкість з якою дві популяції схрещуються, причому батьки належать до обох популяцій); плаваюча елітарність (поточний рівень елітизму, розрахований на кожне покоління, таким чином, що він базується на загальній пристосованості населення та адитивному рівні смертності); мікроскопічна елітарність (придатність розчину обчислюється на основі пристосованості окремих генів у хромосомі, що дозволяє домінувати та рецесивно розглядати гени); багатоцільовий фітнес (для оцінки рішень використовуються множинні фітнес-функції і їх не слід плутати з концепцією мікроскопічної або плаваючої елітарності); рівень смертності (адитивне значення до плаваючого елітизму для визначення кількості непридатних членів для повторного заселення); коефіцієнт народження дітей (верхня межа кількості дітей, яких двоє батьків можуть народити); розмір мутаційного гена (верхня межа кількості генів, які можуть мутувати під час однієї мутації); компенсація популяції (кількість поколінь до введення мігруючої популяції); тривалість життя (кількість поколінь, які використовується популяцією до її повного відновлення); популяційна мутація (коли всі члени популяції перевищують плаваюче значення

елітарності, тоді використовується фіксований коефіцієнт елітарності, спаровування пропускається, і вся популяція мутує до тих пір, поки не з'являться нові члени); максимальні мутації (верхня межа кількості мутацій, яких може зазнати один член популяції під час популяційної мутації); стохастичний дарвінівський підхід (кількість дітей, кількість мутацій, мутовані гени, батьківський відбір, успадкування рецесивних генів і популяційний кросовер визначаються кожним поколінням; це зроблено для того, щоб змоделювати теорії Дарвіна про мінливість, зумовлену природою, а не одомашненням, тобто впливом людини або деяким контролем над процесом).

Хоча концепція застосування багатоцільових фітнес-функцій до ГА існує, але застосування як мікроскопічних, так і макроскопічних елітизмів, не досліджувалося. Крім того, застосування частини з розглянутих параметрів до традиційних ГА ще не було. Спрощений розгляд процесу ГА та його компонентів ілюструє однопопуляційні та міжпопуляційні процеси спарювання, які можуть бути показані як єдиний процес, а популяційна мутація теж можлива. Єдиною відмінністю в одиничному і міжпопуляційному спарюванні є використання вторинної популяції для використання в процесі батьківського відбору, таким чином, щоб згрупувати процеси і зменшити повторення.

Визначаємо кодування проблеми як відображення станів у сторонньому знаку з станами в оригінальному алгоритмі. Де одна пара стану вважається геном, що утворює хромосому, членом популяції або потенційним рішенням. Практикою підвищення ефективності ГА є використання так званої острівної моделі, методу розпаралелювання для обробки субпопуляцій у межах ГА. Замість того, щоб використовувати цей існуючий підхід, ГА включає підходи щодо споріднених видів того ж роду та міграції нових мешканців. Таким чином, замість того, щоб використовувати острівну модель для обробки однієї популяції, концептуалізуємо і використовуємо модель, в якій початкова популяція сама по собі представляє цей острів, де через певний проміжок часу або поколінь унікальна популяція споріднених видів одного роду мігрує на нього.

Такий підхід збільшує зусилля по боротьбі з конвергенцією популяцій, коли

члени популяції з часом тяжіють до єдиного рішення. при цьому дозволяючи генерувати потенційно нові та унікальні рішення шляхом схрещування популяцій між мігруючими та острівними мешканцями. Завдяки використанню кроспопуляції можемо опосередковано включати концепції як варіацій через природу, так і врахування географічного розташування, тобто варіація від острівного до мігруючого виду шляхом спарювання і, хоча природня варіація залежить від наявності нового географічного місця. Для наочності розглянемо наступний приклад. Популяція вовків, що проживають від народження поблизу зони відчуження, очікувано має набагато більше відмінностей від зони, ніж ті, що живуть безпечніше за межами цієї території. Однак, якщо член популяції покидає цю зграю і переходить в інші, менш вразливі, зграї для спаровування, то існує потенціал для генетичної диверсифікації популяцій за межами зони. Ця концепція також була продемонстрована в експериментах, коли менші члени певної популяції, які шукали їжу, могли ненавмисно об'єднатися, коли альфи прокидалися і починали захищати шляхи між популяціями, запобігаючи поверненню.

Оцінка окремих генів дозволяє розглядати їх домінантним і рецесивним чином, забезпечуючи можливість використовувати квадратну методологію. Домінантним геном вважається ген, який відповідає цільовим критеріям проблеми, у цьому сценарії це картування, яке не потребує додавання країв, а рецесивними генами вважаються ті гени, які не відповідають цільовим критеріям. Таким чином, це за своєю суттю створює елітарність і пристосованість у мікроскопічному масштабі та слідує з медичних міркувань.

Прикладом цього є дослідження, які показують, що синдром пов'язаний з неспадковими мутаціями одного гена, зокрема гена зв'язуючого білка серйозно впливає на розвиток мозку. Мікроскопічно один ген, який зазнає мутації, може значною мірою вплинути на ціле, таким чином, замість того, щоб просто сказати, що еволюція в цілому непридатна, звужуємо фокус до уточнення, що ген непридатний, навіть якщо розчин в цілому представлений негативно, то він може складатися з набагато більшої кількості корисних генів.

Оскільки проблема стосується конкретно алгоритму ГА, то використовуємо

багатоцільову придатність, так що, подібно до того, як це було помічено з недоліками та методів сторонніх знаків на основі країв (генерація вхідних бітів), використовуваний засіб все ще може мати ті ж недоліки. Є сценарії, в яких додавання бітів неминуче. Таким чином, придатність оцінюється як з точки зору кількості як ребер, так і бітів, які потрібно додати, щоб вмістити сторонній знак.

Використання плаваючого елітизму має на меті пом'якшити заміну змішувань нижче середньої, так що з часом пристосованість усіх членів поступово зростатиме і падатиме, але врешті-решт зближуватиметься до оптимального рішення. Він керується значенням смертності, що є додатковим значенням до середньої пристосованості популяції, і натхненний концепцією швидкості навчання в нейронних мережах. І навпаки, популяційна мутація використовує фіксований елітизм і концептуально моделює соціально-економічний статус альфа-членів в анімалістичних ієрархіях, згідно з теорією дарвінівської концепції мінливості в природі. Таким чином, лише членам альфа-верств найвищого рівня будуть гарантовані права на все при змінах, зумовлених природою (зміни в екологічній стійкості).

Розглянемо кросовери. Процес схрещування популяцій досить простий, при розгляді декількох незалежних популяцій кожна популяція випадковим чином визначає, чи буде вона виконувати схрещування популяцій. При цьому, якщо розглянути сценарій, коли популяція А визначає, що вона повинна перехресно заселятися, то кожен член, повторно заселений в процесі спарювання, буде складатися з батька А з популяції А і батька В з популяції В, таким чином, кожен з батьків вибирається випадковим чином з кожної популяції. Аналогічно, під час процесу спарювання двох батьків випадковим чином вибираються гени, які обидва є рецесивними. У процесі спарювання народжується кілька дітей на основі: батьківських домінантних генів А і рецесивних В; батьківських домінантних генів В і рецесивних А; домінантних А з випадковими рецесивними; домінантних В з випадковими рецесивними. При цьому повертається дитина з найкращою фізичною підготовкою.

Процес мутації - це просто обмін, коли розміри їх однакові, або заміна, коли

один менший за другого. Спосіб заміщення є рандомізованим, таким чином, гени ітеративно обробляються, і випадковим чином вирішується, мутувати кожному гену чи ні.

Експериментальна установка може полягати в тому, що єдиною модифікацією, яка була б зроблена, можна примусово здійснити введення значень кодування станів (використання присвоєння стану). Для порівняння результатів використовуємо ті ж файли з набору тестів, довжини сторонніх знаків, що вбудовуються, і сценарії синтезу. Тим не менш, використовуватимемо різні довжини кодування, що неможливо в інших роботах. Результати можуть бути отримані для процесу нанесення сторонніх знаків, Інформація про еталонний набір позначає кількість станів, Продукти, що використовуються в процесі, спеціально відповідають підмножині, щоб забезпечити порівняння методів.

Інформація про сторонні знаки, що використовуються з точки зору мінімальних і максимальних значень стану є достатньою. Для наочності зменшуємо кількість сторонніх знаків, перелічених у міру того, як хеш-сигнатура була оброблена так, що кожен з них перетворюється з використанням довжин кодування. Таким чином, було досліджено варіації стороннього знака.

Параметри, які використовуватимуться для ГА в процесі експерименту такі, що можуть бути обрані методом проб і помилок. При цьому для найкращого відбору необхідно провести додатковий аналіз. Порівняння результатів синтезу запропонованого методу ГА з результатами буде ілюструвати зміни площі і затримки по відношенню до розмірів. В середньому буде спостерігатися скорочення як площі, так і затримки з сторонніми знаками, тоді як у деяких випадках буде спостерігатися помітно більша площа та затримка додаткових витрат. Це також може бути пов'язано зі способом додавання додаткових ребер і кодування необмежених станів, в тому сенсі, що перший доступний вхід використовується для доданого краю, і так само перше доступне кодування стану використовується для стану без сторонніх знаків, який повинен бути перепризначений. Необхідно буде використовувати подальші інструменти синтезу, щоб визначити, чи є ці методи причиною і чи можуть вони потенційно

бути зменшені за допомогою додаткових методів оптимізації.

В середньому запропонований метод дає кращі результати після синтезу, ніж сучасний метод сторонніх знаків на основі країв, який в даний час базується на введенні-виведенні, так і методи, засновані на стані. Крім того, додаткові витрати від методу призначення стану зменшуються з точки зору літералів, площі і затримки. Таким чином, цей метод ГА в поєднанні з фреймворком сторонніх знаків на основі кодування стану може бути високоефективним методом вбудовування сторонніх знаків в послідовні схеми. Розглядаючи використання сторонніх знаків над обфускацією, можна встановити, що дана система може просто використовуватися спільно з такою схемою. Інструменти синтезу, що можуть використовуватися, булуть лише для еквівалентності в порівнянні з існуючими, після того, як сторонній знак вбудований, отриманий результат може бути перетворений в еквівалентне моделювання мови опису апаратного забезпечення, а потім синтезований за допомогою стандартних інструментів, або оброблений відповідним чином для використання з такими схемами обфускації. Єдиною частиною процесу, яка дійсно заблокована, є картографування та вбудовування сторонніх знаків.

2.2 Модель процесу кодування станів на основі скінчених автоматів зі сторонніми знаками

Наявність сторонніх знаків скінчених автоматів на основі кодування станів має кращу ефективність ніж аналогічно змодельована сучасна методологія, яка використовує альтернативну схему сторонніх знаків. І навпаки, результати синтезу таких систем є кращими не в кожному випадку, і незрозумілі додаткові витрати для невеликої підмножини оціночних критеріїв. Крім того, хоча інтуїтивно відомо, що вибір використовуваних кодів станів може істотно вплинути на результати синтезу, які отримані для будь-якої даної конструкції, неясно, яка точна причина значних додаткових витрат. Зокрема, при розгляді синтезу та результатів зіставлення, отриманих як алгоритмом і методологія може мати сценарії, в яких будуть

отримані більш оптимальні результати та умови для вставки стороннього знаку, тобто кількість додаткових ребер або бітів для розміщення бажаного стороннього знаку при розгляді даної оцінки буде нижчою. Таким чином, потрібно визначити, що саме конкретно є причиною цих аномалій синтезу.

Розглянемо поняття про сторонні знаки на основі кодування стану. Хоча цей метод детально описаний, але він має особливість, в якій він перетворює хеш-сигнатуру стороннього знаку в граф переходу стану, який потім відображається за допомогою ізоморфізму підграфу і прийняття рішення про завершення стану, де коди станів синтезу будуть застосовуватися до кодів хеш-функції. Геш-функція накладається на вихідний стан оптимальним чином, а додавання необхідних бітів/ребер та часткове застосування кодів станів для бієктивного відображення станів з хешу та оригінального ГА є достатнім. Розмір і складність цього ГА залежить від обох значень, обраних для змінних. Загальна кількість бітів, оброблених з гешу для генерації ГА, наприклад, розмір даних буде обробляти тільки перші біти будь-якої геш-сигнатури. Кількість бітів, що складають стан у геш-функції, наприклад, довжина кодування в бітах розглядатиме всі біти як символьний стан. Після цього результуючий ГА обробляється за допомогою ізоморфізму підграфу та прийняття рішення про завершення циклу, щоб вставити ГА та забезпечити дотримання кодів станів для оцінювання. Це досягається шляхом розгляду гешу як запиту обчислення, в той час як алгоритм також виробляє необхідні місця, для яких ребра повинні бути додані, щоб уможливити циклічність кодів станів, тобто, одночасно вирішуючи як ізоморфізм підграфу, так і проблеми завершення.

Метод сторонніх знаків на основі країв працює так, що геш використовувався для побудови ГА. Послідовність перетравлюється на основі довжини введення-виведення, а потім розкладається на ребра в певній послідовності, щоб її можна було реконструювати. Замість побудови графа, метод має на меті побудувати серію ребер, придатних для повторного використання, які можуть бути відображені/вставлені в оціночний критерій, де сигнатура може бути відтворена за допомогою комбінацій вводу/виводу. Таким чином, цей метод

значною мірою покладається на відображення та вставку ребер/бітів для розміщення сторонніх знаків.

Потрібно забезпечити певний підхід до оцінювання результатів виконання методу. Щоб дослідити вплив методів на синтез, використаємо підмножину файлів для ГА, що перекриваються. Вони мають периферійну мобільність і існують комбінації введення-виведення, для яких поведінка не була явно заявлена. Такі ребра можуть бути додані до системи. І навпаки, решта з них обмежені сценаріями, в яких буде отримана будь-яка з бажаних переваг. При цьому уточнюється введення-виведення або знаходять ізоморфний розв'язок. Коли ці умови не спрацьовують, вся система повинна бути розширена принаймні на біт, щоб забезпечити необхідний простір для вставки країв і сторонніх знаків. Необхідна кількість бітів інтуїтивно визначається характеристиками і загальною кількістю ребер або комбінацій введення-виведення, які необхідно вставити.

Таким чином, на цьому етапі можемо розглянути зміну ребер/вхідних бітів для кожного з викидів між оціночним ГА та результуючим ГА з стороннім знаком, щоб краще зрозуміти, чи метод просто забезпечив неоптимальне рішення відображення та вставки ребер. Для ясності, значення можуть представляти загальне число переходів, коли всі умови не задіюються так що це задає розбіжності. Крім того, при розгляді значень вони позначають збільшення до показаних значень. Хоча інтуїтивно додавання ребер і бітів збільшило б додаткові витрати системи, можна побачити, що викид знайшов ідеальне ізоморфне рішення під час картографування сторонніх знаків, але повідомлені значення площі та затримки додаткових витрат значно перевищили допустимі. Таким чином, він зміг вмістити геш ГА і розміри коду стану в кінцевому підсумку не були збільшені під час процесу відображення та вбудовування сторонніх знаків. Це може означати лише те, що найбільш правдоподібна причина додаткових витрат пов'язана як з частково виконаними, так і з вільними значеннями стану кодування. Крім того, хоча відомо, що багато робіт вивчали вплив значень кодування станів на результуючий синтез, повинні переконатися, що це точна причина додаткових витрат. Спосіб, у який виконується вільне часткове кодування та додавання країв

не є змінним/недетермінованим процесом і проводиться з використанням класичної методології. Таким чином, є дуже мала ймовірність, що цей контрольований процес прийняття рішень у системі міг призвести до негативного впливу на результат синтезу. Крім того, ще одним важливим фактором є те, що метод отримання результатів синтезу як базового, так і післястороннього знака за допомогою ГА є надзвичайно розпливчастим і явно не такий, що було б найкраще використовувати при тиражуванні бібліотеки з попередньо сторонніми знаками. При цьому все, що знаходиться між ними, залишається для найкращого припущення і екстраполяції.

Таким чином, для встановлення базового методу синтезу несторонніх оціночних критеріїв, які будуть використовуватися для порівняння з тими, що наведені і результатів, отриманих при вивченні впливу вільного часткового присвоєння стану, використовуємо точну методологію. Сценарій синтезу, який використовували при генерації нової базової лінії, буде включати фактори, а саме: використання алгебраїчного скрипта; відображення з використанням бібліотек; використання інструменту кодування, оскільки він не дозволяє використовувати задані користувачем значення кодування стану. Конкретний перелік операцій, що виконуються в цьому сценарії враховує результуючі значення синтезу, отримані для кожної оцінки в термінах літералів, площі і затримки.

Спочатку наведемо порівняння результатів синтезу. Багато оціночних критеріїв при розгляді числа істотно не змінилися в розмірах станів. Однак слід зазначити, що хоча деякі відсотки можуть здатися великими, але вони є відносними до розміру оціночного критерію. Розглянемо порівняння додаткових витрат при розгляді результатів синтезу після внесення сторонніх знаків. Крім того, хоча могли б надати порівняння вихідного рівня з результатами, але цього не зроблено заради простору, бо результати так само покажуть більший і негативніший загальний результат методу. Бітний геш ГА довжиною кодування 2 призвів до значень синтезу, значно кращих, ніж згенерований базовий рівень. Таким чином, може виникнути питання, чому бітний геш ГА однакової довжини кодування не кращий. Для подальшого розгляду цього сценарію слід розглянути дві

характеристики: кількість бітів та ребер, доданих з сторонніми знаками, враховуючи дані; складність хешу ГА, який використовувався для встановлення сторонніх знаків як графік запиту. Враховуючи кількість доданих біт/ребер, видно, що обидва розміри давали ізоморфні збіги. Таким чином, повинні врахувати складність гешу ГА для кожного екземпляру. При розгляді цих характеристик можемо виділити топографічні відмінності між ними: петля; двонаправлені ребра. Однак, розглядаючи повністю зв'язуючий граф з графом, який не є таким, можна було б очікувати, що графік дасть кращі результати з більшою гнучкістю. І навпаки, повністю зв'язний граф дає значно менші додаткові витрати і навіть покращує результати базового синтезу. Вироблені геш-функції мають однакові символьні стани, один і той же ГА був з стороннім знаком за допомогою обох графіків, але з якоїсь причини додаткові витрати між ними мають більший діапазон для літералів, площі та затримки.

Таким чином, відображення, отримані для кожного з розмірів, були різними, в тому сенсі, що, хоча ізоморфні, кожен з них був зіставлений по-різному зі станами. Вільні і примусові стани кодів в ГА з стороннім знаком не були абсолютно однаковими в процесі синтезу. Оскільки метод є недетермінованим і в кінцевому підсумку залишений стохастичним процесом генерації рішень відображення, проте, після окремих запусків алгоритму сторонніх знаків, з метою створення невеликого діапазону потенційних стохастичних сторонніх знаків, єдиними рішеннями, що генеруються, є ті, що призводять до більш високих додаткових витрат на синтез. Однак, за допомогою транзитивних властивостей і теорії графів, які є ізоморфними, також матимемо ізоморфне відображення.

Видалення кратних ребер не вплине на розв'язок, так що відображення, може бути безпечно змінено, що в кінцевому підсумку зменшить додаткові витрати обох екземплярів сторонніх знаків.

Розглянемо сценарій додавання стороннього знака нової базової лінії з графіками запитів, таким чином, що замість станів, буде становий оцінювальний ГА. Істотна відмінність полягає в тому, що істотний стан для включення графіків запитів більше відсутній, тому повинні повторно запустити процес для отримання

нових результатів додаткових витрат синтезу, щоб провести порівняння впливу на базовий рівень, оскільки ізоморфізм більше не існує. Незважаючи на те, що додаткові витрати в даний час є значущими для обох випадків, повинні навіть якщо вони можуть бути, то базовий рівень в порівнянні з показником ГА, як правило, збільшив додаткові витрати. Додавання ребер і бітів збільшує додаткові витрати. Однак важливим фактором для розгляду є раніше згадані значення кодування вільного стану. Хоча вони, як правило, призначаються за принципом стратегії «першим прийшов першим обслуговується», можемо дослідити їх вплив на отримані додаткові витрати, що утворюються в результаті синтезу. Враховуючи, що базовий рядок має стани, а графіки запитів містять їх, тоді може дослідити справжній вплив вільного призначення на окремий стан. Таким чином, при наявності простору кодування, можливого для них і підмножини непримусових кодів станів, виконано синтез для кожного з цих значень стану. Для наочності наведемо перелік кодів, які застосовуються для конкретного рішення, що розглядається. Із зібраних даних видно, що не тільки сторонні знаки послідовних схем на основі кодування станів є багатометричною оптимізаційною задачею, але й потенційно недостатньо оптимізована структура ГА може призвести до кращих результатів синтезу. Більш строгий стандартизований метод оптимізації та синтезу оціночних критеріїв може не в ідеалі дати кращі загальні результати, але може тільки покращити заданий показник. Таким чином, вплив одного вільного призначення, коли ГА обмежений частково примусовою схемою кодування, то потенційно може зробити або порушити допуск прийнятних додаткових витрат. Зокрема, при зміні коду стану зменшуються додаткові витрати з точки зору літералів, площі і затримки. Однак ці результати також дозволили реалізувати такі недоліки, які існують. Наприклад, використання вільного часткового присвоєння стану та/або додавання країв у спосіб стратегії «першим прийшов першим обслуговується» виявляється шкідливим для процесу синтезу. Крім того, було встановлено, що стохастичний метод, не завжди може дати найбільш бажані результати для даного розміру, що призводить до вимоги використовувати класичну методології розгортки. І навпаки, через це існують потенційні експлойти

властивості транзитивного ізоморфізму, які можуть бути використані, що дозволяє нижчий розмір, який буде використовуватися навіть тоді, коли знайдено більш ідеальні результати при еквівалентній довжині кодування більшого розміру. Зокрема, якщо розглянемо транзитивні властивості графа та ізоморфізм сторонніх знаків при еквівалентних довжинах кодування для різних розмірів файлу, тоді можна використовувати менший розмір та маніпулювати кодами станів для забезпечення присвоєння коду стану стороннього знака вищого рішення.

Розглянемо вибір цифрового підпису. Деякі значущі, унікальні дані в форматах відео, звук, файл, зображення тощо, які вибираються власником ядра і будуть використовуватися для стороннього знаку цільового представлення ГА, призначені для захисту. Геш-функція переправлює цифровий підпис для подальшої обробки та перетворення в процесі додавання сторонніх знаків. Вона перетворює геш-сигнатури в граф запитів. Це перетворення геш-сигнатури на представлення ГА, засноване на параметрах конфігурації, яке можна запитати у цільового алгоритму ГА.

Гібридизований генетичний алгоритм виконує картування на основі ізоморфізму субграфа та гамільтоніанське завершення для відображення графу запиту в цільовий ГА.

Розглянемо картографування та завершення ГА. Він використовує інформацію про відображення та доповнення для виконання частково примусового присвоєння стану сторонніх знаків для підмножини цільових станів та необхідних ребер/бітів, які потрібно додати для реконструкції стороннього знаку, генеруючи результуючий ГА з стороннім знаком.

Сценарії синтезу, що стосуються інструменту синтезу, цільового ГА та значень призначення стану з сторонніми знаками, генеруються автоматично.

Інструмент синтезу використовує скрипти та пов'язані з ними файли з сторонніми знаками, створені для виконання фізичного синтезу та генерації результатів постсинтезу сторонніх знаків.

Хеш-функція спочатку намагатиметься обробити цифрові підписи в їх необробленому двійковому представленні, доводячи підхід до основного

результату, оскільки отриманий сторонній знак, побудований ГА, завжди був повністю пов'язаним. Оптимальним знайденим рішенням було використання стандартної геш-функції для переправлення цифрового підпису та перетворення отриманого геш-рядка у формат, придатний для алгоритму зіставлення. Важливо, щоб для використання була обрана відповідна геш-функція, вільна від атак, але не містила надлишку бітів. Таким чином, після розгляду функцій, що стосуються вразливостей геш-функцій, що містяться в бібліотеці, можна стандартизувати використання гешу на цьому етапі.

При розгляді атак на геш-функції є дві основні проблеми при їх використанні для додавання сторонніх знаків за допомогою кодування стану: попереднє зображення; колізія. Для даного гешу та функції ці атаки можуть бути визначені як з'ясування, який цифровий підпис зробив геш, і що виробляє даний геш. Крім того, враховуючи, що стандартизація геш-функції є спільною і тому доступною всім, то атака не повинна мати можливості: відтворити оригінальний підпис зі значень коду даного алгоритму; створити інший підпис зі значень коду стану; реконструювати цифровий підпис, який використовується у сторонніх знаках, щоб підтвердити право власності. Поки що не існує цілком правдоподібної методології атаки. Однак, було досягнуто значного прогресу, і оскільки ці рішення також стають більш досконалими, поточна геш-функція також може вимагати змін.

Перетворення хешу в ГА здійснюється після стиснення цифрового підпису геш-функцією. Отриманий геш-рядок додатково обробляється і перетворюється в граф запитів ГА представлення на основі параметрів, що налаштовуються: розмір і довжина кодування. Однак, перш ніж розглянути ці параметри, результуючий формат файлу, створений на цьому кроці, щоб забезпечити краще розуміння, буде першочерговим.

Формат файлу ГА, який отриманий, використовує формат файлу, отриманий з набору. Структура цього формату файлу розбита на дві частини: заголовок і тіло. Попереднє визначення необхідної інформації про послідовну схему алгоритму і визначення поведінки системи отримується на цьому етапі. Інформація, що міститься в заголовку, описує склад основного кроку алгоритму.

Оператори визначають поведінку на системному рівні для заданого символьного стану, заснованого на вхідному значенні в систему, диктуючи як вихідне значення, так і символьний наступний стан, в який система повинна перейти. Перший параметр, тобто довжина файлу, дозволяє використовувати геш-рядок, оскільки на основі методів він визначений прийнятним для обробки підблоків з біт геш-рядка. Таким чином, цей крок чи інструмент дозволяє отримати розмір файлу, який є прийнятним.

Другий параметр, тобто довжина кодування, дозволяє будувати символічні стани з обраної довжини файлу, при цьому переправлений рядок розбивається на двійкові блоки довжини, налаштованої на це значення довжини кодування. У випадку, якщо довжина кодування не є коренем обчислювального стандартного рівняння, тоді до останнього побудованого символьного стану додається значення нуля, поки довжина не стане правильною. Це гарантує, що при виході необхідної послідовності для реконструкції геш-сигнатури на виході значень стандартного регістру все одно буде видаватися правильний геш-підрядок налаштованої довжини файлу.

За допомогою цих двох параметрів можна контролювати щільність, або розрідженість, стороннього знака гешу ГА. Однак, він все ще базується на обох значеннях цих налаштованих параметрів щодо заданого геш-рядка. Нарешті, використовуючи блоки, побудовані з довжин кодування та файлу геш-рядка, ГА будується циклічно, щоб створити гамільтонівський цикл і метод безперервного відтворення значень сторонніх знаків у стандартному реєстрі. Крім того, при побудові гамільтонового циклу, який потрібно знайти/вставити в цільовий ГА, складність, пов'язана зі спробою знайти набір станів сторонніх знаків-кандидатів зловмисником, за своєю суттю збільшується та ускладнюється.

Вибравши цифровий підпис переправляємо його за допомогою функції, щоб отримати результуючий геш-рядок: 345ghbdbv67lbnbnb78ssvnnv9595g. Якщо розглядати можливі конфігуровані довжини файлу, то набір переправлюваних рядків для трансформування наведено в табл. 2.1. В той час як довжина файлу зазвичай не використовується в існуючому методі, то для ілюстрації використаємо

саме її.

Таблиця 2.1 – Шифр-таблиця

Текст	Шифр
5	345gh
2	bd
17	bv67lbnbnb78ssvnn
6	v9595g

Додатково для більшої стислості використовуємо довжину кодування. Це призводить до того, що рядок геш-файлу розбивається на блоки символьного стану. Ці блоки станів символів згодом обробляються циклічно кільцевим способом, і в тому порядку, в якому вони з'являються в геш-рядку для побудови списків суміжності станів. Зображення на рис. 2.1.

Використовуючи інформацію про список суміжностей для кожного символічного стану, можемо скласти представлення ГА. Ці дані представляють структуру графіку, зображену на рис. 2.2.

Після того, як рядок оброблений і зібрано достатньо даних для реалізації ГА, генерується файл для стороннього знака запиту, встановлюючи кількість бітів введення/виведення до довжин, відповідних зі значеннями стану скидання, який опускається, оскільки решта даних заголовку визначені відповідним чином. Файл, отриманий під час цього процесу, який представляє рис. 2.1 , подано на рис. 2.2.

Значення введення-виведення в ГА встановлюються спеціально для того, щоб збільшити можливість відображення цих станів запиту на цільові стани, оскільки умови вводу/виводу та поведінка для системи та ж, то сторонній знак може бути відновлений у регістрі стану. Знайшовши цей сторонній знак запиту ГА у вихідному цільовому запиту і змусивши ці ребра або прийняти поведінку цього знайдені в цільовій функції або явно не визначені.

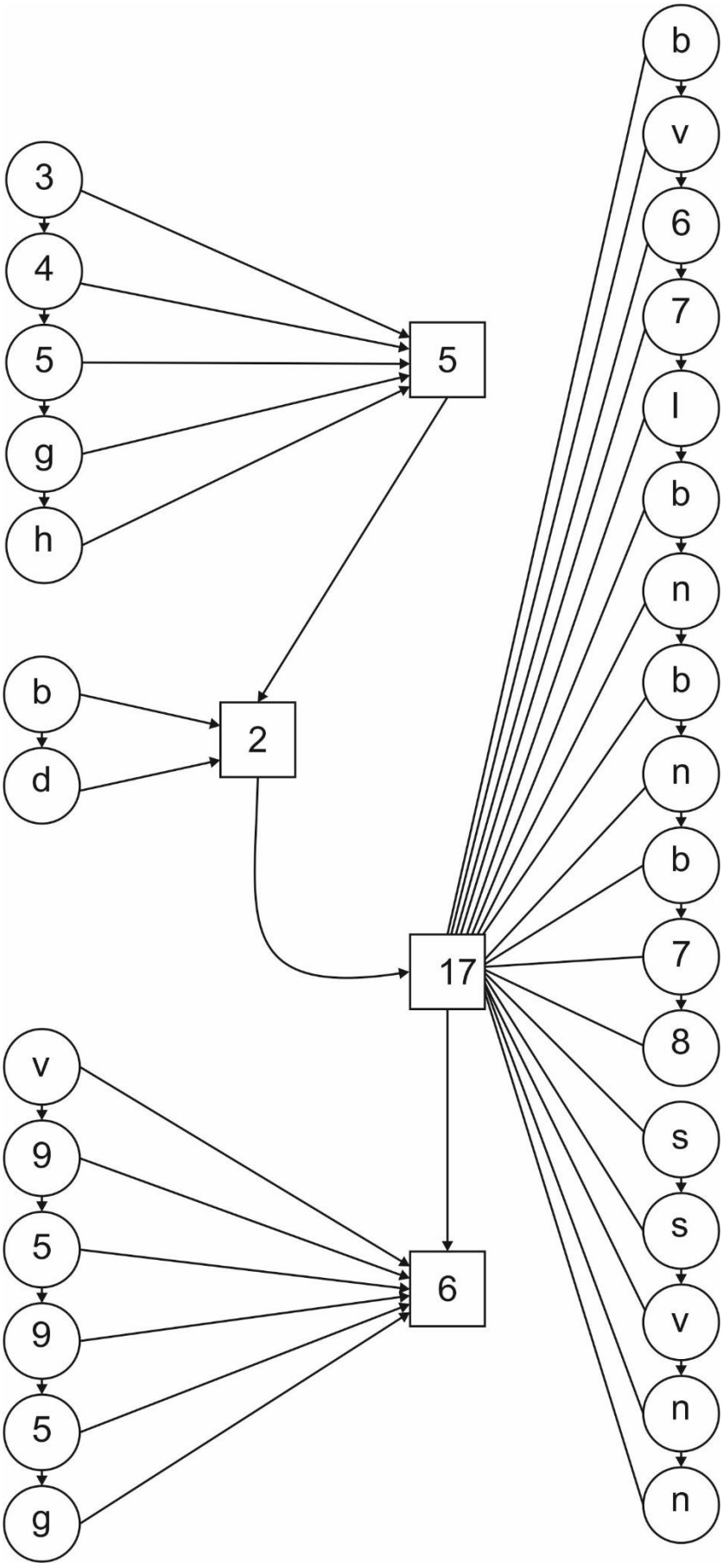


Рисунок 2.1 - Побудова станів символів і суміжностей

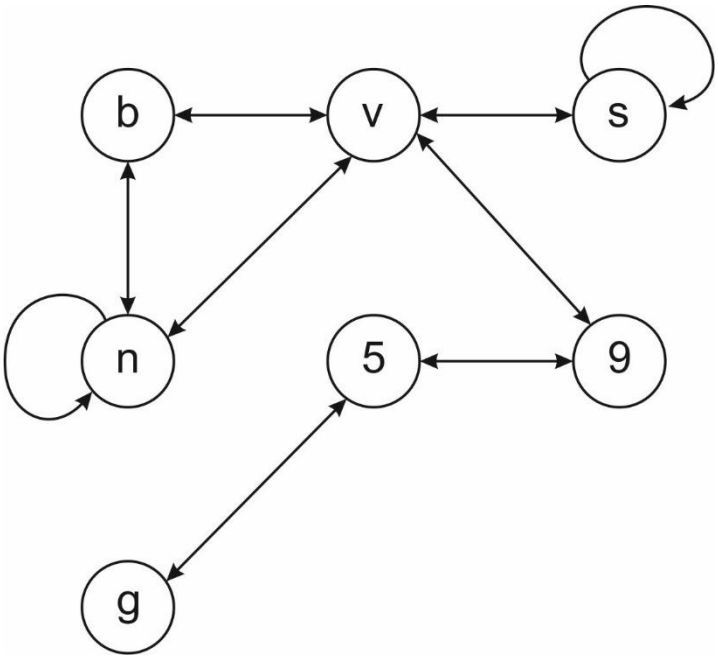


Рисунок 2.2 - Представлення графіка символьних станів та списку суміжностей

Розроблений алгоритм є гібридизованим ГА і частиною класу еволюційних алгоритмів (ЕА). в той час як типовий ГА заснований на дарвінівських принципах та включає концепції Дарвіна, а також додає багато нових концепцій і параметрів для кращого моделювання природного відбору, спарювання, мутації та процесу кросинговеру. Це основний алгоритм, що лежить в основі процесу зіставлення та доповнення між запитом і цільовою функцією. Незважаючи на те, що відомо, що схеми кодування станів впливають на синтез заданим чином, значення кодування одного стану має здатність повністю вилучити синтезовану систему з діапазону допустимих додаткових витрат. Проблема сторонніх знаків за допомогою коду тепер ще більше заглиблюється в багатометричну проблему оптимізації на багатьох фронтах що полягає в тому, який розмір, довжину кодування, вільні значення кодування та підпис вибрати. Кожен з них може негативно вплинути не тільки на складність графіку сторонніх знаків/запитів, але й на коди станів, що застосовуються, і на їх вільний вибір. Компроміси щодо продуктивності відомі, але є спосіб співіснування продуктивності та безпеки.

ГА — це ЕА, які моделюють природний відбір та еволюцію видів за допомогою спрощеної моделі, яка використовує оператори кросинговеру, спарювання та мутації, які використовуються на окремих членах популяції. Ці типи

алгоритмів, як правило, сконструйовані таким чином, щоб краще забезпечувати більш ефективні рішення складних оптимізаційних задач. Типова структура, якої дотримуються ці ГА, така, що придатність є цільовою функцією для оцінки членів генеральної сукупності, а ген позначає ітерацію алгоритму.

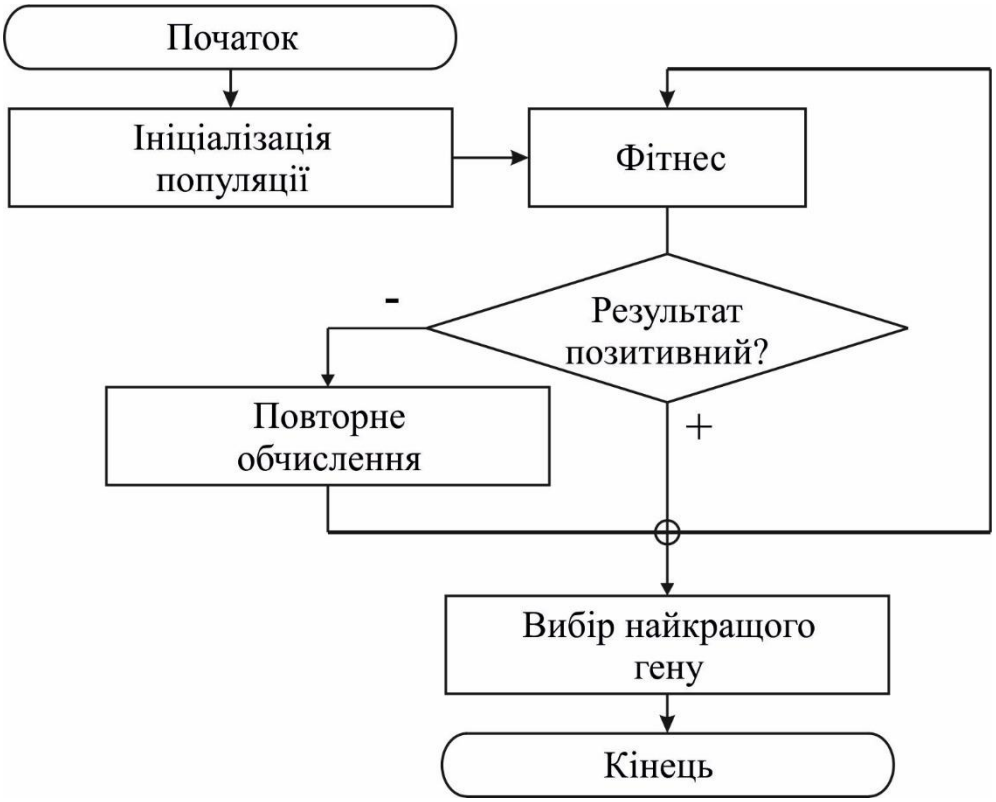


Рисунок 2.3 - Модель ГА

Параметри ГА, які зазвичай використовуються в ГА, і їх описи означають, що включення їх залишається за розробником програмного забезпечення, і хоча є загальним, та не завжди включаються.

Пристосованість обчислюється за допомогою деякої математичної функції, специфічної для задачі оптимізації, призначеної для якісної оцінки членів популяції; по суті, це «відстань від цілі» для члена популяції і, інтуїтивно, чим більша відстань від цілі, тим нижча якість індивіду. Наприклад, якщо хочемо реалізувати ГА для визначення змінних, то відомо, що ціль як індивід у популяції матиме «гени», і результуюче значення математичної функції для цих генів визначається як відстань від бажаної цілі, так і загальна якість особистості; це

також означає, що придатність відповідає введеному рівнянню.

Під час цього процесу ініціалізації створюються та випадковим чином ініціалізуються загальні члени, визначені параметром розміру генеральної сукупності. Процес випадкової ініціалізації знову визначається проблемою, що розв'язується, і додатковими перевірками або правилами, необхідними для забезпечення генерації лише дійсних рішень. Наприклад, якщо потрібно випадковим чином побудувати графові структури, то всі вузли повинні бути унікальними і пов'язаними.

У процесі спарювання відбувається основна частина операцій в ГА, до поточної популяції застосовується фіксована елітарність, використовуються оператори батьківського відбору, кросинговер батьківських генів для створення нових членів популяції і мутація. Процес спаровування може бути розширений і краще проілюстрований. Батьківський відбір може здійснюватися різними способами, в тому числі: рулетка, стохастичний універсальний, випадковий і ранговий. Це кілька відомих методів. Аналогічним чином, дочірні рішення або рішення для повторної популяції, створені з обраних батьків, генеруються на основі використовуваного оператора кросоверу. Там оператори, як правило, дотримуються методології перехрещення точок, в яких батьківські гени будуть розділені на групи з чергуванням груп, що складають дочірній розв'язок. Наприклад, якщо розглянемо установку, використану в попередньому прикладі з фітнесом, то геномний набір дає одноточковий кросовер, то можемо вибрати їх з першого і з другого наборів. Після цього визначається, мутувати ген у чи ні на основі параметру частоти мутацій, у випадку, якщо дочірній розчин мутує, то, як правило, значення одного гена в геномному наборі дочірнього розчину змінюється. Наприклад, якщо дочірній розчин має мінімальні параметри і вибрана мутація, то геномні значення тепер можуть бути досягнуті. У типовій моделі ГА батьки народжують лише двох дітей і вибирають найкращу дитину з розв'язків, щоб стати її членом, який потім знову увійде до популяції для оцінки та можливого спаровування в наступних поколіннях. Розширений процес спарювання моделі ГА на рис. 2.4.

Розглядаючи проблему сторонніх знаків, традиційна модель сама по собі не підходила. Це пов'язано з проблемами складності, які за своєю суттю виникають як при перетворенні гешу в функції, так і при відображенні запиту на цільовий параметр. Зокрема, було проблемно створити задачі гамільтонового завершення в графі. Гамільтоніанське завершення — це задача, заснована на рішеннях, для визначення мінімальної кількості ребер у графі, яку необхідно додати, щоб забезпечити існування гамільтонового циклу. Відомо, що ця задача має компутаційну складність недетермінованого многочлена повного, крім того, знаходження самого гамільтонового циклу має складність NP-повного. Відомо також, що задача має NP-повну складність. Таким чином, розглянемо модифікації, внесені в традиційну модель ГА на додаток до нових функцій, операторів тощо, які були використані.

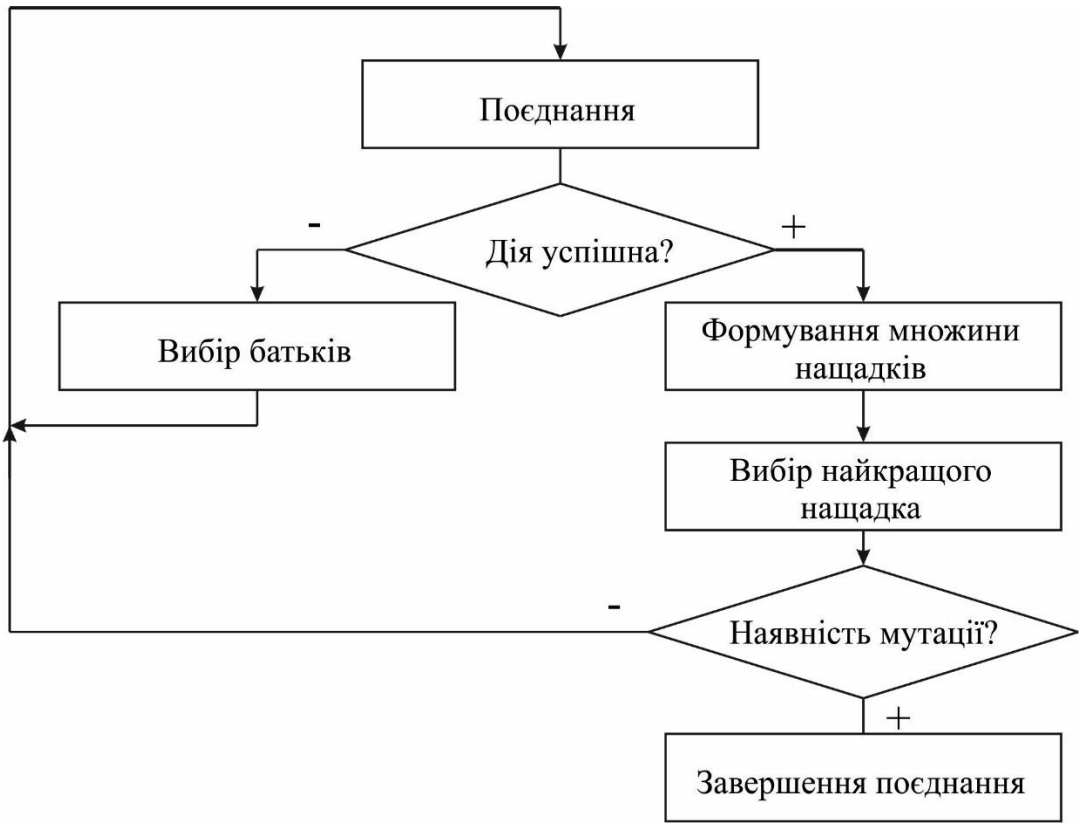


Рисунок 2.4 – Схема розширення процесу спарювання моделі ГА

Параметри дарвінівського генетичного алгоритму є достатніми для реалізації. На відміну від традиційної моделі ГА, націленої на проблему з однією

ціллю в техніці сторонніх знаків на основі кодування стану, немає простої задачі з одним пристосунком, тому мета зіставлення запиту з цільовим значенням включає розгляд як ребер, так і бітів. У випадку, якщо значення є повністю визначеним, тоді повинні додати біт до системи, щоб врахувати необхідні додаткові переваги. І навпаки, коли значення є неповністю визначеним, то треба мінімізувати кількість ребер доданих до системи, зазначаючи, що ця стратегія також застосовна до наступного кроку. Таким чином, замість традиційної «цілі» маємо «цільові краї» та «цільові біти» в новій моделі ГА, яку зображено алгоритмом на рис. 2.5.

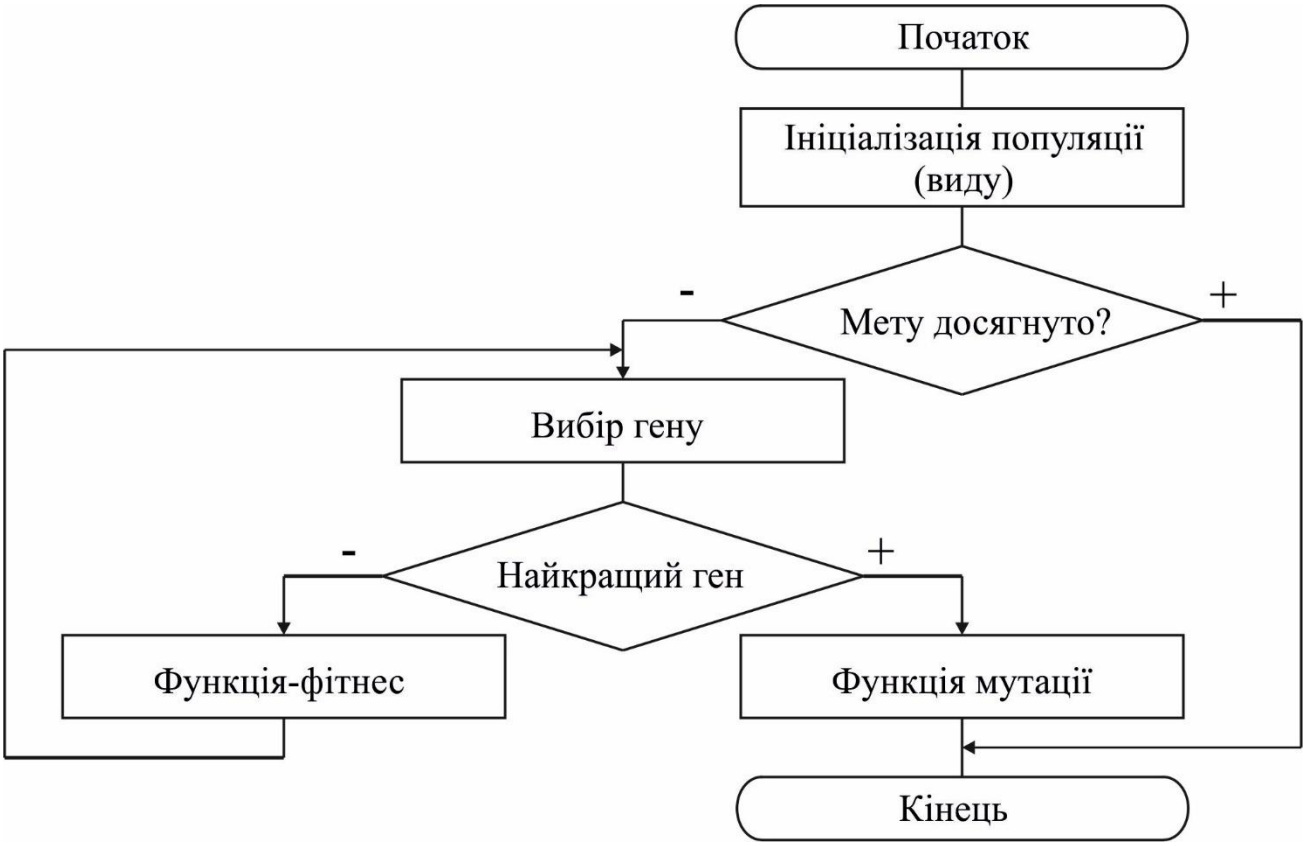


Рисунок 2.5 – Схема розширеної моделі ГА

Не розглядаємо окрему популяцію, яка працює над якоюсь проблемою, в загальній схемі ГА. Хоча існує «острівна» модель для традиційної ГА, але для паралельної обробки в ГА розглядаємо «острів» у фізичному сенсі до обробки. Наприклад, існують дві популяції незалежного розміру, що мають однакову структуру генерації, які можуть взаємодіяти шляхом схрещування популяцій.

Будемо вважати, що це не тільки для того, щоб точніше зобразити еволюційний процес, але й дозволлити те, що можна розглядати як концепцію мінливості Дарвіна, обумовлену природою, оскільки подібно до мутації, вказуємо ймовірнісне значення, в якому ці популяції можуть здійснювати спарювання.

Розглянемо, таким чином, компенсація популяції ГА та тривалість життя. Зсув є параметром і контролюючим фактором для визначення кількості поколінь, які мають відбутися до того, як вторинний мігруючий той самий рід споріднений вид буде введений до аборигену острова і почне взаємодію. Поки немає періодичності його міграції,

У тому, що після прибуття вони не йдуть, проте є тривалість життя. Хоча тривалість життя не застосовується і не відстежується для окремих членів популяції, то вона відстежується за поколіннями, оскільки кожна тривалість життя поколінь весь мігруючий або місцевий вид гине і повторно ініціалізується. Це допомагає забезпечити генетичне різноманіття та уникнути конвергенції до єдиного рішення, тобто популяція складається з дублікатів відповідних рішень, які були згенеровані.

Розглянемо в номвій моделі ГА плаваючу елітарність і рівень смертності. Замість того, щоб використовувати фіксовану схему елітарності, використовуємо «плаваючу» схему, яка базується на поточному поколінні на додаток до адитивного рівня смертності, еквівалентного з швидкістю навчання нейронних мереж. Цей адитивний рівень смертності служить для допомоги плаваючому елітизму у визначенні числа особин, визнаних «непридатними» в даному поколінні, відібраних для повторної популяції. Крім того, рівень смертності є параметром, який можна налаштувати, тоді як плаваючий елітизм є варіантом, заснованим на поколінні та популяції.

Тому, розглянемо додатково мікроскопічну елітарність і геномний кросинговер. При розгляді графових структур, пристосованості та членів популяції застосовуємо мікроскопічний елітизм, тоді як фіксований та плаваючий елітизм застосовується до популяції та її членів, мікроскопічний елітизм застосовується до члену та його геномних рішень. Оцінюючи придатність відображення однієї

вершини від запиту до цільового значення, таким чином, що кожна вершина має придатність, яка складає загальну придатність для індивіда з населення. Це дозволяє використовувати для виробництва дітей перехресний оператор квадратного типу. При цьому пристосовані батьківські гени домінують над рецесивними непридатними генами батьків під час створення потомства. У випадку, якщо будь-який даний ген двох батьків визнається придатним або непридатним, то ген вибирається випадковим чином з будь-якого з батьків. Однак це може спричинити ускладнення в дитячому поколінні, наприклад, якщо якийсь ген був уже визнаний придатним/непридатним для батьків, але гени для батьків є придатними/непридатними і містять значення, яке вже було обрано як ген придатності/непридатності, тоді матимемо дублікати генів і генеруватимемо недійсні рішення членів. Таким чином, при розгляді графіків або аналогічних обставин необхідно провести додаткові перевірки, щоб переконатися, що дуплікація генів не відбувається. Щоб впоратися з цим, потрібно розділити геномне призначення на двофазний процес: призначити відповідні гени, переконавшись, що вони не дублюються, випадковим чином встановити решту рецесивних генів з пулу генів, що залишилися (вершини графа). Це, знову ж таки, також сприяє створенню генетичного різноманіття та дослідженню загального простору рішень шляхом генерації потенційно недосліджених рішень.

2.3 Висновки

Таким чином, необхідно провести додаткові перевірки, щоб переконатися, що дуплікація генів не відбувається. Потрібно розділити геномне призначення на двофазний процес: призначити відповідні гени, переконавшись, що вони не дублюються, випадковим чином встановити решту рецесивних генів з пулу генів, що залишилися.

3 УДОСКОНАЛЕНИЙ МЕТОД КРИПТОГРАФІЧНОГО ЗАХИСТУ ВІД ВРАЗЛИВОСТЕЙ В АПАРАТНОМУ ЗАБЕЗПЕЧЕННІ

3.1 Основи методу криптографічного захисту від вразливостей в апаратному забезпеченні

Так само, як і в житті, види не просто народжують двох дітей або одного з них. Представники деяких видів можуть ніколи не розмножуватися, деякі мають менше десяти нащадків, деякі з десятками або сотнями за одне народження. Таким чином, замість того, щоб контролювати кількість дітей, які є у батьків, обмежуємо його верхньою межею, параметром максимальної кількості дітей, який дозволяє батькам потенційно народити цю верхню межу дітей для подальшого дослідження простору рішень і відбору найкращих для виживання в наступному поколінні як новому члену популяції.

Оскільки кросинговер домінантних парних і рецесивних парних генів є рандомізованим, то між двома заданими ітераціями створення дитини для виживання можуть бути відібрані абсолютно унікальні діти. Після того, як всі діти створені, з усіх дітей вибирається найкращий для нинішніх батьків. Кожна група батьків для кожного покоління має кількість дітей, яку вони будуть випадково згенеровані та обмежені параметром максимуму кількості дітей. В міру рандомізації значення стає можливим, що батьки народжують нуль дітей, а згодом просто вибирається інша група батьків і процес продовжується.

Максимальні мутації та розмір мутаційного гена є важливими параметрами для аналізу. Подібно до більшості дітей, припускаємо, що як кількість мутацій, так і мутованих генів обмежені поодинокими імовірнісними випадками. Наприклад, найпростішим сценарієм, який можемо представити, є сценарій, де з географічною консиденцією знаємо, що дика природа в зоні відчуження постійно перебуває під загрозою або активно піддається впливу різних рівнів радіації. Відома причина клітинних і генетичних мутацій. Таким чином, замість того, щоб припускати, що мутація є однією клітиною та екземпляром, встановлюємо верхню межу як максимальної кількості мутацій, яку може зазнати індивідуум, так і максимальної

кількості генів, які можуть мутувати під час однієї мутації.

І навпаки, якщо продовжимо припускати модель багатопопуляційного острову з мігруючими спорідненими видами, то виявимо мінливість, яку обумовлену природою, і покажемо модель тієї, що показана на рис. 2.4 в алгоритмі. В цілому, процес мутацій, який використовується ГА, найкраще можна проілюструвати на рис. 2.4.

Процес популяційної мутації є процесом, який дає підстави для розгляду стійкості навколишнього середовища, ієрархічного статусу, а також здатності особин і видів адаптуватися за певних обставин. Наприклад, експеримент показав, що коли вид не турбується про загрозу або екологічну стійкість, то це суттєво впливає на поведінку виду. Таким чином, коли плаваючий елітизм виявляє, що всі члени популяції розглядаються як придатні, то вважаємо це нестійкою популяцією, і при використанні фіксованого елітизму решта популяції повинна або адаптуватися через мутацію, поки не буде визначено через плаваючий елітизм, що члени стали непридатними і для повторної популяції може відбутися спарювання.

Процес спарювання застосовується, коли визначається, що «місцевий» і «мігруючий» види не будуть спаровуватися між популяціями, і навпаки, «спаровування» — це процес, який використовується, коли визначається, що «місцевий» і «мігруючий» види будуть спаровуватися один з одним. Ці процеси можна проілюструвати на схемі, де єдиною відмінністю між «спарюванням» є включення «мігруючих» видів для розгляду під час батьківської селекції, тобто з кожного виду буде відібрано по одному.

Щоб проілюструвати вихідні дані ГА, наведемо такий приклад. Коли перевести структури у формат і передати їх ГА, то отримаємо відображення, яке потім можна проілюструвати у вигляді графіка. Для ясності важливо відзначити, що станам без мітки ще не присвоєно значення кодування, тои будемо називати такі стани «вільними», тоді як ті, що вже позначені, є «примусовими» значеннями кодування для фізичного синтезу. Присвоєння цих значень вільного кодування є одним з небагатьох процесів, контрольованих рішеннями в ГА, оскільки весь процес відображення є максимально стохастичним, а зовнішній вплив

максимально усунений. Однак, спосіб, у який присвоюємо ці значення, є дуже простим, принципово простим підходом і просто використовує послідовне присвоєння у порядку черги. Це той самий підхід, який використовуємо для крайової вставки, коли алгоритм може використовувати явно невизначену поведінку для заданих вхідних комбінацій; додавання префіксів бітів, вставлених країв та оригінальних країв для збереження функціональності. Якщо розглядати цей процес для результуючого графіку, то почнемо цей процес з початкового впорядкування цільового графіка. Починаючи з цілі на результуючому графіку, знаходимо перший «вільний» стан, і враховуючи, що видалили декілька доступних значень, то повинні присвоїти їх стану. Таким чином, значення кодування верхнього лівого вузла є лише символічним для фізичного синтезу. Цей процес повторюється до тих пір, поки всім вільним станам не буде присвоєно значення кодування. Сценарії синтезу, згенеровані інструментом ГА, в даний час націлені на інструмент синтезу, оскільки він був фундаментальним.

Однак, це гнучке рішення і може бути змінено з мінімальними зусиллями, оскільки створено інструмент і дозволено генерувати сценарії, а не функції, які можуть бути використані такими стандартними інструментами. Це дозволяє наносити сторонні знаки, інтегрувати архітектуру як є або в більшу систему, а також маскувати макет з мінімальними зусиллями та часом.

Сценарій не виконує жодної оптимізації архітектури стороннього знака, щоб запобігти видаленню станів і зберегти примусові значення кодування стану стороннього знака, це досягається за допомогою інструменту призначення станів з опцією кодування користувача та файлом кодів станів, зіставленням символьних кодів з фізичними значеннями, пов'язаними з ним. Крім того, конструкція зіставляється з фізичними компонентами з бібліотек з отриманими значеннями синтезованої площі і затримки, обчисленими щодо цих фізичних компонентів, що використовуються в проектуванні.

Метод вивчення впливу зусиль синтезу на розвідувальний простір для нанесення сторонніх знаків за допомогою сторонніх знаків на основі кодування станів та його вплив на отримані результати синтезу після сторонніх знаків має

дати точний результат з невеликими відхиленнями. Тому, встановлюємо різний рівень зусиль синтезу за допомогою інструменту синтезу і скриптів, що використовуються на цільових значеннях з попередніми сторонніми знаками. У цьому це значною мірою зводиться до доступних попередньо визначених вихідних скриптів синтезу та команд, що мають відношення до оптимізації. Виконуємо синтез при різних рівнях зусиль без або мінімальних, низьких, середніх, високих, використовуючи підкреслену підмножину еталонних функцій ГА. Крім того, визначаємо, що мається на увазі під рівнями зусиль синтезу, які перераховано для націлювання на ці цільові значення і сторонні знаки.

Для стислості у звітності про дані, при цьому охоплюючи відповідні консиденції, підмножина використовуваних функцій базується на їх загальній кількості станів і на тому, чи є вони рекурсивними. Набір охоплює малі, середні та великі значення, що містять одночасно різні рівні синтезу для забезпечення ретельного вивчення впливу на зусилля синтезу до сторонніх знаків та його впливу на синтез після сторонніх знаків.

Далі визначаємо засоби синтезу, які будуть використовуватися для заданих процесів мінімізації стану, призначення стану тощо, варіанти, специфічні для інструментів. У той час як цільова функція надає варіанти для мінімізації та евристики відображення, не вказуємо алгоритми, які будуть використовуватися, окрім стандартних, оскільки при вивченні їх впливу навіть на найбільші у підмножині еталонних показників, не було помічено змін між результуючою схемою ГА при використанні іншого алгоритму евристичної мінімізації або відображення. Аналогічно, додаткова функція може націлюватися на багато різних критеріїв оптимізації, але для простоти досліджуємо лише присвоєння станів без сторонніх знаків за допомогою комбінації домінуючих алгоритмів введення-виведення. Далі, в описі попереднього рівня зусиль, це означає, що всі попередні оптимізації синтезу мають місце. Зокрема, і мінімізація стану, присвоєння стану, алгебраїчна та булева оптимізація. Включаємо параметри, які використовуються з ГА при нанесенні сторонніх знаків на ці новостворені цільові значення.

Результати синтезу для попередньої підмножини еталонних файлів при

різних рівнях синтезу є достатніми. Крім того, повні результати синтезу після стороннього знака для підмножини при використанні сторонніх знаків ГА при довжині кодування для кожного з розмірів файлу і на кожному відповідному рівні синтезу теж є достатніми.

Тому, надмірні додаткові витрати з точки зору площі або затримки можуть бути просто пов'язані з одним неоптимальним значенням стану через раніше згадану техніку «вільного» призначення стану в процесі ГА. Розглянемо ці результати і припустимо лише, що додаткові витрати або їх відсутність безпосередньо пов'язані з тим, як призначаються «вільні» стани, коли не можемо точно визначити чітку причину розбіжностей.

Очікувана тенденція збільшення зусиль синтезу буде в ідеалі меншою площею, затримкою для схеми, якщо розглянемо результати, то можна побачити, що інтуїтивно між збільшенням рівня зусиль синтезу деякі показники конструкції збільшуються, а не зменшуються. Наприклад, розглянемо взаємозв'язок між низькими і середніми зусиллями синтезу, природно, можна очікувати, що високе зусилля синтезу ще більше оптимізує і зменшить результуючу схему, однак добре видно, що і площа збільшуються при збереженні затримки. Подібна поведінка також проявляється і з протилежними значеннями, але, навпаки, і при цьому зберігає очікувану тенденцію до збільшення оптимізації та зниження пов'язаних з цим витрат на продуктивність.

Алгоритм має задає багато станів перед будь-якою формою синтезу та оптимізації на основі впорядкування рівня синтезуючих зусиль.

Результати після нанесення сторонніх знаків при нульовому рівні зусиль синтезу можна побачити, що конструкція рідко виграє від такого рівня зусиль у порівнянні з аналогом до нанесення сторонніх знаків. Тим не менш, можу бути мінімум один варіант, що є допустимим, який отримає перевагу під час цього процесу з точки зору компромісів між показниками продуктивності, і коли і біт, і край були додані під час додавання сторонніх знаків. Це зменшує затримку при одночасному збільшенні площі. Цей сценарій сам по собі є досить несподіваним, оскільки функція просто виконає присвоєння стану, повертаючи логічну

реалізацію, доступну для оптимізації, а метод після сторонніх знаків не робить нічого, крім виконання присвоєння стану з примусовими кодами станів, створеними за допомогою ГА. У той час як додавання ребер і використання невизначених комбінацій входів з неважливими умовами виведення теоретично може принести користь за рахунок вилучення та використання їх за допомогою методів оптимізації, оптимізація не виконується. Таким чином, повинні вважати за необхідне віднести це до відображення синтезу, створеного ГА у поєднанні з корисними кодуваннями вільного стану, а не з якимось значущим впливом відсутності зусиль на синтез попередніх сторонніх знаків на синтез після сторонніх знаків.

Це стає більш очевидним, якщо врахувати низькі зусилля синтезу, коли знову спостерігаються лише незначні збільшення площі зі зменшенням затримки, що можна пояснити зусиллями синтезу перед стороннім знаком, оскільки обидва випадки додають трохи і краю системі. І навпаки, при розгляді значного зменшення затримки, при цьому найменший приріст площі серед інших пробігів є постійним. Подібна тенденція продовжується, коли розглядати як середні, так і високі зусилля синтезу, де для цього малого значення специфічний пробіг сторонніх знаків для будь-якого заданого рівня синтезуючих зусиль створювали затримку схеми більш оптимальну, ніж затримка значення без сторонніх знаків.

У кожному тексті, розмірі стороннього знака для всіх заданих рівнів синтезуючих зусиль видно, що кількість бітів і ребер, доданих до системи при заданій довжині кодування, не змінюється. Однак у кожній підмножині простір дослідження синтезу для архітектури за своєю суттю був розширений і давав результати впливу в обох напрямках спектру, тобто додаткові витрати та зменшення, при розгляді показників продуктивності. Архітектура зазнала негативного впливу після внесення сторонніх знаків в певних випадках. В наборі значень наявна значна різниця між відсутністю зусиль синтезу і простим використанням низьких зусиль для даних. І навпаки, дані, для ГА при низьких зусиллях синтезу або без них, зберегли переваги від оптимізації та синтезу перед нанесенням сторонніх знаків, що прирівнює результати синтезу після сторонніх

знаків, більше схожі на те, щоб просто не виконувати будь-яку попередню оптимізацію взагалі. Однак, більша частина додаткових витрат, імовірно, може бути пов'язана з кількістю бітів і ребер, доданих до системи, які залишаються необробленими / використаними в синтезі після внесення сторонніх знаків і багато випадків зі сторонніми знаками призводили до перевищення фактичної кількості ребер. Їх додано до системи з мінімальною кількістю ребер, які вимагаються, з бітного тексту зі сторонніми знаками. Незважаючи на це, навіть найоптимальніші результати синтезу після внесення сторонніх знаків для ГА при розгляді велико бітного файлу більш ніж удвічі перевищують кількість бітів, площі та затримки.

При розгляді представленого методу і результатів в порівнянні з існуючими, не існує послідовного або ефективного способу відтворення використовуваних сценаріїв синтезу, а є тільки можливість екстраполювати явне використання таких засобів, як алгебраїчне письмо. Однак, у будь-якому випадку, можемо встановити, що зміна і використання додаткових скриптів, не принесли користі цим конструкціям протягом усього процесу нанесення сторонніх знаків. І навпаки, у випадку тексту зусилля з синтезу виявилися корисними і з точки зору підтримки нижчих результатів після синтезу сторонніх знаків, а також розширення простору пошуку рішень для синтезу, оскільки надані рішення для сторонніх знаків потенційно призвели до того, що певні показники продуктивності перевершили інструменти та методи цільової оптимізації архітектури.

З отриманих результатів можна сказати, що вплив початкових зусиль синтезу нового ГА на отримані результати синтезу після внесення сторонніх знаків в значній мірі варіює, як позитивно, так і негативно впливаючи на схеми. Тим не менш, він розширює простір синтезуючого розчину і може допомогти в тісному утриманні додаткових витрат на схему, навіть виявляється корисним для отриманого процесу синтезу після внесення сторонніх знаків. І навпаки, ці результати зібрані та проаналізовані показали б, що було б корисніше спрямувати зусилля синтезу після сторонніх знаків із застосуванням методів збереження сторонніх знаків. Наприклад, зміну методу присвоєння стану «вільний» у новій схемі ГА слід або залишити для подальших дій користувача, наприклад,

застосування вторинної схеми кодування до підмножини вільного стану або більш схожу на оптимальний розклад. При цьому нова схема ГА має посилання на оптимальне присвоєння коду стану і може стабільно відповідати на основі примусового кодування сторонніх знаків. У будь-якому випадку, для того, щоб ця техніка або інші подібні до неї стали більш бажаними для використання в якості галузевої практики, необхідно вдосконалити і інтегрувати їх з галузевими стандартними рішеннями. Це може означати, що дослідження цієї техніки у форматі з використанням синтезу буде використано для подальшого розгляду результатів синтезу до та після внесення сторонніх знаків для більш модернізованих інструментів синтезу.

Таким чином, вплив початкових зусиль синтезу нового ГА на отримані результати синтезу після внесення сторонніх знаків в значній мірі варіює, як позитивно, так і негативно впливаючи на схеми. Він розширює простір синтезуючого розчину і може допомогти в тісному утриманні додаткових витрат на схему і навіть виявляється корисним для отриманого процесу синтезу після внесення сторонніх знаків.

3.2 Організація протидії текстовим атакам на шифри перестановок

У міру того, як технології інтегральних схем постійно розвиваються, спостерігається тенденція, як багато комп'ютерних пристроїв стають широко доступними для громадськості, стимулюючи цікавість та інновації для багатьох завдяки швидкому створенню, розгортанню або навіть експлуатації апаратних систем. Аналогічним чином, технологія і можливості їх також значною мірою зросли протягом десятиліть. Наприклад, іграшки, автомобілі, гаражні ворота, різні господарські машини, які зараз оснащуються вбудованими системами та можливостями бездротового зв'язку для покращення життя кінцевого користувача. Однак у міру того, як технології продовжували розвиватися, зростала і велика кількість інформації, яка ставала легкодоступною майже для всіх у всьому світі, причому зі швидким доступом. В результаті для багатьох людей без попередніх

знань або досвіду стало можливим адекватно отримати навички, необхідні для проектування та впровадження апаратних систем, які можуть покращувати або використовувати повсякденні технології. Незважаючи на те, що пристрої, які сприяють цьому Інтернету речей (IoT), були спрямовані на покращення якості життя та сприяння революції розумного будинку. З тих пір вони стали звичайною мішенню для багатьох хакерів апаратного забезпечення. При цьому домашні пристрої стають мішенню для виявлення вразливостей, покращення або просто перепрофілювання. Хоча може бути зрозуміло, що багато з цих домашніх пристроїв не обов'язково потребують належних механізмів доступу та контролю. Очікувано, що пристрої, які в іншому випадку призначені для контролю доступу, будуть реалізовувати певну форму безпечного протоколу зв'язку. Однак веб-сайти стали звичайним явищем для хакерів спрямованих на обладнання, щоб змінити експлуатацію цих пристроїв IoT, які в іншому випадку повинні захистити будинки від інших впливів. Хоча це можна розглядати як невинне поширення знань та освітнього контенту, його негативна сторона викриває загальну відсутність турботи про безпеку з боку виробників пристроїв та існуючу сприйнятливість систем до технологій, що постійно розвиваються. І це дозволяє людям зі зловмисними намірами легко створювати зловмисні системи, які можуть бути використані для особистої вигоди.

Розглянемо удосконалення методу проектування для включення ключ-залежних мереж заміщення і перестановки в криптографічних алгоритмах і апаратному забезпеченні за допомогою примітивів мереж взаємозв'язку. Цей метод проектування демонструється шляхом модифікації шифру з декількома архітектурними варіаціями для демонстрації гнучкості. Крім того, встановлено, що для цього шифру включення нової методології проектування, яка залежить від ключів, практично не впливає на кінцеву продуктивність або результати потужності, отримані в процесі синтезу, і служить лише для підвищення стійкості, діючи як контрзахід критичної сприйнятливості, яку демонструють шифри на основі перестановок.

Удосконалений метод застосовний до криптосистем в обсязі одного

комутатора, а не просто з надання методів перестановки через маршрутизації. Тому, необхідні дослідження того, як стійкість криптосистем може бути підвищена за допомогою комутаційних мереж і не вимагає використання додаткових складних математичних операцій для забезпечення безпеки. Крім того, враховані і застосовані контрзаходи в цій методології проектування і вона може бути застосована до альтернативних ключових методів або навіть до побудови шифрів виключно на основі цієї методології.

Тому, в контексті удосконалення методу розглянемо криптографію та інформаційну безпеку. Вони базуються на тріаді, яка складається з трьох основних послуг, необхідних для інформаційної безпеки: конфіденційність; цілісність; доступність. Тому, вони вимагають: конфіденційності та захисту даних від сторонніх осіб; дані не можуть бути змінені неавторизованими особами; щоб конфіденційні дані були легкодоступними та своєчасними. Однак прийнято вважати, що інформаційна безпека неповно представлена лише цією тріадою, що потребує додаткових послуг: автентичність; незаперечення. Щоб дані можна було перевірити як справжні – це четверта послуга і вимога. А остання така, що дані / дії могли б бути адекватно пов'язані з відповідним вихідним суб'єктом. Незважаючи на те, що цей набір послуг є основними принципами та вимогами до інформаційної безпеки, то вони також застосовні до криптографії та повинні бути дотримані. Однак, за будь-якими криптографічними алгоритмами стоять властивості Шеннона: заплутування; змішування. Тобто, отриманий шифротекст повинен залежати від більше ніж однієї частини ключа і зміна одного біту у відкритому тексті повинна привести до більших змін в отриманому зашифрованому тексті. Ця невелика зміна тексту, яка призводить до більшої зміни в шифротексті, відома як ефект лавини. Плутанина, яка зазвичай виконується за допомогою поля підстановки, замінює блоки бітів у відкритому тексті якимось іншим блоком, наприклад, символ один замінюється на другий. Змішування потенційно з використанням поля перестановки переставляє відкритий текст. Ці дві концепції в кінцевому підсумку пов'язані з інтеграцією підстановки і транспозиції для перетворення відкритого тексту в шифротекст за допомогою

деякого залученого процесу.

Для того, щоб забезпечити краще розуміння реляційної моделі атаки на шифри, засновані на пермутації, спочатку детально розглянемо шифр. Наприклад, 160-бітна реалізація шифру використовує 160-бітний секретний ключ, в якому тільки 132-біт ключа використовується в кожному раунді і в поєднанні з адитивним значенням з круглим лічильником. Так як 64 раунди виконуються на відкритих текстових даних, то цей розклад клавіш зображує круглий підрозділ та операцію оновлення для побудови наступного раундового підрозділу. Крім того, бітна операція відображає додаткові дані лічильника раундів. Кругла архітектура містить ряд прямокутників, що відображатимуться подальшою маршрутизацією, що зображуватиме операцію ОТ.

Модель атаки, яку розглядаємо, будує метод атаки на мережу перестановок шифру, який в кінцевому підсумку виявляється застосовним до всіх шифрів на основі перестановок. Методологія використовує диференціально-відкриту атаку і спочатку формується шляхом захоплення витоку потужності / електромагнітного поля з метою отримання диференціального ЗТ на круглому виході. Щоб краще проілюструвати це, розглянемо приклад, в якому, якщо розглянемо будь-яку зміну одного біту на крайній лівий стороні, то результуючі біти / місця, які потенційно завжди зазнають впливу, будуть змінені. Метод служить для використання цього обмеження / конвенції архітектурного проектування, івсі шифри на основі стратегії з витісненням використовують таку методологію проектування, яку можна експлуатувати. Таким чином, біти, на які потенційно впливає зміна одного стану, завжди будуть знаходитися в одному і тому ж положенні.

Міжмережеві мережі можна використовувати для різних застосувань маршрутизації: комутації мереж; високопродуктивних / паралельних обчислень; сортування. Неблокуючі міжмережеві мережі — це мережі, в яких кінцеві точки знаходяться поза межами і будь-яка перестановка джерела-призначення можлива через комутацію. Найпростіший неблокуючий елемент непрямої комутації, відомий як комутатор є основою для проміжного управління потоками даних для різних топологій мережі. В нього така особливість, що жодні дві початкові кінцеві

точки не намагатимуться використовувати один і той самий маршрут. Крім того, основна увага буде зосереджена не на топологіях маршрутизації між цими структурами, а буде зосереджена на використанні виключно цих структур у криптографічних архітектурах.

Міжмережеві мережі в криптографії досліджують їх використання поза телекомунікаційними системами для створення перестановок з низькою латентністю. Аналогічним чином, вона була запропонована для їх вивчення в криптографічних алгоритмах. Так є концепція інтеграції їх в криптографічні архітектури, що робить це шляхом демонстрації архітектури одноключової криптосистеми з використанням комбінації і булевих функцій. Для наочності, система з одним ключем - це замок з поворотним набором, тобто тільки один ключ відкриє замок. Для боротьби з проблемами перемикання елементів використовується функція налаштування керування для керування прохідними та кросоверними операціями. Ця функція, породжена деяким набором булевих операцій, керує всіма ними в топології. Завдяки їй може бути покращена за допомогою генератору псевдовипадкових функцій стійкість. В кінцевому підсумку використання їх в криптографії є життєздатним методом для виконання перестановок при виконанні лавинної властивості. Інтеграція їх для виконання перестановок в криптографічному програмному забезпеченні дозволяє скоротити кількість інструкцій і циклів, необхідних в бітових перестановках. Це досягається шляхом розробки різних методів пермутації, в яких показано, що кожен з цих методів скорочує цикл і інструкції в порівнянні з використанням примітивів таблиці пошуку. Тим самим встановлено, що застосування мережевих примітивів може бути корисним у криптосистемах. Істотна відмінність полягає в тому, як генеруються керуючі біти для комутаційних елементів, на відміну від відомих раніше методів. Різноманітні методи генерують керуючі послідовності, а функціональність комутаційного елемента розширена для обробки керуючих значень. Крім того, цей метод не реалізується в мережі комутації. Він використовує ряд регістрів з топологіями маршрутизації і з'єднує регістрові локації з мультиплексорами. Вибраний керуючим значенням вихід передається в

тимчасовий регістр. У ньому використовується проблемна одиниця, яка заснована на методі управління, а не на генераторах функцій. Це дані, що містяться в пакеті інформації. Це задано на основі великої сітки, де передача пакетів здійснюється за допомогою окремих частин пакета і таким чином встановлено, що кінець списку слідує за конкретними кроками, які слідують за заголовком списку, який спочатку відправляється через перевірену проблемну одиницю системи. І навпаки, оскільки попередні методи використовують неблокуючі мережі, проблемна одиниця за своєю суттю є блокуючою і не має маршрутизації без суперечок. У випадку, якщо два пакети спробують використовувати один і той же маршрут, то один буде заблокований, а всі наступні частини, що слідують за заголовком списку, також зупиняться. Крім того, цей метод досліджує використання як рандомізованого, так і конвеєрного планування для серійного шифрування даних. Результати, отримані за допомогою методу рандомізованого планування, ще більше підтверджують той факт, що колізія / суперечка не обробляється в цій системі. Цей варіант удосконалення методу вимагає більш тривалого часу передачі через збільшення колізій і невпорядкованості пакетів і при цьому конвеєрний метод представлений для пом'якшення цього недоліку.

Залежність ключа в криптографії досліджена достатньо. Але у випадку потенційної проблеми зі сторонніми знаками потрібно розглянути особливості його впливу. В даний час вони стосуються лише підвищеної безпеки та лавинних властивостей для динамічних ключозалежних мереж ЗТ. Незважаючи на те, що підвищення стійкості ЗТ за допомогою механізмів, що залежать від ключа, забезпечує підвищену безпеку в одному аспекті, воно все ще не має на меті усунути інші вразливості багатьох алгоритмів, заснованих на раніше деталізованій моделі атаки.

Існуючі реалізації, залежні від ключів, не мають на меті усунути сприйнятливість криптографічних алгоритмів. З цієї причини було створено механізм, який може створювати як залежні від ключа мережі ЗТ, так і ОТ. Таким чином, представляємо запропонований підхід до боротьби з такими диференціально-відкритими текстовими атаками, застосовними не тільки до

шифрів на основі IoT, але й до всіх шифрів, заснованих на перестановках.

Запропонований підхід використовує керовану версію комутатору для індукції ключових перестановок круглих даних. Однак цей підхід все ще застосовний до будь-якого шифру, що використовує мережі перестановок, а не тільки до прямих варіантів. Дана архітектура була побудована на основі оригінального представлення та деталізації шифру за допомогою модифікованих типів перестановок для включення шару керованих структур, реалізованих в різних точках круглої архітектури. Він був побудований навколо шифру таким чином, що ширина шини вводу / виводу комутаторів базувалася на кількості блоків, які були інтегровані. Оскільки біти бітного ключа залишалися невикористаними за раунд.

Архітектури були побудовані і перевірені з точки зору даних про продуктивність, площу та потужність, зібрані в процесі синтезу для кожної з архітектурних реалізацій, порівнюючи їх з незміненим шифром.

Спочатку представляємо результати синтезу для кожної з цих архітектур де таблиці пошуку та регістри відображаються в термінах використання логіки зрізів, при використанні мети та стратегії проектування синтезу за замовчуванням. Крім того, для наочності, відсотки позначають загальний відсоток використання логіки зрізів для використовуваної плати розробки. Загальна кількість зрізів для регістрів для цього пакета становить прийнятну кількість. Після цього архітектури були відображені на платі і згенеровані звіти про потужність. Кожна з архітектури була запущена з сигнальною активністю за замовчуванням. Тому, детально описані загальна, динамічна та статична потужність для кожної з архітектур шифрів.

Для того, щоб підтвердити / спростувати цю гіпотезу, необхідно буде провести подальші експерименти з використанням інших криптографічних алгоритмів і цього методу. На додаток до цього, оскільки логіка зрізів відрізняється між платами розробки та логічними ядрами, її також потрібно досліджувати на кількох платах розробки та пакетах. Таким чином, ця перевага не є винятковою як для шийру, так і для плат.

Розглянемо концепції, які мають на меті підвищити стійкість шифрів, як це може бути застосовано до інших шифрів, а також які ще більше підтримують

запропонований підхід. Диференціально-відкриті атаки є доповненням до типів атак. Розглянемо наслідки цих доповнень та архітектурних модифікацій щодо існуючої сприйнятливості, яку демонструє алгоритм. Ця методологія використовує диференціально-відкриту атаку та ініціалізовано виконується шляхом захоплення витоку живлення / електромагнітного поля з метою отримання диференціала ЗТ на круглому виході. Однак, в той час як вона фіксує витік для захоплення ентропії і визначення диференціала ЗТ на круглому виході, враховуючи, що перестановка бітів відома, або статична, враховувати треба цей сценарій при вивченні архітектури. Якби ці перестановки і спостережувані круглі виходи не були статичними, а скоріше залежними від ключа результатом, тоді спостережуваний диференціал представляє два можливих значення. Наприклад, при використанні для бітів даного раунду може бути отримано значення на основі залежності від ключа та контрольованих перестановок. При розгляді шифру це має істотне значення, так як ЗТ заздалегідь визначений для розрядних круглих даних. Це означає, що на основі такої ключово-залежної перестановки замінені дані спочатку могли бути не простими числами і, таким чином, порушують можливість чітко визначити точне значення підстановки з інакшої статичної мережі перестановок. І навпаки, без використання їх, значення результату статично представляло заміну і дозволяло визначати фрагмент відкритого тексту з отриманих переставлених круглих даних. Потенційні біти, на які можна вплинути, змінивши перший фрагмент відкритого тексту в шифрі, і навпаки, використовуючи ключ-залежний шар ілюструє біти, які потенційно можуть бути уражені, а це збільшує число можливостей і зменшує величину статичного диференціала.

У зв'язку з упорядкуванням структур потенційно може бути порушено лише приблизно вдвічі більше бітів, однак можна реалізувати архітектуру, яка дає більшу кількість бітів, що зачіпаються при зміні першого шифру. Аналогічно, при використанні побітових операцій число бітів більше, а кругла залежність може бути інтегрована за допомогою використання бітів з круглого лічильника. Хоча демонстрація цієї методології була застосована лише до ОТ шифру, але вона може бути розширена до уможливлення використання ключових перестановок ЗТ. Якщо

застосувати керовані комутатори до певної унікальної схеми даних ЗТ, де не виникне жодних колізій для всіх комбінацій, тоді включення ключових даних ЗТ буде обчислене.

Цей метод застосовний до інших шифрів і може вимагати мінімальних зусиль для включення. Це буде продиктовано власником шифру, а розширення довжини ключа для включення мереж, заснованих на перестановках, що залежать від ключа, є здійсненним рішенням, як це було встановлено для існуючих шифрів, в той час як додатковий розклад ключів для таких мереж на основі перестановки та заміни ключів також може бути розроблений і реалізований, що ще більше розширює простір ключів на цінні папери, пов'язані з атакою на такі системи.

Хоча цей удосконалений метод показує збільшення диференціалу уражених бітів за допомогою ключ-залежних структур, шифр також застосовний для ін'єкції помилок, так і для аналізу бічних каналів, з доведеним контрзаходом до обох цих атак. Застосування цього контрзаходу полягає в розкладанні ЗТ на загальні реєстратори за допомогою системи і реалізації схеми маскування. Оскільки зміни, які представляємо в шифрі, то кругла структура має мінімальний вплив і лише служить для подальшого зменшення витоку.

Використання керованих примітивів мереж, залежних від ключів, підвищує стійкість до диференціально-відкритих атак, при цьому жодних порушень продуктивності не буде, а в найоптимальнішому сценарії потужність системи буде знижена. Крім того, використання цієї методології проектування є дуже гнучким рішенням, застосовним до мереж на основі стратегії, яка інтегрується в існуючі архітектури шифрів і може бути використаною з іншими методами та контрзаходами для подальшого підвищення загальної безпеки шифрів.

3.3 Висновки

Таким чином, удосконалений метод показує збільшення диференціалу уражених бітів за допомогою ключ-залежних структур. Шифр також застосовний для ін'єкції помилок, так і для аналізу бічних каналів, з доведеним контрзаходом

до обох цих атак. Застосування цього контрзаходу полягає в розкладанні ЗТ на загальні реєстратори за допомогою системи і реалізації схеми маскування. Оскільки зміни, які представляємо в шифрі, то структура має мінімальний вплив і лише служить для подальшого зменшення витоку.

4 ПІДВИЩЕННЯ БЕЗПЕКИ ШИФРУ ЗА ДОПОМОГОЮ ДИНАМІЧНИХ МЕРЕЖ ПЕРЕСТАНОВОК З КЛЮЧЕМ

4.1 Вибір типу архітектури для підвищення безпеки шифру за допомогою динамічних мереж перестановок з ключем

Апаратні реалізації алгоритму шифру і шифри з використанням структур є чутливими через невипадкову поведінку мереж перестановок всередині круглих архітектур. Хоча шифр, спочатку забезпечував умови стійкості до диференціальних атак, для шифру та архітектури він не тільки сприйнятливий до диференціальних атак відкритого тексту, але й дана архітектура не вимагає використання методів для виконання диференціальних атак з відкритим текстом. Розглянемо заходи протидії таким атакам, застосовні до всіх шифрів, поряд з простішими контрзаходами для загальної стійкості та врахування криптографічного обладнання.

Розглянемо атаку з відкритим текстом, яка використовує статистичний аналіз для кореляції відмінностей у виводі шифротексту на основі змін у вхідних даних відкритого тексту, щоб краще розрізняти, які операції виконуються з метою реконструкції секретного ключа. При цьому використовується невипадкова поведінка мереж перестановок для його виконання на імплементації базового шифру. Хоча реалізація базового шифру, істотно відрізняється від тієї, що спочатку була спочатку аналізована, але це є незначним впливом на результат, і використання побайтових операцій замість побітових не має жодного відношення до стійкості проти цієї атаки. Успіх в атаці досягається за рахунок вимірювання електромагнітної індукції мікроконтролеру, в роботі якого знаходиться реалізація шифру. Це дозволяє зашифрувати серію вибраних значень відкритого тексту, а потім зафіксувати круглий вихідний диференціал для статистичного аналізу. Метод може містити такі наступні кроки:

1) вибір тексту для атаки;

2) вибір звичайного тексту для використання його як сторонніх знаків;

3) сформувати початковий текст з вставленням звичайного тексту, який

додається;

4) зашифрувати два тексти: початковий без змін; початковий текст із вставленням сторонніх знаків;

5) визначення електромагнітного випромінювання;

6) обчислення різники електромагнітної індукції;

7) генерація ключових кандидатів в зашифрованих текстах;

8) застосування формули для визначення побітових раундів:

9) якщо на кроці 8 не встановлено фрагменти зі сторонніми знаками, тоді повторити кроки 5-8;

10) якщо встановлено фрагменти зі сторонніми знаками, тоді завершити роботу.

Важливість наданої інформації ЗТ і ОТ полягає в тому, що зміна лише одного фрагменту має заздалегідь визначену, невипадкову, поведінку на результуючому виході раунду, оскільки архітектура раунду шифру за такого сценарію, вказують на вставку.

Таким чином, ця атака статистично корелює зміни між відкритим текстом і зашифрованим текстом. Метод використовує значення для вимірювання витоку та отримання статистичної різниці між заданим набором відкритого тексту. Невипадкова поведінка ОТ дозволяє генерувати ключових кандидатів відносно змінних, що задовольняють умові дії кроку 5.

У той час як метод використовує потенційно дорогі інструменти для вимірювання та збору даних, знайдено альтернативний метод виконання атаки на шифр, який не вимагає методів дороговартісних і може бути використаний проти архітектури шифру або потенційно будь-якої реалізації, яка виводить подібним чином, з цією архітектурою. Хоча метод виконання атаки аналогічний методу ручного шифрування, але не вимагаємо вимірювання електромагнітної індукції, а лише вимагаємо спостереження за напругами для фактичних вихідних бітів шифротексту. Щоб продемонструвати це, використали реалізацію шийру з відкритих джерел, таким чином, що це була архітектура разом зі стандартною бібліотекою. Використовуючи її, їсинтезували, спланували та направили

архітектуру на макет, проектування і виконання атаки. Тому, обрані відкриті тексти були використані для виконання атаки.

Таким чином, архітектура шифру за своєю суттю сприйнятлива до диференціальних атак у першому раунді або другому кадрі вихідних даних з використанням цієї аналізованої техніки. Це дозволяє зловмиснику остаточно повернути біти в оригінальному ключі, який використовується як початковий до шифру, а не просто будь-який круглий ключ. Запропонований спосіб протидії за допомогою мережі з динамічними ключами є методом, який використовує динамічні мережі перестановок з ключем. Під ключем розглядатимемо керовану деякою функцією з потенційно ключовими бітами схему, але це може бути якась інша структура або функція. У той час як концептуально використання динамічних та/або ключових перестановок у криптографічних алгоритмах саме по собі з'являлося раніше, застосовність методу, який розроблено, не тільки до шифру, але й до всіх шифрів та його використання для підвищення стійкості є методом, бо спеціально розроблений і поширюється для динамічних перестановок для всіх компонентів матриць.

Використання динамічних або ключових структур у криптографічних алгоритмах та апаратному забезпеченні може слугувати для посилення можливостей змішування, підвищення стійкості або запобігання атакам.

Таким чином, замість того, щоб використовувати єдину статичну мережу, використовуємо серію маршрутизаційних мереж і контролюємо, яка з них буде використовуватися під час раунду, задавши деяку інвертовану функцію. Називатимемо їх мережами маршрутизації. Введення більшої кількості мереж не повинно призводити до витоку додаткової інформації. Мережі повинні ефективно створювати плутанину. Головною метою метою розробленого методу є примусити зловмисника працювати на ключ та викликати при цьому сумніви в нього.

Для експериментальної установки можна використовувати те саме немодифіковане ядро, яке реалізовано на додаток до створення модифікованого ядра, яке реалізувало три додаткові біті мережі перестановок поряд зі стандартним шаром перестановки базового шифру. Ці мережі можуть бути довільними, а

маршрутизація задана структурами мереж взаємоз'єднання.

Хоча ці мережі задані як входові, алгоритмічно маршрутизація була розширена для розміщення розрядної шини, а використовувані мережі не використовують комутацію. І навпаки, задані входові реалізації можуть бути використані для виконання перестановок шляхом включення керованих перемикачів. Однак, це згодом вимагатиме додавання принаймні бітових керуючих сигналів. Єдиною зміною, внесеною в ядро системи, було включення пакета функцій, що містить функції для вищезгаданих мереж, з модифікацією файлу, що змінило поведінку.

До сутності значення також може бути додано двобітний порт і, він може бути з'єднаний у верхньому модулі з бітами круглого ключа. Це дозволяє змінювати використання мережі протягом раундів при зсуві ключа. Функція представляє собою стандартне змішування, виконане шифром.

Таким чином, розглянемо виходи в кожному з наступних сценаріїв, в яких зловмисник: приймає стандартну поведінку шифру; спостерігає нерегулярну поведінку, але не знає про будь-яку конкретну схему; спостерігає за нерегулярною поведінкою і добре знайомий з конкретною схемою.

Перший сценарій полягає в тому, що якщо розглядати ту ж атаку, виконану без додаткових знань про те, що ядро модифіковане, тоді, хоча прогнозовано атака зазнає невдачі, або у випадку цього вибору відкритого тексту насправді можна встановити, що при обчисленні для будь-якого з цих диференціалів отримуємо результат для шифру, який охоплює кілька спроб зловмисника. Це в кінцевому підсумку запобігає подальшій атаці зловмисника і генерації ключових кандидатів під приводом того, що ядром є заданих шифр, і потрібно ретельно вивчити поведінку шифру, щоб визначити його основну поведінку.

Модифікований перший сценарій полягає в тому, що зловмисник розпізнає нормальну поведінку і зможе виконати ГА на раунді. Однак, незважаючи на те, що наступні раунди можуть не демонструвати стандартної поведінки шифру, і потенційні біти, на які впливає мережа перестановок, не будуть заздалегідь визначені, то існують додаткові міркування щодо цього сценарію: як зберігається

ключ; який рівень контролю має зловмисник.

Повинні враховувати це через вплив функції перестановки на загальну безпеку системи, оскільки якщо ключ зберігається всередині та завантажується під час запуску / скидання шифрування/системи, зловмиснику з повним контролем над системою не потрібно турбуватися про змінну мережу перестановок. Зловмисник може просто скинути пристрій після виведення другого кадру, звести систему нанівець і виконати атаку. І навпаки, якщо зловмисник може вибрати лише відкритий текст і спостерігати за результатом кожного раунду, то потрібно буде розпізнати загальну поведінку на системному рівні та реконструювати кожну таблицю мережевого міксу.

Суть другого сценарію в тому, що здійснюється спостереження за нестандартною поведінкою. Зловмисник регулярно спостерігає за нестандартною поведінкою шифру, але не знає про зміни в ядрі. У цьому випадку зловмиснику доведеться реконструювати всю мережу на рівні перестановки. Враховуючи, що зловмисник не має контролю над ключовим значенням і має можливість використовувати лише вибрані відкриті тексти, то це був би надзвичайно виснажливий і складний процес. Якщо припустити, що зловмисник знає поведінку ЗТ у шифрі, то йому спочатку потрібно побудувати відкритий текст, який гарантує одноразову активацію ЗТ для раунду та обнуляє нецільові спроби ЗТ. Це, по суті, прирівнюється до спроби грубої сили щодо решти ключових бітів, щоб визначити поведінку мережі, заснованої на реакції, і є марною. І навпаки, якщо припустити, що зловмисник має рівень контролю з першого сценарію, то цей процес може бути прискорений лише за допомогою вибраного відкритого тексту та скидання системи, як тільки буде помічено бажану зміну вихідного сигналу раунду.

В третьому сценарії визначається модифікована реалізація. Тепер припускаємо, що зловмисник спостерігає за нерегулярною поведінкою і може визначити, що це конкретно використовуваний стандартно шифр і знає багатомережевий рівень перестановки. Зловмисник, як і раніше, розпочне з обробки диференціалу, але тепер повинен визначити, яка мережа використовується. Інтуїтивно, якщо диференціал не є жодним елементом реакції,

то він може бути виключений, і зловмисник зменшує набір можливостей для ключа для раунду. Це також явно пояснює, чому використовуємо нотацію з ключем, в той час як прив'язано елемент управління до ключових бітів для простоти, то видно безпосередній недолік у цьому виборі, і саме тому залишили його як вільне прив'язування того, що буде використовуватися для керуючих значень. Якщо розглянемо мережі, то встановимо, що кожна довільно створена мережа має на виході унікальні набори перестановок, що не перетинаються, в такому випадку, якщо зловмисник знає відображення ОТ для кожної мережі, то продовження атаки стає елементарним. Таким чином, було б важливо, щоб нестатична поведінка мережі частково перетиналася з іншими включеними мережами, що в кінцевому підсумку вимагатиме вивчення більшої кількості відкритих текстів, щоб зловмисник звузив поведінку до однієї мережі для даного раунду. В кінцевому підсумку це збільшує складність і час, необхідний для реконструкції ключів.

При розгляді простого методу атаки для немодифікованого шифру може бути обчислень мінімально, або, в кращому випадку, оптимально підібрані пари ключів. Аналогічно, якщо збережемо шар шифру в модифікованій динамічній мережі, то в кращому випадку для зловмисника це значення справедливе, коли керуючі біти створюють шар змішування шифру. І навпаки, коли спочатку вибираються інші мережі, то він стає в значній мірі недетермінованим, так як і перестановка, і управління стають круглим варіантом в порівнянні з немодифікованим круглим інваріантом. Крім того, тому що контроль в даному випадку базується на верхніх бітах невикористаного ключа, перестановки між раундами базуються на неспостережуваних диференціалах ЗТ при оновленні ключа. Це значною мірою залежить від здатності зловмисника розпізнати, яка мережа перестановок використовується, зіставити це з набором перестановок вводу-виводу, щоб дізнатися справжню диференціал, згенерувати кандидатів і виконати атаку.

Найпростішим і безпосереднім рішенням було б просто включити умовну фіксацію виводу для шифротексту при розгляді реалізацій шифру. Однак ця концепція застосовна до будь-якої архітектури шифру, оскільки методи,

продемонстровані в цій роботі, ілюструють важливість запобігання появі проміжних даних шифротексту на виході системи. Хоча це за своєю суттю збільшить як додаткові витрати на системному рівні, так і енергоспоживання, не слід припускати, що зловмисник не матиме прямого доступу до реалізації під час проектування архітектури.

Ще одним важливим фактором для запобігання атакам є швидкість активації ЗТ, і хоча шифр розглядає такі міркування, показуючи, що раундовий диференціал має мінімум активованих ЗТ, але вони не опрацюють критичність активації ЗТ під час першого раунду та її вплив на архітектуру; Таким чином, вони аналізують безпеку практичності диференціального криптоаналізу. В ідеалі, кругла архітектура при розгляді побітових операцій повинна бути більш схожою на структуру, таку, що: початкова перестановка ефективно розповсюджує біти з кожного угруповання ефективно по всьому результуючому впорядкуванню, що передається ЗТ; функція ЗТ ефективно виконує деяке перетворення над полем; що ці зміни ентропії ефективно розповсюджуються по всьому результуючому впорядкуванню через деяку перестановку, так що не утворюється єдиного групового диференціала.

Таким чином, оригінальний алгоритм і архітектура шифру піддаються диференціальним атакам з відкритим текстом, але при цьому показано практичність і доцільність виконання такої атаки на реальні фізичні реалізації, вибираючи використання цієї архітектури в апаратному забезпеченні. Крім того, показано, що завдяки використанню мереж перестановок з ключем така атака на шифр може бути успішно або повністю зірвана, або призвести до подальших витрат часу та труднощів для зловмисника з відновлення секретного ключа. Крім того, оскільки представлена техніка ключової мережі перестановок є загальною і націлена лише на зміну самої функції перестановки, то вона має незначний загальний вплив, і багато архітектур шифрів, що використовують структури подібним чином, можуть використовувати представлену техніку, так і поєднувати її з додатковими методами зміцнення: динамічним ЗТ; ключовим ЗТ; маскуванням; ключовими субперестановками тощо.

4.2 Проєктування архітектури для мережевих шифрів підстановки-перестановки

Запропонована методика об'єднує концепції розширюючи їх можливості та загальну ефективність. Крім того, представимо нові примітиви для побудови розрізаної одноступінчастої архітектури для забезпечення максимальної стійкості шифру.

Для цього розширюємо логіку керованого комутатора спочатку деталізовану для використання в криптографічних структурах і розширенючи керований комутатор, який супроводжується набором операцій маршрутизації, які можуть бути виконані запропонованим комутатором. Є кілька керуючих бітів на відміну від однорозрядного, необхідного для простого керованого комутатора: два біти для управління напрямком (зверху, знизу, посередині); один біт для перемикання. За допомогою них можемо побудувати кільцеподібну з'єднану структуру, яка також може імітувати прямий прохід і кросоверну маршрутизацію, що дозволяє ефективно створювати одноступінчастий тор. Всім їм передається однакове значення для керування спрямованістю, що дозволяє виконувати індивідуальні операції проходження / перехресні операції.

Розширюємо структуру шифру, включаючи динамічні мережі через коробку з перестановкою ключів. Перевага використання динамічних мереж діапазону полягає в тому, що багато існуючих з них є самоінвертуючими або легко інвертуються через два етапи. Вони є самоінвертуючою мережею і після двох етапів виробляється початкове впорядкування вхідних даних. Аналогічним чином, незважаючи на те, що не всі мережі є самоінвертуючими, то вони мають додаткові етапи і це створить початковий порядок введення.

При розгляді методу атаки на шифр, який покладається на одноразові зміни, інтуїтивно найбільш доцільно з точки зору безпеки використовувати мережі. Однак, міркування безпеки та стійкості до атак здійснено за допомогою алгоритмічного вибору побітово, а не апаратного забезпечення під час створення архітектури, оптимізованої для області. Крім того, якщо розглядаємо структуру як

є, то використовуємо відображення вводу / виводу ЗТ до ОТ один до одного, і навпаки, якщо використовуємо комутатори, тоді потрібно на бітовому рівні два входи. Таким чином, питання полягатиме в тому, чи можна зменшити кількість мереж шляхом об'єднання обох методів. Якщо так, то яка ефективна конфігурація структур і які мереді повинні міститись на основі конфігурації, щоб забезпечити ідеальне перемішування. Якби розглядали побудову ОТ як набору, то потрібна така конфігурація, яка дозволяє зовнішнім впливам завершуватися на всіх виходах раніше. Недоліком є те, що для перемішування потрібно більше шарів на внутрішніх виходах. Щоб змінити цю конфігурацію і зменшити кількість шарів треба, щоб використовувати те, що вже існує в цих алгоритмах або вимагати мінімальних змін, і лише після перевірки двох шляхів було понесено значні бітові витрати, які просто нездійсненні. Таким чином, щоб інтегрувати цей діапазон покриття від входу до виходу, можемо збільшити мобільність даних у поєднанні зі зменшеним розміром, і знову ще один зріз. В результаті виходить архітектура, яка дозволяє зменшити кількість мереж за рахунок мобільності даних. Потрібно визначити належну кількість мереж для забезпечення імовірнісного ідеального перемішування від входу до виходу, а також як структуру мережі, так і самоінверсію. Це стає простішим для усвідомлення при використанні наборів пермутації, створених лише з операцій з маршрутизацією прямо від зрізу до зрізу. Для цього припустимо, що ширина шини вводу/виводу становить три біти, тому кожен пристрій обробляє спробу злому, таким чином, що це дозволить використовувати невикористані біт ключа під час раунду і вимагатиме лише мінімальної логіки для генерації інших керуючих сигналів, які потенційно є похідними від цих бітів.

Загальна кількість можливих наборів перестановок і схема покриття, кожна з яких випливає з входу на вихід, базується на тому, що існує розрив в три покриття, який буде циклічно зміщуватися вниз при розгляді наступних біт у вхідному ряду. Таким чином, можемо створити мережі, які будуть задовольняти ідеальним умовам перемішування і дозволяти будь-якому входу досягати будь-якого виходу.

Мережі можуть бути просто створені, щоб задовольнити вимоги до покриття для ідеального перетасовування варіанту керування і при цьому ці мережі повинні бути реалізовані. В поєднанні з перестановками ці дві мережі в кінцевому підсумку охоплюють всі можливі перестановки для входу на вихід. Таким чином, якщо використовуємо мережу з простим проходом, то між двома мережами потрібен лише один біт для керування і дозволяє однобітовій зміні впливати на будь-який вихід. Крім того, враховуючи, що мережа є самоінвертуючою, то вона не потребує жодних додаткових мереж ні для шифрування, ні для дешифрування.

І навпаки, якби робити це за допомогою типових керованих комутаторів в поєднанні з мережами то потрібна принаймні структура, а це означало б, що знову потрібні додаткові біти, якщо використовуємо половинчасту ширину введення-виведення, і це, в кінцевому підсумку, не дуже практично для більшості шифрів.

Якщо хочемо зберегти побітові операції, то знадобиться багато біт, тоді як при використанні розробленої методики потрібно принаймні втричі менше біт, а при розгляді стандартного шифру це дуже надмірно.

Однак, якщо застосуємо це з таким алгоритмом, то можна буде модифікувати алгоритм розкладання круглого ключа для генерації додаткових бітів, необхідних з мінімальними зусиллями та додатковими витратами. Аналогічним чином, просте використання подібної схеми розширення ключів може дозволити оптимально згенерувати необхідну кількість керуючих бітів з невикористаних біт в раунді шифру з мінімальними зусиллями шляхом збільшення кількості цих бітів в певні місця таким чином, щоб їх можна було легко інвертувати в процесі дешифрування.

В результаті реалізації мережі будуємо структуру, в якій блок даних розбивається навпіл, оперується круглою функцією, а потім перехрещується. Якщо вважати операції зсуву / перехресного зсуву круглою функцією, то маємо підстановку, пермутацію і перестановку. Крім того, якщо далі розглянемо ряд операцій, що відбуваються між раундами для структури, то отримуємо, що концепції з цієї структури також зараз присутні і включені в круговий розгляд і засновані на пермуаціях варіантів управління. Таким чином, замість простого прямолінійного ОТ, який використовується в типовому шифрі, маємо гібридну

структуру. Оскільки в криптографічних алгоритмах не існує стандартизованого методу розкладу або управління ключами, то треба використовувати те, що потрібне як найпростіший метод для реалізації архітектури при розгляді круглих ключів і управління.

Зміни, внесені в архітектуру шифру в кінцевому підсумку були розглянуті в структурі ОТ, однак в них використовували більшу ширину шини, щоб забезпечити більш мінімальну кількість додаткової логіки управління.

На додаток до видалення стандартного шифру ОТ, замінивши його архітектурою з використанням маршрутів реалізовано половинчасту ширину шини і в кінцевому підсумку кожен шифр в торі обробляє один байт і прирівнюється до тієї ж структури мережі і реалізації. Вибір половинчастої ширини шини призводить для кожного з зрізів і це дозволить використовувати верхні біт ключа, який не використовується під час звичайного раунду, і використовувати бітний круглий ключ. Будемо використовувати ці окремі біти для управління кожною з операцій передачі / перетину у першому зрізі. Застосуємо ці ж біти у зворотному порядку до другого зрізу, щоб не скасувати операції, якщо потрібно використати інверсію цих бітів. Для генерації спрямованих (зверху / знизу / посередині) та керуючих бітів обираємо зменшення кількох варіантів раундів. Вибір для генерації та використання цих бітів був досить довільним, і просто ґрунтувався лише на тому, що було доступно під час будь-якого раунду і як це можна було б використовувати просто та унікально.

Послідовність операцій і те, як вони використовуються в якості керуючих бітів в модифікованому шифрі проаналізуємо прикладом. Для реалізації та експериментів з цією технікою використовували реалізацію шифру, яка моделює оптимізовану для області архітектуру. Після того, як внесено необхідні зміни, необхідні для включення запропонованої методики, спочатку розглядаємо модифіковану архітектуру, оскільки доведення архітектури до макета з використанням тих самих методів є марним, оскільки все ще можемо виконувати атаки за допомогою даних симуляції осцилограм, зібраних завдяки загальній реалізації, і це нічим не відрізнялося б, якби використовували інструменти

стандартної бібліотеки комірок для їх отримання. Знову виконуємо диференціальну атаку відкритого тексту, використовуючи ті самі значення відкритого тексту та досліджуючи вихідні напруги шифротексту другого раунду під час процесу шифрування. Значення відкритого тексту, що використовуються в моделюванні для реплікації атаки та оцінки є достатніми. Крім того, шифр є додатковим відкритим текстом для вивчення як мережевого, так і поточного методу атаки і відсутній у попередніх наборах відкритого тексту. Це додатково дозволить вивчити вихідні диференціали декількох відкритих текстів за різних ключових умов і здатність супротивника обчислити диференціал ЗТ і визначити значення ключового гризу.

Після моделювання модифікованої конструкції шифру і аналізі відмінностей у вихідних даних під час другого раунду можна встановити, що конструкція все ще дає результати, які, здавалося б, можуть бути атаковані на перший погляд. При розгляді диференціалу відкритих текстів для першої використовуваної пари ключів вихідний диференціал, то він може бути обчислений для бітових позицій шифротексту, якщо припустити стандартний шифр. Це означало б, що вихідний диференціал охоплює два впливи, і, таким чином, атака не може бути здійснена. Однак, оскільки не можна припустити, що супротивник не знав про застосування цієї модифікованої системи, то повинні врахувати таку особливість для представленої техніки та можливих очікуваних змін.

Якщо поміняємо місцями ці біти, скасовуючи маршрутизацію, то отримаємо диференціал, що утворюється на відстані трьох впливів від місця, де відбулася зміна відкритих текстів. Якби зловмисник продовжив атаку з цим припущенням, що саме так ці спроби за своєю суттю перестановлені, і ця зміна вихідного сигналу дійсно відображає зміну відкритого тексту, тоді потрібно було б згенерувати кандидатів на вихідний диференціал. Це знову означає ітерацію можливих ключів, що задовольняють рівнянню. Роблячи це для заданих відкритих текстів і вихідних диференціалів, зловмисник не знайде ключових кандидатів, то це додатково підтверджується для цього диференціала введення-виведення при розгляді шифру і отриманих наборів-кандидатів можливих для диференціальних комбінацій

введення-виведення, що мають відношення до ЗТ. При вивченні наборів-кандидатів для даного диференціала введення-виведення, а також до ключових кандидатів знову немає для будь-якого диференціала. У порівнянні з цим, у звичайній мережі могли б створити вихідний диференціал і використовуючи ці самі відкриті тексти, згенерували б ключовий набір кандидатів. Крім того, при вивченні виходу можна встановити, що вихідні значення є просто значеннями ЗТ, в той час як на виході виходить їх більше ніж очікувано, якби при цьому продовжувалася та ж поведінка. Таким чином, це поставило б під сумнів той факт, що значеннями ЗТ є без будь-якого адитивного значення ключа, оскільки він більше не демонструє ту саму поведінку, що і раніше, тоді як зловмисник не робив нічого, крім додавання значень відкритого тексту до вектору атаки, і при цьому зміна вхідних даних для отримання більшої кількості наборів кандидатів і виведення ключа була здійснена.

Розглянемо сценарій, коли зловмисник повинен припустити, що виконується якийсь набір додаткових операцій, і що це може бути операція одного перехресного, правого або лівого зсуву, виконана на вхідних напів спробах. Завдяки цьому припущенню можна отримати диференціал, але для трьох різних реакцій. Хоча атака може згенерувати ключовий набір кандидатів та стає позиційно незрозумілою, якому впливу відповідає цей набір кандидатів, то можна лише припустити, що вона базується на змінах, внесених до відкритого тексту. Оскільки ще не існувало сценарію, в якому деякий набір операцій та їх скасування призводило б до диференціалу виходу, що має позиційне відношення до того, як змінюються вхідні дані відкритого тексту.

Розглянемо отримані виходи на основі цих операцій, причому дві з трьох операцій призводять до того, що вона має диференціал. Однак можемо припустити, що в результаті того, що всі три спроби мають однакове диференціальне значення для цих сценаріїв, зловмисник може припустити одне з двох: позиція є нерелевантною і можуть бути просто згруповані варіанти як можливі; позиція є релевантною і ключ вплинув на диференціали, навіть якщо положення для зміни значень реакцій однакові і вхідні диференціали певним чином впливають на вихід.

Крім того, ці міркування щодо цього вхідного диференціалу не враховують поведінку, яку демонструють ключі, що підвищує додаткову можливість виконання операцій, заснованих лише на відмінностях у цих відкритих текстах.

Якщо супротивник тепер припускає, що кожен тор виконує зсув вправо або вліво, тоді можемо розглянути зворотний бік таких операцій у мережі перестановок. Якщо припустимо подвійне схрещування без зсуву, то результуюча перестановка зсувається і виробляється вихідний вхід, тому немає необхідності розглядати це в табличній формі. І навпаки, якщо припустимо, що дані були двічі зміщені вправо, і до них застосували зсув даних вправо, тоді спостерігаємо перму таційну множину. Немає безперервної лінійки бажаних відмінностей між бітами і розрядів і породжених вихідних диференціалів. Подвійний зсув вправо виробляє перестановку, задану ще далі від виробленої.

Таким чином, на основі вивчення та аналізу отриманих результатів можна зробити висновок про те, що це ефективний метод як для маскування ключових значень, так і для запобігання диференціальним атакам відкритого тексту, одночасно заплутуючи виконувані операції та індукуючи нестатичну поведінку за рахунок використання круглих варіантів та представлених структур. Крім того, цей метод є спрощеним і застосовним до будь-якого шифру, який в даний час використовує статичну мережу перестановок, і може бути легко інтегрований при розгляді сценаріїв і мати невикористані частини даних / ключів тощо.

4.3 Висновки

В цілому, це простий метод захисту зашифрованого тексту, що включає динамічні керовані перестановки, які виробляють зміни, щоб гарантувати, що вихід під час другого раунду не містить одноразових змін і не може бути реалізована класична методологія атаки.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено новий метод забезпечення пропускної здатності дисків для застосунків з інтенсивним обсягом даних та отримано такі результати.

1. Проаналізовано відомі методи та засоби криптографічного захисту від вразливостей в апаратному забезпеченні.

2. Розроблено удосконалення методу криптографічного захисту від вразливостей в апаратному забезпеченні.

3. Здійснено реалізацію розробленого методу криптографічного захисту від вразливостей в апаратному забезпеченні.

4. Здійснено есперименталні дослідження згідно розроблених рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Kumar A. Machine learning-based early detection of IoT botnets using network-edge traffic. *Computers & Securityю.* 2022.

2. Brooks R. Distributed denial of service (DDoS): a history. *IEEE Annals of the History of Computing.* 2021. V. 44. No. P. 44-54.

3. Janiesch C., Patrick Z., Kai H. Machine learning and deep learning. *Electronic Markets.* 2021. V. 31. No. 3. P. 685-695.

4. Han Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021. V. 44. No. 11. P. 7436-7456.

5. Swamy S. N., Solomon R. K. An empirical study on system level aspects of Internet of Things (IoT). *IEEE Access* 8. 2020. P. 188082-188134.

6. Denysiuk D., Savenko O., Lysenko S., Savenko B., Kashtalian A. Method for Detecting Steganographic Changes in Images Using Machine Learning. In: *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece. 2023. P. 1-6. doi:10.1109/DESSERT61349.2023.10416453.

6. F5 Labs. "H1 2023 Bad Bots Review." URL: https://www.f5.com/labs/articles/threat-intelligence/monthly-bot-stats-report-h1-2023.

7. Spamhaus. "Botnet Threat Updates." URL: https://info.spamhaus.com/botnet-threat-updates.

8. Aslan Ö., Refik S. A comprehensive review on malware detection approaches. *IEEE Access 8*. 2020. P.6249-6271.

9. Letteri I., Antonio D. C., Giuseppe D. P. New optimization approaches in malware traffic analysis. In: *International Conference on Machine Learning, Optimization, and Data Science*. Cham: Springer International Publishing. 2021. P. 57-68.

10. Markowsky G., Savenko O., Lysenko S., Nicheporuk A. The technique for metamorphic viruses' detection based on its obfuscation features analysis. *CEUR-WS*

2104. 2018. P. 680-687.

11. Lysenko S., Savenko Ol., Bobrovnikova K. DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering. *CEUR-WS* 2104. 2018. P. 688-695.

12. Savenko B., Lysenko S., Bobrovnikova K., Savenko O., Markowsky G.. Detection DNS Tunneling Botnets // *Proceedings of the 2021 IEEE 11th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), IDAACS'2021*, Cracow, Poland, September 22-25, 2021.

13. Suryati O. T., Avon B. Impact analysis of malware based on call network API with heuristic detection method. *International Journal of Advances in Data and Information Systems.* 2020. V.1. No. 1. P. 1-8.

14. Wurzinger P. Automatically generating models for botnet detection. In: *Computer Security–ESORICS 2009: 14th European Symposium on Research in Computer Security*, *Saint-Malo, France, September 21-23*, *Springer Berlin Heidelberg Proceedings.* 2009. 2009. V. 14. P. 232-249.

15. Haddadi F., Zincir-Heywood A. N. Botnet detection using network flow analysis and support vector machines. *Computer Networks.* 2020. V. 181.

16. Zhao J., Liu Y., Luo X. Deep learning for botnet detection: A survey. *IEEE Access* 9. 2021. P. 82771-82785.

17. Mirsky Y., Doitshman T., Elovici Y., Shabtai A. Kitsune: An ensemble of autoencoders for online network intrusion detection. *IEEE Transactions on Information Forensics and Security.* 2021. V. 16. P. 122-133.

18. Moshkovitz M. Explainable k-means and k-medians clustering. In: *International Conference on Machine Learning*. PMLR. 2020. P. 7055-7065.

19. Xu X., Zheng Y., Liu X. Unsupervised Botnet Detection using Network Traffic Clustering Techniques. *Journal of Computer Networks and Communications.* 2021.

20. Ribeiro M., Vieira M. Deep Learning Clustering for Botnet Detection. *Cybersecurity and Privacy Journal.* 2020. V. 1. No. 1. P. 45-60.

21. Zhang Y., Jiang T., Wang H. Challenges and Solutions in Botnet Detection

Using Clustering Algorithms. *International Journal of Network Security*. 2022. V. 24. No. 2. P. 112-124.

22. Zhu D. Efficient precision-adjustable architecture for softmax function in deep learning. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2020. V. 67. No. 12. P. 3382-3386.

23. Lerke A., Heßling H. On Strange Memory Effects in Long–term Forecasts using Regularised Recurrent Neural Networks. *IJC*. 2022. V. 21. No. 1. P. 19-24. https://doi.org/10.47839/ijc.21.1.2513

24. Bodyanskiy Y., Kostiuk S. Learnable Extended Activation Function for Deep Neural Networks. *IJC*. 2023. V. 22. No. 3. P. 311-318. https://doi.org/10.47839/ijc.22.3.3225.

25. Savenko B., Kashtalian A. Method for Determining the Efficiency of a Distributed Anomaly Detection System. *CSIT*. 2022. V. 2. P. 14-22. https://doi.org/10.31891/csit-2022-2-2.

26. *Acalvio ShadowPlex. Autonomous Deception*. Available at: https://www.acalvio.com/product/ 04.09.2023 (accessed 06.04.2024).

27. *SentinelOne*. Available at: https://www.sentinelone.com/surfaces/identity/ (accessed 06.04.2024).

28. *Proofpoint Identity Threat Defense*. Available at: https://www.proofpoint.com/us/illusive-is-now-proofpoint (accessed 06.04.2024).

29. *Counter Craft Security*. Available at: https://www.countercraftsec.com/ (accessed 06.04.2024).

30. *Fidelis Security*. Available at: https://fidelissecurity.com/fidelis-elevate/ (accessed 06.04.2024).

31. *The Commvault Data Protection Platform*. Available at: https://www.commvault.com/ (accessed 06.04.2024).

32. *Labyrinth Deception Platform*. Available at: https://labyrinth.tech/platform (accessed 06.04.2024).

33. *Labyrinth Deception Platform. Datasheet*. Available at: https://labyrinth.tech/assets/media/pdf/labyrinth-data-sheet.pdf (accessed 06.04.2024).

34. Feng M., Xiao B., Yu B., Qian J., Zhang X., Chen P., Li B. A Novel Deception Defense-Based Honeypot System for Power Grid Network. *International Conference on Smart Computing and Communication*. Cham: Springer International Publishing. 2021, Vol. 13202, pp. 297-307. DOI: 10.1007/978-3-030-97774-0_27.

35. Walter E., Ferguson-Walter K., Ridley A. Incorporating deception into cyberbattlesim for autonomous defense. 2021. *arXiv preprint arXiv:2108.13980*. DOI: 10.48550/arXiv.2108.13980.

36. Anwar A. H., Kamhoua C. A., Leslie N. O., Kiekintveld C. Honeypot Allocation for Cyber Deception Under Uncertainty. *IEEE Transactions on Network and Service Management*, 2022. Vol. 19. No. 3. P. 3438-3452. DOI: 10.1109/TNSM.2022.3179965.

37. Sayed M. A., Anwar A. H., Kiekintveld C., Kamhoua C. Honeypot Allocation for Cyber Deception in Dynamic Tactical Networks: A Game Theoretic Approach. *14th International Conference on Decision and Game Theory for Security. GameSec 2023*. 2023. arXiv preprint. arXiv:2308.11817. DOI: 10.48550/arXiv.2308.11817.

38. Anwar A. H., Kamhoua C. A. Cyber Deception using Honeypot Allocation and Diversity: A Game Theoretic Approach. *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2022. P. 543-549. DOI: 10.1109/CCNC49033.2022.9700616.

39. Anwar A. H., Kamhoua C., Leslie N. Honeypot allocation over attack graphs in cyber deception games. *International Conference on Computing, Networking and Communications (ICNC)*. 2020. P. 502-506. IEEE. DOI: 10.1109/ICNC47757.2020.9049764.

40. Acosta J. C., Basak A., Kiekintveld C., Kamhoua C. Lightweight On-Demand Honeypot Deployment for Cyber Deception. In Gladyshev, P., Goel, S., James, J., Markowsky, G., Johnson, D. (eds) Digital Forensics and Cyber Crime. ICDF2C 2021. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. 2022. Vol. 441. P. 294-312. Springer, Cham. DOI: 10.1007/978-3-031-06365-7_18.

41. Priya D., Chakkaravarthy S. Containerized cloud-based honeypot deception for

tracking attackers. *Scientific Reports.* 2023. V. 13. DOI: 10.1038/s41598-023-28613-0.

42. Al-Shaer E., Wei J., Hamlen K. W., Wang C. Autonomous Cyber Deception. Reasoning. *Adaptive Planning. and Evaluation of HoneyThings. Springer Nature Switzerland AG*. 2019. DOI: 10.1007/978-3-030-02110-8.

43. Wegerer M., Tjoa S. Defeating the Database Adversary Using Deception – A MySQL Database Honeypot. *International Conference on Software Security and Assurance (ICSSA),* Saint Pölten. Austria. 2016. P. 6-10. DOI: 10.1109/ICSSA.2016.8.

44. Kedrowitsch A., Danfeng Y., Gang W., Cameron K. A First Look: Using Linux Containers for Deceptive Honeypots. *Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense (SafeConfig '17)*. Association for Computing Machinery, New York, NY, USA. 2017. P. 15–22. DOI: 10.1145/3140368.3140371.

45. Almeshekah M. H., Spafford E. H. Cyber Security Deception. In: Jajodia. S., Subrahmanian. V., Swarup. V., Wang. C. (eds) *Cyber Deception.* 2016. P. 318, Cham. Springer. DOI: 10.1007/978-3-319-32699-3_2.

46. Zobal L., Kolář D., Fujdiak R. Current State of Honeypots and Deception Strategies in Cybersecurity. *11th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT)*. Dublin. Ireland. 2019. P. 1-9. DOI: 10.1109/ICUMT48472.2019.8970921.

47. Dahbul R. N., Lim C., Purnama J. Enhancing honeypot deception capability through network service fingerprint. *Journal of Physics: Conference Series.* 2017. V. 801. Article no. 012057. DOI: 10.1088/1742-6596/801/1/012057.

48. Razali M. F., Razali M. N., Mansor F. Z., Muruti G., Jamil N. IoT Honeypot: A Review from Researcher's Perspective. *IEEE Conference on Application. Information and Network Security (AINS).* Langkawi. Malaysia. 2018. P. 93-98. DOI: 10.1109/AINS.2018.8631494.

49. La Q. D., Quek T. Q. S., Lee J., Zhu H. Deceptive Attack and Defense Game. Honeypot-Enabled Networks for the Internet of Things. *IEEE Internet of Things Journal*. 2016. V. 3. No. 6. P. 1025-1035. DOI: 10.1109/JIOT.2016.2547994.

50. Rowe N. C. Honeypot Deception Tactics. In: Al-Shaer, E., Wei, J., Hamlen, K., Wang, C. (eds) *Autonomous Cyber Deception*. Springer. Cham. 2019. DOI:

10.1007/978-3-030-02110-8_3.

51. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A. Self-adaptive system for the corporate area network resilience in the presence of botnet cyberattacks. *Communications in Computer and Information Science*. 2018. V. 860. P. 385-401. DOI: 10.1007/978-3-319-92459-5_31.

52. Pomorova O., Savenko O., Lysenko S., Kryshchuk A., Bobrovnikova K. A Technique for the Botnet Detection Based on DNS-Traffic Analysis. In: Gaj, P., Kwiecień, A., Stera, P. (eds) *Computer Networks. CN 2015. Communications in Computer and Information Science*. 2015. V. 522. P. 127-138. DOI: 10.1007/978-3-319-19419-6_12.

53. Bobrovnikova K., Lysenko S., Savenko B., Gaj P., Savenko O. Technique for IoT malware detection based on control flow graph analysis. *Radioelectronic and Computer Systems*. 2022. V. 1. P. 141–153. DOI: 10.32620/reks.2022.1.11.

54. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A., Savenko B. Information technology for botnets detection based on their behaviour in the corporate area network. *Communications in Computer and Information Science*. 2017. V. 718. P. 166–181. DOI: 10.1007/978-3-319-59767-6_14.

55. Moskalenko V., Zarets'kyy M., Moskalenko A., Kudryavtsev A., Semashko, V. Multi-layer model and training method for malware traffic detection based on decision tree ensemble. *Radioelectronic and Computer Systems*. 2020. V. 2. P. 92-101. DOI: 10.32620/reks.2020.2.08.

56. Morozova O., Nicheporuk A, Tetskyi A., Tkachov V. Methods and technologies for ensuring cybersecurity of industrial and web-oriented systems and networks. *Radioelectronic and Computer Systems*. 2021. V. 4. P. 145-156. DOI: 10.32620/reks.2021.4.12.

57. Dovbysh A., Liubchak V., Shelehov I., Simonovskiy J., Tenytska A. Information-extreme machine learning of a cyber attack detection system. *Radioelectronic and Computer Systems*. 2022. V. 3. P. 121-131. DOI: 10.32620/reks.2022.3.09.

58. Fursov I., Yamkovyi K., Shmatko O. Smart Grid and wind generators: an

overview of cyber threats and vulnerabilities of power supply networks. *Radioelectronic and Computer Systems*. 2022. V. 4. P. 50-63. DOI: 10.32620/reks.2022.4.04.

59. Ahmed J., Karpenko A., Tarasyuk O., Gorbenko A., Sheikh-Akbari, A. Consistency issue and related trade-offs in distributed replicated systems and databases: a review. *Radioelectronic and Computer Systems*. 2023. V. 2. P. 171-179. DOI: 10.32620/reks.2023.2.14.

60. Markoulidakis I., Rallis I., Georgoulas I., Kopsiaftis G., Doulamis A., Doulamis N. Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*. 2021. V. 9. DOI: 10.3390/technologies9040081.

61. Tharwat A. Classification assessment methods. *Applied Computing and Informatics*. 2021. V. 17. No. 1. P. 168-192. DOI: 10.1016/j.aci.2018.08.003.

62. Powers D. Evaluation: From Precision. Recall and F-Measure to ROC. *Informedness. Markedness & Correlation*. arXiv. 2020. DOI: 10.48550/arXiv.2010.16061.

63. Markoulidakis I., Rallis I., Georgoulas I., Kopsiaftis G., Doulamis A., Doulamis N. A Machine Learning Based Classification Method for Customer Experience Survey Analysis. *Technologies*. 2020. V. 8. Article no. 76. DOI: 10.3390/technologies8040076.

64. Lysenko S., Savenko O., Bobrovnikova, K. DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering. *CEUR-WS*. 2018. V. 2104. P. 688-695.

65. Lysenko S., Bobrovnikova K., Shchuka R., Savenko O. A Cyberattacks Detection Technique Based on Evolutionary Algorithms. *11th International Conference on Dependable Systems. Services and Technologies (DESSERT)*. 2020. V. 1. P. 127-132. DOI: 10.1109/DESSERT50317.2020.9125016.

66. Chatterjee A., Ghosal S.K., Sarkar R., LSB based steganography with OCR: an intelligent amalgamation. *Multimedia tools and applications*. 2020. V. 79. P. 11747-11765.

67. Astuti E.Z., Setiadi D., Rachmawanto E.H., Sari C.A., Sarker M.K., LSB-

based bit flipping methods for color image steganography. *Journal of Physics: Conference Series*. 2020. V. 1501. 012019.

68. Jaeyoung K., Hanhoon P., Jong-Il P. CNN-based image steganalysis using additional data embedding. *Multimedia Tools and Applications.* 2020. V. 79. P. 1355–1372.

69. Zhang R., Zhu F., Liu J., Liu G. Depth-Wise Separable Convolutions and Multi-Level Pooling for an Efficient Spatial CNN-Based Steganalysis. *IEEE Transactions on Information Forensics and Security*. 2020. V. 15. P. 1138-1150.

70. You W., Zhang H., Zhao X. A Siamese CNN for image steganalysis. *IEEE Transactions on Information Forensics and Security*. 2020. V. 16. P. 291-306.

71. Li F., Yu Z., Qin C. GAN-based spatial image steganography with cross feedback mechanism. *Signal Processing.* 2022. V. 190. P. 108341.

72. Yuan C., Wang H., He P., Luo J., Li B. GAN-based image steganography for enhancing security via adversarial attack and pixel-wise deep fusion. *Multimedia Tools and Applications.* 2022. V. 81. P. 6681-6701.

73. Li D., Deogun J., Spaulding W., Shuart B. Towards missing data impu-tation: A study of fuzzy k-means clustering method. In *Rough Sets and Current Trends in Computing*, Springer, 2004. P. 573–579.

74. Vishnu A., Agarwal K. Ieee cluster. chapter Large Scale Frequent Pattern Mining using MPI One-sided Model. 2015.

75. Haoyuan L., Wang Y., Zhang D., Zhang M., Chang E. Pfp: parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM conference on Rec ommender systems*, ACM, 2008. P. 107–114.

76. Yinan L., Patel J. Bitweaving: fast scans for main memory data processing. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data,* ACM, 2013. P. 289–300.

77. Lin W.-T., Chu C.-P. Determining the appropriate number of nodes for fast mining of frequent patterns in distributed computing environments. *International Journal ofParallel, Emergent and Distributed Systems*, (ahead-of-print):1–13, 2014.

78. Lisboa C., Argyrides C., Pradhan D., Carro L. Algorithm level fault tolerance:

a technique to cope with long duration transient faults in matrix multiplication algorithms. In *VLSI Test Symposium, 2008. VTS 2008. 26th IEEE*, pages 363–370. IEEE, 2008.

79. Lei L. A review of missing data treatment methods. *International journal of intelligent information systems and Tech*, 2005. P. 412–419.

80. Lu L., Shi X., Zhou Y., Zhang X., Jin H., Pei C., He L., Geng Y. Lifetime-based memory management for distributed data processing sys- tems. *arXiv preprint arXiv:1602.01959*, 2016.

81. Смірнов О.П., Поплавський С.Ю., Ковальчук В.К., Лутюк Л.І. Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні / Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький, 2023, С. 278-279. https://kn.khmnu.edu.ua/wp-content/uploads/sites/18/apkn-2023-corpuspaper.pdf

ДОДАТОК А

(обов’язковий)

ПРЕЗЕНТАЦІЯ РОБОТИ

Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні

Олексій СМІРНОВ

Науковий керівник
к.т.н., доцент Катерина БЕРЕЗЬКА

Хмельницький
2024

Актуальність роботи.

Апаратне забезпечення піддається складним атакам. Не тільки програмна частина комп’ютерної системи зазнає впливів та інфікувань, але і певні складові частини апаратного забезпечення. Виробники апаратних засобів і пристроїв можуть виробляти їх з вкладеними в архітектуру частинами, які потім можуть використовувати із певною метою. Криптографічні алгоритми є безпечними в програмному забезпеченні і також захищені в апаратному забезпеченні. В апаратному забезпеченні можуть бути вставки, які можуть бути націлені на певні криптографічні алгоритми. Тому, потребують аналізу та розроблення методи протидії таким атакам із використанням нових стратегій.

Перспективним для розроблення є підхід з використанням генетичного алгоритму, який може бути застосовано для ефективного розв'язання складної задачі зіставлення підграфів, які можуть бути використані для подання архітектури апаратних засобів. Криптографічний алгоритм може бути зламаний за допомогою диференціальної атаки з відкритим текстом зі значно обмеженими ресурсами. Тому, щоб запобігти таким атакам, потрібно в розроблюваному методі враховувати контрзаходи не тільки для криптоалгоритму, але і для всіх мережевих шифрів підстановки-перестановки. Також, крім подання нових примітивів мереж взаємозв'язку, динамічних мереж маршрутизації, потрібно модифікуємо елементи, які вимагають прийняття рішень для зловмисника, щоб створити йому проблеми із виконанням зловмисних дій.

2

Актуальність роботи полягає в розробці методу і засобу криптографічного захисту від вразливостей в апаратному забезпеченні.

Метою кваліфікаційної роботи магістра є покращення ефективності криптографічного захисту від вразливостей в апаратному забезпеченні.

Поставлена мета досягається розв'язанням таких основних завдань:

- проаналізувати відомі методи криптографічного захисту від вразливостей в апаратному забезпеченні;
- розробити удосконалення методу криптографічного захисту від вразливостей в апаратному забезпеченні;
- здійснити реалізацію розробленого методу криптографічного захисту від вразливостей в апаратному забезпеченні;
- здійснити есперименталь<u>ні</u> дослідження згідно розроблених рішень.

Об'єктом дослідження є процес криптографічного захисту від вразливостей в апаратному забезпеченні.

Предметом дослідження є методи та засоби криптографічного з[illegible]исту від вразливостей в апаратному забезпеченні.

3

Наукова новизна отриманих результатів:

- удосконалено метод криптографічного захисту від вразливостей в апаратному забезпеченні, в якому на відміну від відомих було розширено його межі застосування для внесених сторонніх знаків в текст.

На основі проведених досліджень розроблена архітектура і засоби виявлення та захисту від вразливостей в апаратному забезпеченні.

Практична значимість отриманих результатів полягає у розроблених засобах виявлення та захисту від вразливостей в апаратному забезпеченні.

Для розв'язання поставлених задач використовувалися методи криптографії, методи виявлення вразливостей, методи приховування сторонніх знаків.

За темою кваліфікаційної роботи опубліковано одну публікацію [81] у Збірнику наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». (Хмельницький – 2023. – С. 278-279).

4

Проектування апаратних засобів здійсеюється із застосуванням сучасних інформаційних технологій. По суті таке проектування розробляється програмно. І лише повністю відтестовані і промодельовані програмні моделі передаються на реалізацію з метою їх створення як апаратних пристроїв та засобів.

Автоматизація проектування до певної міри є самовизначальною. Вона включає інструменти та процеси, пов'язані з автоматизацією високорівневої архітектури аж до макета плати, готової до виготовлення. На найвищому рівні як правило розпочинають з графа переходу станів (ГПС) і за допомогою певного інструменту перевести його в апаратну модель (АМ). У зв'язку з цим, модель АМ може бути оброблена за допомогою таких інструментів, що дозволяє транслювати структури полів для перекладу в примітиви на рівні макета плати, де архітектура може бути спланована, маршрутизована і згенерована готовою до виготовлення як макет плати, враховуючи, що він формально перевірений за допомогою симуляції та тестування або інших програмних засобів.

5

Довіра до такої моделі апаратних пристроїв та засобів на сьогодні може бути нівельована через ризик внесення в неї програмним чином фрагменту, який в подальшому на апаратному рівні надаватиме доступ стороннім особам да цього пристрою чи засобу. До початку активного використання зловмисниками кіберсередовища замовники плат не дуже зосереджувались на такій проблемі. Але в поточний момент часу кіберризики зросли в усіх сферах використання інформаційних технологій і на усій етапах їх розроблення. Довіра протягом усього процесу створення апаратного пристрою до будь-якої конкретної системи або архітектури була високою, бо компанії припускали, що ліцензовані ядра інтелектуальної власності і готові компоненти за своєю суттю безпечні і не поставлять під загрозу довіру до даного архітектури. І навпаки, не тільки тому, що процес проектування зазнає багато різних форматів і переходів з рук в руки для даної архітектури, але може включати зовнішні, здавалося б, надійні джерела, яких не вистачає загальній довірі.

6

Практика проектування інтегральних схем складається з безлічі різних етапів та тактик скорочення часу виходу на ринок і неповторюваних витрат на проектування при одночасному збереженні та виробництві високоякісних і надійних конструкцій чи схем. Таким чином, багато замовників можуть вирішити здійснити корпоративні придбання, щоб отримати перевірену інтелектуальну власність у існуючих розробників у сфері, на яку вони зараз не орієнтуються або не мають кваліфікованих інженерів. Аналогічним чином, часті випадки, коли компанії використовують готові компоненти від великих корпорацій або ліцензійні IP-ядра від проектних організацій третьої сторони.

Довіра до виробництва плат набуває актуальності. Особливо з розвитком систем з IoT, від яких залежатиме надійне функціонування, наприклад, систем в розумному будинку тощо. На сьогодні немає беззастережної довіри до виробничого процесу. Мається на увазі, що компанії, які вирішують перевести процес в офшор і передати процес на аутсорсинг, щоб належним чином обробляти макети архітектури або іншу конфіденційну інтелектуальну власність, не зловживаючи і не використовуючи їх поза узгодженими умовами виробництва. На основі таких проблем вже сформувалась стратегія «нульової довіри» [1, 2]. І навпаки, при розгляді цієї стратегії виявляється, що її метою є зміщення парадигми від одноразової перевірки до парадигми постійної [3]. Передумовою цієї ініціативи є як створення довірених моделей, так і створення довіреного середовища.

7

Суть удосконалення методу:

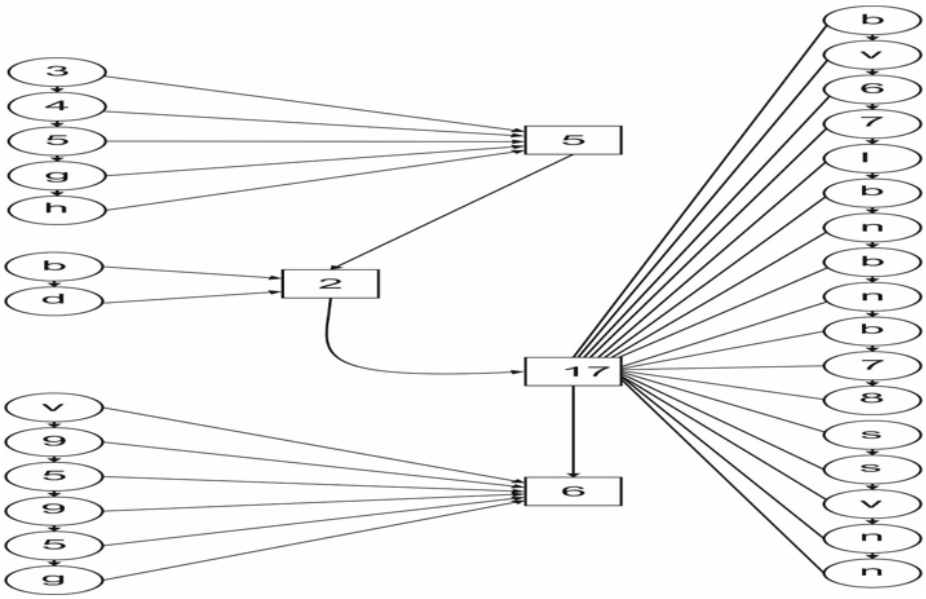
Вибравши цифровий підпис переправляємо його за допомогою функції, щоб отримати результуючий геш-рядок: 345ghbdbv67lbnbnb78ssvnnv9595g. Якщо розглядати можливі конфігуровані довжини файлу, то набір переправлюваних рядків для трансформування наведено в табл. 2.1. В той час як довжина файлу зазвичай не використовується в існуючому методі, то для ілюстрації використаємо саме її.

Шифр-таблиця

Текст	Шифр
5	345gh
2	bd
17	bv67lbnbnb78ssvnn
6	v9595g

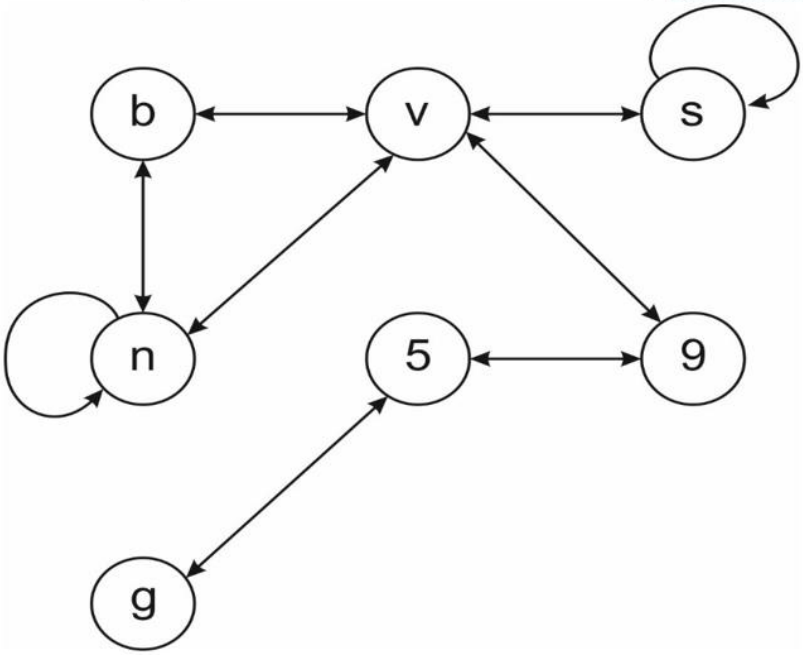
8

- Побудова станів символів і суміжностей – приклад:



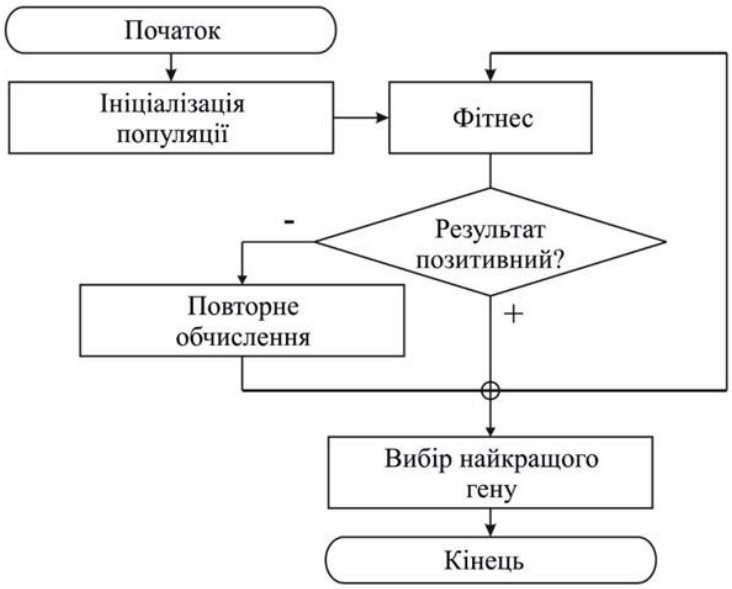
9

Представлення графіка символьних станів та списку суміжностей



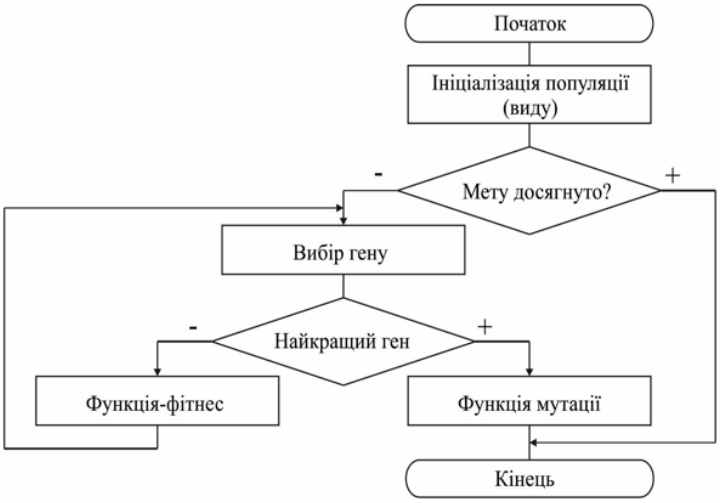
10

Модель генетичного алгоритму



11

Схема розширеної моделі ГА



12

Метод може містити такі наступні кроки:

1) вибір тексту для атаки;

2) вибір звичайного тексту для використання його як сторонніх знаків;

3) сформувати початковий текст з вставленням звичайного тексту, який додається;

4) зашифрувати два тексти: початковий без змін; початковий текст із вставленням сторонніх знаків;

5) визначення електромагнітного випромінювання;

6) обчислення різники електромагнітної індукції;

7) генерація ключових кандидатів в зашифрованих текстах;

8) застосування формули для визначення побітових раундів:

9) якщо на кроці 8 не встановлено фрагменти зі сторонніми знаками, тоді повторити кроки 5-8;

10) якщо встановлено фрагменти зі сторонніми знаками, тоді завершити роботу.

13

ВИСНОВКИ

- У роботі за результатами виконаних теоретичних та практичних досліджень розроблено новий метод забезпечення пропускної здатності дисків для застосунків з інтенсивним обсягом даних та отримано такі результати.
- 1. Проаналізовано відомі методи та засоби криптографічного захисту від вразливостей в апаратному забезпеченні.
- 2. Розроблено удосконалення методу криптографічного захисту від вразливостей в апаратному забезпеченні.
- 3. Здійснено реалізацію розробленого методу криптографічного захисту від вразливостей в апаратному забезпеченні.
- 4. Здійснено еспериментальні дослідження згідно розроблених рішень.

14

ДОДАТОК Б

(обов’язковий)

НАУКОВА ПРАЦЯ ЗДОБУВАЧА

Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп’ютерних наук АПКН-2023»

17-18 листопада 2023

Хмельницький 2023

Слутяк Є.І., Радельчук Г.І., Балицький Б.І.
Удосконалення передачі даних у мережі інтернет з використанням алгоритму верифікації повідомлень 274

Смірнов О.П., Поплавський С.Ю., Ковальчук В.К., Лутюк Л.І.
Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні 278

Смолієнко Д.В., Петровський С.С.
Метод прогнозування забруднення громадських доріг на основі підходу глибокого активного навчання 280

Собко В.В.
Метод поєднання технологій Redux-Toolkit та Redux-Saga для роботи з API веб-ресурсів 284

Собкова Ю.В., Міхалевський В.Ц., Скрипник Т.К.
Метод автоматизованого підбору тимчасового житла для категорій споживачів за генетичним алгоритмом 286

Стецюк Ю.В.
Методи обробки даних з обмеженим доступом в мультикомп'ютерних системах із застосуванням хмарних технологій їх зберігання 289

Тоцький О.П.
Методи обробки кардіограм 293

Уваров В.С., Чабан О.Р., Манзюк Е.А.
Метод діагностики захворювань серця на основі аналізу зображень, отриманих методом магнітно-резонансної томографії 296

Федоренко В.В., Пасічник О.А., Скрипник Т.К.
Технологія блокчейн у сфері реєстрації майнових прав 300

Хміль О.О, Праворська Н.І.
Веб-сайт біржі фрілансу 304

Швайко В.К., Ільчишина Ю.В.
Метод вибору виду спорту на основі морфофункціональних показників людини 308

Шебетко О.В., Кліменко В.І., Мазурець О.В.
Метод адаптивного тестування з використанням продукційних правил 311

УДК 004.5

Смірнов О.П., Поплавський С.Ю., Ковальчук В.К., Лутюк Л.І.

Хмельницький національний університет

УДОСКОНАЛЕНИЙ МЕТОД ТА ЗАСОБИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ВІД ВРАЗЛИВОСТЕЙ В АПАРАТНОМУ ЗАБЕЗПЕЧЕННІ

Удосконалено метод криптографічного захисту від вразливостей в апаратному забезпеченні. Для запобігання крадіжці пропонується удосконалений метод і алгоритм відображення. Розроблений криптографічний алгоритм можна зламати за допомогою диференціальної атаки відкритого тексту із значно обмеженими ресурсами. Нарешті, щоб запобігти таким атакам, представляємо серію контрзаходів для всіх мережевих шифрів із заміною та перестановкою, шляхом стимулювання нестатичної поведінки. Представляємо нову мережу взаємозв'язку мережі динамічної маршрутизації та модифіковані інваріантні елементи раунду на варіанти на основі раунду, які вимагають прийняття рішення зловмисником. Показано практичність і можливість виконання такої атаки на реальні фізичні реалізації.

The method of cryptographic protection against vulnerabilities in hardware has been improved. To prevent theft, an improved display method and algorithm is proposed. The developed cryptographic algorithm can be broken using a differential plaintext attack with significantly limited resources. Finally, to prevent such attacks, we present a series of countermeasures for all network substitution and permutation ciphers by encouraging non-static behavior. We present a new dynamic routing network interconnection network and modified round-invariant elements on round-based variants that require an attacker to make a decision. The practicality and possibility of performing such an attack on real physical implementations is shown.

Протягом останніх десятиліть апаратне забезпечення все частіше піддається складним атакам [1-3]. Немає явної впевненості в тому, що виробники/постачальники не виробляють [4-6] підроблені проекти або що криптографічні алгоритми безпечні в програмному забезпеченні, також безпечні в апаратному забезпеченні. Новизна роботи полягає в наступному: вдосконалений метод захисту, який забезпечує підхід для перевірки права власності, демонстрація того, як зламати криптографічний алгоритм із значно обмеженими можливостями ресурсів і надати безліч апаратних засобів протидії таким атакам.

Схеми кодування стану впливають на синтез певним чином. Одне значення кодування стану має здатність повністю видалити синтезовану систему з діапазону допустимих накладних витрат. За допомогою коду стану тепер заглиблюються в проблему багатометричної оптимізації на багатьох напрямах. Аналізується розмір, довжина кодування, вільні значення кодування та підпис для вибору. Кожен з них

може негативно вплинути на результат не лише складності графа запиту, але й кодів стану, які застосовуються та вибираються вільно.

Щоб запобігти крадіжці, пропонується удосконалений метод і алгоритм відображення. В ньому застосовуємо гібридний генетичний алгоритм, який отримав назву для ефективного вирішення складної проблеми зіставлення підграфів. У результаті покращення складає максимум на 10-12%. Розроблений криптографічний алгоритм можна зламати за допомогою диференціальної атаки відкритого тексту із значно обмеженими ресурсами. Нарешті, щоб запобігти таким атакам, представляємо серію контрзаходів для всіх мережевих шифрів із заміною та перестановкою, шляхом стимулювання нестатичної поведінки. Представляємо нову мережу взаємозв'язку мережі динамічної маршрутизації та модифіковані інваріантні елементи раунду на варіанти на основі раунду, які вимагають прийняття рішення зловмисником. Показано практичність і можливість виконання такої атаки на реальні фізичні реалізації, які вирішують використовувати цю технологію архітектури в апаратному забезпеченні. Крім того, за допомогою мереж перестановок із «ключами» такій атаці можна або повністю запобігти, або призвести до додаткового часу та труднощів для зловмисника, щоб відновити секретний ключ. Крім того, представлена мережева техніка перестановки з «ключем» є загальною і спрямована лише на зміну самої функції перестановки. Він має незначний загальний вплив і багато архітектур шифру, що використовують структури SPN подібним чином, можуть використовувати представлену техніку, так і поєднувати її з додатковими методами захисту.

Перелік посилань

1. Counterfeit integrated circuits: detection and avoidance. Springer, 2015.
2. U.S. Department of Defense Partners with GLOBALFOUNDRIES to Manufacture Secure Chips at Fab 8 in Upstate New York, Feb 2021.
3. N.A. Beresford, C.L. Barnett, S. Gashchak, A. Maksimenko, E. Guliaichenko, M.D. Wood, and M. Izquierdo. Radionuclide transfer to wildlife at a 'Reference site' in the Chernobyl Exclusion Zone and resultant radiation exposures. Journal of Environmental Radioactivity, 211:105661, January 2020.
4. Matthew Lewandowski and Srinivas Katkoori. Lightweight Countermeasure to Differential-Plaintext Attacks on Permutation Ciphers. In Augusto Casaca, Srinivas Katkoori, Sandip Ray, and Leon Strous, editors, Internet of Things. A Confluence of Many Disciplines, pages 159–176, Cham, 2020. Springer International Publishing.
5. Fukang Liu, Christoph Dobraunig, Florian Mendel, Takanori Isobe, Gaoli Wang, and Zhenfu Cao. Efficient Collision Attack Frameworks for RIPEMD-160. In Alexandra Boldyreva and Daniele Micciancio, editors, Advances in Cryptology – CRYPTO 2019, pages 117–149, Cham, 2019. Springer International Publishing.
6. Zachary J. McDowell and Matthew A. Vetter. It Takes a Village to Combat a Fake News Army: Wikipedia's Community and Policies for Information Literacy. Social Media + Society, 6(3):205630512093730, 2020.

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016292508

Дата перевірки:
29.05.2024 05:21:26 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
29.05.2024 07:10:59 EEST

ID користувача:
100005591

Назва документа: **Смірнов_Удосконалений метод та засоби криптографічного захисту від вразливостей в ап...**

Кількість сторінок: **87** Кількість слів: **21508** Кількість символів: **162979** Розмір файлу: **935.38 KB** ID файлу: **1016086536**

1.34%
Схожість

Найбільша схожість: **0.8%** з джерелом з Бібліотеки (ID файлу: **1016010041**)

0.91% Джерела з Інтернету	74	Сторінка 89
1.08% Джерела з Бібліотеки	88	Сторінка 90

0.01% Цитат

Цитати	4	Сторінка 91
Посилання	1	Сторінка 91

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	9

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 7%**

	Документ		Сумарний збіг по Базі Даних	
ID: 127611 Назва: МКР Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні Додано в БД: 2024-05-28 Автора: Олексій СМІРНОВ Керівники: Катерина БЕРЕЗЬКА Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	147719	920	4972 (3%)	61 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Олексій СМІРНОВ

Тема: Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень -; кількість сторінок записки 97

1. Короткий зміст роботи та прийнятих рішень У роботі розроблено удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, наукову новизну та практичну цінність роботи, а також характеристику структури роботи.

У першому розділі проведено аналіз відомих рішень щодо криптографічного захисту від вразливостей в апаратному забезпеченні.

У другому розділі здійснено дослідження предметної області та визначено стратегію забезпечення криптографічного захисту від вразливостей в апаратному забезпеченні.

У третьому розділі розроблено спосіб криптографічного захисту від вразливостей в апаратному забезпеченні. Його реалізація базується на використанні апаратного пристрою. Також, розроблено метод криптографічного захисту від вразливостей в апаратному забезпеченні.

У четвертому розділі здійснено здійснено розроблення криптографічного захисту від вразливостей в апаратному забезпеченні.

У висновках підведено підсумки досягнення результатів з розв'язання завдань дослідження.

4. Позитивні сторони роботи: ___

5. Негативні сторони роботи: немає.

6. Оцінка графічного оформлення та пояснювальної записки роботи: __=_____

7. Відгук про роботу в цілому: __Робота виконана на належному рівні.

8. Інші зауваження: ___=_____

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) __Корецька Людмила Олександрівна, к.т.н., доцент кафедри АКІТР ХНУ

"_24_" __травня__2024р. _____

Завідувачу кафедри КІІС
д-р.техн.наук, проф. Тетяні ГОВОРУЩЕНКО

Олексій СМІРНОВ
ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-22-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

30 березня 2024 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Удосконалений метод та засоби криптографічного захисту від вразливостей в апаратному забезпеченні

Автор: Олексій СМІРНОВ

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Катерина БЕРЕЗЬКА, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
2) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає менше 4% і адресується до джерел з інтернету та бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру завдання і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Катерина БЕРЕЗЬКА

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КІІС _____ Тетяна ГОВОРУЩЕНКО