УДК 004.056:061.68:004.375:061.68
УКПП
№ держреєстрації 0115U002431
Інв. №

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
Харківський національний університет радіоелектроніки
61166, м. Харків, пр. Науки,14; тел./факс. (057)702-13-97

ЗАТВЕРДЖУЮ
Ректор
Харківського національного
університету радіоелектроніки
д.т.н., проф.

_____В.В.Семенець

М.П.

ЗВІТ
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ
МЕТОДИ, СИСТЕМИ І ЗАСОБИ ІНФОРМАЦІЇ З ГАРАНТОВАНИМ РІВНЕМ СТІЙКОСТІ ТА ПІДВИЩЕНОЮ ШВИДКОДІЄЮ
№295
(заключний)

Науковий керівник НДР
д-р техн. наук, Г.З.Халімов

2017

Рукопис закінчено « » грудня 2017 р.

Результати роботи розглянуто науково-технічною радою ХНУРЕ
протокол від 26 грудня 2017 р. № 7/2

СПИСОК АВТОРІВ

Науковий керівник НДР зав. кафедрою БІТ д.т.н., професор	_____ “__” _____2017 р	Халімов Г.З.
Відповідальний виконавець роботи доцент кафедри БІТ к.т.н., доцент	_____ “__” _____2017 р	Гріненко Т.О.
Виконавці:		
професор кафедри БІТ д.т.н., доцент	_____ “__” _____2017 р	Руженцев В.І.
професор кафедри БІТ д.т.н., професор	_____ “__” _____2017 р	Горбенко І.Д.
професор кафедри БІТ д.т.н., професор	_____ “__” _____2017 р	Рубан І.В.
професор кафедри БІТ д.т.н., доцент	_____ “__” _____2017 р	Олійников В.Р.
професор кафедри БІТ д.т.н., доцент	_____ “__” _____2017 р	Замула О.А.
професор кафедри БІТ д.т.н., професор	_____ “__” _____2017 р	Лисицька І.В.
професор кафедри БІТ к.т.н., доцент	_____ “__” _____2017 р	Заболотний В.І.
доцент кафедри БІТ к.т.н	_____ “__” _____2017 р	Мельникова О.А.
доцент кафедри БІТ к.т.н., доцент	_____ “__” _____2017 р	Федюшін О.І.
доцент кафедри БІТ к.т.н., доцент	_____ “__” _____2017 р	Сєвєрінов О.В.
м.н.с, к.т.н.	_____ “__” _____2017 р	Котух Е.В.

м.н.с, к.т.н.	_____	Кайдалов Д.С.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Стеценко П.І.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Перекопський О.О.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Сергейчук Ю.О.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Халімов О.Г.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Цапко Д.П.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Мартовицький В.О.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Худов В.Г.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Перепадя В.І.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Черниш В.І.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Кохан С.А.
	“__” _____2017 р	
аспірант кафедри БІТ	_____	Штефан О.А.
	“__” _____2017 р	
студент	_____	Малєєва Г.
	“__” _____2017 р	

РЕФЕРАТ

Звіт містить 6 розділів. Повний обсяг складає 473 сторінок, у тому числі 423 сторінок основного тексту, 63 таблиці, 90 рисунків. Перелік посилань містить 313 найменуваннь.

У звіті містяться узагальнені результати виконання третього етапу НДР «Методи, системи і засоби криптографічного захисту інформації з гарантованим рівнем стійкості та підвищеною швидкодією»:

- розробка математичних моделей проведення ефективних криптоаналітичних атак на основі таблиць передобчислень (із застосуванням точок розрізнення, rainbow та fuzzy-rainbow) на симетричні криптопримітиви успадкованого рівня стійкості й обґрунтування параметрів перспективних алгоритмів;

- розробка методу порівняння високорівневих конструкцій симетричних блокових шифрів на основі ланцюга фейстеля, схеми лей-мессі і SPN-структури з відповідною кількістю раундів для оцінки ефективності основних конструкцій, та обґрунтування вибору структури перспективного алгоритму шифрування

- вдосконалення методів синтезу ефективних в реалізації схем генерації циклових ключів симетричних блокових шифрів для захисту від атак на зв’язаних ключах;

- вдосконалення методів оцінки стійкості Симетричних блокових шифрів до диференціального криптоаналізу, до атак усічених та нездійсненних диференціалів;

- розробка методів побудови криптосистем над групами;

- розробка методів побудови криптоперетворень та оцінка стійкості на основі обчислень над функціональними полями проективних різноманіть.

РЕФЕРАТ

Отчет содержит 6 разделов. Полный объем составляет 473 страниц, в том числе 423 страниц основного текста, 63 таблицы, 90 рисунков. Перечень ссылок содержит 313 наименований.

В отчете содержатся обобщенные результаты выполнения третьего этапа НИР «Методы, системы и средства криптографической защиты информации с гарантированным уровнем устойчивости и повышенным быстродействием»:

- разработка математических моделей проведения эффективных криптоаналитических атак на основе таблиц предвычислений (с применением точек различия, rainbow и fuzzy-rainbow) на симметричные криптопримитивы унаследованного уровня стойкости и обоснование параметров перспективных алгоритмов;

- разработка метода сравнения высокоуровневых конструкций симметричных блочных шифров на основе цепи Фейстеля, схемы Лей-Месси и SPN-структуры с соответствующим количеством раундов для оценки эффективности основных конструкций, и обоснование выбора структуры перспективного алгоритма шифрования

- совершенствование методов синтеза эффективных в реализации схем генерации цикловых ключей симметричных блочных шифров для защиты от атак на связанных ключах;

- совершенствование методов оценки устойчивости симметричных блочных шифров к дифференциальному криптоанализу, атакам усеченных и невыполнимых дифференциалов;

- разработка методов построения криптосистем над группами;

- разработка методов построения криптопреобразований и оценка устойчивости на основе вычислений над функциональными полями проективных многообразий.

ABSTRACT

The report contains 6 sections. The full volume is 473 pages, including 423 pages of the main text, 63 tables, 90 figures. The list of links contains 313 titles.

The report contains the summarized results of the third stage of research work "Methods, systems and means of cryptographic protection of information with a guaranteed level of stability and high-speed performance":

- development of mathematical models of conducting effective cryptanalytic attacks based on tables of pre-calculations (using points of distinction, rainbow and fuzzy-rainbow) on symmetric cryptoprimitives of inherited level of stability and substantiation of parameters of advanced algorithms;

- the development of a method for comparing high-level structures of symmetric block ciphers based on a Feistel chain, Lai-Massey and a SPN-structure with an appropriate number of rounds for evaluating the effectiveness of the main constructions, and justifying the choice of the structure of a advanced encryption algorithm;

- improvement of methods of synthesis of effective circuits for the generation of round keys of symmetric block ciphers for protection against attacks on associated keys;

- improving the methods for assessing the stability of the symmetric block ciphers to differential cryptanalysis, to the attacks of truncated and unrealizable differentials;

- development of methods for constructing cryptosystems over groups;

- development of methods for constructing cryptographic transformations and assessment of stability based on calculations over functional fields of projective varieties.

ЗМІСТ

ВСТУП

Побудова криптографічних перетворень з гарантованим рівнем стійкості в умовах ресурсних обмежень визначається вирішенням ряду наукових задач, щодо розробки математичних моделей атак на симетричні та асиметричні криптосистеми, методів аналізу та синтезу крипто перетворень, які задовольняють вимогам стійкості та реалізації.

Метою роботи є обґрунтування принципів побудови та оцінки властивостей криптографічних перетворень, що задовольняють вимоги щодо стійкості і швидкодії.

Напрями роботи: розробка математичних моделей криптоаналітичних атак на основі передобчислень на симетричні криптопримітиви; методу порівняння високорівневих конструкцій симетричних блокових шифрів, оцінка ефективності та обґрунтування структури алгоритму шифрування; вдосконалення методів оцінки стійкості блокових шифрів до диференціального крипто аналізу; розробка методів аналізу стійкості асиметричних криптографічних перетворень, які базуються на алгебричних системах еліптичних кривих, кілець зрізаних поліномів, матричних (лінійних) груп;. методів побудови криптографічних систем на основі обчислень над функціональними полями проективних різноманіть та оцінка стійкості.

Очікувані результати: сукупність моделей та методів побудови низки атак переборного типу та диференційних атак на симетричні криптографічні перетворення, побудови асиметричних криптосистем стійких до атак з використанням хмарних обчислень та квантових обчислень.

Об’єкт дослідження – процеси захисту інформації із використанням криптографічних перетворень та їх властивості.

Предмет дослідження – рівень стійкості та ефективність реалізації існуючих та перспективних криптографічних перетворень.

Фундаментальна проблема, на вирішення якої спрямовано проект полягає в розробці теорії побудови криптографічних перетворень, що забезпечують необхідний рівень стійкості щодо алгебричних методів криптоаналізу, високу швидкодію і компактність реалізації в умовах обмеження часових, дослідницьких і обчислювальних ресурсів під час розробки перспективних криптографічних алгоритмів.

Найбільш ефективні сучасні атаки на симетричні криптосистеми реалізуються на основі методів диференційного та лінійного криптоаналізу Е.Бихома і М.Матсуі. Основною проблематикою є необхідність мати велику кількість пар відкритих текстів-криптограм. Складність криптоаналізу є експоненційною від розміру ключових даних. В роботах А.М.Олексійчук та Л.В.Ковальчук показано, що оцінки безпеки шифрів прямо пов'язуються з затраченими ресурсами час/простір. Вперше лінійну апроксимацію блочних шифрів запропонувала К.Нуберг. Ідея масштабування для зменшення складності криптоаналізу широко використана в роботах В.І.Долгова, І.Д.Горбенка, Р.В.Олійникова, І.В.Лисицької, В.І.Руженцева. Іншим механізмом підвищення швидкості криптоаналізу є використання передобчислень на основі застосування моделі Ґеллмана, точок розрізнення, rainbow-таблиць та fuzzy-rainbow-таблиць. Застосування rainbow-таблиц дозволяє отримати мінімальну складність етапу передобчислень для таблиць, з обсягом близьким до потужності множини ключів.

Атаки на асиметричні криптосистеми за останні двадцять п’ять років призвели до появи алгоритмів субекспоненційної складності. Використання хмарних обчислень вимагає збільшувати довжину безпечної довжини ключів кожні декілька років. Алгоритм П.Шора (1997р.) допускає високий рівень розпаралелювання та має поліноміальну складність рішення проблем факторизації та дискретного логарифмування. Загроза появи квантових комп’ютерів призводить до необхідності перегляду безпеки криптопротоколів стійкість яких базується на вирішенні цих задач.

Концептуальними положеннями вирішення фундаментальної проблеми

проекту є наступні:

1. Застосування методу передобчислень для криптоаналізу симетричних криптоперетворень дозволяє отримати компромісне рішення досягнення високої ймовірності успіху, порівняно малий час проведення атаки і припустимий рівень складності на етапі передобчислень. Додаткове експоненційне зниження складності попереднього етапу можливо досягти за рахунок лінійного збільшення складності часу виконання атаки. Обґрунтування вибору структури перспективного алгоритму шифрування можливо на основі розробки математичних моделей криптоаналітичних атак з використанням rainbow-таблиць, fuzzy-rainbow таблиць передобчислень, методу порівняння високорівневих конструкцій симетричних блокових шифрів.

2. Криптоаналіз на основі усічених диференціалів та нездійсненних диференціалів потенційно розширює пошук змін в текстах на кроках обчислення циклових функцій. Ефективність таких методів доказана при оцінці кандидатів на стандарт гешування SHA 3 і являється продуктивним при невеликій кількості раундових перетворень. Вдосконалення методів оцінки стійкості симетричних блокових шифрів до диференціального криптоаналізу, атаки усічених диференціалів, атаки нездійсненних диференціалів надає можливість оцінити колізійні властивості циклових перетворень симетричних крипто перетворень.

3. Циклічні та інші кінцеві групи (групи точок еліптичних кривих, матричні групи, групи кіс Артіна, полугрупи, лупи), як алгебричні структури, широко використовуються в криптографічних протоколах типу Діффі-Хелмана-Меркля. Існує 44 прості та спорадичні групи, які потенційно можна використати в асиметричній криптографії. Для багатьох протоколів де використовуються скінченні алгебричні структури можливий перехід до матричної алгебри. Для криптосистем з матричним групами існує універсальний алгоритм криптоаналізу на основі апарату лінійної алгебри. Практичне використання хмарних обчислень створює пряму загрозу

сучасним асиметричним криптосистемам. Розробка методів аналізу стійкості асиметричних криптографічних систем, побудови загальносистемних параметрів, прискорення часової та просторової складності асиметричних криптоперетворень, які базуються на використанні еліптичних кривих, кілець зрізаних поліномів, матричних (лінійних) груп над полями, кільцями, алгебрами дозволить забезпечити необхідний рівень стійкості щодо алгебричних методів криптоаналізу.

Поставлена мета досягається вирішенням таких науково-технічних задач:

- розробка математичних моделей проведення ефективних криптоаналітичних атак на основі таблиць передобчислень (із застосуванням точок розрізнення, rainbow та fuzzy-rainbow) на симетричні криптопримітиви успадкованого рівня стійкості й обґрунтування параметрів перспективних алгоритмів;

- розробка методу порівняння високорівневих конструкцій симетричних блокових шифрів на основі ланцюга Фейстеля, схеми Лей-Мессі і SPN-структури з відповідною кількістю раундів для оцінки ефективності основних конструкцій, та обґрунтування вибору структури перспективного алгоритму шифрування;

- вдосконалення методів синтезу ефективних в реалізації схем генерації циклових ключів симетричних блокових шифрів для захисту від атак на зв’язаних ключах;

- вдосконалення методів оцінки стійкості симетричних блокових шифрів до диференціального криптоаналізу, до атаки усічених диференціалів, до атаки нездійсненних диференціалів;

- розробка методів аналізу стійкості асиметричних криптографічних систем, які базуються на використанні еліптичних кривих, кілець зрізаних поліномів, матричних (лінійних) груп над полями, кільцями, алгебрами;.

- розробка методів прискорення часової та просторової складності асиметричних криптографічних систем, які базуються на використанні

еліптичних кривих, кілець зрізаних поліномів, матричних (лінійних) груп над полями, кільцями, алгебрами;

- розробка методів побудови загальносистемних параметрів асиметричних криптографічних систем, які базуються на використанні еліптичних кривих, кілець зрізаних поліномів, матричних (лінійних) груп над полями, кільцями, алгебрами;

- розробка методів побудови криптоперетворень та оцінка стійкості на основі обчислень над функціональними полями проективних різноманіть.

1 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ПРОВЕДЕННЯ ЕФЕКТИВНИХ КРИПТОАНАЛІТИЧНИХ АТАК НА ОСНОВІ ТАБЛИЦЬ ПЕРЕДОБЧИСЛЕНЬ (ІЗ ЗАСТОСУВАННЯМ ТОЧОК РОЗРІЗНЕННЯ, RAINBOW ТА FUZZY-RAINBOW) НА СИМЕТРИЧНІ КРИПТОПРИМІТИВИ УСПАДКОВАНОГО РІВНЯ СТІЙКОСТІ Й ОБҐРУНТУВАННЯ ПАРАМЕТРІВ ПЕРСПЕКТИВНИХ АЛГОРИТМІВ

Стійкість сучасних криптографічних систем розглядається тільки при наступних мінімальних умовах для криптоаналітика [1, 2]:

– криптоаналітик має доступ до каналу зв'язку або пристрою зберігання (пасивний або активний в залежності від різновиду атаки);

– криптоаналітику відомий опис існуючих криптографічних перетворень, тощо, за винятком секретного ключа шифрування (принцип Кергоффса).

Відповідно до цієї моделі, стійка криптографічна система забезпечує захист в умовах передачі інформації за межами контрольованої території і в рамках моделі загроз, коли засіб криптографічного захисту може виявитися в руках зловмисника (при цьому потрібною є тільки зміна ключа, але не заміна всього криптографічного обладнання).

Атаки можуть здійснюватися як пасивно (прослуховування каналу зв'язку або створення копій захищених повідомлень, що зберігаються), так і активно (зі спробами відправки зловмисником власних повідомлень в канал зв'язку, на вхід шифратора тощо).

При аналізі схем шифрування розглядають такі типи атак [1, 2]:

– тільки на основі шифртексту;

– з відомими відкритими текстами;

– з вибраними відкритими текстами;

– з вибраними шифртекстами;

– з адаптивно обраними відкритими або шифртекстами.

Сучасні симетричні примітиви (блокові шифри, хеш-функції) можуть використовуватися як базові блоки для інших криптографічних перетворень, таких як протоколи аутентифікації і ін. Тому перспективні криптографічні алгоритми повинні проектуватися для використання з урахуванням принципу Кергоффса і забезпечувати стійкість до атак на основі адаптивно обраних відкритих або шифртекстів.

З точки зору забезпечення стійкості шифри діляться на наступні класи [1-3]:

– безумовно стійкі (теоретично недешифруємі);

– обчислювально стійкі;

– ймовірнісно стійкі;

– обчислювально нестійкі.

Теоретично недешифруємі системи, хоча і забезпечують високий рівень захищеності, вкрай незручні для використання через громіздку і незручну схему управління ключами. Відповідно, перспективні симетричні примітиви повинні забезпечувати обчислювальну стійкість в моделі застосування сучасних атак на основі передобчислення (rainbow-таблиці, fuzzy-rainbow-таблиці та ін.).

1.1 Умови проведення криптоаналітичної атаки

Для забезпечення високого рівня стійкості симетричні криптографічні примітиви повинні моделювати властивості випадкових відображень [1-7], таких як випадкові перестановки (блокові шифри) і випадкові функції (хеш-функції, генератори псевдовипадкових послідовностей, потокові шифри).

Навіть при наявності високої надмірності деякої вхідної послідовності симетричного криптографічного примітиву його вихідна послідовність повинна відповідати критеріям випадковості, рівновірогідності і незалежності. Для шифруючих перетворень ця вимога виводиться з

необхідних властивостей теоретико-інформаційної моделі [3]. Аналогічні результати справедливі і для хеш-функцій, ГПСЧ і ін. [2].

Більшість широко поширених криптографічних перетворень відповідають відомим вимогам випадковості (псевдовипадковості) і успішно проходять статистичне тестування вихідних послідовностей для різних наборів текстів, зокрема, NIST STS, AIS 20 і ін. Відповідно, розробка і реалізація критеріїв, що дозволяють відрізнити вихідну послідовність сучасного симетричного криптопримітиву від випадкової, є практично неможливою. Як правило, алгоритми розрізнення для повних (НЕ усічених) варіантів перетворень мають експоненційну складність. Наприклад, навіть 5 раундів ланцюга Фейстеля з розміром блоку $2n$ біт можна відрізнити від випадкової перестановки зі складністю $O(2^{\frac{3n}{2}})$ при наявності $O(2^{\frac{3n}{2}})$ пар обраного відкритого і відповідного шифрованого текстів [8]. При цьому слід зазначити, що поширені блокові симетричні шифри, які використовують ланцюг Фейстеля, виконують 16 або більше раундів перетворення. Статистичні методи криптоаналізу (диференційний, лінійний і ін.) Для поширених сучасних симетричних перетворень також мають вкрай високу обчислювальну складність (яку можна порівняти з повним перебором), що додатково підтверджує гарну відповідність моделі випадкової функції або перестановки.

Для виключення впливу властивостей різних моделей джерел відкритих повідомлень [4] при аналізі стійкості криптографічних примітивів розглядаються, як правило, атаки з відомими або обраними відкритими текстами. Як виняток, для дешифрування можливий розгляд атак на основі шифртекстів, але більшість методів, описаних у відкритій літературі, вимагають будь-якої додаткової інформації.

Використання такого підходу дозволяє виконувати оцінку властивостей саме криптографічного перетворення в несприятливих умовах, коли криптоаналітику невідомий тільки один секретний параметр (ключ блочного

алгоритму, прообраз значення хеш-функції (хеш-коду), внутрішній стан поточного шифру). Якщо для реально експлуатованої криптографічної системи зловмисникові вдасться якимось неаналітичним способом (ПЕМВН, розкрадання документації тощо) отримати додаткову інформацію (крім шуканого секретного параметра), то це не знизить розрахункову стійкість криптографічного перетворення.

Тому оцінку стійкості симетричних криптографічних примітивів доцільно проводити для атак на основі обраних і / або відомих відкритих текстів.

1.2 Математичні моделі симетричних криптографічних перетворень

Симетричний шифр може бути представлений у вигляді алгебраїчної системи [1-4]:

$$\Sigma_A = \langle X, K, Y, E, D \rangle, \tag{1.1}$$

де X – множина відкритих текстів над алфавітом шифрвелечин Q_X;

K – множина ключів (як правило, задається у вигляді рядків фіксованої довжини над алфавітом Q_K);

Y – множина шифртекстів над алфавітом шифрвизначень Q_Y;

$E: X \times K \to Y$ – множина правил шифрування на основі параметризованих відображень $e_k(x) = y,\ k \in K, x \in X, y \in Y$;

$D: Y \times K \to X$ – множина правил розшифрування на основі параметризованих відображень $d_k(y) = x,\ k \in K, x \in X, y \in Y$;

при цьому для будь-якого $k \in K$ відображення $e_k(\cdot)$ і $d_k(\cdot)$ є бієктивним і забезпечує виконання умови $d_k(e_k(x)) = x$ (як наслідок, $e_k(d_k(y)) = y$).

Для всіх поширених сучасних симетричних шифрів алфавіти відкритих і шифртекстів (шифрвелечін і шифрвизначень) збігаються ($Q_X = Q_Y$), відповідно, такі шифри є ендоморфними: $X = Y$.

Як вже зазначалося вище, для криптографічно стійких шифрів безліч правил шифрування і розшифрування повинні мати властивості випадкового відображення, а $e_k(\cdot)$ і $d_k(\cdot)$ – випадкових перестановок.

Хеш-функція як алгебраїчна система представляється у вигляді:

$$\Psi_A = \langle M, Y, H \rangle, \tag{1.2}$$

де M – множина повідомлень над алфавітом Q_M;

Y – кінцева множина значень хеш-функції, як правило, задається у вигляді безлічі рядків фіксованої довжини над алфавітом Q_Y;

$H : M \to Y$ – відображення хешування $h(m) = y$, $m \in M, y \in Y$.

Для поширених хеш-функцій $Q_M = Q_Y = \{0,1\}$.

Хеш-функція повинна забезпечувати властивості випадкового відображення [5], при цьому безліч значень є кінцевою, а безліч прообразів – нескінченною (за рідкісним винятком, коли довжина вхідного повідомлення обмежена, наприклад, в алгоритмі SHA-1).

Якщо хеш-функція забезпечує властивості випадкового відображення, то складність пошуку колізії або прообразу для ймовірності успіху $p = \frac{1}{2}$ оцінюється як $O\left(\sqrt{\#Y}\right)$.

Відзначимо, що хеш-функція є підалгеброю алгебри кодів аутентифікації повідомлень, при цьому потужність безлічі ключів дорівнює 1.

Код аутентифікації повідомлень (КАП) у вигляді алгебраїчної системи представляється як

$$\Xi_A = \langle M, K, Y, H \rangle, \tag{1.3}$$

де M – множина повідомлень над алфавітом Q_M;

K – множина ключів (як правило, задається у вигляді рядків фіксованої довжини над алфавітом Q_K);

Y – кінцева множина значень КАП, як правило, задається у вигляді безлічі рядків фіксованої довжини над алфавітом Q_Y;

$H: M \times K \to Y$ – безліч параметризованих відображень $h_k(m) = y$, $k \in K, m \in M, y \in Y$.

Для поширених кодів аутентифікації повідомлень на основі блокових симетричних шифрів або хеш-функцій $Q_M = Q_Y = Q_K = \{0,1\}$.

До кодів аутентифікації повідомлень, які використовуються в криптографії, пред'являються ті ж вимоги, що і до хеш-функцій, з урахуванням складності визначення застосовуваного ключа, невідомого зловмиснику.

Генератори псевдовипадкових чисел / послідовностей (ГПСЧ / ГПСП) або генератори гами (ГГ) представляються як

$$\Gamma_A = \langle V, K, Y, G \rangle, \tag{1.4}$$

де V – кінцева множина значень векторів ініціалізації, як правило, задається у вигляді рядка фіксованої довжини над алфавітом Q_V;

K – множина ключів, зазвичай задається у вигляді рядків фіксованої довжини над алфавітом Q_K;

Y – множина вихідних послідовностей, які подаються нескінченними рядками над алфавітом Q_Y, мають період і допускають наявність передперіода (tail);

$G: V \times K \to Y$ – множина параметризованих відображень $g_k(v) = y$, $k \in K, y \in Y$.

Відрізки рядків вихідних послідовностей (до виходу на період) криптографічно стійких генераторів повинні мати властивості випадкового відображення.

Для сучасних генераторів алфавіти векторів ініціалізації, ключів і вихідних послідовностей збігаються: $Q_V = Q_K = Q_Y$ (але довжини рядків, які задають вектори і ключі, як правило, різні).

Генератори гами є основними компонентами поточних шифрів, повністю визначають їх стійкість і криптографічні властивості.

Крім того, слід зазначити особливість побудови сучасних генераторів гами: безліч ключів K відображається в якійсь додатковій множині внутрішніх станів, яка служить аргументом для формування вихідної послідовності (гами): $G_0 : V \times K \to S$, $G_+ : S \to Y$, $G = G_+ \circ G_0$. Проміжна множина необхідна для ефективної технічної реалізації, але може викликати додаткові колізії при генерації (існування еквівалентних ключів, тощо).

Слід зазначити подібність моделей Ξ_A і Γ_A. Можливе створення розширення цих алгебр з множинами вхідних і вихідних параметрів над відповідними алфавітами, але без обмеження їх розміру (довжин вхідних / вихідних рядків).

Завдання обмеження на вихідні значення (обмеження потужності Y) призводить до завдання підалгебри, що визначає коди аутентифікації повідомлень: $\Xi_A \subset \mathrm{A}$. Аналогічне визначення множини вхідних параметрів $(I = V)$ задає підалгебру генераторів $\Gamma_A \subset \mathrm{A}$.

У моделі Кергоффса [1, 3, 4] при проведенні атаки перебором криптоаналітику відомий опис криптосистеми – відома відповідна алгебра $(\Sigma_A, \Psi_A, \Xi_A, \Gamma_A)$, тобто всі теоретично допустимі вхідні і вихідні послідовності, ключова множина, функції, що задають відображення входу в вихід і конкретне перехоплене вихідне значення (шифртекст, КАП і т.п.). Для атаки на основі обраних / відомих відкритих текстів криптоаналітик має додатковий параметр, і йому необхідно визначити деяке невідоме значення (табл. 1.1).

Таким чином, в даній моделі атаки зловмисникові невідомий тільки єдиний параметр: ключ шифру, КАП, генератор гами або ж прообраз значення хеш-функції.

Відповідно, криптоаналітик має деяке параметризоване відображення $F_\chi(x) = y$, деякі доступні йому (перехоплені) множини використаних вхідних

і вихідних параметрів $\{x_i\} \neq \varnothing$, $\{y_i\} \neq \varnothing$ (для хэш-функції $\#\{x_i\} = \#X = 1$), і його завдання – визначити невідомий параметр χ.

Таблиця 1.1 – Параметри криптосистеми, відомі криптоаналітику або задані ним при реалізації атаки

№	Примітив/ алгебра	Основний параметр	Додатковий параметр	Невідомий параметр
1	шифр Σ_A	шифртекст, $c_i \in Y$	відкритий текст, $m_i \in X$	ключ $k_i \in K$
2	хеш-функція Ψ_A	значення хеш-функції, $y_i \in Y$	–	прообраз $m_i \in M$
3	КАС Ξ_A	імітовставка, $y_i \in Y$	відкритий текст, $m_i \in M$	ключ $k_i \in K$
4	генератор гами Γ_A	блок гами, $\gamma_i \in Y$	вектор ініціалізації, $v_i \in V$	ключ $k_i \in K$

Один з варіантів практичного застосування цієї моделі наступний.

Для блокових шифрів: частина відкритого повідомлення (відмітка часу, ідентифікатор абонента, тощо) відома криптоаналітику або обрана ним (наприклад, при атаці на блоковий шифр у складі протоколу автентифікації). На основі перехопленого (отриманого) шифртекста криптоаналітику необхідно отримати використовуваний ключ.

Для поточних шифрів: на основі знання (або вибору) відкритого тексту (і синхровідправки / вектора ініціалізації) з подальшим перехопленням шифртекста криптоаналитик отримує блок гами шифрувальної. Надалі реалізується модель атаки на генератор гами.

Для хеш-функцій: підсистеми аутентифікації поширених операційних систем зберігають паролі користувачів в хешованому вигляді, доступному безпосередньо або через резервні копії системи. Завдання криптоаналітика –

по хешованому значенню отримати прообраз, тобто один з варіантів паролів, чий хеш-код дозволить успішно пройти автентифікацію.

Для КАП: за відомим (вибраному) відкритим повідомленням і його КАП визначити використовуваний ключ.

Для ГПСЧ / генераторів гами: маючи блок псевдовипадкової послідовності (гами), відновити внутрішній стан генератора і / або ключ ініціалізації, що дозволяє незалежно сформувати ПСП (для дешифрування частини повідомлення, визначення ключів інших примітивів, тощо).

Переборні атаки розглядають криптографічне перетворення як деякий «чорний ящик», без використання особливостей побудови, і орієнтовані на пошук невідомого параметра з урахуванням можливості оптимізації тимчасової або просторової складності, а також реалізації різних компромісів між ними.

1.3 Огляд підходів до реалізації атак переборного типу

В рамках даної моделі пошук невідомого параметра криптографічного примітиву може виконуватися за рахунок тимчасових / обчислювальних або просторових ресурсів. У першому випадку виконується повний перебір всіх можливих значень параметра, у другому-пошук в заздалегідь передобчислюваної таблиці. Крім того, припустимо комбінування методів для досягнення компромісів типу «час / пам'ять».

При реалізації атаки з повним перебором всі допустимі значення шуканого параметра послідовно апробуються на деякій обчислювальній системі. Порядок вибору значень для перевірки визначається законом розподілу вихідних послідовностей генератора криптографічного системи. Якщо закон розподілу є рівномірним (всі значення невідомого параметра різновірогідні або різниця ймовірностей незначна), то при переборі доцільно використовувати словниковий порядок. Часова складність реалізації такої

атаки пропорційна потужності множини невідомого параметра, просторова складність дорівнює нулю (не потрібна додаткова пам'ять).

Атака на основі передобчислення (словникова атака) вимагає побудови відсортованої таблиці, в яку на попередньому етапі заносяться відповідність між вихідними (при необхідності – і вхідними) і шуканими параметрами. Складність побудови такої таблиці для повної множини допустимих параметрів збігається зі складністю повного перебору і дорівнює $O(\#K)$. Виконання атаки полягає в пошуку в відсортованій таблиці відомого параметра і витяг відповідного шуканого. Тимчасова складність реалізації дорівнює $O(\log(\#K))$, просторова складність – $O(\#K)$.

Розмір таблиці для атаки на основі передобчислення і час її побудови можна скоротити за умови різновірогідності різних значень невідомого параметра на виході генератора криптографічної системи. Побудувавши таблицю з найбільш ймовірних значень шуканого параметра, можна побудувати досить ефективну криптоаналітичну систему при допустимому рівні вартості і складності. На цьому принципі засновані системи відновлення паролів з хеш-значень, що містять кілька мільйонів найбільш поширених користувальницьких паролів і їх хеші. Відповідно, для захисту від таких атак ряд сучасних операційних систем використовує рандомізацію паролів («salt»).

Методи комбінування просторових і часових атак (компроміс «час-пам'ять», time-memory trade-off, TMTO) припускають як наявність деякого об'єму заздалегідь передобчислюваних таблиць, так і перебір певної множини значень шуканого параметра на оперативному етапі. У ряді випадків такий підхід дозволяє знайти технічно реалізоване рішення з необхідним обсягом пам'яті менше, ніж для словникової атаки, і що проводиться швидше, ніж атака перебору. Слід зазначити, що TMTO-методи є імовірнісними (не гарантують відновлення довільного ключа, а тільки лише деяку їх підмножину, враховану на етапі передобчислення). Це обумовлено

тим, що складність побудови таблиць зростає експоненційно в залежності від ймовірності успіху, і побудова «досконалих» таблиць, які враховують всі ключі, є вкрай ресурсоємною і нівелює всі переваги ТМТО-методів.

Перший ефективний варіант реалізації був запропонований Хеллманом [9]. Для криптосистеми з N ключами цей метод вимагає обсяг пам'яті об'ємом $N^{\frac{2}{3}}$ слів, розділених між $N^{\frac{1}{3}}$ таблицями. Етап передобчислення вимагає N операцій прямого криптографічного перетворення. На оперативному етапі для відновлення невідомого ключа потрібно близько $N^{\frac{2}{3}}$ операцій прямого перетворення і пошуку в кожній з $N^{\frac{1}{3}}$ таблиць.

Таблиці складаються з рядків виду $k_0 \xrightarrow{F_{\delta_j,k_0}(\cdot)} k_1 \xrightarrow{F_{\delta_j,k_1}(\cdot)} \dots \xrightarrow{F_{\delta_j,k_{t-1}}(\cdot)} k_t$. Параметризована функція переходу F_{δ_j,k_i}, унікальна для кожної таблиці, дозволяє обчислити наступний враховуваний ключ шифрування через попередній із застосуванням прямого криптографічного перетворення. Зберігання тільки першого і останнього елемента послідовності дозволяє значно скоротити необхідний обсяг пам'яті, але на оперативному етапі, крім $t-1$ операцій пошуку серед кінцевих точок $\left\{k_t^{l,j}\right\}$ кожної таблиці, необхідно виконати таку ж кількість шифрувань.

Зі збільшенням розміру таблиці збільшується ймовірність появи колізійних елементів і злиття різних ланцюжків (різні рядки містять одні й ті ж послідовності). Для формування вибірки унікальних ключів на етапі передобчисленням виконується побудова великої кількості різних таблиць з унікальними функціями переходу.

Перевагами методу Хеллмана є можливість значного скорочення часової і просторової складності аналізу (в порівнянні з повним перебором і словниковою атакою), фіксований обсяг таблиць і постійнй час аналізу. До недоліків методу слід віднести неможливість (вкрай високу складність) виявлення злиття ланцюжків, що призводить до необхідності побудови

великої кількості таблиць, і збільшення часу (або складності обладнання) для пошуку ключа в кожній таблиці на оперативному етапі. Крім того, метод Хеллмана не враховує особливість, що симетричні криптографічні операції, як правило, виконуються значно швидше, ніж звернення до довготривалої пам'яті значного обсягу.

Модифікація цього методу була запропонована Ривестом [10]. Основною відмінністю є змінна довжина рядка, ознакою закінчення якої є фіксовані значення деяких бітів чергового елемента (вихід на так звану точку розрізнення, distinguishing point). Це дозволяє виявляти цикли і злиття ланцюжків на етапі побудови (таблиця складається тільки з унікальних елементів), а при проведенні криптоаналіза – зменшити кількість звернень до пам'яті за рахунок збільшення операцій криптографічного перетворення (які виконуються значно швидше, ніж звернення до довготривалої пам'яті). Метод на основі точок розрізнення також передбачає побудову великої кількості таблиць через високу ймовірність злиття ланцюжків (рядків).

Зменшити обчислювальну складність етапу передобчислень при збереженні кількості унікальних елементів і необхідного обсягу пам'яті дозволяє застосування rainbow-таблиць [11]. Метод передбачає побудову таблиці фіксованого розміру, але кожна колонка використовує унікальну функцію переходу. Колізії в різних колонках не призводять до злиття ланцюжків, а менш ймовірне злиття при збігу в одній і тій же колонці виявляється за співпадаючими кінцевим точкам. Перевагами методу є низька в порівнянні з описаними вище методами складність передобчислень, простота виявлення колізій і фіксований розмір таблиці. Недоліком методу є порівняно великий час аналізу в порівнянні з таблицями на основі точок розрізнення. Крім того, на момент проведення досліджень у відкритій пресі не було представлено адекватної математичної моделі при обсягах вибірки, близької до потужності ключової множини.

В роботі [12] обґрунтована верхня межа ефективності (ймовірності успішного відновлення і часу проведення оперативного етапу) для

довільного ТМТО-методу. Крім того, запропоновано декілька варіантів комбінування відомих методів. На основі rainbow-таблиць і методу Хеллмана запропоновані thick- і thin-rainbow (в залежності від правила зміни функції), а комбінація rainbow-таблиць і точок розрізнення призводить до fuzzy-rainbow-таблиць. Істотні переваги перед вже відомими методами має тільки останній. В [13] наводиться тільки співвідношення залежності параметрів для фіксованої ймовірності додавання крайнього рядка таблиці. Фактично, для запропонованих методів також відсутня адекватна математична модель, що дозволяє реалізувати атаку на практиці.

1.4 Модель проведення атаки перебору на симетричний криптографічний примітив

Як було зазначено вище, передбачається, що до початку проведення атаки криптоаналітику відомий опис криптосистеми – відповідна алгебра $\mathrm{A} \in \{\Sigma_A, \Psi_A, \Xi_A, \Gamma_A\}$, $\mathrm{A} = \langle X, K, Y, G \rangle$ з заданими множинами вхідних (X) і вихідних (Y) послідовностей, безліччю ключів або прообразів (K) і безліччю параметризованих відображень ($G = \{G_\chi\}: X \to Y, \chi \in K$). Надалі буде передбачатися, що слова цих множин задані над двійковим алфавітом, причому довжина невідомого параметра (ключа, прообразу) фіксована і дорівнює символам (бітам для двійкового алфавіту).

На початку атаки криптоаналітик отримує вхідне і відповідне вихідне значення, отриманих на основі використання одного невідомого параметра: $\{x_i\} \neq \varnothing$, $\{y_i\} \neq \varnothing$, $x_i \in X$, $y_i \in Y$, $G_\chi(x_i) = y_i$. Передбачається, що потужність множини отриманих вихідних значень достатня для досягнення відстані єдності.

Завдання криптоаналітика – на основі перебору або пошуку в таблиці передобчислених значень визначити невідомий параметр $\chi \in K$, використаний при формуванні вихідних послідовностей $G_\chi(x_i) = y_i$.

При розгляді атак на основі компромісу «час-пам'ять» (time-memory trade-off, TMTO) використовується функція переходу, що дозволяє реалізувати псевдовипадкове відображення елементів ключової безлічі на себе (обмеження на властивості сюр'єктивності або ін'єктивності не накладаються). Функція переходу повинна забезпечувати властивості випадкового відображення і, як правило, є композицією прямого криптографічного перетворення і додаткового параметризованих відображення:

$$F_q : K \to K,\ F_q = f_q \circ G_{k_i}, \qquad (1.5)$$

де G_{k_i} – пряме криптографічне перетворення, параметризовані $k_i \in K$;

$f_q : \Delta \times Y \to K$, $q \in \Delta$ – псевдовипадкове параметризоване відображення безлічі вихідних послідовностей в ключову множину.

Функція переходу призначена для побудови послідовності пар вихідних послідовностей і випадкових (псевдовипадкових) значень шуканого параметра без використання додаткової пам'яті (зберігається тільки початкова і кінцева точка послідовності).

Застосування функцій переходу дозволяє значно скоротити просторову складність (обсяг пам'яті) при реалізації атаки при збереженні потужності безлічі передобчислення значень.

Додаткове відображення f_q з різними параметрами використовується для зниження ймовірності появи колізій (злиття послідовностей з різними початковими точками). Крім того, якщо аналізований симетричний криптографічний примітив за деякими показниками НЕ задовольняє критеріям випадкового відображення, застосування f_q дозволяє отримати необхідні характеристики для всієї функції переходу F_q.

Приклади визначення додаткового параметризованого відображення f_q

– вибір з вихідної послідовності слова з довжиною, що дорівнює довжині слова елемента ключової множини, з подальшою перестановкою символів, що задається параметром q;

– вибір з вихідної послідовності слова з довжиною, що дорівнює довжині слова елемента ключової множини, з подальшим зашифруванням, причому додатковим параметром q є використовуваний ключ шифрування.

Перший варіант використовує допущення, що аналізований криптографічний примітив уже має властивості випадкового відображення. Другий варіант передбачає використання додаткового примітиву для забезпечення необхідних властивостей.

Далі розглянуті приклади визначення функції переходу для двох розглянутих вище симетричних криптографічних примітивів.

Для потокових шифрів: вихідний ключ довжиною l бітів завантажується в шифр, після чого виконується генерація гами шифрувальної (не менше l бітів). Отриманий блок гами зашифровується блоковим шифром на фіксованому ключі (заданим як додатковий параметр), причому довжина блоку становить як мінімум l бітів. Отриманий шифртекст усікається до довжини ключа поточного шифру, і розраховане значення є наступним ключем в послідовності.

Для блокових шифрів (довжина ключа не перевищує розмір блоку): вихідний ключ k бітів використовується для шифрування заданого відкритого тексту. Отриманий шифртекст при необхідності усікається до довжини ключа, після чого виконується бітова перестановка, яка визначається параметром q. Сформований ключ використовується як наступний в послідовності.

1.5 Оцінка ефективності таблиць на основі точок розрізнення

Таблиця на основі точок розрізнення є набором рядків (ланцюжків) виду $k_{i,1}^{(q)} \xrightarrow{F_q} k_{i,2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{i,\rho(i,q)}^{(q)}$. довжина ланцюжка $\rho(i,q)$ є випадковою величиною, розподіленою згідно з геометричним розподілом (визначення і властивості F_q див. в п. 1.4). Точкою розрізнення є чергове значення невідомого параметра (ключа) $k_{i,\rho(i,q)}^{(q)}$, що містить заздалегідь задане значення в d фіксованих символах (бітах). Як правило, для технічної реалізації найбільш зручно використовувати перевірку наявності d молодших (або старших) нульових бітів. Таблиця, що складається з m строк, представлена на рис. 1.1.

$$k_{1,1}^{(q)} \xrightarrow{F_q} k_{1,2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{1,\rho(1,q)}^{(q)}$$

$$k_{2,1}^{(q)} \xrightarrow{F_q} k_{2,2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{2,\rho(2,q)}^{(q)}$$

… … … … … … … …

$$k_{m,1}^{(q)} \xrightarrow{F_q} k_{m,2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{m,\rho(m,q)}^{(q)}$$

Рисунок 1.1 – Таблиця на основі точок розрізнення

Відповідно, довжини рядків в таблиці рівні $\rho(1,q)$, $\rho(2,q)$, …, $\rho(i-1,q)$ елементів. Кожен елемент таблиці є унікальним і колізії відсутні. Дійсно, колізія елементів різних рядків проявляються як колізія кінцевих точок ($k_{j,\rho(j,q)}^{(q)} = k_{u,\rho(u,q)}^{(q)}, j \neq u$), а на етапі побудови в таблицю додаються тільки рядки з унікальними кінцевими точками. У разі колізії елементів всередині одного рядка неможливий вихід на точку розрізнення (відбувається вхід в короткий цикл, де така точка відсутня).

Теорема 1.1 (ефективність таблиць на основі точок розрізнення).

Для таблиці на основі точок розрізнення довжиною d бітів, використовуваної для реалізації атаки переборного типу на симетричний криптографічний примітив з потужністю допустимих значень невідомого параметра 2^k, що складається з m рядків ($m \le 2^{k-d}$) з довжиною $\rho(1,q), \rho(2,q), \ldots, \rho(m-1,q)$ кожна:

на етапі передобчислення – математичне очікування верхньої межі складності побудови таблиці (кількість генеруємих рядків) не перевищує значення

$$\Theta^{dp} \le m \cdot \left(1 - \frac{1}{2^k}\sum_{j=1}^{m-1}\rho(j,q)\right)^{-\rho(m,q)} \cdot \left(1 - \frac{\rho(m,q)}{2^k}\right)^{-\frac{\rho(m,q)\cdot(\rho(m,q)-1)}{2}} \approx$$
$$\approx m \cdot e^{\frac{2^{2d}\cdot(2m-1)-2^d}{2^{k+1}}}; \quad (1.6)$$

на оперативному етапі – ймовірність успішного відновлення при випадковому, рівновірогідному і незалежному виборі невідомого параметра дорівнює:

$$P_{succ}^{dp} = \frac{1}{2^k} \cdot \sum_{j=1}^{m}\rho(j,q) \approx \frac{m}{2^{k-d}}. \quad (1.7)$$

Доказ.

Складність побудови таблиці.

При додаванні до таблиці, що містить $(i-1)$ рядків ($i \ge 2$) наступного, i-го рядка $k_{i,1}^{(q)} \xrightarrow{F_q} k_{i,2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{i,\rho(i,q)}^{(q)}$ з унікальною початковою точкою $k_{i,1}^{(q)} \ne k_{j,1}^{(q)}, 1 \le j < i$, можлива колізія кожного з $\rho(i,q)$ елементів, що додаються, з уже врахованими в таблиці. Таким чином, для кожного елемента i-го рядка можлива колізія з $\rho(1,q)+\rho(2,q)+\ldots+\rho(i-1,q) = \sum_{j=1}^{i-1}\rho(j,q)$ елементами. Імовірність появи

колізії для одного елемента який додається до рядка дорівнює $\frac{1}{2^k}\cdot\sum_{j=1}^{i-1}\rho(j,q)$ (див. доказ ймовірності успіху на оперативному етапі). Оскільки елементи таблиці формуються випадковою функцією F_q (див. п.2.4), то події виникнення колізії для кожного елемента $k_{i,j}^{(q)}$ рядка, що додається, є незалежними. Звідси ймовірність повної відсутності колізій з елементами рядків $1..(i-1)$ при додаванні i-го рядка, який складається з $\rho(i,q)$ елементів, дорівнює $\left(1-\frac{1}{2^k}\sum_{j=1}^{i-1}\rho(j,q)\right)^{\rho(i,q)}$.

Крім того, успішне додавання рядка можливе за умови відсутності колізії всередині елементів рядка (зациклення). В цьому випадку є $C^2_{\rho(i,q)}=\frac{\rho(i,q)\cdot(\rho(i,q)-1)}{2}$ пар елементів, і ймовірність відсутності колізії оцінюється як $\left(1-\frac{\rho(i,q)}{2^k}\right)^{\frac{\rho(i,q)\cdot(\rho(i,q)-1)}{2}}$. Як вже зазначалося, F_q – випадкова функція, події відсутності колізії з уже доданими рядками і відсутності циклу в новому рядку є незалежними, і ймовірність їх спільної появи дорівнює добутку ймовірностей кожної з подій.

Звідси ймовірність успішного додавання i-го рядка з виходом на нове (неповторне) значення кінцевої точки дорівнює $p_i^{dp}=\left(1-\frac{1}{2^k}\sum_{j=1}^{i-1}\rho(j,q)\right)^{\rho(i,q)}\cdot\left(1-\frac{\rho(i,q)}{2^k}\right)^{\frac{\rho(i,q)\cdot(\rho(i,q)-1)}{2}}$.

Математичне сподівання кількості спроб, необхідних для додавання до таблиці на основі точок розрізнення i-го рядка дорівнює $\theta_i^{dp}=\left(p_i^{dp}\right)^{-1}=\left(1-\frac{1}{2^k}\sum_{j=1}^{i-1}\rho(j,q)\right)^{-\rho(i,q)}\cdot\left(1-\frac{\rho(i,q)}{2^k}\right)^{-\frac{\rho(i,q)\cdot(\rho(i,q)-1)}{2}}$.

Оскільки зі збільшенням кількості рядків ймовірність появи колізій, і відповідно, обчислювальна складність додавання нового рядка збільшується,

то математичне очікування верхньої межі складності побудови таблиці з рядків не перевищить значення

$$\Theta^{dp} \le m \cdot \left(1 - \frac{1}{2^k} \sum_{j=1}^{m-1} \rho(j,q)\right)^{-\rho(m,q)} \cdot \left(1 - \frac{\rho(m,q)}{2^k}\right)^{-\frac{\rho(m,q)\cdot(\rho(m,q)-1)}{2}}. \quad (1.8)$$

Оскільки $\rho(1,q), \rho(2,q), \ldots, \rho(m-1,q)$ є випадковими величинами, то оцінка ймовірності бесколізіоного додавання m-го рядка і оцінка складності побудови всієї таблиці відповідно до отриманих формул можлива тільки після формування попередніх $m-1$ рядків.

Проте, в переважній більшості випадків кількість елементів, що містяться в одному рядку, значно менше потужності множини значень невідомого параметра: $\rho(i,q) << 2^k$, і при цьому кількість рядків є досить великою, щоб середня довжина рядка в таблиці стала близькою до математичного сподівання. Виходячи з цього, можна зробити ряд припущень, які дозволять значно спростити (2.8).

Як було зазначено вище, довжина рядка $\rho(i,q)$ є випадковою величиною, розподіленою згідно з геометричним законом розподілу. Звідси математичне очікування середньої довжини рядка, який завершується точкою розрізнення довжиною d бітів дорівнює:

$$l = E\left(\rho(i,q)\right) = 1 + \frac{1-p}{p} = 1 + \frac{1-2^{-d}}{2^{-d}} = 2^d. \quad (1.9)$$

При генерації одного рядка формується $C_l^2 = \frac{l \cdot (l-1)}{2}$ пар, елементи яких повинні бути різні, і оскільки $l << 2^k$, ймовірність зациклення приблизно оцінюється як $p_{nl}^{dp} \approx \left(1 - \frac{1}{2^k}\right)^{\frac{l\cdot(l-1)}{2}}$.

При додаванні i-го рядка з l елементів можливі колізії з елементами, вже врахованими в рядках $1..(i-1)$. Таким чином, у кожного елемента, що додається можлива колізія $(i-1)\cdot l$ елементами таблиці, звідки загальна кількість потенційно колізійних пар дорівнює $(i-1)\cdot l^2$. Враховуючи, що $l << 2^k$, ймовірність додавання i-го рядка за відсутності колізій дорівнює $p_{nc}^{dp} \approx \left(1-\frac{1}{2^k}\right)^{l^2\cdot(i-1)}$.

Оскільки колізії при генерації рядка (зациклення) і збіг з уже доданими елементами є незалежними подіями, то ймовірність успішного додавання i-го рядка з виходом на нове (неповторне) значення кінцевої точки приблизно дорівнює

$$p_i^{dp} \approx \left(1-\frac{1}{2^k}\right)^{l^2\cdot(i-1)} \cdot \left(1-\frac{1}{2^k}\right)^{\frac{l\cdot(l-1)}{2}} = \left(1-\frac{1}{2^k}\right)^{l^2\cdot\left(i-\frac{1}{2}\right)-\frac{l}{2}} \approx e^{-\frac{l^2\cdot(2i-1)+l}{2^{k+1}}}. \qquad (1.10)$$

Звідси оцінка верхньої межі складності побудови таблиці дорівнює

$$\Theta^{dp} \le m\cdot\left(1-\frac{1}{2^k}\right)^{-l^2\cdot\left(m-\frac{1}{2}\right)+\frac{l}{2}} \approx m\cdot e^{\frac{2^{2d}\cdot(2\cdot m-1)-2^d}{2^{k+1}}}. \qquad (1.11)$$

Імовірність успіху на оперативному етапі.

Всі враховані елементи є унікальними, звідки у всій таблиці міститься $\sum_{j=1}^{m}\rho(j,q)$ значень невідомого параметра. При випадковому, рівновірогідному і незалежному виборі з множини потужністю 2^k ймовірність вибору параметра, що міститься в таблиці, дорівнює $P_{succ}^{dp} = \frac{1}{2^k}\cdot\sum_{j=1}^{m}\rho(j,q)$.

При використанні (1.9) для $l = E(\rho(i,q)) = 2^d$

$$P_{succ}^{dp} = \frac{\rho(1,q) + \rho(2,q) + ... + \rho(m,q)}{2^k} \approx \frac{m \cdot l}{2^k} = \frac{m}{2^{k-d}}.$$

Обмеження на кількість рядків таблиці $m \le 2^{k-d}$ в формулюванні теореми випливає з того, що кожен елемент таблиці є унікальним, і обсяг таблиці не може перевищити потужність безлічі невідомого параметра. Імовірність побудови досконалої таблиці (що містить всі значення невідомого параметра) без виникнення колізій (рівність $m = 2^{k-d}$) практично дорівнює нулю.

Теорема доведена.

В якості ілюстрації можна навести такий приклад. При потужності множини невідомого параметра $\#K = 2^{64}$, кількості бітів в точці розрізнення $d = 16$, рядків $m = 2^{35}$ розмір таблиці складе приблизно $2^{16} \cdot 2^{35} = 2^{51}$, а ймовірність успішного проведення криптоаналізу $2^{51-64} = 2^{-13} \approx 1{,}2 \cdot 10^{-4}$ (відновлюється всього 1 ключ з 10 тис.). При цьому для обліку 2^{51} елементів потрібно $2^{46} \cdot 2^{16} = 2^{62}$ операцій прямого криптографічного перетворення, що викликано вкрай низькою ймовірністю (високою обчислювальною складністю) успішного додавання нових рядків при їх великій кількості.

Графік залежності ймовірності успішного додавання рядка від їх поточної кількості для обраних параметрів на підставі (1.11) представлений на рис. 1.2.

Наведений приклад показує, що складність побудови таблиці на основі точок розрізнення з однією функцією переходу залежить експоненційно збільшується в залежності від ймовірності успіху криптоаналіза. Для отримання досить високу ймовірність успіху доцільно використовувати велику кількість таблиць з різними функціями переходу або комбінування методу на основі точок розрізнення з іншими способами побудови таблиць передобчислення.

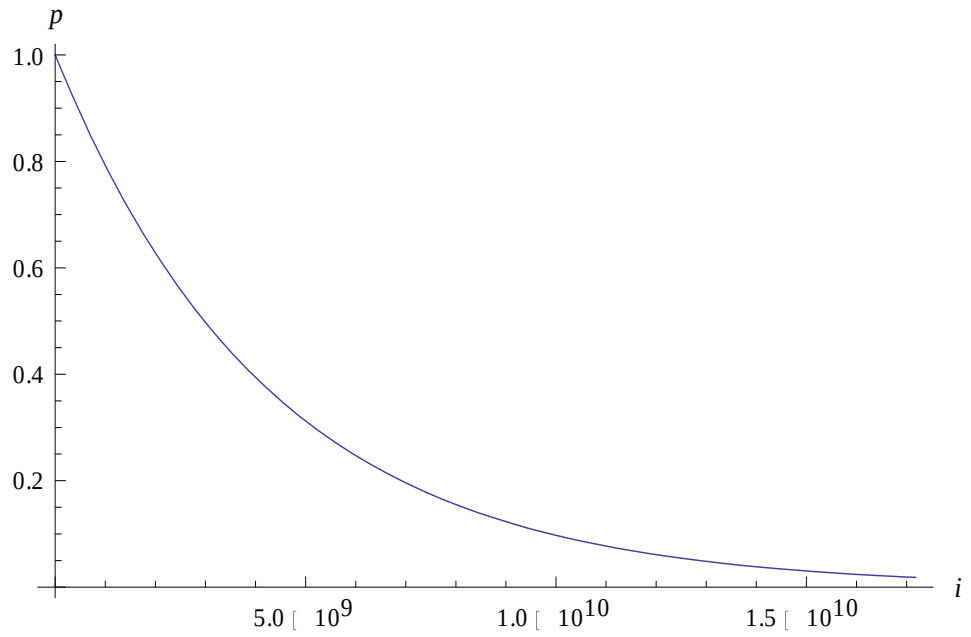


Рисунок 1.2 – Залежність ймовірності успішного додавання нового рядка (p) від поточного розміру таблиці (i)

Як вже було зазначено, етап передобчислення для методу на основі точок розрізнення є ресурсоємним (вимагає великої кількості операцій прямого криптографічного перетворення, як в комбінації з іншими методами побудови таблиць, так і окремо). Особливістю є те, що при побудові рядка необхідно контролювати виникнення циклу, тобто послідовності виду:

$$k_{i,1}^{(q)} \xrightarrow{F_q} k_{i,2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{i,j}^{(q)} \xrightarrow{F_q} k_{i,j+1}^{(q)} \xrightarrow{F_q} k_{i,j+2}^{(q)} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k_{i,j+t}^{(q)} = k_{i,j}^{(q)} \xrightarrow{F_q} k_{i,j+1+t}^{(q)} = k_{i,j+1}^{(q)} \xrightarrow{F_q} \ldots$$

,

в якій повторюються кожні t елементів після перших j елементів послідовності ($j \geq 0$) без попадання в точку розрізнення. Допустимість наявності передперіода значно ускладнює завдання. Неефективний алгоритм контролю може призвести до збільшення складності побудови таблиці.

Формована послідовність має властивості випадкового відображення, і тому складність алгоритму прямої перевірки попадання в цикл дорівнює $O(n^2)$, де n – довжина сформованої послідовності. Оптимізація цього алгоритму, яка полягає у використанні ефективного сортування (наприклад,

швидкого [14]) і пошуку повторюваних елементів знизить складність тільки до $O(n \cdot \log n)$.

Для оптимізації контролю появи циклу в рядку який формується пропонується використовувати імовірнісний алгоритм, який має тимчасову і просторову складність $O(1)$, з можливістю завдання необхідної ймовірності успіху.

Пропонований імовірнісний алгоритм є надзвичайно простим і передбачає тільки контроль поточної довжини послідовності. При перевищенні порогового значення, обчисленого на основі заздалегідь заданій ймовірності помилки першого роду, приймається рішення про зациклення послідовності без виходу на контрольну точку.

Визначити граничне значення можна на підставі наступної теореми.

Теорема 1.2 (порогове значення допустимої довжини рядка таблиці на основі точок розрізнення).

При побудові рядка таблиці на основі точок розрізнення довжиною d бітів, використовуваної для реалізації атаки переборного типу на симетричний криптографічний примітив з потужністю допустимих значень невідомого параметра 2^k (при цьому $2^d << 2^k$), ймовірність того, що рядок до виходу на точку розрізнення буде містити t елементів і більш, дорівнює:

$$p(l_c \geq t) = 1 - 2^{-d} \cdot \sum_{i=0}^{t-1} (1 - 2^{-d})^i . \qquad (1.12)$$

Доказ.

Рядок складається як мінімум з двох елементів (початкова та кінцева точки). довжина рядка $l_c = \rho(i, q)$ є випадковою величиною, розподіленою згідно з геометричним розподілом (визначення і властивості F_q див. в п. 2.4). Імовірність того, що довжина дорівнюватиме точно t елементам ($t \geq 2$ с урахуванням початкового та кінцевого елемента), дорівнює $p(l_c = t) = (1-p)^{t-2} \cdot p$, де $p = 2^{-d}$ – ймовірність появи в послідовності $k_{i,1}^{(q)}, k_{i,2}^{(q)}, \ldots$ значень $k_{i,\rho(i,q)}^{(q)}$ з ознакою точки розрізнення. Події формування

рядків довжиною $t+1, t+2, \ldots$ є несумісними, звідки ймовірність того, що довжина ланцюжка t елементів і більш, дорівнює $p(l_c \geq t) = p \cdot \sum_{i=t-2}^{\infty} (1-p)^i$. Події появи рядка довжиною до t елементів і більше – протилежні, звідки:

$$p(l_c \geq t) = 1 - p \cdot \sum_{i=0}^{t-1} (1-p)^i = 1 - 2^{-d} \cdot \sum_{i=0}^{t-1} (1-2^{-d})^i$$

Теорема доведена.

Обираючи необхідне значення $p(l_c \leq t)$ як ймовірність помилки першого роду (ймовірність відкинути довгий рядок, в якій немає циклу, але ще не з'явився елемент, відповідний точці розрізнення), можна визначати граничне значення довжини ланцюжка для імовірнісного алгоритму.

Наприклад, для потужності множини невідомого параметра $\#K = 2^{64}$ при використанні 8-бітових точок розрізнення довжина 99% ланцюжків буде менш ніж 1177 елементів:

$$p(l_c \geq 1177) = 1 - 2^{-d} \cdot \sum_{i=0}^{t-1} \left(1-2^{-d}\right)^i = 1 - 2^{-8} \cdot \sum_{i=0}^{1177-1} \left(1-2^{-8}\right)^i \approx 0{,}01.$$

Перевищення цієї довжини означає або знаходження дуже довгого ланцюжка (ймовірність такої події 0,01), або увійти в короткий цикл випадкової функції, де відсутня точка розрізнення.

Обчислювальна складність перевірки наявності елемента в таблицях на оперативному етапі дорівнює $O\left(l \cdot 2^d\right)$, де l – кількість таблиць (з різними функціями переходу), побудованих на етапі передобчислення.

Таким чином, розроблена модель таблиць на основі точок розрізнення з використанням сформульованих і доведених теорем дозволяє оцінити складність побудови і ймовірність успіху криптоаналіза для різних комбінацій потужності ключової множини, кількості рядків і фіксованих значень бітів в точці розрізнення. Для отримання досить високої ймовірності успіху доцільно використовувати велику кількість таблиць з різними функціями переходу або комбінування методу на основі точок розрізнення з іншими способами побудови таблиць передобчислення.

Модель може бути використана як при виборі оптимальних параметрів криптоаналітичної системи, так і при обґрунтуванні стійкості симетричного криптографічного примітиву до переборних атак.

1.6 Уточнена оцінка rainbow-таблиць

Метод передобчислень на основі rainbow-таблиць був представлений в 2003 році [11]. Завданням його розробки було зниження ймовірності злиття різних ланцюжків, що визначає основну складність етапу передобчислення, зі збереженням фіксованого розміру таблиці. Основною відмінністю rainbow-таблиць від попередніх методів є використання унікальної функції переходу в кожному стовпчику, тобто рядок має фіксовану довжину і представляється як $k_{i,1} \overset{F_1}{\to} k_{i,2} \overset{F_2}{\to} \ldots \overset{F_{t-1}}{\to} k_{i,t}$. Застосування унікального відображення для кожного елемента дозволяє істотно знизити ймовірність колізії, і для ряду завдань криптоаналізу досить побудови однієї таблиці великого розміру (замість значної кількості порівняно малих для методу Хеллмана і точок розрізнення).

Rainbow-таблиця, що складається з m рядків і t стовпців, приведена на рис. 1.3.

$$
\begin{array}{l}
k_{11} \overset{F_1}{\to} k_{12} \overset{F_2}{\to} \ldots \overset{F_{t-1}}{\to} k_{1,t} \\
k_{21} \overset{F_1}{\to} k_{22} \overset{F_2}{\to} \ldots \overset{F_{t-1}}{\to} k_{2,t} \\
\ldots \qquad\quad \ldots \qquad\quad \ldots \\
k_{m,1} \overset{F_1}{\to} k_{m,2} \overset{F_2}{\to} \ldots \overset{F_{t-1}}{\to} k_{m,t}
\end{array}
$$

Рисунок 1.3 – Rainbow-таблиця з m рядків і t стовпців

Оцінки ефективності rainbow-таблиць, представлені авторами методу і в ряді інших робіт у відкритій пресі, дають деяке наближене значення для

ймовірності успіху і складності побудови таблиці. Проте, при необхідності отримання високої ймовірності успіху (значному обсязі таблиці), відомі методи дають значну похибку, викликану використанням ряду припущень. В роботі [12] наводиться більш точна формула, але вона визначає лише критерій зупинки збільшення обсягу таблиці. Наші результати обчислювального моделювання додатково підтверджують необхідність розробки методики, що дає більш точні результати.

Для побудови такої таблиці необхідно застосування $t-1$ різних функцій переходу $F_1, F_2, ...F_{t-1}$. Всі початкові точки кожного рядка є унікальними: $k_{i,1} \neq k_{j,1}$, $i \neq j$. При генерації рядків необхідно перевіряти, що всі кінцеві точки є унікальними ($k_{i,t} \neq k_{j,t}$, $i \neq j$) для виявлення злиття ланцюжків (появи однакових елементів в одному стовпці).

Імовірність появи циклу в рядку rainbow-таблиці дорівнює $p_L^{rb} = \left(\left(\frac{1}{2^k} \right)^{-l} \right)^{t/l} = \frac{1}{2^{k \cdot t}}$ для циклу довжиною l елементів. Відповідно, така подія є практично неможливою.

Теорема 1.3 (ефективність rainbow-таблиць).

Для rainbow-таблиць, які містять m рядків і t стовпців, використовуваної для реалізації атаки переборного типу на симетричний криптографічний примітив з потужністю допустимих значень невідомого параметра 2^k, $(m << 2^k)$:

на етапі передобчислення – математичне очікування верхньої межі складності побудови таблиці (кількість рядків, що генеруються) не перевищує значення:

$$\Theta^{rb} < m \cdot \left(1 - \frac{m-1}{2^k} \right)^{1-t} \approx m \cdot e^{\frac{(m-1)(t-1)}{2^k}}; \qquad (1.13)$$

на оперативному етапі – ймовірність успішного відновлення при випадковому, рівновірогідному і незалежному виборі невідомого параметра дорівнює

$$p^{rb} = 2^{-k}\sum_{i=1}^{t} m_i ,$$
$$m_j = m\cdot\left(1-2^{-k}\sum_{i=1}^{j-1} m_i\right), m_1 = m \qquad (1.14)$$

Доказ.

Імовірність успіху при виконанні криптоаналізу на основі rainbow-таблиці може бути обчислена через кількість унікальних внутрішніх станів, врахованих в таблиці.

Відповідно до правил побудови rainbow-таблиць, в першому стовпці присутні $m_1 = m$ різних елементів. У другому стовпі присутні m попарно різних елементів (в іншому випадку, при збігу двох елементів в стовпці, це б проявилося однаковою кінцевою точкою для обох рядків). Проте, можливі колізії між елементами першої і другої колонок.

Перший елемент, що додається в другу колонку таблиці, з ймовірністю $\frac{m}{2^k}$ співпаде з одним з елементів першого стовпця, або з ймовірністю $1-\frac{m}{2^k}$ буде унікальним. Другий елемент обов'язково відрізняється від першого (через перевірки унікальності кінцевих точок), і вибирається з множини потужністю 2^k-1, з відповідними імовірностями $\frac{m}{2^k-1}$ і $1-\frac{m}{2^k-1}$. Для елемента з номером m, що додається в другу колонку, ймовірності відповідно рівні $\frac{m}{2^k-m+1}$ і $1-\frac{m}{2^k-m+1}$.

З огляду на те, що для випадку практичного застосування $m << 2^k$ (в іншому випадку виходить трохи спрощений варіант словникової атаки),

нижче допускається, що ймовірність колізії та успішного додавання кожного елемента другої колонки дорівнює $\frac{m}{2^k}$ та $1-\frac{m}{2^k}$.

Кількість колізій між елементами першої і другої колонки є випадковою величиною, розподіленою згідно з біноміальним законом. Імовірність отримання в другій колонці s унікальних елементів (що відрізняються від всіх елементів першої колонки) дорівнює $p_2(k)=\binom{m}{s}\left(1-\frac{m}{2^k}\right)^s\left(1-\left(1-\frac{m}{2^k}\right)\right)^{m-s}=\binom{m}{s}\left(1-\frac{m}{2^k}\right)^s\left(\frac{m}{2^k}\right)^{m-s}$.

Після додавання m елементів в другу колонку (повного заповнення) математичне сподівання кількості унікальних елементів другої колонки дорівнює $m_2=m\left(1-\frac{m}{2^k}\right)$.

Для третьої колонки необхідно брати до уваги колізії додаються елементи з унікальними внутрішніми станами, врахованими в першій і другій колонках: $m_3=m\left(1-\frac{m+m_2}{2^k}\right)$.

Відповідно, для j-ї колонки математичне сподівання кількості унікальних ключових станів дорівнює:

$$m_j=m\cdot\left(1-2^{-k}\sum_{i=1}^{j-1}m_i\right),\ m_1=m \qquad (1.15)$$

Таким чином, rainbow-таблиця містить $m_1+m_2+\ldots+m_t$ унікальних елементів, наступний елемент обчислюється через попередні через рекурентне співвідношення (1.15). Відзначимо, що аналітичне подання (1.15) тільки через m,t і j можливо, але воно є набагато більш громіздким як при виведенні, так і для розрахунків при програмній реалізації.

У відповідності з властивостями функцій переходу $F_1, F_2, ...F_{t-1}$, елементи таблиці є значеннями випадкової величини, розподіленої відповідно за рівномірним законом розподілу. При випадковому, рівновірогідному і незалежному виборі внутрішнього стану для криптоаналізу ймовірність успіху rainbow-таблиці оцінюється як

$$p^{rb} = 2^{-k} \sum_{i=1}^{t} m_i \,. \tag{1.16}$$

Обсяг передобчислення залежить як від обчислювальної складності розрахунку функцій переходу $F_1, F_2, ...F_{t-1}$, так і від кількості колізій кінцевих точок при побудові rainbow-таблиці.

Імовірність виникнення колізії кінцевої точки при додаванні чергового рядка можна оцінити таким чином.

Рядок rainbow-таблиці має t елементів, перший з яких є унікальним (вибір при генерації початкової точки). Решта можуть повторювати будь-якій з уже доданих в своїй колонці (для другого рядка і далі), що призводить до колізії кінцевої точки і необхідності вибору нового початкового значення і повторного формування рядка. Відповідно, при додаванні другого рядка є $t-1$ пар елементів, для яких можлива колізія. Імовірність відсутності колізії в кожній парі дорівнює $1-\frac{1}{2^k}$, відповідно, при обліку всіх пари ймовірність безколізійного додавання другого рядка дорівнює $p_2 = \left(1-\frac{1}{2^k}\right)^{t-1}$. Для третього рядка ймовірність відповідно дорівнює $p_3 = \left(1-\frac{1}{2^k}\right)^{t-1}$.

Для l-го рядку ймовірність безколізійного додавання дорівнює:

$$p_{nc}^{rb} = \left(1-\frac{l-1}{2^k}\right)^{t-1}. \tag{1.17}$$

Оцінку складності побудови rainbow-таблиці (за кількістю генерованих рядків) можна отримати як величину, рівну $\Theta^{rb} = 1 + (p_2)^{-1} + (p_3)^{-1} + ... + (p_m)^{-1}$, яка обмежена зверху

$$\Theta^{rb} = 1 + (p_2)^{-1} + (p_3)^{-1} + ... + (p_m)^{-1} < m \cdot \left(1 - \frac{m-1}{2^k}\right)^{1-t}. \qquad (1.18)$$

При $m << 2^k$ можливо використовувати наближення (1.18) через $\Theta^{rb} < m \cdot \left(1 - \frac{m-1}{2^k}\right)^{1-t} \approx m \cdot e^{\frac{(m-1)(t-1)}{2^k}}$.

Теорема доведена.

Слідство 1.1.

З теореми 1.3 випливає, що навіть при великих обсягах таблиці ($m \cdot t \approx 2^k$) при $m << 2^k$ ймовірність відсутності колізії при завершенні побудови таблиці залишається досить високою: $p_{nc}^{rb} \approx e^{-1}$, що означає, що навіть для додавання m-го рядку потрібно в середньому не більше 3-х спроб.

Таким чином, застосування rainbow-таблиці для криптоаналізу дозволяє отримати мінімальну складність етапу передобчислення навіть для таблиць, за обсягом близьких до потужності множини внутрішніх станів. У порівнянні з іншими типами таблиць (Хеллман, точки розрізнення, fuzzy-rainbow) в rainbow-таблиці буде максимальна кількість унікальних елементів.

Ще однією перевагою при практичній реалізації є те, що rainbow-матриця (множина початкових і кінцевих точок) такого обсягу буде зберігатися на декількох носіях, що дозволить значно прискорити виконання криптоаналіза для декількох блоків гами.

Недоліком методу є наступне. Незважаючи на найменшу обчислювальну складність генерації і найбільшу кількість унікальних елементів, застосування тільки rainbow-таблиці для криптоаналізу

призводить до необхідності перевірки шуканого внутрішнього стану на приналежність до кожної колонці, що, в свою чергу, веде до значного збільшення часу виконання атаки.

Обчислювальна складність перевірки наявності елемента в rainbow-таблиці на оперативному етапі дорівнює $O\left(t^2 \cdot \log m\right)$, де t – довжина рядка.

Наприклад, при використанні таблиці розміром $m \cdot t = 2^{58}$ внутрішніх станів, складність побудови rainbow-матриці не перевищить $e \cdot 2^{58} \approx 2^{59,44}$ операцій прямого криптографічного перетворення.

Таким чином, застосування виключно rainbow-матриці дозволить максимально знизити вартість етапу предвирахування, але оперативний етап виходить досить тривалим навіть при побудові таблиці великого розміру.

1.7 Fuzzy-rainbow-таблиці

Цей метод об'єднує можливості rainbow-таблиць і ланцюжків на основі точок розрізнення, дозволяючи отримати компромісне рішення для досягнення високої ймовірності успіху, порівняно малий час криптоаналіза і прийнятний рівень складності на етапі передобчислення. На момент проведення наших досліджень у відкритій пресі представлені результати тільки для визначення співвідношення обсягу необхідної пам'яті і кількості різних функцій переходу для фіксованої ймовірності виникнення колізії [11]. Оскільки fuzzy-rainbow-таблиці є найбільш ефективним методом в порівнянні з вже описаними, для цього методу буде приведено детальний опис.

Криптоаналіз починається з етапу передобчислення, в ході якого проводиться побудова fuzzy-rainbow-таблиці, що складається з врахованих значень невідомого параметра і її перетворення в fuzzy-rainbow-матрицю для оперативного етапу.

Fuzzy-rainbow-таблиця складається з m рядків, кожна з яких отримана шляхом багаторазового застосування функцій переходу $F_1, F_2, ..., F_s$ (рис. 1.4).

Зміна функції в таблиці $F_i \to F_{i+1}$ проводиться після появи в ланцюжку значень $k_{i,r}^{(q)}, k_{i,r+1}^{(q)}, k_{i,r+2}^{(q)}, ...$ деякої ознаки. Як правило, такою ознакою є поява заданої кількості (d)нульових бітів в фіксованих позиціях змінної k, описує внутрішній стан генератора.

Таблиця будується по рядках (послідовно, починаючи з рядка 1 до рядка m), відповідно до алгоритмів 1.1 і 1.2. Передбачається, що вхідне значення (послідовність) є фіксованим. Складність побудови таблиці (матриці) оцінюється як $O\left(m \cdot e^{s \cdot m \cdot 2^{2d}}\right)$, див. теорему 1.4.

F_1	F_2	...	F_s
$k_{1,1}^{(1)} \xrightarrow{F_1} k_{1,2}^{(1)} \xrightarrow{F_1} \cdots \xrightarrow{F_1} k_{1,\rho(1,1)}^{(1)}$	$\xrightarrow{F_2} k_{1,1}^{(2)} \xrightarrow{F_2} k_{1,2}^{(2)} \xrightarrow{F_2} \cdots \xrightarrow{F_2} k_{1,\rho(1,2)}^{(2)}$	...	$\xrightarrow{F_s} k_{1,1}^{(s)} \xrightarrow{F_s} k_{1,2}^{(s)} \xrightarrow{F_s} \cdots \xrightarrow{F_s} k_{1,\rho(1,s)}^{(s)}$
$k_{2,1}^{(1)} \xrightarrow{F_1} k_{2,2}^{(1)} \xrightarrow{F_1} \cdots \xrightarrow{F_1} k_{2,\rho(2,1)}^{(1)}$	$\xrightarrow{F_2} k_{2,1}^{(2)} \xrightarrow{F_2} k_{2,2}^{(2)} \xrightarrow{F_2} \cdots \xrightarrow{F_2} k_{2,\rho(2,2)}^{(2)}$	...	$\xrightarrow{F_s} k_{2,1}^{(s)} \xrightarrow{F_s} k_{2,2}^{(s)} \xrightarrow{F_s} \cdots \xrightarrow{F_s} k_{2,\rho(2,s)}^{(s)}$
...	...	...	...
$k_{m,1}^{(1)} \xrightarrow{F_1} k_{m,2}^{(1)} \xrightarrow{F_1} \cdots \xrightarrow{F_1} k_{m,\rho(m,1)}^{(1)}$	$\xrightarrow{F_2} k_{m,1}^{(2)} \xrightarrow{F_2} k_{m,2}^{(2)} \xrightarrow{F_2} \cdots \xrightarrow{F_2} k_{m,\rho(m,2)}^{(2)}$	...	$\xrightarrow{F_s} k_{m,1}^{(s)} \xrightarrow{F_s} k_{m,2}^{(s)} \xrightarrow{F_s} \cdots \xrightarrow{F_s} k_{m,\rho(m,s)}^{(s)}$

Рисунок 1.4 – Fuzzy-rainbow-таблиця

Алгоритм 1.1 (побудова рядка fuzzy-rainbow-таблиці).

Вхід: початкове значення $k_{i,1}^{(1)} \in K$.

Вихід: етап передобчислення – повідомлення «успішне завершення» і кінцева точка рядки $k_{i,\rho(i,s)}^{(s)} \in K$ або повідомлення «неуспішне завершення»; оперативний етап – рядок $k_{i,1}^{(1)} \xrightarrow{F_1} k_{i,2}^{(1)} ... \xrightarrow{F_s} k_{i,\rho(i,s)}^{(s)}$ повністю.

1. Номер функції переходу q задається 1: $q \leftarrow 1$.

2. Номер поточного елемента r в ланцюжку застосування функції F_q встановлюється в 1: $r \leftarrow 1$.

3. До отриманого значення застосовується чергова функція переходу для отримання наступного враховується внутрішній стан: $k_{i,r+1}^{(q)} = F_q\left(k_{i,r}^{(q)}\right),\ r \leftarrow r+1.$

4. Якщо кількість застосування функції переходу F_q перевищило порогове значення ($r > l_{\max}$, див. теорему 2.2), вихід з повідомленням «неуспішне завершення».

5. Якщо в черговому значенні відсутня ознака зміни функції, виконання починається з кроку 3.

6. При виявленні ознаки зміни функції і поточному номері функції $q < s$, виконується зміна функції переходу F_q на F_{q+1} ($q \leftarrow q+1$), потім слід перейти до кроку 2.

7. При наявності ознаки зміни функції і максимально можливого номері функції переходу $q = s$ формування рядка завершено, проводиться вихід з повідомленням «успішне завершення» і поверненням кінцевої точки $k_{i,r}^{(q)} = k_{i,\rho(i,s)}^{(s)}$ або всього рядка $k_{i,1}^{(1)} \overset{F_1}{\to} k_{i,2}^{(1)} \ldots \overset{F_s}{\to} k_{i,\rho(i,s)}^{(s)}$.

Очевидно, що кількість кроків $\rho(i,q)$, необхідних для появи ознаки зміни функції переходу F_q в послідовність $k_{i,1}^{(q)} \overset{F_q}{\to} k_{i,2}^{(q)} \overset{F_q}{\to} \ldots \overset{F_q}{\to} k_{i,\rho(i,q)}^{(q)}$, є випадковою величиною. Тому кількість застосувань кожної функції переходу в кожному рядку не є фіксованим в fuzzy-rainbow-методі, відповідно, довжини рядків в таблиці різні.

Надалі послідовність $k_{i,1}^{(q)} \overset{F_q}{\to} k_{i,2}^{(q)} \overset{F_q}{\to} \ldots \overset{F_q}{\to} k_{i,\rho(i,q)}^{(q)}$ буде називатися ланцюжком (застосування функції F_q в рядку i), а набір ланцюжків, отриманих в результаті застосування функції F_q у всіх рядках: $k_{1,1}^{(q)} \overset{F_q}{\to} k_{1,2}^{(q)} \overset{F_q}{\to} \ldots \overset{F_q}{\to} k_{1,\rho(1,q)}^{(q)}$, $k_{2,1}^{(q)} \overset{F_q}{\to} k_{2,2}^{(q)} \overset{F_q}{\to} \ldots \overset{F_q}{\to} k_{2,\rho(2,q)}^{(q)}$, ... , $k_{m,1}^{(q)} \overset{F_q}{\to} k_{m,2}^{(q)} \overset{F_q}{\to} \ldots \overset{F_q}{\to} k_{m,\rho(m,q)}^{(q)}$ – колонкою таблиці.

При побудові fuzzy-rainbow-таблиці зберігаються перша і остання колонки (початкова та кінцева точки $k_{i,1}^{(1)}$, $k_{i,\rho(i,s)}^{(s)}$). Безліч пар таких точок для всіх рядків таблиці $\left\{\left(k_{i,1}^{(1)}, k_{i,\rho(i,s)}^{(s)}\right) \middle| 1 \le i \le m\right\}$ називається fuzzy-rainbow-матрицею, необхідною для оперативного етапу криптоаналіза.

Алгоритм 1.2 (побудова fuzzy-rainbow-матриці).

Вхід: –.

Вихід: fuzzy-rainbow-матриця.

1. Вибирається випадкове початкове значення $k_{1,1}^{(1)}$. Відповідно до Алгоритму 1 формується рядок з номером 1 (у разі неуспішного завершення Алгоритму 1 поточний крок виконується повторно).

2. Номер рядка задається рівним 2: $i \leftarrow 2$.

3. Вибирається випадкове початкове значення $k_{i,1}^{(1)}$, відрізняється від усіх вже використаних початкових значень: $k_{i,1}^{(1)} \ne k_{j,1}^{(1)}, 1 \le j < i$.

4. Відповідно до Алгоритму 1 формується рядок з номером i. У разі неуспішного завершення Алгоритму 1 відбувається перехід на крок 3.

5. Перевіряється, що її кінцева точка сформованого рядка $k_{i,\rho(i,s)}^{(s)}$ є унікальною: $k_{i,\rho(i,s)}^{(s)} \ne k_{j,\rho(j,s)}^{(s)}, 1 \le j < i$ (не збігається з жодною кінцевою точкою вже згенерованих рядків). У разі повтору кінцевого значення йде перехід до кроку 3.

6. Сформована пара $\left(k_{i,1}^{(1)}, k_{i,\rho(i,s)}^{(s)}\right)$ з номером i добавляється в матрицю.

7. При i, менше кількості рядків в таблиці $(i < m)$,виконується збільшення номера на 1: $i \leftarrow i + 1$ і перехід до кроку 3.

8. При $i = m$ формування таблиці завершено.

При проведенні оперативного етапу криптоаналізу в розпорядженні криптоаналітика є fuzzy-rainbow-матриця, побудована на попередньому етапі, а також деякі непусті множини вхідних і вихідних послідовностей заданого

примітиву. Завданням є отримання невідомого параметра криптографічного перетворення.

Слід зазначити, що в загальному випадку, кількість рішень задачі може бути різним – від нуля (на фазі передобчислення такі послідовності не були враховані при побудові fuzzy-rainbow-таблиці) до довільної кількості (значення невідомого параметра є еквівалентними і формують однакові вихідні послідовності).

Криптоаналіз виконується за алгоритмом 1.3. Складність оперативного етапу оцінюється як $O\left(s^2 \cdot 2^d \cdot \log m\right)$.

Алгоритм 1.3 (Відновлення значення невідомого параметра по вхідний і вихідний послідовності).

Вхід: вихідна послідовність заданого симетричного криптографічного примітиву γ, сформована на основі невідомого параметра; fuzzy-rainbow-матриця.

Вихід: множина (допустимо – пустих) значень невідомого параметра, що формують вихідну послідовність.

1. Номер функції переходу q задається s: $q \leftarrow s$. Номер поточного стовпця таблиці c задається s: $c \leftarrow s$.

2. До отриманої вихідний послідовності γ застосовується випадкова функція f_q: $u_1 = f_q(\gamma)$.

3. Номер поточного елемента r в ланцюжку застосування функції F_q встановлюється в 1: $r \leftarrow 1$.

4. Якщо в отриманому значенні u_r присутня ознака зміни функції, виконання триває з кроку 8.

5. До значення u_r застосовується функція переходу F_q для отримання наступного значення в ланцюжку: $u_{r+1} = F_q(u_r)$, лічильник елементів збільшується на 1: $r \leftarrow r + 1$.

6. Якщо кількість застосування функції переходу F_q перевищило порогове значення ($r > l_{\max}$), відбувається вихід з алгоритму (завершення роботи).

7. Відбувається перехід на крок 4.

8. Якщо поточний номер функції q дорівнює s ($q = s$), відбувається перехід на крок 10.

9. Відбувається збільшення номера функції переходу на 1: $q \leftarrow q+1$, задається $u_1 \leftarrow u_r$, $r \leftarrow 1$ і виконується перехід на крок 5.

10. Виконується пошук значення u_r з ознакою зміни функції в другій колонці fuzzy-rainbow-матриці $\left\{k^{(q)}_{i,\rho(i,q)} \,\middle|\, 1 \le i \le m\right\}$. Якщо збігається значення $k^{(q)}_{j,\rho(j,q)}$ знайдено, то виконується відновлення рядка j fuzzy-rainbow-таблиці по початковому значенню $k^{(1)}_{j,1}$ з fuzzy-rainbow-матриці (див. Алгоритм 1, значення $k^{(1)}_{i,1} = k^{(1)}_{j,1}$). Для ланцюжка рядка, що відповідає функції F_q: $k^{(q)}_{j,1} \xrightarrow{F_q} k^{(q)}_{j,2} \xrightarrow{F_q} \ldots \xrightarrow{F_q} k^{(q)}_{j,\rho(j,q)}$ вибирається елемент $k^{(q)}_{\rho(j,q)-r+1,2}$ (який стоїть від початку ланцюжка на позиції $\rho(j,q)-r+1$), який і буде одним з варіантів шуканого внутрішнього стану. Елемент $k^{(q)}_{\rho(j,q)-r+1,2}$ повертається як черговий результат роботи алгоритму.

11. Якщо номер поточного стовпця $c > 2$, то виконується його зменшення на 1: $c \leftarrow c-1$, номер функції переходу q задається c: $q \leftarrow c$, і виконується перехід на крок 2.

12. При номері поточного стовпця $c = 1$, то відбувається вихід з алгоритму (завершення роботи).

Для оцінки ефективності методу необхідно отримати значення математичного очікування кількості унікальних елементів у всій таблиці (для визначення ймовірності успіху оперативного етапу криптоаналіза) і ймовірність безколізійного додавання чергового рядка в таблицю (для оцінки

складності етапу передобчислення). Час проведення криптоаналізу прямо пропорційний кількості стовпців в усій таблиці.

Теорема 1.4 (ефективність fuzzy-rainbow-таблиць).

При реалізації атаки переборного типу на симетричний криптографічний примітив з потужністю допустимих значень невідомого параметра 2^k, а основі fuzzy-rainbow-таблиці, що містить m рядків ($m << 2^k$) і s різних функцій переходу, змінюваних при виході на точку розрізнення довжиною d бітів:

на етапі передобчислення - математичне очікування верхньої межі складності побудови таблиці не перевищує значення

$$\theta^{fr} < m \cdot \left(1 - \frac{1}{2^k}\right)^{s \cdot l - s \cdot l^2 \left(m - \frac{1}{2}\right)} \cdot \left(1 - \frac{m-1}{2^{k-d}}\right)^{1-s} \approx \approx m \cdot e^{2^{-k}\left(s 2^{2d}\left(m - \frac{1}{2}\right) + 2^d((s-1)(m-1) - s)\right)}, \quad (1.19)$$

на оперативному етапі – ймовірність успішного відновлення при випадковому, рівновірогідному і незалежному виборі невідомого параметра дорівнює:

$$p_{succ}^{fr} = 2^{-k} \sum_{i=1}^{s} m_i ,$$
$$m_j = m \cdot l \cdot \left(1 - \frac{1}{2^k} \sum_{i=1}^{j-1} m_i\right), m_1 = m \cdot l \quad (1.20)$$

Доказ.

Кожна колонка fuzzy-rainbow-таблиці може бути представлена як таблиця на основі точок розрізнення, і всі елементи такої таблиці (колонки) є унікальними. Відповідно, в межах допущення про середній довжині ланцюжка в $l = 2^d$ елементів (2.9), кожна колонка таблиці має, в свою чергу, приблизно l підколонок. Колізії в цьому випадку можуть бути тільки лише з підколонки попередніх колонок (властивість таблиці на основі точок

розрізнення), звідки математичне сподівання кількості унікальних елементів другої колонки дорівнює $m_2 = m \cdot l \cdot \left(1 - \frac{m \cdot l}{2^n}\right)$.

Фактично, тут використовується модель для rainbow-таблиць, з наступним уточненням: до першої колонки, що містить $m \cdot l$ унікальних елементів, додається друга, в якій знаходиться така ж кількість нових елементів. Підрахунок як і раніше ведеться в рамках припущення, що $m << 2^k$ і $m \cdot l << 2^k$.

Таким чином, для j-ї колонки fuzzy-rainbow-таблиці математичне сподівання кількості унікальних ключових станів дорівнює

$$m_j = m \cdot l \cdot \left(1 - \frac{1}{2^k} \sum_{i=1}^{j-1} m_i\right). \quad (1.21)$$

Відповідно, fuzzy-rainbow-таблиця містить $m_1 + m_2 + m_3 + \ldots + m_s$ унікальних елементів, $m_1 = m \cdot l$, наступний елемент обчислюється через попередні через рекурентне співвідношення (1.21).

При випадкових функціях переходу $F_1, F_2, \ldots, F_s$ і випадковому, рівновірогідному і незалежному виборі внутрішнього стану для криптоаналізу ймовірність успіху fuzzy-rainbow-таблиці оцінюється як:

$$P_{succ}^{fr} = \frac{1}{2^k} \sum_{i=1}^{s} m_i \,. \quad (1.22)$$

Безколізійне додавання нового рядка з номером i відбувається, коли виконуються всі перераховані умови:

– відсутність циклу в кожному ланцюжку застосування функцій переходу $F_1, F_2, \ldots, F_s$ для додаваємого рядка;

– для кожної колонки вже побудованої частини таблиці: відсутність колізій між елементами ланцюжка рядки i і елементами відповідного стовпця;

– відсутність колізій в точках зміни функції переходу.

Відповідно до того, що функції $F_1, F_2, ..., F_s$ – випадкові, приймаємо, що події відсутності колізій умов 1-3 є незалежними. Події відсутності зациклення ланцюжків кожного стовпця одного рядка та інших також вважаємо незалежними.

Ймовірність виконання умови 1 без урахування колізії по вже побудовані частині колонки, але з урахуванням можливості зациклення (див. доведення теореми 1.1) для s різних функцій переходу:

$$p_i^{fr,nl} = \left(\left(1-\frac{1}{2^k}\right)^{\frac{l\cdot(l-1)}{2}}\right)^s = \left(1-\frac{1}{2^k}\right)^{\frac{s\cdot l\cdot(l-1)}{2}}. \qquad (1.23)$$

Ймовірність виконання другої умови без зациклення, але з урахуванням можливих колізій для різних функцій переходу:

$$p_i^{fr,nc} = \left(\left(1-\frac{1}{2^k}\right)^{l^2\cdot(i-1)}\right)^s = \left(1-\frac{1}{2^k}\right)^{s\cdot l^2\cdot(i-1)}. \qquad (1.24)$$

Ймовірність виконання умови 3 розраховується за аналогією з (1.17), але з урахуванням того, що зміна функції відбувається на множині, потужність якої менше, ніж потужність множини внутрішніх станів, у 2^d раз через використання d нульових бітів в якості точки розрізнення. Відповідно, ймовірність дорівнює:

$$p_i^{fr,nc} = \left(1-\frac{i-1}{2^{k-d}}\right)^{s-1}. \qquad (1.25)$$

Таким чином, за умови незалежності подій відсутності колізій ймовірність безколізійного додавання рядка з номером i в fuzzy-rainbow-таблицю дорівнює добутку (1.23)-(1.25):

$$p_i^{fr} = \left(1-\frac{1}{2^k}\right)^{\frac{s\cdot l\cdot(l-1)}{2}} \cdot \left(1-\frac{1}{2^k}\right)^{s\cdot l^2\cdot(i-1)} \cdot \left(1-\frac{i-1}{2^{k-d}}\right)^{s-1} = \\ = \left(1-\frac{1}{2^k}\right)^{s\cdot l^2\left(i-\frac{1}{2}\right)-s\cdot l} \cdot \left(1-\frac{i-1}{2^{k-d}}\right)^{s-1} . \quad (1.26)$$

З (1.26) можна визначити кількість рядків в таблиці, при яких імовірність безколізійної генерації залишиться досить високою, відповідно, трудомісткість побудови fuzzy-rainbow-матриці не перевищить задану величину.

Оцінку складності побудови таблиці можна отримати як величину, рівну $\theta^{fr} = (p_1)^{-1} + (p_2)^{-1} + (p_3)^{-1} + \ldots + (p_m)^{-1}$, яка обмежена зверху значенням:

$$\theta^{fr} = (p_1)^{-1} + (p_2)^{-1} + \ldots + (p_m)^{-1} < m\cdot(p_m)^{-1} = \\ = m\cdot\left(1-\frac{1}{2^k}\right)^{s\cdot l - s\cdot l^2\left(m-\frac{1}{2}\right)} \cdot \left(1-\frac{m-1}{2^{k-d}}\right)^{1-s} \approx \quad . \quad (1.27) \\ \approx m\cdot e^{2^{-k}\left(sl^2\left(m-\frac{1}{2}\right)-sl+2^d(s-1)(m-1)\right)} = m\cdot e^{2^{-k}\left(s2^{2d}\left(m-\frac{1}{2}\right)+2^d((s-1)(m-1)-s)\right)}$$

Теорема доведена.

Таким чином, складність побудови fuzzy-rainbow-таблиці вище, ніж просто rainbow-таблиці, але застосування точок розрізнення дозволяє значно скоротити час виконання оперативного етапу.

Проте, потужність множин невідомих параметрів застосовуваних в даний час криптографічних примітивів може призводити до вкрай низької ймовірність безколізійного додавання нового рядка (1.26) до таблиці, і, відповідно, високої складності її побудови.

Отримати прийнятну складність переобчислення і допустимий час проведення криптоаналізу можна на основі наступного компромісу. Використання декількох таблиць замість однієї призведе до лінійного збільшення часу виконання оперативного етапу (або зростання складності апаратури) і експоненціально знизить складність побудови кожної таблиці.

При цьому для ефективного криптоаналітичного комплексу кількість таблиць повинна бути порівняно малою (інакше час проведення оперативного етапу буде високим), а обсяг кожної таблиці – досить великим (в іншому випадку ймовірність успіху буде низькою).

Теорема 1.5 (ефективність кількох fuzzy-rainbow-таблиць).

При реалізації атаки переборного типу на симетричний криптографічний примітив з потужністю допустимих значень невідомого параметра 2^k, на основі побудови r різних fuzzy-rainbow-таблиць, кожна з яких містить m рядків і унікальний набір з s різних функцій переходу, змінюваних при виході на точку розрізнення довжиною d бітів:

на етапі передобчислення – математичне очікування верхньої межі складності побудови таблиці (кількість генерованих рядків) не перевищує значення:

$$\theta^{frn} < r \cdot m \cdot \left(1 - \frac{1}{2^k}\right)^{s \cdot l - s \cdot l^2 \left(m - \frac{1}{2}\right)} \cdot \left(1 - \frac{m-1}{2^{k-d}}\right)^{1-s} \approx \approx r \cdot m \cdot e^{2^{-k}\left(s 2^{2d}\left(m - \frac{1}{2}\right) + 2^d((s-1)(m-1) - s)\right)} \quad (1.28)$$

на оперативному етапі – ймовірність успішного відновлення при випадковому, рівновірогідному і незалежному виборі невідомого параметра дорівнює:

$$P_{succ}^{frn} = \frac{u \cdot r}{2^k} \cdot \left(1 - \frac{u \cdot (r-1)}{2^{k+1}}\right) \quad (1.29)$$

$$u = m_1 + m_2 + m_3 + \ldots + m_s,$$

$$m_j = m \cdot l \cdot \left(1 - \frac{1}{2^k} \sum_{i=1}^{j-1} m_i\right), m_1 = m \cdot l$$

Доказ.

Припустимо, що сформовано r таблиць, в кожній з яких міститься u_1, u_2, …, u_r унікальних елементів (в межах кожної таблиці). Очевидно, що елементи можуть повторюватися в різних таблицях, відповідно, кількість унікальних внутрішніх станів, врахованих в таблицях, менше суми $u_1 + u_2 + ... + u_r$. Оскільки кількість таблиць r має бути порівняно малою, а обсяг кожної таблиці - досить великим, використання припущення $u_j << 2^k$ неможливо.

За аналогією з (2.15), кількість унікальних елементів другої таблиці можна оцінити як $u_2 \cdot \left(1 - \frac{u_1}{2^n}\right)$ на основі виключення колізійних елементів, вже врахованих в першій. Відповідно, для третьої таблиці це значення дорівнює $u_3 \cdot \left(1 - \frac{u_1 + u_2}{2^n}\right)$, для j-го – $u_j \cdot \left(1 - \frac{1}{2^n} \sum_{i=1}^{j-1} u_i\right)$.

Оскільки значення u_1, u_2, …, u_r є випадковими величинами, то для таблиць одного розміру доцільно замінити їх значення математичним очікуванням u. У цьому випадку вводиться допущення, побудовано r таблиць, кожна з яких містить u унікальних елементів, $u = m_1 + m_2 + m_3 + ... + m_s$, а m_i обчислюються відповідно до (1.15).

Таким чином, математичне очікування кількості унікальних елементів в межах всіх r таблиць визначається як:

$$U = u + u \cdot \left(1 - \frac{u}{2^k}\right) + u \cdot \left(1 - \frac{2 \cdot u}{2^k}\right) + u \cdot \left(1 - \frac{3 \cdot u}{2^k}\right) + ... + u \cdot \left(1 - \frac{(r-1) \cdot u}{2^k}\right), \text{ або}$$

$$U = u \cdot r \cdot \left(1 - \frac{u \cdot (r-1)}{2^{k+1}}\right). \quad (1.30)$$

Імовірність успіху при використанні r fuzzy-rainbow-таблиць при випадковому, рівновірогідному і незалежному виборі значення невідомого параметра для криптоаналізу дорівнює:

$$P_{succ}^{frn} = \frac{U}{2^k} = \frac{u \cdot r}{2^k} \cdot \left(1 - \frac{u \cdot (r-1)}{2^{k+1}}\right). \quad (1.31)$$

Час побудови таблиць підсумовується, тому верхня межа математичного очікування часу побудови набору таких таблиць, відповідно до (1.19), дорівнює:

$$\theta^{frn} < r \cdot m \cdot \left(1 - \frac{1}{2^k}\right)^{s \cdot l - s \cdot l^2 \left(m - \frac{1}{2}\right)} \cdot \left(1 - \frac{m-1}{2^{k-d}}\right)^{1-s} \approx$$
$$\approx r \cdot m \cdot e^{2^{-k}\left(sl^2\left(m-\frac{1}{2}\right) - sl + 2^d (s-1)(m-1)\right)} = r \cdot m \cdot e^{2^{-k}\left(s 2^{2d}\left(m-\frac{1}{2}\right) + 2^d ((s-1)(m-1) - s)\right)}.$$

Теорема доведена.

Таким чином, fuzzy-rainbow-таблиці дозволяють отримати компромісне рішення для досягнення високої ймовірності успіху, порівняно малий час криптоаналізу і прийнятний рівень складності на етапі передобчислення. Додаткове експоненційне зниження складності попереднього етапу можна досягти за рахунок лінійного збільшення складності проведення етапу криптоаналізу.

1.8 Порівняння ефективності різних типів таблиць передобчислення

Ефективність таблиць предвирахування визначається на основі наступних чинників:

– ймовірність успіху на оперативному етапі;

– складність (час) виконання оперативного етапу;

– необхідний обсяг пам'яті для реалізації оперативного етапу;

– складність етапу передобчислення.

Метод на основі таблиць Хеллмана дозволяє досягти досить високої ймовірністі успіху, але при цьому необхідний обсяг пам'яті має порядок $O\left(2^{\frac{2k}{3}}\right)$, і на оперативному етапі необхідно близько $O\left(2^{\frac{k}{3}}\right)$ операцій пошуку в таблицях, що робить метод менш ефективним (вкрай повільним або ресурсоємним) в порівнянні з іншими.

Таблиці на основі точок розрізнення мають найменший час оперативного етапу, але при цьому практично досяжна складність побудови не дозволяє добитися високої ймовірності успіху криптоаналізу.

Rainbow-таблиці мають низьку складність побудови, можливість використання однієї таблиці з високою ймовірністю успіху криптоаналізу, але порівняно тривалий оперативний етап.

Fuzzy-rainbow-таблиці є компромісним рішенням, що поєднує переваги двох попередніх методів, і дозволяють отримати високу ймовірність успіху, малий час виконання оперативного етапу і прийнятну складність передобчислення. Застосування декількох fuzzy-rainbow-таблиць дозволяє експоненціально знизити складність передобчислення з лінійним зростанням часу аналізу.

З теорем 1.1, 1.3 і 1.4 маємо, що складність побудови таблиць по методам точок розрізнення, rainbow і fuzzy-rainbow дорівнює $O\left(m \cdot e^{m \cdot 2^{2d}}\right)$, $O\left(m \cdot e^{m \cdot s \cdot 2^{d}}\right)$ і $O\left(m \cdot e^{m \cdot s \cdot 2^{2d}}\right)$ відповідно, де m – число рядків, s – кількість різних функцій переходу для fuzzy-rainbow-таблиць, d – біти в точці розрізнення, а для rainbow-таблиць приймається $t = s \cdot 2^d$.

Порівняння складності побудови доцільно проводити для одного розміру таблиці (фіксованої кількості рядків m і стовпців t). В цьому

випадку для методу на основі точок розрізнення $t = 2^d$, $d = \log_2 t$ і складність дорівнює $O\left(m \cdot e^{m \cdot t^2}\right)$. Побудова rainbow-таблиць вимагає $O\left(m \cdot e^{m \cdot t}\right)$ операцій. Для метода fuzzy-rainbow $s \cdot 2^{2d} = s \cdot 2^d \cdot 2^d = t \cdot 2^d = t \cdot \left(\frac{t}{s}\right) = \frac{t^2}{s}$, звідки складність $O\left(m \cdot e^{\frac{m \cdot t^2}{s}}\right)$. Слід зазначити, що для граничних значень ($s = 1$ або $s = t$) fuzzy-rainbow вироджується в таблицю на основі точок розрізнення або в rainbow-таблицю.

Порівняння складності обох етапів різних методів наведено в табл. 1.2, де m – число рядків, t – число стовпців, s – кількість різних функцій переходу для fuzzy-rainbow-таблиць, при цьому завжди $s < t$.

Таблиця 1.2 – Порівняння складності різних методів криптоаналізу на основі передобчислення

Етап/метод	Точки розрізнення	Rainbow	Fuzzy-rainbow
Предвирахування	$O\left(m \cdot e^{m \cdot t^2}\right)$	$O\left(m \cdot e^{m \cdot t}\right)$	$O\left(m \cdot e^{\frac{m \cdot t^2}{s}}\right)$
Оперативний	$O(\log m)$	$O\left(t^2 \cdot \log m\right)$	$O(t \cdot s \cdot \log m)$

Порівняння обчислювальної складності побудови таблиць однакового розміру (яка призводить до однакової ймовірності успіху на оперативному етапі наведено на рис. 1.5). Потужність множини значень $2^k = 2^{32}$, кількість рядків у таблицях прийнято $m = 2^{16}$. Для fuzzy-rainbow-таблиці задано використання $s = 4$ різних функцій. По осі абсцис задано кількість стовпців в таблиці t для rainbow-метода, для точок розрізнення $t = 2^d$ (d – число нульових бітів в кінці ланцюжка), для fuzzy-rainbow-метода $t = s \cdot 2^d$. Верхня крива відповідає методу на основі точок розрізнення, середня - fuzzy-rainbow, нижня - rainbow.

Для 500 тис. генеруємих рядків по методу точок розрізнення досяжна середня довжина рядка в 365 елементів. Складність передобчислення дорівнює $5 \cdot 10^5 \cdot 365 \approx 2^{27,44}$, а ймовірність успіху дорівнює $\frac{365 \cdot 2^{16}}{2^{32}} = 6 \cdot 10^{-3}$. Така ж ймовірність успіху досягається при генерації 65700 рядків для rainbow-таблиць (складність $2^{24,5}$) або 106280 рядків для rainbow-таблиць (складність $2^{25,2}$). Як видно з графіка на рис. 2.5, для методу на основі точок розрізнення досягнуто практично порогове значення ймовірності успіху, при експоненційному зростанні складності довжина рядка збільшується незначно. Графік додатково ілюструє зроблений раніше висновок, що метод на основі точок розрізнення доцільно використовувати тільки спільно з іншими методами.

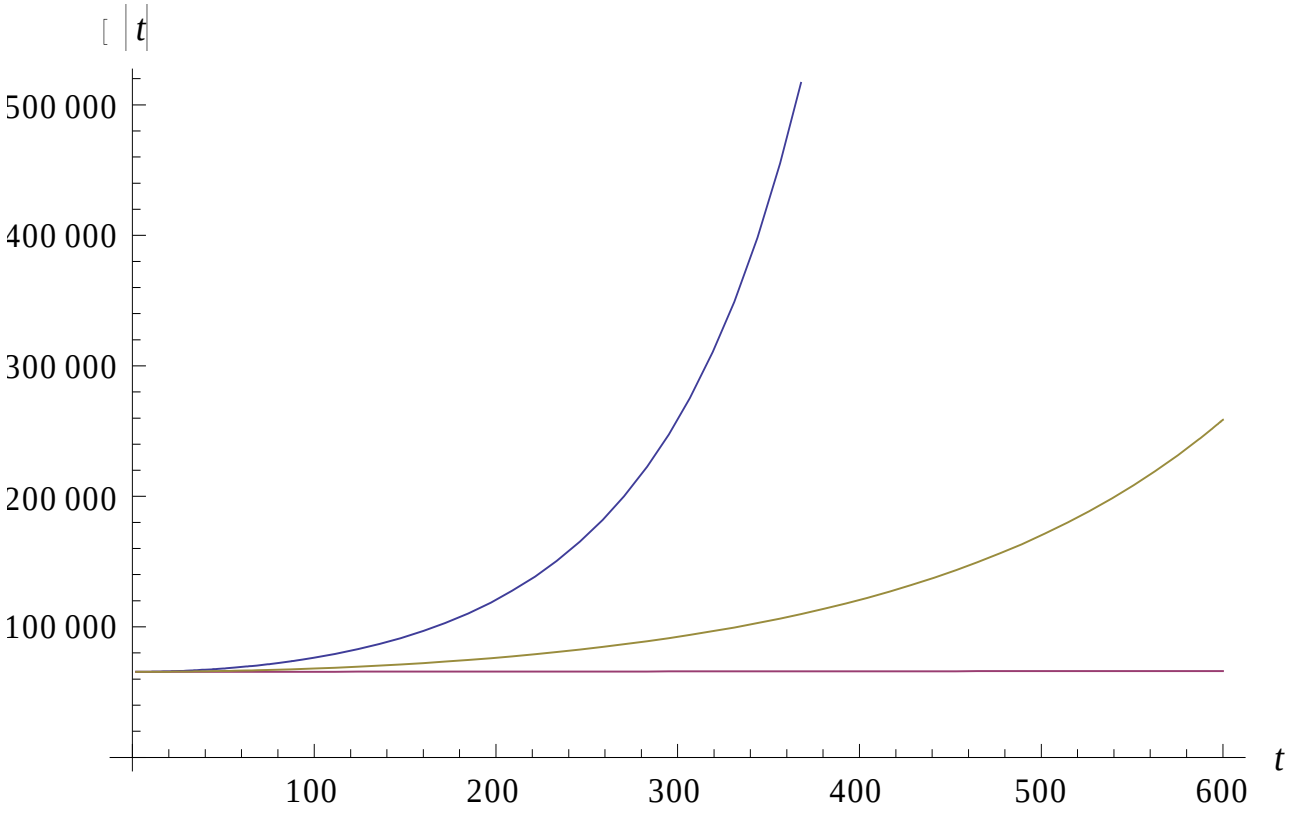


Рисунок 1.5 – Порівняння складності побудови різних таблиць для однакової ймовірності успіху криптоаналіза

Порівняння складності побудови rainbow- і fuzzy-rainbow-таблиць наведено на рис. 1.6. Тут також потужність множини значень $2^k = 2^{32}$,

кількість рядків у таблицях прийнято $m = 2^{16}$, але fuzzy-rainbow-таблиця використовує $s = 2048$ різних функцій. Для 500 тис. генеруються рядків по методу fuzzy-rainbow середня довжина рядка дорівнює 16262 елемента. Складність передобчислення дорівнює $5 \cdot 10^5 \cdot 16262 \approx 2^{32,9}$ (вище прямого перебору, але практично досяжна за рахунок різних початкових точок), а ймовірність успіху дорівнює $\frac{16262 \cdot 2^{16}}{2^{32}} \approx 0{,}25$. Для rainbow-таблиць в цьому випадку необхідно згенерувати 83992 рядки (складність $83992 \cdot 16262 \approx 2^{30,3}$).

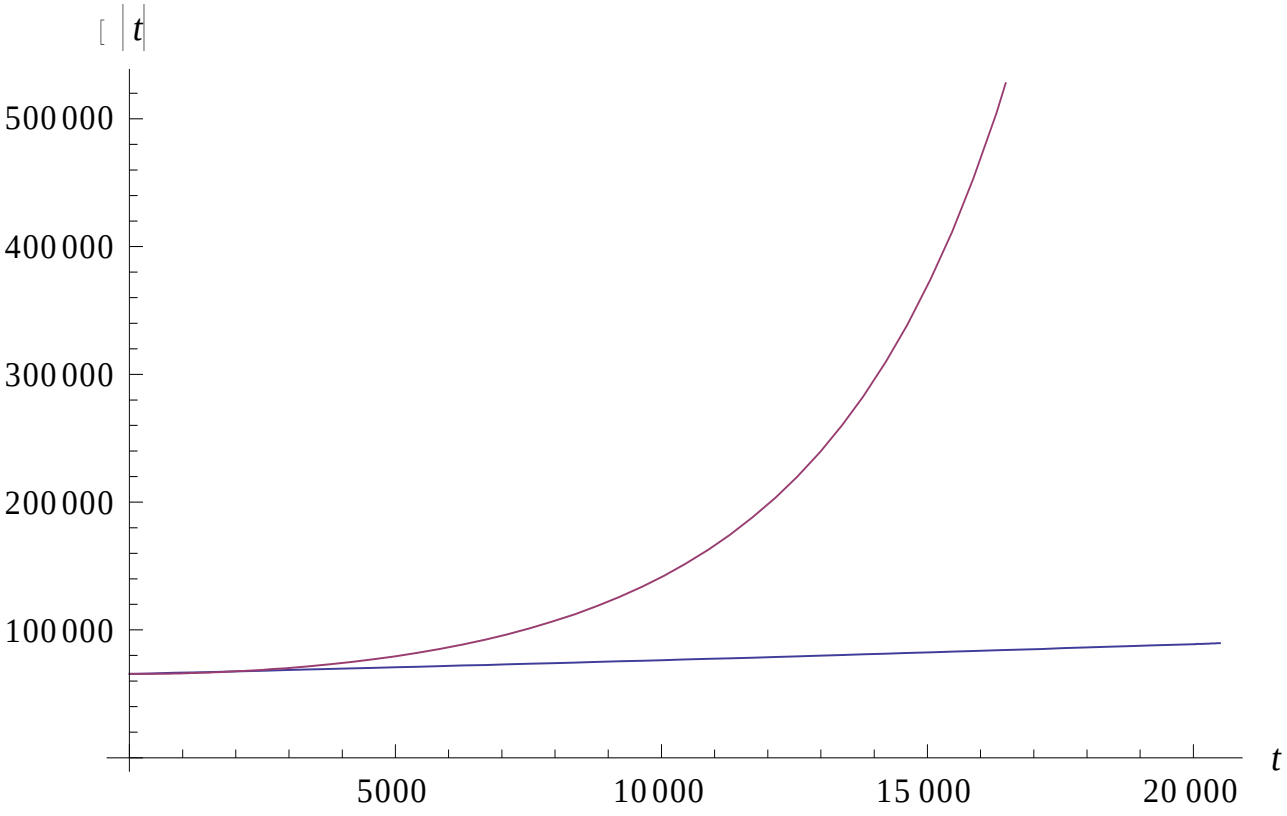


Рисунок 1.6 – Порівняння складності передобчислення rainbow і fuzzy-rainbow для однакової ймовірності успіху криптоаналізу

З розрахунків видно, що для розглянутих параметрів приблизно 8 кратне зростання складності побудови fuzzy-rainbow-таблиці ($2^{32,9} / 2^{30,3}$)дає можливість 8-кратного скорочення $(16262 / 2048)$ часу виконання оперативного етапу.

Таким чином, комбінування методу на основі точок розрізнення і rainbow дозволяє збільшити ймовірність успіху криптоаналіза через зростання часу побудови таблиці.

Додаткове збільшення ймовірності успіху можна отримати за рахунок використання декількох fuzzy-rainbow-таблиць.

1.9 Розрахунок безпечних параметрів симетричних криптографічних примітивів в умовах реалізації атак на основі передобчислення

Алгебраїчні моделі (1.1) - (1.4), призначені для аналізу стійкості до атак переборного типу, розглядають криптографічний примітив без урахування його внутрішньої структури. Відповідно, стійкість до цих методів криптоаналізу визначається виключно зовнішніми параметрами, такими як довжина ключа, значення на виході хеш-функції тощо.

Для успішної протидії порушника 3-го рівня (спеціальна служба економічно і технологічно розвиненої держави [15]) доцільно розглянути наступну модель атаки.

Час експлуатації криптографічного алгоритму - 30 років. Зокрема, TDEA (TripleDES) введено в дію в 1998 році і дозволений до використання до і після 2015 року [16], ГОСТ 28147 89 використовується вже більше 20 років, і т.д.

В державі кожну секунду формується близько 10 тис. повідомлень, для яких використовується криптографічний захист для передачі по відкритому каналу зв'язку.

Криптоаналітична служба порушника має можливість перехоплювати і зберігати кожну криптограму [17], яка залишається актуальною для розкриття протягом усього терміну використання алгоритму.

У розпорядженні криптоаналітика є ефективна обчислювальна система, що складається з 100 млн. процесорів, що працюють на частоті 5 ГГц кожен

(близько до практично досяжному межі масових напівпровідникових технологій) з конвеєрною архітектурою.

В рамках традиційної моделі безпечний час визначається як:

$$t_б = p_{вскр} \frac{N_{кл}}{\vartheta}, \qquad (1.32)$$

де $p_{вскр}$ – допустима максимальна ймовірність успіху криптоаналітика;

$N_{кл}$ – потужність множини невідомого параметра, найчастіше - кількість ключів алгоритму шифрування;

ϑ – продуктивність криптоаналітичної системи.

Задавши допустиму максимальну ймовірність успіху криптоаналітика $p_{вскр} = 10^{-9}$, і час аналізу 30 років, з (1.32) можна визначити необхідну потужність множини невідомого параметра (ключів) в рамках даної моделі атаки:

$$N_{кл} = \left(p_{вскр}\right)^{-1} \cdot \vartheta \cdot t_a = 10^9 \cdot \left(100 \cdot 10^6 \cdot 5 \cdot 10^9\right) \cdot \left(30 \cdot 365 \cdot 24 \cdot 60^2\right) \approx 2^{118,5},$$

тобто, наприклад, для алгоритму шифрування досить використання 128-бітного ключа (за умови неефективності всіх методів аналізу, крім перебірних).

У той же час, якщо розглянути модель із застосуванням rainbow-таблиці (максимальна ймовірність успіху при найменших ресурсах для передобчислення), буде потрібно збільшити довжину ключа.

Обмеживши час проведення оперативного етапу криптоаналізу 1 місяць, з умови нерозпаралелюваності генерації ланцюжка для таблиці передобчислення максимальна довжина рядка дорівнює $t = \left(5 \cdot 10^9\right) \cdot \left(30 \cdot 24 \cdot 60^2\right) \approx 2^{53,5}$.

За 30 років роботи криптоаналітичний комплекс сформує таблицю, яка містить $m \times t = \left(100 \cdot 10^6 \cdot 5 \cdot 10^9\right) \cdot \left(30 \cdot 365 \cdot 24 \cdot 60^2\right) \approx 2^{88,6}$ елементів і складається з $2^{88,6-53,5} = 2^{35,1}$ рядків. Слід зазначити, що для зберігання такої

таблиці буде потрібно всього $2^5 \cdot 2^{35,1} = 2^{40,1} \approx 1$ ТБ довготривалої пам'яті. Збільшивши цей обсяг, можна значно скоротити час виконання оперативного етапу.

Користувачі криптографічного алгоритму за 30 років роботи сформують $n_{\text{сообщ}} = 10^4 \cdot \left(30 \cdot 365 \cdot 24 \cdot 60^2\right) \approx 2^{43,1}$ повідомлень. Імовірність забезпечення безпеки всіх повідомлень $(m \times t >> n_{\text{сообщ}})$ дорівнює $p_{ст} = \left(1 - \frac{m \times t}{N_{кл}}\right)^{n_{\text{сообщ}}} \approx e^{-\frac{(m \times t) \cdot n_{\text{сообщ}}}{N_{кл}}}$.

З $p_{вскр} = 10^{-9}$ визначається $p_{ст} = 1 - p_{вскр}$, звідки необхідна потужність множини значень невідомого параметра дорівнює:

$$N_{кл} = -\frac{(m \times t) \cdot n_{\text{сообщ}}}{\ln\left(1 - p_{вскр}\right)} \tag{1.33}$$

Для даної моделі атаки:

$$N_{кл} = -\frac{\left(\left(100 \cdot 10^6 \cdot 5 \cdot 10^9\right) \cdot \left(30 \cdot 365 \cdot 24 \cdot 60^2\right)\right) \cdot \left(10^4 \cdot \left(30 \cdot 365 \cdot 24 \cdot 60^2\right)\right)}{\ln\left(1 - 10^{-9}\right)} \approx 2^{162}.$$

Таким чином, для розглянутої моделі атаки потужність множини значень невідомого параметра застосовуваного симетричного криптографічного примітиву повинна бути не менше 2^{162} за умови відповідності властивостей випадковому відображенню і неможливості реалізації аналітичної атаки, що використовує особливості внутрішньої структури перетворення.

Слід зазначити, що для TDEA (TripleDES) навіть з 3-ма ключами не задовольняє вимогам стійкості в розглянутої моделі атаки, оскільки атака типу «зустріч посередині» зменшує ефективну довжину ключа з 168 до 112 бітів.

Модель атаки бере до уваги сучасні напівпровідникові технології з перспективами їх вдосконалення, але не розглядає гіпотетичні квантові комп'ютери з великою кількістю елементарних частинок, що мають «заплутане» стан. Складність побудови таких систем зростає експоненціально, і за 15 років досліджень вдалося побудувати систему на основі всього 14 кубітів (збільшити довжину від 3 до 14 кубітів), в той час як для аналізу сучасного симетричного примітиву потрібний квантовий регістр довжиною понад 100 кубітів, причому вони повинні мати загальний «заплутаний» стан. Анонсовані квантові комп'ютери компанії D Wave [18] (США) такою властивістю не володіють.

Потужність множини значень 2^{162} отримана з використанням ряду припущень, зокрема, ймовірності компрометації 10^{-9} і повної відсутності ефективних криптоаналітичних атак на протязі 30 років експлуатації. Для проектованих симетричних криптографічних перетворень необхідно реалізувати запас стійкості з урахуванням особливостей існуючих і перспективних архітектур. Зокрема, з урахуванням запасу стійкості і вирівнюванням слова по 64 і 128-бітової границі, доцільно використовувати потужність множини значень не менше 2^{192} або 2^{256}. Для ефективної програмної або програмно-апаратної реалізації доцільно використовувати значення $N_{кл} \geq 2^{256}$ при розробці перспективних високостійких симетричних криптографічних примітивів.

Відповідно, перспективний продуктивний блоковий симетричний алгоритм шифрування, що забезпечує високий рівень криптографічнї стійкості, повинен підтримувати довжину ключа 256 бітів і більш, як це реалізовано в алгоритмах ГОСТ 28147 89, AES-256 або «Калина».

Перспективний поточний шифр (генератор гами), крім ключів такої довжини, повинен мати більш довгий регістр для внутрішніх станів. Дійсно, через подвійне відображення (ключ - внутрішній стан - гамма, див. п.2.2), можлива поява еквівалентних ключів. Імовірність їх виникнення можна

оцінити як $P_{эк} = \left(1 - \frac{1}{2^s}\right)^{2^k} \approx e^{-2^{s-k}}$, де $2^k = \#K$ – потужність множини ключів, $2^s = \#S$ – потужність множини внутрішніх станів. Звідси $s = v + k$, де $2^{-v} = P_{эк}$ – ймовірність появи еквівалентних ключів. Якщо $\#K = 2^{256}$, тобто довжина ключа поточного шифру дорівнює 256 бітів, і $P_{эк} = 2^{-256}$, то необхідна довжина регістра для зберігання внутрішнього стану дорівнює 512 бітам.

Для хеш-функцій (і кодів аутентифікації повідомлень) необхідно брати до уваги, що криптоаналітик задіє потужності комплексу для генерації вибірки хешированих значень з метою пошуку колізій. Імовірність їх відсутності дорівнює $p_{нк} = 1 - p_к = \left(1 - \frac{m \times t}{N_{зн}}\right)^{m \times t} \approx e^{-\frac{(m \times t)^2}{N_{зн}}}$, звідки:

$$N_{зн} = -\frac{(m \times t)^2}{\ln(1 - p_к)} \tag{1.34}$$

При $p_к = 10^{-9}$ потужність множини значень хеш-функції $N_{зн} \approx 2^{207}$ аналогічно шифрам, з урахуванням можливості розробки ефективних аналітичних атак і необхідності ефективної програмної, апаратної та програмно-апаратної реалізації на сучасних і перспективних архітектур, потужність множини значень хеш-функцій і кодів аутентифікації повідомлень доцільно поставити не менше ніж $N_{зн} \geq 2^{256}$.

ВИСНОВКИ

1. Для забезпечення високого рівня стійкості симетричні криптографічні примітиви повинні моделювати властивості випадкових відображень, таких як випадкові перестановки (блокові шифри) і випадкові функції (хеш-функції, генератори псевдовипадкових послідовностей, потокові шифри).

2. Для виключення впливу властивостей різних моделей джерел відкритих повідомлень при аналізі стійкості (але не дешифрування) сучасних криптографічних примітивів розглядаються, як правило, атаки з відомими або обраними відкритими текстами.

3. Аналіз стійкості симетричних криптографічних перетворень на основі параметрів вхідних і вихідних послідовностей (розмір блоку, ключа тощо) в рамках припущення, що примітив має властивості випадкового відображення. Для генераторів гами необхідно розглядати потужність множини, що описує внутрішній стан (для обліку ймовірності появи еквівалентних ключів).

4. Метою реалізації атак переборного типу є визначення значення невідомого параметра з множини обмеженою потужності на основі аналізу вхідних і вихідних послідовностей.

5. Пошук невідомого параметра криптографічного примітиву може виконуватися за рахунок тимчасових / обчислювальних або просторових ресурсів, а також при виборі компромісу між ними.

6. Найбільш відомими підходами до реалізації компромісів при передобчисленній таблиці Хеллмана, на основі точок розрізнення, rainbow- і fuzzy-rainbow-таблиці. При практичній реалізації атаки метод Хеллмана є менш ефективним, ніж інші методи.

7. Таблиці на основі точок розрізнення містять виключно унікальні елементи і мають найменший час криптоаналіза. Однак, цей метод має

найбільшу складність передобчислення і не дозволяє досягти досить високу ймовірність успіху на оперативному етапі.

В рамках розробленої математичної моделі доведено, що складність побудови експоненційно залежить від кількості рядків в таблиці і від довжини кожного ланцюжка, в свою чергу, експоненційно залежить від числа бітів в точці розрізнення. Імовірність успіху лінійно залежить від кількості рядків і їх довжини.

При побудові рядка таблиці для пошуку зациклення доцільно користуватися запропонованим імовірнісним алгоритмом, що мають складність $O(1)$, замість детермінованого зі складністю $O(n \cdot \log n)$.

8. Rainbow-таблиці допускають існування колізійних елементів, однак імовірність їх появи порівняно невелика в порівнянні з іншими типами. Цей метод має найменшу складність попереднього етапу, але оперативний етап вимагає найбільшого часу в порівнянні з іншими типами таблиць.

В рамках розробленої математичної моделі доведено, що складність побудови експоненційно залежить від кількості рядків в таблиці і їх довжини (кількості різних функцій переходу). Імовірність успіху лінійно залежить від розміру таблиці.

9. Fuzzy-rainbow-таблиці об'єднує можливості rainbow-таблиць і ланцюжків на основі точок розрізнення, дозволяючи отримати оптимальне компромісне рішення для досягнення високої ймовірності успіху, порівняно малий час криптоаналізу і прийнятний рівень складності на етапі передобчислення.

В рамках розробленої математичної моделі доведено, що складність побудови експоненційно залежить від кількості рядків в таблиці і їх довжини і кількості застосовуваних різних функцій переходу. Імовірність успіху лінійно залежить від розміру таблиці.

Застосування декількох fuzzy-rainbow-таблиць дозволяє досягти високої ймовірності успіху на оперативному етапі, порівняно малого часу криптоаналізу і прийнятного рівня складності на етапі передобчислення.

Додаткове експоненціальне зниження складності попереднього етапу можна досягти за рахунок лінійного збільшення складності проведення етапу криптоаналізу.

10. З розглянутих методів перед вирахування найбільш ефективним з точки зору максимізації ймовірності успіху, зменшення часу передобчислення і реалізації атаки в реальному масштабі часу є fuzzy-rainbow-таблиці за умови можливості виконати перебір здебільшого множини значень невідомого параметра. При відсутності такої можливості для досягнення найбільшої ймовірності успіху на оперативному етапі найбільш ефективним методом є rainbow-таблиці.

11. В рамках розглянутої моделі криптоаналітичної атаки, коли криптографічне перетворення експлуатується протягом 30 років, причому кожну секунду легітимні користувачі формують 10 тис. Криптограми, і необхідно забезпечити захист від порушника 3-го рівня (спеціальна служба економічно і технологічно розвиненої держави), який володіє суперкомп'ютером на основі напівпровідникової технології, які виконують $5 \cdot 10^9 \cdot 10^8 = 5 \cdot 10^{17}$ операцій прямого криптографічного перетворення в секунду, з імовірністю розтину не більше 10^{-9}, для забезпечення стійкості ефективно реалізується (апаратно і програмно) криптографічного примітиву:

– довжина ключа перспективного блочного симетричного алгоритму шифрування повинна бути не менше 256 бітів;

– перспективний потоковий шифр (генератор гами), крім ключів такої довжини, при ймовірності наявності еквівалентних внутрішніх станів $P_{эк} = 2^{-256}$, повинен мати регістр для внутрішніх станів довжиною не менше 512 бітів;

– для хеш-функцій і кодів аутентифікації повідомлень необхідно мати сформоване значення довжиною не менше 256 бітів.

З урахуванням досить тривалого терміну експлуатації, протягом якого можуть бути запропоновані нові ефективні методи криптоаналізу і

технологічні засоби, доцільно передбачити додатковий запас стійкості за зовнішніми параметрами перспективного симетричного криптографічного примітиву.

2 РОЗРОБКА МЕТОДУ ПОРІВНЯННЯ ВИСОКОРІВНЕВИХ КОНСТРУКЦІЙ СИМЕТРИЧНИХ БЛОКОВИХ ШИФРІВ НА ОСНОВІ ЛАНЦЮГА ФЕЙСТЕЛЯ, СХЕМИ ЛЕЙ-МЕССІ І SPN-СТРУКТУРИ З ВІДПОВІДНОЮ КІЛЬКІСТЮ РАУНДІВ ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ ОСНОВНИХ КОНСТРУКЦІЙ, ТА ОБҐРУНТУВАННЯ ВИБОРУ СТРУКТУРИ ПЕРСПЕКТИВНОГО АЛГОРИТМУ ШИФРУВАННЯ

2.1 Високорівневі конструкції симетричних блокових шифрів

Сучасні блокові шифри представляють ітеративні перетворення на основі рандомізуючого бієктивного або сюр'єктивного відображення (див. розділ 2.4), яке використовується в складі однієї з таких високорівневих конструкцій:

– ланцюг Фейстеля;

– SPN-структура;

– схема Лей-Мессі.

Ланцюг Фейстеля використовується в алгоритмах DES [19], ГОСТ 28147 89 [20], Camellia [21] та ін. Шифрувальне перетворення одного раунду ланцюга Фейстеля (рис. 1.1) представляється у вигляді:

$$\varphi_{i,k_i}\left(x_i^L, x_i^R\right) = \left(x_i^R, f\left(x_i^R, k_i\right) \oplus x_i^L\right), \tag{2.1}$$

де x_i^L, x_i^R – ліва і права половина вхідного значення циклу шифрування;

$f(\cdot)$ – раундова функція блочного шифру;

k_i – раундовий ключ.

Крім ланцюга Фейстеля, однією з найбільш поширених високорівневих конструкцій симетричних блокових шифрів є SPN-структура

(substitution-permutation network). Зокрема, це перетворення використано в найбільш широко поширеному в усьому світі алгоритмі AES / Rijndael, шифрах Anubis [22], GrandCru [23], Noekeon [24], «Калина» [25] та ін.

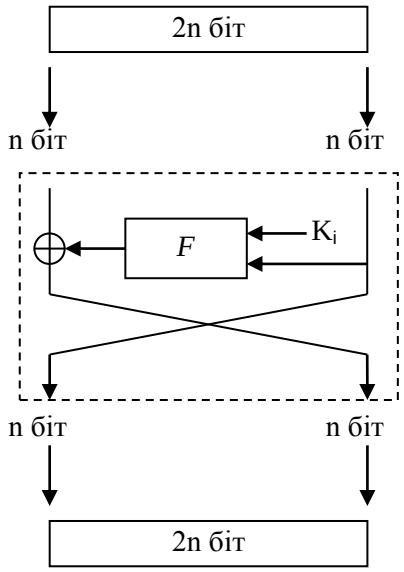


Рисунок 2.1 – Один раунд ланцюга Фейстеля

Шифрувальне перетворення одного раунду SPN-структури (рис. 2.2) представляється у вигляді:

$$\phi_{i,k_i}(x_i) = P\left(S(x_i)\right) \oplus k_i, \tag{2.2}$$

де x_i – вхідне значення циклу шифрування;

$S(\cdot)$ – шар нелінійного перетворення;

$P(\cdot)$ – шар лінійного перетворення;

k_i – раундовий ключ.

Третьою основною високорівневою конструкцією блокових шифрів є схема Лей-Массі [26,27]. Це перетворення зустрічається рідше, ніж вже розглянуті, і на його базі побудовані шифри FOX (IDEA-NXT) [27], IDEA [28], «Мухомор» [29].

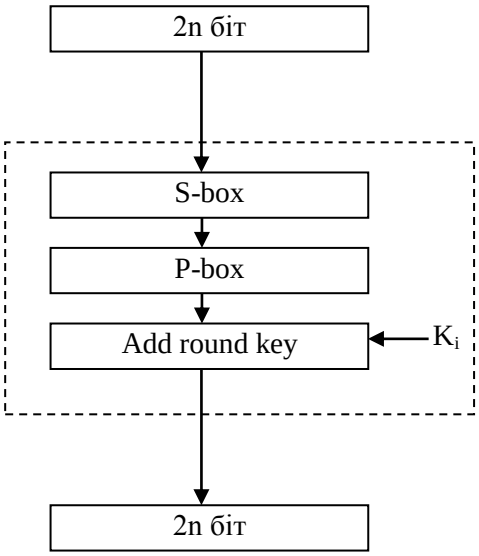


Рисунок 2.2 – Один раунд SPN-структури

Шифрувальне перетворення одного раунду схема Лей-Массі без завершального ортоморфного перетворення (рис. 2.3) представляється у вигляді:

$$\varphi_{i,k_i}\left(x_i^L, x_i^R\right) = \left(x_i^L \oplus f\left(x_i^L \oplus x_i^R, k_i\right), x_i^R \oplus f\left(x_i^L \oplus x_i^R, k_i\right)\right), \qquad (2.3)$$

де x_i^L, x_i^R – ліва і права половини вхідного значення циклу шифрування;

$f(\cdot)$ – раундова функція блочного шифру;

k_i – раундовий ключ.

Крім розглянутих структур, можуть використовуватися узагальнення варіації розглянутих конструкцій. Зокрема, шифр Skipjack [30], блоковий алгоритм в основі хеш-функцій SHA-1, SHA-2, малоресурсний (lightweight) шифр Clefia зі стандарту ISO / IEC 29192 2:2012 [31] побудовані на узагальненому ланцюгу Фейстеля, де на одному раунді обробляється не половина, а $1/4$ або $1/8$ вхідного блоку з відповідним збільшенням кількості циклів шифрування. За аналогічним принципам побудовані алгоритми FOX і

«Мухомор» для обробки блоків даних великої довжини, коли вихідне значення раундової функції має довжину менше, ніж половину блоку.

Крім того, слід зазначити, що з усіх розглянутих схем тільки SPN-структура вимагає бієктивне відображення в раундовому перетворенні для реалізації коректного шифрування і розшифрування. Для інших конструкцій така властивість забезпечується при будь-яких вихідних значеннях $f(\cdot)$, що може використовуватися для реалізації недокументованих можливостей шифру (детальніше див. п. 2.4.8).

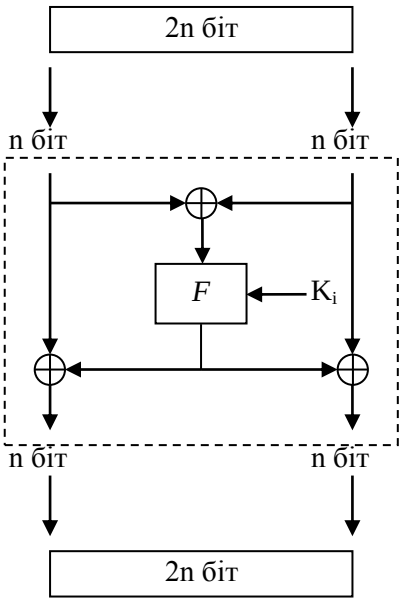


Рисунок 2.3 – Один раунд схеми Лей-Массі

За результатами проведених криптографічних конкурсів [32-34] автори різних шифрів здійснюють вибір високорівневих конструкцій на рівні переваг, без аналітичного обґрунтування ефективності.

У відкритих публікаціях є обґрунтування верхньої межі розрізнення 3-раундового ланцюга Фейстеля, побудованого на основі випадкових функцій (див. рис. 2.1), і випадкової перестановки (використовуються позначення з п. 2.3.3) [35,36].

Точне значення ймовірності розрізнення 3-раундового ланцюга Фейстеля на основі випадкових функцій і випадкових перестановок, а також апроксимації для більш зручних розрахунків наведені у розділі 2.3.

Для SPN-структури і схеми Лей-Мессі відсутні навіть такі результати, і за відкритими публікаціями в зарубіжній і вітчизняній пресі неможливо виконати аналітичну оцінку ефективності та порівняння різних високорівневих конструкцій блокових шифрів.

Крім того, для блокових шифрів актуальним завданням є побудова схеми розгортання ключа, що забезпечує стійкість до атак на зв'язаних ключах і при цьому ефективної програмної і апаратної реалізації (складність такої атаки на ГОСТ 28147-89 еквівалентна 2^{12} для відновлення 256-бітного ключа, для AES – 2^{119} для 192- і 256-бітного ключа).

2.2 Оцінка ефективності розрізнення БСШ на основі ланцюга Фейстеля

Даний розділ присвячений аналізу властивостей такої загальновідомої конструкції побудови блочного шифру, як ланцюг Фейстеля. Особливу популярність дана конструкція придбала після створення одного з найбільш відомих блокових симетричних шифрів – алгоритму DES [19].

У розділі наводиться загальний опис конструкції, проводиться аналіз існуючих робіт в даній області, а також представлені нові результати, отримані в ході досліджень.

2.2.1 Загальна схема БСШ на основі ланцюга Фейстеля

У 1971 р Хорстом Фейстелем був запропонований метод побудови блочного шифру на основі певної багаторазово повторюваної (ітераційної) структури [37]. На кожній ітерації виконується раундове перетворення під дією раундового ключа. Операції шифрування і розшифрування можна виразити таким чином:

$$L_i = R_{i-1} \oplus F(L_{i-1}, K_{i-1}), \quad (2.4)$$

$$R_i = L_{i-1}. \quad (2.5)$$

У даному випадку R_i и L_i – це права і ліва частини повідомлення на *i*-ому раунді (відповідно R_0 и L_0 – це вихідне повідомлення, а R_n и L_n – це зашифроване повідомлення, де *n* – кількість раундів).

Загальна схема приведена на рис. 2.1.

Операції шифрування і розшифрування на кожному етапі дуже прості, і при певній доробці збігаються, вимагаючи тільки зворотного порядку використовуваних ключів. Шифрування за допомогою даної конструкції легко реалізується як на програмному рівні, так і на апаратному, що забезпечує широкі можливості застосування. Більшість сучасних блокових шифрів використовують мережу Фейстеля в якості основи. Першим шифром, побудованим за допомогою такої моделі, був шифр Люцифер [38]. Найпоширенішим є DES (Data Encryption Standart) [19], який довгий час залишався стандартом блокового шифрування в США. Також відомими представниками даного сімейства шифрів є Camelia [39-40], IDEA [28, 41-42], ГОСТ 28147-89 [20].

2.2.2 Теоретична оцінка ефективності розрізнення ланцюга Фейстеля

При оцінці ефективності високорівневої конструкції шифру стоїть завдання наступного плану: потрібно з певною ймовірністю на заданому наборі вихідних даних (відкриті і відповідні їм зашифровані тексти) визначити структуру, яка використовувалася для генерації цих даних. Якщо ми можемо побудувати деякий алгоритм-розрізнювач, який з вірогідністю, відмінною від нуля, може відрізнити ланцюг Фейстеля від випадкової функції, то можна зробити висновок про те, що така конструкція має невисоку ефективність. Чим більша ймовірність розрізнення, тим гірше шифр. Оцінкою ефективності шифру може стати максимально можлива

ймовірність відрізнення його від випадкової функції. Така ймовірність буде різна для різної кількості раундів, проте верхню межу можна отримати і для всіх раундів, оскільки зі зростанням їхньої кількості ймовірність розрізнення буде тільки зменшуватися.

Слід зазначити, що для того, щоб залишити вплив на ймовірність розрізнення тільки високорівневої конструкції слід в якості раундових перетворень використовувати випадкові функції.

Для 1- та 2-раундової ланцюга Фейстеля побудова алгоритму-розрізнювача є тривіальним завданням, тому в подальшому всі дослідження будуть проводитися для кількості раундів 3 і більше.

Розглянемо загальну модель алгоритму-розрізнювача, яка представлена на рис. 2.4.

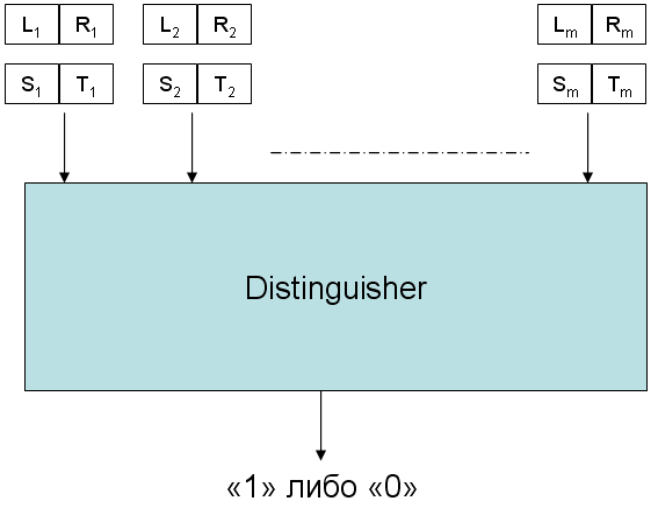


Рисунок 2.4 – Загальна модель алгоритму-розрізнювача

Як уже згадувалося вище, маємо деякий алгоритм-розрізнювач, на вхід якого подається певна кількість пар відкритих / зашифрованих текстів. По суті, блоковий шифр реалізує певну підмножину перестановок, кількість яких дорівнює кількості можливих ключів шифрування. Оскільки вибір такої підмножини визначається структурою шифру і не є випадковим, то можлива побудова алгоритму-розрізнювача, який міг би визначити, чи є конкретна перестановка випадково обраної із загального безлічі, або отриманої в результаті дії блочного шифру. Таким чином, аналізуючи вхідні дані, подані

на вхід алгоритму-розрізнювача, на виході алгоритму матимемо «1» або «0». «1» в тому випадку, якщо вважається, що вхідні дані були отримані за допомогою блочного шифру (ланцюга Фейстеля в даному випадку) і «0», якщо вважається, що це результат дії випадкової функції. Для ланцюга Фейстеля ймовірність появи «1» матиме певне значення. Однак і для випадкової функції ймовірність появи «1» на виході алгоритму-розрізнювача не дорівнює нулю, тому ймовірністю розрізнення ланцюга Фейстеля і випадковою функцією вважається різниця цих двох значень, яка також називається перевагою.

2.2.2.1 Максимальна ймовірність розрізнення ланцюга Фейстеля і випадкової функції

Значну кількість робіт присвячено дослідженням максимально можливого переваги ланцюга Фейстеля над випадковою функцією [26,43-46]. Зокрема, У. Маурер в своїй роботі [43] приводить лемму, що визначає межу переваги алгоритму-розрізнювача 3-раундового (і з кількістю раундів більше 3) ланцюга Фейстеля над випадковою функцією. Наведемо цю лему і її доказ повністю, оскільки в подальшому викладі будемо посилатися на окремі результати доказу.

При подальшому викладі будуть використовуватися наступні позначення:

I_n – безліч всіх можливих бітових векторів довжини *n*;

I_{2n} – безліч всіх можливих бітових векторів довжини *2n*;

F_n – множина функцій $F : I_n \to I_n$;

F_{2n} – множина функцій $F : I_{2n} \to I_{2n}$;

P_1 – ймовірність розрізнення ланцюга Фейстеля алгоритмом-розрізнювачем;

P_1^* – ймовірність розрізнення випадкової функції алгоритмом-розрізнювачем;

$x \bullet y$ – конкатенація двох значень x та y.

Нехай ми маємо 3-раундовий ланцюг Фейстеля (рис. 2.3). Позначимо її як функцію $f = \psi(F^n, F^n, F^n)$, що включає в себе також раундові функції f_1^*, f_2^*, f_3^* випадково вибрані з множини F_n. Вхідними аргументами функції f є блоки даних $x_i = L_i \bullet R_i$ довжиною $2n$ біт. L_i і R_i є відповідно лівою і правою частиною вхідного аргументу довжиною по n біт кожна. Вихідні значення будемо позначати як $f[L_i \bullet R_i] = [V_i \bullet T_i]$.

Нехай також g – це деяка функція, яка реалізує алгоритм-розрізнювач. На вхід функції g подаються $k \geq 2$ аргументів $x_1 \ldots, x_k$. Для кожного вхідного аргументу обчислюється вихідне значення $f[x_i] = y_i$. Значення y_i також має довжину $2n$ біт. Далі кожна пара значень $\{[x_i, y_i], [x_j, y_j]\}$, де $1 \leq i < j \leq k$, обробляється алгоритмом-розрізнювачем, результатом роботи якого є значення «0» або «1». Якщо хоча б одна пара значень привела до «1», то і повертається значення функції g також одно «1», інакше повертається значення дорівнює «0».

Лема 2.1. Для кожної функції $g : (\{0,1\}^{2n})^k \to \{0,1\}$ і для кожного набору з k аргументів $x_1 \ldots, x_k$ вірна наступна нерівність:

$$| P[g(f(x_1), \ldots, f(x_k)) = 1 : f \in_R \psi(F_n, F_n, F_n)] - P_g | \leq \frac{k^2}{2^n}. \qquad (2.6)$$
$$P_g = P[g(f(x_1), \ldots, f(x_k)) = 1 : f \in_R F_{2n}]$$

Трьохраундовий ланцюг Фейстеля наведено на рис. 2.5.

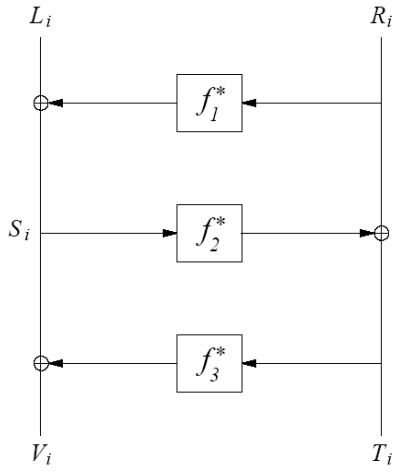


Рисунок 2.5 – Трьохраундовий ланцюг Фейстеля

Доказ леми 2.1. Визначимо S_i, T_i та V_i для всіх $1 \le i \le k$ в такий спосіб: $S_i = L_i \oplus f_1^*(R_i)$, $T_i = R_i \oplus f_2^*(S_i)$, $V_i = S_i \oplus f_3^*(T_i)$.

Відзначимо, що в разі, коли функція f з аргументами x_i розглядається як трираундовий процес, виходи першого, другого і третього раундів позначаються як $R_i \bullet S_i$, $S_i \bullet T_i$ та $T_i \bullet V_i = f(L_i \bullet R_i)$ відповідно. В подальшому можемо припустити, що всі x_i, при $1 \le i \le k$, різні. Вибір повторюваних аргументів не надасть ніякої нової інформації про структуру перетворення.

Нехай ε_S та ε_T позначають такі події, що всі $S_1,...,S_k$ та $T_1,...,T_k$ різні. Також нехай ε позначає подію, коли події ε_S та ε_T і обидва сталися.

Якщо відбулася подія ε_S, тоді значення $T_1 = R_1 \oplus f_2^*(S_1),...,T_k = R_k \oplus f_2^*(S_k)$ абсолютно випадкові, оскільки f_2^* – випадкова функція і, отже, $f_2^*(S_1),...,f_2^*(S_k)$ також випадкові величини. Аналогічно, якщо відбулася подія ε_T, тоді значення $V_1 = S_1 \oplus f_3^*(T_1),...,V_k = S_k \oplus f_3^*(T_k)$ абсолютно випадкові, оскільки f_3^* – випадкова функція.

Якщо обидві події ε_S та ε_T відбулися, тоді значення $f(x_1) = T_1 \bullet V_1, ..., f(x_k) = T_k \bullet V_k$ абсолютно випадкові і тому, функція $f = \psi(f_1^*, f_2^*, f_3^*)$ має поведінку випадкової функції.

Тому, верхня межа переваги ланцюга Фейстеля над випадковою функцією обмежена значенням:

$$| P[g(f(x_1),...,f(x_k)) = 1 : f \in_R \psi(F^n, F^n, F^n)] - P_g | \le 1 - P[\varepsilon]. \qquad (2.7)$$

Визначимо верхню межу як $1 - P[\varepsilon] = P[\bar{\varepsilon}]$, де $\bar{\varepsilon}$ – це подія протилежне ε. Подія $\bar{\varepsilon}$ – це набір з C_k^2 подій $\{S_i = S_j\}$, де $1 \le i < j \le k$, і C_k^2 подій $\{T_i = T_j\}$, де $1 \le i < j \le k$. Верхню межу ймовірності події $\bar{\varepsilon}$ можна отримати як суму ймовірностей кожного з подій складових $\bar{\varepsilon}$. Отже, отримаємо наступну нерівність:

$$1 - P[\varepsilon] = P[\bar{\varepsilon}] \le \sum_{1 \le i < j \le k} P[S_i = S_j] + \sum_{1 \le i < j \le k} P[T_i = T_j]. \qquad (2.8)$$

Для $i \ne j$ маємо:

$$P[S_i = S_j] = \begin{cases} 2^{-n}, якщо\_R_i \ne R_j; \\ 0, якщо\_R_i = R_j. \end{cases} \qquad (2.9)$$

Відзначимо, що коли $R_i = R_j$, тоді $P[S_i = S_j] = 0$. Це пов'язано з тим, що $L_i \bullet R_i \ne L_j \bullet R_j$ і, отже, $L_i \ne L_j$. Рівняння (2.7) показує, що $P[S_i = S_j] \le 2^{-n}$ для будь-яких $i \ne j$.

Аналогічно можна отримати, що $P[T_i = T_j] \le 2^{-n}$.

Отже, загальна кількість подій в правій частині (2.6) становить $2 \cdot C_k^2 = k(k-1) < k^2$.

Доказ леми 2.1 закінчено.

2.2.2.2 Уточнена оцінка максимально можливої ймовірності розрізнення ланцюга Фейстеля і випадкової функції

Максимально можливе значення переваги можна визначити більш точно, ніж у нерівності (2.3). Зокрема можна уточнити рівняння (2.6). Події $\{S_i = S_j\}$ та $\{T_i = T_j\}$ не є рівноймовірними, як передбачається в доказі леми 2.1 [47-50]. Також слід врахувати, що ймовірність виконання одного з сумісних подій не є простою сумою ймовірностей таких подій, оскільки слід враховувати також можливість їх спільного появи [51].

Більш точне значення $P[\bar{\varepsilon}]$ може бути представлене в такий спосіб.

Лема 2.2. Імовірність події $\bar{\varepsilon}$ для кожного набору з $k \geq 2$ аргументів $x_1,...,x_k$ дорівнює:

$$\begin{aligned} \mathrm{P}[\bar{\varepsilon}] \leq 1-(1-\frac{1}{2^n})^{2(k-1)}(1-\frac{1}{2^n-1})^{2(k-2)}(1-\frac{1}{2^n-2})^{2(k-3)}... \\ ...(1-\frac{1}{2^n-(k-2)})^{2(k-(k-1))} = 1-\prod_{i=0}^{k-2}(1-\frac{1}{2^n-i})^{2(k-(i+1))}. \end{aligned} \quad (2.10)$$

Доказ леми 2.2. Розглянемо спочатку ймовірність події $\bar{\varepsilon}_S$, – це ймовірність знаходження серед k аргументів такої пари, що $\{S_i = S_j\}$ при $i \neq j$. Кількість можливих пар аргументів становить $C_k^2 = \frac{k(k-1)}{2}$. Однак, у міру перевірки на рівність певних значень $\{S_i; S_j\}$ ймовірність рівності наступних значень буде змінюватися.

Розглянемо перші $k-1$ пар значень $[\{S_1;S_2\},\{S_1;S_3\},...,\{S_1;S_k\}]$. S_1 і S_i, $2 \leq i \leq k$, в даному випадку є випадковими величинами, тому що функція f_1^* є випадковою (приймаємо умову, що $R_i \neq R_j$, інакше $P[S_i = S_j] = 0$). Імовірність події $\{S_1 = S_i\}$ в такому випадку становить $P[S_1 = S_i] = 2^{-n}$.

Загальна ймовірність того, що жодна з подій $\{S_1 = S_i\}$, $2 \le i \le k$, не відбудеться, дорівнює $(1-\frac{1}{2^n})^{(k-1)}$.

Розглянемо наступні $k-2$ значень $[\{S_2;S_3\},\{S_2;S_4\},...,\{S_2;S_k\}]$. S_1 і S_i, $3 \le i \le k$, в даному випадку також є випадковими величинами, проте точно відомо, що $S_2 \neq S_1$ і $S_i \neq S_1$, тому що такі події вже були розглянуті на попередньому етапі. Відповідно ймовірність події $\{S_2 = S_i\}$ в такому випадку становить $P[S_2 = S_i] = \frac{1}{2^n - 1}$. Загальна ймовірність того, що жодна з подій $\{S_2 = S_i\}$, $3 \le i \le k$, не відбудеться, дорівнює $(1-\frac{1}{2^n-1})^{(k-2)}$.

Аналогічно можна порахувати ймовірності для інших наборів пар. Всього таких наборів може бути $k-1$. Кожен з них матиме на один елемент (одну пару) менше. Останній набір складатиметься з однієї пари $\{S_{k-1} = S_k\}$ і ймовірність такої події становитиме $P[S_{k-1} = S_k] = \frac{1}{2^n - (k-2)}$.

Загальну ймовірність $P[\bar{\varepsilon}_S]$ можна знайти відповідно до закону додавання ймовірностей для сумісних подій. Така ймовірність матиме таке значення:

$$\begin{aligned} \mathrm{P}[\bar{\varepsilon}_S] \le 1-(1-\frac{1}{2^n})^{(k-1)}(1-\frac{1}{2^n-1})^{(k-2)}(1-\frac{1}{2^n-2})^{(k-3)}... \\ ...(1-\frac{1}{2^n-(m-2)})^{(k-(k-1))} = 1-\prod_{i=0}^{k-2}(1-\frac{1}{2^n-i})^{(k-(i+1))}. \end{aligned} \quad (2.11)$$

Імовірність події $\bar{\varepsilon}_T$ – це ймовірність знаходження серед k аргументів такої пари, що $\{T_i = T_j\}$ при $i \neq j$. $\bar{\varepsilon}_T$ знаходиться аналогічно $\bar{\varepsilon}_S$. Фактично, якщо розглядати подію $\bar{\varepsilon}$, при яких виконується хоча б одне з рівності $\{T_i = T_j\}$ або $\{S_i = S_j\}$, то кожен з наборів, що розглядаються вище, буде мати в 2 рази більше можливих пар. Тобто перший набір складатиметься з

пар $[\{S_1;S_2\},\{S_1;S_3\},...,\{S_1;S_k\}]$ і $[\{T_1;S_2\},\{T_1;T_3\},...,\{T_1;T_k\}]$. Тоді загальна ймовірність того, що жодна з подій $\{S_1 = S_i\}$ і $\{T_1 = T_i\}$, $2 \le i \le k$, не відбудеться, дорівнює $(1-\frac{1}{2^n})^{2(k-1)}$. Аналогічно можна знайти ймовірності для інших наборів пар.

Тоді ймовірність події $\bar{\varepsilon}$ матиме значення, представлене у формулі (2.8).

Доказ закінчено.

Формула (2.10) є досить складною. Хорошу апроксимацію можна отримати, якщо допустити, що події $\{T_i = T_j\}$ і $\{S_i = S_j\}$ є рівноімовірними. В такому випадку отримаємо більш просту форму підрахунку ймовірності.

Лема 2.3. Апроксимоване значення ймовірності події $\bar{\varepsilon}$ для кожного набору з $k \ge 2$ аргументів $x_1...,x_k$ може бути отримано наступним чином:

$$\mathrm{P}[\bar{\varepsilon}] \le 1-(1-\frac{1}{2^n})^{2C_k^2} = 1-(1-\frac{1}{2^n})^{2^{\frac{k(k-1)}{2}}} = 1-(1-\frac{1}{2^n})^{k(k-1)}. \qquad (2.12)$$

В даному випадку розглядається $2C_k^2 = k(k-1)$ подій $\{T_i = T_j\}$ і $\{S_i = S_j\}$. Імовірність кожного, як і в лемі 2.1, прийнята за $P[S_i = S_j] = P[T_i = T_j] \le 2^{-n}$. Відмінність від формули (2.8) полягає в тому, що ймовірність знаходиться у відповідності з законом додавання ймовірностей сумісних подій [51]. У (2.8) відбувається просте підсумовування ймовірностей, що дає хорошу апроксимацію при малих кількостях аргументів x_i і при досить великій довжині блоку.

Узагальнюючи описані вище вираження можна записати формулу для уточненої оцінки ймовірності розрізнення:

$$|P[g(f(x_1),...,f(x_k))=1: f \in_R \psi(F_n,F_n,F_n)]-P_g| \le$$
$$\le 1-\prod_{i=0}^{k-2}(1-\frac{1}{2^n-i})^{2(k-(i+1))} \le 1-(1-\frac{1}{2^n})^{k(k-1)} \quad . \qquad (2.13)$$

Порівняльна оцінка значень переваги ланцюга Фейстеля над випадковою функцією отриманих за допомогою формул (2.5), (2.10) і (2.12) для блоків довжини 16 ($n=8$) і 32 ($n=16$) біта показана на рис. 2.6.

Як видно з графіка, формула (2.12) задає дуже хорошу апроксимацію для точного значення імовірності переваги, отриманого за допомогою формули (2.10). Слід зазначити, що межа ймовірності, отримана за допомогою формули (2.5) стає більше 1 приблизно при $\sqrt{2^n}$ аргументах, що унеможливлює її використання для оцінки відмінності ланцюга Фейстеля від випадкової функції при $|\{x_i\}|>\sqrt{2^n}$.

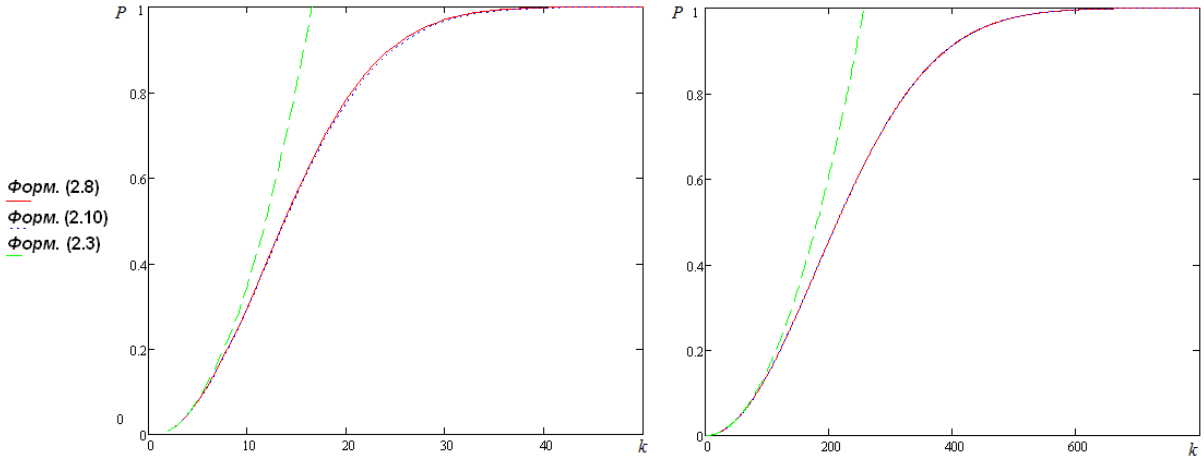


Рисунок 2.6 – Порівняльна характеристика ймовірностей переваги, отриманих за допомогою різних виразів для $n=8$ и $n=16$

У загальному випадку перевага ланцюга Фейстеля над випадковою функцією (перестановкою) визначається за формулою:

$$Advantage(\psi, F^*) = | P_\psi - P_{F^*} | . \qquad (2.14)$$

P_{ψ} – ймовірність розрізнення алгоритмом-розрізнювачем ланцюга Фейстеля;

P_{F*} – ймовірність розрізнення алгоритмом-розрізнювачем випадкової функції (перестановки).

Саме ймовірність P_{ψ} була оцінена в лемі 2.1, проте як було видно, ймовірність P_{F*} була прийнята рівною нулю. Ймовірно, автори не враховували дане значення в зв'язку з тим, що воно може змінюватися в залежності від алгоритму-розрізнювача і складно оцінити в такому випадку максимальне значення переваги, тому що потрібно вказати тоді найкращий алгоритм-розрізнювач.

Насправді $P_{F*} \neq 0$, тому максимальне значення переваги може бути виражено ще більш точно.

2.2.3 Практична оцінка ефективності розрізнення ланцюга Фейстеля

Оцінка ймовірності розрізнення, наведена в розділі 2.2, є теоретично максимально можливою вірогідністю розрізнення для ланцюга Фейстеля з кількістю раундів 3 і більше. Однак для реальних алгоритмів розрізнення дана ймовірність буде ще менше. Крім того, з ростом числа раундів ймовірність розрізнення також буде зменшуватися. В даному розділі будуть розглянуті конкретні алгоритми-розрізнювачі для ланцюга Фейстеля. Також, слід зазначити, що в подальшому буде проводитися розрізнення ланцюга Фейстеля від випадкової перестановки, а не функції, як було раніше. Це пов'язано з тим, що саме ланцюг Фейстеля є перестановкою, тому логічно порівнювати її також з перестановкою.

Варто відзначити, що високорівнева конструкція може мати кілька алгоритмів-розрізнювачів одночасно, кожен з яких дозволяє отримати певну ймовірність. Зокрема, для 3-раундового ланцюга Фейстеля (рис. 2.3) з випадковими функціями у якості раундових перетворень буде розглянуто два алгоритми-розрізнювачі. Цілком можливо, що існують ще які-небудь

варіанти алгоритмів-розрізнювачів, які дозволяють отримати ще більш високу ймовірність розрізнення.

2.2.3.1 Алгоритм-розрізнювач №1 для 3-раундового ланцюга Фейстеля з випадковими функціями у якості раундових перетворень

Алгоритм-розрізнювач g для k пар вхідних аргументів $\{x_i, x_j\}$ перевіряє виконання наступних двох рівностей: $L_i \oplus V_i = L_j \oplus V_j$ і $T_i = T_j$ (рис. 2.5). У разі виконання рівностей хоча б для однієї пари повертається значення буде «1», інакше «0». Такий алгоритм вже розглядався в [52].

Лема 2.4. Ймовірність виконання рівностей $L_i \oplus V_i = L_j \oplus V_j$ і $T_i = T_j$ для кожного набору з $k \geq 2$ аргументів $x_1 \ldots, x_k$ для 3-раундового ланцюга Фейстеля (рис. 2.5) дорівнює наступному значенню:

$$\mathrm{P}_1 = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^n - i}\right)^{k-(i+1)} \leq 1 - \left(1 - \frac{1}{2^n}\right)^{\frac{k(k-1)}{2}}. \qquad (2.15)$$

Доказ леми 2.4. Оскільки $R_i = R_j$ і $L_i \neq L_j$, то отже $S_i \neq S_j$. В такому випадку, обидві умови $L_i \oplus V_i = L_j \oplus V_j$ і $T_i = T_j$ будуть виконані в тому випадку, якщо $f_2^*(S_i) = f_2^*(S_j)$. Імовірність такої події для однієї пари буде дорівнювати $\frac{1}{2^n}$. Для k аргументів можна скласти C_k^2 пар, відповідно сумарна ймовірність виконання умови $f_2^*(S_i) = f_2^*(S_j)$ дорівнює $\mathrm{P} = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^n - i}\right)^{k-(i+1)}$. Спосіб обчислення такої сумарної ймовірності розглядався при доказі леми 2.2. Також у формулі (2.15) наведено апроксимувати значення ймовірності.

Доказ закінчено.

Лема 2.5. Ймовірність виконання умов $L_i \oplus V_i = L_j \oplus V_j$ і $T_i = T_j$ для випадкової перестановки дорівнює наступному значенню:

$$P_1^* = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^{2n} - 1 - i}\right)^{k-(i+1)} \le 1 - \left(1 - \frac{1}{2^{2n} - 1}\right)^{\frac{k(k-1)}{2}}. \qquad (2.16)$$

Доказ леми 2.5. Для випадкової перестановки вихідне значення $(V_i \bullet T_i)$ має абсолютно випадкове значення. Для однієї пари вхідних аргументів значення ймовірності має значення $\frac{1}{2^{2n} - 1}$. Це пов'язано з тим, що вихідні значення перестановки при різних вхідних аргументах не можуть повторюватися, тобто гарантовано $V_i \neq V_j$ і / або $T_i \neq T_j$. З огляду на, що $L_i \neq L_j$, така умова виключає один з невідповідних варіантів для виконання заданих рівностей.

Для k аргументів можна скласти C_k^2 пар, відповідно сумарна ймовірність виконання умов $L_i \oplus V_i = L_j \oplus V_j$ і $T_i = T_j$ дорівнює $P = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^{2n} - 1 - i}\right)^{k-(i+1)}$. Спосіб обчислення такої сумарної ймовірності розглядався при доказі леми 2.2. Також у формулі (2.16) наведено апроксимоване значення ймовірності.

Доказ закінчено.

Таким чином, маючи ймовірності розрізнення алгоритмом-розрізнювачем випадкової перестановки і перестановки побудованої за допомогою ланцюга Фейстеля, можна знайти перевагу алгоритму-розрізнювача:

$$\mathrm{Adv}(\psi, F^*) = |\mathrm{P}_1\text{-}\mathrm{P}_1^*| \le |\prod_{i=0}^{k-2}(1-\frac{1}{2^n-i})^{k-(i+1)} - \prod_{i=0}^{k-2}(1-\frac{1}{2^{2n}-1-i})^{k-(i+1)}| \le$$
$$\le |(1-\frac{1}{2^n})^{\frac{k(k-1)}{2}} - (1-\frac{1}{2^{2n}-1})^{\frac{k(k-1)}{2}}| \quad . \quad (2.17)$$

Розглянемо, яке значення переваги дозволяє досягти описаний алгоритм-розрізнювач для $n = 8$ і $n = 16$ (рис. 2.7). На наведених графіках по осі абсцис наведені значення переваги (тобто по суті це ймовірність розрізнення випадкової перестановки від ланцюга Фейстеля), по осі ординат наведено кількість аргументів, які подаються на вхід алгоритму-розрізнювача.

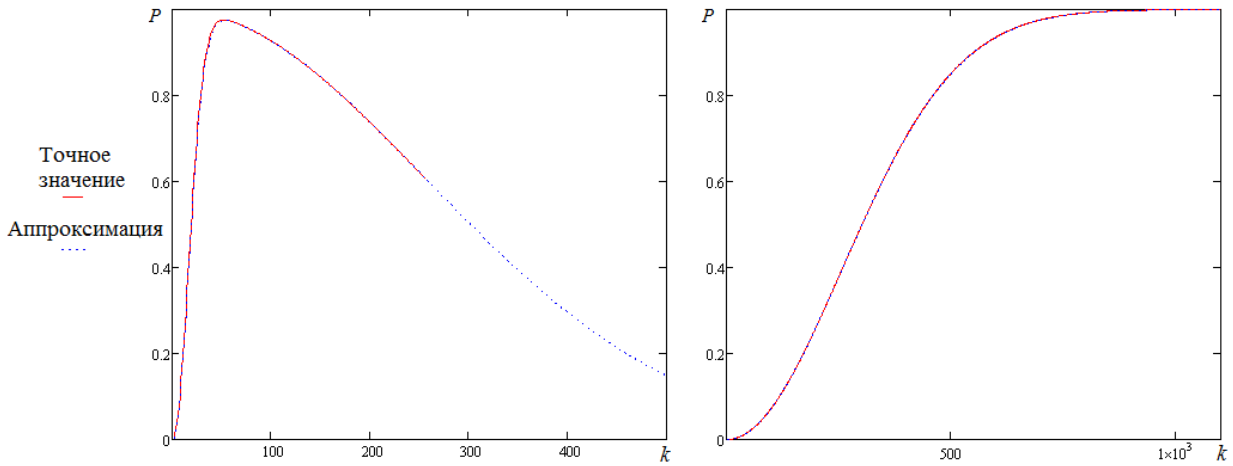


Рисунок 2.7 – Порівняльна характеристика ймовірностей переваги описаного алгоритму-розрізнювача, отриманих за допомогою різних виразів для $n = 8$ (лівий графік) та $n = 16$ (правий графік)

Слід зазначити, що правий графік (рис. 2.7) не прагне до одиниці як може здатися. При досить великій кількості аргументів графік піде на спадання і в кінцевому підсумку досягне нуля, як і лівий графік.

Як видно з цих графіків, за допомогою описаного алгоритму-розрізнювача можна практично з максимальною вірогідністю відрізнити ланцюг Фейстеля від випадкової функції при кількості вхідних аргументів порядку $\sqrt{2^n}$.

2.2.3.2 Алгоритм-розрізнювач №2 для 3-раундового ланцюга Фейстеля з випадковими функціями у якості раундових перетворень

Наведений вище алгоритм не є єдиним алгоритмом-розрізнювачем для 3-раундового ланцюга Фейстеля. У цьому розділі буде розглянуто ще один алгоритм, який також згадувався в [53].

Нехай алгоритм-розрізнювач g для k пар вхідних аргументів $\{x_i, x_j\}$ перевіряє виконання наступної рівності – $R_i \oplus T_i = R_j \oplus T_j$ (рис. 2.5). Вхідні аргументи вибираються такі, що $R_i \neq R_j$ і $L_i = L_j$. У разі виконання зазначеної рівності хоча б для однієї пари значення, що повертається, буде «1», інакше – «0».

Лема 2.6. Ймовірність виконання рівності $R_i \oplus T_i = R_j \oplus T_j$ для кожного набору з $k \geq 2$ аргументів $x_1 \ldots, x_k$ для 3-раундового ланцюга Фейстеля (рис. 2.5) дорівнює наступному значенню [47-50]:

$$\mathrm{P}_2 = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^n - i}\right)^{2(k-(i+1))} \leq 1 - \left(1 - \frac{1}{2^n}\right)^{k(k-1)}. \qquad (2.18)$$

Доказ леми 2.6. $R_i \oplus T_i = f_2(L_i \oplus f_1(R_i))$, відповідно $f_2(L_i \oplus f_1(R_i)) = f_2(L_j \oplus f_1(R_j))$. Розглядаються 2 незалежних події $f_2(L_i \oplus f_1(R_i)) = f_2(L_j \oplus f_1(R_j))$ і $L_i \oplus f_1(R_i) = L_j \oplus f_1(R_j)$, виконання будь-якого з них призведе до виконання заданої рівності. Імовірність такої події для однієї пари буде дорівнювати $\frac{2}{2^n}$. Для k аргументів кількість можливих подій $2C_2^m$, відповідно сумарна ймовірність виконання умови $f_2(L_i \oplus f_1(R_i)) = f_2(L_j \oplus f_1(R_j))$ дорівнює $\mathrm{P} = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^n - i}\right)^{2(k-(i+1))}$. Спосіб обчислення такої сумарної ймовірності розглядався при доказі леми 2.2. Також у формулі (2.18) наведено апроксимувати значення ймовірності.

Доказ закінчено.

Лема 2.7. Ймовірність виконання умови $R_i \oplus T_i = R_j \oplus T_j$ для випадкової перестановки дорівнює наступному значенню:

$$\mathrm{P}_2^* = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n}\right)^{k-(i+1)} \le 1 - \left(1 - \frac{2^n}{2^{2n} - 1}\right)^{\frac{k(k-1)}{2}}. \qquad (2.19)$$

Доказ леми 2.7. Для випадкової перестановки вихідне значення $(V_i \bullet T_i)$ має абсолютно випадкове значення. Для однієї пари вхідних аргументів значення ймовірності має значення $\frac{2^n}{2^{2n} - 1}$. Нехай *[R1, L1] => [V1, T1]*, отже *[R2, L2] ≠> [V1, T1]*, тобто безліч можливих значень для другого запиту зменшилася на один $N = 2^{2n} - 1$. При цьому $R_1 \oplus T_1 \neq R_2 \oplus T_1$, тобто по суті відкидається свідомо не задовольняють рівняння варіант. З усієї безлічі можливих варіантів $N = 2^{2n} - 1$ рівнянню задовольнятимуть 2^n варіантів, тому що шукаємо колізію на півблоці, а не на всьому блоці. Тобто ймовірність виконання рівності $R_i \oplus T_i = R_j \oplus T_j$ буде дорівнювати кількості виходів, що задовольняють, розділеному на загальну кількість можливих виходів для другого запиту $P = \frac{2^n}{2^{2n} - 1}$.

Для k аргументів можна скласти C_k^2 пар, відповідно сумарна ймовірність виконання умови $R_i \oplus T_i = R_j \oplus T_j$ дорівнює $\mathrm{P} = 1 - \prod_{i=0}^{k-2} \left(1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n}\right)^{k-(i+1)}$. Спосіб обчислення такої сумарної ймовірності розглядався при доказі леми 2.2. Також у формулі (2.19) наведено апроксимоване значення ймовірності.

Доказ закінчено.

Таким чином, маючи ймовірності розрізнення алгоритмом-розрізнювачем випадкової перестановки і перестановки побудованої за допомогою ланцюга Фейстеля, можна знайти перевагу алгоритму-розрізнювача:

$$\mathrm{Adv}(\psi, F^*) = |\mathrm{P}_2 - \mathrm{P}_2^*| \le$$
$$\le \left| \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^n - i}\right)^{2(k-(i+1))} - \prod_{i=0}^{k-2} \left(1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n}\right)^{k-(i+1)} \right| \le . \qquad (2.20)$$
$$\le \left| \left(1 - \frac{1}{2^n}\right)^{2\frac{k(k-1)}{2}} - \left(1 - \frac{2^n}{2^{2n} - 1}\right)^{\frac{k(k-1)}{2}} \right|$$

Розглянемо, яке значення переваги дозволяє досягти описаний алгоритм-розрізнювач для $n = 8$ і $n = 16$ (рис. 2.8). На наведених графіках по осі абсцис відзначені значення переваги (тобто по суті це ймовірність розрізнення випадкової перестановки від ланцюга Фейстеля), по осі ординат наведено кількість аргументів, які подаються на вхід алгоритму-розрізнювача.

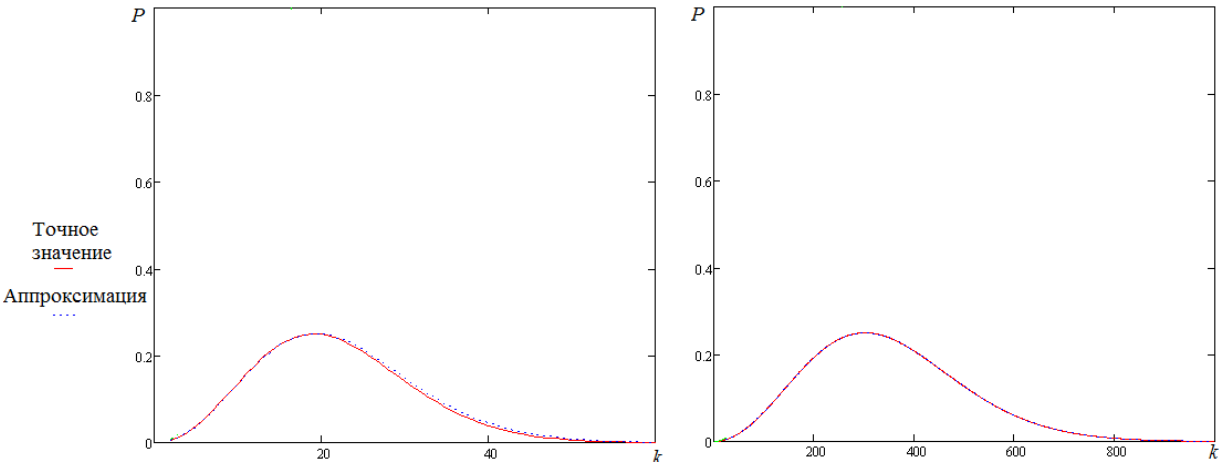


Рисунок 2.8 – Порівняльна характеристика ймовірностей переваги описаного алгоритму-розрізнювача, отриманих за допомогою різних виразів для $n = 8$ (лівий графік) и $n = 16$ (правий графік)

Як можна бачити на графіках на рис. 2.6, за допомогою описаного алгоритму-розрізнювача можна з імовірністю, близькою до $\frac{1}{3}$, відрізнити ланцюг Фейстеля від випадкової функції. Дана ймовірність розрізнення досягається при кількості вхідних аргументів порядку $\sqrt{2^n}$. Звідси можна зробити висновок, що розрізнювач №1, описаний в розділі 2.3.1 є більш ефективним, ніж розглянутий вище, оскільки дозволяє досягти більшої ймовірності розрізнення.

2.2.3.3 Алгоритм-розрізнювач для 3-раундового ланцюга Фейстеля з випадковими перестановками в якості раундових перетворень

Наведені вище алгоритми-розрізнювачі були орієнтовані на те, що в якості раундових функцій використовувалися випадкові функції. Якщо ж в якості раундових функцій використовувати випадкові перестановки (як, наприклад, в ГОСТ 28147), то можна досягти ще кращих результатів за розрізненням [54].

Нехай *f ([L, R]) = [V, T]*, де *f* – схема, наведена на рис. 2.3, f_i^* – це випадкові перестановки. Нехай $R_i = R_j$, а $L_i \neq L_j$, де (k – кількість запитів).

Алгоритм-розрізнювач перевіряє виконання умови $T_i \neq T_j$. Якщо вона виконується, то вважається, що *[V, T]* були отримані в результаті дії ланцюга Фейстеля, інакше – в результаті дії випадкової перестановки [54].

Лема 2.8. Для ланцюга Фейстеля ймовірність виконання події $T_i \neq T_j$ (тобто отримання «1» на виході алгоритму-розрізнювача) буде дорівнювати:

$$\mathrm{P}_1 = 1. \qquad (2.21)$$

Для випадкової перестановки ймовірність такої події буде дорівнює:

$$\mathrm{P}_1^{\,*} = \prod_{i=0}^{k-2} \prod_{j=0}^{k-i-2} \left(1 - \frac{2^n - 1}{2^{2n} - 1 - j - i \cdot 2^n}\right). \qquad (2.22)$$

Відповідно ймовірність розрізнення ланцюга Фейстеля і випадкової перестановки дорівнює різниці цих двох значень:

$$\mathrm{Adv} = |\, \mathrm{P}_1 - \mathrm{P}_1^{\,*} \,| = 1 - \prod_{i=0}^{k-2} \prod_{j=0}^{k-i-2} \left(1 - \frac{2^n - 1}{2^{2n} - 1 - j - i \cdot 2^n}\right). \qquad (2.23)$$

На рис. 2.9 наведені графіки ймовірностей розрізнення при різних кількостях вхідних аргументів для блоків різної довжини.

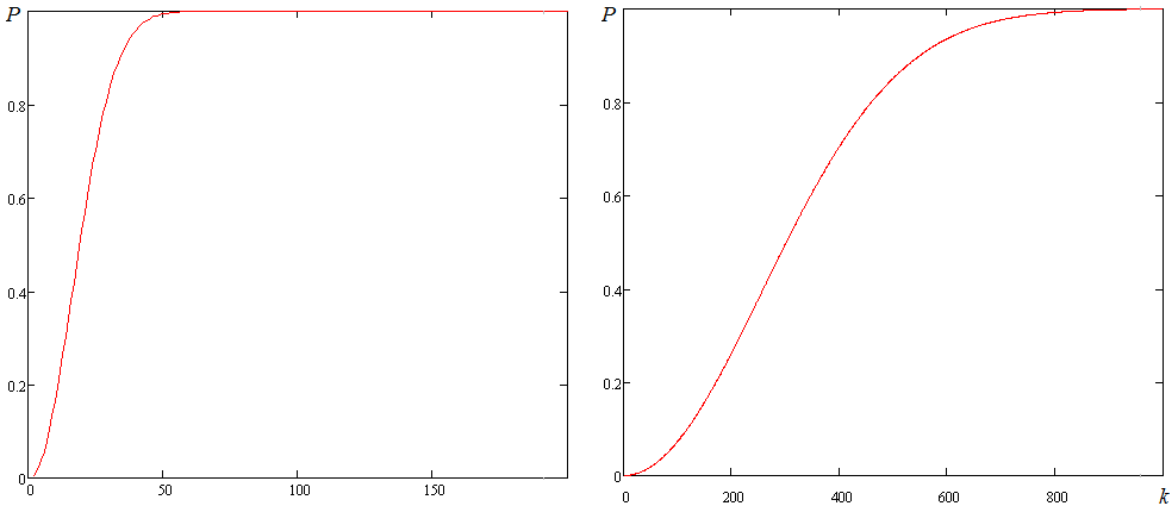


Рисунок 2.9 – Імовірність переваги для блоку $n = 8$ (зліва) и $n = 16$ (справа)

Доказ леми 2.8. Для 3-раундового ланцюга Фейстеля нерівність $T_i \neq T_j$ виконується з імовірністю $\mathrm{P}_1 = 1$. Тому що $R_i = R_j$, то отже $S_i \neq S_j$. З огляду на той факт, що f_2^* – перестановка, отримаємо $f_2^*(S_i) \neq f_2^*(S_j)$. Оскільки $R_i = R_j$, то $f_2^*(S_i) \oplus R_i \neq f_2^*(S_j) \oplus R_j$, тобто отримаємо $T_i \neq T_j$.

Для випадкової перестановки ймовірність виконання нерівності $T_i \neq T_j$, для будь-яких $0 \leq i < j < k$, вказана у формулі (2.22). Для того щоб отримати

цей вислів підемо від зворотного і знайдемо ймовірність події $T_i = T_j$, для будь-яких $0 \le i < j < k$.

Загальну ймовірність $P(T_i = T_j / 0 \le i < j < k)$ можна отримати як суму $P(T_0 = T_1) + ... + P(T_0 = T_{k-1}) + P(T_1 = T_2) + ... + P(T_1 = T_{k-1}) + ... + P(T_{k-2} = T_{k-1})$. Однак тут слід враховувати, що зазначені події є сумісними, тому ймовірність їх появи різна.

Розглянемо перші *m-1* пари запитів:

$[V_0;T_0],[V_1;T_1]$
$[V_0;T_0],[V_2;T_2]$
$[V_0;T_0],[V_3;T_3]$
…
$[V_0;T_0],[V_{k-1};T_{k-1}]$

Ймовірність $P[T_0 = T_i] = \dfrac{2^n - 1}{2^{2n} - i}$, $1 \le i \le k-1$.

$P(T_0 = T_1) = \dfrac{2^n - 1}{2^{2n} - 1}$, оскільки T_i – випадкова величина. Однак точно відомо, що $[V_0;T_0] \ne [V_1;T_1]$ (оскільки це перестановка), тому безліч всіх можливих значень для запиту $[V_1;T_1]$ (знаменник) скорочується на одиницю, як і підмножина задовольняють значень (чисельник – кількість значень $[V_1;T_1]$, при яких $T_0 = T_1$). Відзначимо, що це справедливо для всіх можливих пар $[V_iT_i;V_jT_j]$.

Для T_0 і T_2 маємо ймовірність $P(T_0 = T_2) = \dfrac{2^n - 1}{2^{2n} - 2}$. Оскільки точно відомо, що $[V_1;T_1] \ne [V_2;T_2]$, то безліч всіх можливих значень (знаменник) зменшилася ще на одиницю, але підмножина задовольняють значень (чисельник) не зменшилася, оскільки $V_0 \ne V_1$. Це справедливо для всіх пар $[V_0T_0;V_iT_i]$, де $0 < i < k$. Також дані міркування справедливі і для наступних наборів пар.

Розглянемо наступні *k-2* пари запитів:

$$\left.\begin{array}{l}[V_1;T_1],[V_2;T_2]\\ [V_1;T_1],[V_3;T_3]\\ \\ [V_1;T_1],[V_4;T_4]\\ \\ \ldots\\ [V_1;T_1],[V_{k-1};T_{k-1}]\end{array}\right\} \quad P[T_1 = T_i] \le \frac{2^n - 1}{2^{2n} - i - 2^n},\ 1 \le i \le k-2.$$

При обробці даної множини пар нам відомо, що $T_i \neq T_0,\ i = 1..k-1$, що знову таки скоротить загальну безліч можливих значень для $[V_i;T_i]$, $i = 2...k-1$, оскільки всього є 2^n можливих запитів, в яких $T_i = T_0$ (при всіх можливих $V_j = 0...2^n - 1$). Тобто загальна безліч можливих значень (знаменник) скоротиться на 2^n значень. Ці міркування є закономірними і для всіх наступних наборів запитів.

Аналогічно можна порахувати ймовірності для інших наборів пар. Всього таких наборів може бути *k-1*. Кожен буде мати на одну пару менше. Останній, що складається з однієї пари буде мати ймовірність:

$$P[V_{k-2} = V_{k-1}] = \frac{2^n - 1}{2^{2n} - 1 - (k-2)2^n}. \qquad (2.24)$$

Всього розглянутих подій буде C_2^k. Сумарну ймовірність цих подій можна знайти, як зазначено в наступній формулі, відповідно до законів додавання ймовірностей сумісних подій:

$$P[T_i = T_j] = 1 - (1 - \frac{2^n - 1}{2^{2n} - 1})(1 - \frac{2^n - 1}{2^{2n} - 2})...(1 - \frac{2^n - 1}{2^{2n} - (k-1)})(1 - \frac{2^n - 1}{2^{2n} - 1 - 2^n}) *$$

$$* (1 - \frac{2^n - 1}{2^{2n} - 2 - 2^n}) * ... * (1 - \frac{2^n - 1}{2^{2n} - (k-2) - 2^n})(1 - \frac{2^n - 1}{2^{2n} - 1 - 2 \cdot 2^n})...$$

$$...(1 - \frac{2^n - 1}{2^{2n} - (k-3) - 2 \cdot 2^n})...(1 - \frac{2^n - 1}{2^{2n} - 1 - (k-2) \cdot 2^n}) = \prod_{i=0}^{k-2} \prod_{j=0}^{k-i-2} (1 - \frac{2^n - 1}{2^{2n} - 1 - j - i \cdot 2^n}).$$

Відповідно перевага 3-раундового ланцюга Фейстеля (з випадковими перестановками в якості раундових функцій) над випадковою перестановкою становить різницю розглянутих двох ймовірностей (ймовірність розрізнення ланцюга Фейстеля і випадкової перестановки розглянутим алгоритмом) і виражено у формулі (2.23).

Доказ закінчено.

Оскільки представлена формула для підрахунку переваги є досить громіздкою, пропонується також вираз, що дає гарне апроксимаційне значення з мінімальною похибкою:

$$\mathrm{Adv\_approx} = |\, \mathrm{P}_1 - \mathrm{P}_1^{*} \,| = 1 - (1 - \frac{1}{2^n})^{\frac{k(k-1)}{2}} . \qquad (2.25)$$

Розглянемо, яке значення переваги дозволяє досягти описаний алгоритм-розрізнювач для $n = 8$ і $n = 16$ (рис. 2.10). На наведених графіках по осі абсцис відзначені значення переваги (тобто по суті це ймовірність розрізнення випадкової перестановки від ланцюга Фейстеля), по осі ординат наведено кількість аргументів, які подаються на вхід алгоритму-розрізнювача.

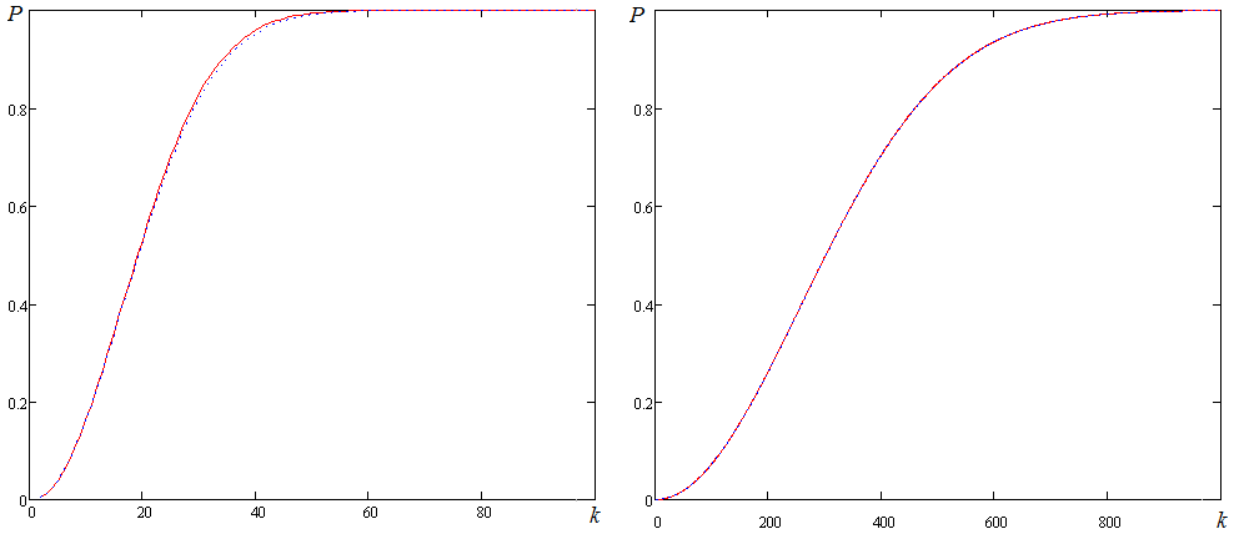


Рисунок 2.10 – Порівняльна характеристика ймовірностей переваги, отриманих за допомогою формул (2.23) і (2.25) (пунктиром) для *n=8* і *n=16*

Як видно з цих графіків, за допомогою описаного алгоритму-розрізнювача можна практично з максимальною вірогідністю відрізнити ланцюг Фейстеля від випадкової функції при кількості вхідних аргументів порядку $\sqrt{2^n}$.

2.2.3.4 Алгоритм-розрізнювач для 4-раундового ланцюга Фейстеля з випадковими функціями у якості раундових перетворень

Для 4-раундового ланцюга Фейстеля (рис. 2.11) буде розглянутий алгоритм-розрізнювач, який аналогічний розрізнювачу №2 для 3-раундового ланцюга Фейстеля. Тому докази формул приводитися не будуть.

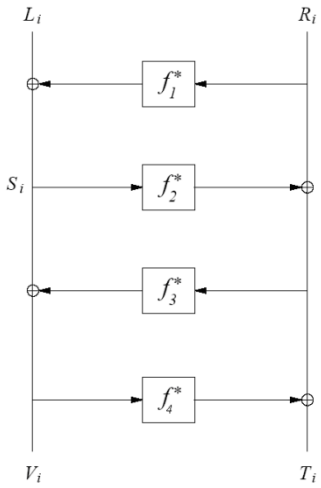


Рисунок 2.11 – 4-раундовий ланцюг Фейстеля

Нехай алгоритм-розрізнювач g для k пар вхідних аргументів $\{x_i, x_j\}$ перевіряє виконання наступної рівності – $L_i \oplus V_i = L_j \oplus V_j$ (рис. 2.9). Вхідні аргументи вибираються такі, що $R_i = R_j$ і $L_i \neq L_j$. У разі виконання зазначеного рівності хоча б для однієї пари повертається значення «1», інакше – «0».

Лема 2.9. Ймовірність виконання рівності $L_i \oplus V_i = L_j \oplus V_j$ для кожного набору з $k \geq 2$ аргументів для 4-раундового ланцюга Фейстеля (рис. 2.11) дорівнює наступному значенню:

$$\mathrm{P}_2 = 1 - \prod_{i=0}^{k-2} (1 - \frac{1}{2^n - i})^{2(k-(i+1))} \leq 1 - (1 - \frac{1}{2^n})^{k(k-1)}. \qquad (2.26)$$

Лема 2.10. Ймовірність виконання умови $L_i \oplus V_i = L_j \oplus V_j$ для випадкової перестановки дорівнює наступному значенню:

$$\mathrm{P}_2^* = 1 - \prod_{i=0}^{k-2} (1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n})^{k-(i+1)} \leq 1 - (1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}}. \qquad (2.27)$$

Таким чином, маючи ймовірності розрізнення алгоритмом-розрізнювачем випадкової перестановки і перестановки побудованої за допомогою ланцюга Фейстеля, можна знайти перевагу алгоритму-розрізнювача:

$$\mathrm{Adv}(\psi, F^*) = |\mathrm{P}_2 - \mathrm{P}_2^*| \leq$$
$$\leq |\prod_{i=0}^{k-2} (1 - \frac{1}{2^n - i})^{2(k-(i+1))} - \prod_{i=0}^{k-2} (1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n})^{k-(i+1)}| \leq . \qquad (2.26)$$
$$\leq |(1 - \frac{1}{2^n})^{2^{\frac{k(k-1)}{2}}} - (1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}}|$$

Розглянемо значення переваги, яке дозволяє досягти описаний алгоритм-розрізнювач для $n = 8$ і $n = 16$ (рис. 2.12). На наведених графіках по осі абсцис відзначені значення переваги (тобто по суті це ймовірність розрізнення випадкової перестановки від ланцюга Фейстеля), по осі ординат наведено кількість аргументів, які подаються на вхід алгоритму-розрізнювача.

Як видно з цих графіків, за допомогою описаного алгоритму-розрізнювача можна з імовірністю, близькою до $\frac{1}{3}$, відрізнити ланцюг Фейстеля від випадкової функції при кількості вхідних аргументів порядку $\sqrt{2^n}$.

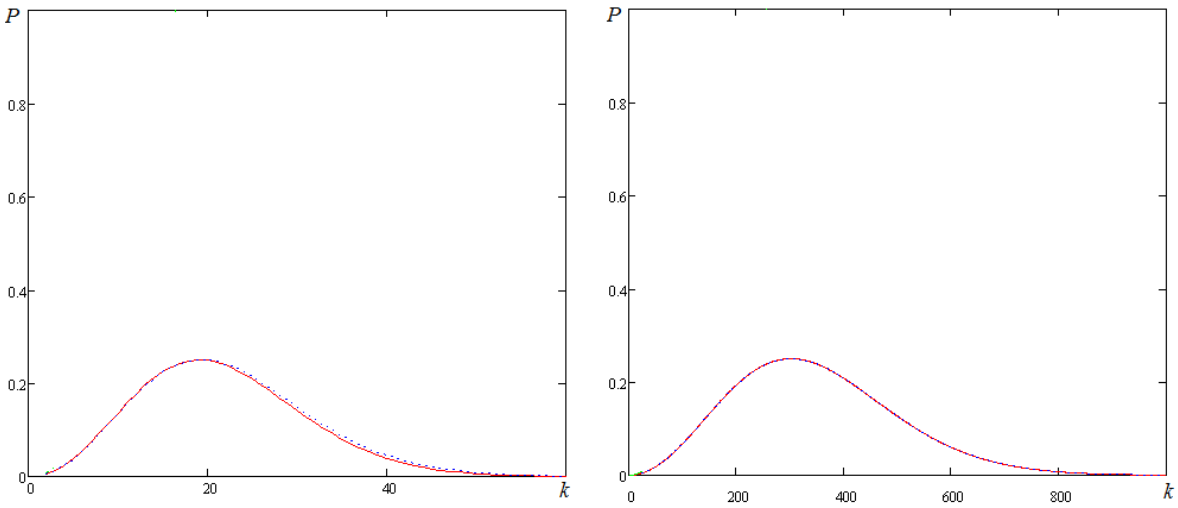


Рисунок 2.12 – Порівняльна характеристика ймовірностей переваги описаного алгоритму-розрізнювача для $n=8$ и $n=16$

Таким чином, на підставі вищенаведеного можна зробити наступні висновки:

1. Для оцінки ефективності ланцюга Фейстеля потрібно визначити ймовірність його відрізнення від випадкової функції. Для цього слід використовувати спеціальний алгоритм, який, приймаючи на вхід пари відкритих і закритих текстів, з певною ймовірністю може визначити за допомогою якої структури дані пари були отримані. Чим більшу ймовірність дає такий алгоритм, тим менш ефективною є високорівнева конструкція, оскільки вона погано приховує свою внутрішню структуру.

2. На даний момент існує вираз для оцінки максимально можливої ймовірності розрізнення ланцюга Фейстеля і випадкової функції [55]. В ході досліджень вдалося вдосконалити цей вислів і отримати уточнену оцінку максимальної ймовірності розрізнення.

3. Для 2-раундового ланцюга Фейстеля побудова алгоритму розрізнювача є тривіальним завданням. Для 3- і 4-раундових ланцюгів Фейстеля були знайдені алгоритми-розрізнювачі, які дозволяють отримати досить велику ймовірність розрізнення.

2.3 Оцінка ефективності розрізнення БСШ на основі схеми Лей-Мессі

У 1990 р вийшла перша версія шифру IDEA, в якому була реалізована структура, відмінна від популярних на той момент ланцюга Фейстеля і SP-мережі (хоча слід врахувати, що ланцюг Фейстеля також використовувався в цьому шифрі) [56]. Дана схема отримала назву за іменами її творців Сюецзя Лайа і Джеймса Мессі – схема Лей-Мессі.

Даний розділ присвячений аналізу властивостей схеми Лей-Мессі, представлені нові результати, отримані в ході досліджень.

2.3.1 Загальна схема БСШ на основі схеми Лей-Мессі

Як і ланцюг Фейстеля, схема Лей-Мессі є багаторазово повторюваною ітераційною структурою. На кожній ітерації виконується раундове перетворення під дією раундового ключа. Операції шифрування і розшифрування можна описати за допомогою таких висловів [26]:

$$L_i = \sigma(L_{i-1} \oplus F(L_{i-1} \oplus R_{i-1}, K_{i-1})), \tag{2.27}$$

$$R_i = R_{i-1} \oplus F(L_{i-1} \oplus R_{i-1}, K_{i-1}), \text{ при } i \in 1..n-1. \tag{2.28}$$

На останньому раунді функція σ опускається, оскільки вона є лінійною і не вносить ніякої складності на останньому етапі перетворення:

$$L_n = L_{n-1} \oplus F(L_{n-1} \oplus R_{n-1}, K_{n-1}), \tag{2.29}$$

$$R_n = R_{n-1} \oplus F(L_{n-1} \oplus R_{n-1}, K_{n-1}). \tag{2.30}$$

В даному випадку R_i і L_i – це права і ліва частини повідомлення на i-му раунді (відповідно R_0 і L_0 – це вихідне повідомлення, а R_n й L_n, де n – кількість раундів). Функція σ – це деяке ортоморфне перетворення. Функція F – це функція ускладнення, залежна від ключа. Загальна схема приведена на рис. 2.13.

Далі в цьому розділі передбачається, що функція σ – це ланцюг Фейстеля (рис. 2.14), оскільки в багатьох шифрах, побудованих за схемою Лей-Массі (IDEA [56], FOX [27], Мухомор [28]), використовується саме така конструкція.

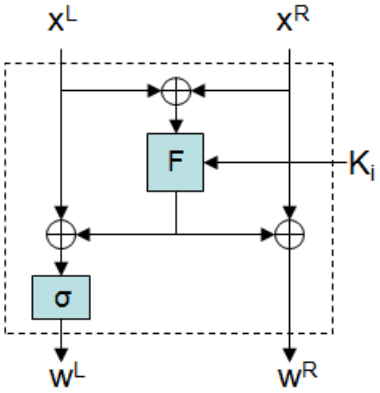


Рисунок 2.13 – Схема Лей-Массі

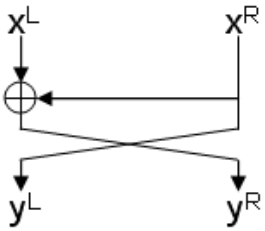


Рисунок 2.14 – Функція σ

Операції шифрування і розшифрування на кожному етапі дуже прості, і при певній доробці збігаються, вимагаючи тільки зворотного порядку використовуваних ключів. Шифрування за допомогою даної конструкції легко реалізується як на програмному рівні, так і на апаратному, що забезпечує широкі можливості застосування.

2.3.2 Теоретична оцінка ефективності розрізнення схеми Лей-Мессі

В даному розділі буде розглянута максимальна ймовірність розрізнення випадкової функції і схеми Лей-Мессі з кількістю раундів $r \geq 3$. Доказ буде приводитися для 3-раундової схеми Лей-Мессі, проте вираз буде справедливо і для конструкцій з великою кількістю раундів, оскільки складність розрізнення в такому випадку буде тільки рости.

Розглянемо загальну схему побудови 3-раундової схеми Лей-Мессі (рис. 2.15).

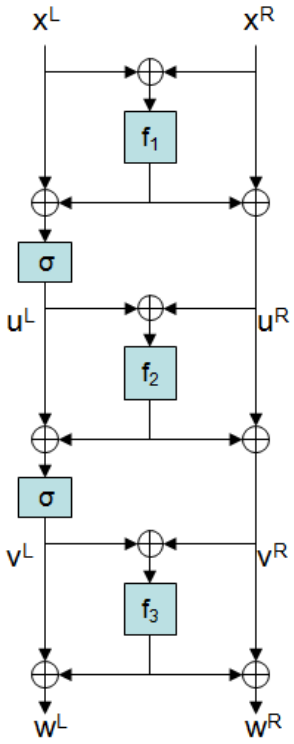


Рисунок 2.15 – Схема Лей-Мессі

x^L, x^R – ліва і права половини вхідного блоку даних, кожна по n біт. Відповідно загальна довжина вхідного блоку становить $2n$ біт;

$\Delta x = x^L \oplus x^R$ – XOR-різниця між лівою і правою половинами бітового вектора;

u^L, u^R – ліва і права половини проміжного результату після 1-го раунду перетворень;

v^L, v^R – ліва і права половини проміжного результату після 2-го раунду перетворень;

w^L, w^R – ліва і права половини вихідного значення;

f_1, f_2, f_3 – випадкові функції;

σ – ланцюг Фейстеля (рис. 2.14);

$\sigma^2(x) = \sigma(\sigma(x))$.

Відзначимо, що даний алгоритм-розрізнювач матиме сенс для будь-якої лінійної функції σ (тобто такий, що $\sigma(x+y) = \sigma(x) + \sigma(y)$). У нашому випадку розглядається саме ланцюг Фейстеля, оскільки така схема використовується найбільш часто (шифри FOX, Мухомор). Введемо також функцію $\sigma'(x) = \sigma(x) + x$. Якщо σ – лінійна, то σ' теж лінійна.

Можна показати, що якщо $\Delta U_i \neq \Delta U_j$ для всіх пар запитів, то побудова алгоритму-розрізнювача неможлива, відповідно звідси можна отримати теоретично максимальну ймовірність розрізнення для схеми Лей-Мессі з кількістю раундів три і більше. Висунемо і доведемо наступну лему:

Лема 2.11. Максимальна ймовірність розрізнення схеми Лей-Мессі (рис. 2.15), з кількістю раундів $r \geq 3$, і випадкової функції при k запитах дорівнює наступному значенню:

$$Adv_{\max}(LM, PRF) \leq 1 - \left(\frac{2^n - 2}{2^n - 1}\right)^{\frac{k(k-1)}{2}}. \qquad (2.31)$$

Доказ леми 2.11. Доведемо спочатку, що при $\Delta U_i \neq \Delta U_j$ (рис. 2.15) розрізнення неможливо.

Отже, нехай $\Delta U_i \neq \Delta U_j$ для всіх можливих пар запитів. Тоді маємо наступне:

$$W_i^L = V_i^L \oplus f_3(\Delta V_i) = \sigma^2(x_i^L \oplus f_1(\Delta x_i)) \oplus \sigma(f_2(\Delta U_i)) \oplus f_3(\Delta V_i), \quad (2.32)$$

$$W_i^R = V_i^R \oplus f_3(\Delta V_i) = x_i^R \oplus f_1(\Delta x_i) \oplus f_2(\Delta U_i) \oplus f_3(\Delta V_i), \quad (2.33)$$

$$W_j^L = V_j^L \oplus f_3(\Delta V_j) = \sigma^2(x_j^L \oplus f_1(\Delta x_j)) \oplus \sigma(f_2(\Delta U_j)) \oplus f_3(\Delta V_j), \quad (2.34)$$

$$W_j^R = V_j^R \oplus f_3(\Delta V_j) = x_j^R \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_j) \oplus f_3(\Delta V_j). \quad (2.35)$$

Оскільки f_2 – випадкова функція (перестановка), то $f_2(\Delta U_i)$ і $f_2(\Delta U_j)$ абсолютно випадкові значення і знайти якусь кореляцію між ними неможливо. Отже, вихідні значення також будуть випадковими.

Розглянемо також значення ΔW_i та ΔW_i:

$$\Delta W_i = W_i^L \oplus W_i^R = \sigma^2(x_i^L \oplus f_1(\Delta x_i)) \oplus \sigma(f_2(\Delta U_i)) \oplus x_i^R \oplus f_1(\Delta x_i) \oplus \\ \oplus f_2(\Delta U_i) = \sigma^2(x_i^L \oplus f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i) \oplus \sigma'(f_2(\Delta U_i)) \quad (2.36)$$

$$\Delta W_j = W_j^L \oplus W_j^R = \sigma^2(x_j^L \oplus f_1(\Delta x_j)) \oplus \sigma(f_2(\Delta U_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus \\ \oplus f_2(\Delta U_j) = \sigma^2(x_j^L \oplus f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus \sigma'(f_2(\Delta U_j)) \quad (2.37)$$

Оскільки f_2 – випадкова функція (перестановка), то $\sigma'(f_2(\Delta U_i))$ і $\sigma'(f_2(\Delta U_j))$ також абсолютно випадкові значення і знайти якусь кореляцію між ними неможливо. Відповідно ΔW_i і ΔW_j в такому випадку теж випадкові значення.

Тепер знайдемо ймовірність виконання $\Delta U_i = \Delta U_j$. Запишемо детально дану рівність:

$$\begin{gathered}
\sigma(x_i^L \oplus f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i) = \sigma(x_j^L \oplus f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j), \\
\sigma(x_i^L) \oplus \sigma(f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i) = \sigma(x_j^L) \oplus \sigma(f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j), \\
\sigma(f_1(\Delta x_i)) \oplus \sigma(f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j) = \sigma(x_j^L) \oplus \sigma(x_i^L) \oplus x_j^R \oplus x_i^R, \\
\sigma'(f_1(\Delta x_i)) \oplus \sigma(f_1(\Delta x_j)) = \sigma(x_j^L) \oplus \sigma(x_i^L) \oplus x_j^R \oplus x_i^R.
\end{gathered} \quad (2.38)$$

Якщо f_1 – випадкова функція, то максимальна ймовірність виконання (2.38) (при підібраних відкритих текстах) має наступне значення:

$$P_1 \le \frac{1}{2^n}. \tag{2.39}$$

Якщо ж f_1 – випадкова перестановка, то максимальна ймовірність виконання (2.38) (при підібраних відкритих текстах) має наступне значення:

$$P_2 \le \frac{1}{2^n - 1}. \tag{2.40}$$

Для перестановки (формула 3.14) ймовірність збільшилася тому, що завжди виконується нерівність $\sigma^{/}(f_1(\Delta x_i)) \oplus \sigma(f_1(\Delta x_j)) \neq 0$, тоді можемо підібрати відповідні відкриті тексти.

Оскільки визначаємо верхню межу розрізнення, то візьмемо велику ймовірність P_2 у якості ймовірності $P(\Delta U_i = \Delta U_j)$. У такому випадку ймовірність того, що для двох запитів буде виконуватися $\Delta U_i \neq \Delta U_j$ має наступне значення:

$$P(\Delta U_i \neq \Delta U_j) = 1 - P_2 \le 1 - \frac{1}{2^n - 1} = \frac{2^n - 2}{2^n - 1}. \tag{2.41}$$

Для k запитів ймовірність того, що для кожної пари виконується нерівність $\Delta U_i \neq \Delta U_j$, матиме таке значення:

$$P(\Delta U_i \neq \Delta U_j) \le \left(\frac{2^n - 2}{2^n - 1}\right)^{\frac{k(k-1)}{2}}. \tag{2.42}$$

Отже, ймовірність розрізнення схеми Лей-Мессі не перевищує величини зворотної (2.42).

Максимальна перевага тоді матиме таке значення:

$$Adv_{\max}(LM, PRF) = | P_{\max}(LM) - P_{\max}(PRF) | . \qquad (2.43)$$

Припустимо, що для випадкової функції ймовірність появи «1» на виході алгоритму-розрізнювача дорівнює нулю (тому що не розглядаємо конкретний алгоритм-розрізнювач). Тоді перевага дорівнюватиме наступному значенню:

$$Adv_{\max}(LM, PRF) = | P_{\max}(LM) - P_{\max}(PRF) | \le 1 - (\frac{2^n - 2}{2^n - 1})^{\frac{k(k-1)}{2}} . \qquad (2.44)$$

Приходимо до (2.41).

Доказ леми 2.11 закінчено.

На рис. 3.4 наведені теоретично максимальні ймовірності розрізнення ланцюга Фейстеля (пунктиром) і схеми Лей-Мессі. Максимальна ймовірність розрізнення для ланцюга Фейстеля розглядалася в розділі 2.2 і, зокрема, графік був побудований за допомогою формули (2.11). Для схеми Лей-Мессі (суцільна лінія) використовувалася формула (3.18).

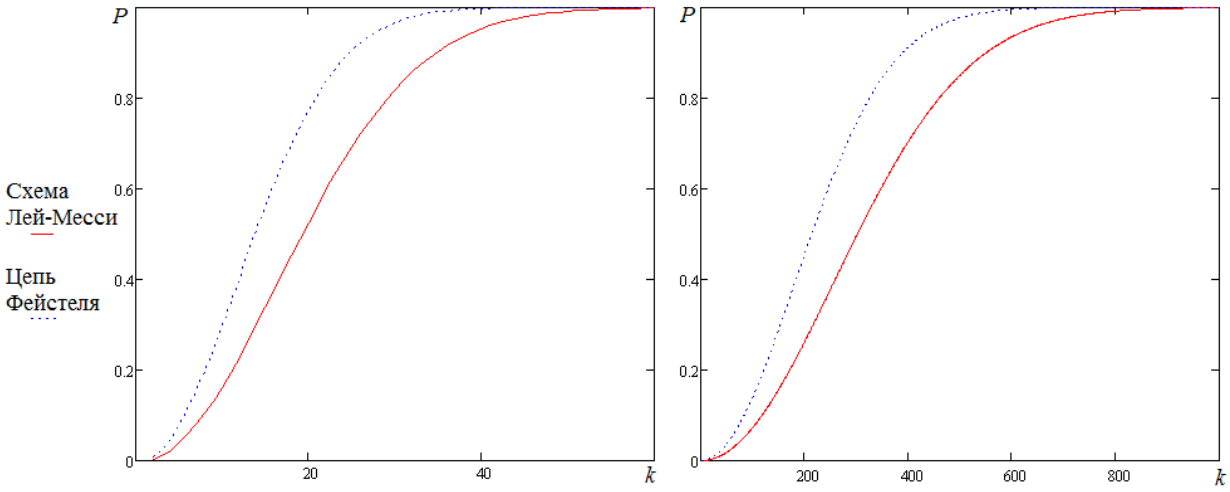


Рисунок 2.16 – Максимальні ймовірності розрізнення для ланцюга Фейстеля і схеми Лей-Мессі для блоків $n=8$ (зліва) та $n=16$ (справа)

Як видно ймовірність розрізнення схеми Лей-Мессі менше при однаковій кількості запитів, що може свідчити про те, що схема Лей-Мессі більш ефективна, ніж ланцюг Фейстеля.

Розглянута в даному розділі ймовірність є лише теоретичною верхньою межею розрізнення. Це означає, що реальна максимальна ймовірність розрізнення може бути і менше. У наступних розділах будуть розглянуті конкретні алгоритми-розрізнювачі для 3-раундової схеми Лей-Мессі.

2.3.3 Практична оцінка ефективності розрізнення схеми Лей-Мессі

Оцінка ймовірності розрізнення, наведена в розділі 2.3.2, є теоретично максимально можливою вірогідністю розрізнення для схеми Лей-Мессі з кількістю раундів 3 і більше. Для розглянутих далі реальних алгоритмів розрізнення дана ймовірність буде ще менше. В даному розділі будуть розглянуті конкретні алгоритми-розрізнювачі для схеми Лей-Мессі.

2.3.3.1 Алгоритм-розрізнювач для 3-раундової схеми Лей-Мессі з випадковими функціями у якості раундових перетворень

Алгоритм-розрізнювач g для k пар вхідних аргументів $\{x_i, x_j\}$ перевіряє виконання наступної рівності: $\Delta W_i \oplus \Delta W_j = \sigma(x_i^R \oplus x_j^R) \oplus x_i^R \oplus x_j^R$ (рис. 3.3). У разі виконання рівності хоча б для однієї пари повертається «1», інакше – «0».

Лемма 2.12. Нехай ми маємо k запитів (x_i, w_i), причому для кожної можливої пари вхідних аргументів (x_i, x_j) виконується така умова: $\sigma(x_i^L \oplus x_j^L) = x_i^R \oplus x_j^R$ і $\Delta x_i \neq \Delta x_j$. Алгоритм-розрізнювач заснований на перевірці виконання рівності (2.45) для всіх можливих пар $[x_i, w_i];[x_j, w_j]$, де $0 \le i < j < k$. Якщо хоча б для однієї пари дана рівність виконується, то на виході алгоритму-розрізнювача маємо «1».

$$\Delta W_i \oplus \Delta W_j = \sigma(x_i^R \oplus x_j^R) \oplus x_i^R \oplus x_j^R. \qquad (2.45)$$

Тоді ймовірність розрізнення 3-раундової схеми Лей-Мессі і випадкової перестановки має наступне значення:

$$Adv(LM, PRP) \le | (1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}} - (1 - \frac{2^{n+1} - 1}{2^{2n}})^{\frac{k(k-1)}{2}} |. \qquad (2.46)$$

Доказ лемми 2.12. Розглянемо вірогідність виконання (2.45) для схеми Лей-Мессі при $k = 2$. У такому випадку ймовірність буде дорівнювати наступному значенню:

$$P(LM) = 1 - (1 - \frac{1}{2^n})^2 = \frac{2^{n+1} - 1}{2^{2n}}. \qquad (2.47)$$

Розглянемо докладніше формулу (3.21).

Запишемо повні вирази для ΔW_i та ΔW_j:

$$\begin{gathered}\Delta W_i = W_i^L \oplus W_i^R = V_i^L \oplus V_i^R = \\ = \sigma(\sigma(x_i^L \oplus f_1(\Delta x_i)) \oplus f_2(\Delta U_i)) \oplus x_i^R \oplus f_1(\Delta x_i) \oplus f_2(\Delta U_i)\end{gathered}, \tag{2.48}$$

$$\begin{gathered}\Delta W_j = W_j^L \oplus W_j^R = V_j^L \oplus V_j^R = \\ = \sigma(\sigma(x_j^L \oplus f_1(\Delta x_j)) \oplus f_2(\Delta U_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_j)\end{gathered}. \tag{2.49}$$

З огляду на те, що функція $\sigma-$ лінійна, можна записати вирази (2.48) і (2.49) наступним чином:

$$\begin{gathered}\Delta W_i = W_i^L \oplus W_i^R = V_i^L \oplus V_i^R = \\ = \sigma^2(x_i^L) \oplus \sigma^2(f_1(\Delta x_i)) \oplus \sigma(f_2(\Delta U_i)) \oplus x_i^R \oplus f_1(\Delta x_i) \oplus f_2(\Delta U_i),\end{gathered} \tag{2.50}$$

$$\begin{gathered}\Delta W_j = W_j^L \oplus W_j^R = V_j^L \oplus V_j^R = \\ = \sigma^2(x_j^L) \oplus \sigma^2(f_1(\Delta x_j)) \oplus \sigma(f_2(\Delta U_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_j).\end{gathered} \tag{2.51}$$

Таким чином, рівність (2.45) буде виконуватися в двох випадках:

1. У разі виникнення колізії на функції f_1: якщо сталася колізія і $f_1(\Delta x_i) = f_1(\Delta x_j)$, то також виконується умова $\Delta U_i = \Delta U_j$, а відповідно і $f_2(\Delta U_i) = f_2(\Delta U_j)$. Розглянемо рівність $\Delta U_i = \Delta U_j$:

$$\Delta U_i = U_i^L \oplus U_i^R = \sigma(x_i^L \oplus f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i), \tag{2.52}$$

$$\Delta U_j = U_j^L \oplus U_j^R = \sigma(x_j^L \oplus f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j). \tag{2.53}$$

Відповідно, якщо $\Delta U_i = \Delta U_j$, то маємо такий вираз:

$$\begin{gathered}\sigma(x_i^L) \oplus \sigma(f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i) = \\ = \sigma(x_j^L) \oplus \sigma(f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j)\end{gathered}. \tag{2.54}$$

Враховуючи, що $f_1(\Delta x_i) = f_1(\Delta x_j)$, отримаємо:

$$\sigma(x_i^L) \oplus x_i^R = \sigma(x_j^L) \oplus x_j^R. \quad (2.55)$$

Як видно, це відповідає вхідній умові, за якою відбираються пари відкритих текстів, тому при виникненні колізії на f_1 ми маємо також $f_2(\Delta U_i) = f_2(\Delta U_j)$.

В такому випадку різниця виразів (2.50) і (2.51) приймає такий вигляд:

$$\begin{aligned} \Delta W_i \oplus \Delta W_j = \sigma^2(x_i^L) \oplus x_i^R \oplus \sigma^2(x_j^L) \oplus x_j^R = \\ = \sigma^2(x_i^L \oplus x_j^L) \oplus x_i^R \oplus x_j^R \end{aligned}. \quad (2.56)$$

Тобто прийшли до вираження (2.45). Це означає, що при колізії на функції f_1 вираз (2.45) дійсно виконується.

f_1 – випадкова функція і $\Delta x_i \neq \Delta x_j$ за умовою, тому маємо наступну ймовірність колізії:

$$P(f_1(\Delta x_i) = f_1(\Delta x_j)) = \frac{1}{2^n}. \quad (2.57)$$

2. У разі відсутності колізії на функції f_1 і виконання наступної рівності:

$$\begin{aligned} \sigma^2(f_1(\Delta x_i) \oplus f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j) = \\ = \sigma'(f_2(\Delta U_i) \oplus f_2(\Delta U_j) \end{aligned}. \quad (2.58)$$

Запишемо повне рівняння для $\Delta W_i \oplus \Delta W_j$:

$$\begin{gathered}\Delta W_i \oplus \Delta W_j = \sigma^2(x_i^L) \oplus \sigma^2(f_1(\Delta x_i)) \oplus \\ \oplus\sigma(f_2(\Delta U_i)) \oplus x_i^R \oplus f_1(\Delta x_i) \oplus f_2(\Delta U_i) \oplus \\ \oplus\sigma^2(x_j^L) \oplus \sigma^2(f_1(\Delta x_j)) \oplus \sigma(f_2(\Delta U_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_j) = \\ = \sigma^2(x_i^L \oplus x_j^L) \oplus x_i^R \oplus x_j^R \oplus \sigma^2(f_1(\Delta x_i) \oplus f_1(\Delta x_j)) \oplus \\ \oplus\sigma(f_2(\Delta U_i) \oplus f_2(\Delta U_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_i) \oplus f_2(\Delta U_j).\end{gathered} \quad (2.59)$$

Перенесемо в ліву частину умову розрізнення (2.45) і скоротимо решту:

$$\begin{gathered}\Delta W_i \oplus \Delta W_j \oplus \sigma^2(x_i^L \oplus x_j^L) \oplus x_i^R \oplus x_j^R = \\ = \sigma^2(f_1(\Delta x_i) \oplus f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus \\ \oplus f_1(\Delta x_j) \oplus \sigma'(f_2(\Delta U_i) \oplus f_2(\Delta U_j))\end{gathered} \quad (2.60)$$

Звідси видно, що для виконання умови (2.45) має виконуватися рівність:

$$\begin{gathered}\sigma^2(f_1(\Delta x_i) \oplus f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus \\ \oplus f_1(\Delta x_j) \oplus \sigma'(f_2(\Delta U_i) \oplus f_2(\Delta U_j)) = 0\end{gathered}, \quad (2.61)$$

$$\begin{gathered}\sigma^2(f_1(\Delta x_i) \oplus f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j) = \\ = \sigma'(f_2(\Delta U_i) \oplus f_2(\Delta U_j))\end{gathered}. \quad (2.62)$$

Оскільки функції f_1 і f_2 випадкові, то ймовірність виконання рівняння (2.62) має наступне значення:

$$P = \frac{1}{2^n}. \quad (2.63)$$

Виходячи з двох вищеописаних випадків, можна знайти загальну ймовірність виконання рівності (2.45) для схеми Лей-Мессі при одній парі вхідних запитів. Дана ймовірність є сумою ймовірностей обох випадків. Сума

ймовірностей знаходиться за формулою складання ймовірностей для сумісних подій:

$$P_1(LM) = 1 - (1 - \frac{1}{2^n})^2 = \frac{2^{n+1} - 1}{2^{2n}}. \quad (2.64)$$

Якщо прийняти, що ймовірність виконання рівності (2.45) для всіх пар запитів однакова, то загальну ймовірність отримання «1» на виході алгоритму-розрізнювача можна отримати за допомогою наступного виразу:

$$P_2(LM) = 1 - (1 - \frac{2^{n+1} - 1}{2^{2n}})^{\frac{k(k-1)}{2}}. \quad (2.65)$$

де k – кількість запитів. Імовірність визначається за формулою додавання для сумісних подій, тому що одночасно кілька пар можуть задовольняти заданій рівності. Загальна кількість можливих пар становить $C_k^2 = \frac{k(k-1)}{2}$.

Насправді, формула (2.65) є апроксимаційним значенням. Точне значення ймовірності можна отримати за допомогою такої формули:

$$P_3(LM) = 1 - \prod_{i=0}^{k-2} (1 - \frac{1}{2^n - i})^{2(k-(i+1))}. \quad (2.66)$$

Це пов'язано з тим, що пари запитів не є незалежними і рівноімовірними. Однак відмінності в ймовірностях мінімальні, тому для спрощення доцільно користуватися апроксимованим значенням (2.65). Залежності між запитами розглядалися в попередніх розділах. В даному випадку має місце такий же випадок, як і в розділі 2.4, тому доказ формули (2.66) опускається. Оскільки апроксимаційна ймовірність більше точної, то

верхню межу ймовірності розрізнення можна задати як нерівність, що і зроблено в формулі (2.46). Надалі буде використовуватися саме апроксимаційне значення ймовірності.

Тепер розглянемо ймовірність виконання рівності (2.45) для випадкової перестановки. Значення ймовірності для $k = 2$ в такому випадку можна буде знайти за допомогою наступного виразу:

$$P(PRP) = \frac{2^n}{2^{2n} - 1}. \tag{2.67}$$

В даному випадку віднімання одиниці з знаменника обумовлено тим, що ми маємо випадкову перестановку, для якої вихідні значення, при різних вхідних значеннях, не повторюються. Тобто $W_i \neq W_j$ і відповідно неможливий випадок коли $W_i^L \oplus W_j^L = W_i^R \oplus W_j^R = 0$. Оскільки права частина виразу (2.45) ніколи не дорівнює нулю (за умовою, для ланцюга Фейстеля в якості лінійної функції), то неможливий один з невідповідних варіантів, тобто ймовірність збільшується.

Якщо прийняти, що ймовірність виконання рівності (2.45) для всіх пар запитів однакова, то загальну ймовірність отримання «1» на виході алгоритму-розрізнювача для випадкової перестановки можна отримати за допомогою наступного виразу:

$$P_2(LM) = 1 - (1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}}. \tag{2.68}$$

Спосіб знаходження такої ймовірності аналогічний описаному вище. Точне значення ймовірності можна отримати використовуючи такий вираз:

$$P_3(LM) = 1 - \prod_{i=0}^{k-2} (1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n})^{k-(i+1)} . \quad (2.69)$$

Доказ цієї формули вже наводився в попередніх розділах і тут опускається.

Таким чином, знаючи ймовірності розрізнення алгоритмом-розрізнювачем схеми Лей-Мессі і випадкової перестановки можна знайти значення переваги для даного алгоритму-розрізнювача. Перевага знаходиться як різниця значень (2.66) і (2.69) і в такому значенні:

$$\begin{aligned} Adv(LM, PRP) &= |\prod_{i=0}^{k-2} (1 - \frac{1}{2^n - i})^{2(k-(i+1))} - \prod_{i=0}^{k-2} (1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n})^{k-(i+1)}| \leq \\ &\leq |(1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}} - (1 - \frac{2^{n+1} - 1}{2^{2n}})^{\frac{k(k-1)}{2}}| . \end{aligned} \quad (2.70)$$

Доказ леми 2.12 закінчено.

Розглянемо, яку максимальну ймовірність розрізнення можна досягти за допомогою описаного алгоритму-розрізнювача. На рис. 2.17 наведено графік залежності ймовірності (вісь *Oy*) розрізнення для $n = 8$ і $n = 16$ від кількості вхідних запитів (вісь *Ox*).

Як видно з графіків, за допомогою описаного алгоритму-розрізнювача можна досягти ймовірності розрізнення порядку $P_{\max} \approx 0.3$. При цьому потрібно близько $\sqrt{2^n}$ обраних відкритих текстів.

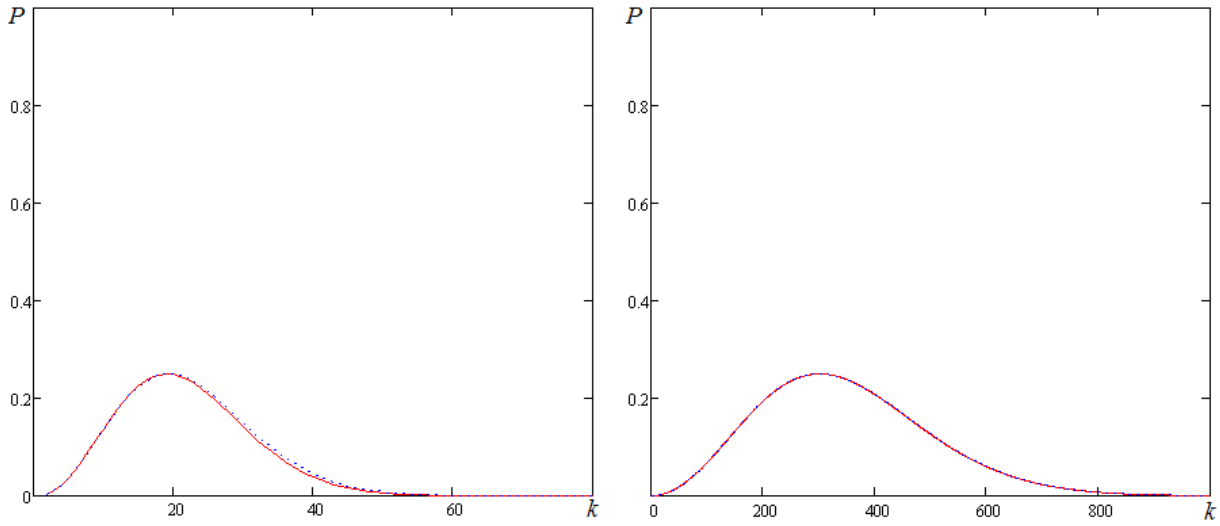


Рисунок 2.17 – Залежність ймовірності розрізнення від кількості вхідних запитів для $n = 8$ (зліва) і $n = 16$ (справа)

2.3.3.2 Алгоритм-розрізнювач для 3-раундової схеми Лей-Мессі з випадковими перестановками в якості раундових перетворень

Загальна схема приведена на рис. 2.15, тільки в даному випадку функції f_1, f_2, f_3 – це випадкові перестановки. В такому випадку алгоритм-розрізнювач зміниться.

Лема 2.13. Нехай ми маємо k запитів (x_i, w_i), причому для кожної можливої пари вхідних аргументів (x_i, x_j) виконується така умова: $\sigma(x_i^L \oplus x_j^L) \oplus x_i^R \oplus x_j^R \neq 0$ і $\Delta x_i \neq \Delta x_j$. Алгоритм-розрізнювач заснований на перевірці виконання рівності (2.71) для всіх можливих пар $[x_i, w_i]; [x_j, w_j]$, де $0 \le i < j < k$. Якщо хоча б для однієї пари дане рівність виконується, то на виході алгоритму-розрізнювача маємо «1».

$$\Delta W_i \oplus \Delta W_j = \sigma(x_i^R \oplus x_j^R \oplus x_i^L \oplus x_j^L). \qquad (2.71)$$

Тоді ймовірність розрізнення 3-раундової схеми Лей-Мессі і випадкової перестановки має наступне значення:

$$Adv(LM, PRP) \le |\,(1-\frac{2^n}{2^{2n}-1})^{\frac{k(k-1)}{2}} - (1-\frac{1}{2^n-1})^{k(k-1)}\,|. \qquad (2.72)$$

Доказ леми 2.13. Розглянемо ймовірність виконання (2.71) для схеми Лей-Мессі при $k = 2$. У такому випадку ймовірність буде дорівнювати наступному значенню:

$$P(LM) = 1-(1-\frac{1}{2^n-1})^2. \qquad (2.73)$$

Розглянемо докладніше формулу (2.73).

Рівність (2.71) буде виконуватися в двох випадках:

1. У разі виникнення колізії $\Delta U_i = \Delta U_j$. Тоді маємо наступне:

$$\begin{gathered}\sigma(x_i^L \oplus f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i) = \sigma(x_j^L \oplus f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j),\\ \sigma(x_i^L) \oplus \sigma(f_1(\Delta x_i)) \oplus x_i^R \oplus f_1(\Delta x_i) = \sigma(x_j^L) \oplus \sigma(f_1(\Delta x_j)) \oplus x_j^R \oplus f_1(\Delta x_j), \qquad (2.74)\\ \sigma(f_1(\Delta x_i)) \oplus \sigma(f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j) = \sigma(x_j^L) \oplus \sigma(x_i^L) \oplus x_j^R \oplus x_i^R.\end{gathered}$$

Оскільки $\Delta x_i \neq \Delta x_j$, а f_1, f_2, f_3 – це перестановки, то для всіх запитів маємо наступне:

$$\sigma(f_1(\Delta x_i)) \oplus \sigma(f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j) \neq 0. \qquad (2.75)$$

З (2.74) і (2.75) також випливає, що $\sigma(x_j^L) \oplus \sigma(x_i^L) \oplus x_j^R \oplus x_i^R \neq 0$ – це початкова умова для вхідних текстів.

Якщо сталася колізія і рівність (2.74) виконується, то отримуємо наступний вираз:

$$\begin{gathered}\Delta W_i \oplus \Delta W_j = \sigma^2(x_i^L) \oplus \sigma^2(f_1(\Delta x_i)) \oplus \sigma(f_2(\Delta U_i)) \oplus \\ \oplus x_i^R \oplus f_1(\Delta x_i) \oplus f_2(\Delta U_i) \oplus \sigma^2(x_j^L) \oplus \sigma^2(f_1(\Delta x_j)) \oplus \\ \oplus \sigma(f_2(\Delta U_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_j) = \\ = \sigma^2(x_i^L \oplus x_j^L) \oplus x_i^R \oplus x_j^R \oplus \sigma^2(f_1(\Delta x_i) \oplus \\ \oplus f_1(\Delta x_j)) \oplus f_1(\Delta x_i) \oplus f_1(\Delta x_j)\end{gathered} . \quad (2.76)$$

З (2.74) і (2.76) можемо отримати такий вираз:

$$\Delta W_i \oplus \Delta W_j = \sigma(x_i^L \oplus x_j^L \oplus x_i^R \oplus x_j^R). \quad (2.77)$$

Вираз (2.77) відповідає умові розрізнення. Тобто вираз (2.77) буде виконуватися при виконанні (2.74). Ймовірність виконання (2.74) має таке значення:

$$P_1(LM) = \frac{1}{2^n - 1}. \quad (2.78)$$

Зменшення знаменника на одиницю маємо через умову $\sigma(x_j^L) \oplus \sigma(x_i^L) \oplus x_j^R \oplus x_i^R \neq 0$ для вхідних текстів. Це збільшує ймовірність, оскільки ліва частина (2.74) ніколи не дорівнює нулю.

2. У разі відсутності колізії $\Delta U_i = \Delta U_j$ і при виконанні наступної рівності:

$$\begin{gathered}\Delta W_i \oplus \Delta W_j = \sigma^2(x_i^L) \oplus \sigma^2(f_1(\Delta x_i)) \oplus \sigma(f_2(\Delta U_i)) \oplus \\ \oplus x_i^R \oplus f_1(\Delta x_i) \oplus f_2(\Delta U_i) \oplus \sigma^2(x_j^L) \oplus \sigma^2(f_1(\Delta x_j)) \oplus \\ \oplus \sigma(f_2(\Delta U_j)) \oplus x_j^R \oplus f_1(\Delta x_j) \oplus f_2(\Delta U_j) = \\ = \sigma(x_i^L \oplus x_i^R \oplus x_j^L \oplus x_j^R).\end{gathered} \quad (2.79)$$

Оскільки функції f_1 і f_2 – це випадкові перестановки, а $\Delta W_i \oplus \Delta W_j$ – випадкове значення (з деякими застереженнями), то ймовірність виконання рівняння (2.79) має таке значення:

$$P = \frac{2^n + 1}{2^{2n} - 1} = \frac{1}{2^n - 1}. \qquad (2.80)$$

Виходячи з двох вищеописаних випадків можна знайти загальну ймовірність виконання рівності (2.71) для схеми Лей-Мессі з випадковими раундовому перестановками при одній парі вхідних запитів. Дана ймовірність є сумою ймовірностей обох випадків. Сума ймовірностей знаходиться за формулою складання ймовірностей для сумісних подій:

$$P_1(LM) = 1 - (1 - \frac{1}{2^n - 1})^2. \qquad (2.81)$$

Якщо прийняти, що ймовірність виконання рівності (2.71) для всіх пар запитів однакова, то загальну ймовірність отримання «1» на виході алгоритму-розрізнювача можна отримати за допомогою наступного виразу:

$$P_2(LM) = 1 - (1 - \frac{1}{2^n - 1})^{2^{\frac{k(k-1)}{2}}}. \qquad (2.82)$$

де k – кількість запитів. Імовірність визначається за формулою додавання для сумісних подій, тому що одночасно кілька пар можуть задовольняти заданій рівності. Загальна кількість можливих пар становить $C_k^2 = \frac{k(k-1)}{2}$.

Насправді, формула (2.82) є апроксимаційним значенням. Точне значення ймовірності можна отримати за допомогою такої формули:

$$P_3(LM) = 1 - \prod_{i=0}^{k-2} (1 - \frac{1}{2^n - 1 - i})^{2(k-(i+1))}. \quad (2.83)$$

Це пов'язано з тим, що пари запитів не є незалежними і рівноімовірними. Однак відмінності в можливостях мінімальні, тому для спрощення доцільно користуватися апроксимувати значенням (2.82). Залежності між запитами розглядалися в попередніх розділах. В даному випадку має місце такий же випадок, як і в розділі 2.4, тому доказ формули (2.83) опускається.

Тепер розглянемо ймовірність виконання рівності (3.81) для випадкової перестановки. Значення ймовірності для *k = 2* в такому випадку можна буде знайти за допомогою наступного виразу:

$$P(PRP) = \frac{2^n}{2^{2n} - 1}. \quad (2.84)$$

В даному випадку віднімання одиниці з знаменника обумовлено тим, що ми маємо випадкову перестановку, для якої вихідні значення, при різних вхідних значеннях, не повторюються. Тобто $W_i \neq W_j$ і відповідно неможливий випадок коли $W_i^L \oplus W_j^L = W_i^R \oplus W_j^R = 0$. Оскільки права частина виразу (2.71) ніколи не дорівнює нулю (за умовою, для ланцюга Фейстеля в якості лінійної функції), то неможливий один з невідповідних варіантів, тобто ймовірність збільшується.

Якщо прийняти, що ймовірність виконання рівності (2.71) для всіх пар запитів однакова, то загальну ймовірність отримання «1» на виході алгоритму-розрізнювача для випадкової перестановки можна отримати за допомогою наступного виразу:

$$P_2(LM) = 1 - (1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}} . \quad (2.85)$$

Спосіб знаходження такої ймовірності аналогічний описаному вище для схеми Лей-Мессі. Точне значення ймовірності можна отримати використовуючи такий вираз:

$$P_3(LM) = 1 - \prod_{i=0}^{k-2} (1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n})^{k-(i+1)} . \quad (2.86)$$

Доказ цієї формули вже наводилося в попередніх розділах і тут опускається.

Таким чином, знаючи ймовірності розрізнення алгоритмом-розрізнювачем схеми Лей-Мессі і випадкової перестановки можна знайти значення переваги для даного алгоритму-розрізнювача. Перевага знаходиться як різниця значень (2.71) і (2.74) і в такому значенні:

$$\begin{aligned} Adv(LM, PRP) = & \left| \prod_{i=0}^{k-2} (1 - \frac{1}{2^n - 1 - i})^{2(k-(i+1))} - \right. \\ & \left. - \prod_{i=0}^{k-2} (1 - \frac{2^n}{2^{2n} - 1 - i \cdot 2^n})^{k-(i+1)} \right| \leq \\ \leq & \left| (1 - \frac{1}{2^n - 1})^{k(k-1} - (1 - \frac{2^n}{2^{2n} - 1})^{\frac{k(k-1)}{2}} \right| . \end{aligned} \quad (2.87)$$

Доказ леми 2.13 закінчено.

Розглянемо, яку максимальну ймовірність розрізнення можна досягти за допомогою описаного алгоритму-розрізнювача. На рис. 2.18 наведено

графік залежності ймовірності (вісь *Oy*) розрізнення для $n = 8$ і $n = 16$ від кількості вхідних запитів (вісь *Ox*).

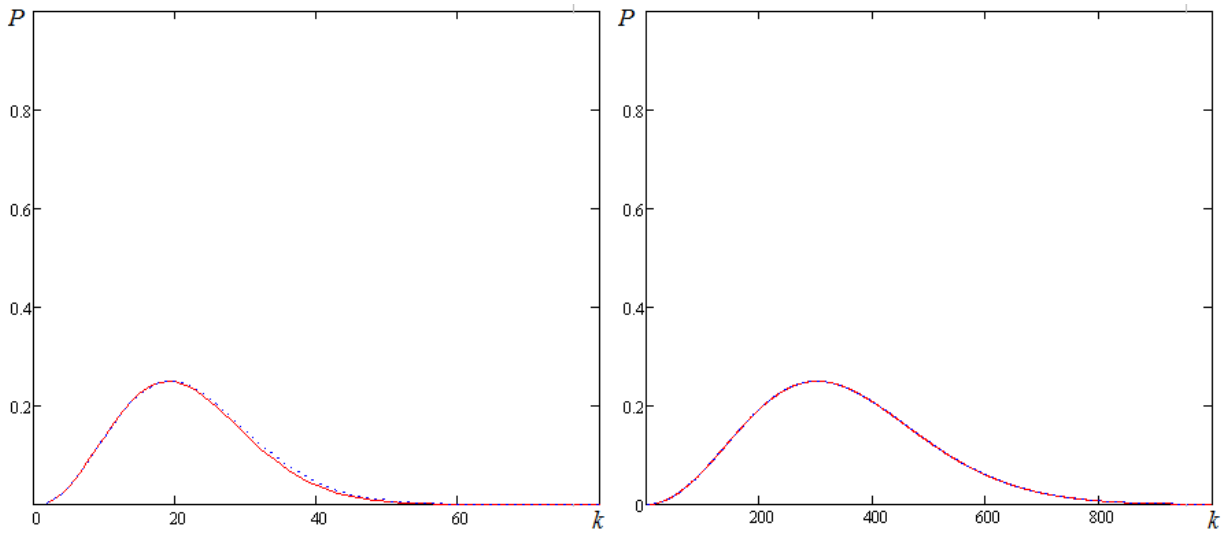


Рисунок 2.18 – Залежність ймовірності розрізнення від кількості вхідних запитів для *n=8* (зліва) та *n=16* (справа)

Як видно з графіків, за допомогою описаного алгоритму-розрізнювача можна досягти ймовірності розрізнення порядку $P_{\max} \approx 0.3$. При цьому потрібно близько $\sqrt{2^n}$ обраних відкритих текстів. Даний алгоритм має практично такі ж показники, як і алгоритм, описаний вище для 3-раундової схеми Лей-Мессі з випадковими функціями у якості раундових перетворень.

На підставі вищенаведеного можна зробити наступні висновки:

1. Схема Лей-Мессі поряд з ланцюгом Фейстеля і SPN-структурою є однією з найбільш популярних конструкцій для побудови блокових симетричних шифрів.

2. Для оцінки ефективності схеми Лей-Мессі застосовується такий же підхід, як і для ланцюга Фейстеля. Оцінити ефективність даної конструкції кількісно можна завдяки алгоритмам-розрізнювачам, які оцінюють пари вхідних / вихідних текстів, вироблених схемою Лей-Мессі. Порівнюючи

ймовірність її розрізнення з ймовірністю розрізнення випадкової підстановки можна отримати кількісну оцінку ефективності.

3. Для схеми Лей-Мессі може бути знайдена максимально можлива ймовірність розрізнення. При однаковій кількості вхідних даних дана ймовірність буде менше, ніж у ланцюзі Фейстеля, що свідчить про те, що вона більш ефективна.

4. Було знайдено два алгоритми-розрізнювачі для 3-раундової схеми Лей-Мессі. Один для конструкції з випадковою функцією як раундового перетворення, інший для конструкції з випадковою підстановкою в якості раундового перетворення. В обох випадках ймовірність розрізнення становить приблизно $P_{\max} \approx 0.3$. Імовірність розрізнення менше, ніж у ланцюзі Фейстеля, що свідчить про більшу ефективність схеми Лей-Мессі.

2.4 Оцінка ефективності розрізнення БСШ на основі SPN-структури

На даний момент SPN-структура поряд з ланцюгом Фейстеля і схемою Лей-Мессі є однією з найпопулярніших високорівневих конструкцій для побудови блочного симетричного шифру. На основі SPN-структури засновані такі шифри як Anubis [57], GrandCru [58], Noekeon [59], «Калина» [25], а також найбільш поширений шифр в світі AES та ін. Розділ присвячений опису результатів, отриманих в ході дослідження даної конструкції.

2.4.1 Загальна схема БСШ на основі SPN-структури

Основу SPN-структури складають операції перемішування (diffusion) і розсіювання (confusion), які реалізуються за допомогою лінійних і нелінійних перетворень відповідно. Даний принцип був запропонований Клодом Шенноном [60].

Внутрішній стан оброблюваного шифром блоку може бути представлено як прямокутна матриця розміром $n_c \times n_b$ елементів. Це подання

використовується в шифрі AES. Кожен елемент має розмір n_e бітів, тоді розмір блоку буде складати $2n = n_c \times n_b \times n_e$ бітів. Для забезпечення необхідних криптографічних властивостей доцільно вибирати $n_c = n_b \cdot 2l, l \in \{1,2,..\}$, що також дозволить збільшити ефективність програмної реалізації. Також слід використовувати $n_e = 8$, оскільки всі сучасні алгоритми є байт-орієнтованими.

Один цикл шифрування при $n_c = n_b$ може бути представлений за допомогою наступної схеми на рис. 2.19.

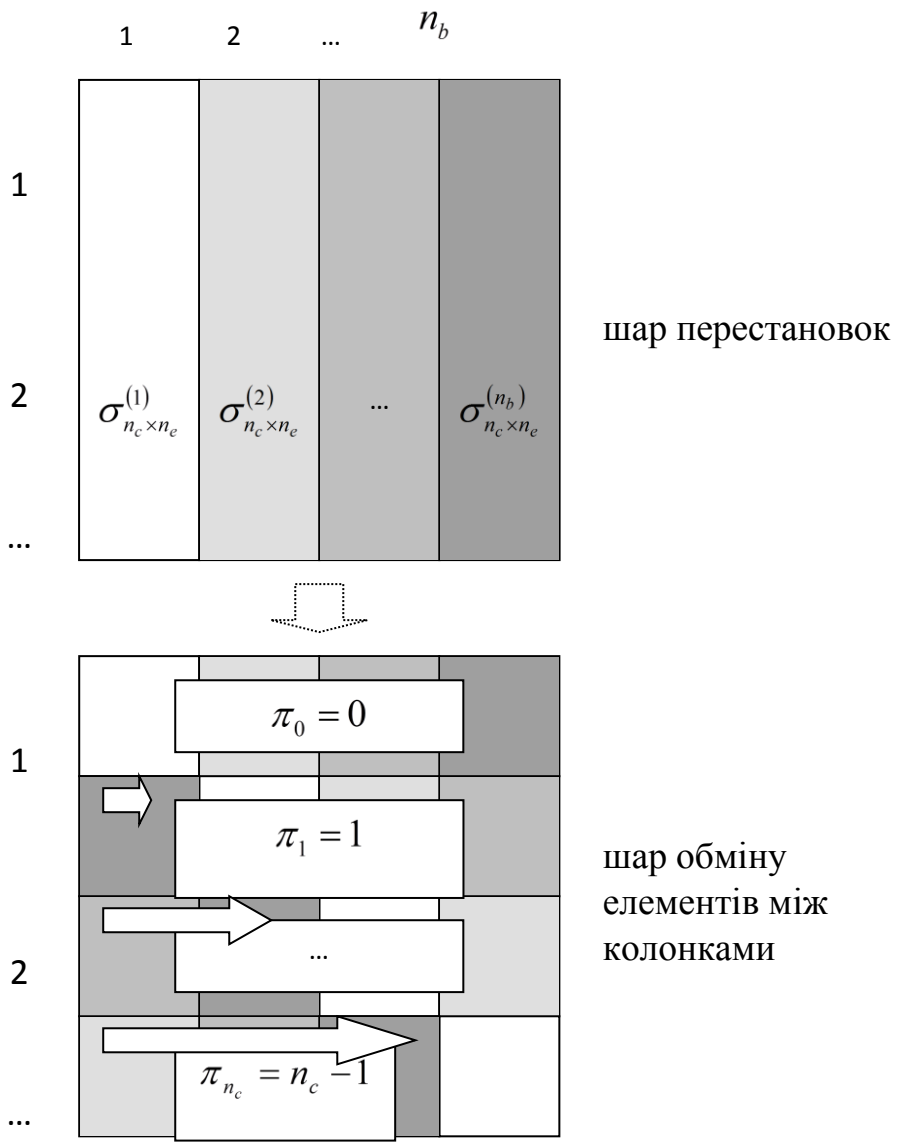


Рисунок 2.19 – Раунд шифрування алгоритмом на основі SPN-структури

Як видно на рис. 2.19 в основі SPN-конструкції лежать підстановчі таблиці, які реалізують нелінійний блок, і операції зсуву рядків, що реалізують відповідно лінійні перетворення.

Загальну модель шифру на основі даної структури можна представити за допомогою наступного виразу:

$$\vartheta = \prod_{i=1}^{s} \left(\pi_0 \bullet \pi_1 \bullet \ldots \bullet \pi_{n_c} \right) \circ \left(\sigma_{n_c \times n_e}^{(1,i)} \bullet \sigma_{n_c \times n_e}^{(2,i)} \bullet \ldots \bullet \sigma_{n_c \times n_e}^{(n_b,i)} \right), \qquad (2.88)$$

де s – кількість раундів шифрування. На вхід шифрувального перетворення подається значення $x_i = \left(x_i^{(1)} \bullet x_i^{(2)} \bullet \ldots \bullet x_i^{(n_b)} \right)$, на виході формується $y_i = \left(y_i^{(1)} \bullet y_i^{(2)} \bullet \ldots \bullet y_i^{(n_b)} \right)$.

Таким чином, дана конструкція являє собою ітеративне перетворення, яке на кожній окремій ітерації включає в себе шар перестановок і шар обміну елементів між колонками матриці стану.

Для виключення впливу раундових перетворень, як такі слід використовувати випадкові перестановки. На відміну від ланцюга Фейстеля і схеми Лей-Мессі для SPN-структури не можуть бути використані випадкові функції в якості раундових перетворень, оскільки в такому випадку неможливо буде реалізувати коректне розшифрування, отже, при аналізі варто розглядати тільки биєктивні раундові перетворення. За аналогією з AES таке перетворення можна назвати Super-S-box (див., наприклад, [61]).

2.4.2 Оцінка ефективності розрізнення 2-раундової SPN-структури

В даному розділі будуть описані вираження для знаходження максимально можливої ймовірності розрізнення для 2-раундової SPN-структури. Як і у випадку з ланцюгом Фейстеля і схемою Лей-Мессі основою методу оцінки є знаходження ймовірності розрізнення SPN-структури і

випадкової перестановки σ_{2n}. Чим менше така ймовірність, тим ефективніше Високорівнева конструкція. В ідеальному випадку ймовірність розрізнення алгоритмом-розрізнювачем дорівнює нулю. Отже описані в даному розділі теореми дозволяють отримати верхню межу розрізнення, що само по собі є кількісною оцінкою ефективності конструкції.

2.4.2.1 Максимальна ймовірність розрізнення 2-раундової SPN-структури і випадкової перестановки

Доведемо наступну теорему щодо максимальної ймовірності розрізнення 2-раундової SPN-структури:

Теорема 2.1 (Верхня межа розрізнення 2-циклової SPN-структури і випадкової перестановки на обмеженому числі запитів). Нехай ми маємо SPN-структуру з розміром блоку $2n$ бітів і внутрішнім станом, що подаються у вигляді матриці розміром $n_c \times n_b$ елементів, кожен з яких має розмір бітів ($2n = n_c \times n_b \times n_e$). При цьому виконується умова $n_c = n_b \cdot 2l, l \in \{1,2,..\}$ і у якості раундового перетворення використовуються випадкові перестановки $\sigma_{\frac{2n}{n_c \cdot n_e}}$. Тоді максимальна ймовірність розрізнення випадкової перестановки σ_{2n} і 2-циклової SPN-структури при k запитів $\left(2 \le k \le 2^{\frac{n_c \cdot n_e}{n_b}}\right)$ на вході алгоритму-розрізнювача не перевищує наступного значення:

$$Adv_{\eta^*}(\vartheta,\sigma_{2n}) =$$

$$= \left| \begin{matrix} P_1\left(\eta_*(f(x_1),...,f(x_k)) = 1 : f \in_R \vartheta^{(2)}\right) - \\ -P_1^*\left(\eta_*(f(x_1),...,f(x_k)) = 1 : f \in_R \sigma_{2n}\right) \end{matrix} \right| \le$$

$$\le \left| \prod_{i=0}^{\frac{k(k-1)}{2}-1} \frac{\left(\left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{n_b} - i \right)}{\left(2^{n_c \cdot n_e} - i \right)} - \prod_{i=0}^{k-2} \left(1 - 2^{i - \frac{n_c \cdot n_e}{n_b}} \right)^{(n_b-1)(k-i-1)} \right| \approx \qquad 2.89$$

$$\approx \left| 2^{-\frac{n_c \cdot n_e \cdot k \cdot (k-1)}{2}} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{\frac{n_b k(k-1)}{2}} - \left(1 - 2^{-\frac{n_c \cdot n_e}{n_b}} \right)^{\frac{k(k-1)(n_b-1)}{2}} \right|.$$

Доказ теореми 2.1.

Імовірність розрізнення SPN-структури можна оцінити таким чином. На кожному етапі роботи алгоритму оцінюється одна пара елементів $(x_i, x_j), 1 \le i < j \le k$. При цьому, вхідні стану елементів з пари повинні відрізнятися тільки для одного стовпчика (тобто буде задіяний тільки один Super-S-box, який ми будемо називати активним). В такому випадку, після 1-го раунду в кожному стовпці (тобто на кожну перестановку подальшого раунду) буде $\frac{n_c}{n_b}$ активних елементів (байтів). Для того щоб відрізнити SPN-структуру повинна виникнути колізія хоча б для одного стовпчика (тобто значення двох запитів (x_i, x_j) повинні співпасти) на вході перестановки другого раунду. Всього таких колізій може бути від нуля до $(n_b - 1)$ (колізії для всіх елементів можливі для випадкової функції, але не перестановки).

Тоді ймовірність появи колізії на вході однієї перестановки буде дорівнює $p_\sigma = 2^{-\frac{n_c \cdot n_e}{n_b}}$, а ймовірність появи хоча б однієї колізії на вході всіх перестановок другого раунду можна знайти за допомогою наступного виразу:

$$p_2 = 1 - \left(1 - 2^{-\frac{n_c \cdot n_e}{n_b}}\right)^{n_b - 1}. \qquad (2.90)$$

Вираз (4.3) визначає ймовірність розрізнення SPN-структури на одному запиті, що складається з двох вхідних значень. Для k вхідних значень можна скласти $C_k^2 = \frac{k(k-1)}{2}$ різних запитів. Імовірність появи колізії серед всіх можливих запитів дорівнює:

$$p_{\eta_*} = 1 - \left(\left(1 - \frac{1}{2^{\frac{n_c \cdot n_e}{n_b}}}\right)^{n_b - 1}\right)^{k-1} \left(\left(1 - \frac{1}{2^{\frac{n_c \cdot n_e}{n_b}} - 1}\right)^{n_b - 1}\right)^{k-2} \ldots \left(\left(1 - \frac{1}{2^{\frac{n_c \cdot n_e}{n_b}} - (k-2)}\right)^{n_b - 1}\right)^{k-(k-1)} =$$

$$= 1 - \prod_{i=0}^{k-2} \left(1 - \frac{1}{2^{\frac{n_c \cdot n_e}{n_b}} - i}\right)^{(n_b - 1)(k-i-1)}.$$

При малому розмірі вибірки $\left(\#\{x_i\} << 2^{\frac{n_c \cdot n_e}{n_b}}\right)$ ймовірність появи колізії змінюється незначно, внаслідок чого можна скористатися наступною апроксимацією:

$$p_{\eta_*} \approx 1 - \left(1 - \frac{1}{2^{\frac{n_c \cdot n_e}{n_b}}}\right)^{\frac{k(k-1)(n_b-1)}{2}}. \qquad (2.91)$$

Імовірність розрізнення випадкової перестановки σ_{2n} аналогічним алгоритмом, при якому перевіряються колізії на кордонах Super-S-box-ів, можна оцінити наступним чином.

Для одного запиту (пари значень (x_i, x_j)) ймовірність колізії хоча б в одному з стовпців дорівнює $p_1 = 1 - 2^{-n_c \cdot n_e} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{n_b}$. Для k вхідних значень можна скласти $C_k^2 = \frac{k(k-1)}{2}$ різних запитів, тоді ймовірність того, що колізія трапиться хоча б в одному запиті буде відповідати наступному виразу:

$$P_1^* \left(\eta_*(f(x_1),..., f(x_k)) = 1 : f \in_R \sigma_{2n} \right) =$$

$$= 1 - \frac{\left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{n_b} \cdot \left(\left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{n_b} - 1 \right) \cdot ... \cdot \left(\left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{n_b} - \frac{k(k-1)}{2} + 1 \right)}{2^{n_c \cdot n_e} \cdot \left(2^{n_c \cdot n_e} - 1 \right) \cdot ... \cdot \left(2^{n_c \cdot n_e} - \frac{k(k-1)}{2} + 1 \right)} =$$

$$= 1 - \prod_{i=0}^{\frac{k(k-1)}{2}-1} \frac{\left(\left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{n_b} - i \right)}{\left(2^{n_c \cdot n_e} - i \right)}.$$

При малому розмірі вибірки $\left(\#\{x_i\} << 2^{\frac{n_c \cdot n_e}{n_b}} \right)$ ймовірність появи колізії змінюється незначно, тому, з метою спрощення наведеної вище формули, можна скористатися наступним виразом для знаходження апроксимованого значення ймовірності з мінімальною похибкою:

$$P_1^*\left(\eta_*(f(x_1),\ldots,f(x_k))=1:\ f\in_R \sigma_{2n}\right)\approx 1-2^{-\frac{n_c\cdot n_e\cdot k\cdot(k-1)}{2}}\cdot\left(2^{\frac{n_c\cdot n_e}{n_b}}-1\right)^{\frac{n_b k(k-1)}{2}}.$$

Звідси можна знайти перевагу розрізнення SPN-структури над розрізненням випадкової перестановки, воно становитиме абсолютну різницю двох ймовірностей:

$$Adv_{\eta^*}(\vartheta,\sigma_{2n})=$$

$$=\left|P_1\left(\eta_*(f(x_1),\ldots,f(x_k))=1:\ f\in_R \vartheta^{(2)}\right)-P_1^*\left(\eta_*(f(x_1),\ldots,f(x_k))=1:\ f\in_R \sigma_{2n}\right)\right|\le$$

$$\le\left|1-\prod_{i=0}^{k-2}\left(1-2^{i-\frac{n_c\cdot n_e}{n_b}}\right)^{(n_b-1)(k-i-1)}-\left(1-\prod_{i=0}^{\frac{k(k-1)}{2}-1}\frac{\left(\left(2^{\frac{n_c\cdot n_e}{n_b}}-1\right)^{n_b}-i\right)}{\left(2^{n_c\cdot n_e}-i\right)}\right)\right|=$$

$$=\left|\prod_{i=0}^{\frac{k(k-1)}{2}-1}\frac{\left(\left(2^{\frac{n_c\cdot n_e}{n_b}}-1\right)^{n_b}-i\right)}{\left(2^{n_c\cdot n_e}-i\right)}-\prod_{i=0}^{k-2}\left(1-2^{i-\frac{n_c\cdot n_e}{n_b}}\right)^{(n_b-1)(k-i-1)}\right|.$$

Тоді апроксимаційний вираз буде виглядати наступним чином:

$$Adv_{\eta^*}(\vartheta,\sigma_{2n}) =$$

$$= \left| P_1\left(\eta_*(f(x_1),...,f(x_k)) = 1 : f \in_R \vartheta^{(2)}\right) - P_1^*\left(\eta_*(f(x_1),...,f(x_k)) = 1 : f \in_R \sigma_{2n}\right)\right| \le$$

$$\le \left| 1 - \left(1 - 2^{-\frac{n_c \cdot n_e}{n_b}}\right)^{\frac{k(k-1)(n_b-1)}{2}} - \left(1 - 2^{-\frac{n_c \cdot n_e \cdot k \cdot (k-1)}{2}} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1\right)^{\frac{n_b k(k-1)}{2}}\right)\right| =$$

$$= \left| 2^{-\frac{n_c \cdot n_e \cdot k \cdot (k-1)}{2}} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1\right)^{\frac{n_b k(k-1)}{2}} - \left(1 - 2^{-\frac{n_c \cdot n_e}{n_b}}\right)^{\frac{k(k-1)(n_b-1)}{2}}\right|.$$

Теорема доведена.

2.4.2.2 Алгоритм-розрізнювач для 2-раундової SPN-структури

Для 2-раундової SPN-структури можна побудувати наступний алгоритм розрізнювач.

Алгоритм-розрізнювач 1 для 2-раундової SPN-структури.

Для k вхідних запитів (x_i, x_j), де $\left(2 \le k \le 2^{\frac{n_c \cdot n_e}{n_b}} + 1\right)$, що виконують активізацію однієї перестановки: $x_i^{(t)} \ne x_j^{(t)}, x_i^{(l)} = x_j^{(l)}, 1 \le l \le n_b, l \ne t, t = const$, $x_i = \left(x_i^{(1)} \bullet x_i^{(2)} \bullet ... \bullet x_i^{(n_b)}\right)$, $x_j = \left(x_j^{(1)} \bullet x_j^{(2)} \bullet ... \bullet x_j^{(n_b)}\right)$, $1 \le i < j \le k$ перевіряється наявність колізій на виході кожної перестановки другого циклу: $y_i^{(l)} = y_j^{(l)}, 1 \le l \le n_b$, $1 \le i < j \le k$.

Якщо хоча б для одного запиту була знайдена колізія, то вихідне значення алгоритму-розрізнювача буде «1» (виявлена SPN-структура), інакше «0».

Для такого алгоритму-розрізнювача ймовірність розрізнення можна отримати за допомогою виразу з теореми 2.1.

Розглянемо графік залежності ймовірності розрізнення 2-раундової SPN-структури з параметрами $n_c = 4, n_b = 4, n_e = 8$ (така конфігурація еквівалентна 2-раундовому шифру AES з випадковими перестановками

Super-S-box в якості раундовий перетворень). На рис. 2.20 наведено графік залежності ймовірності від кількості вхідних текстів.

Як видно з графіка існує певна кількість вхідних аргументів, при якому ймовірність розрізнення досягає максимального значення, яке становить приблизно $Adv(2rSPN) = 0.6$. При малій кількості вхідних аргументів ймовірність появи «1» на виході алгоритму-розрізнювача як для SPN-структури, так і для випадкової перестановки близька до нуля. При значній кількості вхідних аргументів ймовірність появи «1» як для SPN-структури, так і для випадкової перестановки буде прагнути до одиниці, відповідно ймовірність розрізнення тоді буде прагнути до нуля.

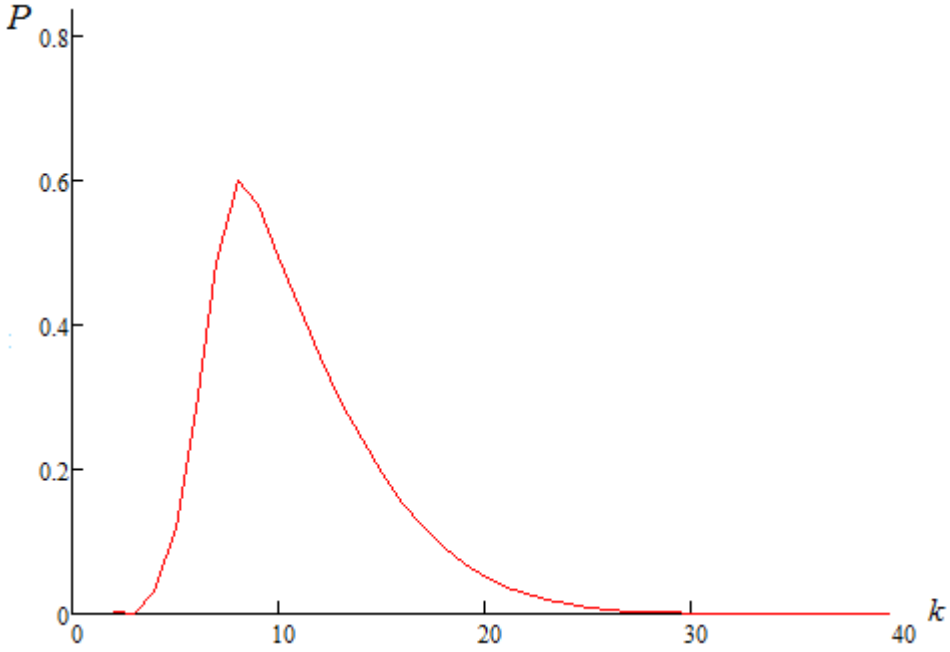


Рисунок 2.20 – Графік залежності максимально можливої ймовірності розрізнення 2-раундової SPN-структури від кількості вхідних аргументів k

На рис. 2.21 зображений графік залежності ймовірності появи «1» на виході алгоритму-розрізнювача від кількості вхідних аргументів для SPN-структури. Такий же графік, тільки для випадкової перестановки зображений на рис. 2.22. Як видно обидва графіка прагнуть до одиниці при збільшенні числа вхідних аргументів, однак швидкість досягнення одиниці різна, саме це

дозволяє нам вдало провести відрізнення SPN-структури від випадкової перестановки. По суті, ймовірність розрізнення на графіку 4.2 є різницею графіків 4.3 і 4.4. Це також підтверджується і теоремою 4.1.

Як видно з графіків на рис. 2.20-2.22, розрізнення навіть 2-раундової SPN-структури є непростим завданням. Значення, що повертається алгоритмом-розрізнювачем, в значній мірі залежить як від досліджуваного перетворення (блочний шифр або випадкова перестановка), так і від кількості вхідних даних.

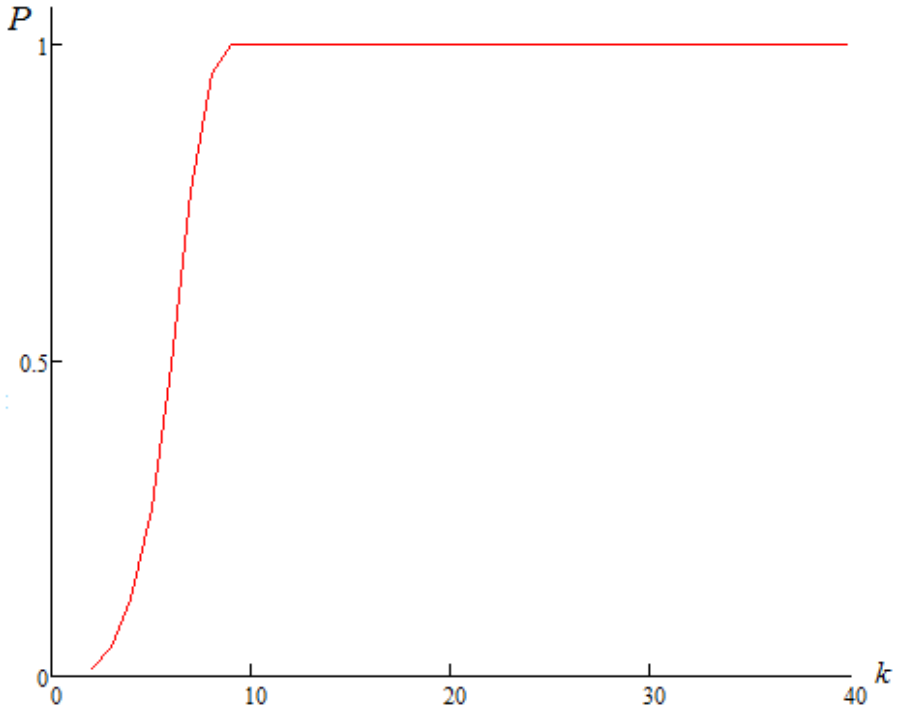


Рисунок 2.21 – Графік залежності ймовірності появи «1» на виході алгоритму-розрізнювача для SPN-структури

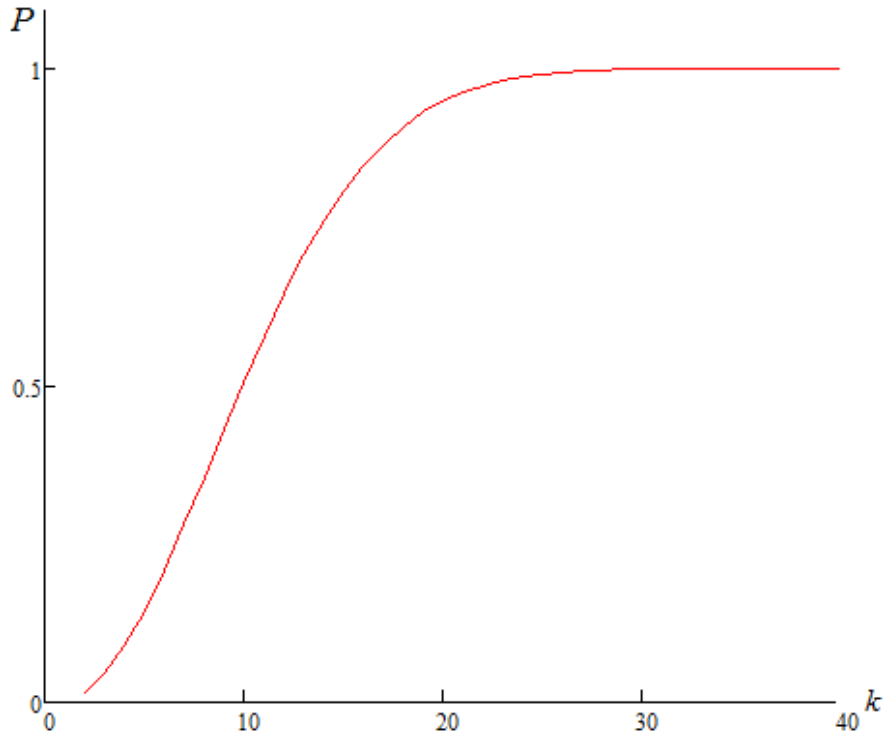


Рисунок 2.22 – Графік залежності ймовірності появи «1» на виході алгоритму-розрізнювача для випадкової перестановки

Як і для інших високорівневих конструкцій існує деяка оптимальна кількість вхідних аргументів, які слід подати на вхід алгоритму-розрізнювача, щоб отримати максимальну ймовірність розрізнення.

2.4.3 Оцінка ефективності розрізнення 3-раундової SPN-структури

Як і для 2-раундового перетворення, на вході SPN-структури ми маємо відкритий текст $x_i = \left(x_i^{(1)} \bullet x_i^{(2)} \bullet \ldots \bullet x_i^{(n_b)}\right)$. На виході формується зашифрований текст $y_i = \left(y_i^{(1)} \bullet y_i^{(2)} \bullet \ldots \bullet y_i^{(n_b)}\right)$. Для розрізнення 3-раундової SPN-структури і випадкової перестановці, як і в попередньому випадку, нам потрібно знайти колізійні властивості перетворення. Однак тепер колізії слід шукати не тільки на виході 1-го раунду, але і на виході 2-го раунду.

Теорема 2.2 (ефективність розрізнення 3-раундової SPN-структури і випадкової перестановки). Нехай ми маємо 3-раундову SPN-структуру з розміром блоку $2n$ бітів. Внутрішній стан блоку представляється у вигляді матриці розміром $n_c \times n_b$ елементів, кожен з яких має розмір n_e бітів

$2n = n_c \times n_b \times n_e$. При цьому виконується умова $n_c = n_b \cdot 2l, l \in \{1,2,..\}$, а в якості раундових перетворень використовуються випадкові перестановки $\sigma_{\frac{2n}{n_c \cdot n_e}}$. Тоді перевага алгоритму-розрізнювача випадкової перестановки і 3-раундової SPN-структури для довільної кількості запитів дорівнює нулю:

$$Adv_{\eta^*}(\vartheta, \sigma_{2n}) = \left| \begin{matrix} P_1\left(\eta_*(f(x_1),...,f(x_k)) = 1 : f \in_R \vartheta^{(3)}\right) - \\ P_1^*\left(\eta_*(f(x_1),...,f(x_k)) = 1 : f \in_R \sigma_{2n}\right) \end{matrix} \right| = 0. \qquad (2.92)$$

Доказ.

Для появи колізії на виході 3-го раунду SPN-структури необхідно, щоб на вході 3-го раунду хоча б для одного стовпчика були тільки колізійні значення елементів, тобто потрібно не менше n_c однакових елементів ($2^{n_c \cdot n_e}$ біт) після 2-го циклу.

Оптимальна активізація передбачає подачу різних значень одного елемента на вхід одної перестановки і однакові значення для інших елементів. Тільки в такому випадку можна домогтися максимальної кількості однакових елементів на вході другого раунду і, відповідно, максимальної ймовірності появи колізії для будь-якого елемента.

Однак навіть при оптимальній активізації вхідних елементів одночасна поява n_c однакових елементів у внутрішньому стані після другого циклу неможливо, але ця обставина має ненульову ймовірність при використанні випадкових значень на вході SPN-перетворення, які активізують не менше двох перестановок першого циклу.

Отже, для розрізнення 3-раундової SPN-структури потрібно, щоб в проміжному стані після 2-го циклу утворилося не менше n_c колізійних елементів. Імовірність такої події становить $\left(2^{-\frac{n_c \cdot n_e}{n_b}}\right)^{n_c} = 2^{-\frac{n_c^2 \cdot n_e}{n_b}}$.

Відповідно, для k вхідних елементів ймовірність появи n_c колізійних елементів хоча б в одному із запитів дорівнює наступному значенню:

$$P_1\left(\eta_*(f(x_1),\ldots,f(x_k))=1:\ f\in_R \vartheta^{(3)}\right)=1-\left(1-2^{-\frac{n_c^2\cdot n_e}{n_b}}\right)^{\frac{k(k-1)}{2}}. \qquad (2.93)$$

Для випадкової перестановки ступеня $2n$ ймовірність одночасної появи n_c співпадаючих елементів на виході 2-го раунду визначається наступним чином.

Кожен елемент містить в собі $\frac{n_c}{n_b}\cdot n_e$ бітів, тоді для n_c елементів ймовірність колізії дорівнює $\left(2^{-\frac{n_c\cdot n_e}{n_b}}\right)^{n_c}=2^{-\frac{n_c^2\cdot n_e}{n_b}}$.

Відповідно, для k вхідних елементів можна сформувати $C_k^2=\frac{k(k-1)}{2}$ запитів. Тоді ймовірність появи n_c колізійних елементів хоча б в одному із запитів дорівнює:

$$P_1^*\left(\eta_*(f(x_1),\ldots,f(x_k))=1:\ f\in_R \sigma_{2n}\right)=1-\left(1-2^{-\frac{n_c^2\cdot n_e}{n_b}}\right)^{\frac{k(k-1)}{2}}. \qquad (2.94)$$

Перевага алгоритму-розрізнювача визначається як абсолютна різниця ймовірностей (4.6) і (4.7):

$$Adv_{\eta^*}(\vartheta, \sigma_{2n}) =$$

$$= \left| P_1\left(\eta_*(f(x_1),..., f(x_k)) = 1 : f \in_R \vartheta^{(3)}\right) - P_1^*\left(\eta_*(f(x_1),..., f(x_k)) = 1 : f \in_R \sigma_{2n}\right) \right| =$$

$$= \left| 1 - \left(1 - 2^{-\frac{n_c^2 \cdot n_e}{n_b}}\right)^{\frac{k(k-1)}{2}} - 1 + \left(1 - 2^{-\frac{n_c^2 \cdot n_e}{n_b}}\right)^{\frac{k(k-1)}{2}} \right| = 0.$$

Доказ закінчено.

Наслідок 2.1 (нерозрізненість 3-раундової SPN-структури і випадкової перестановки). Для 3-раундової SPN-структури не існує такого алгоритму-розрізнювача, який би з певною ймовірністю, відмінною від нуля, зміг би відрізнити таку структуру від випадкової перестановки.

На підставі вищенаведеного можна зробити наступні висновки:

1. Виконано аналіз 2-раундової SPN-структури і доведена теорема про максимально можливої ймовірності розрізнення такої конструкції.

2. Був побудований алгоритм-розрізнювач для 2-раундової SPN-структури, який з певною ймовірністю може відрізнити дану структуру від випадкової перестановки. Було показано, що існує деяка оптимальна кількість вхідних даних, при якому ймовірність розрізнення таким алгоритмом максимальна і складає близько $Adv(2rSPN) = 0.6$.

3. Проведено аналіз 3-раундової SPN-структури. Було доведено, що дану структуру неможливо відрізнити від випадкової перестановки з ймовірністю, відмінною від нуля.

2.5 Порівняння ефективності високорівневих конструкцій блокових шифрів

Розглянуті в попередніх розділах високорівневі конструкції ґрунтуються на двох різних підходах до побудови блокових шифрів:

– пряме бієктивне відображення оброблюваних даних на кожному раунді (SPN-структура);

– гамування оброблюваних даних з виходом деякої випадкової функції (ланцюг Фейстеля і схема Лей-Мессі).

Як зазначалося раніше, порівняння різних високорівневих конструкцій доцільно проводити через оцінку складності їх розрізнення з випадковою функцією або перестановкою. Показником ефективності тієї чи іншої конструкції є ймовірність її відрізнення від випадкової функції або перестановки. Порівнюючи такі ймовірності для різних високорівневих конструкцій можна зробити висновок щодо їх ефективності.

2.5.1 Порівняння ефективності ланцюга Фейстеля і схеми Лей-Мессі

Для високорівневих конструкцій побудованих на основі гамування з раундовою функцією (ланцюг Фейстеля і схема Лей-Мессі) оцінка ефективності може бути отримана як на основі порівняння максимальних теоретичних ймовірностей розрізнення, так і на основі порівняння конкретних алгоритмів-розрізнювачів.

Теоретично максимальна ймовірність розрізнення ланцюга Фейстеля з випадковими функціями в якості раундових перетворень оцінюється як

$$Adv_{\eta^*}(\psi, F_{2n}) \le 1 - \prod_{i=0}^{k-2}\left(1 - \frac{1}{2^n - i}\right)^{2(k-(i+1))} \approx 1 - \left(1 - \frac{1}{2^n}\right)^{k(k-1)}$$ (лема 2.3).

Відповідно, максимальна ймовірність розрізнення схеми Лей-Мессі з випадковими функціями в якості раундових перетворень оцінюється як

$$Adv_{\eta^*}(\varsigma, F_{2n}) \le 1 - \left(\frac{2^n - 2}{2^n - 1}\right)^{\frac{k(k-1)}{2}}$$ (лема 3.1).

Таким чином, обчисливши різницю цих двох величин можна зробити висновок щодо ефективності розглянутих конструкцій:

$$Adv_{\eta^*}(\psi,\varsigma) = Adv_{\eta^*}(\psi,F_{2n}) - Adv_{\eta^*}(\varsigma,F_{2n}) =$$
$$= \left(\frac{2^n-2}{2^n-1}\right)^{\frac{k(k-1)}{2}} - \left(1-\frac{1}{2^n}\right)^{k(k-1)} . \qquad (2.95)$$

Якщо $Adv_{\eta^*}(\psi,\varsigma) > 0$, то схема Лей-Мессі більш ефективна, ніж ланцюг Фейстеля, оскільки ймовірність її відрізнення від випадкової функції менше, а це значить, що дана конструкція краще приховує свою внутрішню структуру. Якщо ж $Adv_{\eta^*}(\psi,\varsigma) < 0$, то тоді ланцюг Фейстеля виявляється ефективнішим.

На рис. 2.23 наведено графік залежності різниці максимальних ймовірностей розрізнення для ланцюга Фейстеля і схеми Лей-Мессі в залежності від кількості вхідних текстів для блоку розміром $n = 8$. Аналогічний графік для блоку розміром $n = 16$ зображений на рис. 2.24. Для побудови даних графіків використовувалася формула (2.95).

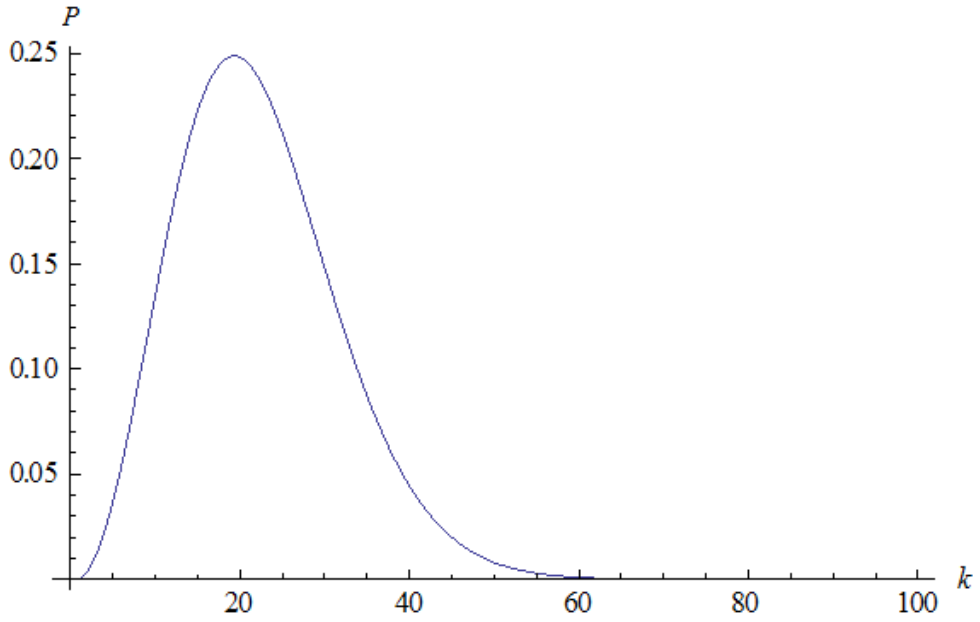


Рисунок 2.23 – Графік залежності різниці максимальних ймовірностей розрізнення ланцюга Фейстеля і схеми Лей-Мессі в залежності від числа запитів для блоку $n = 8$

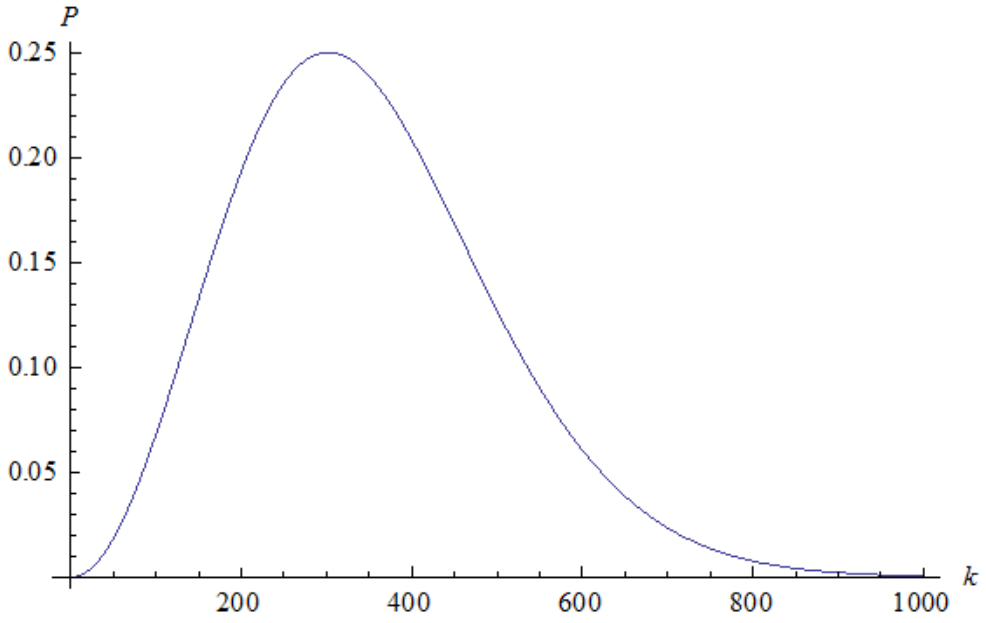


Рисунок 2.24 – Графік залежності різниці максимальних ймовірностей розрізнення ланцюга Фейстеля і схеми Лей-Мессі в залежності від числа запитів для блоку $n = 16$

Таким чином, видно що $Adv_{\eta^*}(\psi, \varsigma) > 0$, звідки випливає, що схема Лей-Мессі має більшу ефективність, ніж ланцюг Фейстеля.

З графіків на рис. 2.23 і 2.24 видно, що суттєву перевагу схеми Лей-Мессі досягається на певній кількості вхідних даних, близькій до $\sqrt{2^n}$. При збільшенні кількості вхідних даних відмінність обох конструкцій від випадкової функції стає неможливою, що робить їх показники еквівалентними.

Якщо порівнювати не максимально можливу вірогідність розрізнення, а конкретні алгоритми-розрізнювачі, то перевага схеми Лей-Мессі виражається ще більш виразно.

Для 3-раундового ланцюга Фейстеля на основі випадкових функцій кращий алгоритм-розрізнювач має наступну перевагу (алгоритм-розрізнювач №1, формула 2.15):

$$Adv_{\eta_1}(\psi, \sigma_{2n}) \le \left| \left(1 - \frac{1}{2^n}\right)^{\frac{k(k-1)}{2}} - \left(1 - \frac{1}{2^{2n} - 1}\right)^{\frac{k(k-1)}{2}} \right|.$$

Для 3-раундової схеми Лей-Мессі на основі випадкових функцій кращий алгоритм-разнювач має наступну перевагу (лема 3.2):

$$Adv_{\eta_4}(\varsigma,\sigma_{2n}) \le \left| \left(1-\frac{2^n}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}} - \left(1-\frac{2^{n+1}-1}{2^{2n}}\right)^{\frac{k(k-1)}{2}} \right|.$$

Звідси можна знайти різницю цих двох виразів і оцінити яка з високорівневих конструкцій є більш ефективною. Для цього можна використовувати такий вираз:

$$Adv_{\eta-opt}(\psi,\varsigma) =$$

$$= \left| \left(1-\frac{1}{2^n}\right)^{\frac{k(k-1)}{2}} - \left(1-\frac{1}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}} \right| - \left| \left(1-\frac{2^n}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}} - \left(1-\frac{2^{n+1}-1}{2^{2n}}\right)^{\frac{k(k-1)}{2}} \right|. \quad (5.2)$$

Аналогічно попередньому випадку, якщо $Adv_{\eta-opt}(\psi,\varsigma) > 0$, то схема Лей-Мессі більш ефективна, ніж ланцюг Фейстеля.

На рис. 2.25 наведено графік залежності різниці ймовірностей розрізнення ланцюга Фейстеля і схеми Лей-Мессі в залежності від кількості вхідних текстів для блоку розміром $n = 8$. Аналогічний графік для блоку розміром $n = 16$ зображений на рис. 2.26. для побудови даних графіків використовувалася формула (2.96).

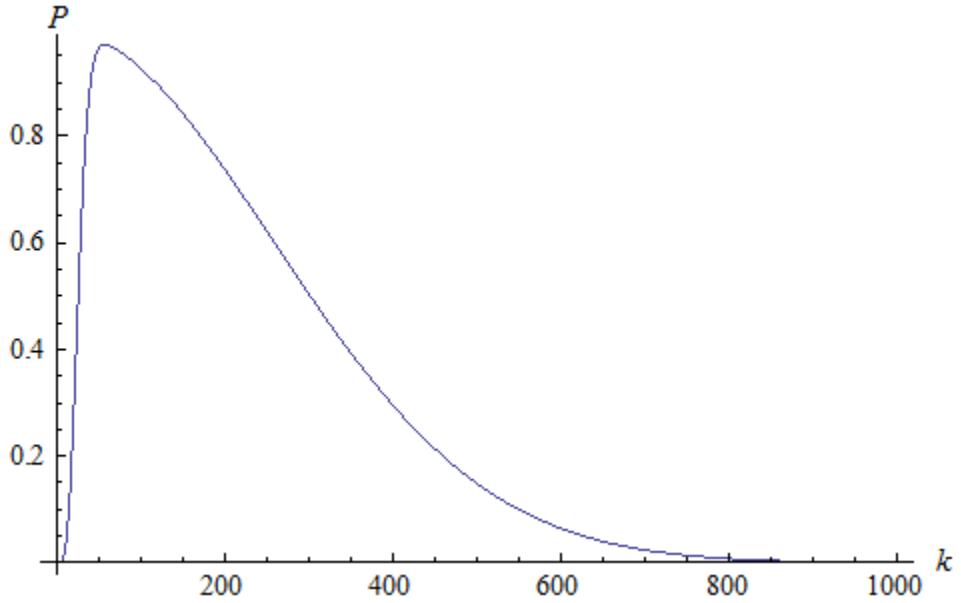


Рисунок 2.25 – Графік залежності різниці переваги розрізнення ланцюга Фейстеля і схеми Лей-Мессі в залежності від числа запитів при використанні випадкових функцій в раундовому перетворенні для кращих алгоритмів-розрізнювачів ($n = 8$)

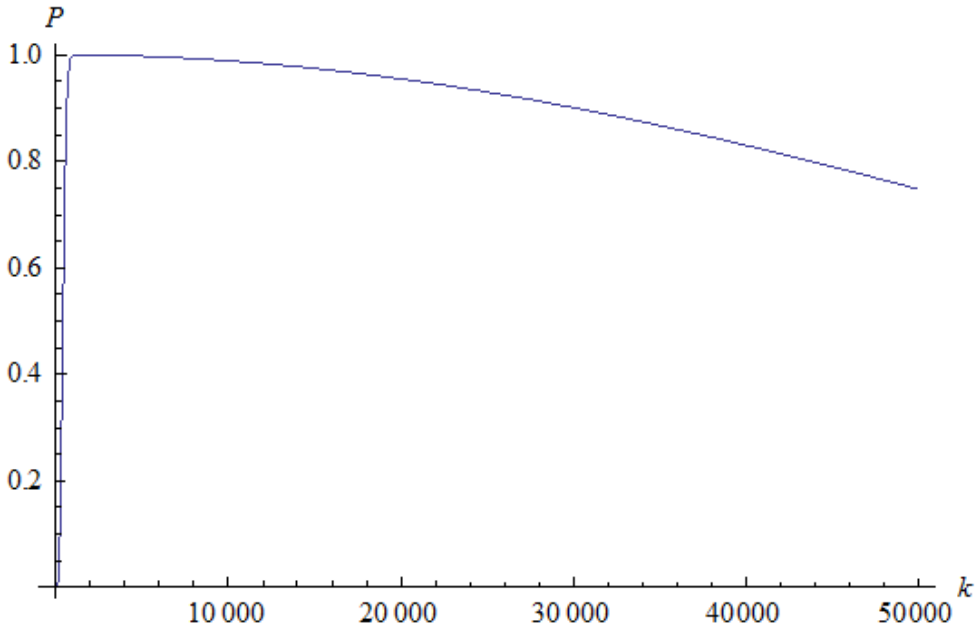


Рисунок 2.26 – Графік залежності різниці переваги розрізнення ланцюга Фейстеля і схеми Лей-Мессі в залежності від числа запитів при використанні випадкових функцій в раундовому перетворенні для кращих алгоритмів-разрізнювачів ($n = 16$)

З графіків на рис. 2.25 і 2.26 видно, що $Adv_{\eta-opt}(\psi,\varsigma)>0$, звідки випливає, що схема Лей-Мессі є більш ефективною ніж ланцюг Фейстеля при застосуванні до них конкретних алгоритмів-розрізнювачів.

З графіків на рис. 2.25 і 2.26 видно, що максимальна перевага досягається на певній кількості запитів. При збільшенні числа запитів відмінність обох конструкцій від випадкової функції стає неможливою, що робить їх еквівалентними.

Переважна більшість сучасних алгоритмів шифрування, які засновані на ланцюгу Фейстеля або схемою Лей-Мессі в якості раундового перетворення мають бієктивне перетворення, а не випадкову функцію.

Доцільно порівняти показники даних конструкцій при використанні саме бієктивних перетворень.

Для 3-раундового ланцюга Фейстеля заснованого на випадкових перестановках в якості раундового перетворення найкращий алгоритм-розрізнювач має перевагу $Adv_{\eta_1}(\psi\psi_{,2n})\le 1-(1-\frac{1}{2^n})^{\frac{k(k-1)}{2}}$ (формула 2.23).

Аналогічно, для 3-раундової схеми Лей-Мессі найкращий алгоритм-розрізнювач має перевагу, яку можна отримати за допомогою виразу $Adv_{\eta_7}(\varsigma,\sigma_{2n})\le\left|\left(1-\frac{2^n}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}}-\left(1-\frac{1}{2^n-1}\right)^{k(k-1)}\right|$ (лема 3.3).

Для оцінки двох конструкцій між собою використовується такий вираз:

$$Adv_{\eta p-opt}(\psi,\varsigma)=1-(1-\frac{1}{2^n})^{\frac{k(k-1)}{2}}-\left|\left(1-\frac{2^n}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}}-\left(1-\frac{1}{2^n-1}\right)^{k(k-1)}\right|. \quad (2.97)$$

Графіки залежності різниці переваг 3-раундового ланцюга Фейстеля і схеми Лей-Мессі з випадковими перестановками в якості раундового перетворення приведені на рис. 2.27 ($n=8$) і рис 2.28 ($n=16$).

Як видно з графіків на рис. 2.27 і рис. 2.28 $Adv_{пр-opt}(\psi,\varsigma)>0$, звідки випливає, що 3-раундова схема Лей-Мессі з випадковими перестановками в якості раундового перетворення є більш ефективною конструкцією, ніж ланцюг Фейстеля аналогічної конфігурації. На певній кількості запитів схема Лей-Мессі стає абсолютно невиразною, в той час як ланцюг Фейстеля може бути визначена з ймовірністю в 1.

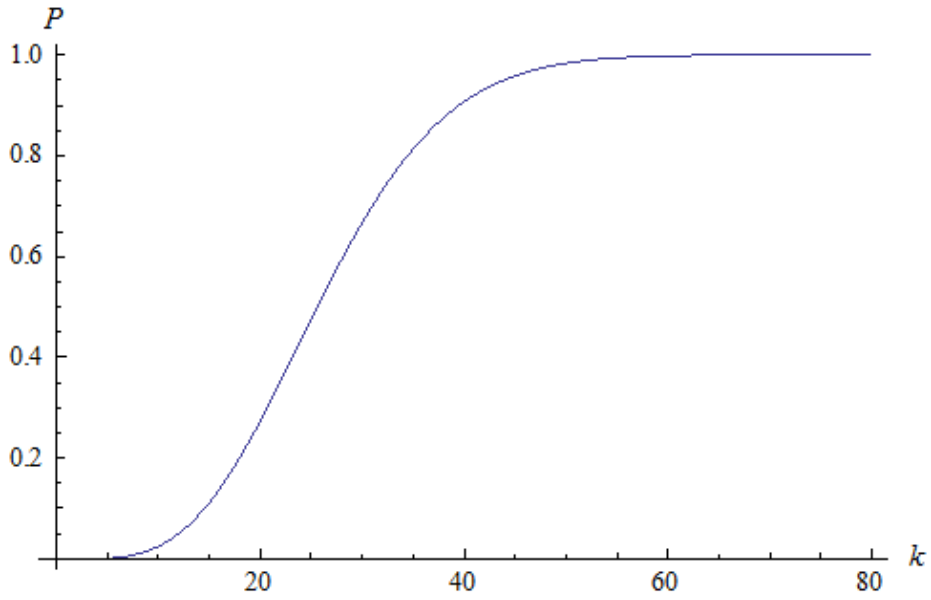


Рисунок 2.27 – Графік залежності різниці переваг розрізнення 3-раундового ланцюга Фейстеля і схеми Лей-Мессі в залежності від числа запитів при використанні випадкових перестановок в раундовому перетворенні ($n=8$)

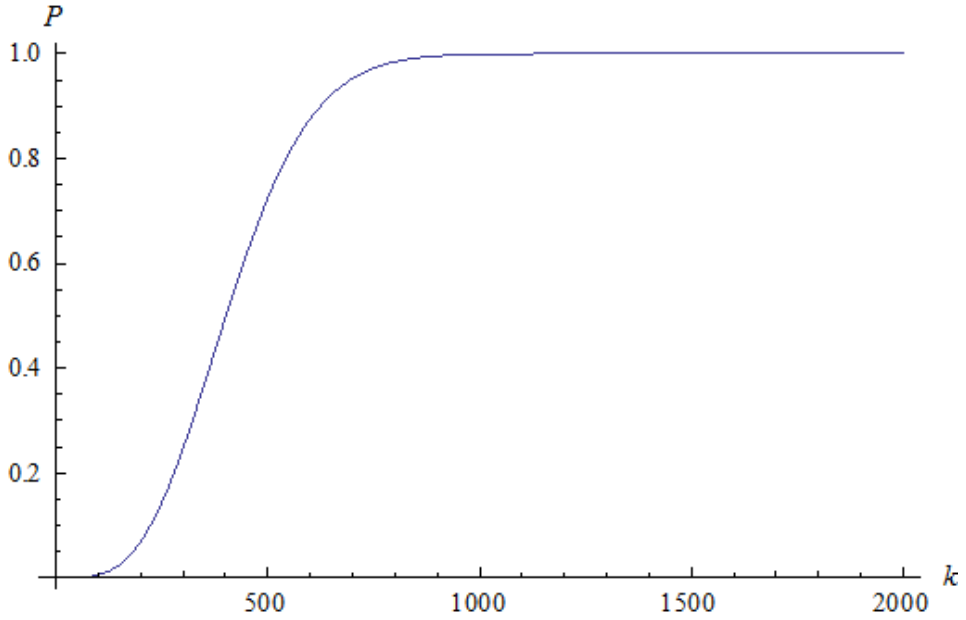


Рисунок 2.28 – Графік залежності різниці переваг розрізнення 3-раундового ланцюга Фейстеля і схеми Лей-Мессі в залежності від числа запитів при використанні випадкових перестановок в раундовому перетворенні ($n = 16$)

Слід звернути увагу на те, що ймовірність розрізнення 3-раундової схеми Лей-Мессі з випадковими перестановками практично дорівнює ймовірності розрізнення 4-раундової схеми Фейстеля з випадковими функціями в раундовому перетворенні.

Також слід зазначити, що дані 2 конструкції мають різну складність реалізації. Ланцюг Фейстеля вимагає одну групову операцію (як правило, додавання по модулю 2) на один раунд шифрування, в той же час схема Лей-Мессі вимагає 3 групових операції, а також додаткові витрати на ортоморфне перетворення (в найпростішому випадку – це ще одна групова операція) .

Як можна бачити, складність реалізації схеми Лей-Мессі є вищою, в порівнянні з ланцюгом Фейстеля. Дана різниця в складності, як правило, незначна в порівнянні зі складністю реалізації раундового перетворення, проте її все одно варто врахувати.

З усього вищесказаного можна зробити висновок, що при малій кількості раундів шифрувального перетворення схема Лей-Мессі є більш ефективною високорівневою конструкцією, ніж ланцюг Фейстеля. Даний

висновок підтверджується як для конструкцій з випадковими функціями як раундового перетворення, так і для конструкцій з випадковими перестановками. На більшій кількості раундів перетворення обидві схеми стають невиразними, тому подальше порівняння їх ефективності описаними методами важко здійсненне.

Також слід зазначити, що ланцюг Фейстеля і схема Лей-Мессі допускають використання несюр'єктивних перетворень в якості раундових функцій. Це дозволяє впроваджувати «закладки» в шифрувальне перетворення знижуючи його криптостійкість.

На даний момент, побудова шифрів, основою яких є ланцюг Фейстеля або схема Лей-Мессі з небієктивною (але при цьому сюр'єктивною) раундовою функцією, є невирішеним завданням. Наприклад, шифр DES, що побудований за такою схемою, має ряд недоліків. Зокрема, його диференціальні характеристики залежать від обраного ключа шифрування, що полегшує криптоаналіз даного шифру. Інші сучасні алгоритми використовують виключно бієктивне перетворення в раундових функціях.

2.5.2 Порівняння ефективності SPN-структури і ланцюга Фейстеля

Побудова SPN-структури можлива тільки на основі бієктивних раундових перетворень. У зв'язку з цим, при порівнянні доцільно орієнтуватися на конструкції з випадковими перестановками в якості раундових функцій, оскільки перестановки є бієктивними перетвореннями.

Порівняння високорівневих конструкцій на основі гамування (таких як схема Лей-Мессі і ланцюг Фейстеля) і SPN-структури є непростим завданням, оскільки саме по собі розрізнення SPN-структури є нетривіальним.

Для SPN-структури був знайдений тільки один алгоритм-розрізнювач для 2-раундового перетворення. Перевага даного алгоритму-розрізнювача можна знайти за допомогою наступного виразу (теорема 2.1):

$$Adv_{\eta^*}(\vartheta,\sigma_{2n}) \approx \left| 2^{-\frac{n_c \cdot n_e \cdot k \cdot (k-1)}{2}} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{\frac{n_b k(k-1)}{2}} - \left(1 - 2^{-\frac{n_c \cdot n_e}{n_b}} \right)^{\frac{k(k-1)(n_b-1)}{2}} \right| .$$

Для 3-раундової ланцюга Фейстеля заснованого на випадкових перестановках в якості раундового перетворення найкращий алгоритм-розрізнювач має перевагу $\mathrm{Adv}_{\eta_1}(\psi,\sigma_{2n}) \approx 1-(1-\frac{1}{2^n})^{\frac{k(k-1)}{2}}$ (формула 2.23).

Для оцінки ланцюга Фейстеля і SPN-структури між собою слід скористатися таким виразом:

$$\mathrm{Adv}_{\eta p-opt}(\psi,\vartheta) = \left| 1-(1-\frac{1}{2^n})^{\frac{k(k-1)}{2}} \right| - \left| 2^{-\frac{n_c \cdot n_e \cdot k \cdot (k-1)}{2}} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{\frac{n_b k(k-1)}{2}} - \left(1 - 2^{-\frac{n_c \cdot n_e}{n_b}} \right)^{\frac{k(k-1)(n_b-1)}{2}} \right| . \quad (2.98)$$

Якщо $Adv_{\eta^*}(\psi,\vartheta) > 0$, то SPN-структура більш ефективна, ніж ланцюг Фейстеля, оскільки ймовірність її відрізнення від випадкової перестановки менше, а це значить, що дана конструкція краще приховує свою внутрішню структуру. Якщо ж $Adv_{\eta^*}(\psi,\varsigma) < 0$, то тоді ланцюг Фейстеля виявляється ефективнішим.

На рис. 2.29 наведено графік залежності різниці максимальних ймовірностей розрізнення для ланцюга Фейстеля і SPN-структури в залежності від кількості вхідних текстів для блоку розміром $n = 16$. Для побудови даного графіка використовувалася формула (2.98).

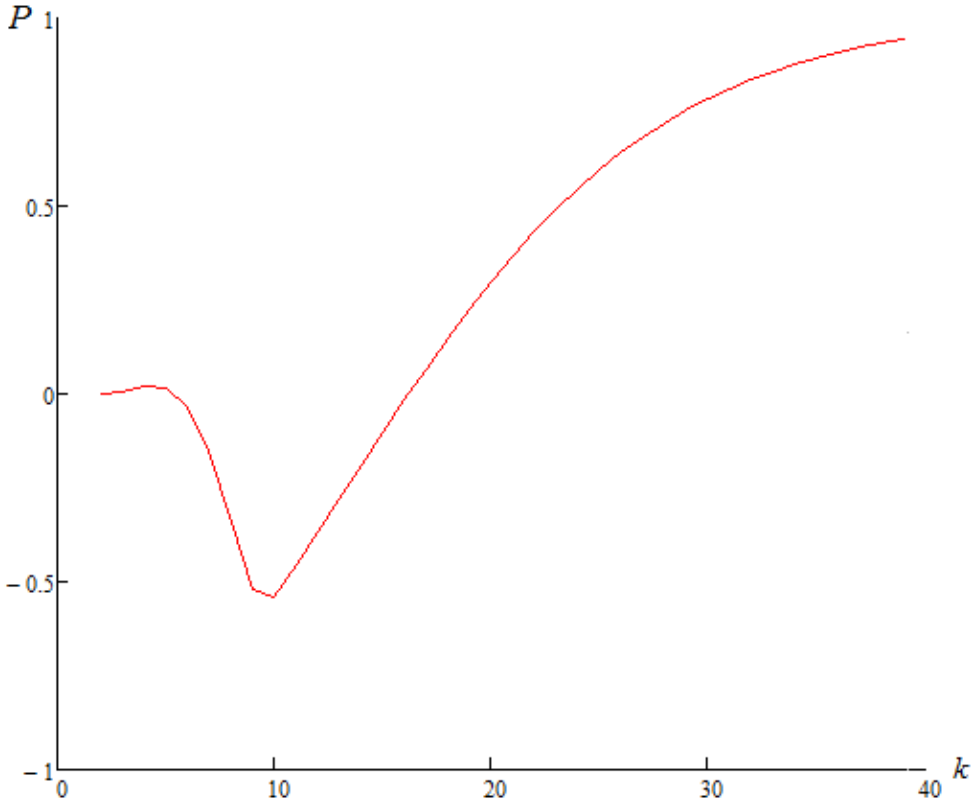


Рисунок 2.29 – Графік залежності різниці переваг розрізнення 3-раундового ланцюга Фейстеля і 2-раундової SPN-стуктури в залежності від числа запитів при використанні випадкових перестановок в раундовому перетворенні ($n = 16$)

Як видно з графіка на рис. 2.29 $Adv_{\eta-opt}(\psi,\varsigma) < 0$ на певній кількості запитів. Це означає, що при певній кількості запитів розрізнити SPN-структуру буде легше, ніж ланцюг Фейстеля. Однак в цілому на більшій частині графіка $Adv_{\eta-opt}(\psi,\varsigma) > 0$, що свідчить про те, що SPN-структура більш ефективна, ніж ланцюг Фейстеля. Також варто враховувати те, що порівняння проводилося між 3-раундової ланцюгом Фейстеля і 2-раундової SPN-структурою, оскільки 3-раундова SPN-структура є взагалі не помітною, а розрізнення 2-раундового ланцюга Фейстеля є зовсім тривіальним завданням.

2.5.3 Порівняння ефективності SPN-структури і схеми Лей-Мессі

Використовуючи аналогічний підхід, проведемо порівняння між схемою Лей-Мессі і SPN-структурою. Для 3-раундової схеми Лей-Мессі найкращий алгоритм-розрізнювач має перевагу, яку можна отримати за допомогою виразу $Adv_{\eta_7}(\varsigma,\sigma_{2n}) \le \left| \left(1-\frac{2^n}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}} - \left(1-\frac{1}{2^n-1}\right)^{k(k-1)} \right|$ (лема 3.3).

Для оцінки двох конструкцій між собою використовується такий вираз:

$$Adv_{\eta p-opt}(\varsigma,\vartheta) = \left| \left(1-\frac{2^n}{2^{2n}-1}\right)^{\frac{k(k-1)}{2}} - \left(1-\frac{1}{2^n-1}\right)^{k(k-1)} \right| - \left| 2^{-\frac{n_c \cdot n_e \cdot k \cdot (k-1)}{2}} \cdot \left(2^{\frac{n_c \cdot n_e}{n_b}} - 1 \right)^{\frac{n_b k(k-1)}{2}} - \left(1-2^{-\frac{n_c \cdot n_e}{n_b}} \right)^{\frac{k(k-1)(n_b-1)}{2}} \right|. \quad (2.99)$$

Якщо $Adv_{\eta^*}(\varsigma,\vartheta) > 0$, то SPN-структура більш ефективна, ніж схема Лей-Мессі, оскільки ймовірність її відрізнення від випадкової перестановки менша, а це значить, що дана конструкція краще приховує свою внутрішню структуру. Якщо ж $Adv_{\eta^*}(\psi,\varsigma) < 0$, то тоді схема Лей-Мессі виявляється ефективнішою.

На рис. 2.30 наведено графік залежності різниці максимальних ймовірностей розрізнення для схеми Лей-Мессі і SPN-структури в залежності від кількості вхідних текстів для блоку розміром $n = 16$. Для побудови даного графіка використовувалася формула (2.99).

Як видно з графіка на рис. 2.30 $Adv_{\eta-opt}(\psi,\varsigma) < 0$ на певній кількості запитів. Це означає, що розрізнення SPN-структури відбувається більш ефективно, ніж розрізнення схеми Лей-Мессі. Однак як і в попередньому випадку, істотним фактором є те, що ми порівнюємо 3-циклову схему Лей-

Мессі з 2-цикловою SPN-структурою. Якщо порівнювати ці структури для однакової кількості раундів, то SPN-структура має більшу ефективність.

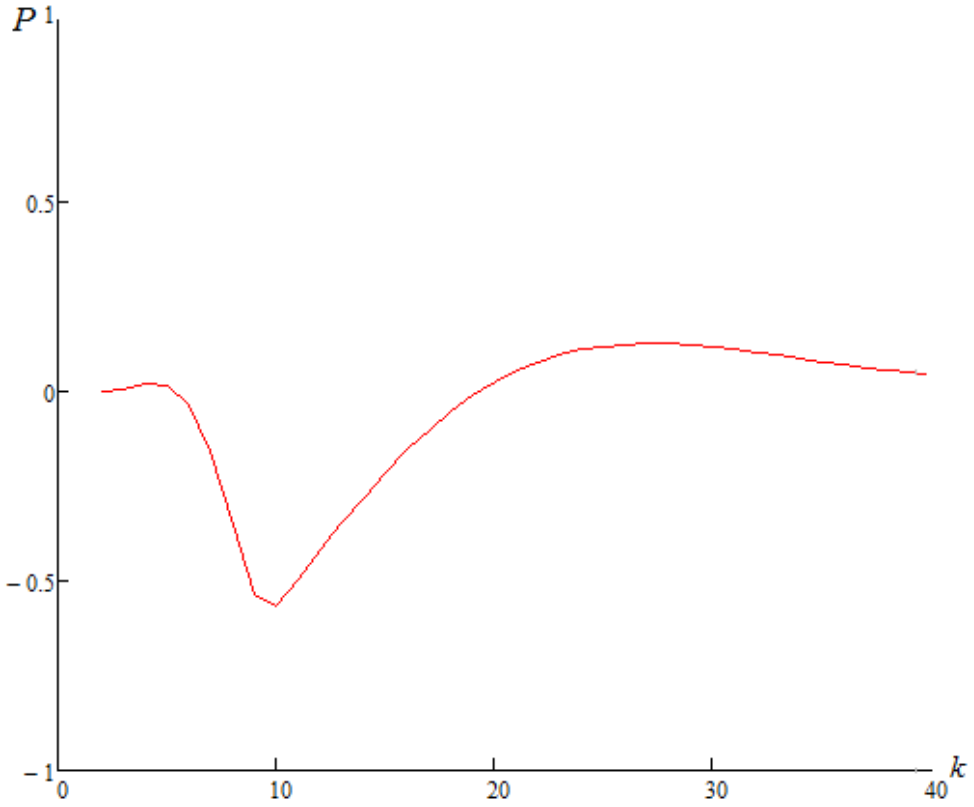


Рисунок 2.30 – Графік залежності різниці переваг розрізнення 3-раундової схеми Лей-Мессі і 2-раундової SPN-структури в залежності від числа запитів при використанні випадкових перестановок в раундовому перетворенні ($n=16$)

Таким чином, з огляду на всі проведені порівняння можна зробити висновок, що SPN-структура є найкращою високорівневою конструкцією для побудови сучасного блочного симетричного шифру.

З урахуванням вищенаведеного можна зробити наступні висновки:

1. Проведена порівняльна оцінка між максимально можливими ймовірностями розрізнення ланцюга Фейстеля і схеми Лей-Мессі. Було встановлено, що схема Лей-Мессі є більш ефективною, ніж ланцюг Фейстеля при теоретичних розрахунках максимально можливої ймовірності розрізнення.

2. Для 3-раундового ланцюга Фейстеля і 3-раундової схеми Лей-Мессі з випадковими функціями як раундового перетворення проведена оцінка найкращих алгоритмів-розрізнювачів. Встановлено, що алгоритм-розрізнювач для ланцюга Фейстеля дає більшу ймовірність розрізнення, ніж аналогічний для схеми Лей-Мессі. Це свідчить про те, що схема Лей-Мессі є більш ефективною, ніж ланцюг Фейстеля.

3. Оскільки більшість сучасних шифрів будується на основі бієктивних раундових перетворень, то для 3-раундового ланцюга Фейстеля і схеми Лей-Мессі з випадковою перестановкою в якості раундового перетворення також проведена порівняльна оцінка найкращих алгоритмів-розрізнювачів. Як і в випадку з випадковими функціями кращий результат показала схема Лей-Мессі.

4. Проведено порівняльну оцінку між 3-раундової ланцюгом Фейстеля і 2-раундової SPN-структурою. На деякій кількості вхідних запитів розрізнення SPN-структури відбувається більш ефективно, проте в загальному випадку ланцюг Фейстеля все ж краще помітний, що свідчить про те, що він має гірші властивості, ніж SPN-структура.

5. В ході порівняння 3-раундової схеми Лей-Мессі і 2-раундової SPN-структури було встановлено, що схема Лей-Мессі більш ефективно приховує свою структуру. Однак, з огляду на що 3-раундова SPN-структура взагалі неможливо розрізнити, то все ж варто вважати її більш ефективною, ніж схема Лей-Мессі.

6. В цілому, серед всіх трьох розглянутих високорівневих конструкцій найбільш кращими властивостями володіє SPN-структура. Її найскладніше відрізнити від випадкової перестановки, що свідчить про її хороші криптографічні властивості.

3 ВДОСКОНАЛЕННЯ МЕТОДІВ СИНТЕЗУ ЕФЕКТИВНИХ В РЕАЛІЗАЦІЇ СХЕМ ГЕНЕРАЦІЇ ЦИКЛОВИХ КЛЮЧІВ СИМЕТРИЧНИХ БЛОКОВИХ ШИФРІВ ДЛЯ ЗАХИСТУ ВІД АТАК НА ЗВ’ЯЗАНИХ КЛЮЧАХ

Основною метою даного розділу є розробка методу оцінки стійкості SPN-подібного блочного шифру проти атаки на пов'язаних ключах. Дана робота була актуальною в зв'язку з тим, що до недавнього часу в Україні проводився конкурсний відбір на національний стандарт блочного симетричного шифрування, в ході якого був відібраний алгоритм Калина [25]. Даний шифр має SPN-подібну структуру в чому повторює структуру всесвітньо відомого алгоритму AES [62]. Однак, як відомо на алгоритм AES були знайдені теоретичні атаки [64-67], які вживають уразливості в схемі розгортання ключа. Оскільки Калина багато в чому схожа з AES, то Першочерговим завданням при аналізі даного шифру стала оцінка його захищеності проти відомих атак, яким був підданий шифр AES і, зокрема, проти атаки на пов'язаних ключах [65-72]. Аналізу даного алгоритму присвячено вже безліч досліджень [73-80], однак метою даної роботи є розробка методу оцінки проти атаки на пов'язаних ключах. Даний метод в подальшому можна буде адаптувати і для інших SPN-подібних шифрів.

Алгоритм Калина став офіційним стандартом блокового симетричного шифрування України ДСТУ 7624:2014 [81]. У наступному підрозділі будуть коротко описані основні компоненти алгоритму Калина, оскільки на цьому буде базуватися весь подальший виклад. Детально ознайомитися з шифром можна в роботі [25].

3.1 Опис основних компонентів алгоритму шифрування Калина

Перш ніж перейти до детального опису методу оцінки стійкості, доцільно привести загальний опис шифру. Основна відмінність алгоритму

Калина від AES полягає в повністю переробленої схемою розгортання ключа, в якій власне і була знайдена уразливість. В даному розділі наводиться короткий опис досліджуваного шифру, а також акцентується увага на відмінностях даного алгоритму.

3.1.1 Загальні параметри

Даний алгоритм може використовуватися з різними варіантами вхідних даних. Блок вхідних даних може становити 128, 256 або 512 біт. Довжина ключа може становити також 128, 256 або 512 біт.

Стан шифру представляється у вигляді матриці розміром $8 * N_b$ байт. Матриці складається з N_b колонок, в кожній по 8 байт. Стан шифру для 128-бітної версії наведено у табл. 3.1.

Таблиця 3.1 – Байтове подання для 128-бітної версії

0	8
1	9
2	10
3	11
4	12
5	13
6	14
7	15

Числами представлена вхідна послідовність байтів (ключа, стану шифру і т.п.). Подання у вигляді колонок по 8 байт використовується для внутрішніх перетворень. Кожна клітинка містить 1 байт стану.

В роботі розглядається тільки 128 бітна версія шифру, тому надалі мова буде йти тільки про цю версію, навіть якщо це і не згадується явно.

3.1.2 Базові перетворення

В алгоритмі Калина, аналогічно алгоритму AES, використовується 4 базових перетворення: додавання з ключем, підстановка байтів, зсув рядків і перемішування в стовпцях. На цих операціях засновані всі перетворення в шифрі. Розглянемо кожну з операцій детальніше.

Додавання з ключем. В алгоритмі використовується два різні способи складання з ключем. Один має на увазі складання по модулю 2^{64}, а другий – класичний XOR (додавання по модулю 2).

Додавання за модулем 2^{64} відбувається наступним чином:

$$b_i = a_i + k_i (\mathrm{mod} 2^{64}),$$

де $b_i = b_{0,i} \cdot 2^{0 \cdot 8} + b_{1,i} \cdot 2^{1 \cdot 8} + b_{2,i} \cdot 2^{2 \cdot 8} + b_{3,i} \cdot 2^{3 \cdot 8} + b_{4,i} \cdot 2^{4 \cdot 8} + b_{5,i} \cdot 2^{5 \cdot 8} + b_{6,i} \cdot 2^{6 \cdot 8} + b_{7,i} \cdot 2^{7 \cdot 8}$,

$a_i = a_{0,i} \cdot 2^{0 \cdot 8} + a_{1,i} \cdot 2^{1 \cdot 8} + a_{2,i} \cdot 2^{2 \cdot 8} + a_{3,i} \cdot 2^{3 \cdot 8} + a_{4,i} \cdot 2^{4 \cdot 8} + a_{5,i} \cdot 2^{5 \cdot 8} + a_{6,i} \cdot 2^{6 \cdot 8} + a_{7,i} \cdot 2^{7 \cdot 8}$

$k_i = k_{0,i} \cdot 2^{0 \cdot 8} + k_{1,i} \cdot 2^{1 \cdot 8} + k_{2,i} \cdot 2^{2 \cdot 8} + k_{3,i} \cdot 2^{3 \cdot 8} + k_{4,i} \cdot 2^{4 \cdot 8} + k_{5,i} \cdot 2^{5 \cdot 8} + k_{6,i} \cdot 2^{6 \cdot 8} + k_{7,i} \cdot 2^{7 \cdot 8}$,

$0 \le i < N_b$ – це номер стовпця.

Як видно, додавання по модулю 2^{64} відбувається за стовпцями і використовується для початкового забілювання, а також в останньому циклі шифрування.

Підстановка байтів. Дане перетворення виконує заміну кожного байта поточного стану відповідно до заданої таблиці підстановки. У шифрі застосовуються 4 різні підстановки «байт-в-байт». Вибір конкретної таблиці залежить від того в якому рядку знаходиться байт. Для 1 і 5 рядка – таблиця 1, для 2 і 6 рядка – таблиця 2, для 3 і 7 рядка – таблиця 3, для 4 і 8 рядків – таблиця 4.

Зсув рядків. В ході даного перетворення проводиться рівномірний розподіл байт кожного 64-бітового стовпця серед інших стовпців. Це досягається шляхом циклічного зсуву рядків вправо на різну кількість байт. Величини зрушень залежать від розміру блоку і представлені в таблиці 3.2.

Таблиця 3.2 – Величини зсуву рядків для різних розмірів блоку даних

	Величина зсуву, байти		
Номер строки	Довжина блоку – 128 біт	Довжина блоку – 256 біт	Довжина блоку – 512 біт
0	0	0	0
1	0	0	1
2	0	1	2
3	0	1	3
4	1	2	4
5	1	2	5
6	1	3	6
7	1	3	7

Зсув рядків для 128-бітної версії шифру наведений на рис. 3.1.

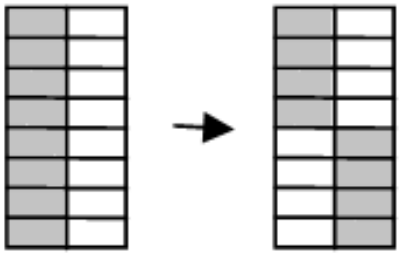


Рисунок 3.1 – Зсув у 128-бітному блоці

Перемішування в стовпцях. В ході даного перетворення відбувається обробка кожного стовпця окремо. Стовпець представляється у вигляді полінома і множиться на певний фіксований поліном у полі $GF(2^8)$. Ця операція також еквівалентна матричному множенню у полі $GF(2^8)$.

Важливою властивістю даного перетворення є те, що кількість активних (ненульових) байтів на вході і виході перетворення пов'язані наступною нерівністю:

$$N_{in} + N_{out} \geq 9. \qquad (3.1)$$

3.1.3 Схема розгортання ключа

Нехай K_M – це головний ключ шифрування блочного шифру і $(K_1, K_2, ..., K_m)$ – раундові ключі, згенеровані схемою розгортання ключа.

Шифр Калина має SPN-подібну структуру, яка описується наступним виразом:

$$Cipher_{K_M} = \prod_{i=1}^{N_r} \theta \circ \gamma \circ \sigma_{K_i} . \quad (3.2)$$

Умовні позначення:

σ_{K_i} – складання з раундовим ключем;

γ – нелінійний шар (підстановка байтів);

θ – лінійне перетворення (перестановка байтів, множення на матрицю);

N_r – кількість раундів блочного шифру.

Схема розгортання ключа складається з двох кроків:

1. Обчислення проміжного значення K_t заснованого на головному ключі шифрування K_M і деякій константі.

2. Обчислення раундових ключів $(K_1, K_2, ..., K_m)$, заснованих на головному ключі шифрування K_M, проміжному значенні K_t і деякій константі.

Проміжне значення K_t формується за допомогою наступного алгоритму (рисунок 3.2):

Як видно на рис. 3.2 формування K_t містить в собі ті ж перетворення, які застосовуються і в основному циклі шифрування – це значно спрощує реалізацію алгоритму, тому що базові операції всюди однакові.

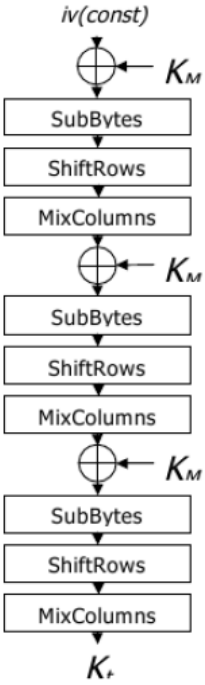


Рисунок 3.2 – Формування проміжного значення K_t

Формально формування проміжного значення може бути описано за допомогою таких висловів:

$$IM_{K_M} = \prod_{i=1}^{3} \theta \circ \gamma \circ \sigma_{K_M}, \tag{3.3}$$

$$K_t = IM_{K_M}(iv), \tag{3.4}$$

де $iv-$ це деяка константа, яка присутня для того, щоб прибрати симетрію, яка може існувати в ключі.

Генерація раундових ключів може бути описана за допомогою наступного виразу:

$$RK_{K_t}[K_M] = \sigma_{Kt+tmv_0} \circ \prod_{i=1}^{2} \theta \circ \gamma \circ \sigma_{K_t+tmv_i}, \tag{3.5}$$

де tmv_i – це константа, яка використовується для генерації раундових ключів.

Для кожного раунду значення tmv_i має бути унікальним, але відмінність між цими константами може бути досить простою (наприклад, звичайний зсув на кілька розрядів).

Загальний алгоритм схеми формування раундових ключів представлений на рис. 3.3.

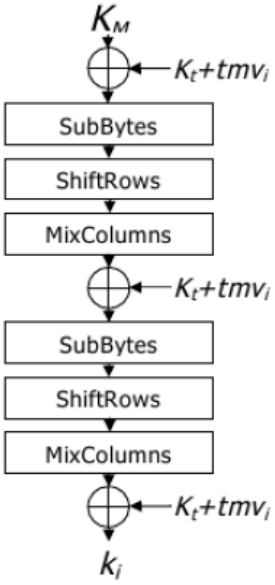


Рисунок 3.3 – Схема формування раундового ключа

Як видно ця схема дуже схожа на схему формування K_t, а також на схему шифрувального перетворення. Це значно спрощує реалізацію.

Така схема розгортання має такі властивості [82]:

1. Незворотність: маючи ключ шифрування можна дуже легко обчислити раундові ключі, але маючи один або кілька раундових ключів практично неможливо обчислити ключ шифрування.

2. Нелінійна залежність кожного біта раундового ключа і кожного біта ключа шифрування.

3. Хороші статистичні властивості (перевірено за допомогою статистичних тестів NIST STS).

4. Простота реалізації. Практично всі компоненти шифру засновані на одних і тих же перетвореннях.

3.1.4 Шифрувальне перетворення

Шифрувальне перетворення алгоритму Калина практично повторює перетворення, яке використовується в алгоритмі AES. Загальна схема для 128 бітової версії шифру представлена на рис. 3.4. Додавання з ключами K_0 і K_{10} проводиться по модулю 2^{64}. В інших випадках складання з ключем проводиться по модулю 2.

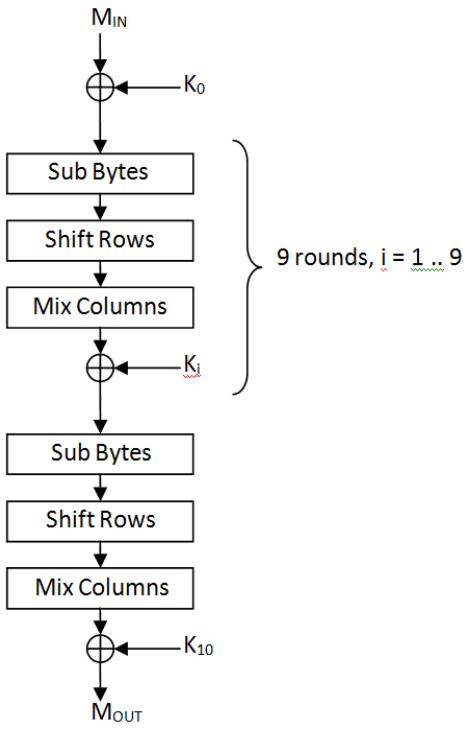


Рисунок 3.4 – Загальна схема шифрування алгоритму Калина

Як видно, даний шифр вдає із себе класичну SPN-структуру. Оскільки та ж структура використовується і в схемі розгортання ключа, то це значно спрощує програмну реалізацію алгоритму.

3.2 Обґрунтування стійкості проти атаки на пов'язаних ключах

Основним завданням, яке ставилося в дослідженнях, було знаходження способу оцінки стійкості досліджуваного алгоритму проти атаки на пов'язаних ключах. Існуючі методи оцінки мають занадто високу складність для того, щоб їх можна було застосувати на практиці, тому була потрібна розробка нового методу, який би дозволив вирішити поставлену задачу за прийнятний час.

3.2.1 Опис атаки на пов'язаних ключах

Суть атаки на пов'язаних ключах полягає в тому, щоб підбираючи деякі різниці для пари вхідних даних на вході шифрувального перетворення домогтися виникнення колізій, які будуть нівелювати поширення вхідної різниці. Така ситуація дозволить надалі відновити ключ з меншою складністю, ніж складність повного перебору. Приклад атаки для одного раунду шифрування алгоритмом AES продемонстрований на рис. 3.5 [66].

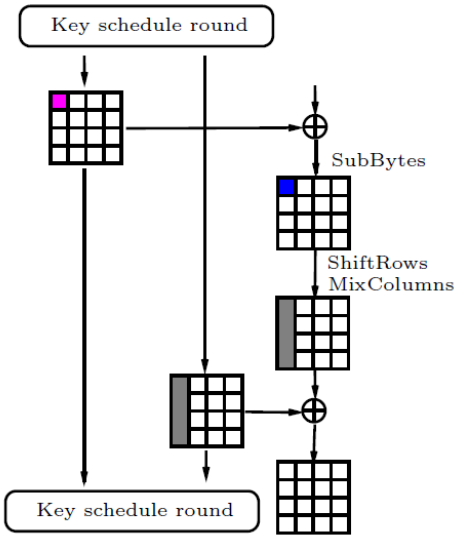


Рисунок 3.5 – Приклад атаки на пов'язаних ключах
для 1-го раунду шифру AES

На рис. 3.5 зображено приклад колізії для першого раунду шифру AES. Як видно на вхід схеми розгортання подані 2 ключа, які відрізняються першим байтом. Згідно зі схемою розгортання ключа дана вхідна різницю в

подальшому перетворюється в різниця для цілого стовпця. У той же час, ці ж ключі використовуються в основному шифрувальні перетворення. Якщо врахувати, що на вхід шифру ми подаємо однакові тексти, то в подальшому, різниця, внесена ключами після проходження таблиць підстановки, перетворитися в іншу різницю, яка з імовірністю 2^{-6} дасть таку ж різницю для першого стовпця після застосування перемішування стовпців. В результаті вийде ситуація, при якій вхідна різниця нівелюється. Така ситуація може призвести до розкриття ключа з меншою складністю, ніж складність повного перебору. Більш детально з атакою на пов'язаних ключах можна ознайомитися в роботі [66].

3.2.2 Запропонований метод оцінки стійкості проти атаки на пов'язаних ключах

Для того щоб довести стійкість алгоритму шифрування проти такого роду атак пропонується визначити найбільш правильну диференціальну характеристику і підрахувати скільки активних байтів задіяно при її побудові. Тобто яка кількість байтів, що мають ненульову різницю, проходить через таблиці підстановки. Якщо кількість активних байтів перевищує визначений поріг, то шифр можна вважати стійким проти такого роду атак. Граничне значення залежить від розміру блоку і розміру ключа шифрування.

Кращу диференціальну характеристику можна знайти шляхом повного автоматизованого перебору всіх можливих вхідних різниць. Як буде показано далі, застосовуючи певні техніки для 128-бітної версії шифру це можливо зробити за короткий час.

3.2.3 Алгоритм підрахунку активних байтів

Як вже було сказано, кращу диференціальну характеристику (з найменшою кількістю активних байтів) можна знайти шляхом перебору всіляких вхідних різниць і підрахунком кількості активних байтів після

схеми розгортання ключа і безпосередньо шифрування. Надалі всі міркування будуть ставитися до 128-бітної версії шифру. Для прикладу розглянемо одну з можливих диференціальних характеристик, яку можна отримати при формуванні проміжного значення K_t для схеми розгортання ключа. Загальна схема формування K_t зображена на рис. 3.6.

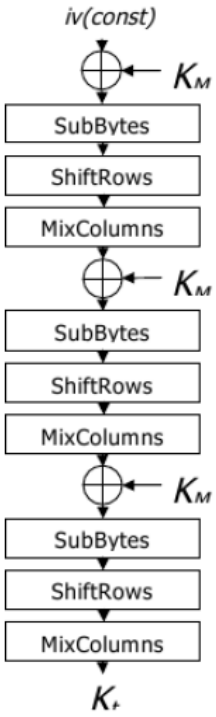


Рисунок 3.6 – Схема формування проміжного значення K_t

Як видно на вхід схеми формування подається деяка константа, а також головний ключ шифрування K_M. Відповідно до алгоритму атаки потрібно підбирати різницю між парою вхідних ключів шифрування і дивитися, як ця різниця в подальшому буде поширюватися. Чим менше поширення різниці, тим менше складність підбору ключів, які зв'яжуть вихідні тексти деяких рівнянням. Приклад побудови диференціальної характеристики для розглянутої схеми наведено на рис. 3.7.

Умовні позначення:

iv – різниця в векторах ініціалізації (завжди однакова і дорівнює нулю);

K_M – головний ключ шифрування;

SB – «sub-bytes» (заміна байтів за допомогою таблиць підстановок);

SR – «shift rows» (зсув рядків);

MC– «mix columns» (перемішування в стовпцях).

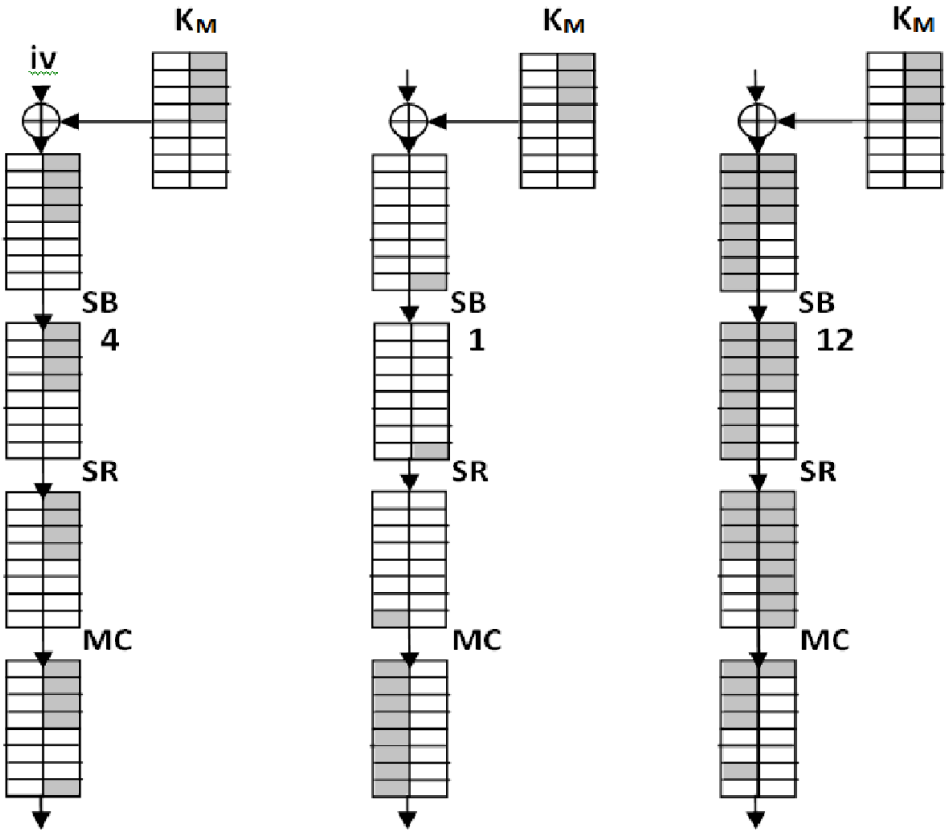


Рисунок 3.7 – Диференціальна характеристика для схеми формування проміжного значення K_t

У прикладі на рис. 3.7 розглянуто 3 раунди формування проміжного значення K_t. Вхідним параметром є різниця ключа шифрування K_M. Надалі можна спостерігати, як вхідна різниця поширюється при застосуванні різних перетворень. Алгоритм підраховує кількість активних байтів (таких, в яких різниця не дорівнює нулю) на вході перетворення «sub-bytes». В даному прикладі це значення дорівнює 4 на першому раунді, 1 – на другому і 12 – на третьому раунді, тобто в сумі виходить 17 активних байтів.

Слід зазначити, що в лінійних перетвореннях (зсув рядків і перемішування в стовпцях) робиться припущення краще для

криптоаналитика. Наприклад, передбачається, що різниця після перетворення «mix columns» буде такою, що при додаванні з ключем у другому раунді відбудеться колізія, і різниця в відповідних байтах нівелюється.

3.2.3.1 Загальний опис алгоритму підрахунку активних байтів

Більш простим варіантом алгоритму підрахунку активних байтів може стати версія з контролем не кожного байту, а кожної окремої колонки стану шифру. Як відомо, в розглянутому алгоритмі шифрування Калина поточний стан шифру представляється у вигляді колонок по 8 байт. Відповідно 128-бітна версія шифру має 2 колонки. При побудові алгоритму пошуку можна враховувати не кожен байт, а кількість активних байт у кожному стовпчику без урахування їх точного місця розташування. Такий підхід значно знизить обчислювальну складність алгоритму. Точність результатів також погіршитися в тому плані, що можуть з'явитися характеристики, які в реальності відтворити неможливо. Такі характеристики будуть показувати меншу кількість активних байтів, яке потрібно для їх побудови. Однак головним тут є те, що даний алгоритм не видасть оцінку більшу, ніж може бути насправді. Таким чином, можна говорити про те, що ми отримуємо мінімально допустиму межу для всіляких диференціальних характеристик досліджуваного шифрувального перетворення, що вже є непоганим результатом. Якщо отриманий результат не виходить за допустимі рамки, то можна такий шифр вважати стійким проти атак на пов'язаних ключах.

Розглянемо роботу такого алгоритму на вже відомому прикладі з формуванням проміжного значення K_t. Приклад зображений на рис. 3.8.

Як видно з прикладу на рис. 3.8 кількість ненульових байтів різниці в колонці перераховується після кожного перетворення. Числа позначають кількість байтів, в яких різниця ненульова. Такі байти при проходженні через таблиці підстановки будуть активними. Як видно даний приклад на виході дає 9 активних байтів, по 3 в кожному раунді.

Для того, щоб було зрозуміліше звідки з'являються ці числа в наступних розділах кожне з шифруючих перетворень буде розглянуто більш детально.

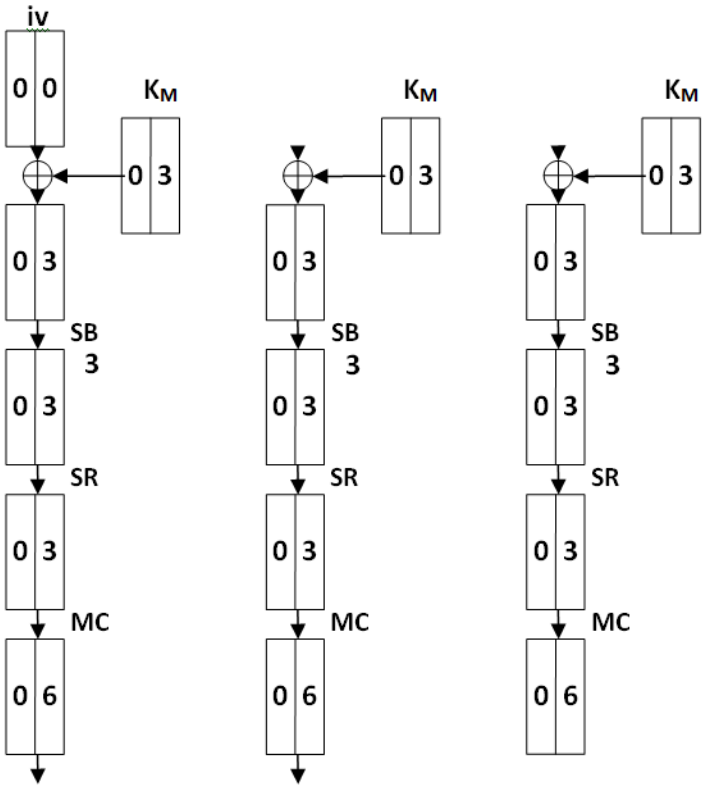


Рисунок 3.8 – Робота алгоритму підрахунку активних байтів при формуванні проміжного значення K_t

3.2.3.2 Додавання з ключем

Операція складання з ключем виконується за наступним принципом: береться різниця станів відповідних колонок по модулю. Більш наочно процедуру складання можна побачити на прикладі на рис. 3.9:

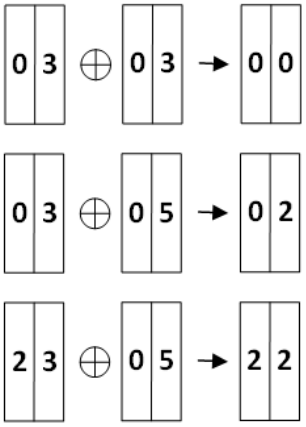


Рисунок 3.9 – Активізація байтів при додаванні з ключем

При додаванні передбачається, що ненульові байти перебувають на таких позиціях і мають такі значення, що станеться колізія. Це припущення в бік криптоаналітика, тому дане припущення може тільки знизити поріг стійкості, але не збільшити його.

Слід зазначити, що в ході експериментів також будувалися моделі, при яких не всі байти ключа і стану шифру входили в колізію, проте ніяких поліпшень при цьому в пошуку кращої характеристики знайдено не було.

3.2.3.3 Підстановка байтів

При підстановці байтів кількість ненульових байтів не змінюється, тому значення в колонках не змінюються. Однак дане перетворення є ключовим, оскільки кількість активних байтів дорівнює кількості ненульових байтів різниці, які проходять через таблиці підстановок. Стійкість шифру залежить від того, скільки ненульових байтів різниці пройде через ці таблиці. Чим більше це значення, тим більш захищеним є шифр.

3.2.3.4 Зсув рядків

У 128-бітної версії шифру Калина операція зсуву виконується наступним чином (рис. 3.10):

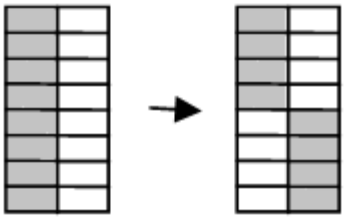


Рисунок 3.10 – Активізація байтів при зсуві рядків

Як видно, останні 4 байти лівої і правої колонки міняються місцями. Оскільки в запропонованому алгоритмі контролюється тільки кількість ненульових байтів в колонці, але не контролюється конкретне місце розташування цих байтів, відповідно виникає питання щодо того, як повинен відбуватися зсув і яка кількість байтів має переходити з однієї колонки в іншу. Дана проблема вирішена таким чином, що перебираються всі можливі варіанти зрушень і диференціальна характеристика отримує додаткові незалежні один від одного в подальшому гілки розвитку. Даний факт становить одну з найбільш трудомістких (в плані обчислювальних витрат) частин алгоритму. Алгоритм буде розгалужуватися експоненційно з наростанням кількості раундів. Розглянемо конкретний приклад на рис. 3.11.

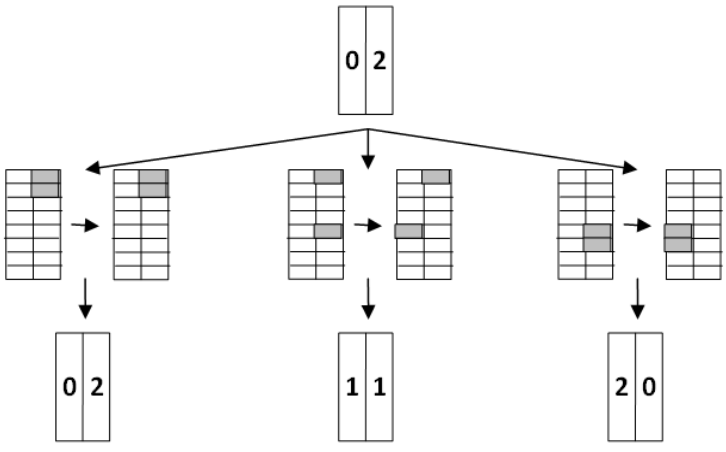


Рисунок 3.11 – Розгалуження диференціальної характеристики при зсуві рядків

Як видно з прикладу на рис. 6.12 стан *(0; 2)* має три незалежних шляху подальшого поширення різниць і всі ці шляхи повинні бути враховані при пошуку кращої диференціальної характеристики.

У табл. 3.3 представлено кількість варіантів зсуву в залежності від кількості ненульових байтів в стовпці.

Загальна кількість розгалужень після зсуву рядків буде дорівнює добутку кількості зрушень лівого і правого стовпчиків. Тобто, наприклад, якщо в лівій колонці 3 ненульових байти, а в правому – 4, то кількість незалежних гілок дорівнюватиме $4 \cdot 5 = 20$.

Таблиця 3.3 – Кількість незалежних шляхів поширення різниць в залежності від кількості ненульових байтів в стовпці

Кількість ненульових байтів в стовпці	Кількість можливих варіантів зсуву для стовпця
0	1
1	2
2	3
3	4
4	5
5	4
6	3
7	2
8	1

3.2.3.5 Перемішування в стовпцях

Перемішування в стовпцях відбувається за рахунок множення стовпця на деяку фіксовану матрицю. Особливість даного перетворення полягає в тому, що сума ненульових байтів на вході і на виході не може бути менше 9,

за винятком ситуації, коли на вході відсутні ненульові байти, тоді і на виході буде нуль: $N_{in} + N_{out} \geq 9$.

У запропонованому алгоритмі робиться припущення в бік криптоаналітика, тобто що завжди буде відбуватися найгірший випадок і на виході буде 9 ненульових байтів різниці: $N_{in} + N_{out} = 9$.

На рис. 3.12 продемонстровано кілька прикладів застосування перемішування в стовпцях.

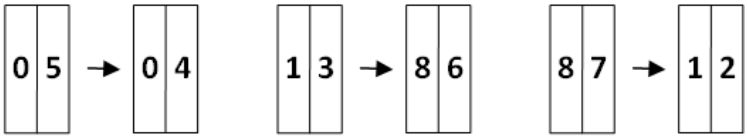


Рисунок 3.12 – Активізація байтів при перемішуванні стовпців

3.2.4 Опис отриманого результату

В результаті проведених експериментів було знайдено найкращу диференціальну характеристику, що має 27 активних байтів. У табл. 3.4 наведено кількість активних байтів для кожного раунду шифрування.

Як видно з табл. 3.4 в основному раунді шифрування диференціальна характеристика стає ітеративною і на кожному раунді додається один активний байт. На рис. 3.13, 3.14, 3.15 знайдена диференціальна характеристика розглянута більш детально.

Умовні позначення:

K – раундовий ключ, для різних фаз шифру має різні значення;

SR – «shift rows» (зсув рядків);

MC – «mix columns» (перемішування в стовпцях).

Таблиця 3.4 – Кількість активних байтів в кожному раунді шифрування для кращої диференціальної характеристики

Частина шифрувального перетворення	Раунд	Кількість активних байтів	Накопичене кількість активних байтів
Формування K_t	1	7	7
	2	4	11
	3	2	13
Схема розгортання	1	2	15
	2	2	17
Основне шифрувальне перетворення	1	1	18
	2	1	19
	3	1	20
	4	1	21
	5	1	22
	6	1	23
	7	1	24
	8	1	25
	9	1	26
	10	1	27
ВСЬОГО			27

На рис. 3.13 розглянуті 3 раунди формування проміжного значення K_t для схеми розгортання ключа. Початкова різниця на вході перетворення дорівнює *(0;0)*, різниця на вході ключа дорівнює *(1;6)* – це майстер-ключ, який підбирався:

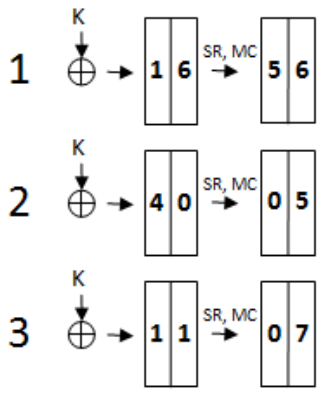


Рисунок 3.13 – Підрахунок активних байтів в схемі формування K_t

Далі на рис. 3.14 представлено 2 раунди схеми розгортання ключа. Початковий стан перетворення одно *(0;7)* – це вихідне значення попереднього перетворення (K_t). Значення ключа одно *(1;6)* – це майстер-ключ.

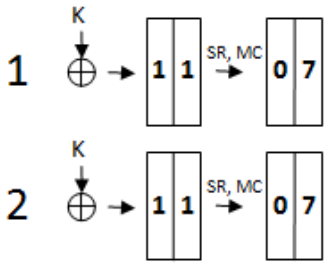


Рисунок 3.14 – Підрахунок активних байтів в схемі розгортання ключа

На рис. 3.15 представлені 10 раундів основного циклу шифрування. На даному етапі є можливість підбирати вхідні стану і в даній диференціальній характеристиці вхідний стан прийнято за *(0;6)*. Раундовий ключ прийме значення *(0;7)* – це вихідне значення попереднього етапу. Різниця *(0;7)* між ключами зберігається на всі раунди завдяки тому, що попередній етап буде для всіх раундових ключів однаковий, відповідно і значення виходять однакові.

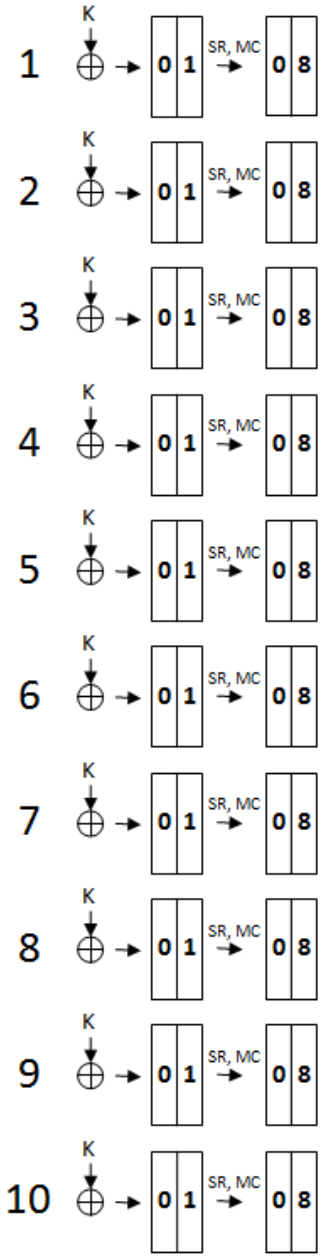


Рисунок 3.15 – Підрахунок активних байтів в основному циклі шифрування

В результаті була отримана диференціальна характеристика, яка містить 27 активних байтів. Однак, як уже згадувалося раніше, отримане значення є лише граничним мінімумом, але насправді реальне значення може бути і більше. Це пов'язано з реалізацією самого алгоритму, в зв'язку з чим можуть з'являтися характеристики, які в реальності побудувати неможливо. Однак навіть такий алгоритм дозволяє переконатися в стійкості випробовується шифрувального перетворення.

Граничним значенням для 128-бітної версії шифру Калина є 26 активних байтів. Це значення ґрунтується на тому факті, що кожен активний байт різниці вносить невизначеність 2^{-5} на виході таблиць підстановок 5 в 5 байтів, які використовуються в Калині. У такому випадку, якщо складність прямого перебору ключем для 128-бітної версти шифру становить 2^{128} операцій, то отримуємо такий вираз для розрахунку безпечної межі кількості активних байтів: $(2^{-5})^x > 2^{-128}$. Тобто при кількості активних байтів $x \geq 26$ складність атаки на пов'язаних ключах буде перевищувати складність атаки прямого перебору, що свідчить про стійкість шифру до подібних атак.

3.2.5 Оцінка складності запропонованого алгоритму

Вхідними значеннями алгоритму можуть бути різні значення різниці майстер-ключа на вході схеми розгортання ключа (етап формування проміжного значення K_t). Оскільки алгоритм контролює кожен стовпець, і значення може бути в межах від 0 до 8 (кількість ненульових байтів різниці), відповідно для кожного стовпця існує 9 різних варіантів вхідних значень. У 128-бітній версії шифру присутні 2 стовпці стану, значення для яких можна підбирати незалежно відповідно загальна кількість вхідних значень досягне 81 (формула 3.6).

$$N_{MK} = N_{Col}{}^{c} = 9^2 = 81. \tag{3.6}$$

Умовні позначення:

N_{MK} – кількість різних різниць майстер-ключа;

N_{Col} – кількість можливих станів однієї колонки стану шифру;

c – кількість колонок в стані шифру.

Також для пошуку найкращої характеристики потрібно підбирати різницю відкритих текстів на вході шифрувального перетворення. Загальна

кількість варіантів відкритих текстів можна розрахувати за такою ж формулою, як і для майстер-ключа (формула 3.7).

$$N_{PT} = N_{Col}{}^{c} = 9^2 = 81, \tag{3.7}$$

де N_{PT} – кількість різних різниць відкритих текстів.

Також слід врахувати, що перетворення «shift rows» може вносити розгалуження, тобто можуть з'являтися додаткові шляхи пошуку диференціальної характеристики. Оскільки алгоритм не контролює позицію кожного байта, а тільки лише загальна кількість активних байтів в колонці, то є різні варіанти зсуву в залежності від припущення на яких позиціях перебувають ненульові байти різниці. Кількість таких варіантів залежить від кількості ненульових байтів різниці в стовпці. У таблиці 3.5 наведено можливу кількість гілок, які з'являються після проходження через зсув рядків для одного стовпчика.

Відповідно, щоб знайти загальне число різних гілок після проходження перетворення потрібно перемножити значення кожного з стовпців.

$$N_{DC} = \prod_{i=0}^{c} N_i \,, \tag{3.8}$$

де N_{DC} – кількість розгалужень після проходження перетворення;

c – кількість колонок в стані шифру;

N_i – кількість розгалужень для кожної колонки (згідно з таблицею 3.5).

Наприклад, якщо поточний стан шифру буде *(3;4)*, то після проходження перемішування в стовпцях з'явиться $4 \times 5 = 20$ незалежних шляхів розвитку диференціальної характеристики.

Якщо врахувати, що відбувається повний перебір текстів, то можна припустити, що в середньому на кожен стовпець буде 3 різних розгалуження, тобто 9 на всі перетворення для 128-бітної версії шифру.

Таблиця 3.5 – Кількість гілок після проходження зсуву рядків в залежності від кількості ненульових байтів різниці на вході перетворення

Кількість активних байтів в стовпці	Кількість шляхів розвитку
0	1
1	2
2	3
3	4
4	5
5	4
6	3
7	2
8	1

Дане перетворення вносить найбільшу складність в алгоритм, тому що воно виконується в кожному раунді, відповідно відбувається експоненціальне зростання складності з кожним раундом. Формула 3.9 демонструє кількість розгалужень в залежності від кількості раундів.

$$N_{DC} = \prod_{i=0}^{r} \prod_{j=0}^{c} N_{ij}\,, \tag{3.9}$$

де N_{DC} – загальна кількість розгалужень для шифру;

c – кількість колонок в стані шифру;

r – кількість раундів;

N_{ij} – кількість розгалужень для кожної колонки в певному раунді (згідно з таблицею 3.5).

Знизити складність алгоритму дозволяє фільтрація свідомо поганих диференціальних характеристик. Тобто по досягненню деякого порога активних байтів обчислення конкретної диференціальної характеристики припиняється. Це дозволяє значно скоротити кількість обчислень.

Загальна формула для підрахунку складності наведена нижче:

$$O = N_{PT} N_{MK} N_{DC} = N_{Col}{}^{c} N_{Col}{}^{c} \prod_{i=0}^{r} \prod_{j=0}^{c} N_{ij} = (N_{Col}{}^{c})^{2} \prod_{i=0}^{r} \prod_{j=0}^{c} N_{ij} . \qquad (3.10)$$

Використовуючи формулу 3.10 можна розрахувати кількість операцій, необхідних для обчислення найкращою диференціальної характеристики для 128-бітної версії шифру. В цьому випадку, передбачається, що для перетворення «shift rows» береться усереднене значення для кожного раунду, що дорівнює 9:

$$O_{128} = 9^2 9^2 9^{15} = 3^{57} = 2^{90}.$$

Ступінь 15 – це кількість раундів шифрувального перетворення (5 раундів схеми розгортання ключа і 10 раундів шифру).

Як видно, кількість обчислень дуже велике, проте, як згадувалося раніше, значно знизити складність дозволяє відсіювання диференціальних характеристик, які переходять граничне значення. Граничне значення для 128-бітної версії шифру одно 27 – це мінімальна кількість активних байтів, яке вдалося отримати для повної версії шифру. Тобто якщо ДХ набирає більшу кількість активних байтів, то далі її обчислення не проводиться.

На комп'ютері з 4-ядерним процесором AMD Phenom з тактовою частотою 3.2 ГГц і 4 Гб оперативної пам'яті даний алгоритм повністю відпрацьовує приблизно за 10 хвилин.

Таким чином, на підставі вищенаведеного можна зробити наступні висновки:

1. Розглянуто сучасний алгоритм блочного симетричного шифрування Калина, який переміг на конкурсі з відбору національного стандарту шифрування в Україні. Цей шифр прийнятий як національний стандарт блочного симетричного шифрування України ДСТУ 7624:2014.

2. Проведено аналіз шифру Калина. Виявлено області, які потребують посиленої аналізу, зокрема потрібно було довести стійкість шифру до атак на пов'язаних ключах. Таке дослідження є важливим, оскільки подібний рід атак застосуємо до шифру AES, який має таку ж базову конструкцію, як і алгоритм Калина.

3. Запропоновано метод оцінки стійкості шифру до атак на пов'язаних ключах. Метод заснований на знаходженні найкращого диференціальної характеристики для шифру. Залежно від кількості активних байтів, присутніх в такій характеристиці, можна зробити висновок щодо його стійкості.

4. За допомогою запропонованого алгоритму була знайдена найкраща диференціальна характеристика для шифру Калина, яка містить в собі 27 активних байтів. Гранична межа для 128-бітної версії шифру становить 26 активних байтів, а це значить, що навіть теоретично найкраща диференціальна характеристика не долає дане граничне значення, що свідчить про те, що шифр є стійким до атак на пов'язаних ключах.

ВИСНОВКИ ЗА РОЗДІЛАМИ 2-3

В ході досліджень були вирішені кілька важливих науково-технічних завдань, що мають практичну значимість для вдосконалення існуючих технологій оцінки стійкості блокових симетричних шифрів. Одне із завдань полягала в оцінці ефективності основних високорівневих конструкцій блокових шифрів, до яких відносяться ланцюг Фейстеля, схема Лей-Мессі і SPN-структура. Оцінювалися дані конструкції за критерієм розрізнення з випадковою функцією або підстановкою. Дане завдання представляла особливу значимість, оскільки до теперішнього моменту не існувало об'єктивних оцінок, за якими можна було б зробити висновок про те, яка з конструкцій є найбільш ефективною.

Інше завдання полягало в розробці методу оцінки захищеності SPN-подібного шифру Калина проти атаки на пов'язаних ключах. Дане завдання також була надзвичайно актуальним, оскільки алгоритм Калина був кандидатом на національний стандарт блочного симетричного шифрування в Україні (а в 2015 році введено в дію в якості стандарту ДСТУ 7624 до: 2014 [81]), внаслідок чого був потрібний всебічний аналіз його захищеності.

Таким чином, основні висновки за результатами роботи полягають в наступному:

1. Запропоновано удосконалене вираз для оцінки максимально можливої ймовірності розрізнення ланцюга Фейстеля з кількістю раундів 3 і більше і випадкової функції. Також було запропоновано апроксимувати вираження для розрахунку даної величини. Застосування уточненого вираження дозволило отримати значно точнішу оцінку максимальної ймовірності розрізнення.

2. Розроблено два методи розрізнення блочного шифру на основі 3-раундової ланцюга Фейстеля і випадкової функції. Один грунтується на застосуванні випадкової функції в якості раундового перетворення, а інший

передбачає використання випадкової підстановки як раундового перетворення. Отримані результати дозволили отримати кількісну оцінку ефективності ланцюга Фейстеля у вигляді ймовірності її розрізнення з випадковою функцією.

3. Розроблено метод розрізнення блочного шифру на основі 3-раундової ланцюга Фейстеля і випадкової підстановки. Отриманий результат дозволив кількісно оцінити ефективність 3-раундової ланцюга Фейстеля з випадковими підстановками як раундового перетворення.

4. Запропоновано метод розрізнення блочного шифру на основі 4-раундової ланцюга Фейстеля і випадкової підстановки. В даному алгоритмі як раундового перетворення ланцюга Фейстеля використовувалася випадкова функція. Отриманий результат дозволив отримати додаткову оцінку ефективності ланцюга Фейстеля.

5. Запропоновано та доведено вираз для розрахунку максимально можливої ймовірності розрізнення схеми Лей-Мессі з кількістю раундів 3 і більше. Отриманий результат дозволив оцінити ефективність схеми Лей-Мессі і порівняти її з іншими високорівневими конструкціями блокових шифрів.

6. Розроблено два методи розрізнення 3-раундової схеми Лей-Мессі з випадковою функцією і випадкової підстановкою в якості раундового перетворення. Отриманий результат дозволив отримати кількісну оцінку ефективності схеми Лей-Мессі.

7. Проведено оцінку ефективності SPN-структури за допомогою критерію розрізнення з випадковою підстановкою. Запропоновано алгоритм-розрізнювач для 2-раундової SPN-структури, а також доведена теорема про неможливість розрізнення SPN-структури з кількістю раундів 3 і більше. Отриманий результат дозволив оцінити ефективність SPN-структури.

8. Всі отримані теоретичні результати по оцінці ефективності високорівневих конструкцій були додатково верифіковані за допомогою розробленого програмного забезпечення.

9. Був проведений порівняльний аналіз всіх трьох розглянутих високорівневих конструкцій за критерієм відрізняються від випадкової функції / підстановки. Отриманий результат дозволив зробити висновок про те, що найбільш ефективною в оцінюваного критерію виявилася SPN-структура і саме її рекомендується використовувати при проектуванні нових блокових симетричних шифрів (результат використаний при розробці національного стандарту ДСТУ 7624 до: 2014 [81]).

10. Розроблено метод автоматизованого пошуку кращої диференціальної характеристики на основі підрахунку кількості активних байтів на різних циклах шифрування. Розроблений метод дозволив оцінити стійкість алгоритму шифрування Калина до атаки на пов'язаних ключах. Було встановлено, що алгоритм Калина є захищеним проти подібного роду атак. Розроблений метод з деякими модифікаціями може бути застосований і до інших шифрів, заснованим на SPN-структурі.

11. Отримані в ході досліджень результати дозволяють вдосконалити існуючі технології оцінки блокових симетричних шифрів і дають в руки проектувальникам нових алгоритмів важливі кількісні оцінки для розробки різних компонентів БСШ.

Достовірність отриманих наукових результатів підтверджується:

– коректністю застосування математичного апарату;

– відсутністю протиріч з існуючою теорією ймовірностей і математичної статистики;

– збігом теоретично отриманих результатів з результатами численних обчислювальних експериментів;

– відсутністю протиріч з існуючими результатами в досліджуваній області криптографії;

Результати дослідження можуть бути використані:

– в організаціях, які займаються проектуванням нових алгоритмів блочного симетричного шифрування, для обґрунтування вибору високорівневої конструкції проектованого шифру;

– в організаціях, які займаються експертизою та оцінкою проектних рішень з побудови сучасних БСШ, для оцінки захищеності шифру проти атак на пов'язаних ключах.

4 ВДОСКОНАЛЕННЯ МЕТОДІВ ОЦІНКИ СТІЙКОСТІ СИМЕТРИЧНИХ БЛОКОВИХ ШИФРІВ ДО ДИФЕРЕНЦІАЛЬНОГО КРИПТОАНАЛІЗУ, ДО АТАК УСІЧЕНИХ ТА НЕЗДІЙСНЕННИХ ДИФЕРЕНЦІАЛІВ

4.1 Проблемні питання оцінювання стійкості сучасних блокових симетричних алгоритмів

4.1.1 Велика кількість відомих і нових атак

На сьогоднішній день відома велика кількість можливих атак на БСШ. Розглянемо спочатку класифікацію атак.

Основними показниками будь криптоаналітичної атаки є складність обчислень і необхідна пам'ять. Необхідна пам'ять вимірюється в кілобайтах, мегабайтах, гігабайтах і т.д., складність атаки зазвичай вимірюють кількістю операцій шифрування з використанням аналізованого шифру. Безумовно, криптоатаки можуть бути класифіковані за цими показниками на більш-менш ефективні. Однак, криптоатаки прийнято також класифікувати ще за двома ознаками:

1) за характером необхідної для криптоаналітика інформації про "поведінку" криптосистеми;

2) за методом розкриття криптосистеми на основі цієї інформації.

Перша ознака відбиває вид взаємодії криптоаналітика з криптосистемою. За цією ознакою відомі атаки можуть бути розбиті на наступні групи [83]:

– атаки з відомим тільки шифртекстом (ciphertext-only attack);

– атаки з відомим відкритим текстом (known plaintext attack);

– атаки з підібраним відкритим текстом (chosen plaintext attack);

– адаптивні атаки з підібраним відкритим текстом (adaptive-chosen-plaintext attack);

– атаки з підібраним шифртекстом (chosen-ciphertext attack);

– адаптивні атаки з підібраним шифртекстом (adaptive-chosen-ciphertext attack);

– атаки з підібраними обома текстами (chosen-text attack);

– атаки з відомими ключовими зв'язками (known related-key attack);

– атаки з підібраними ключовими зв'язками (chosen related-key attack);

– атаки пасивної взаємодії с апаратурою криптозахисту, які передбачають вимір тимчасових затримок, теплових або електромагнітних випромінювань, або рівня енергоспоживання (side-channel attack);

– атаки з впливом на апаратуру криптографічного захисту, спрямованим на внесення певних збоїв в процес криптоперетворень (fault attack).

«Атаки з відомим шифртекст» на практиці зустрічаються частіше інших, так як припускають наявність у криптоаналітика тільки шифрованого тексту і обмежену апріорну інформацію про відповідному йому відкритому тексті.

«Атаки з відомим відкритим текстом» припускають, що атакуючому відомі деякі фрагменти відкритого тексту, відповідні перехопленим криптограмам. На практиці, для цілого ряду додатків, така ситуація є високо ймовірною, оскільки часто в якості відкритого тексту виступають дані, що мають фіксований (відомий) формат з зумовленими значеннями деяких полів. Дана модель є окремим випадком першої моделі, більш сприятливим для криптоаналітика. Тому вважається, що якщо шифр стійкий до атак такого типу, то він також буде стійкий і до атак першої групи.

«Атаки з підібраним відкритим і / або шифрованих текстом» припускають наявність у криптоаналітика можливості управляти входом даних шифратора (для прямого і / або зворотного перетворень). Така умова є досить специфічною і на практиці в більшості додатків є важко виконуваною.

«Адаптивні атаки» є окремим випадком (більш сприятливим для атакуючого) відповідних «атак з підібраним текстом», що передбачає можливість динамічного формування вхідних текстів, виходячи з інформації,

отриманої за раніше накопиченим парам "відкритий-шифрований текст".

Атаки, засновані на пасивній або активній взаємодії з апаратурою криптозахисту, вимагають від атакуючого не тільки можливості фізичного доступу до зазначеної апаратури, а й базуються на методах специфічних для конкретної реалізації відповідних криптоалгоритмів і протоколів. Тому область практичного застосування таких атак сильно обмежена. Захист від атак цього класу швидше відноситься до проблеми не криптографічного, а технічного захисту конкретної реалізації.

«Атаки з відомими або підібраними ключовими зв'язками» припускають, що атакуючий має деяку кількість текстів, отриманих на різних невідомих ключах, але з відомими йому (підібраними їм) залежностями. Для реалізації такої атаки на систему засекречування інформації, на практиці необхідно, щоб процес генерації секретних ключів був заснований на деякому, відомому криптоаналітику, криптографічно слабкому датчику псевдовипадкових послідовностей, або цей датчик повинен бути "закладений" у систему самим криптоаналітиком. Дана модель передбачає наявність вад у реалізації криптосистеми або можливості противника вносити в неї власні "закладки", тому область її практичного застосування в основному обмежена зломом криптосистем підприємств з боку власних співробітників, або з боку сторонніх розробників цих систем.

Друга ознака, за якою можуть бути класифіковані атаки, відображає математичну основу криптоатаки і наявність специфічних властивостей в алгоритмі, що атакується. Більшість методів криптоаналізу може бути реалізовано для різних моделей побудови атак (які класифікуються згідно з першою ознакою), при цьому складність здійснення атаки буде істотно залежати від характеру доступної інформації.

Як видно з класифікаційної схеми методів криптоаналізу (рис. 4.1), все криптоаналітичних атаки можна розділити на два великі класи:

– атаки "грубої сили";

– аналітичні атаки.

Атаки першого класу не враховують внутрішню структуру алгоритму, що атакується, і їх складність залежить тільки від основних параметрів блокового шифру: довжини секретного ключа і довжини блоку даних.

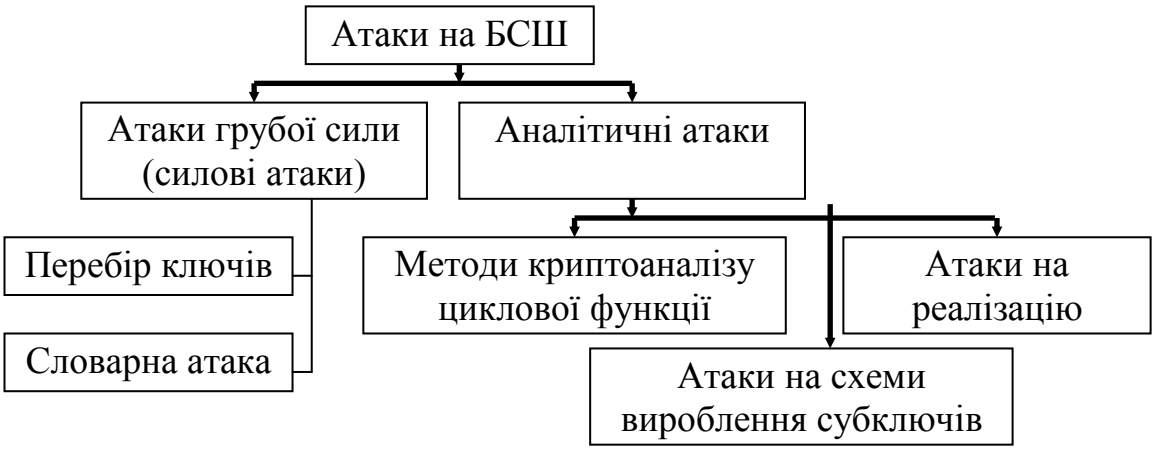


Рисунок 4.1 – Класифікація атак на БСШ

Представлена в класифікації атака повного перебору ключів є найпростішим способом пошуку ключа шифрування. Складність такої атаки залежить від довжини ключа і становить, як відомо [83], не менше 2^{k-1} шифрування за допомогою досліджуваного шифру, де k – розмір ключа в бітах. Для забезпечення захищеності від цієї атаки в шифри використовують ключі великого розміру.

До словникової атаки вразливі шифри, що володіють недостатньою довжиною блоку n. Для виконання атаки потрібно таблиця розміром 2^n блоків, а для побудови такої таблиці необхідно 2^n шифрування [83].

Атаки другого класу використовують специфічні особливості внутрішньої структури конкретного алгоритму. Криптоалгоритм вважається вразливим до деякої аналітичної атаці, якщо складність реалізації цієї атаки менше складності "силової атаки", навіть в тому випадку, коли практичне застосування цієї атаки неможливо на сучасному рівні розвитку обчислювальних засобів. Таким чином, критерій стійкості БСШ з розміром блоку n біт і розміром ключа k біт до аналітичних атак:

$$S \geq \min(2^n, 2^k),$$

де S – складність аналітичної атаки.

Так як для сучасних БСШ зазвичай виконується, то цей вислів зводиться до:

$$S \geq 2^n .$$

При цьому складність будь-якої аналітичної атаки з ростом кількості циклів збільшується і при досягненні певного числа циклів стає вище зазначеної межі, тоді атака перестає бути ефективною і шифр стає захищеним від цієї атаки. Тому, звичайно основний показник стійкості шифру до аналітичної атаці – це число циклів, при якому шифр забезпечує стійкість.

Будь-яка аналітична атака використовує слабкості використовуваних в шифрі перетворень. Зазвичай ці слабкості виражаються в тому, що з високою ймовірністю може бути передбачене значення деякого параметра після великого числа циклів шифру. Тоді, звичайно на останньому циклі, використовуючи відоме значення цього параметра на вході в циклове перетворення і значення криптограми, може бути отримана інформація про підключ, що застосовується в останньому циклі.

Основна ідея аналітичних атак на схеми вироблення підключів пояснена при розгляді атак з відомими ключовими зв'язками і атак з підібраними ключовими зв'язками в класифікації за першою ознакою. Аналогічно, основна суть атак на реалізацію збігається з атаками пасивної і активної взаємодії с апаратурою криптозахисту по першій класифікації.

У даній роботі головна увага приділяється аналітичним атакам на цикловую функцію. На сьогодні відомо близько 30-40 таких атак. До числа найбільш відомих і потужних слід віднести диференційний криптоаналіз, лінійний криптоаналіз, інтегральну атаку, інтерполяційну атаку, атаки усічених і нездійсненних диференціалів, бумеранг-атаку, атаку диференціалів вищих порядків та інші. При цьому щороку з'являються 1-2 нові атаки. Це

можна побачити переглядаючи роботи найвідомішою і престижною конференції в області криптоаналізу симетричних криптоалгоритмів – Fast Software Encryption. Наприклад, серед робіт за 2008 рік до нових атак, напевно, слід віднести роботи [84,85], за 2009 – це роботи [86,87]. При цьому, роботи, де пропонується деяке удосконалення будь-якої атаки за 2008 рік є роботи [88-93], також вимагають перегляду відомих оцінок стійкості до вдосконаленої атаці для існуючих криптоалгоритмів, і таких робіт ще більше на цих конференціях. Зазначені факти вказують на труднощі процесу оцінювання стійкості симетричних криптоалгоритмів.

4.1.2 Висока складність прямого (силового) рішення задачі аналізу стійкості БСШ до аналітичних атак

Ще одним фактором, що ускладнює аналіз стійкості БСШ, є те, що пряме вирішення завдання аналізу стійкості, як правило, пов'язано з необхідністю перебору всіх можливих блоків даних та / або ключів. В результаті, підсумкова складність виходить значно вище складності атак «грубої сили», що робить такий метод аналізу нереалізованим на практиці.

Наприклад, для прямого аналізу стійкості до диференціального криптоаналізу необхідно, по суті, побудувати величезну таблицю різниці. Для БСШ з розміром блоку n і ключа k для цього необхідно перебрати всі ключі, для кожного ключа перебрати всі значення вхідної різниці і всі можливі вхідні значення. Підсумкова ймовірність складе 2^{2n+k} операцій шифрування.

У табл. 4.1 наведені складності прямого (силового) рішення задачі оцінювання стійкості для деяких аналітичних атак.

Таблиця 4.1 – Складності прямого (силового) рішення задачі оцінювання стійкості для деяких аналітичних атак

Атака	Складність аналізу стійкості прямим шляхом
Диференціальний криптоаналіз	2^{2n+k}
Лінійний криптоаналіз	2^{3n+k}
Атака нездійсненних диференціалів	2^{2n+k}

Пряме рішення задачі аналізу стійкості, пов'язане з повним перебором ключа та / або вхідних блоків, можливо тільки для БСШ з невеликими значеннями розміру блоку і ключа (до 32 бітів) і застосовується для БСШ, що відповідає сучасним вимогам (з розміром блоку і ключа 128 і більше бітів). Як наслідок, для отримання оцінок стійкості сучасних симетричних кріптопрімітівов потрібні спеціальні методи.

4.1.3 Відсутність гарантій реальної стійкості для багатьох методів оцінювання стійкості

Багато методів оцінювання стійкості розглядають параметр, значення якого не може гарантувати стійкість шифру до заданого виду атаки. Зазвичай значення обраного для розгляду в такому методі параметра легше оцінити, ніж значення параметра, який дійсно визначає складність атаки.

Прикладом такої ситуації можуть служити методи оцінювання практичної стійкості до диференціального криптоаналізу, в яких розглядаються ймовірності диференціальних характеристик (ДХ), хоча для того, щоб гарантувати стійкість БСШ, необхідно розглядати ймовірності диференціалів (Д) (такі методи отримали назву методів доказової або теоретичної стійкості). Відомо, що саме максимальна ймовірність ймовірності диференціала визначає складність атаки. Також відомо, що верхня межа ймовірності диференціалів не може бути нижче, ніж 2^{-n}, а верхня межа ймовірності ДХ може бути значно нижче (рис. 4.2).

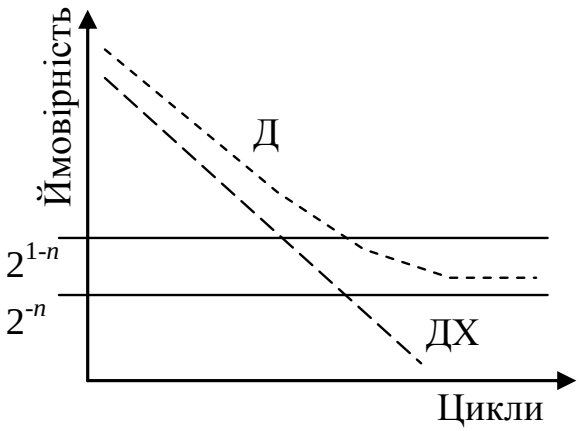


Рисунок 4.2 – Динаміка зміни верхніх меж ймовірностей диференціалів (Д) і диференціальних характеристик (ДХ) для n-бітного БСШ

Тому робити висновки про складність атаки на основі ймовірностей ДХ некоректно, хоча при обґрунтуванні стійкості більшості шифрів частіше використовують саме методи оцінювання практичної стійкості. Більш детально основні ідеї атаки диференціального криптоаналізу будуть розглянуті в розділі 4.2.

Перспективним напрямком удосконалення методів оцінювання стійкості БСШ є їх аналіз на предмет гарантованості одержуваних оцінок стійкості шифру, а також розробка таких методів, які виконують оцінку тих параметрів, які безпосередньо визначають складність атаки, тобто методів доказової стійкості.

4.1.4 Вплив тенденцій розвитку симетричної криптографії на вимоги до методів оцінювання стійкості

Одна з тенденцій розвитку симетричною криптографії пов'язана з постійним збільшенням розміру блоку і ключа БСШ. Наслідком чергового збільшення цих параметрів може стати те, що методи оцінювання стійкості, які пов'язані з вирішенням завдань перебору частини блоку або ключа, можуть не реалізовуватись на практиці за прийнятний час. Тому, з

урахуванням цієї особливості, більш перспективними представляються методи оцінювання, складність яких буде мінімально залежати від розміру блоку і ключа.

Ще однією особливістю розвитку сучасної симетричною криптографії є широке використання в криптоалгоритмах перетворень найпоширенішого шифру Rijndael (AES). Тому актуальним є розвиток методів оцінювання стійкості до тих атак, які є найбільш ефективними для цього шифру.

У специфікації шифру [94], а також деяких інших роботах [95, 96] представлені результати оцінювання стійкості алгоритму шифрування до різних криптоаналітичних атак. Ці результати зведені в табл. 4.2.

Таблиця 4.2 – Оцінки стійкості алгоритму шифрування Rijndael-128 до різних криптоаналітичних атак

Види криптоатак	Мін. число циклів, при якому шифр стійкий	Показники відомих атак на AES (rijndael-128)		
		Макс. число циклів	Обчисл. ресурси, екв.опер.	Пам'ять
Диференціальна	4	3	2^{54}	Мало
Лінійна	4			
Усічен. диффер.	4	3	2^{8}	Мало
Незд. диффер.	6	5	2^{36}	2^{42}
Інтерполяційна	5			
Інтегральна	7	6	2^{72}	2^{32}

Наведені в таблиці 4.2 дані свідчать, що найбільш ефективними з відомих криптоаналітичних методів нападу є інтегральна атака і атака нездійсненних диференціалів. Обидві ці атаки відносяться до класу атак з підібраними відкритими текстами, тобто для їх реалізації криптоаналітику необхідно мати достатню безліч криптограми, отриманих при зашифруванні

спеціально підібраних відкритих текстів на одному і тому ж секретному ключі.

4.1.5 Висновки по розділу

1. Невід'ємним елементом в загальному комплексі засобів і методів захисту інформації є блокові симетричні шифри, які використовуються для забезпечення конфіденційності даних, а також в якості «будівельних блоків» для інших криптопримітивів, таких, як алгоритми хешування, алгоритми поточного шифрування, генератори псевдовипадкових чисел.

2. Один з найбільш важливих, відповідальних і разом з тим складних етапів створення сучасного БСШ полягає в дослідженні можливостей застосування відомих криптоаналітичних методів для цього шифру. Проведений в розділі аналіз факторів дозволив виявити протиріччя, яке полягає в тому, що при створенні блочного симетричного шифру потрібно виконати оцінку стійкості до багатьох видів криптоаналітичних атак, а для прямого вирішення цього завдання потрібні нереалізуємі на практиці обчислювально-часові ресурси (табл. 4.1). Таким чином, необхідно вирішити наступну наукову проблему: розробити методи і засоби оцінювання стійкості сучасних симетричних криптоперетворень для забезпечення уточнених, гарантованих оцінок стійкості в умовах обмежених обчислювально-часових ресурсів.

Найбільш актуальними напрямами розробки методів і засобів оцінювання стійкості сучасних симетричних криптоперетворень є:

1) розробка методів оцінювання стійкості для нових атак;

2) дослідження існуючих методів на предмет достовірності одержуваних висновків про стійкість;

3) розробка методів оцінювання стійкості, заснованих на теоретичному обгрунтуванні нижніх меж складності для відомих атак;

4) вдосконалення методів виконання оцінювання стійкості в напрямку підвищення їх достовірності, точності, розширення сфери застосування, зниження їх обчислювальної складності;

5) широке поширення Rijndael-подібних шифруючих перетворень робить актуальним, в першу чергу, розвиток методів оцінювання стійкості до найбільш ефективним для цього виду шифрів атакам.

За результатами розділу опубліковані роботи [98-107].

4.2 Особливості криптоаналітичних атак і відомих методів оцінювання стійкості

У цьому підрозділі будуть наведені короткі характеристики розглянутих криптоаналітичних атак, а також будуть проаналізовані відомі методи оцінювання стійкості для кожної з атак, з метою виявлення та обґрунтування перспективних напрямків їх вдосконалення.

4.2.1 Диференціальний криптоаналіз

4.2.1.1 Основні поняття і визначення

Диференціальний криптоаналіз не є найефективнішою атакою на Rijndael-подібні шифри (табл. 4.2), але, з одного боку, ця атака – одна з найбільш універсальних і ефективних атак на БСШ в цілому, з іншого боку, такі більш ефективні атаки, як атака нездійсненних диференціалів і атака усічених диференціалів, використовують базові принципи диференціального криптоаналізу. Тому почнемо огляд атак і методів оцінювання стійкості саме з диференціального криптоаналізу.

Ця атака з вибраними відкритими текстами була запропонована для алгоритму DES [108-110] і стала першим теоретичним методом знаходження секретного ключа шифру DES з меншою складністю, ніж повний перебір. На сьогоднішній день диференційний криптоаналіз залишається одним з найбільш ефективних методів криптоаналізу БСШ.

Для організації атаки вибирають пари відкритих текстів (P, P') з певною різницею (difference) $\Delta = P - P'$, де "–" позначає операцію, зворотну операції введення підключів. Наприклад, якщо при шифруванні виконується складання з ключем по модулю 256, то $\Delta = P - P' \bmod 256$, якщо складання з ключем виконується по модулю 2 ($\oplus$), то $\Delta = P \oplus P'$'. Тоді, значення різниці не змінюється при додаванні з ключем:

$$(P + k) - (P' + k) = P - P' = \Delta.$$

Лінійними перетвореннями з точки зору обраної разностной операції називають перетворення, для яких справедливо наступне:

якщо $y_1 = A(x_1)$, $y_2 = A(x_2)$, то $y_1 + y_2 = A(x_1) + A(x_2) = A(x_1 + x_2)$.

Таким чином, лінійні перетворення не вносять невизначеності в проходження різниці. Імовірнісний характер "транспортуванню" різниці через цикли шифру надають нелінійні перетворення.

Зв'язок вхідної та вихідної різниць при проході одного циклу перетворень називається одноцикловою диференціальною характеристикою. Цей зв'язок характеризується ймовірністю того, що випадкова пара відкритих текстів з різницею α перейде після проходу одного циклу перетворень в різницю β. Імовірність такої одноциклової диференціальної характеристики будемо позначати як $p^{(1)}(\alpha \to \beta)$.

При проведенні різниці через кілька циклів шифрування використовується поняття багатоциклової диференціальної характеристики. Багатоциклова характеристика отримується конкатенацією сусідніх одноциклових характеристик, які відповідають умові зшивання ($\beta_{i-1} = \alpha_i$). Імовірність r-циклової характеристики визначається як добуток ймовірностей складових її одноциклових характеристик:

$$p^{(r)} = \prod_{i=1}^{r} p^{(1)}(\alpha_i \to \beta_i).$$

Пару $(\Delta x_j, \Delta y_{j+i}) = (\alpha, \beta)$, визначальну різницю на вході j циклу і різницю після циклу $j + i$, називають i-цикловим диференціалом, а послідовність значень $(\alpha, \beta_1, \beta_2, \ldots, \beta_{i-1}, \beta)$, де кожне β_k ($k = 1, i-1$) – значення різниці після k-го циклу, називається i-цикловою диференціальною характеристикою, що належить i-цикловому диференціалу (α, β). Імовірність i-циклового диференціала $P^{(i)}(\Delta y_i = \beta \mid \Delta x_1 = \alpha)$ визначається як сума ймовірностей належних йому характеристик:

$$P^{(i)}(\Delta y_i = \beta | \Delta x_1 = \alpha) = \sum_{\forall \beta_1,\ldots,\beta_{i-1} \neq 0} p^{(i)}(\Delta y_1 = \beta_1, \ldots, \Delta y_{i-1} = \beta_{i-1}, \Delta y_i = \beta | \Delta x_1 = \alpha)$$

.

Тому завжди справедливою є наступна нерівність:

$$P^{(r)} \geq p^{(r)}. \qquad (4.1)$$

Якщо в розглянутому шифрувальні перетворення використовується апріорно невідомий k-бітний ключ K, то таке перетворення зазвичай характеризується усередненої для всіх варіантів ключа ймовірністю диференціала:

$$P^{(i)}[K](\Delta y_i = \beta | \Delta x_1 = \alpha) = 2^{-k} \sum_{j=0}^{2^k} P_j^{(i)}(\Delta y_i = \beta | \Delta x_1 = \alpha).$$

Так як диференційний криптоаналіз відновлює біти ключа на останньому циклі, то для успішного "нападу" на шифр необхідний багатоцикловий диференціал, що покриває майже все цикли шифру і володіє ймовірністю, що значно перевищує середню ймовірність інших диференціалів з аналогічним входом α. Для випадку n-бітного блоку середня ймовірність деякого диференціала становить $1/(2^n-1) \approx 2^{-n}$. Тому застосування атаки диференціального криптоаналізу виправдано, якщо

ймовірність "транспортування" деякої різниці через всі, крім кількох циклів (зазвичай 1 – для SPN-шифрів і 2 або 3 – для Фейстель-подібних шифрів) значно більше, ніж 2^{-n+1}. Складність атаки диференціального криптоаналізу на r-цикловий шифр $S_{ДК}$ залежить від максимальної ймовірності диференціала $P_{\max Д}^{(r-q)}$ наступним чином:

$$S_{ДК} \approx \frac{1}{P_{\max Д}^{(r-q)}},$$

де q = 1 для SPN-шифрів та q = 2 або q =3 для фейстель-подібних шифрів.

Таким чином, з урахуванням загального критерію стійкості до аналітичних атак, критерій захищеності r-циклового шифру від диференціального криптоаналізу є виконання нерівності:

$$P_{\max Д}^{(r-q)} \le 2^{1-n}. \qquad (4.2)$$

Значення $P_{\max Д}^{(r-q)}$ визначає складність виконання найбільш ресурсомісткого етапу диференціального криптоаналізу – підбору вірної пари відкритих текстів. Використовуючи вірну пару, можуть бути отримані висновки щодо значень підключа останнього циклу.

4.2.1.2 Диференціальні властивості елементів БСШ

У цьому підрозділі розглядаються деякі властивості окремих елементів шифрів, що грають важливу роль в забезпеченні стійкості БСШ до диференціальних атак.

1. Лінійні перетворення.

Диференціальний криптоаналіз базується на лінійності більшості використовуваних в шифрі перетворень. Як вже говорилося, під лінійним перетворенням T, розуміється перетворення, для якого справедливим є:

$$T(X) \oplus T(X') = T(X \oplus X'),$$

тобто значення різниці на виході повністю визначається вхідною різницею. Лінійні перетворення в шифрі вирішують задачу розсіювання. Для оцінювання якості розсіювання, що забезпечується перетвореннями лінійного рівня, в [111] введено поняття числа гілок активізації (branch number). Число гілок активізації B для лінійного перетворення T визначається як:

$$B = \min_{\Delta X \neq 0} (H(X) + H(T(X))),$$

де X – значення на вході лінійного перетворення;

$T(X)$ – результат виконання лінійного перетворення;

$H(X)$ – функція, яка повертає число активних S-блоків (активних байтів) в X.

Чим більше число гілок активізації перетворень розсіювання, тим більше активних S-блоків буде містити диференціальна характеристика, а значить – вище невизначеність проходження різниці, і вище стійкість шифру.

Серед лінійних перетворень широке застосування отримали перетворення на основі МДР-кодів (коди з максимально допустимим відстанню). Вперше такі перетворення були використані при побудові блочного шифру Shark [111]. Подібні перетворення також використовуються в шифри Square [112], Rijndael [94], Khazad [113], Anubis [114]. Головна перевага лінійних перетворень цього класу полягає в тому, що гарантується максимально досяжне число гілок активізації. Тобто якщо МДР-перетворення покриває M S-блоків, то $B = M+1$.

Як вже було зазначено, лінійні перетворення не вносять невизначеності при проходженні через них різниці, тому стійкість шифру до

диференціального криптоаналізу багато в чому визначається нелінійними перетвореннями.

2. Нелінійні перетворення

Нелінійні перетворення забезпечують перемішування. Зазвичай єдиною нелінійної частиною криптосистеми є підстановки, звані S-блоками. Саме нелінійні підстановки вносять невизначеність у проходження різниці через цикли шифру.

У загальному випадку підстановка може визначати відображення m-бітного значення в *k*-бітове. Відповідно до [114], ймовірність переходу заданої вхідної різниці $\Delta x \in \{0,1\}^m$ в вихідну різницю $\Delta z \in \{0,1\}^k$ при проходженні через S-блок визначається так:

$$p_S(\Delta x \to \Delta z) = \frac{\#\{x \in \{0,1\}^m \mid s(x) \oplus s(x \oplus \Delta x) = \Delta z\}}{2^m}.$$

Оскільки не всі переходи можливі, і не всі можливі переходи різновірогідні, то для вивчення проходження різниць через S-блок будують таблицю різниць, в яку заносять значення $\#\{x \in \{0,1\}^m \mid s(x) \oplus s(x \oplus \Delta x) = \Delta z\}$ для всіх $\Delta x \in \{0,1\}^m$ і $\Delta z \in \{0,1\}^k$. Використовуючи таку таблицю, легко визначити максимальну ймовірність проходження ненульовий різниці через S-блок:

$$p_{S\max} = \max_{\substack{\Delta x \in \{0,1\}^m, \Delta x \neq 0 \\ \Delta z \in \{0,1\}^k}} p_S(\Delta x \to \Delta z).$$

Чим менше це значення, тим вище невизначеність значення різниці на виході S-блоку, і, отже, підстановка більш стійка до диференціального криптоаналізу.

S-блок, на вхід якого надходить ненульова різниця, називається активним. При кожному активному S-блоці, в кращому для криптоаналітика випадку, ймовірність диференціальної характеристики множиться на $p_{s\max}$. Відповідно, чим менше $p_{s\max}$, тим вище невизначеність проходження різниці через S-блок. Відомо, що для биективное підстановок ($m = k$) найбільш вдалими в цьому сенсі вважаються, так звані, підстановки з граничною нелінійністю (Almost Perfect Nonlinearity - APN). Відповідно до робіт [115,116], максимальна ймовірність проходження різниці через такий S-блок дорівнює $\frac{4}{2^m}$, якщо m – парне, і $\frac{2}{2^m}$, якщо m – непарне.

4.2.1.3 Відомі методи оцінювання стійкості БСШ до ДК

Обґрунтування стійкості шифру до диференціального криптоаналізу є обов'язковим компонентом будь-якого сучасного БСШ. В даному підрозділі розглядаються основні методи оцінювання стійкості БСШ до диференціальних атак.

Розгляд диференціалів дає більш достовірну оцінку стійкості БСШ, ніж розгляд диференціальних характеристик, але, при цьому, оцінити ймовірність диференціала значно складніше. В результаті таких суперечливих обставин, сформувалися чотири критерії стійкості БСШ з розміром блоку n бітів до диференціального криптоаналізу [117]:

- Точний критерій – максимальне значення ймовірності диференціала нижче, ніж 2^{-n}.

В роботі [118] описано виконання точного розрахунку безпеки шифру проти диференціального криптоаналізу. Однак розрахувати ці ймовірності для сучасного шифру неможливо, тому цей критерій на практиці не застосовується.

- Теоретичний критерій – верхня межа значення ймовірності диференціала нижче, ніж 2^{-n}.

З ряду теоретичних робіт, наприклад [119], відомо кілька теорем про верхню межу ймовірності диференціалів. Однак, як показує практика, теоретично стійкий до диференціального криптоаналізу шифр часто виявляється або вразливим до інших видів атак, або недостатньо швидким.

- Евристичний критерій – максимальна ймовірність диференціальних характеристик нижче, ніж 2^{-n}.

В [120-122], в якості показника можливості організації диференціального криптоаналізу, пропонується використовувати найбільшу ймовірність окремої диференціальної характеристики. У зазначених роботах пропонуються алгоритми для організації пошуку ДХ, що володіють максимальною вірогідністю. Однак, як показали обчислювальні експерименти, результати яких представлені в [123], ці алгоритми можуть використовуватися для БСШ, розмір блоку яких становить менше 64 бітів.

- Практичний критерій – верхня межа ймовірності диференціальних характеристик нижче, ніж 2^{-n}.

В [124] стверджується, що шифри, оцінені відповідно до цього критерію, можна вважати стійкими до диференціального криптоаналізу.

На практиці для оцінювання стійкості сучасних шифрів ($n \geq 64$) зазвичай використовується або теоретичний, або практичний критерії.

а) Методи, засновані на перевірці теоретичного критерію

В [119] запропоновано визначення DES-подібного шифру, під яким розуміється шифр на базі ланцюга Фейстеля, в циклової функції якого присутній додавання (в контексті деякої групової операції) з цикловим підключем, що обирається з простору неменшої розмірності, ніж простір входів.

Якщо для DES-подібного шифру припустити, що підключи різних циклів різновірогідні і незалежні, то можна вважати, що вхідні значення циклових функцій різновірогідні і незалежні. Для даного припущення в [119] доведено теорему про макисмально ймовірності r-циклового диференціала для шифру на базі ланцюга Фейстеля. Відповідно до цієї теореми ймовірність

r-циклового диференціала при $r \geq 4$ задовольняє нерівності $P^{(r)}(\Delta^{(i)}, \Delta^{(i+r)}) \leq 2p_F^2$, де p_F – максимальна ймовірність диференціальної характеристики циклової функції *F*. При цьому, якщо *F* є перестановкою (тобто *F* бієктивна), то при $r \geq 3$ справедливою є нерівність $P^{(r)}(\Delta^{(i)}, \Delta^{(i+r)}) \leq p_F^2$.

В [105] показано, що для того, щоб DES-подібний шифр відповідав теоретичному критерію, необхідно розробити цикловую функцію, що володіє досить низькими значеннями максимуму ймовірностей диференціальних характеристик, що пов'язано зі значними витратами на реалізацію циклової функції і, як наслідок, з невисокою швидкістю процедури шифрування.

Для SPN-шифрів в [124] доведена така теорема.

Теорема 4.1 ([124]). Якщо в SPN-шифрі використовується *n* S-блоків, а перетворення розсіювання забезпечують число гілок активізації рівне *n* – *t*, то ймовірність диференціала, що покриває 2 і більше циклів, буде обмежена значенням p^{n-t-1}, де *p* – максимальна ймовірність проходження ненульовий різниці через S-блок.

Своє продовження теорія про верхню межу ймовірності багатоциклових диференціалів отримала в [125], де розглянуті вкладені SPN-структури (Nested SPN). На рис. 4.3 представлена 2-циклова (число циклів визначається числом XS-рівнів найвищого порядку) дворівнева вкладена SPN-структура, для якої відповідно до теореми 2 з [125], значення ймовірності диференціала обмежено зверху значенням $p^{(n_1-(t_1+1))\cdot(n_2-(t_2+1))}$, де n_1 – число S-блоків в кожному XS-блоці, n_2 – число XS-блоків в блоці, $n_1 - t_1$ і $n_2 - t_2$ число гілок активізації, що забезпечується нижнім і верхнім рівнями перетворень розсіювання, відповідно.

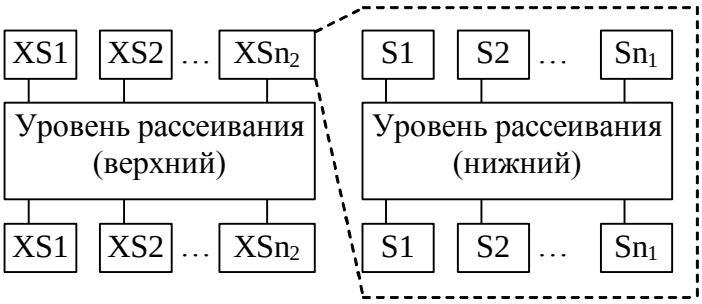


Рисунок 4.3 – Двоциклова дворівнева вкладена SPN-структура

Слід зауважити, що створення теоретично стійкого SPN-шифру згідно з поданими теоремами пов'язане з досить жорсткими вимогами до використовуваних в кожному циклі перетворень, наслідком чого стане уповільнення процедури шифрування. Для підтвердження сказаного зауважимо, що верхня межа для ймовірності багатоциклових диференціалів переможця конкурсу AES, розрахована відповідно до теореми 2 з [95], становить всього 2^{-96}, тобто шифр Rijndael не є теоретично стійким до диференціального криптоаналізу. Але виходячи з того, що за час ретельного тестування цього шифру не було виявлено диференціалів, що володіють високою ймовірністю, то це свідчить про велике завищення верхньої межі ймовірності диференціалів, розрахованої за допомогою відомих на сьогоднішній день методів. Тому важливим завданням є отримання більш точних оцінок верхньої межі ймовірності диференціалів. Разом з тим, методи, засновані на перевірці теоретичного критерію, мають більшою вірогідністю, ніж методи, засновані на перевірці практичного або евристичного критеріїв.

б) Стандартний метод оцінювання стійкості до диференціального криптоаналізу (перевірка практичного критерію).

В основі традиційного методу оцінювання стійкості БСШ до диференціального криптоаналізу лежить підрахунок мінімального числа активних S-блоків в диференціальної характеристиці, що покриває достатнє для організації ефективної атаки число циклів. Далі для обчислення верхньої межі ймовірності диференціальної характеристики максимальну ймовірність

проходження ненульовий різниці через S-блок зводять до рівня числа активних S-блоків.

При обчисленні мінімального числа активних S-блоків в диференціальної характеристиці для SPN-шифру часто використовується той факт, що число активних S-блоків в двох сусідніх циклах буде не менше числа гілок активізації лінійного рівня B [111]. У цьому випадку обчислення граничної ймовірності диференціальної характеристики, що покриває $2r$ циклів, виконується за формулою:

$$p_{dSPN\max}^{(2r)} = (p_{s\max})^{rB}, \tag{4.3}$$

де $p_{s\max}$ – максимальна ймовірність проходження різниці через S-блок.

У випадку, коли блок даних подається у вигляді квадратної матриці розміром $n \times n$ байтів, для цих шифрів показано, що кожна 4-циклова диференціальна характеристика буде містити не менше $(n + 1)^2$ активних S-блоків (Теорема 1 [94]), що призводить до наступного значення граничної ймовірності 4-циклової диференціальної характеристики:

$$p_{dRijn\max}^{(4r)} = (p_{s\max})^{r(n+1)^2}. \tag{4.4}$$

В роботі [123] була доведена більш загальна теорема, на основі якої для Rijndael-подібного шифру з n_c колонками і n_r рядками (n_r кратне n_c) гранична ймовірність 4-циклової диференціальної характеристики складе:

$$p_{dRijn\max}^{(4r)} = (p_{s\max})^{r(n_r+1)(n_c+1)}. \tag{4.5}$$

Максимальна ймовірність диференціальної характеристики для Фейстель-подібних шифрів обговорюється в [126]. Представлена в [126] теорема 3.9 говорить про те, що граничне значення ймовірності диференціальної характеристики, що покриває 4r циклів, дорівнює:

$$p_{dFN\max}^{(4r)} = \left(p_{S\max}\right)^{rB+\left\lfloor \frac{r}{2} \right\rfloor}. \qquad (4.6)$$

Існує кілька недоліків даного підходу до оцінки стійкості шифру. Перший недолік притаманний усім методам, які працюють з диференціальними характеристиками, а не з диференціалами. Розгляд ймовірностей диференціальних характеристик не гарантує стійкість шифру навіть якщо ці ймовірності обмежені досить невеликим значенням. Деякі атаки можуть використовувати диференціали, ймовірність яких може бути вище ймовірності диференціальних характеристик. Другий недолік полягає в тому, що при використанні формул (4.3) або (4.6) виконувана оцінка є грубою і тому можливе отримання занижених показників стійкості шифру. Наприклад, якщо розраховувати максимальну ймовірність 4-циклової диференціальної характеристики шифру Rijndael за допомогою формули (4.3), то отриманий результат буде сильно перевищувати дійсне значення максимальної ймовірності диференціальної характеристики, отримане з використанням формул (4.4) або (4.5). А значить, буде потрібно більше число циклів для забезпечення захищеності шифру від диференціальних атак, що зробить шифр менш продуктивним.

в) Оцінка стійкості Фейстель-подібних БСШ до диференціальних атак на основі пошуку ефективних ДХ (перевірка евристичного критерію).

Якщо попередній метод виконує оцінку стійкості з точки зору розробника, який хоче продемонструвати неможливість організації диференціального криптоаналізу, то в цьому підрозділі розглядається оцінка стійкості з точки зору зловмисника, який намагається провести

диференційний криптоаналіз і для цього шукає найбільш ймовірну диференціальну характеристику. Перевірка евристичного критерію важлива при оцінці стійкості шифрів, для яких не може бути достатньо точно визначена верхня межа ймовірності диференціальної характеристики.

В роботі [120] М. Мацуї запропонував для DES-подібних шифрів метод пошуку багатоциклових диференціальних характеристик, що володіють максимальними можливостями. Для викладу цього методу в [120] введено такі позначення:

$\mathrm{BEST_N}$ – максимальна ймовірність N-циклової диференціальної характеристики;

$\mathrm{CAND_N}$ – передбачуване значення $\mathrm{BEST_N}$;

ΔX_r – різниця на вході r-го циклу;

ΔY_r – різниця на виході r-го циклу;

p_r – ймовірність одноциклової диференціальної характеристики на r-му циклі, тобто ймовірність переходу ΔX_r в ΔY_r.

Для виконання пошуку N-циклової диференціальної характеристики, що володіє найбільшою ймовірністю повинні бути відомі всі BESTr (r < N). Коротко викладемо основні етапи методу Мацуї.

Підготовка: визначаються початкові значення $\mathrm{CAND_N}$. Ці значення повинні бути якомога більше, але менше справжнього максимального значення ймовірності N-циклової диференціальної характеристики.

Пошук на 1-му циклі: для кожного варіанта ΔX_1 виконується наступне:

- вибирається значення ΔY_1, при якому ймовірність p_1 максимальна;

- переходимо до пошуку на 2-му циклі, якщо виконується $p_1 \geq \dfrac{\mathrm{CAND_N}}{\mathrm{BEST_{N-1}}}$.

Пошук на 2-м циклі: для кожного варіанта пари ΔX_2, ΔY_2 виконується наступне:

- переходимо до пошуку на 3-му циклі, якщо виконується $p_1 p_2 \geq \frac{\mathrm{CAND}_{\mathrm{N}}}{\mathrm{BEST}_{\mathrm{N}-2}}$.

Пошук на r-му циклі ($3 \leq r \leq \mathrm{N}-1$): Обчислюється $\Delta X_r = \Delta \mathrm{Y}_{r-1} \oplus \Delta X_{r-2}$. Для кожного варіанта ΔY_r виконується наступне:

- переходимо до пошуку на (r + 1)-му циклі, якщо виконується $\prod_{i=1}^{r} p_i \geq \frac{\mathrm{CAND}_{\mathrm{N}}}{\mathrm{BEST}_{\mathrm{N}-r}}$.

Пошук на N-му циклі:

- обчислюється $\Delta X_{\mathrm{N}} = \Delta \mathrm{Y}_{\mathrm{N}-1} \oplus \Delta X_{\mathrm{N}-2}$;

- вибирається значення ΔY_{N}, при якому ймовірність p_{N} максимальна;

- якщо виконується $\prod_{i=1}^{\mathrm{N}} p_i \geq \mathrm{CAND}_{\mathrm{N}}$, то $\mathrm{CAND}_{\mathrm{N}} = \prod_{i=1}^{\mathrm{N}} p_i$.

В результаті $\mathrm{BEST}_{\mathrm{N}} = \mathrm{CAND}_{\mathrm{N}}$.

Представлений метод можна застосовувати для шифрів з невеликим розміром блоку (не більше 64 бітів), наприклад DES, FEAL, і вимагає при цьому значних обчислювальних ресурсів. Для шифрів з великим розміром блоком розглянутий алгоритм зажадає занадто великих обчислювальних ресурсів.

4.2.1.4 Порівняльний аналіз точності методів оцінювання максимальної ймовірності диференціалів

У цьому пункті за допомогою відомих методів оцінювання максимальної ймовірності диференціалів проаналізуємо зменшені моделі шифрів. Доцільність розгляду зменшених моделей шифрів пояснюється тим, що для вивчення стійкості шифру до диференціальної атаці слід оцінювати ймовірності повних диференціалів – параметр, який на сьогодні точно можна оцінити тільки для шифрів з невеликим розміром блоку. У якості операцій

перемішування і розсіювання були взяті перетворення, запропоновані в [127] для зменшеної версії шифру Rijndael.

Розглянуті модифікації для нелінійної підстановки: оригінальна підстановка з максимальною вірогідністю проходження ненульовий різниці 2^{-2} (sbox_prob = 2^{-2}) і підстановка з більшою максимальною вірогідністю проходження ненульовий різниці 2-1 (sbox_prob = 2^{-1}) [128].

Розглянуті модифікації для лінійного перетворення MixColumn: оригінальне перетворення може конвертувати по 8 бітів (8 bit MixColumn), модифіковане – по 16 бітів (16 bit MixColumn) [128].

До основних особливостей запропонованих зменшених моделей шифрів слід віднести [128-133]:

- розмір блоку 16 біт, розмір ключа 16 біт;

- структура блоку: 2 колонки по 2 4-бітових елемента або одна колонка з 4 4-бітових елемента;

- множення елементів кожної колонки на фіксовану МДР-матрицю розміром 2 на 2 або 4 на 4 проводиться над GF (2^4) (MixColumns);

- підстановка 4 в 4 біта (SubBytes).

Був використаний наступний алгоритм отримання максимальних ймовірностей диференціалів для зменшених шифрів з 16-бітовим блоком [128].

1. Перебираємо всі значення вхідної різниці Δx (2^{16} варіантів).

1.1 Перебираємо значення ключа k (64 випадкових значеннь).

1.1.1 Перебираємо всі значення одного з відкритих текстів x (2^{16} варіантів).

1.1.1.1 Визначаємо вихідну різницю $\Delta y = E_k(x) \oplus E_k(x \oplus \Delta x)$.

1.1.1.2 Інкрементний елемент таблиці різниці з індексами (Δx, Δy).

2. Значення кожного елементу таблиці ділимо на кількість ключів, тобто на 64.

3. Знаходимо максимальне значення в таблиці різниці MAX.

4. Визначаємо максимальну ймовірність передбачення вихідної різниці як MAX / 2^{16}.

Принциповим моментом в представленому алгоритмі є те, що спочатку виконується усереднення значень таблиці по ключам, а потім визначення максимального значення. Таким чином, отримане значення ймовірності переходу різниці дійсне для будь-якого ключа.

У табл. 4.3 наведені отримані значення максимальних ймовірностей диференціалів для зменшених шифрів з різним числом циклів [128].

У табл. 4.3 кольором виділені випадки, коли значення ймовірності диференціала нижче, ніж 2^{-15} і шифр може вважатися захищеним від диференціального криптоаналізу. Наведені дані свідчать про те, що підвищення характеристик лінійного або нелінійного перетворень дозволяють отримати захищений шифр приблизно на один цикл раніше.

На рис. 4.4 Динаміка верхніх кордону ймовірностей для розглянутих шифрів представлена у вигляді графіків.

Тепер розглянемо, які результати для розглянутих шифрів дасть застосування інших методів оцінювання ймовірностей диференціалів.

Перший метод – це оцінка верхньої межі ймовірності 2-циклових диференціалів на основі теореми 4.1 з [124].

Другий метод – оцінка верхньої межі ймовірності 4-циклових диференціалів на основі теореми 2 з [125].

Третій метод – метод оцінка верхньої межі ймовірності 4-циклових диференціалів, який був використаний в [134].

Таблиця 4.3 – Максимальні ймовірності диференціалів при усередненні для 64 випадково вибраних ключів

Шифри	Цикли					
	1	2	3	4	5	6
Mini-aes 1 (sbox_prob = 2^{-2} 16 bit MixCol)	$16384*2^{-16}$	$66*2^{-16}$	$1,47*2^{-16}$	$1,02*2^{-16}$	$1,04*2^{-16}$	$1,28*2^{-16}$
Mini-aes 2 (sbox_prob = 2^{-1} 16 bit MixCol)	$32768*2^{-16}$	$874*2^{-16}$	$27,6*2^{-16}$	$1,66*2^{-16}$	$1,47*2^{-16}$	$1,01*2^{-16}$
Mini-aes 3 (sbox_prob = 2^{-2} 8 bit MixCol)	$16384*2^{-16}$	$1152*2^{-16}$	$124*2^{-16}$	$1,56*2^{-16}$	$1,22*2^{-16}$	$1,19*2^{-16}$
Mini-aes 4 (sbox_prob = 2^{-1} 8 bit MixCol)	$32768*2^{-16}$	$5344*2^{-16}$	$249*2^{-16}$	$3,28*2^{-16}$	$1,13*2^{-16}$	$1,06*2^{-16}$

У табл. 4.4 і 4.5 наведені результати для зменшених версій і для повнорозмірних шифрів, відповідно, отримані за допомогою трьох перерахованих методів (методи позначені як метод 1, метод 2 і метод 3).

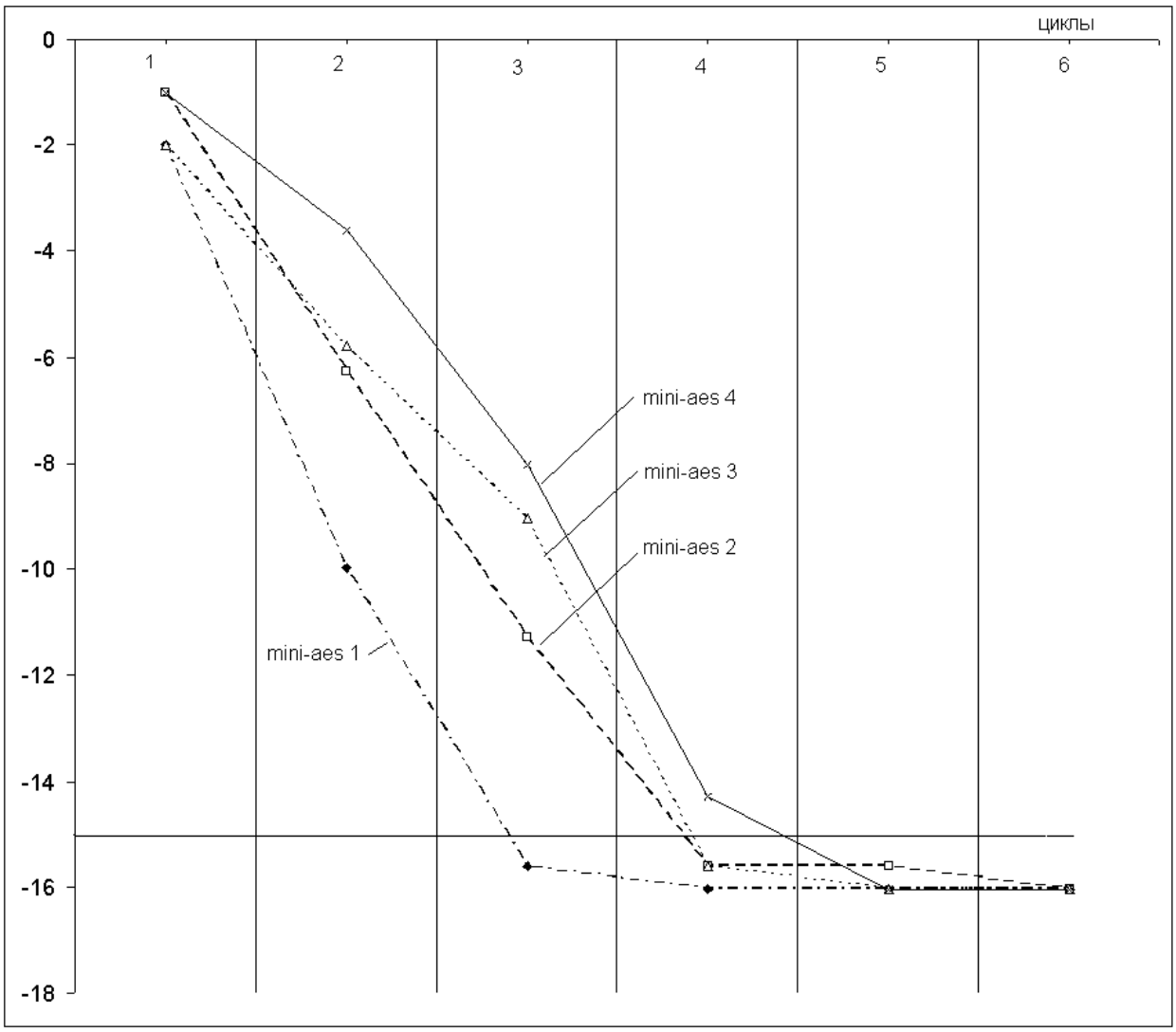


Рисунок 4.4 – Двійковий логарифм верхніх меж ймовірностей диференціалів

Порівнюючи представлені в табл. 4.3 і 4.4 дані видно, що якщо теоретичні оцінки для ймовірностей двоциклову диференціалів близькі до точних, хоча теж можуть бути уточнені, то верхня межа для ймовірностей 4-циклових диференціалів, отримані з використанням методу 2 і методу 3, є дуже неточними. Аналогічна ситуація і з даними з табл. 4.5 для 128-бітових шифрів.

Таблиця 4.4 – Ймовірності диференціалів для 16-бітних шифрів

Назва шифру	Метод 1 (2 цикли)	Метод 2 (4 цикли)	Метод 3 (4 цикли)
Mini-aes 1 (sbox_prob=2^{-2} 16 bit MixCol)	$72*2^{-16}$	$72*2^{-16}$	
Mini-aes 2 (sbox_prob = 2^{-1} 16 bit MixCol)	$2^{11}*2^{-16}$	$2^{11}*2^{-16}$	
Mini-aes 3 (sbox_prob = 2^{-2} 8 bit MixCol)	$1536*2^{-16}$	$36*2^{-16}$	$4,68*2^{-16}$
Mini-aes 4 (sbox_prob = 2^{-1} 8 bit MixCol)	$2^{13}*2^{-16}$	$2^{10}*2^{-16}$	$61,36*2^{-16}$

Таблиця 4.5 – Ймовірності диференціалів для 128-бітних шифрів

Назва шифру	Метод 1 (2 цикли)	Метод 2 (4 цикли)
Aes (sbox_prob = 2^{-6} 64 bit MixCol)	$2^{74}*2^{-128}$	$2^{20}*2^{-128}$
Aes (sbox_prob = 2^{-5} 64 bit MixCol)	$2^{83}*2^{-128}$	$2^{38}*2^{-128}$
Aes(sbox_prob = 2^{-6} 32 bit MixCol)	$2^{99,7}*2^{-128}$	$2^{14,8}*2^{-128}$
Aes (sbox_prob = 2^{-5} 32 bit MixCol)	$2^{103}*2^{-128}$	$2^{28}*2^{-128}$

Таким чином, незалежно від оцінок, отриманих з використанням методів 2 і 3 для 4 циклових диференціалів, можна очікувати, що всі

розглянуті модифікації 128 бітових шифрів, також як і модифікації 16-бітових шифрів, будуть захищені від диференціальних атак після 5-6 циклів.

Використання підстановок з оптимальними характеристиками або зміна структури перетворення MixColumn дозволяють отримати захищений шифр максимум на 1 цикл раніше.

Існуючі на сьогодні методи оцінювання максимальних ймовірностей диференціалів не дозволяють отримувати точні оцінки для більш, ніж 2 циклів шифру Rijndael, отже, питання залишається відкритим.

4.2.2 Атака усічених байтових диференціалів

4.2.2.1 Основні терміни та визначення

В ході атаки усічених байтових диференціалів [135,136] через перетворення шифру намагаються провести вектора активізації. Кожен біт вектора активізації відображає активність одного байта в звичайній різниці. Таким чином, вектор активізації містить стільки бітів, скільки байтів в блоці, а значення біта визначається активністю байта: «1» – байт активний, «0» – байт пасивний.

Сукупність значень вхідного і вихідного векторів активізації для одного циклу перетворень називається одноцикловою байтовою диференціальною характеристикою (БДХ). За аналогією зі звичайним диференціальним криптоаналізом «зшивання» кількох одноциклових БДХ (умова «зшивання»: вхідний вектор активізації кожної наступної одноциклової БДХ дорівнює вихідному вектору активізації попередньої) будемо називати багатоцикловою БДХ. Імовірність такої характеристики обчислюється як добуток ймовірностей всіх вхідних в неї одноціклових характеристик. БДХ, що покривають однакове число циклів і мають однакові значення вхідних векторів активізації і однакові значення вихідних векторів активізації, належать одному і тому ж байтовому диференціалу (БД). Імовірність БД є сума ймовірностей всіх вхідних в нього БДХ. Рис. 4.5

пояснює терміни БДХ, БД і порядок обчислення їх ймовірностей *Р(БДХ)* і *Р(БД)*.

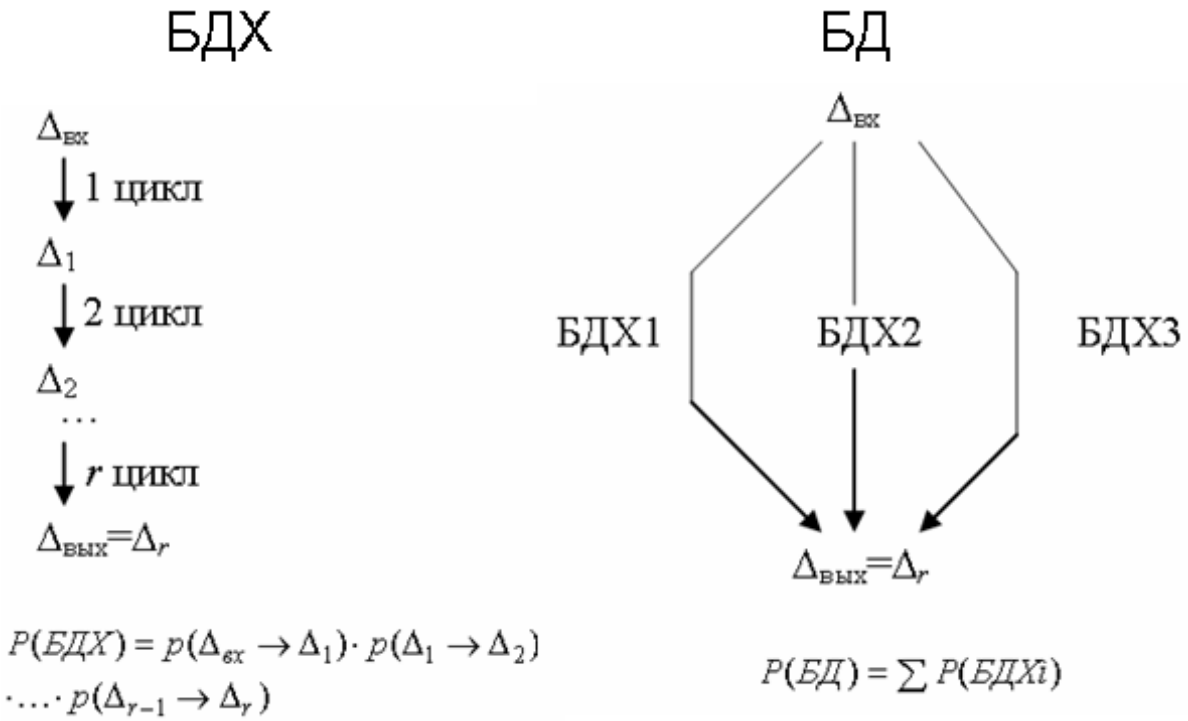


Рисунок 4.5 – БДХ та БД

Слід зауважити, що при такому підході до аналізу проходження байтової різниці основна невизначеність в зміні значення векторів активізації припадатиме на лінійні перетворення розсіювання. Наприклад, для Rijndael-подібних перетворень з ймовірністю 1 відомо як змінюється активність байтів при виконанні операцій ByteSub, AddKey і ShiftRow, а невизначеність виникає тільки при виконанні MixColumn. В роботі [137] був представлений підхід до визначення ймовірностей переходу векторів активізації через це перетворення. Ці ймовірності представлені в табл. 4.6 і 4.7 для різних варіантів перетворення MixColumn.

Таблиця 4.6 – Log_2 ймовірності переходу вектора активізації через 4-байтний MixColumn

Выход / Вход	0	1	2	3	4
0	0	-	-	-	-
1	-	-	-	-	0
2	-	-	-	-7,99	-0,023
3	-	-	-15,99	-8,017	-0,0226
4	-	-23,983	-16,0115	-8,0171	-0,0226

Таблиця 4.7 – Log_2 ймовірності переходу вектора активізації через 8-байтний MixColumn

Вих. / Вх.	0	1	2	3	4	5	6	7	8
0	0	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	0
2	-	-	-	-	-	-	-	-7,99	-0,046
3	-	-	-	-	-	-	-15,9	-8,04	-0,045
4	-	-	-	-	-	-23,9	-16,0	-8,04	-0,045
5	-	-	-	-	-31,9	-24,0	-16,0	-8,04	-0,045
6	-	-	-	-39,9	-32,0	-24,0	-16,0	-8,04	-0,045
7	-	-	-47,9	-40,0	-32,0	-24,0	-16,0	-8,04	-0,045
8	-	-55,9	-48,0	-40,0	-32,0	-24,0	-16,0	-8,04	-0,045

У табл. 4.6 і 4.7 по рядкам змінюється кількість активних байтів у вхідній різниці, а по стовпцях – у вихідний різниці.

Перетворення MixColumn, яке виконує обробку кожної окремої колонки блоку даних, на наш погляд, може бути названо перетворенням з випадковими усіченими байтовими диференціалами [138].

Визначення 4.1. Байт-орієнтоване перетворення покриває m байтів $Y = g(X)$ ($X = (X_1, X_2, ..., X_m) \in GF(2^8)^m$, $Y = (Y_1, Y_2, ..., Y_m) \in GF(2^8)^m$) з числом гілок активізації B називається перетворенням з випадковими усіченими байтовими диференціалами, якщо усереднена для всіх можливих значень $\Delta X \in GF(2^8)^m$ ймовірність переходу вхідного вектора активізації $\alpha = \chi(\Delta X)$ в вихідний вектор активізації $\beta = \chi(\Delta Y)$ може бути обчислена так:

$$P_{БД}(\alpha \to \beta) \approx (2^{-8})^{m-h(\beta)}, \text{ якщо } 2m \geq (h(\alpha) + h(\beta)) \geq B;$$

$$P_{БД}(\alpha \to \beta) = 0, \text{ якщо } (h(\alpha) + h(\beta)) < B;$$

$$P_{БД}(\alpha \to \beta) = 1, \text{ якщо } h(\alpha) = h(\beta) = 0,$$

де $h(f)$ – вага Хеммінга аргументу f;

χ – функція-характеристика, ставить 1 на місці активних S-блоків в аргументі і 0 – в іншому випадку;

$\Delta X = (\Delta X_1, \Delta X_2, ..., \Delta X_m)$ та $\Delta Y = (\Delta Y_1, \Delta Y_2, ..., \Delta Y_m)$ – відповідно, вхідна і вихідна різниці.

Як буде показано пізніше, наявність в цикловом перетворенні шифру такого перетворення грає важливу роль в забезпеченні стійкості до атаки усічених диференціалів.

Нагадаємо також, що БДХ або БД вважаються ефективними, коли їх ймовірність $p_{БДХ}$ або $P_{БД}$ значно більше ймовірності отримання на виході того ж вектора активізації при довільному (випадковому) векторі активізації на вході (випадковий вхідний вектор активізації передбачає рівноймовірність всіх значень вихідної різниці):

$$P_{БД} >> p_{сл}; p_{БДХ} >> p_{сл}, \tag{4.7}$$

де $p_{сл} \approx (2^{-8})^u$ – число неактивних байтів у вихідний різниці або число нульових бітів у вихідному векторі активізації.

Слід зауважити, що для ефективних БД або БДХ неодмінно буде виконуватися і традиційне для звичайних диференціалів обмеження: РБД> $P_{БД} > 2^{-n}$ або $p_{БДХ} > 2^{-n}$, де n – довжина блоку в бітах.

Якщо вдається знайти ефективний диференціал, то зазвичай на останньому циклі, де відомі вихідне значення різниці (на основі відомих значень криптограми) і вхідний вектор активізації (відповідно до використовуваним байтовим диференціалом). Ця інформація дозволяє отримати інформацію про підключ останнього циклу. Таким чином, для виконання атаки необхідно, щоб БД або БДХ покривали майже все цикли шифру.

4.2.2.2 Методи оцінювання стійкості до атаки усічених диференціалів, засновані на розгляді БДХ

Один з методів оцінювання стійкості Фейстель-подібних БСШ до атаки усічених диференціалів запропонований в [139] для тестування шифру E2. Метод заснований на пошуку ефективних байтових диференціальних характеристик (БДХ) для Фейстель-подібних шифрів. Нагадаємо основні етапи цього методу для 128-бітного Фейстель-подібного БСШ:

0) На передобчислювальному етапі формується таблиця, яка відображає ймовірності переходів різних векторів активізації на вході 64-бітної (8-байтной) шифрувальної функції в різні комбінації активних байтів на її виході. Розмір такої таблиці $2^8 \times 2^8$ осередків пам'яті (2^8 можливих варіантів векторів активізації S-блоків на вході і стільки ж на виході). Потім, в кожному рядку, що відповідає певному вхідному вектору активізації, вихідні вектори активізації упорядковано за зменшенням ймовірності переходу в них.

1) Для кожного вхідного вектора активізації (2^{16} варіантів) викликається процедура r_roundf.

2) r_roundf: З таблиці, отриманої на передобчислювальному етапі, в порядку зменшення ймовірності вибираються можливі значення вихідного

вектора активізації. Якщо результуюча ймовірність, з урахуванням ймовірності поточного переходу менше, ніж 2^{-128}, то вибирається таке значення (поточний відкидається). Якщо ж ймовірність більше, ніж 2^{-128}, то викликається процедура r_xor.

3) r_xor: При додаванні по XOR двох 8-байтових напівблоків, відповідні їм біти активізації, складаються за правилами, наведеними в табл. 4.8.

Таблиця 4.8 – Правила складання бітів активізації

Знач. 1-го біту	Знач. 2-го біту	Результат	Ймовірність
0	0	0	1
0	1	1	1
1	0	1	1
1	1	0	1/255
		1	254/255

Для кожного з можливих результатів операції XOR (число можливих результатів одно 2^a, де a – число випадків, коли складаються активні байти) перевіряються дві умови:

а) результуюча ймовірність $p_{rez} < 2^{-128}$;

б) результуюча ймовірність $p_{rez} < 2^{-8c}$, де c – число неактивних байтів у вихідний різниці.

Якщо хоча б одна з умов виконується – вибирається таке значення результату операції XOR (поточне значення відкидається).

Якщо номер поточного циклу дорівнює числу циклів R, то знайдена характеристика записується в файл, якщо немає, – викликається процедура r_roundf.

Даний метод заснований на переборі всіх можливих вихідних векторів активізації в кожному циклі, але оскільки проводиться пошук БДХ, а не БД, результуюча ймовірність може контролюватися в ході побудови БДХ. Якщо

після будь-якого циклу ця ймовірність буде нижче порогового значення, то такий варіант відразу відкидається. Таким чином, якщо серед всіх *r*-циклових БДХ немає ефективних, то пошук характеристик, що покривають більшу кількість циклів, не призведе до збільшення обчислювальної складності цього алгоритму.

Метод чутливий до збільшення розміру блоку. Якщо при пошуку *r*-циклових ефективних БДХ для 128-бітного Фейстель-подібного шифру в граничному випадку (коли в кожному циклі можливі переходи в усі вихідні вектора активізації) доводиться розглядати 2^{8r} варіантів БДХ, то вже для 256-бітного шифру аналогічний пошук зажадає розгляду 2^{16r} варіантів.

В роботі [123] було запропоновано вдосконалений метод пошуку ефективних байтових диференціальних характеристик (БДХ) для БСШ з Rijndael-подібними перетвореннями, який на відміну від відомої стратегії переборного типу орієнтований на виділення і розгляд в кожному циклі найбільш «перспективних» переходів різниці. Це дозволило розширити сферу застосування методу і проаналізувати з його допомогою шифри з розміром блоку до 1024 бітів.

Також в роботі [123] доведено наступні твердження і теорема, які будуть корисні в подальших дослідженнях.

Твердження 4.1 (твердження 3.4 в [123]). Ефективна БДХ Rijndael-подібного SPN-шифру не може містити жодного циклу, в якому активні всі колонки.

Теорема 4.2 (теорема 3.2 в [123]). Для Rijndael-подібного SPN-шифру, блок якого містить n_c n_b-байтових колонок ($n_c < n_b$ та n_c ділить націло n_b) не існує ефективних БДХ для більш, ніж 3 повних циклів.

Загальний недолік розглянутих в цьому пункті методів – низька достовірність, тобто, недостатня гарантованість, реальної стійкості шифру навіть при отриманні позитивного результату. Це є наслідком того, що розглядаються ймовірності БДХ, а не БД.

4.2.2.3 Відомі методи оцінювання стійкості до атаки усічених диференціалів, засновані на розгляді БД

В роботі [96] запропонована наступна рекурсивна процедура обчислення ймовірності БД для SPN-шифру:

$$P_{БД}^{(i)}(\beta'(i),\beta'(0)) = \sum_{\beta'(i-1)} N(F,\beta'(i),\beta'(i-1)) \cdot p^{h(\beta'(i-1))} \cdot P_{БД}^{(i-1)}(\beta'(i-1),\beta'(0)), \quad (4.8)$$

де $p = \frac{1}{2^n - 1}$, n – розмірність використовуваних S-блоків в бітах;

$\beta'(i) = \chi(\beta(i))$, $\beta(i)$ – можлива диференціальна різниця на виході i-го нелінійного рівня.

Значення функції $N(F,\gamma,\delta)$ для циклового перетворення F визначається так:

$$N(F,\gamma,\delta) = \#\{(\Delta X,\Delta Y)/\{0,0\} \mid \Delta Y = F(\Delta X),\ \chi(\Delta X) = \gamma, \chi(\Delta Y) = \delta\}.$$

В останній формулі ΔX – вхідне в циклову шифруючу функцію значення різниці; ΔY – значення різниці на виході циклової шифрувальної функції.

Процедура обчислення ймовірності БД для Фейстель-подібних шифрів також представлена в роботі [96]. Вона дещо відрізняється від представленої процедури для SPN-шифрів, однак основна ідея, як і раніше, полягає в переборі всіх можливих вихідних векторів активізації в кожному циклі.

В роботі [96] показано, що ймовірність усіченого диференціала пов'язана з ймовірністю відповідного звичайного диференціала. Для цього введені такі поняття, як перетворення з випадковими диференціалами і шифр з випадковими диференціалами.

Байт-орієнтоване перетворення $Y = g(X,Z)$ ($X = (X_1, X_2,...,X_m) \in GF(2^n)^m$, $Y = (Y_1, Y_2,...,Y_m) \in GF(2^n)^m$, $Z = (Z_1, Z_2,...,Z_{m'}) \in GF(2^n)^{m'}$) називається перетворенням з випадковими диференціалами, якщо для будь-якої вхідної різниці *a* справедливим є такий вираз:

$$P(\Delta Y = \beta \mid \Delta X = a) = p^{h(\chi(\beta))} P(\chi(\Delta Y) = \chi(\beta) \mid \Delta X = a)),$$

де ключі вибираються випадково;

$h(f)$ – вага Хеммінга аргументу *f*;

χ – функція-характеристика, ставить 1 на місці активних *S*-блоків в аргументі і 0 – в іншому випадку;

$p = \dfrac{1}{2^n - 1}$;

$\Delta X = (\Delta X_1, \Delta X_2,...,\Delta X_m)$ і $\Delta Y = (\Delta Y_1, \Delta Y_2,...,\Delta Y_m)$ є, відповідно, вхідна і вихідна різниці.

Байт-орієнтований шифр з циклової функцією $X(i+1) = f(X(i), Z(i+1))\bullet(i = 0,1,...,r-1)$, де $Z(i)\bullet(i = 0,1,...,r-1)$ – підключи, називається шифром з випадковими диференціалами, якщо для будь-якого перетворення з випадковими диференціалами $X(0) = g(X, Z(0))$ похідне перетворення $X(1) = f(g(X, Z(0)), Z(1))$ також є перетворенням з випадковими диференціалами.

Для шифру з випадковими диференціалами ймовірність звичайного диференціала пов'язана з ймовірністю відповідного усіченого диференціала наступним співвідношенням:

$$P^{(i)}(\beta_i, a) = p^{h(\chi(\beta_i))} P_{БД}^{(i)}(\chi(\beta_i), \chi(a)). \qquad (4.9)$$

Для організації пошуку ефективних БД слід проводити перебір всіх вхідних і вихідних векторів активізації і для кожного варіанту, використовуючи формулу (4.8) обчислювати ймовірність. В результаті будуть знайдені байтові диференціали, які можуть бути покладені як в основу диференціальної атаки або атаки усічених диференціалів, так і в основу атаки нездійсненних диференціалів.

Основним недоліком запропонованого в роботі [96] підходу є високі обчислювальні витрати алгоритму, що реалізує обчислення за формулою (4.8). Обчислювальні витрати можна виміряти кількістю циклових перетворень, для яких виконується обчислення ймовірності одноциклової БДХ. У граничному випадку для оцінки ймовірності *r*-циклового БД для SPN-шифру з m S-блоками кількість розглянутих одноциклових байтових характеристик складе $\left(2^m\right)^{r-1}$. Внаслідок експоненційної залежності між числом циклів і складністю, практично неможливо оцінити ймовірності диференціалів з великим числом циклом. Природно, що процедура пошуку ефективних БД на основі такого методу обчислення ймовірностей зажадає великих обчислювальних витрат. У граничному випадку пошук *r*-циклових БД вимагатиме обчислення ймовірності для $\left(2^m\right)^{r-1} \cdot 2^m \cdot 2^m = \left(2^m\right)^{r+1}$ одноциклових характеристик, так як існує 2^m варіантів вхідних векторів активізації і стільки ж вихідних. Отже, процедура пошуку буде застосовна на практиці для шифрів з ще меншим числом циклів, ніж процедура оцінки ймовірності окремого БД.

Основний недолік відомого методу оцінювання стійкості до атаки усічених диференціалів, заснованого на розгляді БД, – висока складність, що істотно обмежує область його використання. Обчислювальна складність відомого методу експоненціально зростає зі збільшенням розміру блоку і кількості циклів. Як наслідок, метод стає непридатним на практиці для розміру блоку 256 і більше бітів.

4.2.3 Атака нездійсненних диференціалів

4.2.3.1 Загальна характеристика атаки. Основні терміни та визначення

Атака нездійсненних диференціалів (НД) є одним з найбільш ефективних нападів на сучасні блокові симетричні шифри (БСШ). Цей криптоаналітичних метод успішно дозволяє атакувати як SPN-шифри [95,140,141], так і шифри, побудовані з використанням ланцюга Фейстеля і інших структур [142-146]. Підтвердженням сказаного є велика кількість робіт, що з'явилися за останнє десятиліття і спрямованих, головним чином, на пошук нездійсненних диференціалів [95,140-149]. Для шифру Rijndael з зменшеною кількістю циклів дану атаку можна вважати однією з найуспішніших.

Атака НД вперше була запропонована Е. Біхамом в [150,139] для шифрів SkipJack, IDEA, Khufu. Пізніше виявилося, що атака НД застосовна і для інших шифрів, в тому числі і для шифру AES з 5 циклами [85].

Атака НД на блокові симетричні шифри, як більшість криптоаналітичних нападів, відноситься до класу атак на цикловую функцію, і для її реалізації необхідно мати деяку кількість пар відкритий текст-криптограма, отриманих на одному і тому ж секретному ключі.

Дана криптоаналітичних методика називається атакою нездійсненних диференціалів, оскільки в атаці використовуються диференціали спеціального виду – ті, які не можуть виконатися, тобто мають нульову ймовірність. Атака нездійсненних диференціалів на r-цикловий шифр зазвичай стає можливою, коли є $(r-1)$-цикловий нездійсненний диференціал.

На рис. 4.6 представлена схема виконання атаки НД.

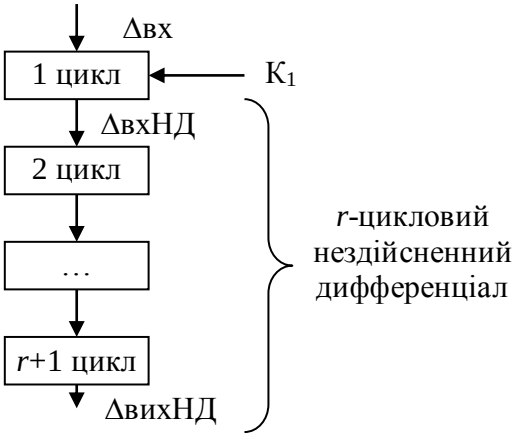


Рисунок 4.6 – Схема атаки НД

При наявності *r*-циклового НД із вхідною різницею $\Delta_{\text{вхНД}}$ і вихідною різницею $\Delta_{\text{вихНД}}$ атака на (*r*+1)-цикловий шифр складається з наступних кроків. Виконується пошук пари з деякою вхідний різницею $\Delta_{\text{вх}}$ і вихідний різницею $\Delta_{\text{вихНД}}$. При цьому перехід $\Delta_{\text{вх}}$ в $\Delta_{\text{вхНД}}$ на першому циклі повинен бути можливий лише для деяких значень ключа першого циклу K_1. Якщо така пара знайдена, то, відповідно до НД, після першого циклу не могла бути різниця $\Delta_{\text{вхНД}}$. І все ключі першого циклу, які будуть призводити до цієї різниці після одноциклового шифрування, є невірними. Шляхом відсіву всіх невірних ключів визначається правильний з'єднання першого циклу K_1.

Один з варіантів атаки – атака байтових або усічених нездійсненних диференціалів (БНД) – була запропонована в роботах [96, 140, 141]. В ході атаки через перетворення шифру намагаються провести вектора активізації. Кожен біт вектора активізації відображає активність одного байта в звичайній різниці. Таким чином, вектор активізації містить стільки бітів, скільки байтів в блоці, а значення біта визначається активністю байта: «1» – байт активний, «0» – байт пасивний.

Перевагою БНД перед просто НД є те, що кожна знайдена правильна пара дозволяє відсіяти не один або кілька неправильних ключів першого циклу, а відразу кілька сотень або тисяч неправильних ключів першого

циклу. В іншій частині роботи будемо, головним чином, обговорювати байтові НД (БНД).

4.2.3.2 Особливості реалізації атаки нездійсненних диференціалів на шифр Rijndael-128 зі зменшеним числом циклів

Використовуючи матеріали роботи [85], викладемо більш детально основні моменти організації атаки нездійсненних диференціалів на шифр Rijndael.

В основі відомої атаки на 5-циклової шифр лежить 4-цикловий нездійсненний диференціал, який використовується для циклів з 2-го по 5-й (рис. 4.7).

На рис. 4.7 зафарбовані клітинки позначають наявність у відповідному байті різниці, відмінною від нуля, незафарбовані осередку позначають нульову різницю.

При проходженні різниці через перетворення шифру позиції активних байтів змінюються при виконанні перетворень *ShiftRow* і *MixColumn* (на рис. 4.7 позначається як SR і MC).

*Я*кщо *ShiftRow* просто змінює позиції активних байтів, то *MixColumn* змінює як позиції, так і кількість активних байтів в рамках кожної колонки таким чином, що сумарне число активних байтів до і після цього перетворення в будь-який колонці не менше 5 (крім випадку, коли різниця на вході в перетворення не містить жодного активного байта в колонці – тоді вихідна різниця також не містити активних байтів). Випадок, коли на вхід операції *MixColumn* надходить 4-байтовая колонка з одним активним байтом, розглянуто при описі інтегральної атаки, і як вже було показано, в цьому випадку на виході перетворення завжди буде виходити 4 активних байта.

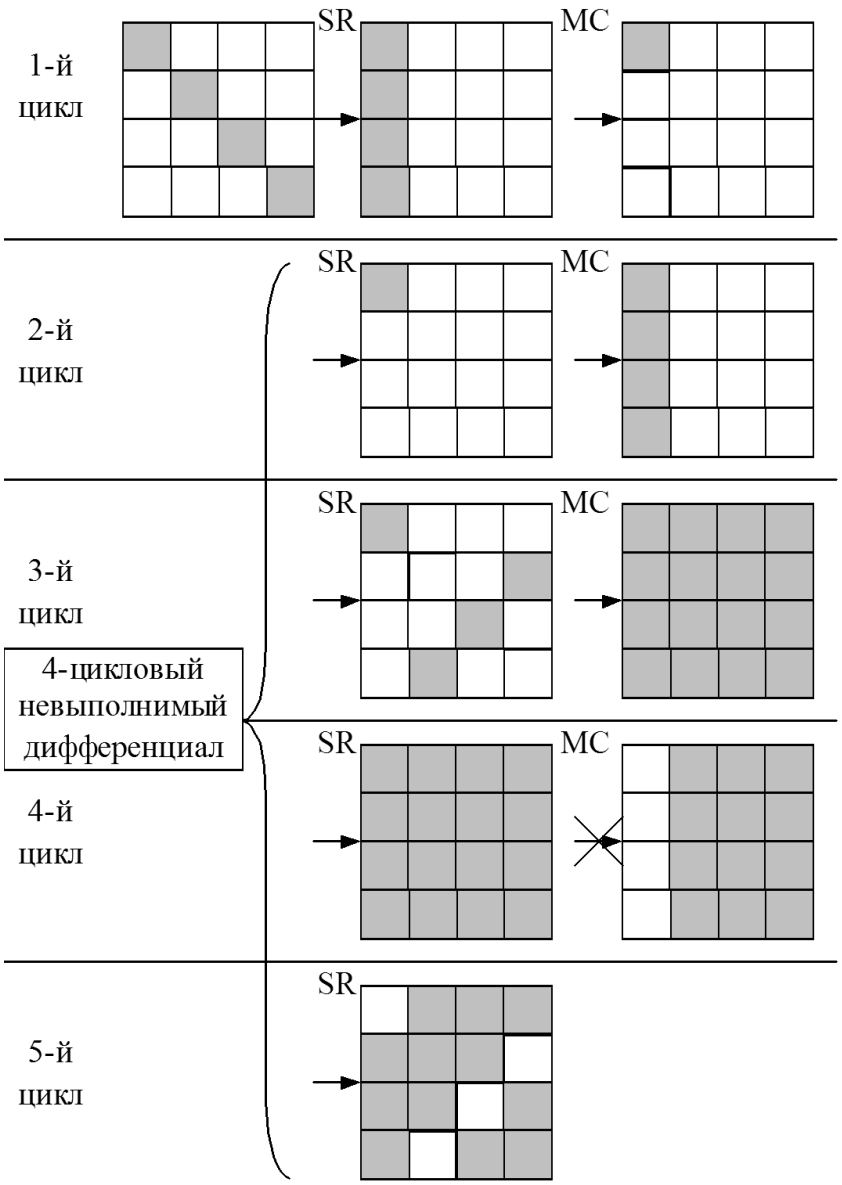


Рисунок 4.7 – Схема атаки НД на 5-цикловий Rijndael-128

Відповідно до наведеного на рис. 4.7 4-цикловим нездійсненним диференціалом, якщо на вході є різниця в одному байті (будь-якому з чотирьох), то після чотирьох циклів не може бути отримана різниця з нульовими значеннями хоча б в одній з комбінацій байтів 0,7,10,13; 1,4,11,14; 2,5,8,15; 3,6,9,12 (кожна комбінація відповідає одній пасивної колонці після операції *MixColumn* 4-го циклу). В ході виконання атаки розглядаються пари відкритих текстів, які мають заданим вхідним значенням різниці (початкове значення на рис. 4.7). Ці пари зашифровуються 5-цикловим алгоритмом. Якщо різниця будь-якої пари криптограми виявляється рівною нулю в одній з

перерахованих комбінацій байтів, то це означає, що після першого циклу не могла бути отримана різниця з єдиним активним байтом, і тому підключи першого циклу, які дозволяють отримати один активний байт після першого циклу, є невірними. Ці варіанти ключів знаходяться традиційним для диференціального криптоаналізу шляхом, розглядаючи послідовно всі можливі варіанти різниці на виході першого циклу, в яких активний один байт.

В ході досліджень нами була запропонована атака на 4-цикловий шифр, яка вимагає значно менших обчислювальних витрат і витрат пам'яті і тому може бути реалізована на практиці. Показники відомої і пропонованої атак представлені в табл. 4.9 [151].

Таблиця 4.9 – Показники відомої і пропонованої атак НД

Кількість циклів	Необхідні ресурси		
	Витрати на обчислення, опер. шифрування	Пам'ять, байти	Кількість пар відкр. текст-криптограма
4	2^{33}	2^{29}	2^{16}
5	2^{36}	2^{42}	$2^{29,5}$

Схема пропонованої атаки представлена на рис. 2.5.

У запропонованій атаці використовується 3-цикловий БНД, який представлений на рис. 4.8. Якщо на вхід 2-го циклу надходить різниця з одним активним байтом (цей байт може перебувати на будь-якій позиції), то після 4-го циклу все байти повинні бути активними, тобто повинні містити ненульову різницю, отже, вихідна різниця хоча б з одним пасивним байтом не може мати місце.

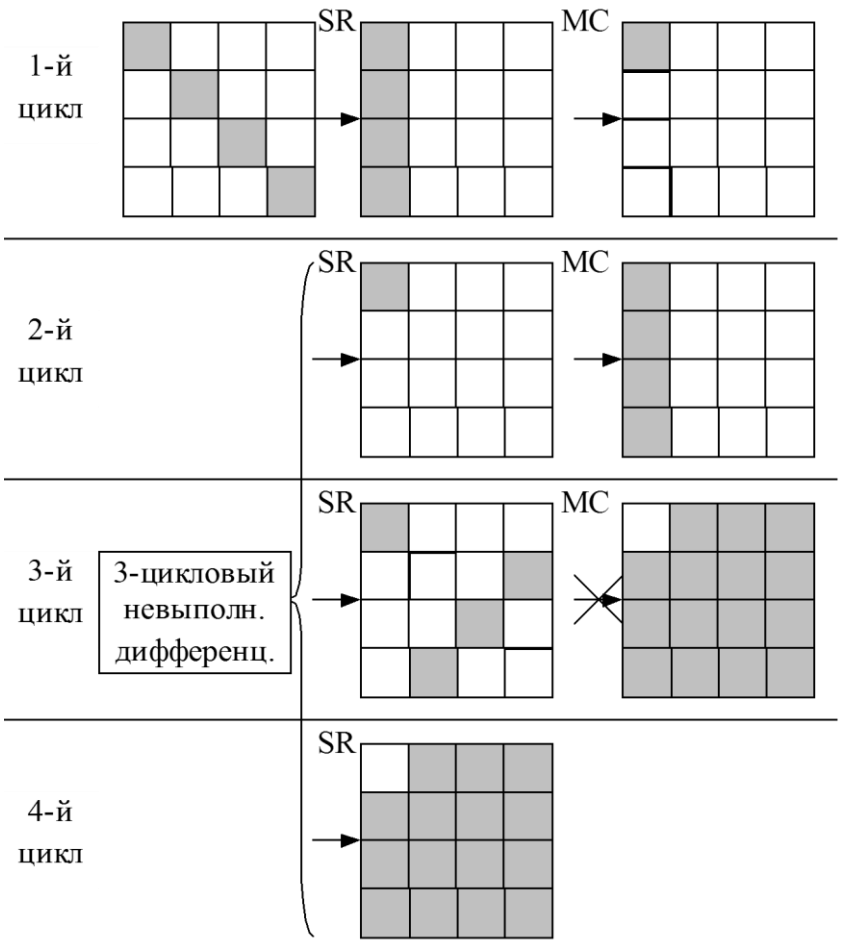


Рисунок 4.8 – Схема пропонованої атаки НД

В ході атаки розглядаються пари відкритих текстів, що володіють заданою вхідною різницею (активні байти розташовані по головній діагоналі (рис. 4.8)). Для цих відкритих текстів виробляються криптограми з використанням 4-циклового алгоритму шифрування. І якщо різниця криптограми дорівнює нулю хоча б в одному з байтів, то це свідчить про те, що на вході другого циклу не могла бути різниця з тільки одним активним байтом. Тому все підключи першого циклу, які призводять до такої ситуації, слід відкинути як невірні. Метою атаки є знаходження і виключення всіх хибних ключів, в результаті повинен залишитися тільки один варіант 32-бітного фрагмента підключа першого циклу, який і буде вірним.

Виконані розрахунки показали, що для виключення всіх неправильних 32-бітних фрагментів ключа необхідно проаналізувати близько 2^{16} відкритих текстів, з яких може бути складено приблизно $(2^{16})^2/2 = 2^{31}$ пар з потрібною

вхідною різницею і знайдено близько $\frac{2^{31}}{16} = 2^{27}$ пар, які мають при цьому і необхідною вихідною різницею. Таким чином, сценарій пропонованої атаки складається з наступних етапів:

1. Формується 2^{16} специфічних відкритих текстів і за допомогою 4-циклового алгоритму виробляються відповідні їм криптограми.

2. Серед криптограм знаходяться такі пари, різниця яких дорівнює 0 хоча б в одному байті.

3. Послідовно розглядаються 2^{10} варіанти різниці з одним активним байтом на виході першого циклу, для кожного варіанта знаходяться ключі, які для даних відкритих текстів дозволяють отримати таку різницю після першого циклу. Всі знайдені на 3-му кроці 32-бітові фрагменти підключа першого циклу виключаються з повної безлічі 2^{32} варіантів.

В результаті атаки необхідно виключити всі неправильні 32-бітові фрагменти підключа першого циклу.

Відповідно до описаної методикою розроблена програмна реалізація атаки на шифр Rijndael с 4-ма циклами. Атака реалізована у вигляді двох додатків. Перше додаток, використовуючи 16-байтовий ключ з файлу, дозволяє набрати необхідну кількість пар відкритий текст-криптограма і зберегти їх в файлах. Другий додаток зчитує відкриті тексти і криптограми з відповідних файлів і діючи далі по викладеній методиці починає знаходити невірні варіанти 32-бітного фрагмента підключа першого циклу. Кожен знайдений невірний варіант фрагмента фіксується одиничним значенням в біті, який відповідає цьому значенню в масиві з 2^{32} бітів. З певним інтервалом масив з 2^{32} бітів (2^{29} байтів) зберігається в файлі.

В ході обчислювального експерименту додаток, яке визначає невірні 32-бітові фрагменти підключа першого циклу, відпрацювало близько 20 годин і за цей час виявило близько 30 000 000 $\approx 2^{24,8}$ невірних варіантів 32-бітного фрагмента, проаналізувавши при цьому ≈29000 пар відкритих текстів, які мають необхідної вхідний різницею і чиї криптограми містять

нульову різницю хоча б в одному байті. Від загального числа таких пар, яке необхідно розглянути, це становить $\frac{29000}{2^{27}} \cdot 100\% = 0{,}02\%$. Виходячи з цього, для завершення роботи потрібно 5000 * 20 = 100000 годин.

Обчислювальний експеримент показав, що методика нездійсненних диференціалів є досить ресурсомісткою, оскільки на відміну від криптоаналітичних атак, в яких виконується пошук вірних значень ключа, тут вірне значення знаходиться шляхом виключення всіх невірних значень з досить великого числа варіантів. Тому навіть для шифру з невеликою кількістю циклів, реалізація атаки нездійсненних диференціалів пов'язана з відносно високими обчислювальними витратами і витратами пам'яті.

В ході виконаного обчислювального експерименту було виявлено значну розбіжність між теоретичними і практичними показниками складності даної атаки. На наш погляд, в першу чергу, це можна пояснити тим, що теоретичний розрахунок складності атаки не враховує високої ресурсоємності роботи з масивами великої розмірності.

При цьому результати обчислювального експерименту підтвердили працездатність даної криптоаналітичної методики, яка для багатьох шифрів, в тому числі і для 4 або 5-циклового алгоритму AES, є однією з найбільш ефективних.

Слід також зазначити, що реалізація атаки нездійсненних диференціалів без особливих труднощів може бути розпаралелена для вирішення на кількох ЕОМ, що дозволить знизити часові витрати на реалізацію даної атаки в відповідне число раз.

4.2.3.3 Аналіз відомих методів оцінювання стійкості до атаки НД

а) Відомі НД для структур, які використовуються в БСШ.

Для багатьох структур, які часто використовуються при побудові БСШ, відомі НД. Повною мірою це відноситься до ланцюга Фейстеля. В роботі [149] згадується про те, що якщо в Фейстель-подібному шифрі

використовується бієктивна шифруюча функція, то завжди існує 5-цикловий НД, який має вигляд $(a, 0) \rightarrow (a, 0)$ для будь-якої ненульової різниці a.

З роботи [96] відомо про наявність 4-циклових НД для Rijndael-подібних БСШ зі скороченим останнім циклом. Вхідна різниця в такому НД містить один активний байт, а вихідна – пасивні байти (з нульовою різницею) на позиціях, які відповідають мінімум однієї пасивної колонці (всі байти колонки містять нульову різницю) до перетворення ShiftRow.

Відомий також ряд робіт, присвячених дослідженню НД для різних узагальнених ланцюгів Фейстеля [145,148].

В роботі [147] представлені критерії наявності НД для Rijndael-подібних БСШ з різним числом циклів. Однак немає гарантій того, що якщо критерії не виконуються, то НД немає.

В цілому, якщо в шифрі використовується одна зі структур, для якої відомо про наявність НД, то НД з аналогічною вхідний і вихідний різницями може існувати і для цього шифру. Однак для таких шифрів може існувати і НД для значно більшої кількості циклів, отже, потрібно докладніше дослідження. Так, наприклад, для Фейстель-подібного шифру Camellia, який використовує ланцюг Фейстеля, а значить – існує 5-цикловий НД, в процесі аналізу були знайдені 8-циклові НД [144].

б) Розбіжність посередині (miss-in-the-middle).

В роботі [149] згадується про досить універсальний підхід до побудови НД для БСШ. Підхід полягає в пошуку двох достовірних диференціалів (ймовірність кожного дорівнює 1), перший з яких визначає рух різниці в першій половині шифру в прямому напрямку, а другий – у другій половині шифруючих перетворень в зворотному напрямку. Якщо кінцеві різниці таких достовірних диференціалів не рівні, то це дає НД.

Використовуючи даний підхід побудовано багато з відомих НД, в тому числі і всі представлені в попередньому підрозділі. Даний підхід не рідко використовується для доказу стійкості БСШ до атаки НД, хоча про суворий доказ неможливості побудови НД іншими способами не відомо.

в) Пошук НД повним перебором.

Цікавий підхід до пошуку НД був запропонований в [149]. Для шифру створювалася зменшена модель (зменшений розмір блоку і ключа) і шляхом перебору всіх можливих вхідних різниць і ключів виконувався пошук НД. Потім результати пошуку аналізувалися і виконувалася спроба побудови НД для повнорозмірного шифру.

Основний недолік методу полягає в тому, що властивості зменшеної моделі і повнорозмірного шифру можуть істотно відрізнятися і довести зворотне дуже складно. Тому і структура НД для шифрів теж може мати істотні відмінності, а відсутність або присутність НД для зменшеної моделі не гарантують того ж для повнорозмірного шифру.

г) U та UID методи пошуку НД.

Спроба автоматизувати процес пошуку НД зроблена в роботах [146,147]. Методи діють відповідно до принципу miss-in-the-middle. Шляхом повного перебору різниць виконується пошук достовірних усічених (байтових) диференціалів для обох половин шифруючих перетворень, а потім перевіряється сумісність цих диференціалів. У разі несумісності знайдений НД.

Недолік методів – значне збільшення складності з ростом розміру блоку і числа циклів в шифрі. Для шифрів, які сьогодні використовуються при побудові хеш-функцій (розмір блоку 512 або 1024 біта) ці методи не будуть працювати.

4.3 Вдосконалений метод оцінювання ймовірностей 2-циклових диференціалів для шифрів із довільними диференціальними показниками підстановок

Як було показано в попередньому підрозділі 4.2, для сучасних блокових шифрів виконати точну оцінку верхньої межі ймовірності диференціалів можна лише для невеликої кількості циклів. Для шифру

Rijndael ця кількість циклів дорівнює 2, а відповідний метод запропонований в роботі [152].

Метою цього підрозділу є виклад нашого підходу до оцінки ймовірностей 2-циклових диференціалів, запропонованого в [153-155]. Для шифру Rijndael, отримані результати збігаються з результатами роботи [152], проте в нашому варіанті оцінка ймовірностей 2-циклових диференціалів відбувається на основі аналізу властивостей таблиць різниць S блоків шифру, що робить можливим застосування нашого методу і для шифрів з довільними підстановками [154].

4.3.1 Основні ідеї пропонованого підходу до визначення верхньої межі ймовірностей 2-циклових диференціалів

В ході численних обчислювальних експериментів (досліджувалися супер-S-блоки розміром від 4 до 32 бітів з розміром S-блоків від 2 до 8 бітів) з пошуку 2-циклових диференціалів, що володіють максимальною вірогідністю, було встановлено, що диференціал, що володіє максимальною ймовірністю, завжди містить диференціальну характеристику, яка також має максимальну ймовірність. Виходячи з цього, в основу запропонованого методу покладено такі основні етапи:

- визначення мінімальної кількості активних S-блоків;
- визначення виду ДХ, що володіє максимальною ймовірностю.;
- визначення кількості і ймовірностей додаткових ДХ;
- визначення максимальної ймовірності 2-циклового диференціала.

Вхідними даними є МДР-матриця, на яку виконується множення в ході перетворення MC, і який використовується S-блок разом з таблицею різниці.

Зазначені етапи є цілком зрозумілими, якщо виходити з визначення диференціала. Однак, найбільш проблематичним для реалізації на практиці видається 3 етап, рішень для якого до теперішнього моменту в доступній літературі не було знайдено. У наступних підрозділах буде

продемонстровано, як пропонований підхід може бути реалізований для різних шифруючих перетворень.

4.3.2 Оцінка ймовірностей двоциклову диференціалів шифру Rijndael 128

4.3.2.1 Розглянуті супер-S-блоки

Як і в [152] будемо розглядати супер-S-блок, який складається з послідовності операцій ByteSub, MixColumns, AddKey і ByteSub і працює з однією колонкою блоку даних. Супер-S-блок AES працює з 32-бітовим блоком, і ми будемо розглядати 16-бітний варіант супер-S-блоку, який містить 4 4-бітних S-блоку. Для такого варіанту супер-S-блоку є можливість більш детально вивчити властивості в ході обчислювальних експериментів.

У якості 4-бітових S-блоків взяті підстановки з шифру baby-rijndael [127]. Таблиця підстановки представлена в табл. 4.10.

Таблиця 4.10 – Підстановка 4 в 4 біта

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A	4	3	B	8	E	2	C	5	7	6	F	0	1	9	D

Подібно підстановці оригінального шифру AES, вона складається з інверсії в поле $GF(2^4)$ з утворюючим поліномом 0x13 (табл. 4.11) і подальшого лінійного перетворення (табл. 4.12).

Таблиця 4.11 – Інверсія в полі $GF(2^4)$

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	9	E	D	B	7	6	F	2	C	5	A	4	3	8

Таблиця 4.12 – Лінійна частина підстановки

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A	4	7	9	1	F	C	2	D	3	0	E	6	8	B	5

При проходженні різниці через S-блок можна виділити відображення різниці, що виконується лінійною частиною підстановки табл. 4.13.

Таблиця 4.13 – Відображення різниці лінійної частиною підстановки

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	E	D	3	B	5	6	8	7	9	A	4	C	2	1	F

У перетворенні MixColumns (MC) використовується множення над $GF(2^8)$ на фіксовану матрицю

$$\begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix}.$$

4.3.2.2 Диференціали з 1 активним S-блоком на вході і 4 – на виході. 16-бітний супер-S-блок

Розглянемо супер-S-блок зменшеного AES на вхід якого подається 2^{16} пар відкритих текстів {P, P'}, де P пробігає всі значення від 0 до 2^{16}-1, а P' = P xor $d_{вх}$, де в $d_{вх}$ перша четвірка приймає одне значення з набору {1, ..., 15}, а інші три тетради рівні 0, наприклад нехай в шістнадцятковому представленні $d_{вх}$ = 1000. відповідно до таблиці різниці (табл. 4.14), на виході 1-го S-блоку першого рівня може з'явитися всього 7 значень, при чому одне значення (для $d_{вх}$ = 1000 – це E000) 2^{14} разів, а решта 6 – по 2^{13} разів. На виході трьох інших S-блоків першого рівня завжди нульові різниці.

В силу лінійності перетворення MC, значення різниці на виході MC повністю визначається значенням різниці на вході. Тому на вході S-блоків другого рівня з'явиться 2^{14} разів одне значення різниці (для $d_{вх}$ = 1000 – це FEE1), а ще 6 значень різниці – по 2^{13} разів.

Розглянемо випадки, коли на вході S-блоків другого рівня 2^{14} раз з'являється однакове значення різниці (для $d_{вх}$ = 1000 – це FEE1). Відповідно

до таблиці різниці (табл. 4.14) в кожному рядку є «4», а значить для будь-якого значення різниці на вході S-блоку завжди є вихідне значення, що найбільш часто зустрічається. Отже, є одне значення вихідний різниці $d_{вих}$, яке буде зустрічатися частіше за інших (для різниці на вході до другого рівня S-блоків FEE1 – це $d_{вих}$ = 733E).

Очікувана кількість таких випадків:

$$2^{14}\left(\frac{4}{16}\right)^4 = 2^6,$$

тобто, з 2^{16} вхідних пар з однаковою різницею 2^6 = 64 пари дадуть одне вихідне значення $d_{вих}$ = 733E. Для цих пар все проміжні значення різниці також будуть збігатися, інакше кажучи, вони пройдуть по одному і тому ж «диференціального сліду» (differential trail) або належать одній і тій же диференціальної характеристиці.

Таблиця 4.14 – Таблиця різниці S-блоку зменшеного AES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	2	0	2	0	2	0	2	2	0	0	0	0	4	0
2	0	0	2	2	0	0	0	0	2	4	2	0	2	0	0	2
3	0	4	0	0	2	0	0	2	2	0	2	0	2	2	0	0
4	0	2	4	0	0	2	2	2	0	0	2	0	0	0	0	2
5	0	0	0	0	4	0	2	2	0	2	0	2	2	0	0	2
6	0	0	0	0	0	2	2	0	4	0	2	2	2	0	2	0
7	0	0	0	2	0	0	4	2	2	0	0	0	0	2	2	2
8	0	2	0	2	2	2	0	0	2	0	0	2	0	0	0	4
9	0	2	0	0	0	2	0	0	0	2	0	0	2	4	2	2
A	0	2	2	2	0	0	2	0	0	0	0	2	4	2	0	0
B	0	0	2	2	2	4	0	2	0	0	0	0	2	0	2	0
C	0	0	0	2	2	2	2	0	0	2	4	0	0	2	0	0

D	0	0	2	0	2	0	0	0	0	0	2	4	0	2	2	2
E	0	2	0	4	0	0	0	2	0	2	2	2	0	0	2	0
F	0	0	2	0	0	2	0	4	2	2	0	2	0	2	0	0

Однак отримання зазначеного значення вихідний різниці можливо і при проходженні іншими «диференціальними слідами». Для оцінки кількості таких пар слід розглянути залишилися 6 2^{13}-елементних структур. Кожній структурі відповідає одне значення різниці на виході активного S-блоку. У табл. 4.15 в першій колонці можливі при $d_{вх}$ = 1000 значення різниці на вході перетворення MixColumns (MC) відзначені «+».

У другій колонці представлені відповідні значення різниці на виході перетворення MC.

Кожне з представлених у другій колонці табл. 4.15 значень складається з чотирьох елементів, кожен з яких є витвором над GF(2^4) значення різниці в активному S-блоці на вході в MC на 1, 2 або 3 (відповідно до використовуваної в MC матрицею). Наприклад, значення в першому рядку 2113 – це добуток кількості 1 на зазначені множники: 2 = 1 * 2, 1 = 1 * 1, 1 = 1 * 1, 3 = 1 * 3. Значення у другому рядку – це добуток кількості 2 на ті ж множники і т.д. Оскільки перехід в вихідне значення різниці для основного шляху відбувається з різниці FEE1 через «4» в таблиці різниці для кожного з S-блоків, то значення різниці 733E пов'язано зі значенням FEE1.

Таблиця 4.15 – Можливі значення на вході і виході MC

Вхід MC	Вихід MC	Можл. перех. у 733e
1000 +	2113	0000
2000 +	4226	2222+
3000	6335	0000
4000 +	844c	0000
5000	a55f	0000

6000 +	c66a	0000
7000	e779	2222
8000 +	388b	2222+
9000 +	1998	0000
A000	7AAD	2222
B000	5BBE	2222
C000	BCC7	2222
D000	9DD4	0000
E000 +	FEE1	4444+
F000	DFF2	0000

733E складається з інверсій в GF(2^4) для кожної тетради вхідних значень різниці з подальшим застосуванням лінійної частини підстановки, яка представлена в табл. 4.13. Позначивши перетворення різниці представлене в табл. 4.13 як *L*, можна записати

$7 = L(F^{-1}) = L(8)$ або $7 = L((2*E)^{-1}) = L(2^{-1}*E^{-1}) = L(9*3) = L(8)$;

$3 = L(E^{-1}) = L(3)$;

$E = L(1^{-1}) = L(1)$ або $E = L((3*E)^{-1}) = L(3^{-1}*E^{-1}) = L(E*3) = L(1)$.

Таким чином, коли ми обговорюємо можливість потрапляння різних варіантів різниці на виході MC (друга колонка табл. 4.15) в вихідне значення різниці 733E, то повинні аналізувати ймовірності переходів різниці $2*L$ в $L((2*E)^{-1}) = 7$ для першого S-блоку другого рівня, різниці X в $L((E)^{-1}) = 3$ для другого і третього S-блоків другого рівня і різниці $3*X$ в $L((3*E)^{-1}) = E$ для четвертого S-блоку другого рівня, де X – різниця на виході активного S-блоку першого рівня. Але, відповідно до наслідком 3 (corollary 3) з [152], всі ці три переходи мають однакову ймовірність і тому кожне з представлених у другій колонці табл. 4.15 значень може одночасно для всіх чотирьох S-блоків другого рівня або мати потенційну можливість переходу в вихідну різницю

$d_{вих}$ = 733E (в таблиці різниці на відповідних позиціях «2»), або не мати (в таблиці різниці на відповідних позиціях «0»).

Відповідно до того, що кожна колонка таблиці різниці (як і рядок) крім «0» містить одну «4», шість «2», то крім варіанту для основного шляху (різниця на виході МС FEE1), є ще 6 варіантів різниці на виході МС, які мають потенційну можливість отримати необхідну вихідну різницю. Ці варіанти відзначені значенням «2222», де кожна з чотирьох цифр – значення з таблиці різниці для переходу на відповідному S-блоці в dвих = 733E. Але з огляду на те, що вхідне значення різниці $d_{вх}$ = 1000 дозволяє отримати лише 6 додаткових варіантів різниці після перетворення МС, то потрібна вихідна різниця $d_{вих}$ = 733E може бути отримана тільки двома додатковими шляхами (ці варіанти разом з варіантом основного шляху відзначені «+» в третій колонці табл. 4.15). Для інших варіантів вхідний різниці кількість додаткових шляхів може відрізнятися. Імовірність того, що виявиться *k* додаткових шляхів, еквівалентна ймовірності того, що в числі навмання обраних 6 куль з кошика з 14 кулями, 6 з яких – білі, а 8 – чорні, виявиться *k* білих куль. Ця ймовірність може бути обчислена так

$$P(k \text{ додаткових шляхів}) = \frac{C_6^k \cdot C_{14-6}^{6-k}}{C_{14}^6} = \frac{C_6^k \cdot C_8^{6-k}}{C_{14}^6}.$$

Для даного виду диференціала можливо 15 варіантів вхідної різниці, тому для отримання очікуваної кількості диференціалів з *k* додатковими шляхами необхідно ймовірність помножити на 15. Результати розрахунків представлені в табл. 4.16.

Таблиця 4.16 – Очікувана кількість диференціалів з *k* додатковими шляхами

k	*P*(*k* додаткових шляхів)	Очікувана кількість диференціалів з *k* додатковими шляхами
0	0,0093	0,14
1	0,112	1,67
2	0,35	5,24
3	0,373	5,59
4	0,14	2,1
5	0,016	0,24
6	0,0003	0,005

Дані табл. 4.16 показують на те, що на практиці повинні зустрічатися варіанти з числом додаткових шляхів від 1 до 4, а максимальна кількість додаткових шляхів – 4. В цьому випадку, очікується, що іншими шляхами до тієї ж вихідної різниці прийде ще $4 \cdot 2^{13}\left(\frac{2}{16}\right)^4 = 8$ пар.

Тоді очікуване загальна кількість пар, які при фіксованому $d_{вх}$ дадуть найбільш ймовірне $d_{вих}$ складе 64 + 8 = 72 пари, а ймовірність відповідного диференціала буде $72/2^{16}$.

На підтвердження правильності наших міркувань наводимо результати обчислювальних експериментів (табл. 4.17), з яких видно, що для диференціала з 1 активним S-блоком на вході і 4 – на виході максимальна кількість пар складає 72.

Таблиця 4.17 – Результати обчислювальних експериментів

Вхідна	Вихідна	Кількість пар	Кількість пар	Загальна

різниця, $d_{вх}$	різниця, $d_{вых}$	по основному шляху	по додатк. шляхах	кількість пар
1000	733E	64	4	68
2000	EDDF	64	8	72
3000	9EE1	64	4	68
4000	2778	64	6	70
5000	F22A	64	4	68
6000	1FF5	64	2	66
7000	A88C	64	8	72
8000	B779	64	8	72
9000	DBB2	64	6	70
A000	5AA6	64	8	72
B000	C447	64	4	68
C000	6CCB	64	8	72
D000	4553	64	4	68
E000	8114	64	8	72
F000	366D	64	2	66

4.3.2.3 Диференціали з 1 активним S-блоком на вході і 4 – на виході. 32-бітний супер-S-блок

Застосуємо ті ж міркування для 32-бітного супер-S-блоку повномасштабного AES. На вхід подамо 2^{32} пар відкритих текстів $\{P, P'\}$, де P пробігає всі значення від 0 до 2^{32}-1, а $P' = P \oplus d_{вх}$, де в $d_{вх}$ перший байт приймає одне значення з набору {0, 1, ... , 255}, а решта три байта рівні 0, наприклад, нехай в шістнадцятковому представленні $d_{вх}$ = 01 00 00 00. відповідно до таблиці різниці S-блоку AES, на виході 1-го S-блоку першого рівня може з'явитися всього 127 значень, причому одне значення (для $d_{вх}$ = 01 00 00 00 – це F1 00 00 00) 226 разів, а решта 126 значень (двійки в рядку

таблиці різниці, відповідної вхідної різниці 1) – по 2^{25} разів. На виході трьох інших S-блоків першого рівня завжди нульові різниці.

Подальші міркування для 32-бітного супер-S-блоку аналогічні зменшеному варіанту. У підсумку, очікувана кількість пар, які пройдуть по основному шляху, складе:

$$2^{26}\left(\frac{4}{256}\right)^4 = 2^2 = 4.$$

Імовірність наявності k додаткових шляхів може бути визначена за формулою:

$$P(k \text{ додаткових шляхів}) = \frac{C_{126}^{k} \cdot C_{254-126}^{126-k}}{C_{254}^{126}} = \frac{C_{126}^{k} \cdot C_{128}^{126-k}}{C_{254}^{126}}.$$

Помноживши отримані значення на 255, отримаємо очікувану кількість диференціалів з k додатковими шляхами. Максимальна кількість додаткових шляхів, при якому очікувана кількість диференціалів вище 1 – 72. Це означає, що очікувана кількість пар, які будуть належати тому ж диференціалу, але пройдуть по додаткових шляхах, складе:

$$72 \cdot 2^{25}\left(\frac{2}{256}\right)^4 = 9 \text{ пар.}$$

Очікуване максимальну кількість пар, що належать диференціалу, 4 + 9 = 13, а ймовірність диференціала $13/2^{32}$.

4.3.2.4 Диференціали з 2 і більш активними S-блоками на вході і мінімальною загальною кількістю активних S-блоків

Розглянемо спочатку для 16-бітного супер-S-блоку ситуацію, коли вхідна різниця містить два активних S-блоки, а вихідна різниця – три. Два активних вхідних S-блоки можуть містити як однакові, так і значення різниці, що відрізняються. При цьому і в одному, і в другому випадку, якщо на всіх п'яти S-блоках відбуваються переходи різниці через «4» в таблиці різниці, то кількість пар для основного шляху буде однаковим і складе $2^{16} \cdot \left(\frac{4}{16}\right)^5 = 64$ пари. Відмінності будуть в кількості додаткових шляхів.

Розглянемо ці два варіанти на прикладі двох диференціалів (B700-0D1D) і (BB00-470C), порівняння яких представлено в табл. 4.18.

По основному шляху кожний з диференціалів проходить через «4» в таблиці різниці. Значення різниці для основного шляху виділені в табл. 4.18 жирним шрифтом.

Основна відмінність між двома диференціалами полягає в кількості можливих варіантів різниці на виході 1 рівня S-блоків, які після MC будуть містити 0 на потрібній позиції (3 рядок табл. 4.18). Для диференціала з однаковими значеннями різниці для двох активних S-блоків на вході завжди гарантовано буде 6 таких варіантів різниці (див. 3 рядок табл. 4.18), – ці варіанти містять однакові значення різниці на виходах обох активних S-блоків. Для диференціала з різними значеннями різниці на вході активних S-блоків кількість додаткових варіантів різниці на виході 1 рівня S-блоків, які після MC будуть містити 0 на потрібній позиції, обмежена зверху значенням 6, але на практиці завжди буде менше (наприклад, в розглянутому прикладі їх 2).

Таблиця 4.18 – Порівняння двох диференціалів (B700-0D1D) і (BB00 470C)

	Диференціал B700-0D1D				Диференціал BB00-470C			
Вхідна різниця	B	7	0	0	B	B	0	0
Вих. різниці для кожного S-блоку	2, 3, 4, 5, 7, C, E	3, 6, 7, 8, D, E, F	0	0	2, 3, 4, 5, 7, C, E	2, 3, 4, 5, 7, C, E	0	0
Можливі різниці на вих. 1 рівня S-блоків, які після MC будуть містити 0 на потрібній позиції	(5600), (2D00), (C800)				(5500), (2200), (3300), (4400), (7700), (CC00), (EE00)			
Різниця після MC	(0939)+, (0BFB)-, (0F4F)+				(5F0A)+, (2604)-, (3506)+, (4C08)-, (790E)-, (C70B)+, (E10F)-			
Вихідна різниця	0D1D				470C			

Аналогічно, як і для диференціалів з одним активним S-блоком на вході, для обох розглянутих диференціалів всі значення різниці, отримані на виході MC можуть одночасно для всіх S-блоків другого рівня або мати, або не мати потенційну можливість переходу в вихідне значення різниці. Кількість варіантів, які мають потенційну можливість переходу в вихідну різницю, для першого диференціала залежить від числа можливих варіантів різниці після першого рівня S-блоків (3 рядок табл. 4.18), а для другого диференціала визначається також як і для випадку з 1 активним S-блоком на вході і складе від 1 до 4 для зменшеної моделі супер-S-блоку і від 53 до 72 для 32-бітного супер-S-блоку.

У підсумку, максимальне значення ймовірності для диференціалів з 2 і більш активними S-блоками на вході складе $72/2^{16}$ для 16-бітного супер-S-блоку і $13/2^{32}$ для 32-бітного супер-S-блоку і може бути досягнуто у разі,

коли різниця в активних S-блоках на вході однакова, а загальна кількість активних S-блоків мінімальна, тобто – 5.

Слід зауважити, що вказані значення ймовірностей диференціалів є максимальними при усередненні по всім ключам. Якщо взяти один ключ, то, наприклад, для 16-бітного супер-S-блоку завжди присутній максимальне значення $128/2^{16}$, а іноді і $132/2^{16}$, однак для таких диференціалів, як показують обчислювальні експерименти, буде набагато менше значення на інших ключах і при усередненні по всім ключам ймовірність не буде перевищувати зазначеного вище максимуму.

4.3.2.5 Диференціали із загальною кількістю активних S-блоків 6 і більше

Розглянемо, до чого буде приводити збільшення кількості активних S-блоків. Проаналізуємо диференціал з шістьма активними S-блоками (2 на вході, 4 на виході). Для прикладу візьмемо диференціал BA00-B3D1, який представлений в табл. 4.19.

Таблиця 4.19 – Диференціал BA00-B3D1

	Диференціал BA00-B3D1			
Вхідна різниця	B	A	0	0
Варіанти вих. різниць для кожного S-блоку	2,3,4,5,7,C,E	1,2,3,6,B,C,D	0	0
Різниця на вході в MC	5C00			
Різниця після MC	DE93			
Вихідна різниця	B3D1			

Основний шлях диференціала містить переходи через «4» в таблиці різниці. Значення різниці для основного шляху представлені в табл. 4.19 і виділені жирним шрифтом.

Відразу можна оцінити кількість пар, які пройдуть основним шляхом:

$$2^{16} \cdot \left(\frac{4}{16}\right)^6 = 2^4 = 16 \text{ пар.}$$

Тепер оцінимо кількість додаткових шляхів і очікувана кількість пар, які пройдуть цими шляхами.

Всього, після 1-го рівня S-блоків, можливо 48 додаткових до основного варіантів різниці. З них 12 різниць повторяться по 2^{11} раз (в одному з активних S-блоків перехід різниці з ймовірністю 4/16, у другому – з ймовірністю 2/16), а 36 різниць повторяться по 2^{10} раз (в обох активних S-блоках переходи різниці з ймовірністю 2/16). Для кожного варіанта різниці на виході 1-го S-блоку існує 4 значення різниці на виході 2-го S-блоку, при яких після перетворення МС буде отримана нульова різниця в одній із тетрад і в цьому випадку потрібне значення вихідний різниці $d_{вих}$ ніяк не зможе бути досягнуто. Оскільки серед можливих варіантів виходу кожного з активних S-блоків завжди присутня приблизно половина всіх можливих значень (7 з 15), то в середньому можна очікувати, що в числі варіантів різниць 2-го S-блоку завжди буде присутній 2 з 4 значень різниці, призводять до нульової різниці в одній із тетрад після МС. З огляду на можливість отримання невідповідних різниць після МС, з 48 додаткових варіантів різниці нам підійдуть 8 з 12 варіантів різниці першого типу і 24 з 36 варіантів різниці другого типу.

Для кожного з варіантів різниці, що залишилися, ймовірність того, що існує потенційна можливість отримання потрібної вихідної різниці (в таблиці різниці на потрібних позиціях не "0") складе $\left(\frac{7}{15}\right)^4 \approx 2^{-4}$.

При наявності потенційної можливості отримання потрібної вихідної різниці переходи різниці для всіх 4 S-блоків зазвичай матимуть можливість 2/16. Тому, в результаті приблизних оцінок, очікуване число пар для додаткових шляхів складе:

$$8 \cdot 2^{11} \cdot 2^{-4} \cdot \left(\frac{2}{16}\right)^4 + 24 \cdot 2^{10} \cdot 2^{-4} \cdot \left(\frac{2}{16}\right)^4 = 2^{-2} + 3 \cdot 2^{-3} = 0{,}625.$$

Таким чином, підсумкова ймовірність диференціала $(16+0{,}625)/2^{16}$.

В результаті обчислювальних експериментів при усередненні по всім ключам очікуване значення ймовірності для диференціала склало $16{,}75 / 2^{16}$. Незначне відхилення від теоретичної оцінки пояснюється округленнями і припущеннями, які були присутні в представлених міркуваннях, а також можливістю потрапляння на переходи різниці з вірогідністю 4/16 для S-блоків другого рівня.

Загальний висновок для диференціалів даного виду полягає в тому, що збільшення кількості активних S-блоків призводить до зниження як кількості пар, які проходять по основному шляху, так і до зниження кількості пар, що проходять по додаткових шляхах, а в результаті і до зниження ймовірності диференціала.

Аналогічна ситуація для 32-бітного супер-S-блоку. Застосовуючи такі ж міркування, отримаємо очікуване число пар для основного шляху $2^{32} \cdot \left(\frac{4}{256}\right)^6 = 2^{-4}$; для додаткових шляхів:

$$2 \cdot 124 \cdot 2^{19} \cdot 2^{-4} \cdot \left(\frac{2}{256}\right)^4 + 126 \cdot 124 \cdot 2^{18} \cdot 2^{-4} \cdot \left(\frac{2}{256}\right)^4 \approx 2^{-5} + 1$$

Загальна очікувана кількість пар – 1, і це значно менше, ніж для диференціалів з 5 активними S-блоками.

Таким чином, в даному підрозділі продемонстровано застосування запропонованого методу для шифру AES для його міні-версії. Серед 2-циклових диференціалів AES максимальною вірогідністю мають ті, які містять мінімальну кількість активних S блоків (тобто 5) і у яких основний

слід (шлях) містить переходи різниці через «4» в таблиці різниць. Максимальна ймовірність диференціала для 16-бітного супер-S-блоку $72/2^{16}$, для 32-бітного супер-S-блоку $13/2^{32}$. Отримані результати для 32-бітного супер-S-блоку збігаються з відомими, представленими в роботі [120]. Правильність отриманих результатів для 16-бітного супер-S-блоку підтверджується збігом з результатами обчислювальних експериментів, в яких виконувався пошук диференціала, що володіє макимальною ймовірністю, за допомогою повного перебору.

4.3.3 Ймовірності 2-циклових диференціалів Rijndael-подібних шифрів з довільними підстановками

Метою даного підрозділу є демонстрація можливості застосування запропонованого методу для шифрів з довільними підстановками. Таким шифром, наприклад, є шифр «Калина» [156], який став переможцем на українському конкурсі блокових алгоритмів [157].

4.3.3.1 Використовувані супер-S-блоки

Як і в попередньому підрозділі, будемо розглядати супер-S-блок, який складається з послідовності операцій ByteSub, MixColumns, AddKey і ByteSub і працює з однією колонкою блоку даних. Супер-S-блок AES працює з 32-бітовим блоком, а в цій роботі ми будемо розглядати 16-бітний варіант супер-S-блоку, який містить 4 4-бітних S-блоки. Для такого варіанту супер-S-блоку є можливість більш детально вивчити властивості в ході обчислювальних експериментів.

Як 4-бітових S-блоків взята довільно побудована підстановка представлена в табл. 4.20.

Таблиця 4.20 – Таблиця використовуваної підстановки

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
5	9	B	2	3	6	F	8	7	C	0	E	1	D	4	A

Ця підстановка не є алгебраїчно побудованою і не володіє граничними диференціальними показниками, що видно з наведеної таблиці різниць (табл. 4.21).

Таблиця 4.21 – Таблиця різниць

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	2	0	2	0	2	0	2	4	0	4	0
2	0	0	2	0	0	2	0	4	0	0	0	2	2	0	4	0
3	0	0	2	0	0	0	0	2	0	6	0	4	2	0	0	0
4	0	2	0	0	6	0	4	0	0	0	2	0	0	0	0	2
5	0	0	0	4	0	0	0	0	0	0	8	0	0	4	0	0
6	0	4	0	4	2	0	2	0	2	0	2	0	0	0	0	0
7	0	2	0	0	0	0	2	0	2	0	0	0	0	8	0	2
8	0	0	6	0	0	2	0	0	0	0	0	6	2	0	0	0
9	0	0	2	0	0	4	0	2	0	2	0	0	2	0	4	0
A	0	0	0	0	0	4	0	4	0	0	0	0	4	0	4	0
B	0	0	4	0	0	2	0	2	0	6	0	2	0	0	0	0
C	0	0	0	0	6	0	2	0	2	0	2	0	0	0	0	4
D	0	6	0	0	0	0	2	0	6	0	0	0	0	0	0	2
E	0	2	0	4	2	0	0	0	4	0	2	0	0	0	0	2
F	0	0	0	4	0	0	4	0	0	0	0	0	0	4	0	4

Для подальших розрахунків важливим є кількість різних значень в цій таблиці для випадків, коли вхідний і вихідний різниці не рівні 0 (табл. 4.22).

Таблиця 4.22 – Статистична інформація для таблиці різниці 4-бітової підстановки

Значення	Кількість значень в таблиці різниць
“8”	2
“6”	8
“4”	22
“2”	36
“0”	157

Таким чином, за умови, що вхідний і вихідний різниці не рівні 0, частка ненульових значень в таблиці різниці становить 68/225 = 0,3.

У перетворенні MixColumns (MC) використовується множення на матрицю:

$$\begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix}.$$

4.3.3.2 Аналіз супер-S-блоку «силовим» способом

Використовуючи повний перебір можливих значень вхідних пар був визначений 2-цикловий диференціал, що володіє максимальною ймовірністю 3/128. Вхідна різниця цього диференціала 5550, вихідна – 00D1. Належить цьому диференціалу тільки одна диференціальна характеристика, яка представлена на рис. 4.9.

5	5	5	0	
B S				8 8 8
A	A	A	0	
M C				
0	0	7	D	
B S				8 6
0	0	D	1	

Рисунок 4.9 – Диференціальна характеристика, що володіє максимальною ймовірністю

На рис. 4.9 праворуч, навпроти перетворень BS (ByteSubstitution), представлені значення з таблиці різниці, на які потрапляють зазначені переходи різниці для окремих S-блоків, які містять на вході ненульову різницю.

Слід зауважити, що використовувати згаданий вище повний перебір можливих значень вхідних пари можливо лише для супер-S-блоку розміром до 32 бітів. Для супер-S-блоків більшого розміру, які, наприклад, використовуються в шифрі «Калина», в шифруючих перетвореннях функцій хешування Whirlpool, Groestl, значення максимальної ймовірності можна отримати тільки на підставі аналізу особливостей шифруючих перетворень. Продемонструємо, як це може бути зроблено для розглянутого шифру. Для цього будемо дотримуватися основних етапів запропонованого підходу:

1) визначення мінімальної кількості активних S-блоків;

2) визначення виду ДХ, що володіє максимальною ймовірністю;

3) визначення кількості і ймовірностей додаткових ДХ;

4) визначення максимальної ймовірності 2-циклового диференціала.

Відповідно до кількості гілок активізації (branch number) МДР-перетворення при ненульовий вхідний різниці завжди буде мінімум п'ять активних S-блоків. Можуть бути різні варіанти для кількості активних S-блоків на вході і виході: 1 і 4, 2 і 3, 3 і 2, 4 і 1. Розглянемо ці варіанти.

4.3.3.3 Кількість активних S-блоків: 1 на вході, 4 на виході

Вихідні значення для перетворення MC є вхідними для другого рівня S-блоків. Як видно з 4-х тетрад в вихідному значенні завжди містять різну різницю. Так як в таблиці різниці (табл. 4.21) є присутнім тільки два переходи з ймовірністю 8/16, то ймовірність того, що для всіх чотирьох активних S-блоків 2-го рівня може бути виконаний перехід різниці з максимальною вірогідністю (8/16) дорівнює 0:

$$P_{1\to 4}\ (\text{ 4 переходи різниці з ймовірністю } \frac{8}{16}) = 0.$$

Оцінимо ймовірність того, що для всіх п'яти активних S-блоків може бути виконаний перехід різниці з ймовірністю не менше, ніж 6/16. З 15 рядків і стовпців з ненульовими вхідною і вихідною різницями значення 6 і більше присутні в 8 рядках і 8 стовпцях. Тому, якщо вважати, що три різних значення різниці на другому рівні S-блоків з'являються випадково, то ймовірність того, що для всіх чотирьох активних S-блоків 2-го рівня може бути виконаний перехід різниці з ймовірністю 6/16 і більше дорівнює:

$$P_{1\to 4}(\text{ 4 переходи різниці з ймовірністю} \geq \frac{8}{16}) = \left(\frac{8}{15}\right)^3.$$

З огляду на наявність 8 варіантів різниці для активного S-блоку першого рівня, коли може бути виконаний перехід різниці з ймовірністю 6/16 і більше, очікувана кількість випадків, коли всі 5 переходів різниці виконуватися з ймовірністю 6/16 і більше, складе:

$$OK_{1\to 4}(\text{ 5 переходів різниці з ймовірністю} \geq \frac{6}{16}) = 8\cdot\left(\frac{8}{15}\right)^3 \approx 1.$$

Цей прогноз підтвердили обчислювальні експерименти. Знайдена диференціальна характеристика (ДХ), в якій всі 5 переходів різниці виконуються з ймовірністю 6/16 (рис. 4.10).

4	0	0	0	
B	S			6
4	0	0	0	
M	C			
8	4	4	C	
B	S			6666
2	4	4	4	

Рисунок 4.10 – ДХ, в якій всі 5 переходів різниці виконуються з ймовірністю 6/16

Оцінимо ймовірність існування додаткових ДХ, що належать такій диференціалу. Додаткові ДХ повинні містити однакові з основною ДХ вхідну і вихідну різниці, але інші проміжні значення різниць.

З огляду на, що частка ненульових значень в таблиці різниці 68/225 = 0,3, а в кожному рядку таблиці різниці зі значенням 6 і більше в середньому присутні 3 інших ненульових значення, то очікувана кількість додаткових ДХ складе:

$$OK_{1\to 4}(\text{додаткових ДХ}) = 3 \cdot 0,3^3 = 0,081.$$

отже, дуже малоймовірно наявність додаткових ДХ, що також підтвердили обчислювальні експерименти.

В значно меншій кількості додаткових ДХ полягає головна відмінність шифрувального перетворення з довільними підстановками від перетворення з алгебраїчно побудованими підстановками з граничними диференціальними і лінійними показниками.

4.3.3.4 Кількість активних S-блоків: 2 на вході, 3 на виході

У разі, коли на вході в МС 2 активних тетради, то тільки при певному співвідношенні різниці в цих тетрадах на виході МС може бути отримана нульова різниця в одній з 4-х тетрад. Наприклад, для отримання нульової різниці в 3-й тетраді на виході МС дві перші тетради на вході в МС повинні містити однакове значення різниці (рис. 4.11).

7	7	0	0	
B	S			88
D	D	0	0	
M	C			
D	4	0	9	
B	S			664
1	4	0	5	

5	5	0	0	
B	S			88
A	A	0	0	
M	C			
A	D	0	7	
B	S			468
5	1	0	D	

Рисунок 4.11 – Приклади ДХ

У будь-якому випадку, з 5 активних S-блоків в 3-х завжди буде різна різниця, а, отже всі 5 переходів різниці з ймовірністю 8/16 відбутися не можуть.

Повернемося до даного варіанту проходження різниці, коли дві перші тетради на вході в МС містять однакове значення різниці і нульова різниця в 3-й тетраді на виході МС. Імовірність того, що 3 переходи різниці на другому рівні S-блоків можуть бути виконані з ймовірністю 6/16 і більше:

$$P_{2\to 3}(3 \text{ переходи різниці з ймовірністю} \geq \frac{6}{16}) = \left(\frac{8}{15}\right)^3 = 0{,}15.$$

З огляду на наявність 8 варіантів різниці для двох активних S-блоків першого рівня, що містять однакову різницю, коли може бути виконаний перехід різниці з ймовірністю 6/16 і більше, очікувана кількість випадків, коли всі 5 переходів різниці виконуватися з ймовірністю 6/16 і більше, складе:

$$OK_{2\to 3}(\text{ 5 переходів різниці з ймовірністю} \geq \frac{6}{16}) = 8 \cdot \left(\frac{8}{15}\right)^3 \approx 1.$$

При цьому остання оцінка справедлива для кожного окремого випадку розташування тетради з нульовою різницею на виході перетворення МС. Так як є 4 варіанти такого розташування, то можна очікувати така ж кількість різних ДХ при фіксованій позиції вхідних активних S-блоків.

На рис. 4.12 а) представлена ДХ з ймовірністю всіх переходів 6/16 і більше і третьою нульовою тетрадою на виході МС, а на рис. 4.12 б) представлена ДХ з ймовірністю всіх переходів 6/16 і більше і другою нульовою тетрадою на виході МС.

4	4	0	0	
B	S			66
4	4	0	0	
M	C			
4	C	0	8	
B	S			666
4	4	0	2	

а)

3	7	0	0	
B	S			68
9	D	0	0	
M	C			
5	0	4	5	
B	S			868
A	0	4	A	

б)

Рисунок 4.12 – Приклади ДХ

Для кожної такої ДХ очікувана кількість додаткових ДХ розраховується також як і в попередньому розділі і складає:

$$OK_{2\to3}(\textit{додаткових ДХ}) = 3\cdot 0,3^3 = 0,081$$

4.3.3.5 Кількість активних S-блоків: 3 на вході, 2 на виході

У разі, коли на вході в МС 3 активних тетради, то тільки при певному співвідношенні різниці в цих тетрадах на виході МС може бути отримана нульова різниця в двох з 4-х тетрад. Наприклад, для отримання нульової різниці в 1-й і в 2-й тетрадах на виході МС три перші тетради на вході в МС повинні містити однакове значення різниці (рис. 4.13).

7	7	7	0	
B		S		888
D	D	D	0	
M		C		
0	0	9	4	
B		S		46
0	0	5	4	

Рисунок 4.13 – Приклад ДХ

При цьому для подальших оцінок важливо, що значення в активних тетрадах на виході МС будуть завжди різні. Оцінимо ймовірність того, що з цих двох різних значень різниці будуть можливі переходи з ймовірністю 8/16.

$$P_{3\to2}(2\ \textit{переходи різниці з ймовірністю}\ \frac{8}{16}) = \frac{2}{15}\cdot\frac{1}{15} \approx 0,009\,.$$

Тоді очікувана кількість випадків, коли всі 5 переходів різниці відбудуться з ймовірністю 8/16:

$$OK_{3\to 2}(5\ переходів\ різниці\ з\ ймовірністю\ \frac{8}{16}) = 2\cdot\frac{2}{15}\cdot\frac{1}{15}\approx 0,018.$$

Обчислювальні експерименти підтвердили відсутність таких випадків.

Імовірність того, що для двох активних тетрад на виході МС будуть існувати переходи з ймовірністю 6/16 і більше, буде значно вище:

$$P_{3\to 2}(2\ переходи\ різниці\ з\ ймовірністю \geq \frac{6}{16}) = \frac{8}{15}\cdot\frac{7}{15}\approx 0,25.$$

Очікувана кількість випадків, коли всі 5 переходів різниці відбудуться з ймовірністю 6/16 і більше складе:

$$OK_{3\to 2}(5\ переходів\ різниці\ з\ ймовірністю \geq \frac{6}{16}) = 8\cdot\frac{8}{15}\cdot\frac{7}{15}\approx 2.$$

Очікувана кількість додаткових ДХ:

$$OK_{3\to 2}(додаткових\ ДХ) = 3\cdot 0,3^2 = 0,27.$$

Ці випадки знайдені в ході обчислювальних експериментів і представлені на рис. 4.14.

5	5	5	0	
B	S			888
a	A	A	0	
M	C			
0	0	7	D	
B	S			86
0	0	d	1	

а)

4	4	4	0	
B	S			666
4	4	4	0	
M	C			
0	0	8	C	
B	S			66
0	0	2	4	

б)

Рисунок 4.14 – Знайдені ДХ

Варіант з 4 активними S-блоками на вході і 1 на виході ідентичний варіанту з переходом 1 в 4 і тому розглядати окремо не будемо.

Таким чином, в цьому підрозділі показано, як можна визначити максимальну ймовірність для 2-циклових диференціалів супер-S-блоку, в якому використовується довільна підстановка.

Одна з головних відмінностей шифрувального перетворення з довільними підстановками від перетворення з алгебраїчно побудованими підстановками з граничними диференціальними і лінійними показниками (які були розглянуті в попередньому підрозділі) полягає в тому, що для всіх розглянутих диференціалів було знайдено значно меншу кількість додаткових ДХ (всі розглянуті диференціали складалися з однієї ДХ).

Ще однією виявленої особливістю стала відсутність серед розглянутих диференціалів таких, ймовірність яких змінювалася б для різних ключів.

4.3.4 Імовірність 2-циклових диференціалів для шифру «Калина»

4.3.4.1 Супер-S-блоки «Калини»

Супер-S-блок складається з послідовності операцій ByteSub, MixColumns, AddKey і ByteSub і працює з однією колонкою блоку даних. Супер-S-блок Калини працює з 64-бітовим блоком і досліджувати такий супер-S-блок «силовим» способом неможливо.

У якості 8-бітових S-блоків використовуються 4 підстановки, які сформовані випадковим чином з контролем наступних параметрів: максимальне значення в таблиці різниці (для всіх підстановок дорівнює 8), максимальне значення в таблиці лінійних апроксимацій (для всіх підстановок одно 26), ступінь нелінійності (для всіх підстановок дорівнює 7). З точки зору ймовірностей диференціалів важливі таблиці різниць цих підстановок. Кількість максимальних значень, «8», в таблицях різниць для цих 4 підстановок становить 15, 9, 7 і 9. Очевидно, що підстановка з 15 вісімками в таблиці різниць дозволить побудувати двоциклову диференціал, який буде володіти максимальною вірогідністю. Тому в подальшому будемо розглядати

найгірший варіант, коли в шифрі використовується тільки одна ця підстановка, що володіє найнижчими показниками стійкості з усіх 4-х підстановок. Очікується, що цей варіант, у порівнянні з оригінальним, матиме більш високі показники ймовірностей диференціалів, а, відповідно, стійкість шифру трохи нижче.

Кількість осередків у таблиці різниці підстановки без урахування першого рядка і першого стовпця становить 255 * 255 = 65025. У табл. 4.23 представлена статистична інформація для таблиці різниці обраної підстановки.

Таблиця 4.23 – Статистична інформація для таблиці різниці 8 бітової підстановки

Значення	Кількість значень в таблиці різниць
“8”	15
“6”	246
“4”	3423
“2”	24996
“0”	36345

Дані з табл. 4.23 демонструють, що 56% значень таблиці різниці – це «0», а 44% - ненульові значення.

Позначимо *M* МДР-матрицю, яка використовується в перетворенні MixColumns (MC):

$$M = \begin{bmatrix} 1 & 1 & 5 & 1 & 8 & 6 & 7 & 4 \\ 4 & 1 & 1 & 5 & 1 & 8 & 6 & 7 \\ 7 & 4 & 1 & 1 & 5 & 1 & 8 & 6 \\ 6 & 7 & 4 & 1 & 1 & 5 & 1 & 8 \\ 8 & 6 & 7 & 4 & 1 & 1 & 5 & 1 \\ 1 & 8 & 6 & 7 & 4 & 1 & 1 & 5 \\ 5 & 1 & 8 & 6 & 7 & 4 & 1 & 1 \\ 1 & 5 & 1 & 8 & 6 & 7 & 4 & 1 \end{bmatrix}.$$

4.3.4.2 Пошук ДХ, що володіє максимальною вірогідністю

Як було показано в попередніх підрозділах, коли для підстановки обмежена кількість переходів різниці з максимальною вірогідністю, слід шукати шлях трансформації різниці з мінімальною загальною кількістю активних підстановок (для Калини – 9) і з максимальною кількістю повторюваних значень різниці на входах обох рівнів підстановок. В ході аналізу операції множення на МДР-матрицю був визначений такий шлях трансформації різниці для шифру «Калина». Вираз (4.10) показує яким буде результат множення на МДР-матрицю колонки, яка містить однакові ненульові значення різниці x в перших двох байтах і нульову різницю в інших байтах:

$$\begin{bmatrix} x \\ x \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times M = \begin{bmatrix} 0 \\ 5x \\ 3x \\ x \\ ex \\ 9x \\ 4x \\ 4x \end{bmatrix}.$$

Зазначений шлях трансформації різниці містить мінімальну загальну кількість активних підстановок – 2 + 7 = 9, при цьому по дві активні

підстановки першого і другого рівня можуть містити однакові значення різниці.

Існує 15 варіантів значення різниці x на виході першого рівня підстановок, в яке може відбутися перехід різниці з ймовірністю 8/256. Імовірність того, що для значення різниці $4x$ також буде існувати перехід з ймовірністю 8/256, становить 15/255. Тоді очікувана кількість випадків, коли два переходи різниці першого рівня і два з семи переходів різниці другого рівня підстановок матимуть можливість 8/256, складе:

$$15 \cdot \frac{15}{255} = \frac{225}{255} \approx 0{,}88 .$$

Очікувана кількість випадків, коли ще хоча б один перехід різниці другого рівня буде мати можливість 8/256, буде ще менше. Однак, в таблиці різниці підстановки багато переходів з ймовірністю 6/256 і тому для решти 5 значень різниці на вході другого рівня підстановок ($5x$, $3x$, x, ex, $9x$) з ймовірністю близькою до 1 існуватимуть переходи з такою ймовірністю. Тоді підсумкова ймовірність такої основної ДХ складе:

$$\left(\frac{8}{256}\right)^4 \cdot \left(\frac{6}{256}\right)^5 = \frac{243 \cdot 2^9}{2^{64}} .$$

4.3.4.3 Кількість і ймовірності додаткових ДХ

Тепер оцінимо кількість і ймовірності додаткових ДХ.

На виході підстановок 1-го рівня існує близько $\approx 254 \cdot 0{,}44 \approx 112$ додаткових варіантів вихідний різниці. Важливо, що значення на виході цих підстановок повинні бути однаковими, так як інакше не вийде нуль в першому байті різниці після множення на МДР-матрицю.

Відповідно до виразу 4.10, на вході 2-го рівня підстановок буде 6 різних ненульових значень різниці. Імовірність того, що для кожної з цих 6 окремо взятої активної підстановки буде існувати перехід в задане основною ДХ вихідне значення, становить 0,44. Тоді ймовірність того, що для всіх 6

випадків будуть можливі потрібні переходи різниці буде $(0{,}44)^6$; а очікувана кількість додаткових ДХ буде $112 \cdot (0{,}44)^6 \approx 0{,}8$, тобто, швидше за все одна додаткова ДХ. У цій додатковій ДХ більшість переходів матиме ймовірність 2/256. Навіть якщо половина з цих переходів матиме можливість 4/256, то в порівнянні з імовірністю основної ДХ ймовірність від додаткових ДХ буде незначною:

$$\left(\frac{4}{256}\right)^4 \cdot \left(\frac{2}{256}\right)^5 = \frac{2^5}{2^{64}}.$$

Максимальна ймовірність 2-циклового диференціала складе:

$$\frac{243 \cdot 2^9}{2^{64}} + \frac{2^5}{2^{64}} \approx 2^{-47,3}.$$

4.3.5 Порівняльний аналіз результатів, одержуваних за допомогою відомих і нового методів

Як було зазначено вище, граничні значення ймовірностей диференціалів можуть бути отримані на основі теореми 4.1 ([124]) для SPN-шифрів і теореми 2 ([125]) для шифрів, які використовують вкладені SPN-структури. Так, відповідно до цих теорем, верхня межа ймовірності 2-циклового диференціала шифру Rijndael-128 складе $\left(2^{-6}\right)^4 = 2^{-24}$, а верхня межа ймовірності 4-циклового диференціала – $\left(2^{-24}\right)^4 = 2^{-96}$. Запропонований в роботі [152] і запропонований в цьому розділі методи дозволяють більш точно оцінити верхню межу ймовірності 2 циклового диференціала – $\frac{13}{2^{32}} \approx 2^{-28,3}$. Тоді відповідна верхня межа ймовірності 4-циклового диференціала складе $\left(2^{-28,3}\right)^4 = 2^{-113,2}$.

Використовуючи запропонований метод, можуть бути істотно уточнені і верхня межа ймовірностей диференціалів для всіх варіантів шифру «Калина» (табл. 4.24).

Таблиця 4.24 – Верхня межа ймовірностей диференціалів для шифру «Калина»

	2-цикловий дифференц.	4-цикловий дифференц., Калина-128	4-цикловий дифференц., Калина-256	4-цикловий дифференц., Калина-512
Відом. метод	$\left(2^{-5}\right)^8 = 2^{-40}$	$\left(2^{-40}\right)^2 = 2^{-80}$	$\left(2^{-40}\right)^4 = 2^{-160}$	$\left(2^{-40}\right)^8 = 2^{-320}$
Проп. метод	$2^{-47,3}$	$\left(2^{-47,3}\right)^2 = 2^{-94,6}$	$\left(2^{-47,3}\right)^4 = 2^{-189,2}$	$\left(2^{-47,3}\right)^8 = 2^{-378,4}$

Представлені в табл. 4.24 верхня межа ймовірностей 2- і 4-циклових диференціалів для шифру «Калина», отримані за допомогою пропонованого в розділі методу, є найбільш точними з відомих.

4.3.6 Висновки по розділу

1. Запропоновано вдосконалений в напрямку розширення області використання метод оцінювання ймовірностей 2-циклових диференціалів для SPN-шифрів. Відмінність від відомого методу полягає в тому, що скорочений набір вимог до підстановлювальний перетворенням, що дозволяє отримати значення 2-циклових диференціалів для шифрів і з не граничними (з довільними) диференціальними показниками підстановок.

2. Продемонстровано застосування запропонованого методу для шифру AES і для його міні-версії. Серед 2-циклових диференціалів AES максимальну вірогідність мають ті, які містять мінімальну кількість активних S-блоків (тобто 5) і у яких основний слід (шлях) містить переходи різниці

через «4» в таблиці різниць. Максимальна ймовірність диференціала для 16-бітного супер-S-блоку $72/2^{16}$, для 32-бітного супер-S-блоку $13/2^{32}$. Отримані для 32-бітного супер-S-блоку збігаються з відомими результатами, представленими в роботі [152]. Правильність отриманих результатів для 16-бітного супер-S-блоку підтверджується збігом з результатами обчислювальних експериментів, в яких виконувався пошук диференціала, що володіє максимальною вірогідністю за допомогою повного перебору.

3. З метою перевірки правильності роботи запропонованого методу для випадків, коли використовуються неалгебраїчні побудовані підстановки, була визначена максимальна ймовірність для 2-циклових диференціалів для 16-бітного супер-S-блоку, в якому використовується довільна підстановка. Правильність отриманих результатів для 16-бітного супер-S-блоку підтверджується збігом з результатами обчислювальних експериментів, в яких виконувався пошук диференціала володіє максимальною вірогідністю за допомогою повного перебору.

4. Практична значимість запропонованого методу полягає в тому, що з його допомогою вдалося значно уточнити верхню межу ймовірності 2-циклових диференціалів для шифру «Калина». Ця межа становила $\approx 2^{-47,3}$, замість 2^{-40} при використанні відомого методу. Уточнене значення верхньої межі ймовірності 2-циклових диференціалів дозволяє уточнити і граничне значення ймовірності 4 циклових диференціалів (табл. 4.24). Для Калини-128 значення уточнено у $2^{14,6}$ раз, для Калини-256 – у $2^{29,2}$ раз, Калини-512 – у $2^{58,4}$ раз.

4.4 Метод оцінювання стійкості шифрів до атаки усічених байтових диференціалів, заснований на теоретичному обгрунтуванні верхніх меж ймовірностей байтових диференціалів

4.4.1 Основні ідеї пропонованого підходу до визначення ймовірностей байтових усічених диференціалів

У підрозділі 4.2 були розглянуті існуючі методи оцінювання стійкості БСШ до атаки усічених байтових диференціалів. Відзначено, що складність всіх цих методів критично зростає зі збільшенням розміру блоку і зростанням числа циклів. В результаті, відомі методи оцінювання стійкості застосовні для шифрів з розміром блоку до 256 бітів (до 32 байт). На додаток до цього, ті методи оцінювання стійкості, які засновані на розгляді байтових диференціальних характеристик (БДХ), не можуть повною мірою .

Відома теорема 4.2 дозволяє обгрунтувати відсутність ефективних байтових диференціальних характеристик (БДХ) для 3 і більше циклів 128-бітного варіанту шифру Rijndael. Однак для обґрунтування стійкості необхідно також доказ відсутності ефективних байтових диференціалів (БД), що і є метою цього розділу.

В основі запропонованого в цьому розділі підходу лежить виконання наступних основних етапів.

1. Визначення кількості циклів, коли для шифру відсутні ефективні БДХ.

2. Визначення ймовірностей основної та додаткових БДХ.

3. Оцінка кількості додаткових БДХ.

4. Оцінка кількості додаткових нетипових БДХ.

5. Оцінка верхньої межі ймовірностей байтових диференціалів (БД).

При виконанні першого етапу запропонованого для шифрів з розміром блоку до 256 бітів можуть бути використані відомі методи, запропоновані в [123,139]. Для деяких шифрів відсутність ефективних БДХ може бути

доведено теоретично, як це зроблено в роботі [123] для Rijndael-подібних шифрів.

Аналіз обчислювальних експериментів з пошуку БДХ і БД для різних шифрів дозволив запропонувати класифікацію БДХ, що входять до складу різних БД. Ця класифікація представлена в наступному підрозділі і вона дозволяє виконати найбільш проблематичні етапи викладеного вище підходу (етапи 2-5) і, в підсумку, прийти до оцінки верхньої межі ймовірностей БД.

4.4.2 Пропонований метод оцінювання шифру Rijndael 128

4.4.2.1 Види БДХ, їх ймовірності і кількість. Основна БДХ

Для шифру Rijndael з 128-бітовим блоком справедливо наступне твердження.

Твердження 4.2. Для кожного не нездійсненного r-циклового ($r \geq 3$) БД завжди є одна і тільки одна БДХ з ймовірністю приблизно $p_{сл} \cdot 2^{-0,0904r}$.

Доказ. Відомо кілька нездійсненних БД, які лежать в основі атаки нездійсненних диференціалів. Для всіх інших БД існує БХ, в якій на виході всіх перетворень MixColumns крім останнього і передостаннього циклів буде формуватися вектор активізації з усіма активними байтами. У більшості випадків (табл. 2.4) ймовірність кожного такого переходу для однієї колонки складе $2^{-0,0226}$. Якщо активні всі 4 колонки, а така ситуація в даному випадку виникне максимум після одного циклу, то ймовірність буде зменшуватися в $2^{-0,0226*4} = 2^{-0,0904}$ раз на кожному циклі. У підсумку, за кожен пасивний байт у вихідний різниці в останньому і, в деяких випадках, передостанньому циклах «заплатимо» зменшенням імовірності в 2^8 разів. Крім того, кожен додатковий цикл буде зменшувати ймовірність приблизно в $2^{-0,0904}$ раз. Таким чином, підсумкова ймовірність такої БХ складе $p_{сл} * 2^{-0,0904*r}$.

Розглянутий вид БДХ містить переходи векторів активізації з ймовірністю близькою до 1 у всіх циклах крім останнього і, іноді,

передостаннього циклів. Надалі подібну БХ будемо називати основною, а решта БХ, що належать цьому ж БД, будемо називати додатковими.

На рис. 4.15 представлений приклад основний БДХ із зазначенням ймовірностей переходів активізації для окремих циклів.

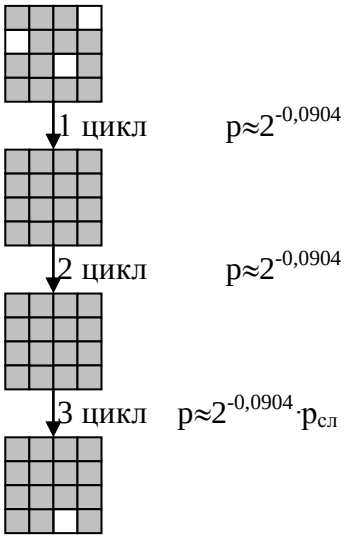


Рисунок 4.15 – Приклад основної БДХ

4.4.2.2 Види БДХ, їх ймовірності і кількість. Додаткові БДХ

Крім основної БДХ до складу БД можуть входити інші БДХ, які будемо називати додатковими. Для більшості додаткових БДХ справедливим є твердження 4.3.

Твердження 4.3. Для кожного БД з 3 або більше циклами будь-яка додаткова БДХ з m додатковими пасивними байтами має ймовірність приблизно в 2^{8m} раз нижче, ніж основна БХ цього БД.

Доказ. Кожна додаткова БХ повинна відрізнятися від основної і ця відмінність має полягати в одному або декількох додаткових пасивних байтах. При чому, додаткові пасивні байти повинні бути в тих циклах, де в основний БХ використовуються переходи з вірогідністю близько 1. Таким чином, відповідно до табл. 1 кожен додатковий пасивний байт буде

зменшувати ймовірність БХ приблизно в 28 разів. У підсумку, ймовірність БХ буде завжди приблизно в 28m раз нижче, ніж основна БХ цього БД.

На рис. 4.16 представлений приклад додаткової 3-циклової БДХ з одним додатковим пасивним байтом після другого циклу і з тими ж значеннями вхідний і вихідний різниць, що і в БДХ на рис. 4.15.

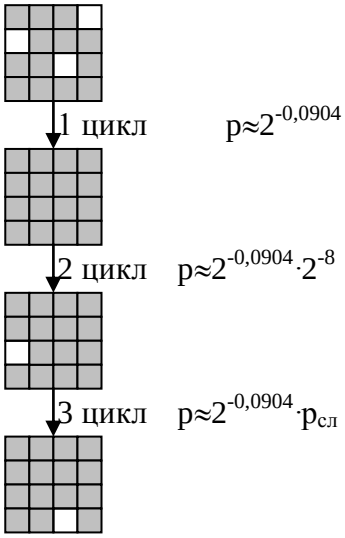


Рисунок 4.16 – Приклад додаткової БДХ

Кількість додаткових БДХ залежить від числа додаткових пасивних байтів і від числа циклів. При цьому максимальна кількість додаткових БДХ випадає на ті випадки, коли додаткові пасивні байти розподілені по одному в різних циклах. Розглянемо приклад БДХ з одним додатковим пасивним байтом. Кількість додаткових БДХ буде $C_R^1 \cdot C_{16}^1 = 16 \cdot R$, де перший множник C_R^1 – це кількість варіантів вибору циклу з цим пасивним байтом, а C_{16}^1 – кількість варіантів розташування цього пасивного байта в блоці шифру. Таким чином, відповідно до твердженням 4.3, сумарна ймовірність БХ з одним додатковим пасивним байтом складе $16R \cdot 2^{-8}$.

Є три варіанти розташування двох додаткових пасивних байтів:

- одна колонка – кількість варіантів $C_R^1 \cdot C_4^1 \cdot C_4^2 = R \cdot 4 \cdot 6 = 24R$;

- один цикл, але різні колонки – кількість варіантів $C_R^1 \cdot C_4^2 \cdot (C_4^1)^2 = 96R$;

- два різних цикли – кількість варіантів $C_R^2 \cdot \left(C_{16}^1\right)^2 = \frac{R!}{(R-2)!2!} \cdot 256 = 128 \cdot R^2 - 128 \cdot R$.

Підсумувавши ці значення, можемо отримати сумарну ймовірність від БХ з двома додатковими пасивними байтами: $(128R^2 - 8R) \cdot 2^{-16}$.

Діючи аналогічним чином, ми оцінили кількість варіантів для додаткових БДХ з трьома додатковими пасивними байтами. У цьому випадку кількість варіантів розташування цих байтів значно більше і складає, в результаті, $683R^3 - 128R^2 + 5R$. Як і у випадку з двома байтами, в підсумковому значенні можна виділити найбільш вагомий доданок $683R^3$. Можна знехтувати іншими складовими і вважати, що сумарна ймовірність від БДХ з трьома додатковими пасивними байтами: $\approx 683R^3 \cdot 2^{-24}$.

Вважаючи, що сумарна ймовірність від БДХ з i додатковими пасивними байтами буде представлена тільки одним найвагомішим елементом, і цей елемент присутній, коли $i \le R$, можна записати загальну формулу для сумарної ймовірності від БДХ з i додатковими пасивними байтами:

$$2^{-8i} \cdot 2^{-0,0904 \cdot R} \cdot p_{сл} \cdot C_R^i \cdot 16^i \approx$$

$$\approx \frac{R^i \cdot 16^i \cdot 2^{-8i}}{i!} \cdot 2^{-0,0904 \cdot R} \cdot p_{сл} = \frac{\left(\frac{R}{16}\right)^i}{i!} \cdot 2^{-0,0904 \cdot R} \cdot p_{сл}$$

для $i = 0,\ldots,R$ (випадок $i = 0$ відповідає основній БХ).

Тоді сумарна ймовірність основної та додаткових r-циклових БДХ, що потрапляють під твердження 4.2 та 4.3, може бути визначена наступним виразом

$$\sum_i \frac{R^i \cdot 16^i \cdot 2^{-8i}}{i!} \cdot 2^{-0,0904R} \cdot p_{сл} \qquad (4.10)$$

Після елементарних перетворень вираз (4.10) можна представити таким чином:

$$\sum_{i=0}^{R} \frac{\left(\frac{R}{16}\right)^i}{i!} \cdot 2^{-0,0904R} \cdot p_{сл}.$$

З теорії рядів відомо, що $\sum_{i=0}^{\infty} \frac{x^i}{i!} = e^x$. Тоді, в нашому випадку:

$$\sum_{i=0}^{R} \frac{\left(\frac{R}{16}\right)^i}{i!} \cdot 2^{-0,0904R} \cdot p_{сл} < \sum_{i=0}^{\infty} \frac{\left(\frac{R}{16}\right)^i}{i!} \cdot 2^{-0,0904R} \cdot p_{сл} \approx e^{\frac{R}{16}} \cdot 2^{-0,0904R} \cdot p_{сл} =$$

$$= \left(e^{\frac{1}{16}} \cdot 2^{-0,0904} \right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл} = p_{сл} \qquad (4.11)$$

Таким чином, сума ймовірностей основної та додаткових БДХ, що потрапляють під умови затвердження 4.3, нічого очікувати перевищувати $p_{сл}$.

Однак, в результаті обчислювальних експериментів було виявлено, що не всі додаткові БДХ потрапляють під умови затвердження 4.3.

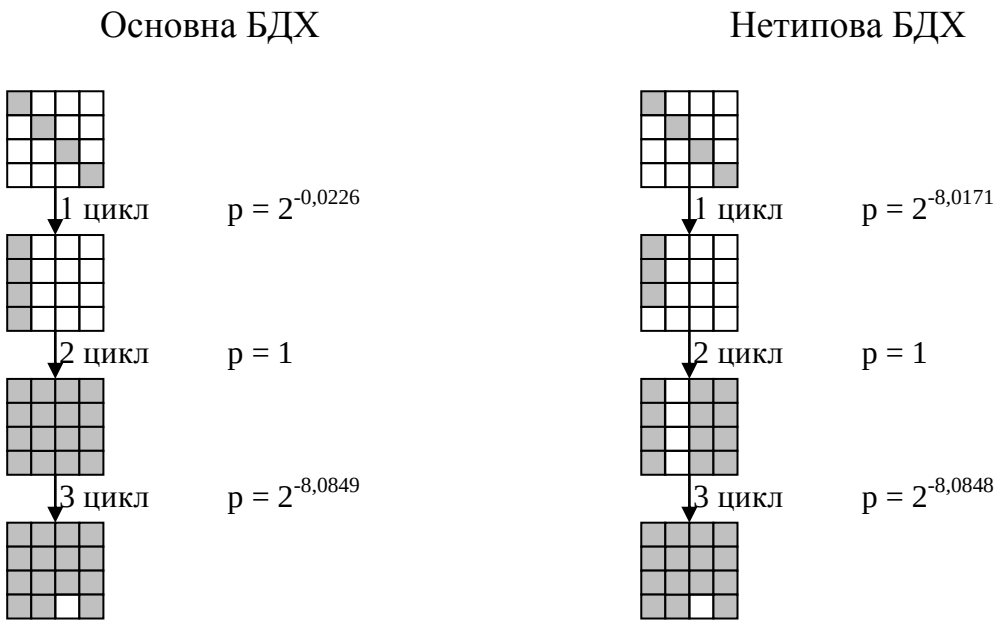
4.4.2.3 Види БДХ, їх ймовірності і кількість. Нетипові додаткові БДХ

Під умову зниження ймовірності додаткових БДХ не потрапляють r-циклові БДХ, перші $r-1$ циклів яких покривають ефективні $(r-1)$-циклові БДХ (які відповідають умові ефективності – вираз (4.7)). Імовірність таких

БДХ вище, ніж вірогідність основний БДХ помножена на 2^{-8k}, але, обов'язково, нижче ймовірності основної БДХ. Будемо називати їх нетиповими додатковими БДХ. Приклади основної і нетипової БДХ наведені на рис. 4.17.

Перші два цикли нетиповою БДХ покриває 2-циклова БДХ, яка є ефективною, тому що її ймовірність $2^{-8,0171}$ з урахуванням чотирьох пасивних байтів у вихідний різниці задовольняє умові ефективності (4.7).

З рис. 4.17 видно, що нетипова БДХ не підпадає під умови затвердження 4.3, так як містить на 5 пасивних байтів більше, ніж основна БДХ, але її ймовірність всього приблизно в 2^8 разів менше основний БДХ.



$$P^{(3)}_{ОснБДХ} = 2^{-8,1075} \qquad P^{(3)}_{НетипБДХ} = 2^{-16,1019} = P^{(2)}_{ЭффБДХ} \cdot 2^{-0,0904} \cdot p_{сл}$$

Рисунок 4.17 – Приклад нетипової додаткової БДХ

Імовірність нетипової БДХ, як зазначено на рис. 4.17, дорівнює $P^{(3)}_{НетипБДХ} = P^{(2)}_{ЭффБДХ} \cdot 2^{-0,0904} \cdot p_{сл}$. Отже, для визначення додаткової ймовірності від нетипових 3-циклових БДХ $P_{НетипБДХ}$ для обраного БД з фіксованим вхідним вектором активізації необхідно визначити кількість і ймовірності ефективних $(r-1)$-циклових БДХ. Тоді

$$P_{НетипБДХ} \approx (A_1 \cdot 2^{-8} + A_2 \cdot 2^{-16} + \ldots) \cdot p_{сл} \cdot 2^{-0,0904}, \qquad (4.12)$$

де A_i – кількість ефективних 2-циклових БДХ, що володіють ймовірністю 2^{-8i} для обраного вхідного вектора активізації.

В результаті обчислювальних експериментів з пошуку ефективних БДХ для шифру Rijndael-128 за допомогою методів з [166,124] було виявлено вхідний вектор активізації, для якого додаткова ймовірність від нетипових 3-циклових БДХ є максимальною. Цей вектор активізації є вхідним для БДХ, яка представлена на рис. 4.17.

Кількість і ймовірності знайдених ефективних БДХ для 2-циклового шифру Rijndael з розміром блоку 128 бітів і зазначеним вхідним вектором активізації представлені в таблиці 4.25.

Таблиця 4.25 – Нетипові 3-циклові БДХ для Rijndael з розміром блоку 128 бітів і зазначеним вхідним вектором активізації

Імовірність 2-циклової ефективної БДХ, $P^{(2)}_{ЕфБДХ}$	Кількість ефективних 2-циклових БДХ, A_i	Додаткова ймовірність від нетипових 3-циклових БДХ, $A_i \cdot P^{(2)}_{ЕфБДХ} \cdot 2^{-0,0904} \cdot p_{сл}$
2^{-8}	4	$2^{-6,0904} \cdot p_{сл}$
2^{-16}	6	$2^{-13,4904} \cdot p_{сл}$
2^{-24}	4	$2^{-22,0904} \cdot p_{сл}$

З табл. 4.25 видно, що зі зменшенням імовірності кількість БДХ може збільшуватися, але загальна додаткова ймовірність скорочується. Отже, найбільша додаткова ймовірність від нетипових БДХ, в основі яких лежать ефективні $(r-1)$-циклові БДХ з максимальною ймовірністю. Максимальна ймовірність для таких БДХ становитиме 2^{-8}. Таким чином, для будь-якого вхідного вектора активізації:

$$P_{НетипБДХ} \approx A_1 \cdot 2^{-8} \cdot p_{сл} \cdot 2^{-0,0904},$$

де A_1 – кількість ефективних 2-циклових БДХ, що володіють ймовірністю 2^{-8} для обраного вхідного вектора активізації.

У підсумку, для Rijndael-128 максимальна додаткова ймовірність від додаткових нетипових БДХ складе:

$$P_{НетипБДХ} = 2^{-6,0904} \cdot p_{сл}. \qquad (4.13)$$

4.4.2.4 Верхня межа ймовірностей БД для шифру Rijndael-128

Результати, представлені в цьому підрозділі для шифру Rijndael з розміром блоку 128 бітів, можуть бути підсумовані у вигляді такої теореми.

Теорема 4.3. Для шифру Rijndael з розміром блоку 128 бітів немає ефективних БД для 3 або більше циклів.

Доказ. Вираз (4.11) показує, що сума ймовірностей основної та додаткових БДХ, що потрапляють під твердження 4.3, обмежена зверху значенням $p_{сл}$. Вираз (4.13) показує верхню межу для додаткової ймовірності від додаткових нетипових БДХ. Отже, верхня межа ймовірності 3-циклового БД буде визначатися як сума виразів (4.11) і (4.13):

$$P_{БД}^{(3)} = \sum_{i=0}^{R} \frac{\left(\frac{R}{16}\right)^i}{i!} \cdot 2^{-0,0904R} \cdot p_{сл} + P_{НетипБДХ}$$

або

$$P_{БД}^{(3)} = p_{сл} + 2^{-6,0904} \cdot p_{сл}.$$

Значення $2^{-6,0904} \cdot p_{сл}$ не можна вважати вирішальною перевагою для того, щоб БД можна було вважати ефективним, тобто таким, що задовольняє вислову (4.7).

Подальше збільшення кількості циклів буде знижувати додаткову ймовірність, яка обчислюється відповідно до вираження (4.13), для кожного $(r+1)$-циклового БД в порівнянні з r-цикловим БД у $2^{-0,0904}$ раз. Походження цього значення було розглянуто раніше в цій роботі. Теорема доведена.

4.4.3 Пропонований метод оцінювання стосовно інших Rijndael-подібних SPN-шифрів

Важлива особливість запропонованого методу полягає в тому, що він може бути використаний для шифру з будь-яким розміром блоку. У цьому підрозділі проаналізуємо стійкість інших Rijndael-подібних SPN-шифрів, які, в деяких випадках, мають значно більшим розміром блоку.

Будемо розглядати шифри з розміром колонок 4 або 8 байтів і різною кількістю колонок в блоці (табл. 4.26).

Таблиця 4.26 – Мінімальна кількість циклів, при якій не існує ефективних БДХ

Розмір колонки	К-ть колонок в блоці	Кількість циклів
4 байти	4	3
	6	4
	8	6
8 байтів	2	3
	4	3
	8	3
	16	7

На першому етапі слід визначити мінімальну кількість циклів, при якому для розглянутих шифрів не існує ефективних БДХ. Для цього були проведені обчислювальні експерименти з пошуку ефективних БДХ з використанням методу Мораї [124] для шифрів з розміром блоку до 128 бітів і з використанням модифікованого методу з [123] для великих розмірів блоку. При цьому, виділені кольором в табл. 4.26 варіанти шифрів потрапляють під умови представленої у 2 розділі теореми 4.2, отже, для них теоретично обґрунтовано відсутність ефективних БДХ для 3 і більше циклів.

Представлені результати підтверджують справедливість теоретичних висновків, а також демонструють брак варіантів шифрів для яких число колонок більше числа рядків ($n > m$). Ефективні БДХ в цих випадках можуть покривати значно більшу кількість циклів.

Ще одне спостереження, пов'язане з аналізом результатів обчислювальних експериментів, полягає в тому, що для всіх розглянутих варіантів шифрів БДХ, які володіють максимальною вірогідністю, зазвичай містять мінімальну кількість активних колонок, але при цьому, відповідно до поданої у 2 розділі лемою 2.1, немає жодного циклу, де одночасно всі колонки активні на вході перетворення MC. Так, для випадків, коли в рамках операції SR виконується циклічний зсув кожної з рядків на різну кількість байт, ефективні БДХ, що покривають максимальну кількість циклів, в основному складаються з циклів з кількістю активних колонок. Наприклад, для всіх варіантів Rijndael це значення дорівнює 3, тобто, в основному, ефективні БДХ, представлені в табл. 3 для 4-байтових колонок, будуть містити по 3 активних колонки в кожному з циклів. При цьому, кожна активна колонка зазвичай містить $\frac{m}{2}+1$ активних байтів на вході і виході. Приклад одного такого циклу для шифру з 128-бітовим блоком представлений на рис. 4.18.

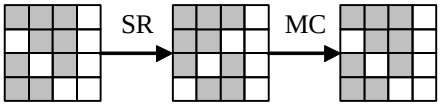


Рисунок 4.18 – Приклад циклу з однією пасивною колонкою на вході і виході

На другому етапі запропонованого підходу слід оцінити ймовірності основної та додаткових БДХ. Для виконання цих оцінок доведені твердження 4.2 та 4.3. Твердження 4.3 справедливо для всіх Rijndael-подібних SPN-шифрів. У твердженні 4.2 чисельний множник в показнику ступеня граничного значення залежить від розміру колонки і відповідних ймовірностей переходів з таблиць 4.6 або 4.7, і від кількості колонок в блоці. Таким чином, ймовірність основної БДХ:

$$P_{осн} = p_{сл} \cdot (p_{MC})^{nr}$$

де r – кількість циклів;

n – кількість колонок в блоці;

p_{MC} – ймовірність переходу вектора активізації для однієї колонки з усіма активними байтами на вході і виході (значення ймовірності в нижньому правому куті табл. 4.6 або 4.7).

Відповідні граничні значення ймовірностей основної та додаткових БДХ представлені в табл. 4.27.

Третій етап пов'язаний з оцінкою кількості можливих додаткових БДХ. У підрозділі 4.2 показано, що найбільша кількість додаткових БДХ набирається у випадках, коли додаткові пасивні байти будуть розташовуватися в окремих циклах. Рештою додаткових БДХ можна знехтувати, так як їх кількість буде значно менше.

Позначивши кількість байтів в блоці b, кількість додаткових БДХ з розподілом додаткових k пасивних байтів по одному в циклі може бути оцінений як

$$C_r^k \cdot b^k = \frac{r!}{k!\cdot(r-k)!} \cdot b^k \approx \frac{(r^k \cdot b^k)}{k!} \qquad (4.14)$$

Таблиця 4.27 – Ймовірності основної та додаткових БДХ

Розмір колонки	К-ть колонок в блоці	Імовірність основної БДХ	Імовірність додаткової БДХ з k додатковими пасивними байтами
4 байти	4	$p_{сл} \cdot 2^{-0,0904r}$	$p_{сл} \cdot 2^{-0,0904r} \cdot 2^{-8k}$
	6	$p_{сл} \cdot 2^{-0,1356r}$	$p_{сл} \cdot 2^{-0,1356r} \cdot 2^{-8k}$
	8	$p_{сл} \cdot 2^{-0,1808r}$	$p_{сл} \cdot 2^{-0,1808r} \cdot 2^{-8k}$
8 байтів	2	$p_{сл} \cdot 2^{-0,08r}$	$p_{сл} \cdot 2^{-0,08r} \cdot 2^{-8k}$
	4	$p_{сл} \cdot 2^{-0,16r}$	$p_{сл} \cdot 2^{-0,16r} \cdot 2^{-8k}$
	8	$p_{сл} \cdot 2^{-0,32r}$	$p_{сл} \cdot 2^{-0,32r} \cdot 2^{-8k}$
	16	$p_{сл} \cdot 2^{-0,64r}$	$p_{сл} \cdot 2^{-0,64r} \cdot 2^{-8k}$

У табл. 4.28 представлено кількість різних r-циклових БДХ з k додатковими пасивними байтами, які належать одному і тому ж БД, тобто мають фіксовані вхідну і вихідну активізації байтів.

Таблиця 4.28 – Кількість різних r-циклових БДХ з k додатковими пасивними байтами

Розмір колонки	К-ть колонок в блоці	Кількість БДХ
4 байти	4	$(r^k \cdot 16^k)/k!$
	6	$(r^k \cdot 24^k)/k!$
	8	$(r^k \cdot 32^k)/k!$
8 байтів	2	$(r^k \cdot 16^k)/k!$
	4	$(r^k \cdot 32^k)/k!$
	8	$(r^k \cdot 64^k)/k!$
	16	$(r^k \cdot 128^k)/k!$

На четвертому етапі слід оцінити сумарну ймовірність від всіх БДХ, що входять до складу одного БД. Для цього слід підсумувати твори кількості БДХ (з табл. 4.28) на їх ймовірність (з табл. 4.27). Результати представлені в табл. 4.29.

Таблиця 4.29 – Ймовірність БД з числом циклів, при якому відсутні ефективні БДХ

Розмір колонки	К-ть колонок в блоці	Імовірність БД
4 байти	4	$\left(e^{\frac{1}{16}} \cdot 2^{-0,0904}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$
	6	$\left(e^{\frac{3}{32}} \cdot 2^{-0,1356}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$
	8	$\left(e^{\frac{1}{8}} \cdot 2^{-0,1808}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$
8 байтів	2	$\left(e^{\frac{1}{16}} \cdot 2^{-0,09}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$
	4	$\left(e^{\frac{1}{8}} \cdot 2^{-0,18}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$
	8	$\left(e^{\frac{1}{4}} \cdot 2^{-0,36}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$
	16	$\left(e^{\frac{1}{2}} \cdot 2^{-0,72}\right)^R \cdot p_{сл} \approx 1^R \cdot p_{сл}$

З представлених в таблиці 4.29 результатів видно, що у всіх випадках отримано значення ймовірності БД рівне $p_{сл}$, а значить БД з більшим числом циклів, ніж може покрити ефективна БДХ, не є ефективними (не задовольняють умові (4.7)).

4.4.4 Пропонований метод оцінювання стосовно шифрів, що використовують ланцюг Фейстеля і Rijndael-подібні перетворення

У цьому підрозділі запропонований підхід застосовується до шифрів, в основі яких лежить ланцюг Фейстеля, а циклове перетворення використовує Rijndael-подібні перетворення. До відомих шифрів цього виду можна віднести «Торнадо» [167] і «Лабіринт» [168].

У табл. 4.30 представлені результати виконання першого етапу дослідження таких шифрів.

Таблиця 4.30 – Мінімальна кількість циклів, при якій не існує ефективних БДХ

Розмір колонки	К-ть колонок в напівблоці	Кількість циклів
4 байти	2	5
	4	8
8 байтів	1	3
	2	6
	4	9
	8	16

Аналіз представлених в табл. 4.30 результатів дозволяє зробити висновок про значно меншу захищеність Rijndael-подібних шифрів із застосуванням ланцюгів Фейстеля в порівнянні з використанням схем SPN (табл. 4.26). Виключенням є варіант ланцюга фейстелі з єдиною колонкою в блоці (МДР-перетворення охоплює весь напівблок).

На другому етапі слід оцінити вірогідність основної та додаткових БДХ.

Як и у випадку з SPN-схемою, основна БДХ буде мати переходи, що мають максимальну вірогідність і максимальну кількість активних байтів при виході всіх операцій МС та всіх XOR-комбінацій півблоків, у всіх циклах, крім декількох останніх. В декількох останніх циклах кількість і позиція активних байтів на виході перетворень буде диктуватися значенням вихідного вектора активізації.

Ймовірність такої *r*-циклової БДХ може бути оцінена таким чином:

$$P_{осн} = \left((p_{MC})^n \cdot \left(\frac{254}{255} \right)^{mn} \right)^r \cdot p_{сл},$$

де m та n – відповідно, число рядків і колонок у напівблоці;

p_{MC} – ймовірність переходу вектора активізації для однієї колонки з усіма активними байтами на вході та виході (значення вірогідності в нижньому правому куті табл. 4.6 або 4.7). В останньому вираженні перший множник у скобках враховує зниження вірогідності БДХ при проходженні перетворень *MC*, а другий – при проходженні XOR-суми напівблоків.

Тоді ймовірність кожної додаткової БДХ з додатковими пасивними байтами буде:

$$P_{дод} = P_{осн} \cdot 2^{-8k}.$$

В табл. 4.31 наведені ймовірності основної та додаткової БДХ для шифрів з використанням ланцюга Фейстеля та кількості циклів, у яких відсутні ефективні БДХ (аналог табл. 4.27 для SPN-шифрів).

Таблиця 4.31 - Вірогідності основних БДХ

Розмір колонки	К-ть колонок у напівблоці	Імовірність основної БДХ
4 байти	2	$p_{сл} \cdot 2^{-0,0902r}$
	4	$p_{сл} \cdot 2^{-0,1811r}$
8 байтів	1	$p_{сл} \cdot 2^{-0,09r}$
	2	$p_{сл} \cdot 2^{-0,1807r}$
	4	$p_{сл} \cdot 2^{-0,3614r}$
	8	$p_{сл} \cdot 2^{-0,7228r}$

На третьому етапі необхідно оцінити кількість можливих додаткових БДХ. Як і у випадку SPN-шифрів, найбільша кількість додаткових БДХ набирається в випадках, коли додаткові пасивні байти будуть розташовуватися в окремих циклах. Іншими варіантами розташування

додаткових пасивних байтів можна знехтувати, так як їх кількість буде значно меншою.

Позначивши кількість байтів в блоці b, кількість додаткових БДХ з розподілом додаткових k пасивних байтів по одному в циклі може бути оцінена як

$$C_r^k \cdot \left(\frac{b}{2}+\frac{b}{2}\right)^k = C_r^k \cdot b^k \approx \frac{(r^k \cdot b^k)}{k!}$$

Отримане значення ідентичне тому, що було отримано в виразі (4.5) для SPN-шифрів, а дані з табл. 4.31 практично повторюють дані з табл. 4.3 для однакових розмірів блоку. Таким чином, в підсумку, як і для схем SPN (табл. 4.29), у всіх випадках шифрів, побудованих з використанням ланцюга Фейстеля, гранична вірогідність БД складатиме $\approx p_{сл}$. Тобто, БД не будуть ефективними.

4.4.5 Достаткові умови відсутності ефективних БД для БСШ. Модель захищенного від атак в'язних байтових диференціалів БСШ

У цьому підрозділі на основі аналізу підходу до підтвердження відсутності ефективних БД, представлених у попередніх підрозділах для різних шифрів, виділімо і загальні основні умови, при виконанні яких можна стверджувати про відсутність ефективних БД для будь-якого блочного шифра. Далі ці умови будуть положені в основу моделі захищеного від атак в'язних байтових диференціалів блочного шифра.

Перша умова є досить очевидною.

Умова 4.1. Необхідною умовою відсутності ефективних БД для r-циклового шифру є відсутність ефективних БДХ для цього шифру.

Очевидність цієї умови випливає з визначень БД і БДХ. Для пошуку ефективних БДХ можуть бути використані відомі методи з [123,140].

Для виділення інших умов слід проаналізувати підсумкову формулу для ймовірності R-циклового БД, отриману на основі представлених в підрозділах 4.2 та 4.3 результатів для Rijndael-подібних шифрів.

$$P_{БД} = \sum_{k=0}^{\infty}(C_R^k \cdot b^k \cdot 2^{-8k} \cdot 2^{-0,0904 \cdot R} \cdot p_{сл}) + P_{НетипБДХ}, \qquad (4.15)$$

де b – розмір блоку в байтах;

k – кількість пасивних байтів в додаткових БДХ;

R – кількість циклів;

$p_{сл} \approx (2^{-8})^u$;

u – число неактивних байтів у вихідний різниці або число нульових бітів у вихідному векторі активізації.

Тепер розглянемо суть основних елементів формули (4.15).

1) $C_R^k \cdot b^k$ – кількість варіантів розміщення k додаткових пасивних бітів (які відповідають пасивним байтам в різниці) у b-бітних векторах активізації. Дане значення відображає ситуацію, коли пасивні байти будуть розміщені по одному в різних циклах. У такому випадку кількість розміщень максимальна і іншими варіантами можна знехтувати. Це значення залежить від розміру блоку b і кількості циклів R і завжди буде залишатися таким самим незалежно від виду використовуваних перетворень.

2) $2^{-0,0904 \cdot R} \cdot p_{сл}$ – ймовірність основний БДХ для R-циклового 128-бітного шифру Rijndael. Більшість циклів основний БДХ – це переходи різниць з усіма активними байтами на вході і виході, тому значення $2^{-0,0904}$, як зазначено в підрозділі 4.2, формується так: ймовірність переходу різниці з одночасно всіма активними байтами в таку ж різницю на виході для однієї колонки (значення в правому нижньому кутку в табл. 2.4) зводиться до степеня кількості колонок в блоці: $2^{-0,0226 \cdot 4} = 2^{-0,0904}$. З таблиць 4.6 і 4.7 видно також, що значення в правому нижньому кутку майже повністю повторюється і в інших осередках останньої колонки починаючи з рядка, яка відповідає двом активним байтам у вхідній різниці. Значення в цьому рядку в останній колонці формується як $1 - C_m^{m-1} \cdot \frac{1}{255} = 1 - \frac{m}{255}$, де m – кількість

байтів в колонці блоку. Тоді можна записати загальну формулу для значення $2^{-0,0904}$ так:

$$2^{-0,0904} \approx \left(1 - C_m^{m-1} \cdot \frac{1}{255}\right)^{\frac{b}{m}} = \left(1 - \frac{m}{255}\right)^{\frac{b}{m}}. \qquad (4.16)$$

З огляду на вираз (4.16) більш загальний вид формули ймовірності БД (4.15) буде наступним:

$$P_{БД}^{(R)} = \sum_{k=0}^{(R-1)b} (C_R^k \cdot b^k \cdot 2^{-8k} \cdot \left(1 - \frac{m}{255}\right)^{\frac{b}{m}R} \cdot p_{сл}) + P_{ДодНетипБДХ} \qquad (4.17)$$

де R – число циклів; b – розмір блоку в байтах; m – кількість байтів в одній колонці блоку.

З формули (4.16) видно, що збільшення m призводить, з одного боку, до незначного зменшення значення в дужках, з іншого боку, – до зменшення показника ступеня. У підсумку, зміна загального значення не суттєве, що і підтверджується обчислювальними експериментами, результати яких представлені в табл. 4.32.

Таблиця 4.32 – Ймовірності переходу векторів активізації з усіма активними байтами на вході і виході перетворення MixColumn

b	m	b/m	Ймовірність переходу
16	16	1	0.93725490196078
	8	2	0.93823913879277
	4	4	0.93871587874477
	2	8	0.93895056138023
32	32	1	0.87450980392157
	16	2	0.87844675124952
	8	4	0.88029268156260
	4	8	0.88118750100757
	2	16	0.88162815671624

На рис. 4.19 наведені в табл. 4.32 дані представлені графічно.

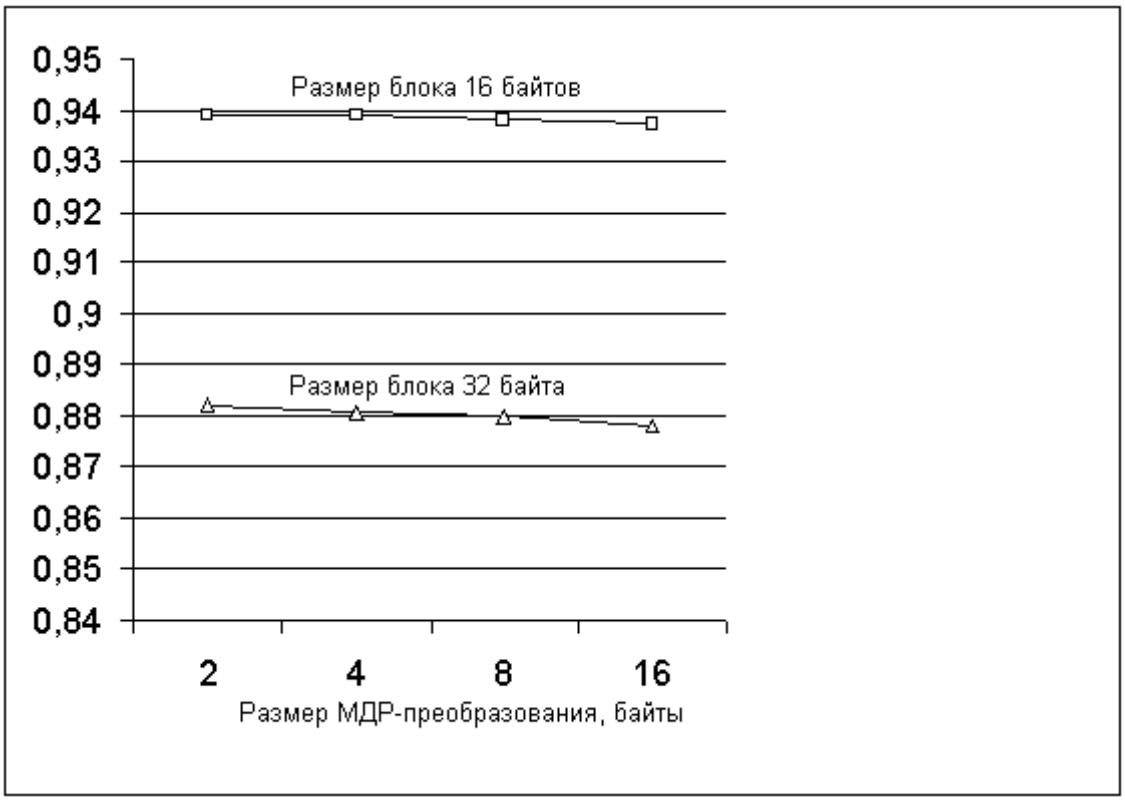


Рисунок 4.19 – Залежність значення виразу (4.7) від розміру блоку і розміру МДР-перетворення

Таким чином, можна сказати, що розглядається значення (вираз (4.16)) більшою мірою залежить від розміру блоку, ніж від структури циклових перетворень, але тільки за умови, що в кожному циклі присутні одне або декілька перетворень з випадковими байтовими диференціалами (див. визначення 4.1), які в сукупності покривають весь блок. Тепер, можна сформувати другу умову.

Умова 4.2. У кожному циклі шифру повинно бути присутнім перетворення з випадковими байтовими диференціалами (визначення 4.1), яке покриває весь блок, або кілька таких перетворень, які покривають весь блок в сукупності.

3) 2^{-8k} – у стільки разів відповідно до твердженням 4.3 повинна знижується ймовірність будь-якої БДХ при додаванні k додаткових пасивних байтів в будь-яких циклах до основної БДХ.

Якщо циклове перетворення містить операцію, яка покриває весь блок і є перетворенням з випадковими усіченими байтовими диференціалами (див. Визначення 4.1), то це гарантує зниження ймовірності БДХ на 2^{-8} для кожного додаткового пасивного байта. Однак, часто зустрічається і ситуація, коли блок даних покривається декількома перетвореннями з випадковими усіченими байтовими диференціалами, наприклад, перетворення MixColumn в шифрі Rijndael з розміром блоку 128 бітів складається з чотирьох таких окремих перетворень для чотирьох колонок. У цьому випадку можливі багатоциклових ефективні БДХ, які виходять за рахунок подачі нульової різниці на одну або кілька частин такого перетворення.

Під умову зниження ймовірності додаткових БДХ не потрапляють *r*-циклові БДХ, перші *r*-1 циклів яких покривають ефективні (*r*-1)-циклові БДХ. Імовірність таких БДХ вище, ніж вірогідність основної БДХ, помноженої на 2^{-8k}, але, обов'язково, нижче ймовірності основної БДХ. Такі БДХ в підрозділі 4.2 були названі нетиповими додатковими БДХ. Як видно з (4.17) ці БДХ також можуть збільшити ймовірність БД. Але для істотного збільшення ймовірності БД необхідна наявність досить великої кількості ефективних (*r*-1)-циклові БДХ. Тому доцільно запропонувати третю умову відсутності ефективних *r*-циклових БД, яке буде обмежувати число ефективних (*r*-1)-циклових БДХ.

Умова 4.3. Кількість ефективних (*r*-1)-циклових БДХ з максимальною вірогідністю $P_{\max}^{(r-1)}$ (зазвичай $P_{\max}^{(r-1)} = 2^{-8}$) для фіксованого вхідного вектора активізації не повинно перевищувати значення $\dfrac{1}{P_{\max}^{(r-1)}}$.

У разі виконання умови 4.3 буде виконуватися нерівність:

$$P_{\max}^{(r)} < 2p_{сл},$$

що свідчить про відсутність ефективних БД.

Таким чином, виконання трьох умов (умови 4.1 - 4.3) для шифру зі стандартним набором перетворень в циклі (нелінійні підстановки, перетворення розсіювання, перетворення введення секретності) свідчить про відсутність ефективних БД і, отже, про захищеність від атаки усічених байтових диференціалів. Набір цих умов становить модель захищеного від атаки усічених байтових диференціалів БСШ. Перевірка цих умов, наприклад, для Rijndael-подібних SPN-шифрів, може бути виконана з мінімальною складністю (виконання всіх умов доводиться теоретично).

4.4.6 Порівняння з відомими результатами

В роботі [96] представлені результати аналізу усічених байтових диференціалів шифру Rijndael-128. Цей алгоритм зі скороченим останнім циклом (без операції MixColumn) розглядався в цих роботах. Операція MixColumn – єдине перетворення, яке вносить невизначеність в проходження вектора активізації. Тому ймовірність проходження вектора активізації через r-циклової шифр зі скороченим останнім циклом і через (r-1)-цикловий шифр з повним набором перетворень буде однаковою. Отже, висновок про відсутність ефективних байтових диференціалів для 3 і більше циклів шифру Rijndael-128 (теорема 4.3) узгоджується з результатами з [96] про відсутність ефективних байтових диференціалів для 4-циклового шифру зі скороченим останнім циклом.

В роботі [96] представлена інформація про 4-цикловий звичайний диференціал з ймовірністю $1{,}0065 \cdot 2^{-128}$ і про 5-цикловий звичайний диференціал з ймовірністю $1{,}00007 \cdot 2^{-128}$ отриманих на основі байтових диференціалів, проте, на наш погляд таке незначне збільшення ймовірності в порівнянні з 2^{-128} не може привести до ефективної атаки.

Таким чином, отримані в розділі результати не суперечать відомим висновкам з роботи [96].

4.4.7 Висновки по підрозділу

1. Вперше запропоновано метод оцінювання стійкості шифрів до атаки усічених байтових диференціалів, заснований на (теоретичному) обгрунтуванні верхніх меж ймовірностей байтових диференціалів (БД). В основі методу лежить запропонована класифікація байтових диференціальних характеристик (БДХ) разом з оцінками ймовірностей і кількості кожного виду БДХ. За рахунок того, що запропонований метод аналізує ймовірності БД, він володіє більш високим ступенем достовірності в порівнянні з методами, заснованими на визначенні максимальної ймовірності БДХ. На відміну від відомого методу, який аналізує ймовірності БД і складність якого експоненціально залежить від розміру блоку і числа циклів, складність запропонованого методу для Rijndael-подібних шифрів не залежить від параметрів шифру, що розширює область використання запропонованого методу і робить можливим його застосування для шифрів з великими розмірами блоків.

2. Продемонстровано застосування запропонованого методу оцінювання стійкості шифрів до атаки усічених байтових диференціалів для шифру Rijndael-128. Вдалося довести відсутність ефективних байтових диференціалів для 3 і більше повних циклів шифру. Отримані результати не суперечать відомим, що свідчить про правильність одержуваних за допомогою запропонованого методу результатів.

3. Практична значимість запропонованого методу полягає в тому, що з його допомогою вдалося вперше обгрунтувати відсутність ефективних БД для SPN і Фейстель-подібних шифрів з певною кількістю циклів, які використовують Rijndael-подібну структуру перетворень і володіють великим розміром блоку (256 і більше бітів). До таких шифрів відносяться:

- «Калина» з усіма розмірами блоку (128, 256 і 512 бітів) з 3 і більше повними циклами;

- використовувані в алгоритмі хешування Groestl 512-бітові шифри P і Q з 3 і більше повними циклами і 1024-бітові шифри P і Q з 7 і більше повними циклами;

- використовуваний в алгоритмі хешування Whirlpool 512 бітний шифр W з 3 і більше повними циклами;

- використовувані в алгоритмі хешування «Купина» 512-бітові шифри P і Q з 3 і більше повними циклами і 1024-бітові шифри P і Q з 7 і більше повними циклами.

4. Запропоновано модель захищеного від атаки усічених байтових диференціалів БСШ. Модель дозволяє на основі результатів пошуку ефективних БДХ і аналізу властивостей використовуваних в шифрі перетворень розсіювання зробити висновок про відсутність ефективних БД і захищеності від атаки усічених байтових диференціалів. В основі моделі для шифру зі стандартним набором перетворень в циклі (нелінійні підстановки, перетворення розсіювання, перетворення введення секретності) лежать три основні умови (умови 4.1 - 4.3), перевірка яких, наприклад, для Rijndael-подібних SPN-шифрів, може бути виконана з мінімальною складністю (виконання всіх умов доводиться теоретично).

За результатами розділу опубліковані роботи [158-165].

4.5 Метод доказу відсутності нездійсненних диференціалів для блокових симетричних шифрів

4.5.1 Критерій відсутності нездійсненних диференціалів для блокових симетричних шифрів

На відміну від більшості відомих підходів, які спрямовані на пошук НД, наш підхід, викладений в [169,170], спрямований на обґрунтування відсутності НД.

Основна ідея пропонованого підходу полягає в тому, щоб обґрунтувати існування деякої різниці на проміжному етапі шифрування, яка може бути

отримана для будь-якої вхідної або вихідної різниці. Рис. 4.20 пояснює цю ідею.

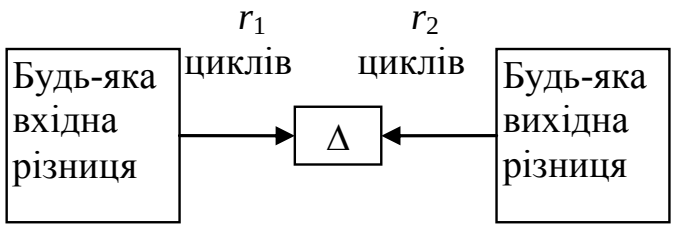


Рисунок 4.20 – Пропонований підхід до обґрунтування відсутності НД

Сформулюємо це у вигляді такої теореми, яку можна вважати критерієм відсутності НД.

Теорема 4.4. Якщо для БСШ існує деяка різниця Δ, яка може бути отримана з будь-який ненульовий вхідний різниці за r_1 циклів перетворень і яка може бути отримана з будь-який ненульовий вихідний різниці за r_2 циклів, які виконуються в напрямку дешифрування, то для такого БСШ не існує НД з r_1+r_2 і більш циклами.

Доказ. Справедливість теореми досить очевидна, тому що якщо будь-яка вхідна різниця і будь-яка вихідна різниця можуть прийти до проміжного значення різниці Δ, то можливий перехід будь-якої вхідної різниці в будь-яку вихідну різницю, а це значить, що не існує НД. Теорему доведено.

Таким чином, для доказу відсутності НД необхідно визначити кількість циклів r_1 та r_2, за які будь-яка вхідна різниця і будь-яка вихідна можуть прийти до деякого значення різниці Δ.

Коли мова йде про вектори активізації або про байтової різниці на вході, виході і на проміжних етапах, то Δ зазвичай містить відразу всі активні байти (вектор активізації складається з усіх «1»). Такі НД будемо називати байтовими НД (БНД). Теорему 4.4 можна переформулювати наступним чином для БНД.

Теорема 4.5. Якщо для БСШ існує деяка байтова різниця Δ, яка може бути отримана для будь-якого вхідного вектора активізації за r_1 циклів перетворень і яка може бути отримана для будь-якого вихідного вектора активізації за r_2 циклів, які виконуються в напрямку дешифрування, то для такого БСШ не існує БНД з r_1+r_2 і більш циклами.

За допомогою теореми 4.5 можна, наприклад, пояснити відсутність БНД для багатьох Rijndael-подібних шифрів, в тому числі для шифру Rijndael з 128-бітовим блоком. При цьому, область використання теореми 4.5 не обмежується тільки цим видом шифрів. У цьому розділі розглянемо також Фейстель-подібні шифри і шифри побудовані з використанням схеми Лея-Мессі (Lai-Massey).

Для кожної з розглянутих різновидів шифрів була побудована зменшена модель, на якій за допомогою обчислювальних експериментів виконувалася перевірка справедливості отриманих теоретичних результатів.

У таблицях 4.33 і 4.34 приведені алгоритми які були використані для обчислювальних експериментів з пошуку НД і БНД для зменшених 16 бітових моделей шифрів.

Таблиця 4.33 – Алгоритм пошуку НД

	Вхідні дані: Шифрувальне перетворення *E*. Порожній рядок таблиці різниці відповідного розміру.
1	Перебір всіх варіантів вхідної різниці *d*
2	Обнулення рядка таблиці різниці
3	Перебір варіантів ключа *k*
4	Перебір всіх варіантів вхідного значення *x*
5	Інкрементуємо елемент з індексом $E_k(x)+E_k(x+d)$
6	Перевіряємо рядок таблиці різниці на наявність «0». Кажен такий «0» відповідає НД
	Вихідні дані: Знайдені НД.

Таблиця 4.34 – Алгоритм пошуку байтових НД (БНД)

	Вхідні дані: Шифрувальне перетворення E. Порожній рядок таблиці різниці відповідного розміру.
1	Перебір всіх варіантів вхідного вектора активізації
2	Обнулення рядки таблиці різниці активізації
3	Перебір всіх варіантів вхідної різниці d, що відповідають заданому вхідному вектору активізації
3	Перебір варіантів ключа k
4	Перебір всіх варіантів вхідного значення x
5	Інкрементуємий осередок з індексом $E_k(x)+E_k(x+d)$
6	Перебір елементів отриманої таблиці різниці
7	Наявність елемента з ненульовим значенням свідчить про відсутність БНД з даним вихідним вектором активізації
8	Якщо відсіяні усі можливі вихідні вектора активізації, то БНД не знайдені і переходимо до наступного значення вхідного вектора активізації
9	Якщо залишилися невідсіяні вихідні вектора активізації, то кожен з них відповідає знайденому БНД
	Вихідні дані: Знайдені БНД.

4.5.2 Використовувані зменшені моделі

В рамках проведених досліджень будуть розглядатися криптографічні властивості Фейстель-подібних, SPN блокових шифрів і шифрів, побудованих з використанням схеми Lai-Massey, зі зменшеним розміром блоку і ключа (8 або 16 бітів). Доцільність розгляду саме зменшених моделей шифрів пояснюється тим, що повноцінний пошук НД можна провести тільки для шифрів з невеликим розміром блоку. Як операцій перемішування і розсіювання були взяті перетворення, запропоновані в [127] для зменшеної

версії шифру Rijndael. На рис. 4.21 і 4.22 схематично представлені перетворення, які виконуються в розглянутих моделях SPN і Фейстель-подібних шифрів.

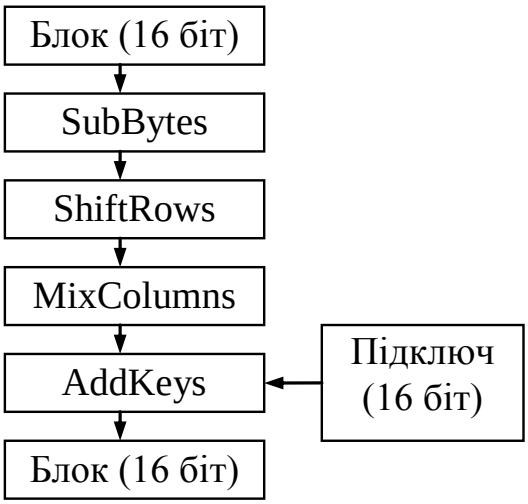


Рисунок 4.21 – Схема одного циклу SPN-шифра

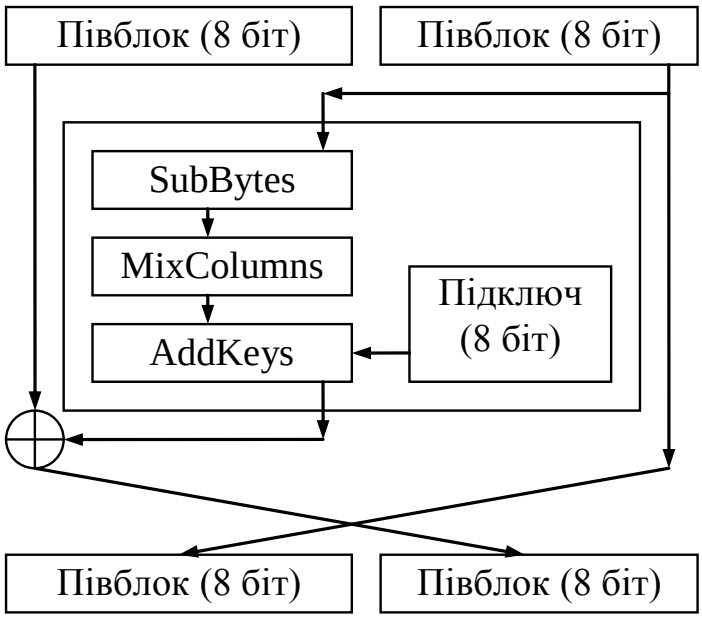


Рисунок 4.22 – Схема одного циклу Фейстель-подібного шифру

До основних особливостей запропонованих зменшених моделей шифрів слід віднести:

- розмір блоку 16 біт, розмір ключа 8 або 16 біт;
- структура блоку для SPN: 2 колонки по 2 4-бітових елемента;

- структура напівблоку для Фейстель-подібного: 2 4-бітових елемента;

- множення елементів кожної колонки на фіксовану МДР-матрицю розміром 2 на 2 над $GF(2^4)$ (MixColumns);

- підстановка 4 в 4 біта (SubBytes);

- число гілок активізації лінійного перетворення MixColumns $B = 3$.

4.5.3 Аналіз Rijndael-подібних шифрів

У цьому підрозділі виконується аналіз умов, при яких теореми 4.4 і 4.5 можуть бути застосовані до Rijndael-подібних SPN-шифрів.

Справедливим є наступне твердження.

Твердження 4.4. Для Rijndael-подібних шифрів з блоком, в якому кількість рядків m не менше, аніж кількість колонок n ($m > n$), не існує байтових НД для 4 і більше циклів з повним набором перетворень.

Доказ. Для доведення твердження необхідно показати, що різниця з одночасно всіма активними байтами може бути отримана при будь-якій початковій різниці як після двоциклового зашифрування так і після двоциклового розшифрування. В цьому випадку виконується теорема 4.5.

Двоциклове шифрування містить послідовність перетворень: MC, SR, MC; а двоциклову розшифрування – послідовність тих же зворотних перетворень: MC^{-1}, SR^{-1}, MC^{-1}.

Розглянемо двоциклове шифрування. Будь-яка ненульова усічена (байтова) різниця має принаймні одну активну колонку на вході першого перетворення MC. Відповідно до табл. 4.6 і 4.7, для будь-якій ненульовій вхідній різниці завжди може бути отримана на виході MC різниця з усіма активними байтами. Перетворення SR поширить активні байти цієї колонки на всі без винятку інші колонки (оскільки $m > n$). Завершальне перетворення MC завжди може перетворити таку різницю на вході в різницю з усіма активними байтами. Аналогічні міркування справедливі і для двоциклового розшифрування. Твердження доведено.

Отриманий результат повністю узгоджується з відомими результатами для шифру Rijndael з 128-бітовим блоком, так як найкращі НД, які були знайдені або використані в відомих роботах, покривають 3 повних і один (останній) неповний цикли [123,125,126,135].

Для Rijndael-подібних шифрів з блоком, в якому рядків менше, ніж колонок ($m < n$), для того, щоб гарантувати відсутність НД потрібно, принаймні, два додаткових циклу перетворень (по одному з кожного боку). Тобто, для таких шифрів можна говорити про доведення відсутності НД не менше, ніж для 6 повних циклів.

За допомогою представлених в таблицях 4.33 і 4.34 алгоритмів був проведений пошук НД і БНД для зменшеної 16-бітової версії алгоритму AES. Результати представлені в таблицях 4.35 і 4.36.

Таблиця 4.35 – Результати пошуку НД для зменшеної версії AES

Кількість циклів	Кількість знайдених НД	Коментарі
4	510	Для кожної вх. різниці з 1 активним S-блоком
5	0	

Таблиця 4.36 – Результати пошуку БНД для зменшеної версії AES

Кількість циклів	Кількість знайдених НД	Коментарі
4 неповних (без MC в останньому циклі)	24	По 6 для кожного вх. вектора активізації з 1 активним S-блоком
4	0	

Результати обчислювальних експериментів з табл. 4.36 підтверджують справедливість доведеного твердження 4.4.

Результати, представлені в табл. 4.35, показують, що при відсутності БНД можуть бути присутніми звичайні НД. Для 4 повних циклів зменшеного AES, не знайдено БНД, але знайдені НД. Знайдені НД можна назвати напівбайтовими, тому що вхідну різницю можна описати вектором активізації з одним активним бітом, а для вихідної різниці важливі самі значення в кожному з активних байтів. У вихідній різниці має бути два активних напівбайти, які до останнього ShiftRow знаходяться в одній колонці. Значення різниці в цих напівбайтів має бути таким, щоб при виконанні останньої операції MC в зворотному напрямку була отримана ненульова різниця тільки в одному напівбайті. Подібні напівбайтові НД для 4 повних циклів існують і для повнорозмірних Rijndael-подібних шифрів, але відомостей про них у доступній літературі знайдено не було. Можливо, використання цих напівбайтових НД може зробити відомі атаки більш ефективними. Однак дане питання потребує більш ретельного дослідження.

Під умови затвердження 4.4 потрапляють всі Rijndael-подібні SPN-шифри, що дозволяє вперше довести відсутність байтових НД для 4 і більше циклів шифру «Калина» з усіма розмірами блоків, для 512 бітових блокових шифрів, які використовуються в хеш-функціях Whirlpool, Groestl і «Купина».

4.5.4 Аналіз шифрів, побудованих з використанням ланцюга Фейстеля

Схема Фейстеля – одна з найбільш поширених схем сучасних БСШ. Як шифру, для досліджень взято алгоритм, який за структурою близький до шифрів «Торнадо» [167] і «Лабіринт» [168]. У кожному циклі виконується SL-перетворення, схема якого представлена на рис. 4.23.

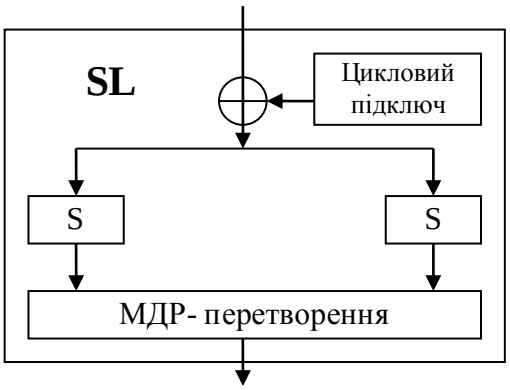


Рисунок 4.23 – SL- перетворення

Важливим моментом є те, що МДР-перетворення (аналог MixColumn в Rijndael-подібних шифрах) охоплює весь опрацьований напівблок. Тому за один цикл таке SL-перетворення може будь-яку ненульову різницю на вході трансформувати в різницю з усіма активними байтами в напівблоці на виході. Загальна схема трьох циклів перетворень представлена на рис. 4.24.

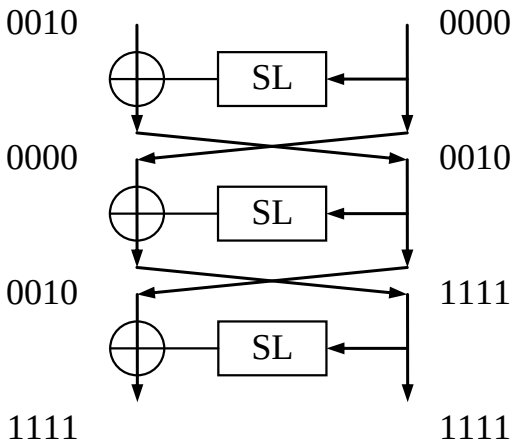


Рисунок 4.24 – Схема трансформації байтової різниці для 3 циклів

Використовуючи теорему 4.5, покажемо справедливість наступного твердження.

Твердження 4.5. Для розглянутого шифру (схема Фейстеля і в цикловому перетворенні МДР-перетворення покриває весь напівблок) не існує БНД, що покривають 6 і більше циклів.

Доказ. Для доведення твердження необхідно показати, що за 3 циклу будь-яка початкова різниця може бути перетворена в різниця з одночасно всіма активними байтами.

Перший цикл може містити тривіальний перехід нульової різниці через SL-перетворення (рис. 4.24). Тоді, незалежно від виду початкової різниці в лівому напівблоці, на вхід SL-перетворення другого циклу надійде ненульова різниця (містить, принаймні, один активний байт). Відповідно до ймовірностей переходів з табл. 4.6, вихід такого SL-перетворення завжди може містити відразу всі активні байти для всього півблоку незалежно від вхідного значення (для всіх ненульових вхідних різниць остання колонка табл. 4.6 містить значення ймовірності близькі до 1).

Далі, це значення різниці надійде на вхід SL-перетворення третього циклу. Отже, на виході знову може бути отримана різниця з одночасно всіма активними байтами (рис. 4.24). Таким чином, після 3 циклів завжди є можливість отримання вихідний різниці з одночасно всіма активними байтами в блоці для будь-якої вхідної різниці.

Так як розшифрування виконується за такою ж схемою, то 3 цикли розшифрування також дозволяють для будь-якої початкової різниці отримати різницю з одночасно всіма активними байтами в блоці. Тоді в термінах теореми 4.5 для даного шифру $r_1 = r_2 = 3$. Твердження доведено.

Як і в попередньому підрозділі для перевірки отриманих теоретичних висновків за допомогою представлених в таблицях 4.33 і 4.34 алгоритмів був проведений пошук НД і БНД для зменшеною 16-бітової версії алгоритму. Результати представлені в таблицях 4.37 і 4.38.

Таблиця 4.37 – Результати пошуку НД для зменшеної версії Фейстель-подібного шифру

Кількість циклів	Кількість знайдених НД	Коментарі
7 (S-блок max_dif = 10)	12	Наприклад, 0x0100-0x0001
7 (S-блок max_dif = 4)	8	Наприклад, 0x0100-0x0001
8	0	

Таблиця 4.38 – Результати пошуку БНД для зменшеної версії Фейстель-подібного шифру

Кількість циклів	Кількість знайдених БНД	Коментарі
5	4	По два для вхідних векторів активізації 1000 та 0100
6	0	

Експерименти з пошуку звичайних НД були проведені для підстановок з різним максимальним значенням в таблиці різниці. Це значення вказано в першій колонці табл. 4.37 для випадків, коли були знайдені НД. У першому випадку (перший рядок табл. 4.37) максимальна ймовірність проходження ненульової різниці для підстановки 4 в 4 біта складає 10/16 (S-блок max_dif = 10), а в другому (другий рядок табл. 4.37) 4/16 (S-блок max_dif = 4).

Представлені результати показують, що диференціальні властивості нелінійних підстановок не роблять вирішального впливу на стійкість БСШ до атаки НД. При цьому, цілком очікувано, що при більшому максимальному значенні в таблиці різниці знайдено більше НД, так як більше максимальне значення свідчить про більшу кількість нулів (заборонених переходів) в таблиці різниці.

Як і в попередньому підрозділі, відсутність БНД не означає відсутність НД для розглянутого шифру. На відміну від SPN-шифрів, де різниця в числі циклів, необхідних для відсутності БНД, від числа циклів, необхідних для відсутності НД, становить 1 цикл (табл. 4.35 і 4.36), в даному випадку ця різниця становить 2 цикли. При цьому результати з табл. 4.38 підтверджують справедливість твердження 4.5.

Під умови твердження 4.5 потрапляють Фейстель-подібні шифри з циклової функцією, в якій використовується МДР-перетворення, що покриває весь напівблок, що дозволяє довести відсутність байтових НД для 6 і більше циклів шифрів Торнадо і Лабіринт з розміром блоку 128 бітів.

4.5.5 Аналіз шифрів, побудованих з використанням схеми Лея-Мессі (Lai-Massey)

До найбільш відомих шифрів, які використовують схему Lai-Massey, відносяться шифри сімейства Fox [171], шифр Мухомор [172]. Схема не є дуже поширеною, тому і недостатньо вивчена. Зокрема, однією з прогалин є стійкість до атаки нездійсненних диференціалів. У доступній літературі ми не зустріли обгрунтування стійкості шифрів з використанням схеми Lai-Massey до атаки нездійсненних диференціалів. В цьому розділі продемонстровано, як з використанням теореми 4.5 може бути обгрунтована стійкість шифру, який використовує схему Lai-Massey.

Як шифру, стійкість якого будемо досліджувати узятий алгоритм, в кожному циклі якого виконується таке ж SL-перетворення як і в попередньому підрозділі (рис. 4.23).

Схема двох циклів схеми Lai-Massey представлена на рис. 4.25.

Використовуючи теорему 4.5 і лему 3.4 з [171], покажемо справедливість наступного твердження.

Твердження 4.6. Для розглянутого шифру (схема Lai-Massey та в цикловому перетворенні МДР-перетворення покриває весь півблок) не існує БНД, що покривають 4 і більше циклів.

Доказ. Відповідно до леми 3.4 [171], в одному з двох поспіль циклів завжди буде ненульова різниця на вході SL-перетворення. Тоді МДР-перетворення може поширити цю різницю на все байти напівблоку. Після такого циклу завжди може бути отримана різниця з усіма активними байтами.

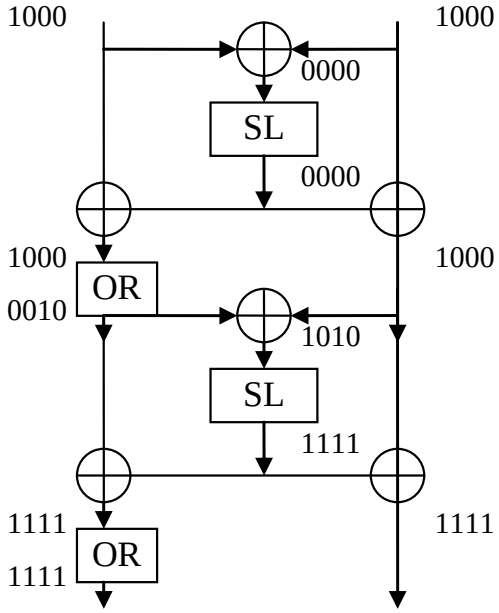


Рисунок 4.25 – Схема трансформації байтової різниці для 2 циклів

Дешифрування виконується за такою ж схемою, як і шифрування (відмінність лише в OR перетворенні), тому два циклу дешифрування також гарантують можливість отримання з будь-якої ненульової різниці різницю з усіма активними байтами. Тоді відповідно до теореми 4.5, для такого шифру не існує БНД, що покривають 4 і більше циклів.

Для перевірки отриманих теоретичних висновків був проведений пошук НД і БНД для зменшеної 16-бітової версії алгоритму. Результати представлені в таблицях 4.39 і 4.40.

Таблиця 4.39 – Результати пошуку НД для зменшеної версії шифру, який використовує схему Lai-Massey

Кількість циклів	Кількість знайдених НД	Коментарі
6 (S-блок max_dif = 10)	Близько 50	Для кожної вхідної різниці типу $0y0y_{16}$, де y – довільне 16-ричне значення
6 (S-блок max_dif = 4)	Близько 20	Для кожної вхідної різниці типу $0y0y_{16}$, де y – довільне 16-ричне значення
7	0	

Таблиця 4.40 – Результати пошуку БНД для зменшеної версії шифру, який використовує схему Lai-Massey

Кількість циклів	Кількість знайдених БНД	Коментарі
3	1	Вх. вект. активіз. 1000, вих. вект. активіз. 0010
4	0	

Для зменшеною моделі шифру, який використовує схему Lai-Massey та Rijndael-подібне циклове перетворення, обчислювальні експерименти підтвердили відсутність 4-циклових БНД (табл. 4.40). У той же час, звичайні НД не були знайдені лише для 7 циклів.

Як і при аналізі Фейстель-подібних шифрів, в табл. 4.39 для максимальної кількості циклів, коли ще існують НД, представлені результати перевірки шифрів з підстановками, що володіють різною максимальною вірогідністю проходження ненульовий різниці. У першому випадку максимальна ймовірність проходження ненульовий різниці для підстановки 4 в 4 біта складає 10/16 (S-блок max_dif = 10), а в другому – 4/16 (S-блок

max_dif = 4). Представлені результати демонструють відсутність вирішального впливу диференціальних властивостей нелінійних підстановок на стійкість БСШ даного виду до атаки НД.

5 РОЗРОБКА МЕТОДІВ ПОБУДОВИ КРИПТОСИСТЕМ НАД ГРУПАМИ

5.1 Теоретико-групові проблеми для побудови симетричних криптосистем на групах

Останнім часом з'явилися численні криптографічні протоколи на основі теоретичних концепцій груп. Такі протоколи ще не привели до практичних схем, щоб конкурувати з подібними RSA і Діффі-Хеллмана, але ідеї цікаві і привели до деякої корисної теорії груп.

З усвідомленням того, що квантові комп'ютери можуть ефективно вирішувати, як завдання з цілочисленної факторизації, так і стандартні варіанти завдання дискретного логарифмування [173] пошук альтернативних криптосистем став більш важливим. Криптосистеми, включаючи групові приклади, які не обов'язково уразливі для квантових супротивників, стали відомі як постквантові криптосистеми. Добре відомим прикладом є криптосистема McEliece, заснована на складності декодування кодів з виправленням помилок. Інші приклади включають криптосистеми на основі решітки та криптосистеми, засновані на великих системах багатовимірних поліноміальних рівнянь.

Цей розділ присвячений груповій криптографії: проектуванню та аналізу криптографічних схем на основі неабелевих груп. Зокрема, ми досліджуємо роль матричних груп в якості платформи для групових криптосистем. Якщо необхідно впровадити криптосистему на основі груп, потрібен ефективний метод подання, зберігання і маніпулювання елементами групи. З цих причин матричні групи є привабливим джерелом неабелевих груп: матриці легко зберігати і представляти на комп'ютері, а лінійна алгебра забезпечує ефективний інструмент для маніпулювання елементами. Але лінійна алгебра також є потужним інструментом криптоаналітика. Дійсно, основна увага приділяється криптоаналізу. Ми демонструємо небезпечність

кількох групових криптосистем, які використовують матричні групи в якості платформи, що робить їх непридатними для використання в реальному світі. Ми розглядаємо кілька протоколів створення ключів, криптосистему з відкритим ключем і групову хеш-функцію. За винятком хеш-функції (де ми будемо мати справу з опором зіткнень), ми працюємо в моделі пасивного противника. Таким чином, ми виходимо з того, що у противника є сила тільки для підслуховування, який знає все про використовувану систему, за винятком секретних ключів і випадкових виборів, зроблених окремими сторонами.

Передбачається, що читач знайомий з основними поняттями абстрактної алгебри і криптографії. Всі необхідні передумови можна знайти, наприклад, в роботах Фрейли [174] і Стінсона [175].

5.1.1 Криптографія з використанням груп

Перша пропозиція використовувати неабелеві групи в криптографії з відкритим ключем належить Вагнеру і Мадьяріку [176] в 1985 році. Криптосистема заснована на жорсткості проблеми Word (або, вірніше, на проблемі вибору слів) для звичайно представлених груп. Однак схема досить теоретична з декількома невирішеними питаннями: критика дається Гонсалесом Васко і Стейнвандтом [177] і Леві-дит-Вехель і Перре [178, 179].

Важливість схеми Вагнера і Мадьяріка полягає в її новизні, яка поклала початок взаємодії між криптографією і комбінаторною теорією груп. Нехай G – група, задана кінцевим поданням. У 1911 році Макс Ден [180] поставив такі завдання:

- Проблема «слова»: задано слово w на образуючих G, визначити чи може $w=1$ в G.

- Проблема спряженості: задані слова u,v на образуючих G, визначити, чи є u,v сполученими елементами в G.

- Проблема ізоморфізму: при заданих двох кінцевих поданнях визначити, чи становлять вони ізоморфні групи.

Вивчення цих проблем виявилося надзвичайно плідним. Наприклад, відомий результат (отриманий незалежно) Новіковим [181] і Буном [182] стверджує, що існує звичайно певна група з рекурсивно нерозв'язною проблемою «слова». Введення групової криптографії поставило під питання деякі з цих добре вивчених проблем. Нехай дано два елементи $g,h\in G$, а також умова, що вони пов'язані в *G*. Завдання полягає в тому, щоб знайти $x\in G$, таке що $g=x^{-1}hx$. Це не проблема рішення, а проблема пошуку, відома як проблема пошуку спряженості. Це становить основу для двох високопрофільних групових протоколів ключових угод запропонованих Аншелом і Голдфелдом [183] в 1999 році і Ко, Лі, Чоном, Ханом, Каном і Парком в 2000 році. До і співавтори також описують схему шифрування з відкритим ключем, безпека якої також заснована на проблемі пошуку спряженості.

Принципову схему узгодження Ко та співавторів можна розглядати як узагальнення протоколу Діффі-Хеллмана на неабелеві групи, причому проблема пошуку спряженості діє як теоретико-груповий аналог завдання дискретного логарифма. Таким чином, до обговорення схем Аншель і співавторів і Ко і співавторів ми згадаємо протокол Діффі-Хеллмана [184] і проблему дискретного логарифма. Пізніше обговоримо теоретико-груповий аналог «безключового» протоколу Шаміра [185].

Протокол узгодження ключів Діффі-Хеллмана.

Нехай $G=\langle g\rangle$ – циклічна група, причому обидва *g* і її порядок *d* є загальновідомими. Щоб створити загальний ключ, Аліса і Боб надходять у такий спосіб:

1. Аліса вибирає рівномірно випадкове ціле число $a\in[2,d-1]$ і відправляє g^a Бобу.

2. Боб вибирає рівномірно випадкове ціле число $b\in[2,d-1]$ і відправляє g^b Алісі.

3. Аліса обчислює $k_a = \left(g^b\right)^a \in G$. Боб обчислює $k_b = \left(g^a\right)^b \in G$. Загальний ключ – $k_a = k_b \in G$.

Безключовий транспортний протокол Шаміра.

Нехай $G = \langle g \rangle$ – циклічна група, причому обидва g і її порядок d є загальновідомими. Щоб створити загальний ключ, Аліса і Боб надходять у такий спосіб:

1. Аліса вибирає рівномірно випадковий ключ $k = g^r \in G$ і ціле число $a \in [2, d-1]$ і посилає $k^a \in G$ Бобу.

2. Боб вибирає рівномірно випадкове ціле число $b \in [2, d-1]$ і посилає $\left(k^a\right)^b = k^{ab} \in G$ Алісі.

3. Аліса обчислює $\left(k^{ab}\right)^{a^{-1}} = k^b \in G$ і відправляє k^b Бобу.

4. Боб обчислює $\left(k^b\right)^{b^{-1}} = k$.

Безпека протоколу Діффі-Хеллмана заснована на припущенні, що, знаючи $g \in G$ і маючи обидва значення g^a і g^b, для зловмисника обчислювально неможливо отримати g^{ab}. Ця проблема відома як проблема Діффі-Хеллмана. Проблема Діффі-Хеллмана пов'язана з відомою проблемою дискретного логарифма (DLP): при заданому $h \in G = \langle g \rangle$ необхідно знайти таке ціле число *t*, щоб $g^t = h$. Якщо буде реалізовано ефективне рішення проблеми DLP, то проблема Діффі-Хеллмана буде ефективно вирішена і протокол Діффі-Хеллмана буде успішно зламаний. Аналогічним чином протокол Шаміра «без ключа» заснований на складності рішення проблеми DLP. Труднощі DLP сильно залежить від того, як представлена група *G*, а не тільки від класу ізоморфізму *G*. Наприклад, проблема DLP тривіальна, якщо $G = \mathbb{Z} / d\mathbb{Z}$ – адитивна група, породжена $g = 1$. Однак, якщо *G* – правильно обрана група, то рішення DLP вважається обчислювально неможливим. На

практиці часто використовують $G = F_{p^l}^*$ для правильно підібраного простого числа *p* і ступеня *l*, або групи точок правильно обраної еліптичної кривої над кінцевим полем.

5.1.2 Від зведення в ступінь до спряженості

Нехай *G* – неабелева група. Для $g, x \in G$ запишемо g^x для $x^{-1}gx$, сполученого з g по *x*. На перший погляд, позначення припускають, що поєднання може використовуватися замість зведення в ступінь в криптографічних контекстах. Припускаючи, що ми можемо знайти групу, в якій проблема пошуку спряженості складна, і припускаючи, що з елементами цієї групи легко виконувати різні дії, можна визначити криптосистеми, які є аналогами криптосистем, заснованих на проблемі DLP. Ко і співавтори запропонували наступний аналог протоколу Діффі-Хеллмана.

Протокол узгодження ключа Ко-Лі-Чона-Хана-Кана-Парка.

Нехай *G* – неабелева група, g – відкритий елемент *G*. Нехай A, B – комутуючі підгрупи в *G*, тобто $ab = ba$ для всіх $a \in A, b \in B$. Групи A, B і *G* – відкриті групи. Щоб створити загальний ключ, Аліса і Боб надходять у такий спосіб:

1. Аліса вибирає елемент $a \in A$ і відправляє $g^a = a^{-1}ga$ Бобу.

2. Боб вибирає елемент $b \in B$ і відправляє $g^b = b^{-1}gb$ Алісі.

3. Аліса обчислює $k_a = \left(g^b\right)^a \in G$. Боб обчислює $k_b = \left(g^a\right)^b \in G$.

4. Оскільки $ab = ba$, ми маємо $k_a = k_b \in G$.

Важлива тонкість протоколу полягає в тому, що, хоча $k_a = k_b$ в якості елементів групи, їх подання до бітового рядку можуть бути різними. Але для багатьох груп загальний ключ може бути отриманий від k_a і k_b. Наприклад, якщо *G* має ефективний алгоритм для обчислення нормальної форми для елемента групи, тоді загальний ключ може бути витягнутий з нормальної

форми k_a і k_b. Ми можемо узагальнити мінімальні вимоги безпеки на платформі групи *G*:

- Група *G* повинна мати дві великі комутуючі підгрупи (наприклад, експоненціального розміру в деякому відповідному параметрі безпеки, наприклад, довжину слова по відношенню до деякої нормальної формі).

- Група *G* повинна мати ефективний алгоритм нормальної форми.

- Проблема пошуку спряженості в *G* повинна бути в цілому складною.

Інтерес до пропозиції Ко і співавторів був зосереджений на їх виборі для групи *G* і підгруп A, B. Візьмемо групу *G* в якості групи кіс B_n на *n* рядках, яка має подання:

$$B_n = \left\langle \sigma_1, \sigma_2, ..., \sigma_{n-1} \,|\, \begin{matrix} \sigma_i, \sigma_j, \sigma_i = \sigma_j, \sigma_i, \sigma_j & for\ |i-j| = 1 \\ \sigma_i, \sigma_j = \sigma_j, \sigma_i & for\ |i-j| \geq 2 \end{matrix} \right\rangle$$

Нехай *l* і *r* будуть такими цілими числами, що $l + r = n$ і визначають підгрупи:

$$A = \langle \sigma_1, \sigma_2, ..., \sigma_{l-1} \rangle, B = \langle \sigma_{l+1}, \sigma_{l+2}, ..., \sigma_{l+r-1} \rangle.$$

Група кіс є привабливим вибором платформи, так як існує ефективна нормальна форма для елементів групи і групове множення, а також може бути ефективно виконана інверсія.

Однак криптосистема в повному обсязі визначена: крім вибору значень *n*, *l* і *r*, необхідно вирішити, як вибирати елементи $g \in G, a \in A$ та $b \in B$. Так як групи G, A та *B* нескінченні, то вибір елементів неочевидний. Наступний протокол узгодження ключів, запропонований Аншелем і Гольдфелдом [183] має перевагу перед протоколом Ко і співавторів в тому, що комутуючі підгрупи *A* і *B* не потрібні.

Протокол узгодження ключів Аншеля-Аншеля-Гольдфелда.

Нехай G – неабелева група, і нехай елементи $a_1,...,a_l,b_1,...,b_m \in G$ будуть відкритими. Щоб створити загальний ключ, Аліса і Боб надходять у такий спосіб:

1. Аліса вибирає слово x в $a_1,...,a_l$ і відправляє $b_1^x,...,b_m^x$ Бобу.

2. Боб вибирає слово y в $b_1,...,b_m$ і відправляє $a_1^y,...,a_l^y$ Алісі.

3. Аліса обчислює x^y, Боб обчислює y^x.

4. Загальний ключ є комутатором $[x,y] := x^{-1}y^{-1}xy$.

Зверніть увагу, що Аліса і Боб можуть обчислювати загальний комутатор: Аліса може помножити x^y на x^{-1}, а Боб може помножити y^x на y^{-1} і потім обчислити зворотне: $[x,y]=\left(y^{-1}y^x\right)^{-1}$. Аншель і співавтори також запропонували використовувати групи кіс в якості платформи для протоколу. Зауважимо ще раз, що схема не повністю визначена: потрібно вказати, як елементи a_i,b_j обрані і як Аліса і Боб генерують слова x,y.

Вибір елементів у випадковому порядку. Дві вищезгадані схеми залишають відкритими деталі специфікації щодо того, як вибирати елементи з нескінченної групи, такої як група кіс. Зауважимо, що протоколи однаково добре працюють для кінцевих груп, і в цьому випадку можна вибирати елементи рівномірно випадковим чином. Але для нескінченних груп значення поняття «випадковим чином» є незрозумілим, і потрібно точно вказати, як будуть відбиратися елементи групи. Нехай $G=\langle X \mid R\rangle$ – кінцево визначена група. Одним з поширених методів вибору елементів з G є вибір випадковим чином послідовності цілих чисел $j_1,...,j_l$ (з деякої кінцевої підмножини в $\mathbb{Z}$) і для кожного $1\le i\le l,$ випадковим чином вибираємо генератор $x_i \in X$. Потім виводиться «випадковий» елемент $x_1^{j_1}\cdots x_l^{j_l}$. Процес вибору елементів групи як для схеми Ко і співавторів, так і для схеми Аншеля і співавторів виявляється критичним для їх безпеки. Ми докладно зупинимося на цьому питанні в розділі 5.1.5.

5.1.3 Заміна спряженості

Схема Ко і співавторів використовує спряженість замість зведення в ступінь в протоколі Діффі-Хеллмана, але є багато інших альтернатив. Наприклад, можна визначити $g^a = \phi(a)ga$ і $g^b = \phi'(b)gb$ для будь-яких фіксованих функцій $\phi: A \to A$ і $\phi': B \to B$ (включаючи карти ідентичності), і схема буде працювати так само добре. У більш загальному випадку ми можемо замінити a і $\phi(a)$ непов'язаними елементами з A.

Після статей Ко і співавторів і Аншеля і співавторів було багато пропозицій, заснованих на варіантах проблеми пошуку спряженості, таких як проблема пошуку розломів і проблема пошуку кручених збігів. М'ясников і співавтори [186] обговорюють ці проблеми.

5.1.4 Симетричні схеми

Теорія груп в основному використовувалася в пропозиціях криптосистем з відкритим ключем і схемах встановлення ключів, але також використовувалася в симетричній криптографії. Одним із прикладів є блоковий шифр PGM, заснований на симетричній групі. У загальному випадку блоковий шифр (такий як DES або AES) можна розглядати як безліч S перестановок на безлічі всіх блоків, індексованих ключем. Питання про те, чи є S фактично групою, впливає на безпеку шифрування в деяких ситуаціях: якщо це група, то шифрування повідомлення двічі з використанням шифрування з різними ключами було б не безпечнішим, ніж одне шифрування. Однак обчислити групу, згенеровану блоковим шифром, часто буває дуже складно. Наприклад, відомо, що група, згенерувала блоковим шифром DES, є підгрупою групи $A_{2^{64}}$, що чергується, [187] з порядком більше ніж 2^{56} (і, отже, S для DES не є групою [188, 189]). Однак про його структуру відомо небагато більше.

Блокові шифри часто будуються як ітеровані конструкції більш простих залежних від ключа перестановок, відомих як раундові функції, і

можна вивчати властивості груп перестановок, згенерованих цими раундовими функціями. Було показано, наприклад, що раундові функції блокових шифрів DES і AES є парними перестановками; крім того, можна показати, що вони генерують змінні групи $A_{2^{64}}$ і $A_{2^{128}}$, відповідно [190, 187, 191].

Конструювання хеш-функцій – це ще одна область симетричної криптографії, в якій цікавим чином використовуються групи. Земор [192] запропонував використовувати проходи через графіки Келі в якості основи для хеш-функцій; найбільш відомою конкретною пропозицією з цієї ідеї є хеш-функція Тілліха і Земора [193].

5.1.5 Криптоаналіз

У цьому розділі ми описуємо методи, розроблені для демонстрації відсутності безпеки багатьох схем, заснованих на групах. У 1969 році Гарсіде [194] дав перший алгоритм вирішення проблеми пов'язаності в групі кіс B_n. Питання ефективності методу Гарсіде залишалося недоторканим до кінця 80-х років. З тих пір було проведено велику кількість досліджень, в значній мірі заснованих на криптографічних додатках, на пошук поліноміального вирішення проблеми спряженості. Для двох кісок $x, y \in B_n$, ідея Гарсіда полягає в тому, щоб побудувати кінцеві підмножини (так звані вершинні безлічі) I_x, I_y від B_n такі, що x спряжений з y тоді і тільки тоді, коли $I_x = I_y$. Ефективне вирішення проблеми спряженості за допомогою цього методу також забезпечило б ефективне рішення проблеми пошуку спряженості (і, отже, робило б протоколи, засновані на проблеми пошуку пов'язаності груп кіс теоретично небезпечними). Однак для даної косички x верхня межа Гарсіда I_x може бути експоненціально великою. Таким чином, завдання полягає в доказі поліноміальної межі розміру відповідної інваріантної безлічі, асоційованої з будь-яким класом спряженості. За минулі роки були зроблені уточнення до методу установки на вищому рівні (такі методи, як –

супервибірка набору, ультравибірка набору і скорочена супервибірка), але межа полінома залишається невловимою. Останнім часом основна увага приділялася ефективному вирішенню кожного з трьох типів кос: періодичної, що зводиться або псевдо-Анасовой відповідно до класифікації Нільсена-Терстона [195, 196, 197].

Однак для цілей криптографії немає необхідності ефективно вирішувати проблему пошуку спряженості, щоб зламати криптосистему на основі коси: можна вільно використовувати специфіку використовуваного протоколу. Алгоритм повинен працювати тільки в значній частині випадків і евристичні алгоритми виявилися вельми успішними.

Дійсно, Хофхейнс і Штейнвандт [198] використовували евристичний алгоритм для вирішення проблеми пошуку спряженості для груп кіс з дуже високими коефіцієнтами успішності. Їх атака заснована на спостереженні, що уявлення спряжених кіс в наборі супервибірки, швидше за все, пов'язані перестановочною косою, яка має особливо просту структуру. Їх атака демонструє невід'ємні недоліки обох пропозицій, як Ко і співавторів так і Аншеля і співавторів. Приблизно в той же час були виявлені кілька інших потужних підходів до реалізації атак, про які ми зараз говоримо.

Атаки по довжині. Представлені Хьюзом і Танненбаумом [199], атаки на основі довжини забезпечують акуратний імовірнісний спосіб вирішення проблеми пошуку спряженості в деяких випадках. Припустимо, що задано екземпляр завдання пошуку спряженості в B_n. Отже, кожному даються коси $x, y^{-1}xy$ і ми хочемо знайти y. Нехай $l : B_n \to \mathbb{Z}$ – придатна функція довжини на B_n (наприклад, довжина нормальної форми елемента). Якщо для деякого числа i можна записати $y = y'\sigma_i$, де y' має коротшу довжину, ніж y, то тоді $l\left(\sigma_i y^{-1} xy \sigma_i^{-1}\right)$ має бути строго менше $l\left(\sigma_j y^{-1} xy \sigma_j^{-1}\right)$ для $j \neq i$. Так що i може бути вгадано і атака повториться для меншого примірника y' від y. Імовірність успіху цієї атаки залежить від використовуваної конкретної

функції довжини. Для груп кіс є ряд відповідних функцій довжини, які відкривають можливість проведення такого роду атак. Гарбер і співавтори [200] і М'ясников і Ушаков [201] дають переконливі атаки як на рішення Ко і співавторів, так і на підхід Аншеля і співавторів.

Атаки лінійної алгебри. Ідея цієї атаки досить проста: візьміть лінійне уявлення групи кіс і вирішите проблему пошуку спряженості за допомогою лінійної алгебри в матричній групі. Існують два добре відомих уявлення групи кіс: уявлення Бурау (зрадницьке для $n \geq 5$) і точне уявлення Лоуренса-Крамера. Хьюз [202] і Лі [203] дають переконливі атаки на протокол Аншеля і співавторів з використанням уявлення Бурау, а Чон і Жун [204] надають алгоритм поліноміального часу злому протоколу Ко і співавторів з використанням уявлення Лоуренса-Крамера.

5.1.6 Обговорення

Багато пропозицій було зроблено для підвищення безпеки схем, обговорюваних у цій главі. Теми варіюються від зміни основної проблеми (і натомість досліджувати такі проблеми, як проблема розкладання, проблема кореневого кореня, проблема зміни спряженості, тощо) до зміни групи платформ (група Томпсона, поліциклічні групи та інші). Крім того, криптографи створили інші криптографічні примітиви, засновані на проблемі пошуку спряженості, такі як схеми аутентифікації і схеми підпису. Тим не менше, немає ніяких відомих криптографічних примітивів, заснованих на будь-якої з цих ідей, які б переконливо переживали вищезгадані атаки.

З іншого боку, групова криптографія викликала деякі природні питання для теоретика в області груп, зокрема вивчення загальних властивостей груп. Наприклад, М'ясников і Ушаков довели, що повні групи кіс PB_n задовольняють сильній загальній властивості вільної групи: для будь-якої породжуючої безлічі PB_n, коли будь-які k елементів обрані «випадково» (як

обговорювалося вище), вони вільно породжують вільну групу рангу k в загальному вигляді.

5.2 Побудова асиметричних криптосистем на основі неабелевих кінцевих груп

На даний момент лише кілька асиметричних криптографічних примітивів залишаються не зламаними. Більшість із них засновані на усвідомлюваних труднощах певних математичних завдань у дуже великих кінцевих абелевих групах, зокрема презентацій. Такими проблемами є:

- проблема факторизації великих чисел;
- задача дискретного логарифмування (DLP) [205], зокрема, подання великих циклічних груп;
- пошук короткого базиса для даної інтегральної решітки $\mathcal{L}$ великої розмірності.

На жаль, через квантові алгоритми П. Шора для розкладання цілого числа й розв'язку DLP [213], відомі криптосистеми відкритого ключа будуть небезпечними, як тільки увійдуть в практику квантові комп'ютери. У недавній доповіді за редакцією П. Нгуен [212] ідентифіковано ці та інші проблеми, з якими стикнеться інформаційна безпека, як наука, у майбутньому.

Теоретичні основи для багатьох сучасних асиметричних криптографічних примітивів полягають у розумінні труднощів розв'язання математичних задач ближчих до теорії чисел, ніж до теорії груп. Теорія чисел в основному займається абелевими групами.

У цій роботі вводиться новий підхід до проектування «закладок» однобічних функцій на основі неабелевих кінцевих груп. Наш основний інтерес виникає зі спостереження, що безпека відкритого ключа криптосистеми MST_2 залежить від вибору секретного епіморфізму. Зокрема, відкритий ключ MST_2 складається з мережі для групи ς і її образу при

деякому епіморфізмі f від ς на групу H, де f є секретним ключем [211]. Використання, що рекомендується, полягає у виборі f як сполучення елементом $g \in \varsigma$.

Дійсно, у деяких класах груп, знання мережі та її образу від g розкриває деяку інформацію про g. Це може використовуватися для атаки на криптосистему MST_2 для цих класів групи [211].

Наше припущення полягає в тому, що випадкові покриття у кінцевих групах породжують однобічні функції. Починаючи з випадкового накриття α для підмножини ς, ми отримуємо двостороннє перетворення $\breve{\alpha}$ від α. Потім, використовуючи $\breve{\alpha}$ й секрет (простий логарифмічний підпис β для центра ς), побудуємо γ, що накриває другу підмножину ς. Зробимо α й γ відкритими та збережемо в секреті «закладку» у системі β, так само, як інформацію про отримання $\breve{\alpha}$ із α.

5.2.1 Вступна інформація

У цьому розділі ми стисло подамо позначення, визначення та деякі основні факти про логарифмічні підписи, накриття для кінцевих груп і їхні породжені відображення. Для отримання більш докладної інформації, ми відсилаємо читача до [210, 211]. Позначення з теорії груп є стандартними й можуть бути знайдені в [207].

Нехай ς – кінцева абстрактна група. Визначимо ширину ς як позитивне ціле число $w = \lceil \log|\varsigma| \rceil$. Позначимо через $\varsigma^{[\mathbb{Z}]}$ сукупність усіх кінцевих послідовностей елементів ς і відобразимо елементи $\varsigma^{[\mathbb{Z}]}$ як однорядкові матриці із записами в ς. Нехай $X = [x_1, x_2 \ldots, x_r]$ і $Y = [y_1, y_2 \ldots, y_s]$ будуть двома елементами у $\varsigma^{[\mathbb{Z}]}$

Визначимо:

$$X \cdot Y = [x_1 y_1, x_1 y_2, \ldots, x_1 y_s, x_2 y_1, x_2 y_2, \ldots, x_2 y_s, \ldots, x_r y_1, x_r y_2, \ldots, x_r y_s].$$

Замість запису $X \cdot Y$ ми також можемо записати $X \otimes Y$ як звичайний тензорний добуток матриць або для короткого написання залишимо XY. Якщо $X = [x_1, \ldots, x_r] \in \varsigma^{|\mathbb{Z}|}$, позначимо через $\bar{X}$ елемент $\sum_{i=1}^{r} x_i$ у груповому кільці $\mathbb{Z}\varsigma$.

Нехай $\alpha = [A_1, A_2, \ldots A_s]$ – послідовність $A_i \in \varsigma^{|\mathbb{Z}|}$, така що $\sum_{i=1}^{s} |A_i|$ обмежена поліномом в $\log|\varsigma|$. Нехай

$$\bar{A}_1 \cdot \bar{A}_2 \cdots \bar{A}_s = \sum_{g \in \varsigma} a_g g, a_g \in \mathbb{Z} \tag{5.1}$$

Нехай S – підмасив ς. Тоді, можемо сказати, що α є:

Накриттям для ς (або S), якщо $a_g > 0$ для всіх $g \in \varsigma$ ($g \in S$).

(i) Логарифмічним підписом для $\varsigma(S)$, якщо $a_g = 1$ для кожного $g \in \varsigma$ ($g \in S$).

Нехай α буде накриттям.

Визначимо $\lambda_{\min} := \min\{a_g : g \in \varsigma\}$, $\lambda_{\max} := \max\{a_g : g \in \varsigma\}$ та $\lambda := \lambda_{\max} / \lambda_{\min}$. Співвідношення λ визначає степінь однорідності α. Говоримо, що α – однорідне накриття, якщо $\lambda = 1$. Зокрема, логарифмічний підпис є однорідним накриттям.

Зверніть увагу, що якщо $\alpha = [A_1, \ldots A_s]$ є логарифмічним підписом для ς, тоді кожен елемент $y \in \varsigma$ може бути однозначно виражений як добуток вигляду:

$$y = q_1 \cdot q_2 \cdots q_{s-1} \cdot q_s \tag{5.2}$$

для $q_i \in A_i$.

Зазвичай, для загальних накриттів, розкладення (5.2) не унікальне й проблема пошуку розкладення для даного $y \in \varsigma$, у загальному випадку, є обчислювально неможливим.

Нехай $\alpha=[A_1,...A_s]$ – накриття для ς з $r_i=|A_i|$. Тоді A_i називаються блоками від α і вектором $(r_1,...,r_s)$ блока довжин r_i і класу α. Визначимо довжину α як ціле число $\ell=\sum_{i=1}^{s} r_i$. Однорідне накриття $\alpha=[A_1,...A_s]$ класу $(r,r,...,r)$ називається $[s,r]$-мережею. Говоримо, що α є нетривіальним, якщо $s\geq 2$ й $r_i\geq 2$ для $1\leq i\leq s$, у протилежному випадку α є тривіальним. Позначимо через $C(\varsigma)$ і $\Lambda(\varsigma)$ відповідні подання накриттів і логарифмічних підписів групи ς.

Нехай $\Gamma=\{(\varsigma_\ell,\alpha_\ell)\}_{\ell\in T}$ – сімейство пар, індексоване за допомогою параметра безпеки ℓ, де ς_ℓ – групи загальної презентації, α_ℓ – особливе накриття для ς_ℓ довжини полінома, який дорівнює ℓ. Вважаємо, що Γ – просте, якщо імовірнісний алгоритм поліноміального часу виконання A, за якого для кожного $g\in\varsigma_\ell$, A приймає (α_ℓ,g) як вхідні й вихідні дані факторизації $\varphi(g)$ від g відносно α_ℓ (як у 5.2) з переважною імовірністю успіху. Або вважаємо, що Γ – випадкове, якщо для будь-якого імовірнісного алгоритму поліноміального часу виконання A, імовірність того, що A успішне у факторизації випадкового елемента g від ς, є незначною.

Для кінцевих груп є елементи $\{(\varsigma_\ell,\alpha_\ell)\}_\ell$, для яких факторизація вважається складною. Для прикладу, нехай q – проста степінь числа, для якого проблема дискретного логарифмування в мультиплікативній групі кінцевого поля $\mathbb{F}_q$ вважається складною. Нехай $2^{\ell-1}\leq q-1<2^\ell$ і нехай ς_ℓ буде раніше згаданою мультиплікативною групою $\mathbb{F}_q^*$. Нехай f – генератор ς_ℓ. Якщо $\alpha_\ell=[A_1,A_2,...A_\ell]$, де $A_i=[1,f^{2^{i-1}}]$, тоді α_ℓ – накриття ς_ℓ й факторизація стосовно α_ℓ зводиться до розв'язання проблеми дискретного логарифмування для ς_ℓ.

Нехай $\alpha=[A_1,A_2,...A_s]$ – накриття групи ς. Нехай $g_0,g_1,...,g_s\in\varsigma$ і розглянемо $\beta=[B_1,B_2,...B_s]$ з $B_i=g_{i-1}^{-1}A_ig_i$ для спеціального випадку, де $g_0=1$ й $g_s=1$, тоді β називається багатошаровим накриттям від α. Зверніть увагу, що β також є накриттям для ς.

Нехай $\alpha=[A_1,A_2,...A_s]$ – накриття класу $(r_1,r_2,...,r_s)$ для ς з $A_i=[a_{i,1},a_{i,2},...,a_{i,r_i}]$ і нехай $m=\prod_{i-1}^{s}r_i$. Нехай $m_1=1$ і $m_i=\prod_{j=1}^{i-1}r_j$ для $i=2,...,s.$ Позначимо τ як канонічну бієкцію від $\mathbb{Z}_{r_1}\oplus\mathbb{Z}_{r_2}\oplus\cdots\oplus\mathbb{Z}_{r_s}$ на $\mathbb{Z}_m$, тобто:

$$\mathbb{Z}_{r_1}\oplus\mathbb{Z}_{r_2}\oplus\cdots\oplus\mathbb{Z}_{r_s}\to\mathbb{Z}_m$$

$$\tau(j_1,j_2,...,j_s):=\sum_{i=1}^{s}j_im_i$$

Використовуючи τ, можемо визначити сюр'єктивне відображення $\breve{\alpha}$, породжене α:

$$\breve{\alpha}:\mathbb{Z}_m\to\varsigma$$

$$\breve{\alpha}(x):=a_{1,j_1}\cdot a_{2,j_2}\cdots a_{s,j_s},$$

де $(j_1,j_2,...,j_s)=\tau^{-1}(x)$. Оскільки τ й τ^{-1} ефективно обчислювані, то відображення $\breve{\alpha}(x)$ також ефективно обчислюване.

З іншого боку, з даним накриттям α і елементом $y\in\varsigma$, щоб визначити будь-який елемент $x\in\breve{\alpha}^{-1}(y)$, необхідно отримати кожне з можливих розкладень класу (2.2) для y і визначити показники $j_1,j_2,...,j_s$ такі, що $y=a_{1,j_1}\cdot a_{2,j_2}\cdots a_{s,j_s}$. Це можливо тільки якщо α – просте. Оскільки вектор $(j_1,j_2,...,j_s)$ визначено, то $\breve{\alpha}^{-1}(y)=\tau(j_1,j_2,...,j_s)$ може бути ефективно обчислено.

Два накриття α й β вважатимуться еквівалентними, якщо $\breve{\alpha}=\breve{\beta}$.

Наведемо невеликий приклад за участю α й β для змінної групи A_5. Класи α й β дорівнюють (5,2,6) і (3,4,5) і $|A_5|=5\cdot2\cdot6=3\cdot4\cdot5=60$

відповідно. У табл. 1 блоки α й β подані вертикально. Щоб ефективно обчислити τ^{-1} й τ прикладемо канонічні логарифмічні підписи τ_α й τ_β адитивної групи $\mathbb{Z}_{60}$ вліво від α і вправо від β. Відповідні класи τ_α й τ_β дорівнюють (5,2,6) і (3,4,5) лише для α й β.

Тепер можемо продемонструвати, як на практиці можна обчислити $\breve{\alpha}:\mathbb{Z}_{60}\to A_5$. Будь-який елемент $x\in\mathbb{Z}_{60}$ можна однозначно записати як суму елементів τ_α, використовуючи тільки один елемент з кожного блоку. Визначення цієї декомпозиції x містить у собі «жадібний» вибір компонентів, одного з кожного блоку, послідовно з нижнього блоку до верхнього й, по суті, визначає $\tau^{-1}(x)=(j_1,j_2,j_3)$. Якщо x_i є елементами A_5 відповідними j_i, то ми обчислюємо $\breve{\alpha}(x)=x_1x_2x_3$. Зокрема, якщо $x=47$, маємо: 47=40+5+2 і компоненти $j_1=2$, $j_2=5$ і $j_3=40$, елементи, що вказують $x_1=(15423)$ $x_2=(24)(35)$ і $x_3=(132)$ від A_5. Тоді можемо обчислити: $\breve{\alpha}(47)=x_1x_2x_3=(15423)\cdot(24)(35)\cdot(132)=(125)$.

Якщо розкладемо $y=\breve{\alpha}(x)$ відносно другого логарифмічного підпису β, то можемо отримати $y=y_1y_2y_3$. З елементів y_i отримуємо відповідні елементи адитивної τ_β і формуємо суму. Для окремого випадку, $y=(125)=y_1y_2y_3=(354)\cdot(253)\cdot(124)$ відповідними компонентами τ_β є 1,3,0.

Таким чином, $\breve{\beta}^{-1}((125))=1+3+0=4$. Необхідно зазначити у даному прикладі, що α й β належать до класу простих логарифмічних підписів, але β, насправді, суперпроста. Далі ми не пояснюватимемо, як ефективно отримати розкладення $y=y_1y_2y_3$, для цього дивіться [6].

Коли група, яка лежить в основі, обрана правильно, бієкція $\breve{\alpha}\breve{\beta}^{-1}$ може використовуватися як криптографічне перетворення з ключем (α,β) у симетричній криптосистемі PGM [5, 6] або як криптографічні примітиви в інших системах.

Таблиця 5.1 – Два логарифмічні підписи від A_5

	τ_α	α		β	τ_β	
	$\mathbb{Z}_{60}$	A_5		A_5	$\mathbb{Z}_{60}$	
$x_1 \to$	0	(1)(2)(3)(4)(5)		(1)(2)(345)	0	$\leftarrow y_1$
	1	(1 2 5 3 4)		(1)(2)(354)	1	
	2	(1 5 4 2 3)		(1)(2)(3)(4)(5)	2	
	3	(1 3 2 4 5)		(1)(23)(45)	0	$\leftarrow y_2$
	4	(1 4 3 5 2)		(1)(253)(4)	3	
$x_2 \to$	0	(1 2 5 3 4)	A_5	(1)(243)(5)	6	
	5	(2 4) (3 5)		(1)(2)(3)(4)(5)	9	
$x_3 \to$	0	(1 3 5 4 2)		(124)(3)(5)	0	$\leftarrow y_3$
	10	(13) (24) (5)		(1)(235)(4)	12	
	20	(1)(2)(3)(4)(5)		(13)(2)(45)	24	
	30	(15)(23)(4)		(1 5 3 4 2)	36	
	40	(132)(4)(5)		(1 4 3 2 5)	48	
	50	(123)(4)(5)				

5.2.2 Опис нової криптосистеми відкритого ключа

Опишемо нову криптосистему, яка називається MST_3. Нехай ς – кінцева неабелева група з нетривіальним центром $\mathbb{Z}$, таким, що ς не розкладаються над $\mathbb{Z}$. Також припустимо, що $\mathbb{Z}$ є досить великим, таким, що пошук перебором у $\mathbb{Z}$ є обчислювально нездійсненним.

Криптографічна гіпотеза, що є основою для нашої криптосистеми полягає в тому, що якщо $\alpha = [\mathrm{A}_1, \mathrm{A}_2, ..., \mathrm{A}_s] := (a_{i,j})$ – випадкове накриття для «великого» підмасива S в ς, то пошук розкладення $g = a_{1j_1} a_{2j_2} \cdots a_{sj_s}$ для будь-якого елемента $g \in S$ відносно α є, загалом, невирішуваною проблемою.

Підготовка. Аліса обирає велику групу ς, описану раніше й генерує:

(1) Простий логарифмічний підпис $\beta = [B_1, B_2, ..., B_s] := (\mathrm{b}_{ij})$ класу $(\mathrm{r}_1, \mathrm{r}_2, ..., \mathrm{r}_s)$ для $\mathbb{Z}$.

(2) Випадкове накриття $\alpha = [\mathrm{A}_1, \mathrm{A}_2, ..., \mathrm{A}_s] := (a_{i,j})$ такого самого класу, як і β для деякої підмножини J від ς такого, що $\mathrm{A}_1, ..., \mathrm{A}_s \subseteq \varsigma \setminus \mathbb{Z}$.

Потім, вона обирає $t_0, t_1 ..., t_s \in \varsigma \setminus \mathbb{Z}$ й обчислює:

(3) $\breve{\alpha} = [\tilde{\mathrm{A}}_1, \tilde{\mathrm{A}}_2, ..., \tilde{\mathrm{A}}_s]$, де $\tilde{A}_i = t_{i-1}^{-1} A_i t_i$ для $i = 1, ..., s$;

(4) $\gamma := (h_{ij}) = (b_{ij} \tilde{a}_{ij})$.

Аліса публікує свій відкритий ключ $(\alpha = (a_{ij}), \gamma = (h_{ij}))$, а $(\beta = (\mathrm{b}_{ij}), (\mathrm{t}_0, ..., \mathrm{t}_s))$ – зберігає як свій закритий ключ.

Шифрування. Якщо Боб хоче відіслати повідомлення $x \in \mathbb{Z}_{|Z|}$ для Аліси, то він:

(1) Обчислює $y_1 = \breve{\alpha}(x)$ і $y_2 = \breve{\gamma}(x)$.

(2) Посилає $y = (y_1 y_2)$ Алісі.

Дешифрування. Тепер, коли Аліса знає y_2, вона визначає:

$$y_2 = \breve{\gamma}(x) = b_{1 j_1} \tilde{a}_{1 j_1} \cdot b_{2 j_2} \tilde{a}_{2 j_2} \cdots b_{s j_s} \tilde{a}_{s j_s} = b_{1 j_1} t_0^{-1} a_{1 j_1} t_1 \cdots b_{s j_s} t_{s-1}^{-1} a_{s j_s} t_s =$$

$$= b_{1 j_1} b_{2 j_2} \cdots b_{s j_s} t_0^{-1} a_{1 j_1} a_{2 j_2} \cdots a_{s j_s} t_s = \breve{\beta}(x) . t_0^{-1} \breve{\alpha}(x) t_s = \breve{\beta}(x) . t_0^{-1} y_1 t_s .$$

І потім вона може обчислити $\breve{\beta}(x) = \mathrm{y}_2 t_s^{-1} y_1^{-1} t_0$.

А згодом Аліса відновлює x з $\breve{\beta}(x)$ використовуючи $\breve{\beta}^{-1}$, який ефективно обчислюємо, оскільки β – проста.

Зауваження 5.1

1. Нехай $\alpha = [\mathrm{A}_1, ..., \mathrm{A}_s]$ – накриття для J, що задовольняє умовам підготовки (2), таким що $\bar{A}_1 \cdot \bar{A}_2 \cdots \bar{A}_s = \sum_{h \in J} a_h h$. І нехай $\lambda = \frac{1}{|J|} \sum_{h \in J} a_h$.

Припущення, що Аліса має можливість побудувати накриття α того самого класу, що й β, має на увазі, що $\lambda |J| \le |Z|$.

Зверніть увагу, що для побудови MST_3, криптографічна гіпотеза, що $\breve{\alpha}$ і $\breve{\gamma}$ є односторонніми функціям, як і раніше, необхідна. Однак, покажемо далі, що гіпотеза може не виконуватися для α, якщо $\lambda_{\min} := \min\{a_h : h \in J\}$ досить велике.

2. Припущення, що ς не розкладається над Z має на увазі, що немає такої підгрупи $H < \varsigma$ із $H \cap Z = 1$, щоб $\varsigma = Z \cdot H$ ($= Z \times H$, де Z – центр групи ς). Без даного припущення система може бути уразливою для атак, заснованих на алгоритмах групової перестановки. Зокрема, якщо наша група є прямим добутком $\varsigma = Z \times H$ і може подаватися як група перестановки розумного степеня (наприклад, ≤ 100000), тоді, використовуючи відповідний сильний генератор для ς й дерева Шрайєра, можна отримати b_{ij} з h_{ij}. Система, таким чином, буде ослаблена.

Подане шифрування є детермінованим шифруванням: той самий відкритий текст даватиме той самий шифротекст щоразу під час шифрування. Проте, випадкове шифрування може бути реалізоване в такий спосіб:

Для шифрування повідомлення $x \in \mathbb{Z}_{|Z|}$, Боб обирає випадкове число $R \in \mathbb{Z}_{|Z|}$, $R \neq 0$ і виконує наступні дії:

(1) обчислює $y_0 = x + R$, де обчислення відбувається в $\mathbb{Z}_{|Z|}$;

(2) обчислює $y_1 = \breve{\alpha}(R)$ й $y_2 = \breve{\gamma}(R)$;

(3) посилає $y = (y_0 y_1 y_2)$ Алісі.

Для дешифрування $y = (y_0 y_1 y_2)$ Аліса перш за все відновлює R з $(y_1 y_2)$, а потім отримає $x = y_0 - R$.

5.2.3 Реалізація та безпека MST_3

У цьому розділі ми пропонуємо клас груп для загальної версії нашої криптосистеми відкритого ключа MST_3. Тут важливим моментом є той факт,

що для довільних членів ς у даному сімействі, ми можемо показати безпеку системи і її стійкість до злому. Для того, щоб зробити аналіз, включаємо в розділ деякі основні позначення й визначення щодо кінцевих p-груп і описання структури Судзукі 2-груп у деяких деталях.

5.2.3.1 Судзукі 2-групи

Перш за все нагадаємо деякі базові факти про кінцеві p-групи, де p – просте число.

Кінцева група ς порядку p називається p-групою, тобто $|\varsigma| = p^n$ для певного позитивного числа n. Найменше спільне кратне порядків елементів ς називається показником ς. Абелева (комутативна) p-група ς з показником p називається елементарною абелевою групою.

Множина $\mathbb{Z}(\varsigma) = \{z \in \varsigma : zg = gz \forall g \in \varsigma\}$ називається центром ς. Відомо, що $\mathbb{Z}(\varsigma)$ є нормальною підмножиною порядку не менше, ніж p для будь-якої p-групи ς. Підмножина ς', що утворена всіма елементами вигляду $x^{-1}y^{-1}xy$ з $x, y \in \varsigma$, називається комутатором підмножини ς. Так звана група Фраттіні ς, що позначається як $\Phi(\varsigma)$, є, за визначенням, перетином усіх максимальних підгруп ς. Якщо ς – p-група, то фактор-група $\varsigma / \Phi(\varsigma)$ є елементарною абелевою групою. Зокрема, якщо ς є 2-групою, то $\Phi(\varsigma) = \langle g^2 \mid g \in \varsigma \rangle$. На закінчення, – елемент порядку 2 у групі називається інволюцією.

Формально, Судзукі 2-група визначена як неабелева 2-група з більш ніж однією інволюцією, що має циклічну групу автоморфізмів, яка переставляє її інволюції транзитивно. Цей клас 2-груп був вивчений і охарактеризований Г. Хігманом [2]. Зокрема, у будь-якій Судзукі 2-групі ς ми маємо $\mathbb{Z}(\varsigma) = \Phi(\varsigma) - \varsigma' = \Omega(\varsigma)$, де $\Omega_1(\varsigma) = \langle g \in \varsigma \mid g^2 = 1 \rangle$ й $|\mathbb{Z}(\varsigma)| = q = 2^m$, $m > 1$. У [2] показано, що порядок ς дорівнює q^2 або q^3. Таким чином, всі

інволюції ς є центром ς, отже $\mathbb{Z}(\varsigma)$ й фактор-група $\varsigma/\Phi(\varsigma)$ є елементарними абелевими.

З цього випливає, що всі елементи не в $\mathbb{Z}(\varsigma)$ мають порядок 4, тобто ς має показник 4. Відомо, що ς має автоморфізм ξ порядку $q-1$, який циклічно переставляє інволюції ς [2,4].

У нашій реалізації MST_3 розглядаємо тільки клас Судзукі 2-груп з порядком q^2. Використовуючи позначення Хігмана, Судзукі 2-група з порядком q^2 буде позначена як $A(\mathrm{m},\theta)$. Нехай $q=2^m$ з $3\le m\in N$ є таким, що поле $\mathbb{F}_q$ має нетривіальний автоморфізм θ непарного порядку. Тут мається на увазі, що m не є степенем 2. Тоді групи $A(\mathrm{m},\theta)$ існують.

Насправді, якщо ми визначаємо $\varsigma := \{S(a,b) \mid a,b\in\mathbb{F}_q\}$, де $S(a,b)=\begin{pmatrix}1 & 0 & 0\\ a & 1 & 0\\ b & a^\theta & 1\end{pmatrix}$ є матрицею 3×3 над полем $\mathbb{F}_q$, це показує, що група ς ізоморфна $A(\mathrm{m},\theta)$. Отже, ς має порядок q^2 і ми маємо

$$Z := \mathbb{Z}(\varsigma)=\Phi(\varsigma)=\varsigma'=\Omega_1(\varsigma)=\{S(0,b)\mid b\in\mathbb{F}_q\}.$$

Оскільки центр $\mathbb{Z}(\varsigma)$ є елементарною абелевою групою порядку q, він може бути ідентифікований з адитивною групою поля $\mathbb{F}_q$. Крім того, фактор-група $\varsigma/\Phi(\varsigma)$ є елементарною абелевою групою порядку q. Тоді легко перевірити, що множення двох елементів у ς здійснюється відповідно до правила:

$$S(a_1,b_1)S(a_2,b_2)=S(a_1+a_2,b_1+b_2+\mathrm{a}_1^\theta a_2).$$

У матричному зображенні Судзукі 2-групи $A(\mathrm{m},\theta)$ можуть бути розглянуті як підгрупи загальної лінійної групи $GL(3,q)$ над $\mathbb{F}_q$.

Зауваження 5.2 Як це було показано в [2], групи $A(\mathrm{m},\theta)$ і $A(\mathrm{m},\phi)$ є ізоморфними тільки якщо $\phi=\theta^{\pm 1}$.

Аналіз безпеки реалізації MST_3 з Судзукі 2-групами $\varsigma = A(\mathrm{m},\theta)$, що розглядається далі, не вимагає явного представлення ς в матричній формі, як це було зроблено вище. Оскільки ми можемо розглядати G як неабелеву 2-групу порядку q^2, $q=2^m$, $m>1$, що має показник 4 такий, що $Z := \mathbb{Z}(\varsigma) = \Phi(\varsigma) = \varsigma'$ з $\mathbb{Z}(\varsigma)$ – елементарна абелева група порядку q. Аргументи дуже залежать від структури ς. Для прикладу, два будь-яких елемента $x, y \in \varsigma$ порядка 4, що належать до різних суміжних підмасивів центра $\mathbb{Z}(\varsigma)$ не комутують, тобто $xy \neq yx$.

У цій реалізації ми обираємо елементи для накриття α відповідно до такої властивості.

Властивість DC. Для кожного A_i, $i=1,...s$ елементи A_i, обрані так, що $x \neq y$, $x, y \in A_i$, тоді xy^{-1} – елемент порядку 4 в ς. Це означає, що різні елементи x та y у A_i не належать тому самому суміжному підмасиву Z.

5.2.3.2 Аналіз безпеки даної реалізації MST_3

Ми можемо передбачити такі типи атак на реалізацію MST_3.

Атака 1. Перша атака намагається витягти інформацію про $(t_0,...,t_s)$ і $\beta = (\mathrm{b}_{ij})$ із знання відкритих $\alpha = (a_{ij})$ і $\gamma = (\mathrm{h}_{ij})$. Однак для зловмисника достатньо отримати логарифмічний підпис β' еквівалентний β, тобто будь-яку зручну β', яка є багатошаровою трансформацією від β. Таким чином, застосовуючи багатошарове перетворення, ми можемо припустити, що перший елемент кожного блока, окрім останнього блока β, тотожний $1 \in \varsigma$. Зловмисник розглядає загальне рівняння:

$$h_{i,j} = b_{i,j} t_{i-1}^{-1} a_{i,j} t_i,\ i=1,...,s,\ i \le j \le r_i, \tag{5.3}$$

де $h_{i,j}$ і $a_{i,j}$ є відкритими.

Оскільки, $b_{i,j}=1$ (4.2) це дає

$$h_{1,1} = t_0^{-1} a_{1,1} t_1 \tag{5.4}$$

Оскільки $t_0 \in \varsigma / Z$ зловмисник має $q^2 - q$ варіантів вибору для t_0, і для кожного такого вибору t_1 повністю визначено з (5.3). Надалі, відібравши t_0, оскільки a_{1j} й h_{1j} відомі, зловмисник може обчислити b_{1j} із $h_{1j} = b_{1j} t_0^{-1} a_{1j} t_1$ для кожного $j \in \{2,...,r_1\}$. Отже, вибір t_0 однозначно визначає всі наступні елементи блока B_1.

За аналогією, знання t_1 й факт того, що $b_{2,1} = 1$ визначає t_2 й усі елементи $b_{2,\mathrm{j}}$ для $j \in \{2,...,r_2\}$. Ітеративно, відібравши t_0, зловмисник може обчислити $t_1,...,t_{s-1}$, всі можливі b_{1j} для $i \in \{1,...,s-1\}$ і відповідні $j \in \{1,...,\mathrm{r}_i\}$.

Зараз, перший елемент $b_{s,1}$ останнього блоку B_s перебуває в Z, але, як і раніше, залишається невизначеним. Усі q варіантів вибору для $b_{s,1}$ й для кожного з таких варіантів, t_s і всі елементи останнього блоку повністю визначені. Отже, варіанти $q^2 - q$ для t_0 й варіанти q для $b_{s,1}$, тобто маємо $(q-1)q^2$ варіантів для $(\mathrm{t}_0, \mathrm{b}_{s,1})$, кожний з яких повністю визначає $(\mathrm{t}_0,...,\mathrm{t}_s;\beta)$.

Якщо t_0 заміняється $t_0 Z$, де $z \in Z$, зберігаючи при цьому відкриті ключі α та γ, як і секретний ключ β, інваріантним, то легко перевірити із (4.2), що $(\mathrm{t}_0, \mathrm{t}_1, ...\mathrm{t}_s)$ заміняється $(\mathrm{t}_0\mathrm{z}, \mathrm{t}_1\mathrm{z}, ..., \mathrm{t}_s\mathrm{z})$. Отже, з погляду зловмисника, варіанти вибору $(\mathrm{t}_0,...\mathrm{t}_s)$ поділяються на класи еквівалентності розміром $|Z| = q$ кожний. Тобто досить обрати одне значення t_0 з кожного різного суміжного підмасиву ς по модулю Z. З цього випливає, що насправді, зловмисник має $\dfrac{(q-1)q^2}{q} = q(q-1)$ можливих варіантів для керуючої пари $(\mathrm{t}_0, \mathrm{b}_{s,1})$. Але оскільки q, можливо, є дуже великим, то даний тип атаки здійснити неможливо.

Атака 2. Метою даної «атаки вибору відкритого тексту» є визначення β й $(\mathrm{t}_0, \mathrm{t}_s)$ з рівнянь:

$$y_2 = \breve{\beta}(x) t_0^{-1} y_1 t_s, \; x \in \mathbb{Z}_{|Z|}, \tag{5.5}$$

що еквівалентно

$$\breve{\beta}(x) = \mathrm{y}_2\, t_s^{-1} y_1^{-1} t_0, \tag{5.6}$$

де $y_1 = \breve{\alpha}(x)$ й $y_2 = \breve{\gamma}(x)$.

Зловмисник вживає спроби обчислити достатню кількість значень $\breve{\beta}(x_i)$ для того, щоб відновити β, використовуючи пропозицію 5.1. в [7]: пропозиція стверджує, що якщо ς – група перестановки степеня N і якщо β є відомим значенням класу $(r_1,...,r_s)$, тоді можна відновити логарифмічний підпис еквівалентний β, використовуючи певне значення $1-s+\sum_{i=1}^{s} r_i$ правильно відібраних значень $\breve{\beta}(x_i)$.

Зазначимо також, що висновок пропозиції 4.1 залишається в силі для абстрактних груп, тобто умова, що ς – група перестановок, не використовується або необхідна в доведенні пропозиції.

Нехай $\{x_1,\ldots,x_n\}$ – набір відкритих текстів, обраних зловмисником, з яких він припускає отримати інформацію про β. Маємо

$$\breve{\beta}(x_i) = \mathrm{y}_{i,2}\, t_s^{-1} y_{i,1}^{-1} t_0,\ i = 1,...,n, \tag{5.7}$$

де $y_{i,1} = \breve{\alpha}(x_i)$ й $y_{i,2} = \breve{\gamma}(x_i)$.

Зловмисник намагається обчислити або вгадати *n* різних значень $\breve{\beta}(x_i)$, для того щоб відновити β. Зверніть увагу, що у кожному (4.6) тільки $y_{i,1}$ і $y_{i,2}$ є відомими значеннями. Насамперед, маємо:

$$\mathrm{y}_{i,2}(y_{i,1}^{-1})^{t_s} t_s^{-1} t_0 = \mathrm{y}_{i,2}\, y_{i,1}^{-1}\, \mathrm{y}_{i,1}(y_{i,1}^{-1})^{t_s} t_s^{-1} t_0 \in Z$$

Оскільки $\mathrm{y}_{i,1}(y_{i,1}^{-1})^{t_s} \in \varsigma' = Z$, це означає, що

$$t_s^{-1} t_0 \in \mathrm{y}_{i,2}\, y_{i,1}^{-1} Z$$

або, що тотожно

$$\mathrm{t}_s \in \mathrm{t}_0 \mathrm{y}_{i,2}\, y_{i,1}^{-1} Z,\ i = 1,...,n. \tag{5.8}$$

Припустимо, що $\mathrm{y}_{i,2}\, y_{i,1}^{-1} Z \neq \mathrm{y}_{i,2}\, y_{\mathrm{j},1}^{-1} Z$ для пари $i \neq j$.

Тоді, $\mathrm{t}_s \in \mathrm{t}_0 \mathrm{y}_{i,2}\, y_{i,1}^{-1} Z \cap \mathrm{t}_0 \mathrm{y}_{i,2}\, y_{\mathrm{j},1}^{-1} Z = \varnothing$, що суперечить факту існування принаймні однієї пари (t_0, t_s), яка задовольняє (5.7).

Отже, маємо

$$\mathrm{y}_{i,2}\, y_{i,1}^{-1} \in \mathrm{y}_{1,2}\, y_{i,1}^{-1} Z \text{ для } i = 1,...,n.$$

Задамо $w := \mathrm{y}_{1,2}\, y_{i,1}^{-1}$.

Оскільки $t_0 \in \varsigma / Z$, є $q^2 - q$ можливостей для t_0. Якщо t_0 обрано, тоді $t_s \in t_0 w Z$, тобто є q можливостей для t_s. Отже, маємо $q(q-1)q$ «прийнятних» пар (t_0, t_s).

Крім того, якщо (t_0, t_s) задовольняє (5.8), то утворюється пара $(t_0\,\mathrm{z}, t_s\,\mathrm{z})$ з $z \in Z$, тобто, для кожної пари розв'язку (t_0, t_s) в (5.7), одна має q асоційованих розв'язків $(t_0\,\mathrm{z}, t_s\,\mathrm{z})$ з $z \in Z$.

Припустимо тепер, що (τ_0, τ_s) й (t_0, t_s) задовольняють

$$\mathrm{y}_{i,2}\, t_s^{-1} y_{i,1}^{-1} t_0 = z = \breve{\beta}(x_i) = \mathrm{y}_{i,2}\, \tau_s^{-1} y_{i,1}^{-1} \tau_0 .$$

Отже, маємо $\tau_0^{-1} y_{i,1} \tau_s = t_0^{-1} y_{i,1} t_s$ для $i = 1,...,n$.

Таким чином,

$$\tau_0^{-1}\, \mathrm{y}_{i,1}\, y_{\mathrm{j},1}^{-1} \tau_0 = t_0^{-1} y_{i,1} y_{\mathrm{j},1}^{-1} t_0, \ \forall i, j = 1,...,n. \qquad (5.9)$$

Якщо достатньо (i, j) пар, таких, що різні елементи $y_{i,1} y_{\mathrm{j},1}^{-1}$ утворюють ς (необхідно щонайменше m таких елементів), тоді τ_0 й t_0 породжують один і той же внутрішній автоморфізм ς, тобто

$$\tau_0 \equiv t_0 \bmod Z .$$

Отже, $\tau_0 = t_0 z$ і тоді $\tau_s = t_s z$ для деяких $z \in Z$. Значить, кількість прийнятних пар (t_0, t_s), що дають відмінний $\breve{\beta}(x_i)$ дорівнює

$$\frac{q^2(q-1)}{q} = q(q-1).$$

Результат даного аналізу показує, що зловмисник має побудувати, щонайменше, $q(q-1)$ розв'язків для наборів значень $(\breve{\beta}(x_1),...,\breve{\beta}(x_n))$. Серед

цих можливих розв'язків, тільки одне є правильним. Тобто, імовірність реалізації зловмисником успішної атаки дорівнює $\frac{1}{q(q-1)}$. Цікаво те, що кількість $q(q-1)$ розв'язків для наборів значень $(\breve{\beta}(x_1),...,\breve{\beta}(x_n))$ є кількістю не асоційованих розв'язків (t_0,t_s) для (5.6).

Зауваження 5.3

1. Якщо зловмисник не має достатньої кількості рівнянь типу (5.9), щоб зробити висновок (5.9), тоді є більше можливостей для (t_0,t_s), отже, більше можливих розв'язків для наборів значень $(\breve{\beta}(x_1),...,\breve{\beta}(x_n))$. Оскільки, лише один з можливих розв'язків є правильним, то імовірність успішної атаки навіть менше значення $\frac{1}{q(q-1)}$.

2. Відповідно до пропозиції 5.1 [7], необхідно $1-s+\sum_{i=1}^{s} r_i$ різних значень щоб відновити логарифмічний підпис еквівалентний β.

Визначаємо β-логарифмічний підпис класу $(r_1,...,r_s)$ для Z та $|Z|=q=2^m$.

Нехай $r_i=2^{e_i}$ для $i=1,...,s$.

Тоді

$$2^m=2^{e_1}\cdots 2^{e_s} \text{ і } \sum_{i=1}^{s} e_i=m.$$

Тепер

$$\sum_{i=1}^{s} r_i-s+1=\sum_{i=1}^{s}(2^{e_i}-1)+1>\sum_{i=1}^{s} e_i=m.$$

Дана нерівність засвідчує твердження, зроблене в ході аналізу атаки 2.

5.2.5 Простір і складність витрат часу для обчислень ς

У цьому розділі обговоримо вимоги до розмірності й часу для обчислення $\varsigma=A(m,\theta)$ у матричному поданні. Як і раніше, нехай $q=2^m$, де

$m \geq 3$ не є степенем 2 і нехай θ буде нетривіальним автоморфізмом непарного порядку поля $\mathbb{F}_q$. Нагадаємо, що ς містить всі матриці розмірності 3×3 вигляду

$$S(a,b) = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & a^\theta & 1 \end{pmatrix},$$

де $a,b \in \mathbb{F}_q$.

Центр $Z := \mathbb{Z}(\varsigma) = \{S(0,b) \mid b \in \mathbb{F}_q\}$ від ς є елементарною абелевою групою порядку q. Отже, Z може бути поданий у вигляді векторного простору розмірністю m над $\mathbb{F}_2$.

Як було зазначено раніше у розділі 5.1, множення двох елементів у ς виконується відповідно до правила

$$S(a_1,b_1)S(a_2,b_2) = S(a_1 + a_2, b_1 + b_2 + \mathrm{a}_1^\theta a_2).$$

Ми могли б зберігати елементи групи $S(a,b)$ як пару (a,b), але це потребувало б обчислення деякого a^θ кожного разу, коли ми обчислюємо добуток елементів групи. У свою чергу, кожне обчислення a^θ вимагає $O(m)$ множення в $\mathbb{F}_q$. Тому ефективніше за часом зберігати елементи групи як трійки (a,b,a^θ). Таким чином, добуток $S(a_1,b_1) \cdot S(a_2,b_2)$ ідентифікується трійкою

$$(a_1 + a_2, b_1 + b_2 + \mathrm{a}_1^\theta a_2, \mathrm{a}_1^\theta + \mathrm{a}_2^\theta).$$

І обчислення добутків вимагає лише одне множення та чотири додавання в $\mathbb{F}_q$.

Зменшення вимог до витрат на зберігання елементів групи й висока ефективність операції в 2-групах ς є важливими позитивними факторами для реалізації криптосистеми з групою $\varsigma = A(m,\theta)$ в основі.

5.2.4 MST_3 без відповідності криптографічній гіпотезі для α

Під час порівняння MST_3 з MST_2 виникає один унікальний факт. Він полягає у нашій криптографічній гіпотезі, суть якої в тому, що випадково утворені накриття для великих кінцевих груп створюють однобічні функції.

Для MST_2 криптографічна гіпотеза має фундаментальне значення. Проте, для MST_3 криптографічна гіпотеза для випадкового накриття α може бути пропущена без порушення безпеки системи, якщо α побудована правильно.

Значення $|Z|/|J|$ може бути зображене як середнє значення кількості подавань для кожного елемента J стосовно накриття α. Мається на увазі, що кожне $y \in J$ буде, у середньому, мати $|Z|/|J|$ прообразів у $\mathbb{Z}_{|Z|}$ відносно $\breve{\alpha}: \mathbb{Z}_{|Z|} \to J$. Коли криптографічна гіпотеза для α не враховується, MST_3 залишається безпечним, якщо $|Z|/|J|$ є великим. Для випадку, коли $\breve{\alpha}$ не однобічна функція, тобто, для будь-якого заданого $y \in J$, знаходження $z \in \mathbb{Z}_{|Z|}$ такого, що $\breve{\alpha}(z) = y$ є обчислювально можливим, тоді використання оракула Ω, який видає $z \in \mathbb{Z}_{|Z|}$ для заданого вхідного значення $y \in J$, такого що $\breve{\alpha}(z) = y$ приведе до злому системи MST_3, у середньому, після $|Z|/2|J|$ запитів.

Припустимо, що $x \in \mathbb{Z}_{|Z|}$ – відкритий текст і $y_1 = \breve{\alpha}(x)$. Тоді, якщо $|Z| \geq 2|J|^2$, то оракулу Ω знадобитися не менше, ніж $|J|$ запитів для вхідного значення y_1 для того, щоб знайти x з імовірністю $\geq 1/2$. Оскільки J є великим, будь-яке обчислення з часовою складністю $O(|J|)$ є нездійсненним і умова $|Z| \geq 2|J|^2$ просто означає, що криптографічна гіпотеза для α має бути застосована.

5.2.5 Висновки

Ми показали новий підхід до розробки криптосистеми відкритого ключа, заснованого на накриттях і логарифмічних підписах неабелевых груп кінцевих у конкретному класі. У реалізації загальної версії системи пропонується клас спеціальних 2-груп, який дозволяє провести детальний аналіз, що показує міцність системи. Отримано нижні оцінки складності реалізації для двох типів атак проти даної системи. Результати показують, що, як передбачалося, криптосистема захищена від таких атак, якщо порядок обраної 2-групи досить великий. Крім того, коли основна 2-група подана у вигляді матричної групи, вона має ефективне подання, що дозволяє мінімальний простір для зберігання його елементів і, що ще більш значно, у мінімальний можливий час терміни для множення елементів групи.

5.3 Методи побудови асиметричних криптосистем на узагальнених групах Судзукі та групі Рі

5.3.1 Криптосистема на узагальненій групі Судзукі

Магліверас, Стінсон і ван Транг розробили два підходи до побудови криптосистем відкритого ключа, які називаються MST_1 і MST_2 [214]. У випадку з MST_1 використовуються логарифмічні сигнатури, друга версія *MST*, базується на накриття, які відрізняються від логарифмічних сигнатур тим, що факторизація для них не є унікальною. У третій версії криптосистеми MST_3, яка була представлена в [215], використовують прості логарифмічні сигнатури, такі як накриття. Практичність MST_3показана на Судзукі 2 групі потужності $|G| = q^2$. Хігман показав існування Судзукі 2 груп потужності q^3, а Ханакі показав, що існують узагальнені Судзукі 2 групи розмірності q^l [216,217].

Дана робота присвячена побудові криптосистеми MST_3 на основі узагальнених Судзукі 2 груп.

Нехай $F = GF(2^n)$ – кінцеве поле ступеня 2^n і нехай θ – автоморфізм F. Отримуємо, для позитивного цілого l і $a_1, a_2, ..., a_l \in F$ такий вираз:

$$u(a_1, a_2, ..., a_l) = \begin{pmatrix} 1 & & & & & \\ a_1 & 1 & & & & \\ a_2 & a_1\theta & 1 & & & \\ a_3 & a_2\theta & a_1\theta^2 & 1 & & \\ ... & ... & ... & ... & ... & \\ a_l & a_{l-1}\theta & a_{l-2}\theta^2 & ... & a_1\theta^{l-1} & 1 \end{pmatrix} \in M_{l+1}(F)$$

та

$$A_l(n,\theta) = \{u(a_1, a_2, ..., a_l) \mid a_i \in F\}.$$

Множення визначається як добуток двох матриць наступним чином:

$$u(a_1, a_2, ..., a_l)u(b_1, b_2, ..., b_l) = u(a_1 + b_1, a_2 + (a_1\theta)b_1 + b_2,$$
$$a_3 + (a_2\theta)b_1 + (a_1\theta^2)b_2 + b_3, ..., a_l + (a_{l-1}\theta)b_1 + ... + (a_1\theta^{l-1})b_{l-1} + b_l)$$

Таким чином, $A_l(n,\theta)$ стає групою порядку 2^{nl}. Якщо $l = 2$, то ця група ізоморфна Судзукі 2-групі $A(n,\theta)$.

Застосування матричної групи основі узагальнених Судзукі 2 груп в якості платформи для групових криптосистем запропоновано вперше. визначимо для $1 \le i \le l$

$$G_i = \{u(0, ..., 0, a_i, a_{i+1}, ..., a_l)\}.$$

Нехай $\phi_{l.i-1} : A_l(n,\theta) \to A_{l-1}(n,\theta)$ і $\phi_{l.i-1}(u(a_1, ..., a_l)) = u(a_1, ..., a_{i-1})$. Тоді $\phi_{l.i-1}$ є епіморфізмом і $\ker \phi_{l.i-1} = G_i$. Тоді G_i є нормальною підгрупою $A_l(n,\theta)$, $A_l(n,\theta) / G_i \cong A_{l-1}(n,\theta)$ і, очевидно, G_l є центром $A_l(n,\theta)$ з операцій множення. Підгрупа G_i є абелевою, якщо і тільки $i \ge (l+1)/2$.

Криптосистема MST_3 будується на абелевому центрі групи. Це дозволяє виконати дешифрування з виокремленням логарифмічної сигнатури і подальшою факторизацією. Алгебраїчні атаки на MST_3 не відомі. Посиленням є те, що параметри криптосистеми вибираються випадковим чином. Потужність центру узагальненої групи Судзукі $Z(A_l(n,\theta))$, як випливає з

подання G_l, дорівнює q і ймовірність вгадування шифр тексту буде дорівнює q^{-1}.

Безпека криптосистеми визначається складністю атак розкриття параметрів секретного ключа, які визначаються значеннями елементів групи $A(n,\theta)$. Порядок узагальненої групи Судзукі дорівнює q^l. Тоді ймовірність вгадування параметрів має граничну оцінку в разі рівновірогідного їх завдання q^{-l}, і l-1 раз за показником експоненти менше в порівнянні з групою Судзукі.

Висновки. Криптосистема за узагальненими групами Судзукі з дешифруванням по центру групи програє класичній криптосистемі MST_3 на Судзукі 2 групі за витратами на ключові дані, для досягнення тієї ж ймовірності вгадування зашифрованого тексту.

5.3.2 Криптосистема на групі Рі

Останнім часом з'явилися численні криптографічні протоколи на основі теоретичних концепцій груп. Такі протоколи ще не привели до практичних схем, щоб конкурувати з подібними RSA і Діффі-Хеллмана, але ідеї цікаві і є корисними для теорії груп.

Актуальність пошуку альтернативних криптосистем визначається перш за все тим, що квантові комп'ютери можуть ефективно вирішувати, завдання факторизації і стандартні варіанти завдання дискретного логарифмування. Криптосистеми, включаючи групові приклади, що не уразливі для квантового криптоаналізу, відомі як постквантові криптосистеми. Хорошим прикладом є криптосистема McEliece, заснована на складності декодування кодів з виправленням помилок. Відомі криптосистеми на основі решітки і, засновані на великих системах багатовимірних поліноміальних рівнянь.

Дана робота присвячена криптографії на основі груп Рі. Застосування матричної групи Рі в якості платформи для групових криптосистем запропоновано вперше.

Група Рі $G = Ree(q)$ визначена над полем F_q, $q = 3^{2m+1}$ і має подання

$$Ree(q) = < \alpha(x), \beta(x), \gamma(x), h(\lambda), Y \mid x \in F_q, \lambda \in F_q^x >,$$

де $\alpha(x), \beta(x), \gamma(x)$ – верхні трикутні матриці, $h(\lambda), Y$ – діагональні матриці. Підгрупа $U(q) = \{S(a,b,c) \mid a,b,c \in F_q\}$, де $S(a,b,c) = \alpha(a)\beta(b)\gamma(c)$ є силова 3-підгрупа $Ree(q)$ потужністю $|U(q)| = q^3$. Групові операції над елементами підгрупи визначаються співвідношеннями:

$$S(a_1,b_1,c_1)S(a_2,b_2,c_2) = S(a_1 + a_2, b_1 + b_2 - a_1a_2^{3t}, c_1 + c_2 - a_2b_1 + a_1a_2^{3t+1} - a_1^2a_2^{3t}),$$

$$S(a,b,c)^{-1} = S(-a, -(b + a^{3t+1}), -(c + ab - a^{3t+2})),$$

$$S(a_1,b_1,c_1)^{S(a_2,b_2,c_2)} =$$

$$S(a_1, b_1 - a_1a_2^{3t} + a_2a_1^{3t}, c_1 + a_1b_2 - a_2b_1 + a_1a_2^{3t+1} - a_2a_1^{3t+1} - a_1^2a_2^{3t} + a_2^2a_1^{3t}),$$

$$S(a,b,c)^{h(\lambda)} = S(\lambda^{3t-2}a, \lambda^{1-3t}b, \lambda^{-1}c),$$

де λ примітивний елемент поля F_q, $t = 3^m$.

Властивості групи $G = Ree(q)$:

- $|G| = q^3(q^3 + 1)(q - 1)$, де $\gcd(q^3 + 1, q - 1) = 2$;

- спряженості $U(q)$ мають тривіальне перетинання;

- $Z(U(q)) = \{S(0,0,c) \mid c \in F_q\}$ – центр групи;

- елементи підгрупи $U(q)' = \{S(0,b,c) \mid b,c \in F_q\}$ мають порядок 3;

- елементи підгрупи $U(q) \setminus U(q)' = \{S(a,b,c) \mid a \neq 0\}$ мають порядок 9.

Першою практичною криптосистемою на групах є криптосистема MST_3 запропонована Lempken, Magliveras, van Trung and Wei. Нехай G є кінцева неабелева група з нетривіальним центром Z, з властивістю, що G не розщеплюється над Z (G не може бути представлена як прямий добуток $G = Z \times H$ для деякої підгрупи H). Криптосистема MST_3 визначається генерацією факторизуємих логарифмічної сигнатури $\beta = [B_1, ..., B_S] := (\beta_{ij})$ і випадкового накриття $\alpha = [A_1, ..., A_S] := (\alpha_{ij})$ типу $(r_1, ..., r_S)$ над центром Z,

вибором випадкових елементів групи G $t_0,...,t_S \in G \setminus Z$ лежать поза центром і обчисленням векторів $\gamma := (\gamma_{ij}) = (\beta_{ij}\alpha_{ij})$. Шифрування визначається обчисленням по відомим параметрам (α, γ), а дешифрування по векторах (β, $(t_0,...,t_S)$). Дешифрування реалізується за рахунок комутативності обчислень в центрі Z групи G. Реалізація криптосистеми на групі Рі вимагає побудови логарифмічною сигнатури β по векторах 3^h, де h визначається розміром типу $r_i = 3^h$. Зауважимо, що всі блоки B_i є підгрупами центру групи G. Розмір масивів β і α визначається типом $(r_1,...,r_S)$ для центру Z. Для 128 бітної криптографії, що еквівалентно обчисленням над полем 3^{80}, буде потрібно 360 рядків для криптографії на групі Рі, якщо в якості типу взяти $r_i = 9, s = 40$. У порівнянні з MST_3 по Судзукі 2-групі, мали б 256 рядків для $r_i = 4, s = 64$, а для $r_i = 16, s = 32$ мали б 512 рядків.

Другий важливий аспект – безпека криптосистеми. Безпека криптосистеми визначається складністю атак розкриття параметрів β і $(t_0,...,t_S)$. Алгебраїчних атак не існує, тому що параметри вибираються випадковим чином. Потужність центру Z групи Рі дорівнює q і ймовірність вгадування буде дорівнює q^{-1}. Тому що параметри $t_0,...,t_S \in G \setminus Z$ лежать поза центром в підгрупі $U(q) = \{S(a,b,c) \mid a,b,c \in F_q\}$, і потужність $|U(q)| = q^3$, тоді ймовірність вгадування t_i буде дорівнювати q^{-3}, в q разів менше в порівнянні з групою Судзукі.

5.3.3 Оцінка складності реалізації криптосистем MST_3 на групі Судзукі

Основні положення, опис криптосистеми.

Криптосистема *MST* будується на поняттях логарифмічних підписів, накриття для кінцевих груп, і їх породжених відображеннях.

Нехай ς буде кінцевою неабелевою групою з нетривіальним центром $\mathbb{Z}$, таким що ς не розкладаються над $\mathbb{Z}$. Також припустимо, що $\mathbb{Z}$ є досить великим, таким, що пошук перебором в $\mathbb{Z}$ є обчислювально нездійсненним.

Криптографічна гіпотеза, яка є основою для нашої криптосистеми полягає в тому, що якщо $\alpha = [A_1, A_2, ..., A_S] := (a_{ij})$ – випадкове накриття для «великого» підмасиву *S* в ς, то пошук розкладання $g = a_{1j}, a_{2j}, ..., a_{sj}$ для будь-якого елементу $g \in S$ щодо α є, взагалі, нерозв'язною проблемою.

Підготовка даних. Аліса вибирає велику групу ς, яка описана раніше та генерує простий логарифмічний підпис $\beta = [B_1, B_2, ..., B_S] := (b_{ij})$ класу $(r_1, r_2, ..., r_s)$ для $\mathbb{Z}$.

Випадкове накриття $\alpha = [A_1, A_2, ..., A_S] := (a_{ij})$ того же класу, як β для деякої підмножини *J* від ς такого, що $A_1, ..., A_s \subseteq \varsigma \setminus \mathbb{Z}$.

Далі, вона вибирає $t_0, t_1, ..., t_s \in \varsigma \setminus \mathbb{Z}$ та обчислює: $\breve{\alpha} = [\tilde{A}_1, \tilde{A}_2, ..., \tilde{A}_s]$, де $\tilde{A}_i = t_{i-1}^{-1} A_i t_i$ для $i = 1, ..., s$, $\gamma := (h_{ij}) = (b_{ij}\tilde{a}_{ij})$.

Аліса публікує відкритий ключ $(\alpha = (a_{ij}), \gamma = (h_{ij})$, а $(\beta = (b_{ij}), (t_0, ..., t_s))$ – зберігає як свій закритий ключ.

Шифрування. Якщо Боб хоче відправити повідомлення $x \in \mathbb{Z}_{|Z|}$ для Аліси, він обчислює $y_1 = \breve{\alpha}(x)$ та $y_2 = \breve{\gamma}(x)$ і посилає $y = (y_1 y_2)$ Алісі.

Дешифрування. Коли Аліса знає y_2, вона визначає

$$y_2 = \breve{\gamma}(x) = b_{1j_1}\tilde{a}_{1j_1} \cdot b_{2j_2}\tilde{a}_{2j_2} \cdots b_{sj_s}\tilde{a}_{sj_s} = b_{1j_1}t_0^{-1}a_{1j_1}t_1 \cdots b_{sj_s}t_{s-1}^{-1}a_{sj_s}t_s =$$

$$= b_{1j_1}b_{2j_2} \cdots b_{sj_s}t_0^{-1}a_{1j_1}a_{2j_2} \cdots a_{sj_s}t_s = \breve{\beta}(x).t_0^{-1}\breve{\alpha}(x)t_s = \breve{\beta}(x).t_0^{-1}y_1t_s$$

Далі вона може обчислити $\breve{\beta}(x) = y_2 t_s^{-1} y_1^{-1} t_0$.

А після Аліса відновлює *x* з $\breve{\beta}(x)$ використовує $\breve{\beta}^{-1}$, який ефективно обчислюється, так як β – просте.

Реалізація та безпека MST_3. Реалізація MST_3 запропонована на основі Судзукі 2-групи порядку q^2. Нехай $q = 2^m, m \geq 3$. Елемент групи $\varsigma := \{S(a,b) \mid a,b \in \mathbb{F}_q\}$, де

$$S(a,b) = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & a^\theta & 1 \end{pmatrix}$$

являється матрицею 3x3 над полем $\mathbb{F}_q$, θ – автоморфізм непарного порядку. Оскільки центр $\mathbb{Z}(\varsigma)$ є елементарною абелевою групою порядку q, він є адитивною групою поля $\mathbb{F}_q$. Добуток двох елементів в ς визначаться за правилом

$$S(a_1,b_1)S(a_2,b_2) = S(a_1 + a_2, b_1 + b_2 + a_1^\theta a_2).$$

Групова операція визначається двома сумами елементів в кінцевому полі, возведенням в ступінь θ та одним добутком. Якщо взяти $\theta = 2$, піднесення в ступінь змінюється на множення елемента самого на себе. Таким чином групова операція проста и спрощує обчислення при шифруванні на відміну від криптосистем у кільці цілих чисел та на точках еліптичних кривих, які вимагають піднесення у ступінь великого числа.

Аналіз безпеки виконано в роботі. Стійкість до крипто аналізу має оцінку складності порядку q, що дорівнює розміру кінцевого поля над яким визначається група Судзукі. Криптосистема MST_3 являється стійкою до квантових алгоритмів криптоаналізу, що визначає її перевагу в порівнянні з криптосистемами у кільці цілих чисел та на точках еліптичних кривих.

Простота застосування криптосистеми MST_3 полягає в тому, що програма зашифрування містить таблиці векторів і повинна знаходитися на пристрої користувача, програма розшифрування знаходиться тільки на сервері. Відсутнє зберігання, узгодження та розповсюдження ключів в системі, що є перевагою в порівнянні з симетричними криптосистемами.

Практичний приклад обчислень представлений в у полі $F_{2^{12}}$. Результати обчислень над полями (не оптимізовані до кінця обчислення) представлені в табл. 5.1-5.3.

Таблиця 5.2 – Результати обчислень над 128-бітним полем

128-бітне поле					
Розмір блоків	Час генерації ключових даних, мс	Розмір приватного ключа, байт	Розмір публічного ключа, байт	Час зашифрування 100 КБайт, мс	Час розшифрування 100 КБайт, мс
128[2]→64[4]	56	78830	39761	4749	2711
64[4]->32[16]	59	111726	75217	2388	1487
32[16]→16[256]	169	671918	590609	1205	888

Таблиця 5.3 – Результати обчислень над 256-бітним полем

256-бітне поле					
Розмір блоків	Час генерації ключових даних, мс	Розмір приватного ключа, байт	Розмір публічного ключа, байт	Час зашифрування 100 КБайт, мс	Час розшифрування 100 КБайт, мс
256[2]→128[4]	57	249630	128593	14811	7911
128[4]->64[16]	106	361502	248657	7540	4196
64[16]→32[256]	798	2193054	1967569	3782	2318

Таблиця 5.4 – Затрати на обчислення RSA алгоритмом

Розрядність ключових параметрів, біт	Час генерації ключових даних, мс	Розмір приватного ключа, байт	Розмір публічного ключа, байт	Час зашифрування 100 КБайт, мс	Час розшифрування 100 КБайт, мс
512	3,368	342	92	66,987	641,277
1024	8,685	632	160	117,947	2116,400
2048	63,658	1214	292	243,887	9853,580
4096	707,645	2373	548	591,868	64250,400

6 РОЗРОБКА МЕТОДІВ ПОБУДОВИ КРИПТОПЕРЕТВОРЕНЬ ТА ОЦІНКА СТІЙКОСТІ НА ОСНОВІ ОБЧИСЛЕНЬ НАД ФУНКЦІОНАЛЬНИМИ ПОЛЯМИ ПРОЕКТИВНИХ РІЗНОМАНІТЬ

6.1 Аналіз сучасних вимог до криптографічних примітивів

У першому кварталі 2015 року компанія Google офіційно припинила підтримку SHA-1 (Secure Hash Algorithm) [218]. Практично це означає, що з виходом браузера Chrome версії 41 всі сайти, підписані сертифікатом із зазначеним алгоритмом, будуть визнаватися небезпечними. У свою чергу, компанія Microsoft анонсувавши, спочатку, відмова від 160-бітних хешів з 1 січня 2017 року, але потім випустила анонс відмови від використання SHA-1 вже з другого кварталу 2016 [219,220]. Це стало можливим завдяки значному підвищенню продуктивності в сукупності з новими методологіями розподільних обчислень, що дозволило реалізувати відомі і нові загрози безпеці, що характеризуються зниженням фактичних витрат часу на подолання практичної стійкості підсистеми автентифікації до неприйнятного рівня. Два останніх десятиліття активно розроблялися математичні методи, які мають практичний інтерес для криптоаналізу. Обчислювальна складність вирішення певного класу математичних задач лежить в основі безпеки ряду систем захисту комерційних інформаційно-комунікаційних технологій.

У зв'язку з цим однією з актуальних наукових завдань є розробка перспективних криптопримітивів з доказовою стійкістю. Завданням розділу є аналіз безпеки і продуктивності сучасних криптопримітивів фіналістів конкурсу NIST (National Institute of Standard and Technology), аналіз причин вибору криптопримітива Keccak в якості фіналіста, сучасних вимог до безпеки, обчислювальної складності і швидкості, методів підвищення швидкості без збільшення складності реалізації .

Результати міжнародних проектів NESSIE (2000 - 2003 рр.) І NIST SHA-3 Competition (2007 - 2012 рр.) Визначають вимоги до побудови колізійно стійких функцій хешування і ключових функцій хешування для обчислення кодів автентифікації повідомлень, автентифікації і встановлення ключів в криптографічних протоколах, перш за все в протоколах електронного цифрового підпису [221,222]. Критеріями оцінювання кандидатів на стандарт SHA-3 є стійкість до атак на хеш-функцію, складність і швидкість обчислення, характеристики і реалізація алгоритму.

6.1.1 Визначення і класифікація MAC-кодів

Код автентифікації повідомлення (MAC-код) визначається як обчислена для блоку даних ключова криптографічна контрольна сума (автентифікатор), за допомогою якої можна перевірити автентичність повідомлення. Маємо таке визначення.

Визначення 6.1 [223] MAC-код є функцією відображення $h: K \times D \to R$, де простір ключів $K = \{0,1\}^k$, простір повідомлень $D = \{0,1\}^*$ і простір MAC значень $R = \{0,1\}^n$ для $k, n \geq 1$. Для заданих значень ключа $k \in K$ і повідомлення $X \in D$ функція виробляє MAC- обчислення $Y \in R$.

Зауваження 6.1

– функція обчислення MAC-кода повинна мати наступні властивості:

1) має бути обчислювально важкою, знаючи M і $CK(M)$, найти повідомлення M', таке, щоб $CK(M) = CK(M')$;

2) значення $CK(M)$ повинні бути рівномірно розподіленими, для будь-яких повідомлень M і M' ймовірність того, що $CK(M) = CK(M')$ повинна дорівнювати 2^{-n}, де n – довжина значення MAC;

– нехай довжина ключа, використовуваного при обчисленні MAC, дорівнює k. За умови сильної MAC-функції криптоаналітику потрібно виконати 2^k спроб для перебору всіх ключів. Якщо довжина значення,

створюваного MAC, дорівнює n, то всього існує 2^n різних значень MAC-кодів;

– припустимо, криптоаналітик має доступ до відкритого повідомленням і відповідного йому значення MAC. Визначимо зусилля, необхідні криптоаналітику для знаходження ключа MAC.

Нехай довжина ключа $k > n$ більша довжини MAC. за відомими M_1 і $MAC_1 = CK(M_1)$ криптоаналітик може обчислити $MAC_1 = CK_i(M_1)$ для всіх можливих ключів K_i. Принаймні, для одного з ключів буде отримано збіг $MAC_i = MAC_1$. Криптоаналітик вирахує 2^k значень MAC. При довжині MAC n біт існує всього 2^n значень MAC. Правильне значення MAC буде отримано для кількох значень ключів. В середньому збіг матиме місце для $2^k / 2^n = 2^{(k-n)}$ ключів. Для обчислення єдиного ключа опонентові буде потрібно знати кілька пар повідомлень і відповідних їм MAC-кодів. Таким чином, простий перебір всіх ключів вимагає більше зусиль, ніж пошук ключа симетричного шифрування тієї ж довжини.

Міжнародні стандарти побудови алгоритмів автентифікації даних представлені в таблиці 6.1.

Таблиця 6.1 – Стандарти побудови MAC-кодів

Стандарт	Механізм вироблення MAC-кода
FIPSPUB 113 Computer Data Authentication (2002)	Алгоритм на основі DES
FIPS PUB 198-1 The Keyed-Hash Message Authentication Code	Алгоритм HMAC
ISO/IEC 9797-1	Алгоритми на основі блочного шифра
ISO/IEC 9797-2 Mechanisms using a dedicated hash-function	Алгоритми на основі хеш-функції

Основні алгоритми обчислення MAC-кодів представлені в таблиці 6.2.

Зауваження 6.2 Можна виділити наступні підходи до побудови кодів автентифікації повідомлень [224, 225]:

– коди автентифікації повідомлень, побудовані із застосуванням блокових шифрів;

- коди автентифікації повідомлень, побудовані на основі безключових хеш-функцій;

- коди автентифікації повідомлень, побудовані з використанням сімейства універсальних хеш-функцій;

- коди автентифікації повідомлень, побудовані на основі спеціалізованих алгоритмів;

– *MAC*-коди на основі блочного шифра.

Таблиця 6.2 Алгоритми обчислення *MAC*-кодів

Алгоритм	Рекомендація		Схема реалізації
	поточна	в майбутньому	
EMAC	Так	Так	любий БСШ як *PRP*
CMAC	Так	Так	любий БСШ як *PRP*
HMAC	Так	Так	любий БСШ як *PRP*
UMAC	Так	Так	люба хеш-функція як *PRP*
GMAC	Так	Ні	операції в кінцевому полі
AMAC	Так	Ні	любий БСШ

Коди автентифікації повідомлень, побудовані із застосуванням блокових шифрів, визначаються стандартом ISO / IEC 9797-1 та використовують шифрування в режимі ЗВС зчеплення шифр текстів [226].

Визначення 6.2 *MAC*-код, заснований на застосуванні блочного шифру в ЗВС режимі, визначається виразом $H_1 = E_K(X_1)$, $H_i = E_K(X_i \otimes H_{i-1})$, $2 \le i \le t$. Безпека *ЗВС MAC*-конструкцій грунтується на криптографічній стійкості блокового шифру, і оцінки стійкості отримані для статистичної моделі блочного шифру як псевдовипадкової функції [226].

Довжина блоку даних повинна бути дорівнює довжині даних блочного шифру. довжина MAC-кода встановлюється рівною довжині блоку блочного шифру. На кожному кроці ітерації ключ MAC-кода використовується як ключ шифру, а блок повідомлення, після побітового складання з результатом обчислення шифру тексту, отриманий на попередньому кроці, подається на вхід блоку шифрування.

Основна CBC-конструкція вразлива до атаки типу exor підробки і, отже, може використовуватися тільки в додатках, де повідомлення мають фіксовану довжину. Також існують кілька більш захищених різновидів цього алгоритму: EMAC і RMAC схеми. EMAC код використовує додаткове шифрування вихідного результату перетворень, ключ для цієї операції шифрування може бути отриманий з ключа MAC. $RMAC$ алгоритм замінює останнім шифрування на шифрування з двома ключами. Безпека цих конструкцій може бути доведена за умови, що основний блоковий шифр є псевдовипадковим [226]. Всі ці схеми містяться в ISO / IEC 9797-1, що є міжнародним стандартом для кодів автентифікації повідомлень, що використовують блокові шифри [226].

Коди автентифікації повідомлень на основі безключових хеш-функцій (HMAC) використовують значення секретного ключа при обчисленні хеш-результату. Ці коди автентифікації повідомлень мають велику швидкість в порівнянні з кодами автентифікації повідомлень, використовують блокові шифри. HMAC є вкладеною конструкцією, обчислює MAC-код на основі хеш-функції, повідомлення і секретного ключа [227].

Визначення 6.3 HMAC код визначається виразом

$$HMAC(K, X) = h\left((K \otimes opad) \| h(K \otimes ipad \| X)\right).$$

Ключ K доповнюється нульовими бітами до повного блоку, opad та ipad - постійні значення. Безпека цієї конструкції була доведена в [222], грунтуючись на таких припущеннях: основна хеш-функція є колізійно стійкою за умови, що початкове значення - секретне; ключова функція

стиснення за допомогою початкового значення є захищеним *MAC*-алгоритмом (для повідомлень розміром в один блок); функція стиснення - слабка псевдовипадкова функція.

Альтернативою HMAC кодами є MDX-MAC конструкції [227], засновані на MD5, SHA та RIPEMD хеш-функціях. Тут основна хеш-функція змінена в *MAC*-код за рахунок внесення невеликих модифікацій: включення секретного ключа на початок, на кінець і до кожного кроку ітераційного обчислення хеш-функції. Захищеність MDX-MAC може бути доведена за умови, що основна функція стиснення є псевдовипадковою.

Практичними алгоритмами хешування в HMAC (ISO / IEC 9797-2) алгоритмі є: ГОСТ 34.311-95, HAVAL, SHA-1, RIPEMD-160, MD-5, ГОСТ 28147-89 режим 4 [228], Whirpool [229], SHA-2 [230].

Три алгоритму серії MD (MD-2, MD-4, MD-5) розроблені Роном Райвест в 1989, 1990 і 1991-му роках відповідно. Вони оперують з блоками даних, що збігаються з довжиною результуючого значення згортки, причому $n=128$ для алгоритма *MD*-4, і $n=160$ – для *MD*-5 і *SHA*. Зазначені алгоритми спроектовані спеціально з урахуванням ефективної реалізації на 32-розрядних процесорах.

Оригінал тексту M розбивається на блоки довжиною 512 біт. Останній блок формується шляхом дописування до кінця повідомлення комбінації до отримання блоку розміру 448 біт, до якого потім додається комбінація з 64 біт, що представляє бітову довжину повідомлення. Потім обчислюється значення згортки згідно з процедурою з використанням однокрокового стискання функції, заданої формулою $f(\chi, H)=E\chi(H)\oplus H$, де χ – блок повідомлення довжиною 512 біт, H – блок з η біт, а $E\chi$ – деяке перетворення множини блоків. Значення початкового вектора визначається в описі перетворення $E\chi$.

Алгоритм *MD*-2 (*RFC* 1319) передбачає:

- доповнення тексту до довжини, кратної 128 біт;

- обчислення 16-бітної контрольної суми (старші розряди відкидаються);

- додавання контрольної суми до тексту;

- повторне обчислення контрольної суми.

Алгоритм MD-4 передбачає:

- доповнення тексту до довжини, що дорівнює 448 біт по модулю 512;

- додається довжина тексту в 64-бітному поданні;

- 512-бітові блоки піддаються процедурі Damgard - Merkle, причому кожен блок бере участь в трьох різних циклах.

В алгоритмі MD-4 досить швидко були знайдені критичні уразливості, тому він був замінений алгоритмом MD5, в якому кожен блок бере участь не в трьох, а в чотирьох різних циклах.

Алгоритм MD-5 призначений для створення «відбитків» або «дайджестів» повідомлень довільної довжини. Є покращеною в плані безпеки версією MD-4. Знаючи MD-5, неможливо відновити вхідне повідомлення, так як одному MD-5 можуть відповідати різні повідомлення. Використовується для перевірки автентичності опублікованих повідомлень шляхом порівняння дайджесту повідомлення з опублікованими. Цю операцію називають «перевірка хешу» (hashcheck). На вхід алгоритму надходить вхідний потік даних, хеш якого необхідно знайти. Довжина повідомлення може бути будь-якою (в тому числі нульовою). Запишемо довжину повідомлення в L. Це число ціле і невід'ємне. Кратність будь-яким числам необов'язкова. Після надходження даних йде процес підготовки потоку до обчислень.

Алгоритм SHA розроблений NIST і повторює ідеї серії MD. У SHA використовуються тексти більш 264 біт, які закриваються сигнатурою довжиною 160 біт. Даний алгоритм передбачається використовувати в програмі Capstone. RIPEMD сімейство функцій хешування використовує дві паралельні схеми обчислення, які є модифікованими версіями MD-4. У таблиці 6.3 представлені порівняльні дані MD подібних алгоритмів [227].

Функція хешування SHA-1 входить в стандарт FIPS 180-1 і рекомендований NIST для цифрових підписів разом з DSA стандартом. NIST оновив цей стандарт, представивши FIPS 180-2 [231], який включає, крім

SHA-1, три нові хеш-функції SHA-2/256, SHA-2/384 і SHA-2/512 з функціями хешування більшої довжини, щоб відповідати рівню захисту нового стандарту блокового шифру AES. ANSI прийняв банківські стандарти криптографії з відкритими ключами: стандарт X9.30 [232], який визначає алгоритм SHA-1 разом з DSA, і стандарт X9.31 [233], який визначає алгоритм MDC-2 з цифровим підписом на базі RSA.

Алгоритм Whirlpool був визнаний кращим в проекті NESSIE в категорії «Стійка до колізій хеш-функція». Алгоритм Whirlpool обчислює 512-бітний хеш-код з використанням в якості опції стиснення блочного шифру, який є модифікацією алгоритму Rijndael. В роботі [229] досліджувався існування квадратичної залежності у вхідних і вихідних значеннях блоку підстановки алгоритму Whirlpool.

Було показано, що для блоків підстановки Rijndael і Serpent існують квадратичні рівняння для вхідних і вихідних біт з імовірністю 1. Такі рівняння завжди існують для n біт n-бітного блоку підстановки, якщо $n \le 6$, але для $n > 6$ – не завжди. Були проведені дослідження з виявлення квадратичних залежностей в блоці перестановки Whirlpool.

Таблиця 6.3 – Параметри *MD* подібних функцій хешування

Алгоритм	Розмір хеш-кода (біт)	Розмір блока (біт)	Розмір слова (біт)	Число циклів на число кроків в циклі
MD-4	128	512	32	3/16
MD-5	128	512	32	4/16
RIPEMD-128	128	512	32	4/16/2
RIPEMD-160	160	512	32	5/16/2
SHA-1	160	512	32	4/20
SHA-2/256	256	512	32	1/64
SHA-2/384	384	1024	64	1/80
SHA-2/512	512	1024	64	1/80

Блок перестановки - це 8-бітна перестановка. Існує максимум 137 можливих ступенів свободи в багатовимірному вираженні для 8 вхідних і 8 вихідних біт. Найпростіший метод перевірки наявності таких залежностей - це обчислення 256 раз визначника 137-мірних довічних матриць. Для повної таблиці підстановки Whirlpool не було знайдено квадратичних залежностей. Проте, так як блок підстановки складається з декількох чотирьох бітових підстановок, завдання побудови маленької системи багатовимірних квадратичних рівнянь істотно спрощується. Алгоритм Whirlpool має дуже високу стійкість, порівняно з SHA-2 (512), але невисока швидкодія і, відповідно, його можна рекомендувати до застосування в системах, де необхідно забезпечити стійкість протягом тривалого періоду часу, де критерій стійкості є визначальним і набагато важливіше швидкості. Математична простота алгоритму, досягнута в процесі розробки, дозволяє спростити і процес аналізу стійкості. Довжина *MAC*-кода в 512 біт забезпечує ефективний захист від атак, заснованих на парадоксі «день народження», а також покращує показники стійкості до колізій. Оптимальна довжина ключових даних, що відповідає сучасним вимогам, дозволила цьому алгоритму стати переможцем в категорії «Алгоритм, стійкий до колізій» в рамках проекту NESSIE.

ISO / IEC розвинув стандарт 10118 для різних класів функцій хешування [224, 230, 234, 235]. У частині 10118-2 [234] визначені функції хешування, засновані на блокових шифри в конструкції Matyas-Meyer-Oseas, коли незалежний блоковий шифр в алгоритмі MDC-2 з двома і більше функціями робить значення хешу подвійною і потрійною довжини відповідно. Частина 10118-3 [230] визначає три алгоритму: RIPEMD-128, RIPEMD-160 і SHA-1. Ця частина стандарту в даний час переглядається з урахуванням оцінки нових криптографічних примітивів, які будуть прийняті як стандарти ISO. Крім зазначених трьох алгоритмів вивчаються функції хешування: SHA-2/256, SHA-2/384, SHA-2/512 і Whirlpool. Частина 10118-4 [235] описує MASH-1 і MASH-2 функції хешування, які використовують

модулярну арифметику. Оцінки стійкості функцій хешування представлені в таблиці 6.4 [236].

Коди автентифікації повідомлень, які засновані на універсальному хешуванні, використовують комбінаторні властивості сімейства хеш-функцій.

Визначення 6.4 MAC-код на основі універсального хешування є відображенням $h = H_k : D \to R$ (де для кожного $k \in K$, H_k – функція з сімейства хеш-функцій $H = \{h : D \to R\}$, D – загальна область визначення, R – кінцевий діапазон значень, таким, що для будь-яких різних $x, x' \in D$ ймовірність того, що $h(x) = h(x')$ буде не більше $\varepsilon \le 1$, при випадковому виборі $h \in H$ [237].

Комбінаторні властивості універсального хеш-сімейства дозволяють отримати точні межі секретності *MAC*-кодів.

Таблиця 6.4 – Верхні межі стійкості функцій хешування

Функція хешування	n	m	Стійкість прообразу	Стійкість до колізії
Матіс – Мейер – Озіса[a]	n	n	2^n	$2^{n/2}$
MDC-2 (*DES*)[b]	64	128	$2 \cdot 2^{82}$	$2 \cdot 2^{54}$
MDC-4 (*DES*)	64	128	2^{109}	$4 \cdot 2^{54}$
Мерклі (*DES*)	106	128	2^{112}	2^{56}
MD-4	512	128	2^{128}	2^{20}
MD-5	512	128	2^{128}	2^{64}
RIPEMD-128	512	128	2^{128}	2^{64}
SHA-1, *RIPEMD*-160	512	160	2^{160}	2^{80}

a – ця ж стійкість передбачається для функцій хешування Девіса - Мейєра і Міягучі - Прінеля;

b – стійкість може бути збільшена шляхом застосування шифру з довжиною ключа, що дорівнює довжині блоку шифру.

Коди автентифікації повідомлення, побудовані на основі спеціалізованих алгоритмів, є модифікацією відомих хеш-функцій і мають, як правило, найвищий рівень захисту для *MAC*-примітивів.

Прикладом таких алгоритмів є Two-Track-MAC (K.U.Leuven, Бельгія і debisAG, Німеччина). TTMAC (Two-Track-MAC) алгоритм заснований на хеш-функції RIPEMD-160 з невеликими модифікаціями [238]. Алгоритм працює на блоках 512 біт, розділених на слова по 32 біт, використовує секретний ключ 160 біт і виробляє вихід до 160 біт. Великий розмір внутрішнього стану (320 біт) в Two-Track-MAC дає алгоритму високий рівень захисту від атак, заснованих на внутрішніх колізіях [238]. Складність основних атак на цей примітив наступна:

– приблизно 2^{159} обчислень *MAC*-кода і $160/m$ відомих пар текста – *MAC* необхідні для вичерпного пошуку ключа, де m – довжина *MAC*-результата (значення для *m* підтримується алгоритмом між 32 і 160 бітами);

– вгадування значення *MAC*-кода має ймовірність успіху 2^{-m};

– атаки, засновані на внутрішніх колізіях, вимагають приблизно 2^{160} відомих пар текст – *MAC*-код і приблизно 2^{320-m} вибраних текстів.

Алгоритм TTMAC має найвищий рівень захисту для *MAC*-примітивів, певні переваги у швидкодії, особливо в разі коротких повідомлень, і оптимальну довжину ключових даних. Разом з тим, TTMAC має низьку швидкість, що робить проблематичним застосування для додатків, де потрібно хешувати дані великих обсягів. У таблиці 6.5 представлені основні результати за параметрами і оцінці швидкодії основних алгоритмів автентифікації. Швидкість обчислень визначається кількістю циклів процесора, що витрачаються на один байт оброблюваного повідомлення.

Таблиця 6.5 – Швидкодія *MAC*-алгоритмів

Алгоритм	Довжина *MAC*-кода (біт)	Довжина ключа (біт)	Тип ПЕВМ, кількість циклів				
			Pentium 2	*P*III/*Linux*	*Pentium* 4	*Xeon*	*AMD*
TTmac	160	160	21	21	40	37	21
UMAC-16	64	128	6.1	6.0	6.2	6.1	6.2
UMAC-32	64	128	2.5	2.9	6.7	6.6	1.9
HMAC-Whirlpool	512	512	86	72	98	103	100
HMAC-MD-4	128	512	4.7	4.7	6.4	6.4	4.7
HMAC-MD-5	128	512	7.2	7.3	9.4	9.4	7.4
HMAC-RIPE-MD	160	512	23	18	27	26	21
HMAC-SHA-0	160	512	16	15	23	23	13
HMAC-SHA-1	160	512	16	15	25	24	12
HMAC-SHA-2	256 384 512	512	40 84 84	39 84 84	40 124 124	39 132 132	33 72 72
HMAC-Tiger	192	512	24	21	28	26	20
CBCMAC-Rijndael	128	128	24	26	26	27	31
CBCMAC-DES	64	56	62	61	72	69	54
CBCMAC-Shacal	512	160	31	31	67	74	29

6.1.2 Вплив обчислювальної потужності на вимоги до хеш-функцій

В якості основних складових приросту потужності обчислювальних систем можна виділити: зростання продуктивності виділених обчислювальних пристроїв; організацію масових обчислень в сукупності пристроїв, в тому числі і мобільних; використання ефективних моделей обчислень на існуючих класах архітектур. Одним з найбільш перспективних напрямків у вирішенні завдань обчислень загального призначення є використання технології GPGPU (General-purpose graphics processing units) [239]. Графічний процесор (GPU) володіє меншим набором виконуваних команд (RISC-подібні архітектури), ніж CPU, але більшою продуктивністю. Технологія GPGPU дозволяє на одному обчислювачі досягати досить

високого рівня паралелізму без тимчасових витрат на передачу даних між вузлами і синхронізацію результатів обчислень. Відносна низька вартість, простота додавання обчислювальних модулів і питоме енергоспоживання в поєднанні з високою питомою продуктивністю GPU дозволяють реалізувати на практиці масові розподілені паралельні обчислення, які доступні ширшому колу потенційних порушників. Порівняльний аналіз можливостей CPU і GPU архітектур в практичній реалізації алгоритмів SHA-1 і MD-5 представлений на рис. 6.1. [239].

Слабка обчислювальна складність, ряд значних недоліків конструкції і математично обгрунтовані атаки на колізійну стійкість дозволили реалізувати для хеш-функцій SHA1 і MD5 атаки повного перебору за допустимий час. У практичній реалізації [239] представлені результати швидкості підбору MD5-паролів на nVidiaTitanX (близько 135,2 мільярдів комбінацій в секунду), що дозволяє знайти пароль довжиною вісім символів за менш ніж п'ять хвилин.

У таблиці 6.6 представлені результати продуктивності реалізації атаки «повного перебору» для сімейства хеш-функцій на тестовому стенді (Ubuntu 14.04, 64 bit, ForceWare 346.29, 8xNVidiaTitanX, stockcoreclock, oclHashcatv 1.3).

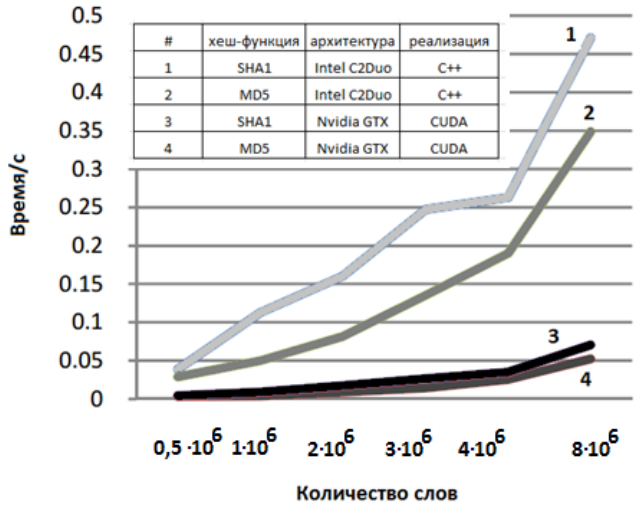


Рисунок 6.1 – Порівняльний аналіз швидкості реалізації хеш-функцій на CPU / GPU архітектурах

Таблиця 6.6 – Продуктивність атак повного перебору для різних хеш-функцій

Алгоритм	Продуктивність (комбінацій / с)
MD-5	$135232*10^6$
SHA-1	$42408*10^6$
SHA-256	$16904*10^6$
SHA-512	$5240*10^6$
RipeMD-160	$28368*10^6$
Whirpool	$1122402*10^6$

Успішні атаки і зростання обчислювальної ємності висувають високі вимоги до стійкості хеш-функцій. Недоліки, виявлені у криптографічних стандартах 90-х, і обгрунтовані теоретичні атаки обумовлюють розвиток нових методів реалізації криптопримітивів. Конкурс SHA-3, організований NIST, показав розвиток вимог до безпеки, архітектурі і продуктивності при реалізації криптографічних примітивів. У фіналі конкурсу SHA-3, організованого NIST, двоє з п'яти фіналістів (Keccak і Skein) виявилися універсальними криптопримітивами, які можуть використовуватися не тільки для хешування, але і для виконання множини інших криптографічних операцій, забезпечуючи спрощення проектованих криптографічних протоколів. Три етапи конкурсу фактично формалізували подальший підхід до процедури вибору криптопримітивів. Розглянемо вимоги до безпеки, архітектурі і продуктивності, що пред'являються до сучасних криптопримітивів.

6.1.3 Вимоги до безпеки сучасних хеш-функцій

Основні визначення безпеки хеш-функцій представлені в [226,227, 229, 240,241].

Визначення 6.5 (Стійкість до обчислення прообразу). Хеш-функція $h: \{0,1\}^* \to R$ є стійкою до обчислення прообразу силою (t, ε), якщо не існує імовірнісного алгоритму Ih, з вхідними значеннями $Y \in_R R$ і значеннями на

виході $X \in \{0,1\}^*$, часом виконання не більше ніж t, де $h(X) = Y$, і ймовірністю не менше ε, оціненої при випадковому виборі Y і Ih.

Стійкість хеш-функцій до обчислення прообразу має важливе значення для систем автентифікації, що використовують хеш-значення паролів і секретних ключів.

Визначення 6.6 (Стійкість до обчислення другого прообразу). Нехай S – кінцева підмножина $\{0,1\}^*$. Хеш-функція $h: \{0,1\}^* \to R$ є стійкою до обчислення другого прообразу силою (t, ε, S), якщо не існує імовірнісного алгоритму Sh, з $X \in_R S$ і $X' \in \{0,1\}^*$, часом виконання не більше ніж t, де $X' \neq X$ і $h(X') = h(X)$, і ймовірністю не менше ε, оціненої при випадковому виборі X і Sh.

Стійкість хеш-функцій до обчислення другого прообразу визначає безпеку систем автентифікації з цифровим підписом.

Визначення 6.7 (Стійкість до колізій). Хеш-функція $h: \{0,1\}^* \to R$ є стійкою до колізій силою (t, ε), якщо не існує імовірнісного алгоритму Ch с відомими вихідними значеннями $X, X' \in \{0,1\}^*$, часом виконання не більше ніж t, де $X' \neq X$ і $h(X) = h(X')$, і ймовірністю не менше ε, оціненої при випадковому виборі Ch.

Основні вимоги до безпеки хеш-функцій, які використовуються NIST, представлені в роботі [242]. Більше 40 кандидатів не задовольнили цих визначень і були виключені в першому раунді конкурсу [222].

Порівняльний аналіз фіналістів конкурсу NIST, які відповідають основним вимогам до безпеки, наведено в таблиці 6.7. [243].

Таблиця 6.7 - Безпека фіналістів конкурсу NIST

Кандидат	Стійкість до колізій	Кількість раундів стиснення	складність обчислення		
			про-образа	другого прообраза	Псевдо pre
Blake-256	Inner-collision	2.5	2^{224}	2^{256}	-
Blake-512	Inner-collision	4	2^{448}	2^{256}	-
Groestl-256	2^{64}	5	2^{256}	$2^{256\text{-}512}$	$2^{244.85}$
Groestl-512	2^{128}	8	2^{512}	$2^{512\text{-}1024}$	2^{248}
JH-256	$2^{96,12}$	16	-	2^{-388}	-
JH-512	$2^{95,63}$	22	-	2^{-900}	-
Keccak-224	близько до 2^{256}	4-5	2^{112}	2^{288}	
Keccak-256	близько до 2^{256}	5-10	2^{1370}	2^{512}	
Keccak-512	2^{512}	24	2^{1590}	$2^{511.5}$	2^{1576}
Skein-256	$>2^{-265}$	32-36	2^{105}	2^{200}-2^{824}	$2^{511.7}$
Skein-512	$>2^{-265}$	32-36	2^{105}	2^{200}-2^{824}	$2^{511.7}$
Skein-1024	$>2^{-265}$	32-36	2^{105}	2^{200}-2^{824}	2^{1045}-2^{125}

6.1.4 Вимоги до архітектури і реалізації MAC-алгоритмів

Узагальнимо вимоги до архітектури і особливості реалізації, які є частиною інтегральної оцінки алгоритму. Функція стиснення, як основний елемент архітектури хеш-функції була заявлена в більшості кандидатів. З точки зору вимог до архітектури важливими атрибутами є: особливості реалізації структури SPN і блоків підстановок (S-box) або блоків перестановок (P-box), схеми Фейстеля для перетворень функцій $F(L_i, K_i)$, математична складність функції ключового розширення (функції розгортання підключів раунду з основного ключа), структури Merkle-Damgard, WidePipe, розмірність MDS-матриці. В роботі [244] розглянуто вплив булевих операцій, OUT-трансформації, FSR, ARX-зрушень на кінцеву реалізацію хеш-функцій.

З точки зору оптимізації обчислень, найбільш важливим параметром є розмір кешу 1-го рівня для інструкцій. Кеш 1-го рівня для даних важливий для функцій, які використовують табличну реалізацію (такі, як хеш-функція ECHO). Невідповідність кількості виконуваних для циклу інструкцій

передбачає серйозне зниження продуктивності (прикладом є деякі реалізації Skein), де згодом передбачається використовувати метод «розгортання» [245,246]. Цей метод полягає в повторенні коду для одного раунду декількох послідовних раундів і дозволяє «обнулити» витрати на маршрутизацію даних в пам'яті. Так, наприклад, в алгоритмі SHA-256 в кожному раунді «обертаються» вісім слів, що для восьми послідовних раундів «розгортання» дозволяє використовувати ті ж змінні і скоротити витрати на копіювання даних між ними. Крім того, якщо «розмотувати» 64 послідовних раунду SHA-256, то можна відмовитися від отримання констант з таблиці і заощадити на непрямої адресації в пам'яті. Такий підхід є загальним інструментом зменшення витрат на маршрутизацію даних під час виконання. Якщо реалізація в процесі повного «розгортання» поміщається в кеш 1-го рівня, то вона оптимізована, в зворотному випадку виникають додаткові витрати на копіювання та непряму адресацію. Важливою особливістю реалізації є можливість використання 64-розрядних цілих чисел для систем з власними 64-розрядними регістрами [246]. На 32-бітних системах без таких регістрів використання 64-бітових цілих неефективно і потребує два регістра. Це збільшує витрати на коди операцій (перенесення між нижніми і верхніми словами) для 32-бітних архітектур. Найбільш очевидно це проявляється на діаграмах з отриманим на 64-бітних архітектур 512-бітовим хеш-кодом. В архітектурі x.86 відсутність 64-розрядної цілого типу часто компенсується в реалізаціях хеш-функцій, з використанням спеціальних блоків з 64-розрядними регістрами (MMX, SSE). Недолік реалізацій в подібних випадках полягає в необхідності використовувати особливі, вбудовані в компілятор інструкції (C / C ++) [246]. Це обмежує можливості реалізації на інших платформах, зокрема в JavaVM. Порядок байт для сучасних алгоритмів хеш-функцій вже не має істотного значення для досягнення високої продуктивності.

Для більшості функцій поточний набір інструкцій також не важливий.

У великих системах коди операцій динамічно транслюються у внутрішні елементарні інструкції, для яких CPU застосовує оптимізації (паралельне виконання, зміна порядку, спекулятивне виконання інструкцій і т. д.).

Більшість з кандидатів NIST другого раунду змагань показали композитну схему реалізації. Деякі з них були представлені парою функцій для різної довжини виходів (224, 256, 384, 512 біт). Fugue і Lufa складалися з трьох функцій, а Keccak - з чотирьох, хоча і з розділяючим ядром.

У свою чергу, такий підхід в архітектурі має ряд негативних наслідків для продуктивності: реалізація сімейства функцій вимагає більше ресурсів для розробки, ніж для оптимізації; збільшується розмір коду; проблеми продуктивності та безпеки більш не є взаємозалежними. Архітектура CubeHash, JH і Shabal дозволила уникнути подібного ефекту і показала стабільну продуктивність для всіх розмірів вихідних даних.

Мінімальний розмір ключа, необхідний для захисту інформації від атак зловмисника, буде рости в міру підвищення швидкодії комп'ютерів, але, тим не менш, наведені обчислення показують, що можна вибрати таку довжину ключа, при якій атаку методом повного перебору провести буде в принципі неможливо, не залежно від підвищення обчислювальної потужності комп'ютерів або успіхів в області класичної теорії алгоритмів.

В роботі [242] проаналізовано підхід до класифікації сучасних хеш-функцій на основі порівняння з еталонною реалізацією алгоритму SHA-256/512 в NIST (IntelCore 2 Duo). За результатами конкурсу клас швидкості в роботі був визначений наступним чином:

$AA = x < 1/2\,SHA\text{-}2$;

$A = 1/2\,SHA\text{-}2 \le x < 3/4\,SHA\text{-}2$;

$B = 3/4\,SHA\text{-}2 \le x < SHA\text{-}2$;

$C = SHA\text{-}2 \le x < 5/4\,SHA\text{-}2$;

$D = 5/4\,SHA\text{-}2 \le x < 2SHA\text{-}2$;

$E = x > 2SHA\text{-}2$.

Таблиця 6.8 – Порівняльний аналіз швидкості хеш-функцій

Хеш-функція	Продуктивність 32-біт		Продуктивність 64-біт	
	cpb	клас швидкості	*cpb*	клас швидкості
SHA-256	29.3	*C*	20.1	*C*
SHA-512	55.2	*C*	13.1	*C*
Blake-256	28.3	*B*	16.7	*B*
Blake-512	61.7	*C*	12.3	*B*
Groestl-256	22.9	*B*	22.4	*D*
Groestl-512	37.5	*A*	30.1	*E*
JH-256	21.3	*B*	16.8	*B*
JH-512	21.3	*AA*	16.8	*D*
Keccak-256	35.4	*C*	10.1	*A*
Keccak-512	68.9	*C*	20.3	*D*
Skein-256	21.6	*A*	7.6	*AA*
Skein-512	20.1	*AA*	6.1	*AA*

Проаналізуємо особливості архітектури і реалізації фіналістів NIST, їх вплив на обчислювальну складність, швидкість і безпеку.

6.1.5 Аналіз схем реалізації фіналістів NIST

6.1.5.1 Алгоритм BLAKE

Представлений в якості альтернативи існуючим SHA-2 / MD-5 застосуванням, значно перевершуючи їх по надійності, практично не поступаючись в продуктивності [247] (рис. 6.2).

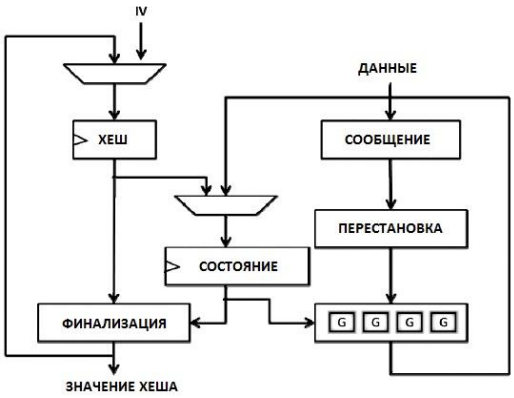


Рисунок 6.2 – Високорівнева архітектура BLAKE

Хеш-функція BLAKE не чутлива до розміру хешируємих даних і захищена від усіх властивих SHA-1 і MD5 видів атак, пов'язаних з виникненням колізій в процесі хешування [246]. Реалізації для різних архітектур з підтримкою розпаралелювання дозволяють істотно збільшити продуктивність.

6.1.5.2 Алгоритм Groestl

Функція хешування Groestl здатна повертати хеш-значення довільної довжини від 1 до 64 байт, тобто від 8 до 512 біт, при цьому хеш-значення має бути кратне байту (див. рис. 6.3). Функція стиснення базується на двох t-бітових перестановках P і Q (1) [249]:

$$F(h,m)=P(h\oplus m)\oplus Q(m)\oplus h. \qquad (6.1)$$

Вихідне хешування можна описати за допомогою формули $\Omega(h)=\text{truncn}(P(x)$

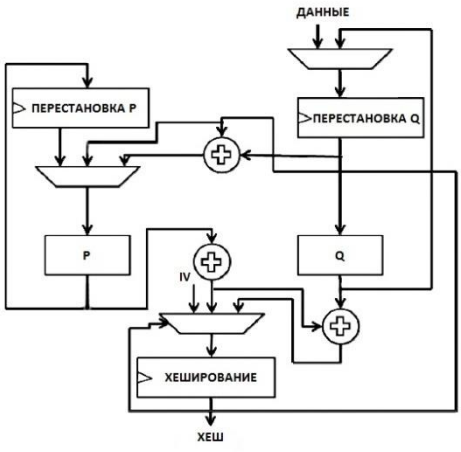


Рисунок 6.3 – Високорівнева архітектура Groestl

6.1.5.3 Алгоритм JH

Алгоритм JH використовує функцію стиснення. Повідомлення розбивається на блоки по 512 біт (хеш-значення може мати розмір 224, 256, 384 і 512 біт). Функція стиснення формує 1024-бітове значення, яке урізається до необхідного розміру хеш-значення. На вхід функції стиснення

надходять 512-бітові блоки повідомлення, а на етапі фіналізації - 1024-бітове вихідне значення після обробки попереднього блоку H_{i-1} [250]. Для обробки першого блока використовується значення вектора ініціалізації IV (рис. 6.4).

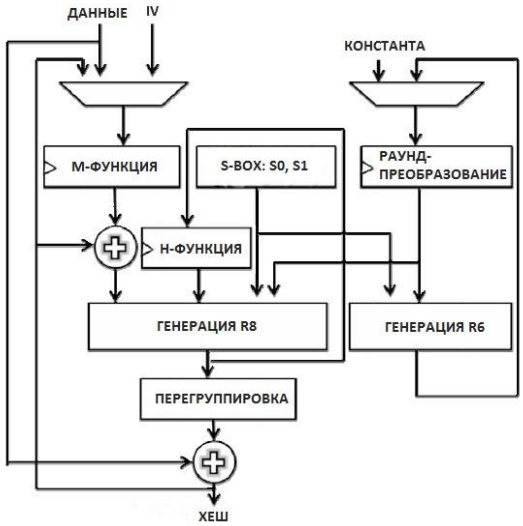


Рисунок 6.4 – Високорівнева архітектура *JH*

Функція стиснення виконує наступні дії. Опрацьований блок повідомлення M_i складається по модулю два з лівої 512-бітної половиною значення H_{i-1}. Результат попередньої операції обробляється функцією перетворення E. Повідомлення M_i складається по модулю два з правою половиною 1024-бітного вихідного значення функції E. В результаті виходить значення H_i. Вихідне 1024-бітове значення функції E представляється у вигляді восьмимірного масиву, кожний вимір якого містить два слова по чотири біта. В основі E функції лежить блоковий шифр, який представляє собою SPN перетворення на основі підстановок і перестановок і складається з 35 раундів. У кожному раунді використовуються: заміна за допомогою двох S-блоків $S0$ або $S1$; лінійне перетворення, по черзі обробляє по два слова стану за допомогою операцій XOR над певними бітами вхідних слів; перестановка слів стану.

6.1.5.4 Алгоритм Skein

Функція хешування Skein завдяки використанню нового класу блокових шифрів TBC (Tweakable Block Ciphers) в сукупності з унікальною блоковою ітерацією (UBI) і можливістю використовувати вибіркову систему параметрів має широкий набір властивостей. Даний підхід дозволив реалізувати множину режимів роботи, таких як: проста хеш-функція, функція деревовидного хешування, *MAC*-код, в якості складової частини HMAC. Skein підтримує рандомізоване хешування, використання в цифрових підписах, в якості опції обчислення похідних ключів (KDF), ключа з пароля (PBKDF), в якості генератора псевдовипадкових чисел (PRNG), в якості потокового шифру [251]. Використання настроюваного блочного алгоритму шифрування з UBI режимом гарантує, що кожен блок буде оброблений с використанням унікальної функції стиснення. На відміну від псевдовипадкових функцій (Pseudo Random Function - PRF) в конструкції функцій стиснення псевдовипадкові перестановки (Pseudo Random Permutation - PRP), які використовуються в блокових шифрах, дозволяють отримати більш швидку реалізацію (досить використовувати шифр з розміром блоку і розміром ключа 512 біт і можна буде отримати функцію стиснення $m \to n\,(m > n)$). Часто конструкція блочного шифру обумовлює використання полегшеного або слабкого ключового розширення. Потенційно це дозволяє реалізувати атаки зі зв'язаними ключами, що послаблює надійність функції стиснення на основі блочного шифру [252]. Ідеальна функція «розгортання ключа» для ідеального блочного шифру повинна мати властивості ідеальної псевдовипадкової функції.

Skein захищена від нових видів специфічних атак на хеш-функції - підбір подовжених повідомлень, псевдоколізії. Замість вибору різних схем і стандартів, вивчення особливостей їх застосування, роботи і реалізації розробникам криптододатків можна використовувати Skein і Keccak з різними параметрами. Традиційне побудова хеш-функцій засновано на використанні

функції стиснення. Ця функція відображає значення $m \to n\,(m > n)$ псевдовипадковим чином. При цьому значення n має бути до 512 біт, а m – порядка $2n$. Повторюючи цю функцію протягом кількох раундів з різними константами, досягають потрібного значення стійкості. Доповнюючи і зчіплюючи між собою блоки від різних фрагментів вихідного тексту, отримують можливість обчислити хеш від повідомлення довільної довжини. Такий метод є алгоритмічно складним. Багатораундові повторення згладжують дефектність, але наявність швидкої можливості знайти часткову колізію в вихідній функції стиснення не гарантує стійкість всієї конструкції (рис. 6.5).

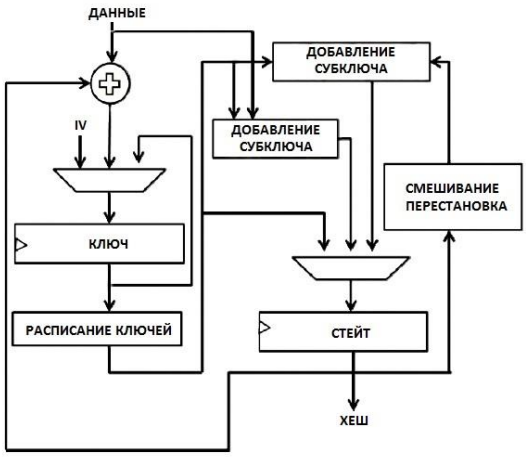


Рисунок 6.5 – Високорівнева архітектура Skein

6.1.5.5 Алгоритм Keccak

Автори алгоритму Keccak («Кеччак») стверджують, що сконструювати надійний функцію стиснення виду $m \to n\,(m > n)$ як однораундовий блок криптопримітива не представляється можливим. У Keccak в якості стійкого криптоперетворення замість функції стиснення була реалізована безключова PRF [253]. Архітектурою всього алгоритму є конструкція Sponge (англ. Sponge construction), що відноситься до класу алгоритмів з кінцевим внутрішнім станом, на вхід якої надходить двійковий рядок довільної довжини, і яка повертає двійковий рядок також довільної довжини $f : \{0,1\} \wedge n \to \{0,1\}^{\wedge *}$ [253]. Губка є узагальненням хеш-функцій, потокових і

блокових шифрів, генераторів псевдовипадкових чисел, що мають довільну довжину вхідних даних. Просте додавання секретного ключа на вхід хеш-функції Keccak перетворює її в код автентифікації повідомлень. Це було неможливо в звичайних хеш-функціях SHA-1 або SHA-2 і вимагало громіздкої конструкції HMAC. Розглянемо архітектуру алгоритму Keccak.

Ініціалізація. Масив з 5×5 рядків, що вказують в напрямку вісь z. В офіційно представленої версії число двійкових елементів в рядку z визначено для обчислень на 64-розрядних процесорах як $w=64$ [253]. Таким чином, стан містить $b=5\times5\times64=1600$ біт. Рядок S з $5\times5\times w$ біт порівняний з b бітами стану a наступним чином [253]:

$$s\left[w\left(5\times y+x\right)+z\right]=a\left[x\right]\left[y\right]\left[z\right]. \qquad (6.3)$$

На фазі ініціалізації блок даних розміру b заповнюється нулями, а вхідні дані M розбиваються на блоки розміру r. Вихідне повідомлення M доповнюється до розміру блоку, рівного частини блоку $b=r+c$.

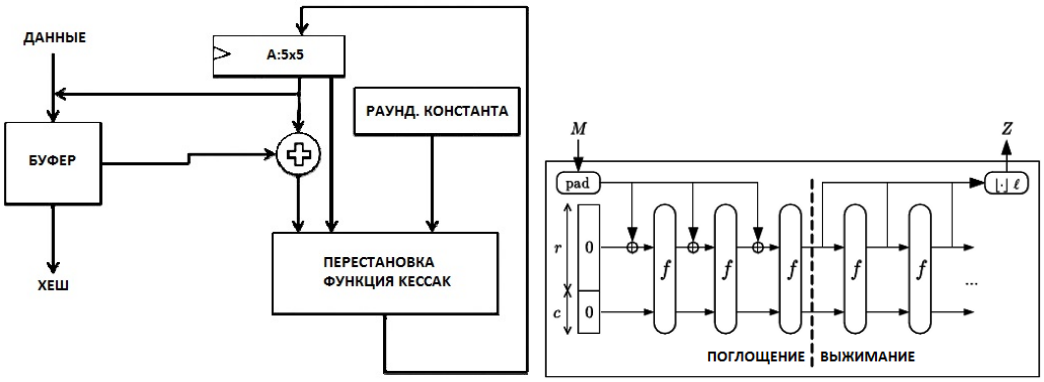
Суміжні b біти в стані можуть бути розділені між «зовнішньою частиною» (перші r біт) і «внутрішньою частиною» (змінне значення c біт). Значення c вибирається як $2N$, де N – розмір вихідних даних, згенерованих алгоритмом.

Фаза «поглинання». У фазі «поглинання» виконується операція XOR чергового блоку вихідного повідомлення з першою частиною стану розмірністю r біт, решта стану, розмірністю c біт, залишається незачепленою. Результат операції і незаймана частина передаються на вхід функції Keccak-f - багатораундові безключові псевдовипадкові перестановки і повторюється до вичерпання блоків вихідного повідомлення. Розглянемо конструкцію багатораундової перестановки:

Хеш-функція Keccak-f. Функція f в алгоритмі Keccak, представлена на рис. 6.6, б, задовольняє вимогам безпеки до хеш-функції. Кількість раундів збільшується на розмір w відповідно до формули $p=12+2*l$, де

$l = \log 2(w)$. При $w = 64$ кількість $R(p) = 24$. Кожен раунд складається з п'яти перестановок стану:

$$R = \theta \circ \rho \circ \pi \circ \chi \circ \iota . \quad (6.4)$$



а *б*

Рисунок 6.6 – Високорівнева архітектура Keccak – *а*;
конструкція «губка» – *б*

Перестановка θ. Перша перестановка конструкції «губка» забезпечує дифузію. Значення кожного біта в матриці стану розраховується як XOR між сумами (над полем GF (2)) двох інших рядів і власного значення біта:

$$a[x][y][z] = a[x][y][z] + \sum_{y'=0}^{4} a[x-1][y'][z] + \sum_{y'=0}^{4} a[x][y'][z-1]. \quad (6.5)$$

Перестановка ρ. Перестановка виконує обертання рядків в стані за допомогою визначеної константи. Для кожного біта в стані в раунді *n* маємо:

$$a[x][y][z] = a[x][y][z + T(n)]. \quad (6.6)$$

Перестановка π. Перестановка при $x = y'$ і $y = 2x' + 3y'$ впорядковує рядки наступним чином:

$$a[x][y][z] = a[x'][y'][z]. \quad (6.7)$$

Перестановка χ. Перестановка являється єдиною нелінійною функцією. Без неї функція Keccak-f була б лінійним відображенням над GF(2) [245]. Перестановка χ застосовується до кожного рядка стану, що практично реалізує відому конструкцію S-boxes для кожного рядка стану:

$$a[x][y][z] = a[x][y][z] + \left(a[x][y][z+1] \wedge a[x][y][z+2]\right). \qquad (6.8)$$

Перестановка ι. Перестановка призначена для внесення асиметрії в конструкцію за рахунок додавання раундовий констант, що нівелює властивості інваріанта і дозволяє уникнути трансляції, тим самим попереджаючи слайд-атаки, що використовують симетрію [253].

Фаза «вичавлювання». У цій фазі стан S подається на функцію f, після чого частина $S1$ подається на вихід. Ці дії повторюються, поки не буде отримана послідовність потрібної довжини (довжин хешу). Останні біти залежать від вхідних блоків лише опосередковано і не виводяться в ході фази «вичавлювання».

Атаки на знаходження колізій і других прообразів мають важливе практичне і теоретичне значення, але поряд з неінвертіруємостю не забезпечують повної оцінки стійкості хеш-функції. Існує клас атак, пов'язаних з практичними і теоретичними уразливими конструкцій хеш-функцій: атаки на подовження повідомлення, атаки на часткові колізії з підібраним префіксом і ін. Для доказовою стійкості функції «губка» авторами Keccak був запропонований критерій «випадкового оракула» (RandomOracle) - ідеалізованої функції, яка описує роботу ідеального автомата з практично нескінченним об'ємом пам'яті, який на будь-який запит видає ідеально випадкове число і запам'ятовує пару «запит-відповідь». При повторі запиту відповідь не генерується, а видається з раніше згенерованого масиву. Якщо функція Keccak-f з п'ятьма багатораундовими перестановками ідеальна, то хеш-функція доказово нерозрізнена з RO. Нерозрізненість хеш-функції з випадковим оракулом вважається єдиним і достатнім критерієм стійкості. Дане твердження дозволило обґрунтувати використання хеш-функції Keccak як практично універсального криптопримітива.

Якщо перед блоками повідомлення ввести блок з секретним ключем K, то вийде код автентифікації повідомлення. Практичний інтерес представляє можливість обчислення MAC-кода в паралельному режимі.

Раніше вважалося, що такі паралельні режими можливі тільки для блокових шифрів (OCB, GCM) або при використанні громіздкого режиму деревовидного хешування.

В роботі [254] запропонований спосіб паралельного обчислення *MAC*-кода на основі конструкції «губка» (рис. 6.7). Ключ об'єднується з вектором ініціалізації (Nonce) і подається на вхід множини паралельних конструкцій, де попередньо об'єднується ще й зі значенням лічильника кожної конструкції. Можливість використовувати Keccak в паралельному режимі істотно спрощує створення протоколів, що вимагають шифрування з автентифікацією, і рятує від безлічі потенційних помилок в їх проектуванні та реалізації. Це є більш стійкою альтернативою патентованим і легалізованим при неправильному виконанні режимам автентифікованого шифрування на блокових шифрах (OCB, GCM). Keccak (на відміну від Skein) не містить у своїй основі блоковий шифр і тому не є універсальним, але може використовуватися в протоколах, де потрібно використовувати блоковий шифр в режимі лічильника. Фактично Keccak може працювати в чотирьох режимах: *MAC*-код, потоковий шифр, потоковий шифр з довільним доступом, генерація симетричних ключів з паролів.

Keccak так само універсальний, як і Skein, але його область застосування може бути ширше. При необхідності цей алгоритм може бути впроваджений як в мініатюрні пристрої з обмеженими ресурсами, так і на високопродуктивні сервери, що працюють з великим обсягом з'єднань.

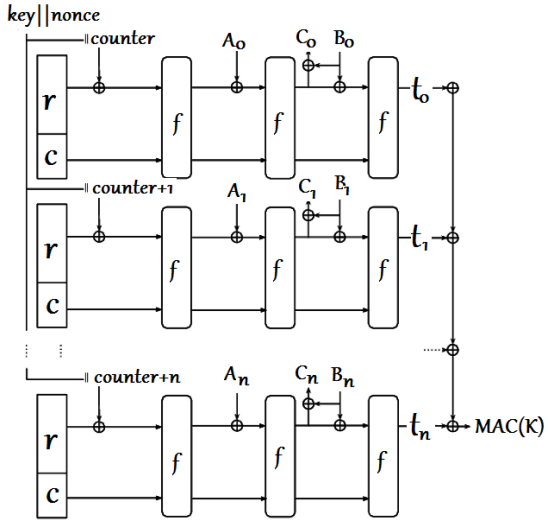


Рисунок 6.7 – Паралельне обчислення Keccak-*MAC*

Обидва алгоритми мають найсильнішу доказову базу серед всіх фіналістів, але, за оцінками Keccak, має більш суворі докази в моделі RO. Зокрема, він не піддається атакам на функції конструкції NarrowPipe. Невирішеними питаннями залишається оптимізація даних алгоритмів виявлення оптимальної кількості раундів хеш-функції.

6.1.6 Колізійні властивості MAC-кодів універсального хешування

Ідея універсального хешування була запропонована Картером і Вегманн для побудови колізійно стійких і високошвидкісних кодів автентифікації [255,256]. Універсальні сімейства хеш-функцій характеризуються прозорими комбінаторними властивостями і мають доказову стійкість. Основні визначення універсальних класів і властивості комбінаторних схем розглянуті в роботах [257,258] і узагальнені в роботах [259,260].

Визначення 6.8 $(N; n, m)$ хеш-сімейство є множина з N функцій H таке, що $h: A \to B$, де $h \in H$, $|A| = n$ и $|B| = m$, $n \geq m$.

Визначення 6.9 $(N; n, m)$ хеш-сімейство є ε-універсальним, якщо для будь-яких двох різних елементів $x_1, x_2 \in A$ існує найбільше εN функцій

$h \in H$ таких, що $h(x_1) = h(x_2)$. Абревіатура $\varepsilon - U$ використовується для позначення ε-універсальних хеш-функцій.

Очевидно, якщо h вибирається випадково з заданого $\varepsilon - U(N; n, m)$ хеш-сімейства, то ймовірність колізії хеш-значень для двох різних вхідних повідомлень $x_1, x_2 \in A$ не перевищує ε:

$$\Pr h \in H\left[h(x_1) = h(x_2)\right] \le \varepsilon.$$

Первісне визначення універсальних хеш-функцій Картера і Вегмана було запропоновано для $\varepsilon = 1/m$. Наступне визначення є узагальненням попереднього.

Визначення 6.10 H являється ε – майже універсальним сімейством хеш-функцій $\left(\varepsilon - AU(N; n, m)\right)$, якщо $\Pr h \in H\left[h(x_1) = h(x_2)\right] \le \varepsilon$, для $x_1, x_2 \in A$, $x_1 \ne x_2$, $1/m < \varepsilon \le 1$.

Наступні визначення визначають класи хеш-функцій суворої універсальності.

Визначення 6.11 H являється ε – строго універсальним сімейством хеш-функцій $\left(\varepsilon - SU(N; n, m)\right)$, якщо для всіх $x_1, x_2 \in A$, $x_1 \ne x_2$ і всіх $a, b \in B$, $\Pr h \in H\left[h(x_1) = a, h(x_2) = b\right] = \varepsilon$, $\varepsilon = 1/|B|$.

Визначення 6.12 H являється $\varepsilon-$ майже строго універсальним сімейством хеш-функцій $\left(\varepsilon - ASU(N; n, m)\right)$, якщо для всіх $x_1, x_2 \in A$, $x_1 \ne x_2$ і всіх $a, b \in B$, $\Pr h \in H\left[h(x_1) = a, h(x_2) = b\right] \le \varepsilon$.

Одним з найбільш відомих універсальних сімейств хеш-функцій є PolyCW хешування (polynomial Carter-Wegman hashing) [255]. Головна властивість PolyCW функції визначена фундаментальною теоремою алгебри: многочлен відмінний від нуля, ступені не менше k, має не менше k корнів. Імовірність колізії для поліноміальної хеш-функції обмежується відношенням k/q, де q – просте число, визначальне поле Z_q для обчислення многочленів. Множина ключів визначається значенням q. Чим більше розмір

ключового простору, тим більшу кількість слів можна хешувати для досягнення допустимої ймовірності колізії. При збільшенні q зростають тимчасові витрати на обчислення хеш-значень в Z_q. При русі q від 2^{32} до 2^{64} зростають тимчасові витрати на обчислення хеш-значень в Z_q, частково знімаються в конструкції сповзаючого поліноміального хешування RPHash (ramped polynomial hashing), використаного в UMAC алгоритмі [261,262]. Відповідно до специфікації UMAC розмір поля обчислення хешу Z_q зростає в міру збільшення довжини повідомлення. Недолік хешування полягає в тому, що ймовірність колізії зростає лінійно з ростом довжини повідомлення і її зменшення можливе шляхом обмеження довжини хешуємого повідомлення або збільшення розміру поля обчислення хешу Z_q.

Ще одним підходом, що знижує протиріччя між обчислювальними витратами в великих полях і необхідністю забезпечити на великій довжині повідомлення мале значення ймовірності колізії, є застосування універсального хешування по алгебраїчним кодами [220]. Зв'язок між універсальним сімейством хеш-функцій і кодовими схемами вперше була відзначена Біербрауером, Джохансоном, Кабатіанскі і Смітсом [263].

У схемах з алгебраїчними кодами (n, k, d) ймовірність колізії визначається значенням $1-d/n$. Для відомих до теперішнього часу кодів ймовірність колізії обмежується, в кращому випадку, значенням обернено пропорційним квадрату розмірності поля Z_q.

В роботі [264] представлений універсальний клас з кодами Ріда - Соломона (Rsh), еквівалентний поліноміальному хешуванню. Відмінність полягає в тому, що обчислення хеш-коду може бути реалізовано не тільки в кінцевому полі простого числа, але і в розширеному полі Галуа. У ряді випадків це може бути кращим, так як спрощується розбиття повідомлення на слова, які повинні бути приведені до розміру поля обчислення хешу. Найбільш прийнятними значеннями q для хешування даних, які лежать в

діапазоні розрядності сучасних процесорів, є 2^{32} і 2^{64}. Імовірність колізії зростає лінійно зі зростанням обсягів даних. Для ймовірності колізії в діапазоні значень $10^{-3}\left(2^{-10}\right)\div 10^{-9}\left(2^{-30}\right)$ розмір хешируємих даних повинен лежати в діапазоні $2^3 \div 2^{22}$ 32-розрядних слів.

При RSh хешуванні обчислення MAC-кодів в кінцевому полі Fq реалізується виразом

$$h_x(m)=\sum_{i=0}^{k} m_i \cdot x_i ,$$

де x – ключове слово, $m=(m_1, m_2, \ldots, m_k)$ – повідомлення, $x, m_i \in F_q$, k – обсяг повідомлення, $k<q$ і визначає $\frac{k-1}{q}$ – $U(q; q_k, q)$ сімейство хеш-функцій $\left(RSh_q\right)$.

У разі, коли $q=p$, де p – просте число, обчислення хеш-кодів в простому полі визначається модулярною арифметикою.

Обчислення RSh_q хеш-функції можна здійснити за ітераційною схемою Горнера з однією операцією множення і складання в кінцевому полі на кожному кроці. Кращий результат досягається в арифметиці $Z_p(w)$ при якомога великому простому числі $p(w)\le 2w$. Ітераційна схема Горнера обчислення хеш-функції передбачає обчислення

$$y \leftarrow xy + m \cdot \bmod p(w). \qquad (6.9)$$

Як випливає з аналізу залежностей ймовірності колізії від значення поля обчислень, розмір кінцевого поля F_q повинен бути якомога більшою. Застосування операції $\bmod p(32)$ потребує $12.4cpb$ (циклів процесора на байт).

Можна використовувати більш ефективний алгоритм для обчислення $y \leftarrow xy + m \cdot \bmod p(w)$ [261]. Простий модуль $p(w)$ можна представити у

вигляді $p(w)=2^w-c$. використовуючи уявлення $xy=a2^w+b$ і враховуючи, що $a\cdot 2^{32}=ca$ в $Z_p(w)$, отримаємо

$$y=xy+m\cdot \bmod p(w)=ca+b+m\cdot \bmod p(w). \qquad (6.10)$$

З аналізу останнього виразу випливає, що при обчисленнях на одному кроці ітерації значення *y* буде потрібно одне множення і два складання на w розрядних регістрах і кілька команд, які дозволяють контролювати вихід результату обчислень за межі діапазону представлення чисел в $Z_p(w)$ Недоліком зазначеного високошвидкісного алгоритму обчислень є зменшення ключового простору до величини $d=2^w/c$ і збільшення ймовірності колізії до $k2^{-d}$.

Для максимального значення модуля $p(32)=2^{32}-5$ ефективний алгоритм використовує вісім рядків асемблерної програми і досягає продуктивності 3.69*cpb* [261]. Зменшення ключового простору призводить до величини $2^{32}/5=2^{29}$ і збільшення ймовірності колізії до $k2^{-29}$. Практична схема ефективного алгоритму обчислення *Rshp* хешу повинна включати ще одне додаткове перетворення. Вектори чисел з w бітами, які мають елементи за діапазоном уявлення $Zp(w)$, необхідно перетворити в вектор, який їх не має, за допомогою так званого подвійного подання, як це зроблено в UMAC алгоритмі. Це призведе до подвоєння розміру даних і до збільшення ймовірності колізії до $k2^{-(d-1)}$ відповідно, знизиться швидкість обчислень. Так, за оцінками розробників UMAC алгоритму, швидкість обчислень для $p(32)$ знижується до значення 3.86*cpb* [261].

Подальше підвищення розрядності регістрів обчислення *RShp* хеш $p(w)$ за значення $w-32$ призводить до значного зниження швидкості обчислень. Так, реалізація *RShp* хеш-функції для $p(64)=2^{64}-59$ має ймовірність колізії $k2^{-49}$, використовує 40 рядків асемблерної програми і має

пікову продуктивність $6.8cpb$ [265]. Зниження швидкості обчислень всього в два рази визначається тим, що автори використовували MMX технологію, яка заснована на чотирьохвекторному поданні 16-розрядних даних і відповідних командах швидкісного множення.

В роботі [265] представлені результати збільшення ймовірності колізії від використовуваних модулів. Аналіз показує, що зменшення ключового простору досягає величини $\approx 2 \div 3$ біт і призводить відповідно до збільшення ймовірності колізії в $2^2 \div 2^8$ раз в залежності від використовуваного модуля.

Лінійне зростання ймовірності колізії для *RShp* хешування обмежує розмір хешируємого повідомлення. Чим більший розмір ключового простору, тим більшу кількість слів можна хешувати для досягнення допустимої ймовірності колізії. Потужність множини ключів визначається значенням простого числа $p(w)$, визначального поля $Z_p(w)$. При збільшенні $p(w)$ зростають тимчасові витрати на обчислення многочленів в $Z_p(w)$. При русі $p(w)$ від 2^{32} до 2^{64} тимчасові витрати збільшуються в два рази.

Арифметика $GF(2w)$ менш зручна для сучасних мікропроцесорів, хоча забезпечує легкий поділ хешируємих бітових рядків на бітові підрядки. При цьому відсутні втрати по ймовірності колізії і за обсягом ключових даних, які виникають при хешуванні в арифметиці $Z_p(w)$.

Обчислення *RSh* хеш-кодів визначається виразом (1.9) в розширеному кінцевому полі F_q характеристики $p = 2$, так же, як і в $Z_p(w)$, реалізуються за схемою Горнера $y \leftarrow xy + m$. Ключовим моментом даної схеми є обчислення добутку елементів поля $GF(2^w)$. Відомо кілька ефективних алгоритмів швидкого множення в $GF(2^w)$, орієнтованих на табличне множення елементів поля або є модифікаціями схеми Монтгомері.

Практичні схеми хешування повинні включати класи хеш-функцій з великим коефіцієнтом стиснення для даних можливо дуже великого обсягу.

Для цих цілей інтерес представляють сімейства хешів на основі довгих алгебраїчних кодів. Довгі алгебраїчні коди реалізуються в класі кодів по алгебраїчних кривих [264]. В основу кодування належить відображення векторного простору Рімана - Роха над кінцевим полем раціональних функцій по точках кривій алгебри. Застосування скалярного твори по раціональним функціям алгебраїчних кривих визначає метод універсального хешування. Алгебраїчні криві з великим числом точок і щільно упакованим по полюсах функціональним полем раціональних функцій реалізують найкращі результати універсального хешування. Універсальне хешування за кодами Ріда - Соломона в алгеброгеометричній інтерпретації визначається як хешування за проективною прямою.

У параметричній інтерпретації є поліноміальним хешуванням.

Параметрично більш складними кривими є максимальні криві, криві Судзукі і криві Рі. Криві мають найменші відносини значення полюса раціональних функцій до числа точок, що визначає найменше значення ймовірності колізії в схемі універсального хешування, і є найкращими для застосування.

6.1.7 Формулювання завдань досліджень

Аналіз методів універсального і строго універсального хешування показує, що рішення задачі побудови колізійно стійких функцій хешування і ключових функцій хешування, які відповідають міжнародним вимогам гарантованої стійкості до атак, складності і швидкості обчислення, характеристикам і реалізацій алгоритму, можливо в теорії доказово стійкою автентифікацією. Основні положення теорії автентифікації визначені в [255, 266]. Наукова задача побудови доказово стійкої автентифікації вперше сформульована в [266]. Вирішення цього завдання було запропоновано в класі універсальних хеш-функцій, як автентифікації з максимальною теоретично досяжною секретністю. Ідеї універсальної автентифікації

отримали розвиток в теорії безумовної автентифікації з використанням строго універсального хешування [237,258,259].

Основне протиріччя доказово стійкої автентифікації полягає в тому, що для забезпечення гарантованої ймовірності обману на рівні нижньої межі розмір ключа повинен бути не менше розміру повідомлення, а фіксування розміру ключа на нижній межі визначається потужністю простору хешів призводить до пропорційного зростання ймовірності колізії від довжини даних. На практиці це означає, що для автентифікації з секретністю на нижній межі $P_{\text{col}} = 1/|B|$ ($|B|$ – потужність простору хеш-кодів) по закритому каналу зв'язку слід передавати ключових даних більше, ніж за відкритим - інформаційних даних.

Аналіз методів універсального хешування показує, що основними шляхами вирішення цієї суперечності є універсальне хешування на основі алгебраїчного кодування за лінійним вектором простору Рімана - Роха, побудованому за раціональним функціям функціонального поля, асоційованого з кривій алгебри на проектному різноманітті її точок. Основний результат залежить від того, що ймовірність колізії випливає з відносини значення полюса раціональних функцій функціонального поля до числа точок кривій алгебри. Імовірність колізії зв'язується з довжиною повідомлення, ключа і полем обчислення хешів, а також з функціональним полем кривій алгебри. Ключове простір визначається числом точок кривій алгебри. Вибір кривій алгебри і асоційованого з нею функціонального поля дозволяє оптимізувати витрати на автентифікацію.

Найкращий результат щодо значення полюса раціональних функцій до потужності точок кривої отримано для кривих Судзукі. Мета роботи - розробка методу універсального хешування по кривій Судзукі для побудови доказово стійкої автентифікації.

Функція мети (Z) полягає в забезпеченні гарантованої ймовірності колізії P_{col} функції хешування, мінімізації витрат на ключовий простір $|K|$,

складності обчислень $N_{\text{выч}}$ в умовах фіксованої довжини повідомлень $\log|M|$, поля обчислення F_q характеристики 2 непарного степеня розширення і проектованого різноманіття по кривій Судзукі $F_q(C)$:

$$Z = \min\left\{P_{\text{col}}, |K|, N_{\text{выч}}\right\} \left| \log|M| = fix,\ F_q(C) = \text{var}. \right. \qquad (6.11)$$

Наукова задача полягає в розробці методу універсального хешування по раціональним функціям кривій алгебри Судзукі для побудови доказово стійкою автентифікації повідомлень із забезпеченням гарантованої ймовірності колізії та мінімізацією витрат на ключове простір, розмір хеш-коду і складність обчислень.

Для досягнення мети слід вирішити такі приватні задачі:

1) провести аналіз методів побудови *MAC*-кодів універсального і строго універсального хешування;

2) розробити метод універсального хешування по раціональним функціям кривій алгебри Судзукі;

3) розробити метод швидкісного універсального хешування по кривій Судзукі на основі застосування схеми обчислення Горнера;

4) розробити метод універсального хешування з обмеженням функціонального поля алгебраїчних кривих зі зменшеною складністю обчислень;

5) розробити метод каскадного універсального хешування по кривій Судзукі на основі твору функціональних полів;

6) розробити практичні рекомендації щодо застосування універсального хешування по кривій Судзукі.

6.1.8 Висновки

Збільшення обсягу доступної обчислювальної потужності, розподілені обчислення і здешевлення вартості обумовлює відмову від комерційного

використання криптографічних стандартів 90-х (MD5, SHA-1). Практична реалізація класичних конструкцій на процесорах і архітектурі нового покоління ведуть до збільшення вразливостей. Особливості атрибутів конструкцій хеш-функцій роблять все більший вплив на безпеку і продуктивність в кінцевій реалізації. У зв'язку з цим актуальною стає задача розробки криптопримітивів з урахуванням нових вимог до колізійної стійкості і безпеки, продуктивності і практичної реалізації хеш-функцій з використанням нових конструкцій з метою зниження уразливості до атак. Нове покоління криптопримітивів підтримує різні комбінації вихідних параметрів без використання додаткових засобів. Перспективним напрямків досліджень є застосування універсальних сімейств хеш-функцій, що характеризуються прозорими комбінаторними властивостями і мають доказову стійкість. Одним з найбільш відомих універсальних сімейств хеш-функцій є PolyCW хешування, недоліком якого є істотне збільшення витрат часу на обчислення хеш-значень. Існуючі практичні реалізації універсального хешування (UMAC) частково знімають обмеження, пов'язані з обчисленнями в великих полях. Одним з підходів, що знижує протиріччя між обчислювальними витратами в великих полях і необхідністю забезпечити на великій довжині повідомлення мале значення ймовірності колізії, є застосування універсального хешування по алгебраїчним кодами. Завданням другого розділу є аналіз методів універсального хешування і оцінка асимптотичних характеристик універсального хешування.

6.2 Доказово стійка автентифікація на основі методів універсального хешування

Побудова доказово стійкої автентифікації визначається класами універсальних хеш-функцій з прозорими комбінаторними властивостями. Універсальне хешування зв'язується з розподілами хешів для повідомлень по ключовому простору, що визначає автентифікацію з лічильником на масивах

автентифікатор. Автентифікація з лічильником передбачає застосування сеансового ключа для кожного MAC-обчислення. Колізійні оцінки майже строго універсального хешування зв'язуються з розподілами хешів для пар повідомлень по ключовому простору, що визначає безумовну автентифікацію на масивах автентифікатор.

Розробка методів універсального хешування лежить в площині оптимізації витрат ключового простору при хешуванні даних фіксованої довжини і ймовірності колізії. Таким чином, завданням є порівняння методів універсального хешування для побудови доказово секретної автентифікації. Класифікація методів універсального хешування представлена в підрозділі 6.2.1 за результатами робіт [237,255,258,259]. Безумовна автентифікація на основі строго універсального хешування розглянута в підрозділі 6.2.2. Визначення універсального хешування на основі алгебраїчного кодування по раціональним функціям алгебраїчних кривих, властивості універсальних хеш-функцій, асимптотичні кордону для ймовірності колізії представлені в підрозділах 2.3, 2.4.

6.2.1 Класифікація методів універсального хешування

Коди автентифікації повідомлень в поданні Картера - Вегмана визначаються сімейством хеш-функцій [255]. масив значень MAC-кодів складається з N рядків, n стовпців, елементи приймають одне з m значень. Кожна функція $f \in H$ визначається значенням використовуваного ключа, зв'язується з рядком і визначає правило відображення елементів множини X (номерів стовпців масиву) в елементи Y (власні значення елементів масиву).

Визначення MAC-кода з урахуванням секретності має такий вигляд.

Визначення 6.13 [267]. MAC-код $f : A \to B$ являється $(t; \varepsilon; q)$ секретним, якщо при випадково взятому ключі k противник не може підробляти нове повідомлення за час t з ймовірністю більше ніж ε, якщо йому надані значення q MAC-кодів інших повідомлень за його вибором.

Зауваження 6.3

1. Представлення MAC-кодів у вигляді масиву значень дозволяє розглядати статистичні розподілу автентифікатор на просторі $A \times B$, що в свою чергу пов'язано з колізійними характеристиками;

2. Для точного обчислення імітаційної і колізійної стійкості MAC-кодів необхідно використовувати статистику спільних розподілів MAC-кодів по ключам для вихідних і нав'язуваних повідомлень. Для практичних MAC-кодів знання такої статистики виглядає проблематичним через надзвичайно великих розмірів масиву автентифікаторів;

3. Нижні межі для ймовірності підміни визначаються потужністю простору ключів і MAC-кодів, не враховують статистичні властивості масивів автентифікаторів. Вимоги до ймовірності підміни визначають мінімальні вимоги до розміру ключового простору і простору MAC-значень;

4. Верхні межі для ймовірностей імітації і підміни пов'язані з комбінаторними властивостями MAC- масивів і визначають значення ймовірності колізій на просторі $A \times B$ для найгіршого випадку вибору ключів і повідомлень.

Колізійні оцінки майже універсального хешування зв'язуються з розподілами хешів для повідомлень по ключовому простору, що визначає автентифікацію з лічильником на масивах автентифікаторів. Автентифікація з лічильником передбачає застосування виняткового сеансового ключа для кожного MAC-обчислення.

Універсальне хешування реалізується на основі таких методів:

- скалярного добутку;
- поліноміального хешування;
- хешування на основі алгебраїчних кодів;
- скалярного добутку по раціональним функціям алгебраїчних кривих.

6.2.1.1 Універсальне хешування на основі скалярного добутку

Хеш-обчислення y над кінцевим полем F_q визначається функцією виду

$$y = \sum_{i=1}^{k} x_i m_i , \tag{6.12}$$

де $y, x_i, m_i \in F_q, m_i$ – слова повідомлення;

x_i – слова ключа;

k – число слів повідомлення.

Зауваження 6.4

1. Хеш-функція на основі скалярного добутку визначає хеш-клас $\varepsilon - U\left(q^k, q^k, q\right)$ з ймовірністю колізії $\varepsilon = 2/q$;

2. Реалізується доказово секретна автентифікація з максимальною теоретично досяжною секретністю;

3. Недоліком універсального хешування (6.1) є вимога - розмір ключового простору повинен бути не менше простору повідомлення.

Обмеження на розмір ключового простору знімається в методі поліноміального хешування.

6.2.1.2 Поліноміальне універсальне хешування

Поліноміальне універсальне хешування (polynomial Carter-Wegman hashing) запропоновано для зняття обмеження на простір ключів.

Поліноміальне хешування над кінцевим полем F_q визначається функцією виду

$$y = \sum_{i=1}^{k} m_i x^i , \tag{6.13}$$

де $y, x_i, m_i \in F_q, m_i$ – слова повідомлення,

x – ключове слово,

k – число слів повідомлення $k < q$.

Зауваження 6.5

1. Хеш-функція на основі поліноміального обчислення визначає хеш-клас $\varepsilon - U\left(q, q^k, q\right)$ з ймовірністю колізії $\varepsilon = k/q$. Імовірність колізії для поліноміальної хеш-функції обмежується відношенням k/q, де q – просте число, що визначає поле F_q. Значення ε визначається фундаментальною теоремою алгебри: многочлен відмінний від нуля, ступені не менше k, має не менше k корнів;

2. Множина ключів визначається значенням q. Чим більший розмір ключового простору, тим більша кількість слів може хешуватися для досягнення допустимої ймовірності колізії;

3. Обчислення хеш-кодів в простому полі визначається модулярною арифметикою. Хеш-функцію (2.2) можна обчислити за ітераційною схемою Горнера з однією операцією множення і складання в кінцевому полі на кожному кроці. Кращий результат досягається в арифметиці Z_q при якомога великому простому числі $q < 2^w$.

Як випливає з аналізу залежностей ймовірності колізії від значення поля обчислень, розмір кінцевого поля F_q повинен бути якомога більшим;

4. Практична схема ефективного алгоритму хеш-обчислення повинна включати ще одне додаткове перетворення. Вектори чисел з w бітами, які мають елементи поза діапазоном уявлення Z_q, необхідно перетворити в вектор, який їх не має, за допомогою, так званого подвійного подання, як це зроблено в *UMAC* алгоритмі. Це призводить до подвоєння розміру даних і до збільшення ймовірності колізії, і відповідно знижується швидкість обчислень;

5. Арифметика над розширеним полем F_q характеристики 2 є менш зручною для сучасних мікропроцесорів, хоча забезпечує легке поділ хешируємих бітових рядків на бітові підрядки. При цьому відсутні втрати по

ймовірності колізії і за обсягом ключових даних, які виникають при хешуванні в арифметиці Z_q. Відомо кілька ефективних алгоритмів швидкого множення в F_{2^w}, орієнтованих на табличне множення елементів поля або є модифікаціями схеми Монтгомері.

Недоліком поліноміального хешування є вимога - розмір простору повідомлень обмежується умовою для ймовірності колізії $\varepsilon = k/q$ і розміром поля обчислень. Обмеження на розмір простору повідомлень знімається в методі на основі алгебраїчного кодування.

6.2.1.3 Універсальне хешування на основі алгебраїчних кодів

Хеш-значення для повідомлення $m = (m_0, m_1, \ldots, m_{k-1})$, $m_i \in F_q$ визначається виразом

$$h_x(m) = \sum_{i=0}^{k-1} f_{i,x} \cdot m_i , \qquad (6.14)$$

де $h_x(m) \in F_q$ и $f_{i,x}$ – значення елементів стовпця x виробленої матриці G лінійного кода $(n, k, d)_q$.

Зауваження 6.6

1. Універсальне хешування, утворене $(n, k, d)_q$ лінійним кодом, визначає $1 - \frac{d}{n} - U(n, q^k, q)$ майже універсальне сімейство хеш-функцій;

2. Верхня межа ймовірності нав'язування для універсального сімейства хеш-функцій на основі алгебраїчного коду $(n, k, d)_q$ без одиниці визначається відносною кодовою відстанню, нижня межа - значності коду;

3. Криптоаналіз прямої атаки на універсальне сімейство хеш-функцій з алгебри кодуванням за умови багаторазових спроб вгадування MAC - значень і ключів розглянуто в [268]. Імовірність нав'язування шляхом підміни і

вгадування повідомлень обернено пропорційно залежить від значності кодових слів і відносної кодової відстані $(n, k, d)_q$ кода;

4. Для практичної автентифікації слід використовувати $(n, k, d)_q$ коди великої розмірності і з великим відносним кодовою відстанню. Найкращим алгебраїчним кодом для побудови універсальних хеш-функцій є код Ріда - Соломона.

6.2.1.4 Універсальне хешування за раціональними функціями алгебраїчних кривих

Хеш-функція $h_{P_j}(m) \in F_q$ для повідомлення $m = (m_1, \ldots, m_k)$, $m_i \in F_q$ в точці P_j визначається виразом

$$h_{P_j}(m) = \sum_{i=1}^{k} f_i(P_j) m_i, \qquad (6.15)$$

де $f_i(P_j)$ – значення раціональної функції в точці P_j кривої χ;

$f_i \in F_q(\chi) \setminus \{0\}$ – раціональні функції функціонального поля кривої χ з упорядкованими порядками полюсів $0 < \rho_1 < \rho_2 < \ldots < \rho_k$.

Хеш-функція $h_{P_j}(m)$ визначає універсальний хеш-клас $\varepsilon - U(N, q^k, q)$, де N – число точок алгебраїчної кривої; q^k – обсяг простору повідомлень; q – обсяг простору хеш-кодів; $\varepsilon = \rho_k / N$ – ймовірність колізії; ρ_k – значення полюса раціональної функцій f_k.

Універсальне хешування за раціональними функціями алгебраїчних кривих визначається властивостями лінійного базисного простору, асоційованого з функціональним полем кривої.

Універсальне хешування за раціональним функціям алгебраїчних кривих має кращі асимптотичні результати. Верхня межа ймовірності колізії для алгебраїчного універсального хешування $\varepsilon = \rho_k / N$ визначається

відношенням значення полюса раціональної функції f_k до числа точок кривій алгебри.

Проблематика побудови схем універсального хешування по раціональним функціям алгебраїчних кривих полягає у виборі алгебраїчних кривих з необхідними параметрами.

6.2.2 Методи суворо універсального хешування

Строго (майже строго) універсальне хешування визначає безумовну автентифікацію, що було представлено Стінсон [258,259]. Колізійні оцінки майже строго універсального хешування зв'язуються з розподілами хешів для пар повідомлень по ключовому простору, що визначає безумовну автентифікацію на масивах автентифікатор.

У загальному випадку можна говорити про розподіли t повідомлень, що визначає t пов'язану автентифікацію.

Для побудови строго універсального хешування використовують методи на основі ортогональних масивів.

Ортогональним масивом $OA_{\lambda}(t,k,\upsilon)$ називається масив елементів $y_i \in Y$ за допомогою стовпців, відповідними елементам множини X, і рядками, які визначаються елементами множини m, в якому для будь-якої вибірки з t елементів $y_1, y_2, \ldots, y_t$ з Y існує тільки λ функцій $f \in m$, для яких справедливо $f(x_i) = y_i$, $i = 1, 2, \ldots, t$, де X, Y є множинами з k і υ елементів відповідно, і H є безліч функцій, які здійснюють відображення $f : X \to Y$ [269].

Строго універсальний клас хеш-функцій, побудований на ортогональному масиві сили $t = 2$, має параметри $\frac{k}{q^b} - SU\left(q^{a+b}, q^{ka}, q^b\right)$.

Сімейство хеш-функцій визначається відображенням $\phi : F_{q^a} \to F_{q^b}$.

X є множиною поліномів $p(X)$, визначеними над F_{q^a} ступені $\le k$, без постійного члена.

Елементи матриці відображення $X \to Y$ на перетині (u, υ) рядка, $u \in F_{q^a}$, $\upsilon \in F_{q^b}$, і $p(X)$ стовпчика можна визначити як $\phi(p(u)) + \upsilon = y$.

Зауваження 6.7

1. Якщо $k = 1$, маємо строго універсальний клас хеш-функцій $\frac{1}{q^b} - SU(q^{a+b}, q^a, q^b)$. Розмір ключових даних N визначається добутком простору автентифікаторів і простору повідомлень.

2. Для майже строго універсального хешування знижуються вимоги до розміру ключових даних, які обмежуються розмірами поля обчислень F_{q^a} і F_{q^b}.

3. Лінійне відображення $\phi: F_q^n \to F_q^m$ визначає множення елементів в F_{q^n}, проектування m координат $F_{q^n} \to F_{q^m}$ і складання в F_{q^m}

Строго універсальне хешування на основі майже незалежних масивів.

Майже незалежні масиви (almost independent arrays) були розглянуті Куросавой, Стінсон [270,271]. Теорія майже незалежних масивів знімає обмеження на рівновірогідний розподіл наборів хешів за стовпцями масиву. Майже незалежні масиви є узагальненням ортогональних масивів.

Нехай $(n, k)_p$ – масив, що містить n рядків, k стовпців і записи з набору p елементів. Для $\forall a \in F_p$ частота $v_a(u)$ появи значення a в стовпцях масиву $u = (u_1, u_2, \ldots, u_n) \in F_p^n$ задовольняє умові $|v_a(u)/n - 1/p| \le \varepsilon_1$, і для любих пар стовпців u, u' частота $v_{(a,a')}(u, u')$ появи в стовпцях значень a і

a' задовольняє умові $\left|v_{(a,a')}(u,u')/n-1/p^2\right|\le\varepsilon_2$. Тоді $(n,k)_p$ – масив є сімейство $\varepsilon-ASU(n,k,p)$ хеш-функцій і $\varepsilon=\left(p^{-2}+\varepsilon_2\right)\Big/\left(p^{-1}-\varepsilon_1\right)$.

Зауваження 6.8

1. Параметр ε визначається умовною ймовірністю появи будь-яких записів a,a' для різних стовпців u, u' при рівновірогідному виборі i рядка $\varepsilon=\Pr(u_i'=a'/u_i=a)$ і характеризує відхилення від рівномірного розподілу спільних ймовірностей появи кодових комбінацій в t довільних стовпцях випадково обраного рядка $(n,k)_p$ масива.

2. Значення параметра залежно від ε визначає ймовірність колізії MAC-кодів і в загальному випадку, як показано в [272], колізійні властивості t кратних кодів автентифікації.

Практична побудова майже незалежних масивів проблематична, так як потрібні методи, які дозволяють формувати масиви хешів з заданими розподілами по стовпцях. В цьому відношенні для побудови строго універсальних хеш-функцій більш продуктивним є застосування слабо зміщених масивів.

Строго універсальне хешування на основі слабо заміщених масивів.

Слабо зміщені масиви розглянуті в роботах [273,274] для масивів дискретних значень великої розмірності з розподілом, незначно відрізняється від рівномірного. Слабо зміщені масиви визначають властивості розподілів хешів в стовпцях масиву. Нехай $(n,k)_p$ – масив, містить n рядків, k стовпців і записи з набору p елементів и $0\le\varepsilon\le1$. Масив $(n,k)_p$ являється ε - зміщеним (ε-biased), якщо будь-яка нетривіальна лінійна комбінація стовпців має зміщення $bias\le\varepsilon$. зсув вектора u визначається як

$$bias(u)=\frac{1}{n}\left|\sum_{i\in F_p}\delta_i(u)\xi^i\right|=\frac{1}{n}\left|\sum_{i\in F_p}v_i(u)\xi^i\right|, \qquad (6.16)$$

де $v_i(u)$ – частота появи елемента i в послідовності u, $v_i(u)=\frac{n}{p}+\delta_i(u)$, де $\delta_i(u)$ – відхилення частоти $v_i(u)$ від середнього значення і $\sum_{i\in F_p}\delta_i(u)=0$; ξ – комплексний корінь p - ступеня з одиниці.

Зауваження 6.9

1. Зсув масиву є властивістю F_p- лінійного коду, побудованого за допомогою стовпців виродженої матриці.

2. Для двійкових масивів параметр ε зміщення прямо зв'язується з ймовірності появи 0 і 1 в стовпцях масиву.

3. Для строго універсального класу, масив хеш-значень визначається $(n,k)_p$ масивом зі зміщенням рівним нулю [275].

Практичним методом побудови слабо зміщених масивів є метод сум експонент Вейля - Карлітца - Ушіями (СКУ).

Метод сум експонент СКУ визначає масив $\left(p^f, f*(n-n/p)\right)_p$ зі зміщенням $bias\le(n-1)p^{-f/2}$, з записами виду $Tr\left(a_j\alpha^i\right)$, де a_j – базис поля $F_{p^f}\left|F_p\right.$, $i\le n$ і i не кратно p, $Tr:F_{p^f}\to F_p$ – слід елемента $a_j\alpha^i$.

Масив автентифікаторів $(n,k)_p$ являється ε майже строго універсальним ASU_2, якщо кожен стовпець має зсув 0 і для двох записів e,e' одного рядка в будь-яких стовпчиках c,c' умовна ймовірність $\Pr(c_i=e\,|\,c_i'=e')\le\varepsilon$ і рівномірний розподіл номера рядока i [276]. Рядок масива $(n,k)_p$ визначається значенням ключа, стовпець - повідомленням джерела і значення запису являє собою автентіфікаційній тег.

Зауваження 6.10

1. Універсальне хешування визначається через слабо заміщені масиви, є узагальненням конструкцій лінійних кодів, СКУ масивів.

2. Побудова ASU_2 автентифікації визначається тим, що використовується спеціальне індексування рядків масиву автентифікатора і записів, що збільшує простір ключів і записів і призводить до кращих оцінками параметрів автентифікації.

6.2.3 Універсальне хешування на основі алгебраїчного кодування

Універсальне хешування за раціональними функціям алгебраїчних кривих визначається виразом (6.14) і асоціюється з кодовими схемами, в основу яких покладено відображення векторного простору над кінцевим полем раціональних функцій по точках кривої алгебри. Побудова кодів по алгебраїчних кривих вперше запропоновано В.Д. Гоппе [266].

Алгеброгеометричний підхід має наступне визначення [264].

Визначення 6.4 Нехай

χ – алгебраїчна крива над полем F_q;

$F_q(\chi)$ – поле F_q раціональних функцій на χ;

$\mathrm{Div}_q(\chi)$ – дільник кривої χ;

$\mathrm{div}(f)$ – дільник, асоційований з $f \in F_q(\chi) \setminus \{0\}$;

$L(G)$ – векторний простір Рімана - Роха, асоційоване з $G \in \mathrm{Div}_q(\chi)$ так, що $L(G) = \{ f \in F_q(\chi) \setminus \{0\} : G + \mathrm{div}(f) \succ 0 \} \cup \{0\}$; $\ell(G) := \dim(L(G))$;

$D := P_1 + \ldots + P_n$, де $P_1, \ldots, P_n$ – раціональні точки кривої χ.

Алгеброгеометричний код $C_{D,G} := e(L(G))$, асоційований з D і G, визначається як F_q – лінійне відображення виду $e = e_{P_1,\ldots,P_n} : L(G) \to F_q^n$:

$$f \to (f(P_1), \ldots, f(P_n)). \tag{6.17}$$

Лема 6.1 [264]. Нехай $k := \dim(C_{D,G})$ і d – мінімальна відстань коду $C_{D,G}$. Тоді 1) $k = \ell(G) - \ell(G - D)$, 2) $d \geq n - \deg(G)$.

Слідство 6.1 [264]. Нехай $C_{D,G}$ – алгеброгеометричний код з параметрами k і d. Нехай g – код алгебраїчної кривої.

1. Якщо $n > \deg(G)$, тоді $k = \ell(G)$. У випадку $k \geq \deg(G) + 1 - g$ має сенс $d + k \geq n + 1 - g$. Породжуюча матриця $C_{D,G}$ має вид

$$M = \begin{vmatrix} f_1(P_1) & f_1(P_2) & \ldots & f_1(P_n) \\ f_2(P_1) & f_2(P)_2 & \ldots & f_2(P_n) \\ \vdots & \vdots & \ldots & \vdots \\ f_k(P_1) & f_k(P_2) & \ldots & f_k(P_n) \end{vmatrix},$$

де $f_1, \ldots, f_k$ утворюють базис $L(G)$.

2. Якщо $n > \deg(G) > 2g - 2$, тоді $k = \deg(G) + 1 - g$.

3. Алгеброгеометричний код має параметри

$$\left[n, \deg(G) - g + 1, d \right], \; d \geq n - \deg(G) \tag{6.18}$$

і визначається як

$$C = \left\{ f(P_1), \ldots, f(P_n) \middle| f \in L(G) \right\}. \tag{6.19}$$

Для хешування на основі алгеброгеометричного кодування отримаємо універсальне сімейство хеш-функцій з параметрами $1 - \frac{d}{n} - U\left(n, q^k, q\right)$ (див. зауваження 6.10).

Універсальне хешування за PC кодам (Rsh) в алгеброгеометричній інтерпретації має наступний вид [277]. Нехай F – алгебраїчне покриття F_q. Точки χ визначаються гомогенними координатами (x, y), мають значення $P_i = (\alpha_i, 1)$, $0 \leq i \leq q - 1$ і $Q = (1, 0)$ – особлива точка (точка невизначеності). Нехай $f \in F_q(\chi)$ – раціональні функції, які визначені в кожній P_i з коефіцієнтами в F_q і які мають полюс порядки менше, ніж m в точці Q, і немає інших полюсів. Раціональна функція f має вид $\frac{\alpha(x, y)}{\beta(x, y)}$, де $\alpha(x, y)$ і

$\beta(x, y)$ – гомогенні поліноми ступеня $< k$. Алгеброгеометричний код C визначимо як

$$C = \left\{ \left(f(P_0),\ f(P_1), \ldots,\ f\left(P_{q-1}\right)\right) \middle| f \in L(mQ) \right\}.$$

Код $\tilde{N}$ має розмірність простору $L(mQ)$, $g = 0$, $k = \dim C = m - g + 1 = m + 1$ і мінімальна відстань $d \geq n - m$.

Таким чином, отримаємо PC код $(q, k, q - k + 1)_q$ в алгеброгеометрична інтерпретація і універсальний клас хеш-функцій $\frac{k-1}{q} - U(q, q^k, q)$.

Наступний приклад – алгеброгеометричного хешування по кодам Ерміта (Hch). Крива Ерміта визначається рівнянням $y^{\sqrt{q}} + y = x^{\sqrt{q}+1}$ над квадратичним полем $F_{q=p^2}$. Число точок кривої $N = q\sqrt{q+1}$, род $g = \sqrt{q}\left(\sqrt{q} - 1\right)/2$. Нехай $P_\infty = (0:1:0)$ и $G = mP_\infty$. Базис простору $L(mQ)$ задається функціями виду $\left\{ x^i \cdot y^j : i\sqrt{q} + j(\sqrt{q} + 1) \leq m \right\}$. При алгеброгеометричному кодуванні код Ерміта має параметри $\left[q\sqrt{q}, k, d \geq q\sqrt{q} - k + 1 - g \right]$, що призводить до параметрів універсального хешування: $\frac{1}{\sqrt{q}}\left(\frac{k-1}{q} + 1 - \frac{1}{\sqrt{q}} \right) - U\left(q\sqrt{q}, q^k, q\right)$ [278].

Залежності ймовірності колізій для універсального хешування на PC кодах і кодах Ерміта від довжини хешованого повідомлення представлені на рис. 6.9. Для асимптотичного кордону побудовані графіки залежності ймовірності колізії для HCh_q від довжини даних і розміру поля обчислень, які представлені на рис. 6.9, де:

– ймовірність колізії для HCh_q

$$\varepsilon = \frac{1}{\sqrt{q}}\left(\frac{k-1}{q}+1-\frac{1}{\sqrt{q}}\right) \text{ при } 1 \le k \le q\sqrt{q};$$

– ймовірність колізії для RSh_q

$$\varepsilon = \frac{k-1}{q} \text{ при } 1 \le k \le q.$$

Для малих довжин даних, коли $k < \sqrt{q}+1$, схема RSh_q по асимптотичній оцінці має більшу перевагу, так як реалізує найменшу ймовірність колізії. Аналіз графіків показує, що HCh_q забезпечує збільшення обсягу хешованих даних в $\sqrt{q}$ раз. Для $q = 2^{32}$ обсяг хешованих даних може досягати 2^{39} 32-разрядних слів, що перекриває весь діапазон практичних застосувань. При цьому ймовірність колізії обмежується значенням 10^{-3} $\left(2^{-10}\right)$. Зменшення ймовірності колізії можливо збільшенням q. Так, при $q = 2^{64}$ ймовірність колізії зменшується до асимптотичного кордону не менше ніж на п'ять порядків до 10^{-8}.

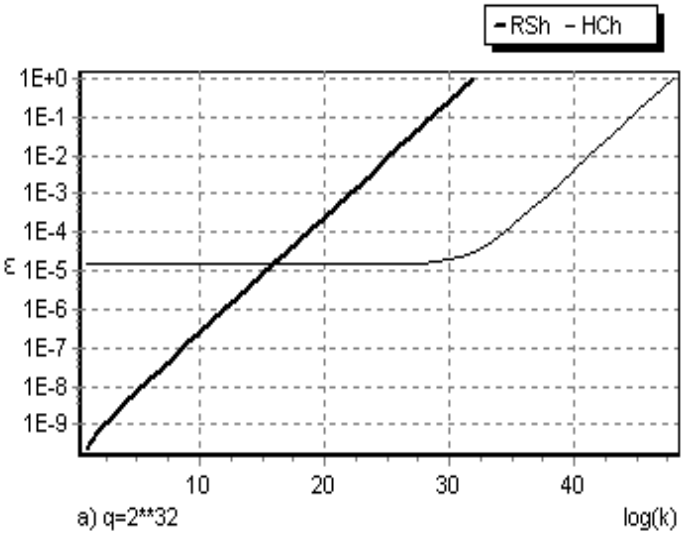


Рисунок 6.9 – Графіки залежності асимптотичного кордону ймовірності колізії для схеми хешування з НС від довжини даних при різних значеннях розміру кінцевого поля обчислення

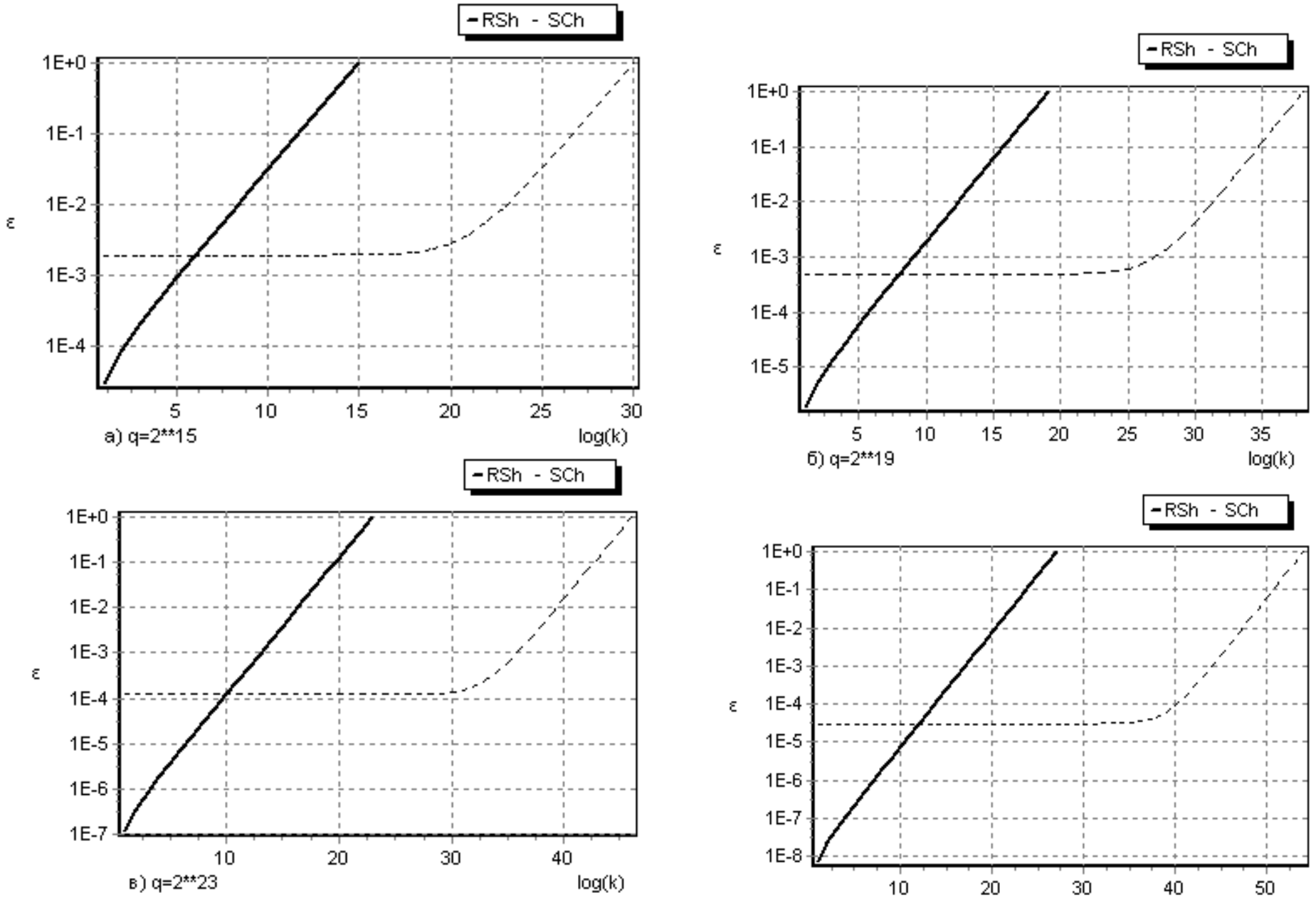


Рисунок 6.10 – Графіки залежності асимптотичних кордонів ймовірності колізій для хешування SChq від довжини даних при різних значеннях розмірів кінцевого поля обчислення

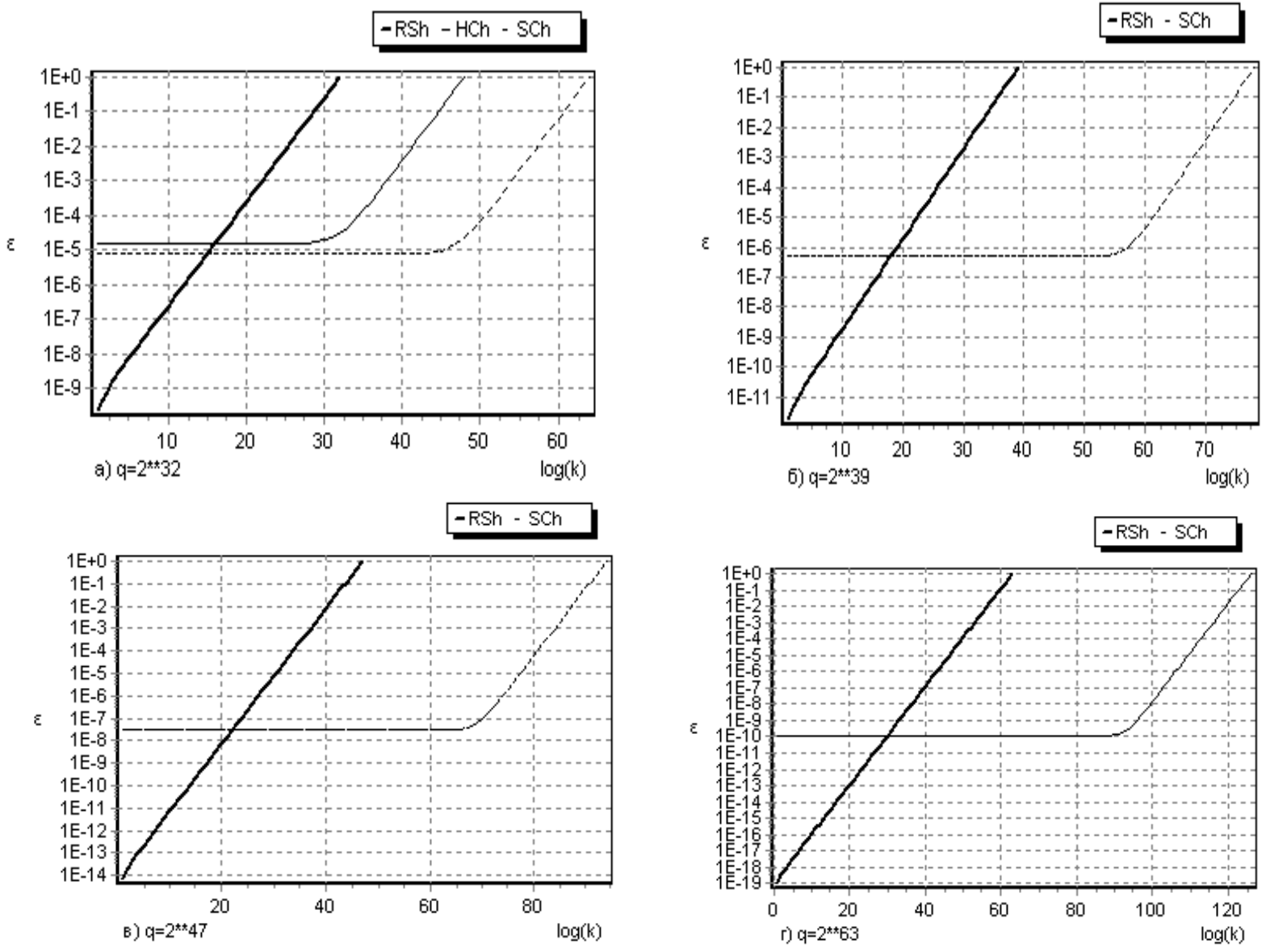


Рисунок 6.11 – Графіки залежності асимптотичних кордонів ймовірності колізій для хешування SChq від довжини даних при різних значеннях розмірів кінцевого поля обчислення

HC кодування є конструктивним, реалізується відносно просто в тій же арифметиці, що і *RS* кодування. HCh_q хешування забезпечує меншу ймовірність колізії в порівнянні з RSh_q.

Для асимптотичної оцінки ймовірність колізії можна ввести порогове значення довжини k^*. Для універсального хешування HCh_q k^*_{HC} визначається значенням $k^*_{HC} = \sqrt{q} + 1$ і ймовірність колізії при $k = k^*_{HC}$ – $\varepsilon(k^*_{HC}) = 1/\sqrt{q}$ [279]. Порогове значення k^* показує, при якій довжині даних ймовірність колізії HCh_q хешування менше в порівнянні з RSh_q хешуванням.

Серед довгих кодів по алгебраїчним кривим практичне значення мають коди Судзукі *SC*. Коди Судзукі визначені над полем GF_q, $q = p^{2f+1}$ і мають параметри $\left[q^2, k, q^2 - k + 1 - g\right]$, где $g = q\sqrt{q}/\sqrt{2}$ випливає з роду кривої [280-283]. Застосування кодів призводить до універсального хешування SCh_q з параметрами $\frac{1}{q}\left[\frac{(k-1)}{q} + \sqrt{\frac{q}{2}}\right] - U(q^2, q^k, q)$ [284]. Графіки залежності ймовірності колізії для SCh_q по асимптотичному кордоні від довжини даних і розміру поля обчислень представлені на рис. 6.10, 6.11, де:

ймовірність колізії для SCh_q

$$\varepsilon = \frac{1}{\sqrt{q}}\left(\frac{k-1}{q} + 1 - \frac{1}{\sqrt{q}}\right) \text{ при } 1 \le k \le q^2;$$

ймовірність колізії для RSh_q

$$\varepsilon = \frac{k-1}{q} \text{ при } 1 \le k \le q.$$

Аналіз графіків показує, що SCh_q забезпечує збільшення обсягу хешируємих даних в q раз в порівнянні з RSh_q. Для $q = 2^{32}$ обсяг хешируємих даних може досягати 2^{54} 32- розрядних слів, що перекриває весь діапазон практичних застосувань. При цьому ймовірність колізії обмежується

значенням $10^{-3}\left(2^{-10}\right)$. Зменшення ймовірності колізії можливо шляхом збільшення q. Так, при $q=2^{64}$ ймовірність колізії зменшується до асимптотичного кордону не менше ніж на сім порядків до 10^{-10}. Це значно краще, ніж при RSh_q і HCh_q.

Порогове значення k_{SC}^{*} при хешуванні повідомлень в схемі SCh_q визначається значенням $k_{SC}^{*}=\frac{\sqrt{q}}{2}+\frac{\sqrt{q}}{2(q-1)}+1$. Ймовірність колізії при $k=k_{SC}^{*}$ – $\varepsilon\left(k_{SC}^{*}\right)=\frac{1}{2\sqrt{q}}+\frac{1}{2\sqrt{q}(q-1)}$. Порогове значення k_{SC}^{*} для SCh_q практично збігається з граничним значенням k_{HC}^{*} для HCh_q. Універсальне хешування SCh_q по асимптотичному кордоні ймовірності колізії також програє на малих довжинах хешування RSh_q.

Основні результати за схемами хеширування за кодовою конструкцією представлені в таблиці 6.7.

Таблиця 6.7 - Основні параметри схем хешування по алгебраїчним кодам

№	Схема хешування	Асимптотичні оцінки для ймовірності колізії ε	Поле обчислень	Довжина хешируємих даних
1	RSh_q	$\frac{k-1}{q}$	$GF_q,\ q=p^m$	$1\le k\le q$
2	HCh_q	$\frac{1}{\sqrt{q}}\left(\frac{k-1}{q}+1-\frac{1}{\sqrt{q}}\right)$	$GF_q,\ q=p^2$	$1\le k\le q\sqrt{q}$
3	SCh_q	$\frac{1}{q}\left[\frac{k-1}{q}+\sqrt{\frac{q}{2}}\right]$	$GF_q,$ $q=p^{2f+1}$	$1\le k\le q^2$

Аналіз показує, що найбільш прийнятними значеннями q для хешування даних, які лежать в діапазоні розрядності сучасних процесорів, є 2^{32} і 2^{64} [259]. Імовірність колізії зростає лінійно зі зростанням обсягу даних.

Для ймовірності колізії в діапазоні значень 10^{-3} (2^{-10}) ÷ 10^{-9} (2^{-30}) розмір хешируємих даних повинен лежати в діапазоні $2^3 \div 2^{22}$ 32-розрядних слів.

Залежності ймовірності колізій для універсального хешування на PC кодах, кодах Ерміта і Судзукі від довжини хешируємого повідомлення представлені на рис. 6.12 [280].

На рис. 6.12 представлені абсолютно кращі результати для універсального хешування по алгеброгеометричним кодам. Алгеброгеометричні коди Ерміта і Судзукі забезпечують меншу ймовірність колізії універсального хешування в порівнянні з алгеброю PC кодом.

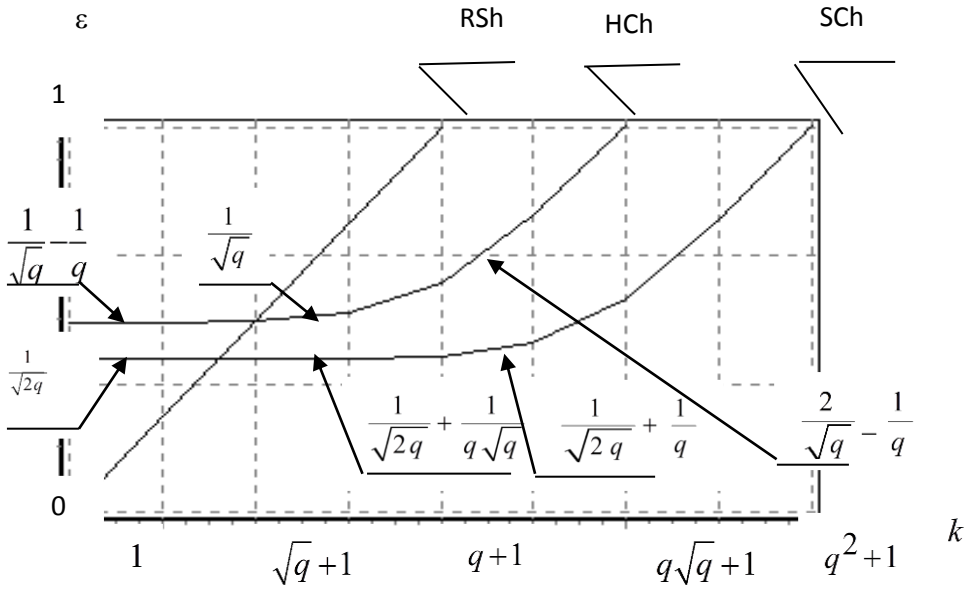


Рисунок 6.12 – Залежності ймовірності колізій універсального хешування з PC кодами, кодами Ерміта і Судзукі від довжини повідомлення

Для великих значень k це видно прямо з графіків. Для малих k програш по ймовірності колізії PC коду (RSh) визначається неточною оцінкою кодової відстані коду Ерміта (HCh) і Судзукі (SCh).

Оцінки ймовірності колізії для інтернету низької кодування не є точними, так як параметри алгеброгеометричних кодів точно визначаються при розмірності $k > g - 1$.

Універсальне хешування за раціональним функціям алгебраїчних кривих знімає обмеження на точність оцінок параметрів хешування за параметрами алгеброгеометричних кодів.

6.2.4 Універсальне хешування за раціональним функціям алгебраїчних кривих

6.2.4.1 Визначення універсального хешування по алгебраїчним кривим

Універсальне хешування за раціональним функціям алгебраїчних кривих визначається властивостями лінійного базисного простору, асоційованого з функціональним полем кривої. Властивості лінійного векторного простору по базису раціональних функцій випливають з теореми Рімана - Роха.

Нехай $D := P_1 + \ldots + P_n$ и $G := \rho_k P_\infty$, де $P_1, \ldots, P_n$ – раціональні точки кривої χ і P_∞ – точка на нескінченності. Нехай порядки полюсів ρ_i раціональних функцій $f_i \in F_q(\chi)$ впорядковані і менше ρ_k; дивізори $G + \mathrm{div}_\infty(f_i) \succ 0$ є ефективними (мають позитивні коефіцієнти по області визначення) і по теоремі Рімана - Роха раціональні функції f_i утворюють векторний простір, асоційоване з G. За слідству 6.1 значення раціональних функцій $f_i(P_j)$ в точках кривої визначають рядки породжує матриці коду $C_{D, \rho_k P_\infty}$ розмірності, рівний числу раціональних функцій k, і кодовою відстанню $d \geq N - \rho_k$ по лемі 6.1. По властивості універсального хешування на основі алгебраїчного кодування (2.3) отримаємо, що хеш-функція $h_{P_j}(m)$ визначає універсальний хеш-клас $\varepsilon - U(N, q^k, q)$ з ймовірністю колізії $P_{\mathrm{col}} \leq \varepsilon = \rho_k / N$, де N – число точок алгебраїчної кривої, q^k – обсяг простору повідомлень, q – обсяг простору хеш-кодів.

Метод універсального хешування по алгебраїчним кривим визначається як скалярний добуток по раціональним функціям алгебраїчних кривих:

$$h_{P_j}(m) = \sum_{i=1}^{k} f_i(P_j) m_i ,$$

що випливає з конструкції алгеброгеометричного коду і визначається наступною послідовністю дій:

- визначити проектне різноманіття - алгебраїчну криву і її точки;

- побудувати лінійний векторний простір для функціонального поля кривій алгебри;

- задати хеш-функцію як скалярний добуток слів даних і значень раціональних функцій в точці кривої.

Параметри універсального хеш-класу $\varepsilon - U(N, q^k, q)$ на основі хешування по раціональним функціям визначаються властивостями кривій алгебри або асоційованим з цієї кривої проективним різноманіттям.

Зауваження 6.11

1. Параметри універсального хеш-класу $\varepsilon - U(N, q^k, q)$ на основі хешування по раціональним функціям визначаються властивостями алгебраїчної кривої. Підгрупа Вейерштрасса $H(P_\infty) = \{\rho_0 = 0 < \rho_1 < \ldots\}$ визначається полюсами раціональних функцій в особливій точці кривої; раціональні функції, впорядковані за значеннями полюсів, утворюють векторний лінійний простір розмірності

$$\dim(L(G)) = v_\ell := \#\{(i, j) \in N^2 : \rho_i + \rho_j = \rho_{\ell+1}\}.$$

2. Ключовий параметр хеш-функції $h_{P_j}(m)$ визначається обчисленням в точці алгебраїчної кривої.

3. Для побудови алгоритму обчислень необхідно визначити проектне різноманіття алгебраїчної кривої (її точки), побудувати лінійне векторне простір для функціонального поля алгебраїчної кривої, задати хеш-функцію

як скалярний твір слів даних і значень раціональних функцій в точці кривої, обраної по ключу.

Проблематика побудови схем універсального хешування по раціональним функціям алгебраїчних кривих полягає у виборі алгебраїчних кривих з необхідними параметрами.

6.2.4.2 Найкращі алгебраїчні криві для універсального хешування

Під кривою будемо розглядати проектну, геометрично нерозкладну і несингулярну алгебраїчну криву, визначену над кінцевим полем F_q з q елементами.

Точка P кривої C називається несингулярною, якщо існує дотична лінія до кривої в точці P. Наприклад, якщо $P=(a,b)\in F_q\times F_q$ є точкою плоскої кривої, асоційованої з поліномом $f(X,Y)\in F_q[X,Y]$, тоді точка P називається несингулярною, якщо

$$f_X(a,b)\neq 0 \text{ або } f_Y(a,b)\neq 0,$$

де f_X і f_Y – часні похідні.

Крива C називається несингулярною, якщо кожна точка $P\in C$ є несингулярною.

Крива $\tilde{C}$ є проективною моделлю афінної кривої C, асоційованої з поліномом $f(X,Y)$ ступеня $d:=\deg f(X,Y)$, має вигляд

$$F(X,Y,Z)=Z^d f(X/Z,Y/Z) \text{ і } \tilde{C}:=\left\{(a:b:c)\in P^2(F_q)\middle| F(a,b,c)=0\right\}.$$

Якщо проектна плоска крива є несингулярною, тоді для роду кривої справедлива оцінка [285]:

$$g(\tilde{C})\leq (d-1)(d-2)/2.$$

Точка $(a:b:c)$ кривої $\tilde{C}$ є раціональною точкою, якщо $a,b,c\in F_q$. Точка $(a:b:c)$ кривої $\tilde{C}$ є раціональною точкою, якщо P_∞, $c=0$.

Нехай $N_q(g)$ позначає максимальне число F_q раціональних точок, які крива роду g може мати. Крива C роду g є оптимальною (максимальної) над F_q, якщо її число F_q раціональних точок $\#C(F_q)$ дорівнює $N_q(g)$. Головний результат для теорії визначається теоремою Хассе - Вейля.

Теорема 6.1 [285]. Нехай C – проектна і несингулярна, абсолютно нерозкладна крива, визначена над кінцевим полем F_q з q елементами. Тоді число F_q раціональних точок кривої визначається нерівністю

$$N_q(g) \le 1 + q + 2\sqrt{q}g(C) \qquad (6.20)$$

Максимальні криві над F_q – це криві, число F_q раціональних точок які задовольняють кордону Хассе - Вейля (2.9).

Для максимальних кривих над кінцевим полем досягається максимальне відношення числа точок кривої до роду, що прямо пов'язується з імовірністю колізії в схемі з універсальним хешуванням, так як $P_{\text{êîë}} \le \varepsilon = \rho_k / N$.

Існують три чудові сімейства таких кривих, які зв'язуються з Делігне - Лустіга (Deligne - Lusztig) різноманіттям розмірності $\dim = 1$. Крива Делігне - Лустіга асоціюється з проективної спеціальної лінійної групою (криві Ерміта), з групою Судзукі (Suzuki) $Sz(q)$ (криві Судзукі) і Ри (Ree) групою $R(q)$ [286].

Не існує максимальних кривих над полем F_q роду більше, ніж $g > \sqrt{q}\left(\sqrt{q} - 1\right)/2$. Класифікація максимальних плоских кривих представлена в [266]. Основний результат за максимальними кривим представлений теоремою [287].

Теорема 6.2 Рід g максимальної кривої над F_{l^2} відповідає значенням $g \le g_3 = \left\lfloor \left(l^2 - l + 4\right)/6 \right\rfloor$ або $g = g_2 = (l-1)^2/4$, або $g = g_1 = l(l-1)/2$.

Випадок, коли $g = g_1$, виконується тільки для кривої Ерміта $y^l + y = x^{l+1}$.

За класифікацією максимальних плоских кривих крива Ерміта є кривої максимального роду $g = g_1 < q$.

Зауваження 6.12

1. В роботі [288] показано, що $N_{l^2}(g) < 1 + l^2 + 2gl$ для $g_2 < g < g_1$, тобто не існує максимальних кривих, рід яких приймає значення $g_2 < g < g_1$.

2. Відомий результат, який визначає, що якщо крива покривається максимальною кривою, то вона також є максимальною, що дозволяє побудувати сімейство максимальних кривих [289].

Максимальні криві C_2 роду g_2, які відносяться до класу кривих, що покриваються кривої Ерміта, розглянуті в [290-295] і мають вигляд:

а) $C_{2a} \Rightarrow y^l + y = x^{(l+1)/2}$;

в) $C_{2b} \Rightarrow \sum_{i=1}^{t} y^{l/2^i} = x^{l+1}$;

с) $C_{2c} \Rightarrow \sum_{i=0}^{t-1} y^{2^i} = x^{l+1},\ l = 2^t > 2$;

Максимальні криві C_3 роду $g = g_3$, які відносяться до класу кривих покриваються кривої Ерміта, розглянуті в [290-293] і мають вигляд:

а) $C_{3a} \Rightarrow x^{(l+1)/3} + x^{2(l+1)/3} + y^{l+1} = 0$, якщо $l \equiv 2 (\mathrm{mod}\, 3)$;

в) $C_{3b} \Rightarrow \omega x^{(l-1)/3} - yx^{2(l-1)/3} + y^l = 0$, якщо $l \equiv 1 (\mathrm{mod}\, 3)$, де $\omega \in F_{l^2}, \omega^{l-1} = -1$;

с) $C_{3c} \Rightarrow y^l + y = \left(\sum_i^t x^{l/3^i} \right)^2$, якщо $l = 3^t$;

d) $C_{3d} \Rightarrow x^{2(l+1)/3} y^{(l+1)/3} + y^{2(l+1)/3} + x^{(l+1)/3} = 0$, якщо $l \equiv 2 (\mathrm{mod}\, 3)$.

Максимальні криві C_4 роду $g < g_3$, розглянуті в [290-293, 295]:

а) $C_{4a} \Rightarrow y^l + y = x^{(l+1)/3},\ g = (l^2 - 3l + 2)/6 < g_3$;

в) $C_{4b} \Rightarrow \sum_{i=0}^{t-1} y^{3^i} = \omega x^{l+1},\ l = 3^t,\ \omega \in F_{l^2},\ \omega^{l-1} = -1,\ g = l(l-3)/6 < g_3$;

с) $C_{4c} \Rightarrow y^l + y = x^m$, де m – дільник $l+1,\ g = (m-1)(l-1)/2$;

d) $C_{4d} \Rightarrow \sum_{i=1}^{t} y^{l/p^i} + \omega x^{l+1} = 0$, $\omega \in F_{l^2}$, $\omega^{l-1} = -1$, $g = l(l-p)/2p$.

Зауваження 6.13

1. Крива C_{3a} є окремим випадком кривої $x^{(l+1)/d} + x^{2(l+1)/d} + y^{l+1} = 0$ роду $g = (l+1)(l-2)/2d + 1$ [268,269,272].

2. Крива $C_{3b}\dfrac{\delta y}{\delta x}$ – окремий випадок кривої $\omega x^{(l-1)/d} - yx^{2(l-1)/d} + y^l = 0$ роду $g = l(l-1)/2d$ [273-275].

3. Крива C_{3c} є окремим випадком кривої $y^l + y = \left(\sum_i^t x^{l/p^i}\right)^2$, $l = p^t$ роду $g = l(l-1)/2p$ [268,269,272].

Оцінки полюсів раціональних функцій, розмірність функціональних полів за максимальними кривим в квадратичному полі роду g_1, g_2 і g_3 представлені в таблиці 6.8.

Таблиця 6.8 - Максимальні криві над F_q, $q = l^2$

Рівняння кривої $C(F_{l^2})$	Значення роду кривої	Полюса раціональних функцій	Значення підгрупи Вейерштрасса
$y^l + y = x^{l+1}$	$g_1 = \dfrac{l(l-1)}{2}$	$(x)_\infty = l$, $(y)_\infty = l+1$	$\langle l, l+1\rangle$
$y^l + y = x^{(l+1)/2}$, l непарне	$g_2 = \dfrac{(l-1)^2}{4}$	$(x)_\infty = l$, $(y)_\infty = (l+1)/2$	$\langle (l+1)/2, l\rangle$
$\sum_{i=1}^{t} y^{l/2^i} = x^{l+1}$, $l = 2^t$	$g_2' = \dfrac{l(l-2)}{4}$	$(x)_\infty = l/2$, $(y)_\infty = l+1$	$\langle l/2, l+1\rangle$
$y^l + y = x^{(l+1)/3}$, $l \equiv 2(\mathrm{mod}3)$	$g_3' = \dfrac{(l^2-3l+2)}{6}$	$(x)_\infty = (l+1)/3$, $(y)_\infty = l$	$\langle (l+1)/3, l\rangle$
$\sum_{i=0}^{t-1} y^{3^i} = \omega x^{l+1}$, $l = 3^t$, $\omega \in F_{l^2}$, $\omega^{l-1} = -1$	$g_3'' = \dfrac{l(l-3)}{6}$	$(x)_\infty = l/3$, $(y)_\infty = l+1$	$\langle l/3, l+1\rangle$
$x^{(l+1)/3} + x^{2(l+1)/3} + y^{l+1} = 0$, $l \equiv 2(\mathrm{mod}3)$	$g_3 = \dfrac{(l^2-l+4)}{6}$	$(x)_\infty = l+1$, $(y)_\infty = (l+1)/3$, $(v)_\infty = l$	$\langle 2(l+1)/3, l, l+1\rangle$

$\omega x^{(l-1)/3} - yx^{2(l-1)/3} + y^l = 0$ $\omega \in F_{l^2},\ \omega^{l-1} = -1,$ $l \equiv 1 (\mathrm{mod}\, 3)$	$g_3''' = \frac{l(l-1)}{6}$	$(x)_\infty = l,$ $(y)_\infty = (l-1)/3,$ $(v)_\infty = l+1$	$\langle (2l-1)/3, l, l+1 \rangle$
$y^l + y = \left(\sum_i^t x^{l/3^i} \right)^2,$ $l = 3^t$	$g_3''' = \frac{l(l-1)}{6},$	$(x)_\infty = l,$ $(y)_\infty = 2l/3,$ $(v)_\infty = l+1$	$\langle 2l/3, l, l+1 \rangle$
$x^{2(l+1)/3} y^{(l+1)/3} +$ $+ y^{2(l+1)/3} + x^{(l+1)/3} = 0$	$g_3 = \frac{\left(l^2 - l + 4\right)}{6}$		$\langle (2l+1)/3, l, l+1 \rangle$

Зауваження 6.14

1. Алгебраїчні криві $y^l + y = x^{l+1}$, $y^l + y = x^{(l+1)/2}$ і $\sum_{i=1}^{t} y^{l/2^t} = x^{l+1}, l = 2^t$ є максимальними кривими першого та другого роду, мають підгрупу Вейерштрасса $H(P_\infty) = \langle \rho_1, \rho_2 \rangle$ розмірності $\dim = 2$, функціональне поле визначається функціями виду $\{x^i \cdot y^j\}$.

2. Алгебраїчні криві $y^l + y = x^{(l+1)/3}, l \equiv 2 (\mathrm{mod}\, 3)$ и $\sum_{i=0}^{t-1} y^{3^t} = \omega x^{l+1}, l = 3^t, \omega \in F_{l^2}, \omega^{l-1} = -1$ – максимальні криві третього роду, мають підгрупу Вейерштрасса $H(P_\infty) = \langle \rho_1, \rho_2 \rangle$ розмірності $\dim = 2$, функціональне поле визначається функціями виду $\{x^i \cdot y^j\}$.

3. Максимальні криві виду:

– $x^{(l+1)/3} + x^{2(l+1)/3} + y^{l+1} = 0, l \equiv 2 (\mathrm{mod}\, 3),$

– $\omega x^{(l-1)/3} - yx^{2(l-1)/3} + y^l = 0, \omega \in F_{l^2}, \omega^{l-1} = -1, l \equiv 2 (\mathrm{mod}\, 3),$

– $y^l + y = \left(\sum_i^t x^{l/3^i} \right)^2, l = 3^t$

мають підгрупу Вейерштрасса $H(P_\infty)$ розмірності $\dim = 3$, функціональне поле визначається раціональними функціями виду $\{x^i \cdot y^j \cdot v^t\}$.

6.2.4.3 Колізійні властивості універсального хешування по алгебраїчним кривим

Визначення хеш-функції випливає з базису простору $L(\rho_k P_\infty)$ раціональних функцій алгебраїчних кривих. Оцінки для ймовірності колізії пов'язані зі значенням ρ_k полюса підгрупи Вейерштрасса $H(P_\infty)$, що, в свою чергу, визначається показниками ступенів раціональних функцій функціонального поля і залежить від числа k слів даних.

Максимальні алгебраїчні криві першого і другого роду мають підгрупу Вейерштрасса $H(P_\infty) = \langle \rho_1, \rho_2 \rangle$ розмірності $\dim = 2$, і функціональне поле визначається функціями виду $\{x^i \cdot y^j\}$.

Хеш-функція $h_{x,y}(m) \in F_{q^2}$ для повідомлення m по раціональним функціям в точці x, y максимальних кривих, визначається виразом

$$h_{x,y}(m) = \sum_{i \geq 0,\ 0 \leq j \leq \rho_1,\ i \cdot \rho_1 + j \cdot \rho_2 \leq \rho_k} m_{i,j} \cdot x^i \cdot y^j, \qquad (6.21)$$

де ρ_k – полюс підгрупи Вейерштрасса $H(P_\infty)$;

$m_{i,j} \in F_{q^2}$ – слова повідомлення m.

Хеш-функція $h_{P_j}(m)$ визначає універсальний хеш-клас $\varepsilon - U(N, q^k, q)$, де $\varepsilon = \rho_k / N$ – ймовірність колізії. Число F_{q^2} раціональних точок максимальної кривої лежить на кордоні Хассе - Вейля і визначається рівністю

$$N_{q^2}(g) = 1 + q^2 + 2qg. \qquad (6.22)$$

Значення роду визначає число точок $N_{q^2}(g)$ і прямо випливає з показників координатних змінних рівняння кривої. Значення ступенів змінних визначають порядки полюсів $\mathrm{div}_\infty(x) = \rho_1$, $\mathrm{div}_\infty(y) = \rho_2$, а аддитивна група полюсів - підгрупу Вейерштрасса $H(P_\infty)$. Таким чином, найкращий результат хешування - найменша верхня межа ймовірності колізії,

досягається на максимальних кривих найбільшого роду. Оцінки ймовірності колізії для кривих першого, другого і третього роду представлені в таблиці 6.9.

Зауваження 6.15

1. Найкраща асимптотична оцінка ймовірності колізії $\varepsilon_{q\to\infty}(k)$ визначається хешуванням по кривій Ерміта - максимальної кривої найбільшого роду за класифікацією.

2. Програш по ймовірності колізії універсальних схем хешування по плоским максимальним кривим другого і третього роду несуттєвий. Асимптотична оцінка трохи гірше в $\sqrt{2}\div\sqrt{6}$ раз у порівнянні з кривою Ерміта.

Точні значення ймовірності колізії універсального хешування по кривим Ерміта HCh_q над F_q від довжини даних представлені на рис. 6.13, 6.14. Для порівняння тут наведено залежності хешування по проективній прямій RSh_q і залежності ймовірності колізії для хешування на основі кодів Ерміта, яка не враховує розподіл полюсів раціональних функцій при малих значеннях k довжин даних $HCh_{\grave{a}\tilde{n}}$.

Таблиця 6.9 – Колізійні оцінки універсального хешування за максимальними кривим над F_{q^2}

Рівняння кривої	Параметри універсального хешування $\varepsilon - U\left(N, q^{2k}, q^2\right)$	Оцінка ймовірності колізії ε, $k < g$	Асимптотична оцінка $\varepsilon_{q\to\infty}(k)$
$y^q + y = x^{q+1}$ F_{q^2}	$U\left(q^3, q^{2k}, q^2\right)$	$k/q^3 + s/q^2 -$ $-s(s-1)/(2q^3)$ $s = \left\lvert (2k+1/4)^{1/2} - 1/2 \right\rvert$	$\sqrt{2}k^{1/2}/q^2$
$y^q + y = x^d$ F_{q^2} $d \mid q+1$	$U\left(q^2 + (d-1)\times\right.$ $\left.\times(q-1)q, q^{2k}, q^2\right)$	$\dfrac{(iq+jd)}{\left(q^2+(d-1)(q-1)q\right)}$ $s = \left\lvert \left(\dfrac{2k}{m}+\dfrac{1}{4}\right)^{1/2} - \dfrac{1}{2} \right\rvert$ $t = \left[k - m(s-1)s/2/s\right]$ $m = (q+1)/d$ $ind = 0, -1$	$\sqrt{2(q+1)/d}$ $k^{1/2}/q^2$
$\sum_{i=1}^{t} y^{q/p^i} + \omega x^{q+1} = 0$ F_{q^2}, $q = p^t$ $\omega^{q-1} = -1$	$U\left(q^3/p, q^{2k}, q^2\right)$	$\left(i(q+1)p + jq\right)/q^3$ $s = \left\lvert \left(\dfrac{2k}{m}+\dfrac{1}{4}\right)^{1/2} - \dfrac{1}{2} \right\rvert$ $t = \left[\dfrac{k - p(s-1)s/2}{2}\right]$ $ind = 0, -1$	$\sqrt{2p}k^{1/2}/q^2$
$x^{\frac{q+1}{d}} + x^{\frac{2(q+1)}{d}} + y^{q+1} = 0$ F_{q^2}, $q = 2 \pmod 3$ $x^{\frac{q+1}{3}} + x^{\frac{2(q+1)}{3}} + y^{q+1} = 0$	$U\left(\dfrac{q^3+2q^2+4q+3}{3}, q^{2k}, q^2\right)$	$\dfrac{3iq + 3j(q+1) + t\cdot 2(q+1)}{q^3+2q^2+4q+3}$ $s = \left\lvert \left(\dfrac{2k}{3}+\dfrac{1}{4}\right)^{1/2} - \dfrac{1}{2} \right\rvert$	$\sqrt{6}k^{1/2}/q^2$
$\omega x^{\frac{q-1}{d}} - yx^{\frac{2(q-1)}{d}} + y^q = 0$ F_{q^2}, $\omega^{q-1} = -1$ $q = 1 \pmod 3$ $\alpha^{(q+1)/2}x^{(q-1)/3} +$ $+\alpha^{(q^2-1)/2}x^{2(q-1)/3}y + y^q = 0$	$U\left(\dfrac{q^3+2q^2-q-2}{3}, q^{2k}, q^2\right)$	$\dfrac{3iq + 3j(q+1) + t(2q+1)}{q^3+2q^2-q-2}$ $s = \left\lvert \left(\dfrac{2k}{3}+\dfrac{1}{4}\right)^{1/2} - \dfrac{1}{2} \right\rvert$	$\sqrt{6}k^{1/2}/q^2$

Графіки ймовірності колізії для HCh_q, $HCh_{àñ}$ і RSh_q хешування обчислені по наступних співвідношеннях:

- асимптотичий кордон ймовірності колізії для $HCh_{àñ}$

$$\varepsilon = \frac{1}{\sqrt{q}}\left(\frac{k-1}{q}+1-\frac{1}{\sqrt{q}}\right) \text{ при } 1 \le k \le q\sqrt{q};$$

– ймовірність колізії для RSh_q

$$\varepsilon = \frac{k-1}{q} \text{ при } 1 \le k \le q.$$

Уточнена межа ймовірності колізії для $HCh_{\text{ôí÷}}$ взята з таблиці 6.9 для кривої $y^q + y = x^{q+1}$.

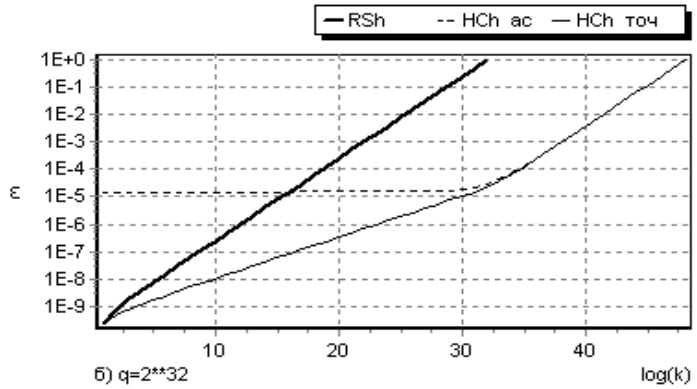


Рисунок 6.13 – Залежності ймовірності колізії для *Hchq* хешування в кінцевому полі F_q, $q = 2^{32}$ від довжини даних

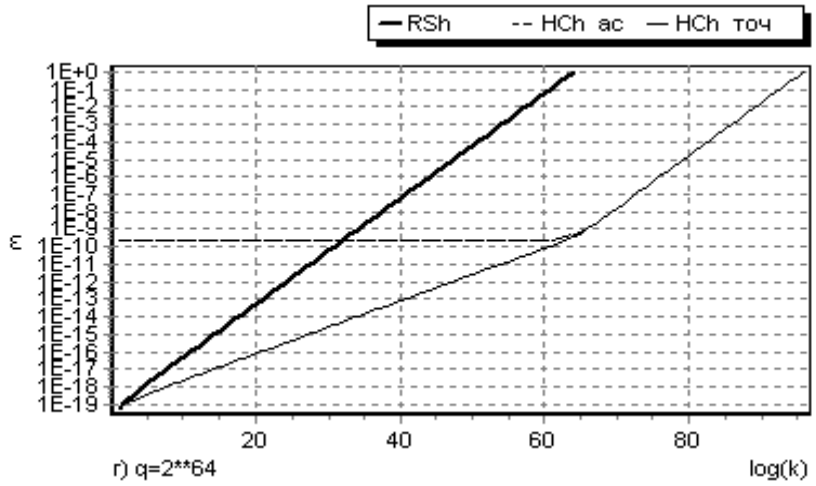


Рисунок 6.14 – Залежності ймовірності колізії для *Hchq* хешування в кінцевому полі F_q, $q = 2^{64}$ від довжини даних

Аналіз кордону для ймовірності колізії показує, що змішування HCh_q має перевагу в порівнянні з RSh_q при всіх значеннях довжини даних на відміну від асимптотичного кордону для хешування на основі кодового уявлення. При обчисленні хеш-кодів в поле однакової розмірності при фіксованій імовірності колізії обсяг даних в схемі хешування по кривій Ерміта HCh_q перевищує допустимий обсяг даних в порівнянні з RSh_q хешуванням в ступені 2 раз.

Висновки за розділом 6.2

1. Застосування універсального хешування для побудови доказово стійкою автентифікації вимагає, щоб ключовий простір було не менше простору повідомлень. Це обмеження знімається в конструкціях, де застосовуються алгебраїчні кодові конструкції.

2. Основний результат універсального хешування по алгебраїчним кодами полягає в тому, що ймовірність колізії визначається відношенням кодового відстані до довжини коду. Коди над великим алфавітом є кращими, тому що забезпечують в універсальних схемах хешування при фіксованій довжині повідомлення і значності коду потенційно менше значення ймовірності колізії.

3. Теорія побудови масивів строго універсальних автентифікаторів визначається ортогональними масивами. Основний результат строго універсального хешування полягає в тому, що воно реалізується за умови, коли розмір ключа не менше твори розмірностей простору повідомлень і хеш-кодів. Застосування слабо заміщених масивів для побудови майже строго універсальних хеш-функцій знімає обмеження на розмір ключів, але при цьому збільшується ймовірність колізії.

4. Проблематика практичної реалізації універсального хешування на основі скалярного твори по раціональним функціям алгебраїчних кривих

визначається складністю побудови точок алгебраїчних кривих за ключовими даними. Обчислювальні витрати на змішування залежать від розмірності функціонального поля раціональних функцій. Основне протиріччя універсального хешування по алгебраїчним кривим полягає в тому, що для забезпечення гарантованої ймовірності обману на нижньому рівні необхідно побудувати обчислення по раціональним функціям алгебраїчних кривих з якомога меншим відношенням значення максимального полюса раціональних функцій до числа точок кривої для фіксованої довжини даних. Застосування максимальних кривих великого роду призводить до збільшення розмірності функціонального поля, асоційованого з кривою, і зростання складності обчислень.

5. Завдання побудови універсального хешування по раціональним функціям алгебраїчних кривих полягає у виборі алгебраїчних кривих, обчисленні їх алгеброгеометричних параметрів, розробці методів, алгоритмів і оцінок універсального хешування. Верхні оцінки ймовірності колізії універсального хешування по раціональним функціям алгебраїчних кривих показують, що найкращі результати досягаються за кривими з великим числом точок.

Криві, асоційовані з групою Судзукі $Sz(q)$ (криві Судзукі), є неплоскими і мають істотно більшу кількість точок [264].

Актуальним завданням є оцінка властивостей групи Судзукі, алгебраїчної кривої, асоційованої з групою Судзукі, побудова її функціонального поля і універсального хеш-класу по кривій.

6.3 Універсальне хешування по кривій Судзукі

Проблематика практичної реалізації універсального хешування на основі скалярного добутку по раціональним функціям алгебраїчних кривих визначається складністю побудови точок алгебраїчних кривих за ключовими

даними. Обчислювальні витрати на змішування залежать від розмірності функціонального поля раціональних функцій. Одне із завдань універсального хешування по алгебраїчним кривим полягає в тому, що необхідно побудувати обчислення по раціональним функціям алгебраїчних кривих з якомога меншим відношенням значення максимального полюса раціональних функцій до числа точок кривої для фіксованої довжини даних для забезпечення гарантованої ймовірності обману на нижньому рівні. Застосування максимальних кривих великого роду призводить до збільшення розмірності функціонального поля, асоційованого з кривою, і зростання складності обчислень.

Завданням розділу є оцінка властивостей групи Судзукі, алгебраїчної кривої, асоційованої з групою Судзукі, побудова функціонального поля кривої Судзукі, універсального хешування по раціональним функціям і оцінка параметрів.

6.3.1 Криві Делігне - Лустіга, асоційовані з групою Судзукі

Крива максимального роду над квадратичним полем F_q, $q = l^2$ є кривою Ерміта. За класифікацією кривих Делігне - Лустіга крива Ерміта асоціюється з проективною спеціальною лінійною групою, покриває максимальні плоскі криві меншого роду в квадратичному полі і має найкращу асимптотичну оцінку:

$$A(q) = \limsup_{g \to \infty} N_q(g) \Big/ g ,$$

$$A(q) \approx 2\sqrt{q} ,$$

де $N_q(g)$ – число точок кривої роду g.

Друга і третя група кривих Делігне - Лустіга асоціюються з групою Судзукі $Sz(q)$ і групою Рі $R(q)$ [296]. Криві Судзукі і Рі широко розглянуті в [297-299]. Ці криві є оптимальними кривими в тому сенсі, що мають число

F_q раціональних точок щодо роду досить близьким до кордону Хассе - Вейля.

Головний результат по кривим Делігне - Лустіга другого типу, асоційованим з $Sz(q)$, визначається наступною теоремою.

Теорема 6.3 [294]. Для позитивного цілого s задані $q = 2q_0^2$ и $q_0 = 2^s$. Нехай X крива над F_q роду g і задовольняються наступні умови:

1) $g = q_0(q-1)$;

2) $\#X(F_q) = q^2 + 1$.

Тоді X являється F_q ізоморфною кривої Делігне - Лустіга, асоційованої з групою Судзукі $Sz(q)$.

Криву з точністю до F_q ізоморфізма, асоційовану з підгрупою $S(a,b)$ групи Судзукі $Sz(q)$, засновану на роді, числі точок і груповому F_q автоморфізмі кривої, визначили Хансен Дж. і Стічтенот Х. [297].

Крива Судзукі S являється F_q ізоморфною плоскій кривій

$$y^q - y = x^{q_0}\left(x^q - x\right), \tag{6.23}$$

де $q = 2q_0^2$ и $q_0 = 2^s$.

Рід кривої $g = q_0(q-1)$ і число F_q раціональних точок дорівнює q^2+1 [297].

Крива, асоційована з підгрупою $S(a,b)$ порядку q^2 з групи Судзукі порядку $q^2\left(q^2+1\right)(q-1)$,

$$S(a,b) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & \pi(a) & 1 & 0 \\ a^2\pi(a) + ab + \pi(b) & a\pi(a) + b & a & 1 \end{pmatrix}.$$

Крива Судзукі, розглянута Хансеном Дж. І Стічтенотом Х. має відносно свого роду максимальне число точок. Максимальне число F_q

раціональних точок кривої визначається чудовою формулою Вейля і для кривої Судзукі трохи менше кордону Хассе - Вейля. Дійсно, пряма підстановка в оцінку числа точок Хассе - Вейля роду кривої дає значення

$$N_q(g)=1+q+2g\sqrt{q}=\sqrt{2}q^2-\left(\sqrt{2}-1\right)q+1 \tag{6.24}$$

і маємо, що $q^2+1<\sqrt{2}q^2-\left(\sqrt{2}-1\right)q+1$.

Число F_{q^r} раціональних точок кривої $N_{q^r}(g)$ можна визначити на основі обчислення спеціального L полінома – енумератора дзета-функції

$$\zeta(X,t)=\exp\left(\sum N_{q^r}t^r/r\right).$$

Відомий результат Хассе - Вейля

$$\zeta(X,t)=L(X,t)/\{(1-t)(1-qt)\},$$

де

$$L(X,t)=\prod_{k=1}^{2g}(1-\alpha_k t),\ L(X,t)\in Z(t); \tag{6.25}$$

$$\alpha_j\alpha_{j+g}=q,\ j=1,\ldots,g; \tag{6.26}$$

$$|\alpha_j|=\sqrt{q},\ j=1,\ldots,g. \tag{6.27}$$

Слідуючи [300,301], якщо

$$L(X,t)=\prod_{k=1}^{2g}(1-\alpha_k t),$$

число точок кривої в розширеному полі F_{q^r} визначається виразом

$$N_{q^r}(g)=q^r+1-\sum_{k=1}^{2g}\alpha_k^r. \tag{6.28}$$

Для кривої Судзукі, як показано в [298], енумератор має вигляд

$$L(X,t)=\left(1+2q_0t+qt^2\right)^g. \tag{6.29}$$

Поліном $L(X,t)$ має $2g$ коренів, і рішення для α_k з (6.30) розбиваються на дві групи по g однакових значень

$$\alpha=q_0(-1+i)$$

і

$$\beta=\tilde{\alpha}=q_0(-1-i).$$

Для α і β легко перевіряються умови (3.11) і (3.12). Для $N_{q^r}(g)$ отримаємо результуючий вираз

$$N_{q^r}(g) = q^r + 1 - g\left(\alpha^r + \beta^r\right). \qquad (6.31)$$

Розглянемо основні випадки розширеного поля F_{q^r} і значення числа точок для кривою Судзукі. Підставляючи в (6.31) ступеня розширення $r = 1, \ldots, 4$, получимо

$$N_q(g) = q + 1 - g(\alpha + \beta) = q + 1 + 2gq_0 = q + 1 + 2q_0 q_0 (q-1) = q^2 + 1;$$

$$N_{q^2}(g) = q^2 + 1 - g\left(\alpha^2 + \beta^2\right) = q^2 + 1 - gq_0{}^2(-2i + 2i) = q^2 + 1;$$

$$N_{q^3}(g) = q^3 + 1 - g\left(\alpha^3 + \beta^3\right) = q^3 + 1 - 4gq_0{}^3 = q^3 + 1 - 4q_0{}^4(q-1) = q^2 + 1;$$

$$N_{q^4}(g) = q^4 + 1 - g\left(\alpha^4 + \beta^4\right) = q^4 + 1 + 8gq_0{}^4 = q^4 + 1 + 2q_0(q-1)q^2.$$

Зауваження 6.16

1. Крива Судзукі над полем F_q є оптимальною для кривої роду $g = q_0(q-1)$, по числу точок лежить близько до кордону Хассе - Вейля.

2. Крива Судзукі над квадратичним і кубічним полем є неоптимальною.

3. Крива Судзукі над кінцевим полем ступеня розширення 4 є максимальною. Підстановка значення роду $g = q_0(q-1)$ в вираз Хассе - Вейля дає

$$N_{q^4}(g) = q^4 + 1 + 2q_0(q-1)q^2,$$

що дорівнює числу точок кривої в F_{q^4}.

4. Більш загальний результат отриманий в [298], де показано, що для розширень $r \equiv 0 \bmod 4$ крива Судзукі є максимальною. Дійсно,

$$N_{q^{4s}}(g) = q^{4s} + 1 - g\left(\alpha^{4s} + \beta^{4s}\right) = q^{4s} + 1 + 2 \cdot 4^s gq_0{}^{4s} =$$

$$= q^{4s} + 1 + 2q_0(q-1)q^{2s}, \quad s \geq 1.$$

5. В роботі [298] розглянута крива Судзукі

$$y^q - y = x^{q_0}\left(x^q - x\right),$$

де $q_0 = 2^s$ и $q = 2q_0$.

Показано, що крива має рід $g = q_0(q-1)$, число точок $N_q(g) = q^2 + 1$ і також є F_q - ізоморфною кривої Делігне - Лустіга, асоційованої з групою Судзукі $Sz(q)$. Крива визначена над меншим в q_0 раз кінцевим полем і містить в ${q_0}^2$ меньше точок.

6. В роботі [302] отримані похідні криві по підгрупах групи Судзукі для випадку $q = 2{q_0}^2$ і $q_0 = 2^s$.

За циклічною підгрупою порядку r, $r \mid q-1$ крива рода $g = q_0(q-1)/r$ має вигляд

$$V^{(q-1)/r} f(U) = \left(1 + U^{q_0}\right)\left(U^{q-1} + V^{2(q-1)/r}\right),$$

де $f(t) = 1 + \sum_{i=0}^{s-1} t^{2^i(2q_0+1)-(q_0+1)}(1+t)^{2^i}$.

Крива по підгрупі Зінгера порядку $q + 2q_0 + 1$ має рід $g = (q + 2q_0 + 1)(q_0 - 1)/r + 1$ і визначається виразом

$$V^{(q+2q_0+1)/r} \tilde{f}(U) = U^{q+2q_0+1} + V^{2(q+2q_0+1)/r},$$

де $\tilde{f}(t) = 1 + \sum_{i=0}^{s-1} t^{2^i q_0}(1+t)^{2^i(q_0+1)+q_0} + t^{q/2}$.

Крива по підгрупі Зінгера порядку $q - 2q_0 + 1$ має рід $g = (q - 2q_0 + 1)(q_0 - 1)/r - 1$ і вигляд

$$bV^{(q-2q_0+1)/r} f(U) = \left(U^{q_0-1} + V^{2q_0-1}\right)\left(U^{q-2q_0+1} + V^{2(q-2q_0+1)/r}\right),$$

де $b = \lambda^{q_0} + \lambda^{q_0-1} + \lambda^{-q_0} + \lambda^{-(q_0-1)}$, $\lambda \in F_{q^4}$, порядку $q - 2q_0 + 1$.

7. В [302] розглянуті криві, асоційовані з підгрупами групи Судзукі порядків $2^u r$, $2r$, $2s$, $4s$, де $r \mid (q-1)$, $s \mid (q \pm 2q_0 + 1)$. Криві за даними підгрупами мають число точок, відповідних порядків підгруп, що менше числа точок кривої, асоційованої з підгрупою $S(a,b)$ порядку q^2.

6.3.2 Функциональное поле кривой Судзуки

Крива Делігне - Лустіга, асоційована з групою Судзукі, визначається повною лінійною серією $D=\left|\left(q+2q_0+1\right)P_0\right|$ розмірності $\dim=4$ і ступеня $q+2q_0+1$, яка виводиться з енумератора дзета-функції [297]. Відображення кривої Судзукі на проективний простір P^4 і підгрупа Вейерштрасса $H\left(P\right)$, $P\in X\left(F_q\right)$ розглянуті в роботах [291,297]. Основні результати узагальнені в затвердженні 6.1.

Затвердження 6.1 [303-305]. F_q - раціональний морфізм кривої Судзукі в P^4 є відображення

$$\pi:=\left(1:x:y:v:w\right),$$

де x, y, v, w визначаються рівняннями [297]

$$y^q-y=x^{q_0}\left(x^q-x\right),$$

$$v:=x^{2q_0+1}+y^{2q_0},$$

$$w:=xy^{2q_0}+x^{2q+2q_0}+y^{2q},$$

і порядки полюсів

$$\mathrm{div}_\infty\left(x\right)=qP_0,\ \mathrm{div}_\infty\left(y\right)=\left(q+q_0\right)P_0,$$

$$\mathrm{div}_\infty\left(v\right)=\left(q+2q_0\right)P_0,\ \mathrm{div}_\infty\left(w\right)=\left(q+2q_0+1\right)P_0.$$

Крива Судзукі може бути представлена в P^4 безліччю точок вигляду

$$P_{(a,b)}:=(1:a:b:\ f\left(a,b\right):af\left(a,b\right)+b^2)\bigcup\pi\left(P_0\right)=\left(0:0:0:0:1\right),$$

де $a,b\in F_q$ и $f\left(a,b\right):=a^{2q_0+1}+b^{2q_0}$ [281].

Підгрупа Вейерштрасса $H\left(P\right)$, $P\in C\left(F_q\right)$ функціонального поля кривої містить підгрупу [306]

$$H\left(P\right)=\left\langle q,q+q_0,q+2q_0,q+2q_0+1\right\rangle.$$

Доказ. Покажемо підгрупу Вейерштрасса. Для цього запишемо рівняння кривої в проективних координатах:

$$Y^q Z^{q_0} - YZ^{q+q_0-1} = X^{q+q_0} - X^{q_0+1} Z^{q+q_0-1}. \qquad (6.32)$$

На кривій C існує особлива точка на нескінченності $P_0 = (0:1:0)$ кратності q_0 і раціональні точки $P_{\alpha,0} = (\alpha:0:1)$, $P_{0,\beta} = (0:\beta:1)$, де $\alpha, \beta \in F_q$.

Нехай $\aleph$ – лінія з рівнянням $X = 0$. Тоді $\aleph$ перетинає криву в точках $P_{0,\beta}$ і P_0. Число точок $P_{0,\beta}$ дорівнює q. Лінія $\aleph$ має тільки одноразові перетини в точках $P_{0,\beta}$, так як $X = 0$ не є дотичною в цих точках. По теоремі Безу кратність перетину лінії $\aleph$ з кривою C дорівнює $q + q_0$. Звідси слідує що $\aleph \cdot C = \sum_{\beta \in F_q} P_{0,\beta} + q_0 P_0$.

Розглянемо лінію $\Re$ з рівнянням $Y = 0$. $\Re$ перетинає криву C в точках $P_{\alpha,0}$ і в точці $P_{0,0} = (0:0:1)$ є дотичною кратності перетину $q_0 + 1$, відповідно $\Re \cdot C = \sum_{\alpha \in F_q,\, \alpha \neq 0} P_{\alpha,0} + (q_0+1) P_{0,0}$. Для лінії $\Im$ з рівнянням $Z = 0$ маємо перетин з кривою тільки в одній точці $P_0 = (0:1:0)$ и $\Im \cdot C = (q + q_0) P_0$.

Для раціональних функцій $x = X/Z$ и $y = Y/Z$ отримаємо наступні дільники:

$$\operatorname{div}(x) = \sum_{\beta \in F_q} P_{0,\beta} + qP_0,\ \operatorname{div}(y) = \sum_{\alpha \in F_q,\, \alpha \neq 0} P_{\alpha,0} + (q_0+1) P_{0,0} - (q + q_0) P_0,$$

відповідно $\operatorname{div}_\infty(x) = qP_0$ і $\operatorname{div}_\infty(y) = (q + q_0) P_0$ – значення полюса дільників.

Розглянемо рівняння $v := x^{2q_0+1} + y^{2q_0}$. Маємо

$$y = \left(v - x^{2q_0+1}\right)^{1/2q_0}.$$

Підставимо в $y^q - y = x^{q_0}\left(x^q - x\right)$ і після перетворень отримаємо

$$v^q - v = x^{2q_0}\left(x^q - x\right).$$

Запишемо рівняння в проективних координатах

$$V^q Z^{2q_0} - VZ^{q+2q_0-1} = X^{q+2q_0} - X^{2q_0+1} Z^{q-1}. \qquad (6.33)$$

Рівняння (3.17) так само, як і рівняння кривої (3.16), має q^2+1 число рішень в F_q.

Розглянемо лінію Ψ з рівнянням $V=0$. Ψ перетинає криву C в точках $P_{\alpha,\beta}$, $\alpha^{2q_0+1}+\beta^{2q_0}=0$ і в точці $P_{0,0}=(0:0:1)$ є дотичною кратності перетину $2q_0+1$.

Відповідно, $\Re\cdot C=\sum\limits_{\alpha,\beta\in F_q,\,\alpha^{2q_0+1}+\beta^{2q_0}=0} P_{\alpha,\beta}+(2q_0+1)P_{0,0}$.

Для лінії $\Im$ з рівнянням $Z=0$ маємо перетин з кривою $V^qZ^{2q_0}-VZ^{q+2q_0-1}=X^{q+2q_0}-X^{2q_0+1}Z^{q-1}$ тільки в одній точці $P_0=(0:1:0)$ і $\Im\cdot C=(q+2q_0)P_0$. Для раціональної функції $v=V/Z$ отримаємо дільник

$$\operatorname{div}(y)=\sum_{\alpha\in F_q,\,\alpha\neq 0} P_{\alpha,0}+(2q_0+1)P_{0,0}-(q+2q_0)P_0$$

і $\operatorname{div}_\infty(y)=(q+2q_0)P_0$ – значення полюса дільника.

Визначимо $w:=y^{2q_0}x+v^{2q_0}$. Рівняння від змінних w, y, x має вигляд

$$w^q-w=y^{2q\cdot q_0}x^q+v^{2q\cdot q_0}-y^{2q_0}x-v^{2q_0}=y^{2q_0}\left(x^q-x\right).$$

Порядок полюса функції w в точці P_0 отримаємо з використанням наступних властивостей дискретної оцінки ϑ_P для раціональних функцій.

Пропозиція.

а) $\vartheta_{P_0}(xy)=\vartheta_{P_0}(x)+\vartheta_{P_0}(y)$ [304];

б) оцінка ϑ_P раціональних функцій x, y в рівнянні $y^q+\mu y=f(x)$ над $F_q, q=p^s$ визначається виразом [304]

$$q\vartheta_{P_0}(y)=\vartheta_{P_0}\left(y^q+\mu y\right)=\vartheta_{P_0}\left(f(x)\right).$$

Обчислимо оцінку ϑ_P для w^q-w

$$\vartheta_{P_0}\left(w^q-w\right)=\vartheta_{P_0}\left(y^{2q_0}\left(x^q-x\right)\right)=\vartheta_{P_0}\left(y^{2q_0}\right)+\vartheta_{P_0}\left(x^q-x\right).$$

Так як $\vartheta_{P_0}(x)=\operatorname{div}_\infty(x)=q$ и $\vartheta_{P_0}(y)=\operatorname{div}_\infty(y)=q+q_0$, отримаємо

$$q\vartheta_{P_0}(w)=(q+q_0)2q_0+q\cdot q=q(q+2q_0+1)$$

І значення полюса дільника $\operatorname{div}_\infty(w)=q+2q_0+1$.

Звідси випливає, що підгрупа Вейерштрасса $H(P)$, $P \in C(F_q)$ функціонального поля кривої містить підгрупу

$$H(P) = \langle q, q+q_0, q+2q_0, q+2q_0+1 \rangle .$$

Число точок розриву визначає рід кривої $|G(P_0)| = \#(N \setminus H) = q_0(q-1)$ [306]. Лінійна серія $q, q+q_0, q+2q_0, q+2q_0+1$ є повною, розмірності $\dim = 4$ і визначається раціональними функціями $x, y, v, w \in F_q(C)$.

Розглянемо уявлення точок кривої Судзукі. Для $P \in C(F_q) \setminus P_0$ нехай $a := x(P), b := y(P)$ і $f(a,b) := v(a,b) = a^{2q_0+1} + b^{2q_0}$. Рівняння для $w := y^{2q_0}x + v^{2q_0}$ приводиться до вигляду $w := xy^{2q_0} + x^{2q+2q_0} + y^{2q}$. Тоді $w(a,b) := af(a,b) + b^2$.

Зауваження 6.17

1. Базис простору $L(\rho_\ell P_0)$, асоційований з кривою

$$y^q - y = x^{q_0}(x^q - x),$$

задається функціями виду

$$S := \{x^r y^t v^i w^j\},$$

де $r \le q-1$, $0 \le t \le 1$, $i \le q_0 - 1$, $j \le q_0 - 1$,

$$r \cdot q + t(q+q_0) + i(q+2q_0) + j(q+2q_0+1) \le \rho_l .$$

2. Функціональне поле максимальної кривої Судзукі над кінцевим полем F_{q^4} розглянуто в [307]. Основний результат визначається наступною теоремою.

Теорема [307]. Нехай $l \in N$, $l \le q^2 - 1$. Тоді

$$S := \left\{ \begin{array}{ll} x^a y^b v^c w^d (x^q + x)^r & aq + b(q+q_0) + c(q+2q_0) + \\ & +d(q+2q_0+1) + rq^2 \le l(q^2+1) \\ & \le a \le q-1, 0 \le b \le 1, 0 \le c \\ & \le\le q_0 - 1, 0 \le d \le q_0 - 1, 0 \le r \le l \end{array} \right\}$$

є базисом лінійного простору $L(lD)$, раціональні функції якого асоційовані з кривою Судзукі над розширеним кінцевим полем ступеня розширення 4. Дільник кривої визначається на точках

$$D = P_\infty + \sum_{\alpha,\beta \in F_q} P_{\alpha,\beta},$$

має ступінь $\deg(D) = q^2 + 1$.

Нехай $f = x^a y^b v^c w^d \left(x^q + x\right)^r$ – одна з раціональних функцій з множини S і нехай ν_∞ – її дискретна оцінка в точці P_∞. Тоді

$$\nu_\infty(f) = aq + b(q + q_0) + c(q + 2q_0) + d(q + 2q_0 + 1) + rq^2$$

і f не має інших полюсів. Раціональні функції множини $S \in L\left(l\left(q^2 + 1\right)P_\infty\right)$. Кодові конструкції, асоційовані з дільником D, визначені на точках

$$N = N_{q^4} - \sum_{\alpha,\beta \in F_q} P_{\alpha,\beta},$$

так як $x^q + x = 0$, для $x \in F_q$.

3. Нехай $q_0 = 2^s$ и $q = 2q_0$. Крива, асоційована з групою Судзукі $Sz(q)$ і дільником $D = \left|\left(q + 2q_0 + 1\right)P_0\right|$, $P_0 \in C\left(F_q\right)$, – точка на нескінченності визначається як F_q раціональний морфізм $\pi := (1 : x : y : v : w)$ в P^4 з порядками полюсів $\mathrm{div}_\infty(x) = qP_0$, $\mathrm{div}_\infty(y) = (q + q_0)P_0$, $\mathrm{div}_\infty(v) = (q + 2q_0)P_0$, $\mathrm{div}_\infty(w) = (q + 2q_0 + 1)P_0$.

6.3.3 Метод універсального хешування по раціональним функціям кривої Судзукі

В основі розробленого методу лежать відомі результати, представлені в утвердженні 3.1. Рівняння кривої в проектному просторі P^2

$$Y^q Z^{q_0} - YZ^{q+q_0-1} = X^{q+q_0} - X^{q_0+1} Z^{q+q_0-1}$$

і в афінному просторі над F_q

$$y^q - y = x^{q_0}\left(x^q - x\right),$$

де $q = 2q_0^2$ і $q_0 = 2^s$.

Рід кривої $g = q_0(q-1)$ і число F_q раціональних точок дорівнює q^2+1. Крива є максимальною і задовольняє кордону Хассе - Вейля. Точками кривої є особлива точка на нескінченності $P_0 = (0:1:0)$ кратності q_0 і раціональні точки $P_{a,b} = (a:b:1)$, де $a, b \in F_{q^2}$ і $b^q - b = a^{q_0}(a^q - a)$.

Підгрупа Вейерштрасса функціонального поля кривої містить підгрупу $H(P_\infty) = \langle q, q+q_0, q+2q_0, q+2q_0+1 \rangle$. Крива Судзукі визначається повною лінійною серією $D = |(q+2q_0+1)P_0|$ розмірності $\dim = 4$.

Базис простору $L(\rho_\ell P_0)$ задається функціями виду $\{w^j \cdot v^i \cdot y^t \cdot x^r : i(q+2q_0) + j(q+2q_0+1) + t(q+q_0) + r \cdot q \le \rho_\ell\}$, що випливає з підгрупи Вейерштрасса $H(P_0)$, представленої порядками полюсів функцій $x = X/Z$, $y = Y/Z$, $v = x^{2q_0+1} + y^{2q_0}$), $w := xy^{2q_0} + x^{2q+2q_0} + y^{2q}$.

Порядки полюсів: $\mathrm{div}_\infty(x) = qP_0$, $\mathrm{div}_\infty(y) = (q+q_0)P_0$, $\mathrm{div}_\infty(v) = (q+2q_0)P_0$, $\mathrm{div}_\infty(w) = (q+2q_0+1)P_0$.

Крива Судзукі представляється в P^4 безліччю точок вигляду $P_{(a,b)} := (1:a:b:f(a,b):af(a,b)+b^2) \cup \pi(P_0) = (0:0:0:0:1)$, де $a, b \in F_q$ і $f(a,b) := a^{2q_0+1} + b^{2q_0}$.

Зауваження 3.10

Крива Судзукі має визначення в полі характеристики 2 непарного степеня розширення.

Визначення 3.8 [303]. Хеш-функція $h_{x,y}(m) \in F_q$, $q = 2q_0^2$, $q_0 = 2^s$ для повідомлення m по раціональним функціям в точці x, y кривої $Y^q Z^{q_0} - YZ^{q+q_0-1} = X^{q+q_0} - X^{q_0+1} Z^{q+q_0-1}$ визначається виразом

$$h_{x,y}(m) = \sum m_{i,j,t,r} \cdot w^j \cdot v^i \cdot y^t \cdot x^r, \qquad (6.34)$$

де ρ_k – полюс підгрупи Вейерштрасса $H(P_\infty)$,

$m_{i,\,j,\,t,\,r} \in F_q$ – слова повідомлення m,

$i \geq 0,\ 0 \leq j \leq 2q_0 - 1,\ 0 \leq t \leq 1,\ 0 \leq r \leq q_0$,

$i\left(q + 2q_0\right) + j\left(q + 2q_0 + 1\right) + t\left(q + q_0\right) + rq \leq \rho_k$,

$x = X/Z,\ y = Y/Z,\ v = x^{2q_0+1} + y^{2q_0},\ w := xy^{2q_0} + x^{2q+2q_0} + y^{2q}$.

Приклад 3.1 Нехай задано F_{2^3}. Крива Судзукі має вигляд $y^8 - y = x^2\left(x^8 - x\right)$. Число точок кривої $N = 65$. Точки кривої в P^4 визначаються рівняннями:

$$x = a,$$

$$y = b,$$

$$v = x^5 + y^4 = a^5 + b^4,$$

$$w = x^6 + y^2 + xy^4 = a^6 + b^2 + ab^4,$$

де $b^8 - b = a^2\left(a^8 - a\right)$.

Значення полюсів дільників $\mathrm{div}_\infty\left(x\right) = 8P_\infty$ і $\mathrm{div}_\infty\left(y\right) = 10P_\infty$, $\mathrm{div}_\infty\left(v\right) = 12P_\infty$, $\mathrm{div}_\infty\left(w\right) = 13P_\infty$. Підгрупа Вейерштрасса точок нерозрива визначається значеннями полюсів $H\left(P_\infty\right) = \left\langle 8, 10, 12, 13\right\rangle$ і має вигляд $\left\{0, 8, 10, 12, 13, 16, 18, 20, 21, 22, 23, 24, 25, 26, 28, \ldots.\right\}$. Точки розриву визначаються множиною $G\left(P_\infty\right) = \left\{1, 2, 3, 4, 5, 6, 7, 9, 11, 14, 15, 17, 19, 27\right\}$, їх число $\left|G\left(P_\infty\right)\right| = 14$ і дорівнює значенню роду $g = q_0\left(q - 1\right) = 14$. Лінійна серія 8,10,12,13 є повною, визначається раціональними функціями x, y, v, w.

Таблиця 6.9 – Точки кривої $y^8 - y = x^2\left(x^8 - x\right)$ над F_{2^3}

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}
z	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
y	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x	0	0	1	α_1	α_2	α_3	α^4	α_5	α_6	0	1	α_1	α_2	α_3	α^4	α_5	α_6
v	1	0	1	α_5	α_3	α_1	α_6	α^4	α_2	1	0	$\alpha 4$	α_1	α_3	α_2	α_5	α_6
w	1	0	1	α_6	α_5	α^4	α_2	α_1	1	1	α^4	α_1	α_2	α_2		α_1	α_3
	P_{17}	P_{18}	P_{19}	P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}	P_{26}	P_{27}	P_{28}	P_{29}	P_{30}	P_{31}	P_{32}	P_{33}
z	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
y	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_2	α_2	α_2	α_2	α_2	α_2	α_2	α_2	α_3
x	0	1	α_1	α_2	α_3	α^4	α_5	α_6	0	1	α_1	α_2	α_3	α^4	α_5	α_6	0
v	α^4	α_5	1	α_6	α_2	α_3	0	α_1	α_1	α_3	α_6	1	0	α_5	α_2	α^4	α_5
w	α_2	α_3	α^4	α^4	α_3	α_6	α_2	α_6	α_5	α_1	α^4	α_1	α^4	α_1	α_5	α_6	α_6
	P_{34}	P_{35}	P_{36}	P_{37}	P_{38}	P_{39}	P_{40}	P_{41}	P_{42}	P_{43}	P_{44}	P_{45}	P_{46}	P_{47}	P_{48}	P_{49}	P_{50}
Z	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
y	α_3	α_3	α_3	α_3	α_3	α_3	α_3	α^4	α^4	α^4	α^4	α^4	α^4	α^4	α^4	α_5	α_5
x	1	α_1	α_2	α_3	α^4	α_5	α_6	0	1	α_1	α_2	α_3	α^4	α_5	α_6	0	1
v	α^4	0	α_2	α_6	α_1	1	α_3	α_2	α_6	α_3	α_5	α^4	1	α_1	0	α_6	α_2
w	α_3	α_6	α_3	1	α_1	α_1	1	α_1	α_5	α_2	α_3	α_3	α_2	α_5	α_1	α_3	α_5
	P_{51}	P_{52}	P_{53}	P_{54}	P_{55}	P_{56}	P_{57}	P_{58}	P_{59}	P_{60}	P_{61}	P_{62}	P_{63}	P_{64}			
z	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
y	α_5	α_5	α_5	α_5	α_5	α_5	α_6	α_6	α_6	α_6	α_6	α_6	α_6	α_6			
x	α_1	α_2	α_3	α^4	α_5	α_6	0	1	α_1	α_2	α_3	α^4	α_5	α_6			
v	α_1	α^4	α_5	0	α_3	1	α_3	α_1	α_2	0	1	α^4	α_6	α_5			
w	α_5	α^4	1	α_3	1	α^4	α_5	α_6	α_2	α_5	α_2	α_6	1	1			

Базисний простір кривої $y^8 - y = x^2\left(x^8 - x\right)$ визначається раціональними функціями x, y, $v = x^5 + y^4$, $w := x^6 + xy^4 + y^2$. Розподіл кратності перетину

поліномів базисного простору і $y^8 - y = x^2\left(x^8 - x\right)$ над F_{2^3} представлено в таблиці 3.2. Хеш-обчислення в кінцевому полі F_{2^3} по поліноміальному базису $L(18P_\infty)$ на кривій $y^8 - y = x^2\left(x^8 - x\right)$ дають оцінку ймовірності колізії $\varepsilon = m/N = 18/64 = 0,28$.

Дійсно, число точок кривої $N = 64$ і число співпадаючих хешів при обчисленні поліноміального базису $L(18P_\infty)$ не перевищує значення 18. Число слів даних $k = 6$. Хеш-обчислення в кінцевому полі F_{2^3} для 6 слів даних по поліноміальному базису $L(6P_\infty)$ на проективній прямій $x + y + z = 0$ дають оцінку ймовірності колізії $\varepsilon = m/N = 6/8 = 0,75$.

Таблиця 6.10 – Розподіл кратності перетину поліномів базисного простору і кривої $y^8 - y = x^2\left(x^8 - x\right)$

Базисний простір	Число випробувань	Розподіл кратності перетину (значення числа точок перетину = число дослідів)		
x, y, v, w	10000	8:=767 9:=1095	12:=2074	13:=6060
x, y, v, w, x^2	10000	8:=730 9:=138 11:=2904	12:=1479 13:=3781 14:=234	15:=717 16:=17
x, y, v, w, x^2, xy	10000	8:=959 9:=10 10:=2683 11:=1192	12:=2722 13:=577 14:=1136 15:=403	16:=259 17:=21 18:=38
x, y, v, w, x^2, xy, y	10000	8:=864 9:=75 10:=2703 11:=897 12:=2969	13:=866 14:=970 15:=307 16:=201	17:=33 18:=108 19:=4 20:=3

Зв`язок значення k з показниками i, j, t, r ступенів раціональних функцій w, v, y, x визначається лемою [303].

Лема 6.2 Нехай $k < q_0(q-1)$. Для кривої Судзукі має сенс $i = s - 1 - r - j - t$, $j = \Delta - t \cdot q_0 - 1$, $r = s - s_2 - dq_0 - t$, $t = t_1 \bmod 2$,

де $s'=\left\lfloor (3k)^{1/3} \right\rfloor$, $\Sigma = s'(s'+1)(2s'+1)/6$, $s = s' + \lfloor k/\Sigma \rfloor$, $s_1 = s - q_0 - 1$, $\Sigma_{s-1} = s(s-1)(2s-1)/6 - s_1(s_1-1)(2s_1-1)/3 - s_1(s_1-1)$, $k' = k - \Sigma_{s-1}$, $k_1 = \lceil k'/2 \rceil$, $d = \lfloor s_2/(s-t) \rfloor$, $k_2 = k_1 + s_1(s_1+1)/2$, $s_2 = \left\lceil (2k_2 + 1/4)^{1/2} - 1/2 \right\rceil$, $s_3 = s_2 - q_0 - 1$, $\Delta = k' - 2k_3$, $t_1 = \lceil \Delta/q - 1 \rceil$, $k_3 = (s_2-1)s_2/2 - (s_1-1)(s_1+1)/2 - s_3(s_3+1)/2$, $p = s_2 - (s_2 - t)d$, $\lceil \cdot \rceil$ – округлення до більшого цілого числа, $\lfloor \cdot \rfloor$ – округлення до меншого цілого числа, $[\cdot]$ – округлення до найближчого цілого числа.

Доказ. Аддитивна підгрупа Вейерштрасса $H(P_\infty) = \{\rho_0 = 0 < \rho_1 < \ldots\}$ кривої $y^q - y = x^{q_0}\left(x^q - x\right)$ визначається значеннями полюсів, $\varphi = q$, $\omega = q + q_0$, $\eta = q + 2q_0$ і $\gamma = q + 2q_0 + 1$.

Розглянемо приклад кривої $y^{32} - y = x^4\left(x^{32} - x\right)$ над полем F_{2^5}, число точок кривої $N = q^2 + 1 = 1025$ і рід $g = q_0(q-1) = 124$. Розміщення полюсів в $H(P) = \langle q, q+q_0, q+2q_0, q+2q_0+1 \rangle$ має вигляд, представлений в таблиці 3.3.

Полюса підгрупи Вейерштрасса $q_0 = 2^2$, $q = 2^5$, $q + q_0 = 36$, $q + 2q_0 = 40$, $q + 2q_0 + 1 = 41$. Полюса діляться на два шари. Відмінність другого шару від першого полягає в тому, що поряд з комбінацією полюсів 32, 40, 41 функцій x, v, w враховується полюс зі значенням 36 раціональною функцією y. Кожен шар складається з рівнів з наростаючим числом рядків. Число рівнів для $k < q_0(q-1) = 124$ буде $2q_0 - 1 = 7$.

Число полюсів на кожному рівні визначається наступним чином: нехай s – номер рівня і $s \le q_0 = 4$, тоді в першому шарі маємо $\Sigma_1 = s(s+1)/2$, в другому $\Sigma_2 = s(s-1)/2$ і результуюче число полюсів на рівні s буде $\Sigma = s(s+1)/2 + s(s-1)/2 = s^2$. Розглянемо випадок $s > q_0 = 4$. Зауважимо, що можна використовувати вираз для суми s членів арифметичної прогресії з вирахуванням відсутніх елементів ряду $1, 2, \ldots, s - q_0 - 1$. У кожному шарі

таких наборів по два, результуючих вирази для суми полюсів на рівні s буде $\Sigma = s^2 - 2s_1(s_1+1)$, де $s_1 = s - q_0 - 1$.

Сумарне число полюсів на всіх рівнях $1, 2, \ldots, s$

$$\Sigma_s = \sum_{i=1}^{s} i^2 - 2\sum_{j=1}^{s_1} j^2 - 2\sum_{j=1}^{s_1} j = \\ = s(s+1)(2s+1)/6 - s_1(s_1+1)(2s_1+1)/3 - s_1(s_1+1). \tag{6.35}$$

Розміщення полюсів ρ_k в порядку зростання в підгрупі $H(P_\infty) = \langle q = 32, q + q_0 = 36, q + 2q_0 = 40, q + 2q_0 + 1 = 41 \rangle$, з урахуванням полюсів раціональних функцій представлено в таблиці 3.4.

Значення k визначається виразом

$$k = s(s-1)(2s-1)/6 + s_2(s_2-1) - s_1(s_1-1) - s_3(s_3-1) + t_1 q_0 + j + 1. \tag{6.36}$$

Тут s – значення рівня, на якому розташовується полюс ρ_k, s_2 – номер рядка розташування полюса на рівні s, s_1 – номер рядка доповнення до повного арифметичного ряду s рядків, s_3 – номер рядка розташування полюсів, які необхідно відняти від суми повного арифметичного ряду s рядків, t_1 – число рядків розміру q_0 рівня s, j – місце розташування полюса в останньому рядку рівня s. Для обчислення значень показників i, j, t, r ступенів раціональних функцій w, v, y, x, відповідних заданому k, слід визначити рівень, на якому знаходиться ρ_k полюс, рядок розташування полюса на рівні, місце розташування ρ_k в цьому рядку і приналежність до першого або другого шару полюсів.

Неважко показати, що значення рівня, на якому знаходиться полюс ρ_k, обчислюється за формулами:

$s = \left[(3k)^{1/3} \right]$ – округлення до найближчого цілого,

$\Sigma = s'(s'+1)(2s'+1)/6$ – сума числа полюсів рівнів $1 \div s'$,

Таблиця 6.11 – Полюса підгрупи Вейерштрасса $H(P_\infty)=\langle 32, 36, 40, 41\rangle$

Полюса другого шару				Полюса першого шару				№
							ρ_0=0	1
			ρ_2=36				ρ_1=32	
						ρ_4=41	ρ_3=40	2
			ρ_6=68				ρ_5=64	
		ρ_{10}=77	ρ_9=76			ρ_8=73	ρ_7=72	
					ρ_{13}=82	ρ_{12}=81	ρ_{11}=80	3
			ρ_{15}=100				ρ_{14}=96	
		ρ_{19}=109	ρ_{18}=108			ρ_{17}=105	ρ_{16}=104	
	ρ_{25}=118	ρ_{24}=117	ρ_{23}=116		ρ_{22}=114	ρ_{21}=113	ρ_{20}=112	
				ρ_{29}=123	ρ_{28}=122	ρ_{27}=121	ρ_{26}=120	4
			ρ_{31}=132				ρ_{30}=128	
		ρ_{35}=141	ρ_{34}=140			ρ_{33}=137	ρ_{32}=136	
	ρ_{41}=150	ρ_{40}=149	ρ_{39}=148		ρ_{38}=146	ρ_{37}=145	ρ_{36}=144	
ρ_{49}=159	ρ_{48}=158	ρ_{47}=157	ρ_{46}=156	ρ_{45}=155	ρ_{44}=154	ρ_{43}=153	ρ_{42}=152	
			ρ_{54}=164	ρ_{53}=163	ρ_{52}=162	ρ_{51}=161	ρ_{50}=160	5
		ρ_{58}=173	ρ_{57}=172			ρ_{56}=169	ρ_{55}=168	
	ρ_{64}=182	ρ_{63}=181	ρ_{62}=180		ρ_{61}=178	ρ_{60}=177	ρ_{59}=176	
ρ_{72}=191	ρ_{71}=190	ρ_{70}=189	ρ_{69}=188	ρ_{68}=187	ρ_{67}=186	ρ_{66}=185	ρ_{65}=184	
ρ_{80}=199	ρ_{79}=198	ρ_{78}=197	ρ_{77}=196	ρ_{76}=195	ρ_{75}=194	ρ_{74}=193	ρ_{73}=192	
		ρ_{86}=205	ρ_{85}=204	ρ_{84}=203	ρ_{83}=202	ρ_{82}=201	ρ_{81}=200	6
	ρ_{92}=214	ρ_{91}=213	ρ_{90}=212		ρ_{89}=210	ρ_{88}=209	ρ_{87}=208	
ρ_{100}=223	ρ_{99}=222	ρ_{98}=221	ρ_{97}=220	ρ_{96}=219	ρ_{95}=218	ρ_{94}=217	ρ_{93}=216	
ρ_{108}=231	ρ_{107}=230	ρ_{106}=229	ρ_{105}=228	ρ_{104}=227	ρ_{103}=226	ρ_{102}=225	ρ_{101}=224	
ρ_{116}=239	ρ_{115}=238	ρ_{114}=237	ρ_{113}=236	ρ_{112}=235	ρ_{111}=234	ρ_{110}=233	ρ_{109}=232	
	ρ_{123}=246	ρ_{122}=245	ρ_{121}=244	ρ_{120}=243	ρ_{119}=242	ρ_{118}=241	ρ_{117}=240	7
ρ_{131}=255	ρ_{130}=254	ρ_{129}=253	ρ_{128}=252	ρ_{127}=251	ρ_{126}=250	ρ_{125}=249	ρ_{124}=248	
ρ_{139}=263	ρ_{138}=262	ρ_{137}=261	ρ_{136}=260	ρ_{135}=259	ρ_{134}=258	ρ_{133}=257	ρ_{132}=256	
ρ_{147}=271	ρ_{146}=270	ρ_{145}=269	ρ_{144}=268	ρ_{143}=267	ρ_{142}=266	ρ_{141}=265	ρ_{140}=264	
ρ_{155}=279	ρ_{154}=278	ρ_{153}=277	ρ_{152}=276	ρ_{151}=275	ρ_{150}=274	ρ_{149}=273	ρ_{148}=272	
ρ_{163}=287	ρ_{162}=286	ρ_{161}=285	ρ_{160}=284	ρ_{159}=283	ρ_{158}=282	ρ_{157}=281	ρ_{156}=280	8

$s = s' + \lfloor k/\Sigma \rfloor$ – уточнення значення рівня, $\lfloor \cdot \rfloor$ – округлення до найменшого цілого.

Результат випливає з кубічної залежності k від s. Для обчислення рядка розташування полюса ρ_k на рівні s слід врахувати, що число полюсів по рядках на рівні визначається арифметичним рядом. Результуючі вирази мають вигляд:

$s_1 = s - q_0 - 1$ – значення числа рядків доповнення,

$$\Sigma_{s-1} = \sum_{i=1}^{s-1} i^2 - 2\sum_{j=1}^{s_1-1} j^2 - 2\sum_{j=1}^{s_1-1} j = s(s-1)(2s-1)/6 - s_1(s_1-1)(2s_1-1)/3 - s_1(s_1-1)$$

– число полюсів на рівнях $1, \ldots, s-1$,

$k' = k - \Sigma_{s-1}$ – число полюсів на рівні s,

$k_1 = \lceil k'/2 \rceil$ – число полюсів на рівні s для одного шару,

$k_2 = k_1 + s_1(s_1+1)/2$ – доповнення числа полюсів на рівні s за рахунок s_1 рядків,

$s_2 = \left\lceil (2k_2 + 1/4)^{1/2} - 1/2 \right\rceil$ – значення рядка розташування полюса.

Число рядків t_1 розміру q_0 на рівні s визначається виразами:

$s_3 = s_2 - q_0 - 1$ – число рядків розташування полюсів, які необхідно відняти від суми повного арифметичного ряду,

$k_3 = (s_2 - 1)s_2/2 - (s_1 - 1)(s_1 + 1)/2 - s_3(s_3 + 1)/2$ – число полюсів на рівні s для одного шару,

$\Delta = k' - 2k_3$ – число полюсів на рядку s_2,

$t_1 = \lceil \Delta/q_0 - 1 \rceil$ – число рядків розміру q_0.

Результуючі вираження для показників i, j, t, r ступенів раціональних функцій w, v, y, x визначаються за формулами:

$t = t_1 \bmod 2$ – ступінь y^t (визначник другого шару),

$j = \Delta - t \cdot q_0 - 1$ – ступінь w^j,

Таблиця 6.12 – Розміщення полюсів підгрупи Вейерштрасса $H(P_\infty)=\langle q, q+q_0, q+2q_0, q+2q_0+1\rangle$

№	Полюса першого шару				Полюса другого шару			
1	$\rho_0=0$							
	$\rho_1=q=\varphi$				$\rho_2=q+q0=\omega$			
2	$\rho_3=q+2q0=\eta$	$\rho_4=q+2q0+1=\gamma$						
	$\rho_5=2\varphi$				$\rho_6=\omega+\varphi$			
	$\rho_7=\eta+\varphi$	$\rho_8=\gamma+\varphi$			$\rho_9=\eta+\omega$	$\rho_{10}=\gamma+\omega$		
3	$\rho_{11}=2\eta$	$\rho_{12}=\gamma+\eta$	$\rho_{13}=2\gamma$					
	$\rho_{14}=3\varphi$				$\rho_{15}=\omega+2\varphi$			
	$\rho_{16}=\eta+2\varphi$	$\rho_{17}=\gamma+2\varphi$			$\rho_{18}=\eta+\omega+\varphi$	$\rho_{19}=\gamma+\omega+\varphi$		
	$\rho_{20}=2\eta+\varphi$	$\rho_{21}=\gamma+\eta+\varphi$	$\rho_{22}=2\gamma+\varphi$		$\rho_{23}=2\eta+\omega$	$\rho_{24}=\gamma+\eta+\omega$	$\rho_{25}=2\gamma+\omega$	
4	$\rho_{26}=3\eta$	$\rho_{27}=\gamma+2\eta$	$\rho_{28}=2\gamma+\eta$	$\rho_{29}=3\gamma$				
	$\rho_{30}=4\varphi$				$\rho_{31}=\omega+3\varphi$			
	$\rho_{32}=\eta+3\varphi$	$\rho_{33}=\gamma+3\varphi$			$\rho_{34}=\eta+\omega+2\varphi$	$\rho_{35}=\gamma+\omega+2\varphi$		
	$\rho_{36}=2\eta+2\varphi$	$\rho_{37}=\gamma+\eta+2\varphi$	$\rho_{38}=2\gamma+2\varphi$		$\rho_{39}=2\eta+\omega+\varphi$	$\rho_{40}=\gamma+\eta+\omega+\varphi$	$\rho_{41}=2\gamma+\omega+\varphi$	
	$\rho_{42}=3\eta+\varphi$	$\rho_{43}=\gamma+2\eta+\varphi$	$\rho_{44}=2\gamma+\eta+\varphi$	$\rho_{45}=3\gamma+\varphi$	$\rho_{46}=3\eta+\omega$	$\rho_{47}=\gamma+2\eta+\omega$	$\rho_{48}=2\gamma+\eta+\omega$	$\rho_{49}=3\gamma+\omega$
5	$\rho_{50}=4\eta$	$\rho_{51}=\gamma+3\eta$	$\rho_{52}=2\gamma+2\eta$	$\rho_{53}=3\gamma+\eta$	$\rho_{54}=4\gamma$			
	$\rho_{55}=\eta+4\varphi$	$\rho_{56}=\gamma+4\varphi$			$\rho_{57}=\eta+\omega+3\varphi$	$\rho_{58}=\gamma+\omega+3\varphi$		
	$\rho_{59}=2\eta+3\varphi$	$\rho_{60}=\gamma+\eta+3\varphi$	$\rho_{61}=2\gamma+3\varphi$		$\rho_{62}=2\eta+\omega+2\varphi$	$\rho_{63}=\gamma+\eta+\omega+2\varphi$	$\rho_{64}=2\gamma+\omega+2\varphi$	
	$\rho_{65}=3\eta+2\varphi$	$\rho_{66}=\gamma+2\eta+2\varphi$	$\rho_{67}=2\gamma+\eta+2\varphi$	$\rho_{68}=3\gamma+2\varphi$	$\rho_{69}=3\eta+\omega+\varphi$	$\rho_{70}=\gamma+2\eta+\omega+\varphi$	$\rho_{71}=2\gamma+\eta+\omega+\varphi$	$\rho_{72}=3\gamma+\omega+\varphi$
	$\rho_{73}=4\eta+\varphi$	$\rho_{74}=\gamma+3\eta+\varphi$	$\rho_{75}=2\gamma+2\eta+\varphi$	$\rho_{76}=3\gamma+\eta+\varphi$	$\rho_{77}=4\eta+\omega$	$\rho_{78}=\gamma+3\eta+\omega$	$\rho_{79}=2\gamma+2\eta+\omega$	$\rho_{80}=3\gamma+\eta+\omega$
6	$\rho_{81}=5\eta$	$\rho_{82}=\gamma+4\eta$	$\rho_{83}=2\gamma+3\eta$	$\rho_{84}=3\gamma+2\eta$	$\rho_{85}=4\gamma+\eta$	$\rho_{86}=5\gamma$		
	…	…	…	…	…	…		
	$\rho_{s(s-1)(2s-1)/6+s_2(s_2-1)-s_1(s_1-1)-s_3(s_3-1)+t_1q_0+j+1}=i(q+2q_0)+j(q+2q_0+1)+t(q+q_0)+r$							

$d=\lfloor s_2/(s-t)\rfloor$ – індикатор останнього рядка s_2 на рівні s,

$r=s-s_2-dq_0-t$ – ступінь x^r,

$i=s-1-r-j-t$ – ступінь v^i.

Приклад. Нехай $y^{32}-y=x^4\left(x^{32}-x\right)$ над полем F_{2^5}, $q=2^5$. Полюса представлені таблицею 6.12. Обчислити значення полюса ρ_l.

Нехай $l=113$. Маємо $k=113+1=114$ і обчислення за формулами леми дають наступне:

$$s'=\left\lceil (3k)^{1/3}\right\rceil=(3\cdot 114)^{1/3}=7,$$

$$\Sigma=s'(s'+1)(2s'+1)/6=7\cdot 8\cdot 15/6=140,$$

$$s=s'+\lfloor k/\Sigma\rfloor=7,$$

$$s_1=s-q_0-1=7-5=2,$$

$$\Sigma_{s-1}=s(s-1)(2s-1)/6-s_1(s_1-1)(2s_1-1)/3-s_1(s_1-1)=$$
$$=7\cdot 6\cdot 13/6-2\cdot 1\cdot 3/3-2\cdot 1=87,$$

$$k'=k-\Sigma_{s-1}=114-87=27,$$

$$k_1=\lceil k'/2\rceil=\lceil 27/2\rceil=14,$$

$$k_2=k_1+s_1(s_1+1)/2=14+2\cdot 3/2=17,$$

$$s_2=\left\lceil (2k_2+1/4)^{1/2}-1/2\right\rceil=\lceil (2\cdot 17+0,25)1/2-0,5\rceil=6,$$

$$s_3=s_2-q_0-1=6-4-1=1,$$

$$k_3=(s_2-1)s_2/2-s_1(s_1+1)/2-s_3(s_3+1)/2=5\cdot 6/2-2\cdot 3/2-1\cdot 2/2=11,$$

$$\Delta=k'-2k_3=27-2\cdot 11=5,$$

$$t_1=\lceil \Delta/q_0-1\rceil=\lceil 5/4-1\rceil=1,$$

$$t=t_1 \bmod 2=1,$$

$$j=\Delta t\cdot q_0-1=5-5-1=0,$$

$$d=\lfloor s_2/(s-t)\rfloor=\lfloor 6/(7-1)\rfloor=1,$$

$$r=s-s_2+d\cdot q_0=7-6+1\cdot 4=5.$$

$$r=\varepsilon-\varepsilon_2 dq_0-t,$$

$$i=\varepsilon-1-r-j-t.$$

Отримаємо значення полюса

$$\rho_{113}=i(q+2q_0)+j(q+2q_0+1)+t(q+q_0)+rq=$$
$$=1\cdot 40+0\cdot 41+1\cdot 36+5\cdot 32=236,$$

що збігається зі значенням в таблиці 6.12.

Затвердження 6.18 [303]. Хешування за раціональними функціями кривої $y^q - y = x^{q_0}\left(x^q - x\right)$, де $q = 2q_0^2$ і $q_0 = 2^s$ над полем F_q, визначає універсальний хеш-клас $\varepsilon - U\left(q^2, q^k, q\right)$, де q^2 – число хеш-функцій (обсяг ключового простору), q^k – обсяг простору повідомлень, q – обсяг простору хеш-кодів. Імовірність колізії ε визначається співвідношеннями:

$$\varepsilon = \left(i\left(q + 2q_0\right) + j\left(q + 2q_0 + 1\right) + t\left(q + q_0\right) + rq\right)/q^2, \text{ якщо } k < q_0\left(q - 1\right), \qquad (6.37)$$

$$\varepsilon = \left(k + q_0\left(q - 1\right)\right)/q^2, \text{ якщо } k \geq q_0\left(q - 1\right), \qquad (6.38)$$

де i, j, t, r визначаються лемою.

Доказ. Параметри універсального класу по раціональним функціям кривої Судзукі випливають з визначення кривої і числа її точок в F_q. Імовірність колізії визначається співвідношенням $\varepsilon = \rho_k / N$, де $\rho_k = i\left(q + 2q_0\right) + j\left(q + 2q_0 + 1\right) + t\left(q + q_0\right) + rq$ – значення полюса раціональної функції $f_k = w^j \cdot v^i \cdot y^t \cdot x^r$, $N = q^2$ – число точок кривої.

Нехай $k < q_0\left(q - 1\right)$. Параметри i, j, t, r визначаються лемою, і по підстановці в $\varepsilon = \rho_k / N$ слідує (6.38).

У вападку $k = q_0\left(q - 1\right)$ маємо $\rho_k = 2g = 2q_0\left(q - 1\right)$ і $\varepsilon = 2g/N$, що узгоджується з (6.38). З іншого боку,
$\rho_k = i\left(q + 2q_0\right) + j\left(q + 2q_0 + 1\right) + t\left(q + q_0\right) + rq$. Обчислення за формулами леми дають $i = q_0 - 1$, $j = 0$, $t = 0$, $r = q_0$ і прямою підстановкою отримаємо перевірку:

$$\rho_k = i\left(q + 2q_0\right) + j\left(q + 2q_0 + 1\right) + t\left(q + q_0\right) + rq = \\ = \left(q_0 - 1\right)\left(q + 2q_0\right) + q_0 q = 2q_0\left(q - 1\right).$$

Нехай $k > q_0\left(q - 1\right)$. Зауважимо, що $\rho_k = k + q_0\left(q - 1\right)$. Пряме обчислення $\varepsilon = \rho_k / N$ призводить до вираження (6.38).

Зауваження 6.19

1. Нехай $k \approx \sqrt{q}\left(\sqrt{q}-1\right)/2$. За лемою маємо $s \approx \left[\left(3k\right)^{1/3}\right] \approx \left(q\right)^{1/3}$, значення параметрів $i \approx s$, $j=0$, $t=0$, $r=0$ і оцінку ймовірності колізії

$$\varepsilon \approx q^{1/3}\left(q+q_0\right)/q^2 \approx q^{-1/6}\varepsilon_{HC}, \tag{6.39}$$

де $\varepsilon_{HC} = 1/\sqrt{q} + 1/q$ – значення ймовірності колізії універсального хешування по кривій Ерміта в квадратичному полі F_q при $k = \sqrt{q}\left(\sqrt{q}-1\right)/2$. З оцінки (6.39) слідує виграш в $q^{1/6}$ раз по ймовірності колізії хешування по кривій Ерміта. Розмір ключових даних $N = q^2$ в порівнянні з хешуванням по кривій Ерміта більше в $\sqrt{q}$ рази.

2. Для $k = q_0\left(q-1\right)$ маємо можливість колізії хешування по кривим Судзукі

$$\varepsilon = 2q_0\left(q-1\right)/q^2 \approx \sqrt{2}q^{-1/2}. \tag{6.40}$$

Підстановка $k = q_0\left(q-1\right)$ в вираз для ймовірності колізії хешування по кривим Ерміта дає

$$\begin{gathered}\varepsilon = k/\ \ q^3 + 1/\left(2q\right) - 1/\left(2q^2\right) = \\ = \frac{1}{\sqrt{2}}\sqrt{q}\left(q-1\right)/q\sqrt{q} + 1/\left(2\sqrt{q}\right) - 1/\left(2q\right) \approx 1/\sqrt{2}.\end{gathered} \tag{6.41}$$

3. Універсальне хешування по кривій Судзукі для випадку $q_0 = 2^s$ і $q = 2q_0$, асоційоване з дільником $D = \left|\left(q+2q_0+1\right)P_0\right|$, визначається на тому ж F_q раціональному морфізмі $\pi := \left(1 : x : y : v : w\right)$ в P^4 з тими ж порядками полюсів, як для випадку кривої з параметрами $q_0 = 2^s$ і $q = 2q_0{}^2$. Отже, справедливо визначення (6.34) і співвідношення для вірогідності колізії (6.37) і (6.38). Для $k = q_0\left(q-1\right)$ отримаємо оцінку ймовірності колізії хешування

$$\varepsilon = 2q_0(q-1)/q^2 \approx \left(4{q_0}^2 - 2q_0\right)/4{q_0}^2 \approx 1,$$

що гірше в порівнянні з хешуванням з параметрами $q_0 = 2^s$ і $q = 2{q_0}^2$:

$$\varepsilon = 2q_0(q-1)/q^2 \approx 1/q_0 .$$

4. Універсальне хешування за максимальною кривою Судзукі над кінцевим полем F_{q^4} обчислюється за п'ятьма параметричними функціями базису $L(lD)$:

$$S := \left\{ \begin{array}{ll} x^a y^b v^c w^d \left(x^q + x\right)^r & aq + b(q+q_0) + c(q+2q_0) + \\ & +d(q+2q_0+1) + rq^2 \le l\left(q^2+1\right) \\ & 0 \le a \le q-1, 0 \le b \le 1, 0 \le c \le \\ & \le q_0 - 1, 0 \le d \le q_0 - 1, 0 \le r \le l \end{array} \right\}. \qquad (6.42)$$

Функціональне поле вибудовується в порядку зростання полюсів $f = x^a y^b v^c w^d \left(x^q + x\right)^r$. Для числа слів даних $k < q(q-1)^2$ обчислення раціональних функцій здійснюється за параметрами x, y, v, w. Для крайніх параметрів a, b, c, d з (6.42) матимемо значення дискретної оцінки

$$\nu_\infty(f) = (q-1)q + (q+q_0) + (q_0-1)(q+2q_0) + \\ +(q_0-1)(q+2q_0+1) = q^2 + 2g - 1.$$

Для числа слів даних $k > q(q-1)^2$ обчислення раціональних функцій слід виконувати за п'ятьма параметрами $x, y, v, w, \left(x^q + x\right)$, що дозволяє побудувати функціональний простір з безперервною послідовністю полюсів для великих значень k.

Для $k = q_0(q-1)$ отримаємо оцінку ймовірності колізії хешування над полем F_{q^4}

$$\varepsilon = 2q_0(q-1)/\ q^4 \approx \frac{1}{q^2 q_0}.$$

Хешування за кривою Судзукі над полем F_p з параметрами $p_0 = 2^s$, $p = 2p_0^{\,2}$ і $p \approx q^4$ для $k = q_0(q-1)$ слів даних аналогічно п.1 зауваження матиме оцінку ймовірності колізії $\varepsilon \approx \rho_k / p^2 \approx \frac{k^{1/3}(q+q_0)}{q^8} \approx \frac{1}{q^6 q_0}$, що істотно краще в порівнянні з хешуванням в розширеному полі F_{q^4}.

Висновки за розділом 6.3

Результатами розділу є оцінки параметрів групи Судзукі, кривих, асоційованих з підгрупами групи Судзукі, обчислення функціонального поля кривої, асоційованої з підгрупою $S(a,b)$ повної групи Судзукі, побудова універсального хешування по раціональним функціям функціонального поля кривої Судзукі, оцінки параметрів універсального хешування. Метод універсального хешування по раціональним функціям кривої Судзукі і його властивості представлені в роботах [295,303,304,308-313]. Основні результати досліджень такі.

1. Крива Судзукі визначена з точністю до F_q ізоморфізма, асоційована з підгрупою $S(a,b)$ групи Судзукі $Sz(q)$, заснована на роді, числі точок і груповому F_q автоморфізмі кривої. Крива має відносно свого роду максимальне число точок і над полем F_q, де $q = 2q_0^{\,2}$, $q_0 = 2^s$ і трохи менше кордону Хассе - Вейля. Рід кривої $g = q_0(q-1)$ і число F_q раціональних точок дорівнює $q^2 + 1$.

2. Крива Судзукі над кінцевим полем ступеня розширення $r \equiv 0 \bmod 4$ є максимальною, має мале значення роду і відповідно щодо розміру поля мале число точок. Крива Судзукі над квадратичним і кубічним полем є неоптимальною.

3. Крива Судзукі над полем F_q, де $q_0 = 2^s$ і $q = 2q_0$ також являється F_q - ізоморфною кривій Делігне - Лустіга, асоційованої з групою Судзукі $Sz(q)$, але має в q_0^2 менше точок в порівнянні з випадком $q = 2q_0^2$.

4. Класи похідних кривих по підгрупах групи Судзукі для випадку $q = 2q_0^2$ і $q_0 = 2^s$ визначені за циклічною підгрупою порядку r, $r|q-1$, по підгрупі Зінгера порядку $q \pm 2q_0 + 1$, по підгрупах порядків $2^u r$, $2r$, $2s$, $4s$, де $r|(q-1)$, $s|(q \pm 2q_0 + 1)$. Криві по даним підгрупам мають число точок, відповідних порядків підгруп, що менше числа точок кривої, асоційованої з підгрупою $S(a,b)$ порядку q^2.

5. Універсальне хешування по кривій Судзукі будується на основі відображення $\pi := (1 : x : y : v : w)$ в проективному просторі P^4. Хешування за раціональними функціями кривої над полем F_q визначає універсальний хеш-клас $\varepsilon - U(q^2, q^k, q)$, де q^2 – число хеш-функцій (обсяг ключового простору), ε – верхня оцінка ймовірності колізії, k – число q-х слів даних.

6. Хешування за кривою Судзукі має виграш в $q^{1/6}$ раз по ймовірності колізії і в $q^{1/2}$ раз - по числу слів даних в порівнянні з універсальним хешуванням по кривій Ерміта. Хешування по кривій з параметрами $q = 2q_0{}^2$ і $q_0 = 2^s$ має перевагу по ймовірності колізії, довжині даних, складності обчислень в порівнянні з хешуванням за похідними кривим Судзукі і кривої над полем четвертого ступеня розширення.

СПИСОК ЛІТЕРАТУРИ

1. Горбенко І. Д. Прикладна криптологія [Текст] / І. Д. Горбенко, Ю. І. Горбенко. – Х. : Форт, 2012. – 870 с.

2. Menezes A. J. Handbook of applied cryptography [Text] / Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. – CRC press, 2010.

3. Шеннон К. Теория связи в секретных системах [Текст] / Клод Шеннон // Работы по теории информации и кибернетике. – М., 1963. – С. 333–369.

4. Бабаш А. В. Криптография [Текст] / А. В. Бабаш, Г. П. Шанкин. – М. : СОЛОН-Р, 2002.

5. NESSIE security report [Electronic resource] : D20 : Version 2.0, IST-1999-12324 / B. Preneel et al. – February 19, 2003. – Mode of access : www. URL: https://www.cosic.esat.kuleuven.be/nessie/deliverables/D20-v2.pdf.

6. Основы криптографии [Текст] : учеб. пособие / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. – М. : Гелиос АРВ, 2005.

7. Харин Ю. С. Математические и компьютерные способы криптографии [Текст] / Ю. С. Харин. – Минск, 2003. – 391 с.

8. Patarin J. Generic attacks on Feistel schemes [Text] / J. Patarin // Advances in Cryptology – ASIACRYPT 2001 : proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9–13, 2001. – Berlin ; Heidelberg : Springer, 2001. – P. 222–238. – (Lecture Notes in Computer Science ; vol. 2248).

9. Hellman M. E. A Cryptanalytic Time-Memory Trade-Off [Text] / Martin E. Hellman // IEEE Transactions on Information Theory. – 1980. – Vol. 26, № 4. – P. 401–406.

10. Denning D. E. Cryptography and Data Security [Text] / Dorothy E. Denning. – Boston, MA : Addison-Wesley, 1982.

11. Oechslin P. Making a Faster Cryptanalytic Time-Memory Trade-Off [Text] / Philippe Oechslin // Advances in Cryptology – CRYPTO 2003 : proceedings of the 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 2003. – Berlin ; Heidelberg : Springer, 2003. – P. 617–630. – (Lecture Notes in Computer Science ; vol. 2729).

12. Barkan E. Rigorous Bounds on Cryptanalytic Time/Memory Tradeoffs [Text] / Elad Barkan, Eli Biham, Adi Shamir // Advances in Cryptology – CRYPTO 2006 : proceedings of the 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 2006. – Berlin ; Heidelberg : Springer, 2006. – P. 1–21. – (Lecture Notes in Computer Science ; vol. 4117).

13. Avoine G. Time-Memory Trade-Off: False Alarm Detection Using Checkpoints [Text] / Gildas Avoine, Pascal Junod, Philippe Oechslin // Progress in Cryptology – INDOCRYPT 2005 : proceedings of the 6th International Conference on Cryptology in India, Bangalore, India, December 10–12, 2005. – Berlin ; Heidelberg : Springer-Verlag, 2005. – P. 183–196. – (Lecture Notes in Computer Science ; vol. 3797).

14. Кнут Д. Искусство программирования для ЭВМ [Текст] : в 7 т. / Д. Кнут. – 2-е изд. – М. : Вильямс, 2007. – Т. 3 : Сортировка и поиск. – 824 с.

15. Положення про порядок розроблення, виробництва та експлуатації засобів криптографічного захисту інформації [Електронний ресурс] : наказ № 141 від 20 лип. 2007 р. / Адміністрація Держ. служби спеціального зв'язку та захисту інформ. України. – Режим доступу : www. URL: http://zakon4.rada.gov.ua/laws/show/z0862-07.

16. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths [Electronic resource] : NIST Special Publication 800–131A / National Institute of Standards and Technology. – January 2011. – Mode of access : www. URL: http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf.

17. N.S.A. Able to Foil Basic Safeguards of Privacy on Web [Text] // The New York Times. – 2013. – September 5th.

18. The D-Wave Two system [Electronic resource]. – Mode of access : www. URL: http://www.dwavesys.com/d-wave-two-system.

19. Data Encryption Standard (DES) [Electronic resource] : FIPS 46–3 / National Bureau of Standards, USA. – 1993. – Mode of access : www. URL: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.

20. ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования [Текст]. – Введ. 01–07–1990. – М. : Изд-во стандартов, 1989. – 28 с.

21. Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms – Design and Analysis [Text] / K. Aoki et al. // Selected Areas in Cryptography. – Berlin ; Heidelberg : Springer, 2001. – P. 39–56.

22. Barreto P. The Anubis block cipher [Electronic resource] / P. Barreto, V. Rijmen // First Open NESSIE Workshop, Leuven, Belgium, November 13–14, 2000. – Mode of access : https://www.cosic.esat.kuleuven.be/nessie/workshop/.

23. Borst J. The block cipher: GrandCru [Electronic resource] / J. Borst // First Open NESSIE Workshop, Leuven, Belgium, November 13–14, 2000. – Mode of access : www. URL: https://www.cosic.esat.kuleuven.be/nessie/workshop/.

24. The Noekeon block cipher [Electronic resource] / J. Daemen et al. // First Open NESSIE Workshop, Leuven, Belgium, November 13–14, 2000. – Mode of access : www. URL: https://www.cosic.esat.kuleuven.be/nessie/workshop/.

25.=56 Перспективний блоковий симетричний шифр «Калина» – основні положення та специфікація [Текст] / І. Д. Горбенко, В. І. Долгов, Р. В. Олійников [та ін.] // Прикладная радиоэлектроника. – Х., 2007. – Т. 6, № 2. – С. 195–208.

26. Vaudenay S. On the Lai-Massey scheme [Text] / S. Vaudenay // Advances in Cryptology – ASIACRYPT'99 : proceedings of the International Conference on the Theory and Applications of Cryptology and Information

Security, Singapore, November 14–18, 1999. – Berlin ; Heidelberg : Springer, 1999. – P. 8–19. – (Lecture Notes in Computer Science ; vol. 1716).

27. Junod P. FOX: a new family of block ciphers [Text] / P. Junod, S. Vaudenay // Selected Areas in Cryptography. – Berlin ; Heidelberg : Springer, 2005. – P. 114–129.

28. Meier W. On the security of the IDEA block cipher / W. Meier // Advances in Cryptology – Eurocrypt'93 : proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23–27, 1993. – Berlin : Springer-Verlag, 1994. – P. 371–385. – (Lecture Notes in Computer Science ; vol. 765).

29. Перспективний блоковий симетричний шифр «Мухомор»: основні положення та специфікація [Текст] / Р. В. Олійников, І. Д. Горбенко, М. Ф. Бондаренко, В. І. Руженцев // Прикладная радиоэлектроника. – 2007. – Т. 6, № 2. – С. 147–156.

30. Skipjack and KEA Algorithm Specifications. Version 2.0 [Electronic resource] / National Institute for Standards and Technology (NIST). – 29 May 1998. – Mode of access : www. URL: http://cryptome.org/jya/skipjack-spec.htm.

31. ISO/IEC 29192–2:2012. Information technology – Security techniques – Lightweight cryptography – Part 2 : Block ciphers [Electronic resource]. – Mode of access : www. URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=56552.

32. Advanced Encryption Standard (AES) Development Effort [Electronic resource]. – Mode of access : http://csrc.nist.gov/archive/aes/index2.html#overview.

33. New European Schemes for Signatures, Integrity, and Encryption (NESSIE), IST-1999-12324 [Electronic resource]. – Mode of access : www. URL: https://www.cosic.esat.kuleuven.be/nessie.

34. CRYPTREC Cryptography Research and Evaluation Committees [Electronic resource]. – Mode of access : www. URL:

http://www.cryptrec.go.jp/english/about.html.

35. Luby M. How to construct pseudorandom permutations from pseudorandom functions [Text] / M. Luby, C. Rackoff // SIAM Journal on Computing. – 1988. – Vol. 17, № 2. – P. 373–386.

36. Naor M. On the construction of pseudo-random permutations: Luby-Rackoff revisited [Text] / M. Naor, O. Reingold // Proceedings of the 29th annual ACM symposium on Theory of computing. – ACM, 1997. – P. 189–199.

37. Feistel H. Cryptography and Computer Privacy / H. Feistel // Scientific American. – 1973. – Vol. 228, No 5. – P. 15–23.

38. Sorkin A. LUCIFER: a cryptographic algorithm / A. Sorkin // Cryptologia. – 1984. – Vol. 8(1). – P. 22–35.

39. Aoki K. Specification of Camelia – a 128-bit Block Cipher [Electronic resource] / K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita // 2001. – Mode of access: http:// www.cryptrec.go.jp/cryptrec_03_spec_ cypherlist_files/PDF/06_01espec.pdf.

40. RFC-4312: The Camellia Cipher Algorithm and Its Use With IPsec [Electronic resource] / A. Kato, M. Kanda, S. Moriai // 2005. – Mode of access: http://tools.ietf.org/html/rfc4312.

41. Lai X. A proposal for a new block encryption standard / X. Lai, J. L. Massey // Advances in Cryptology. – Proc. : EUROCRYPT. – 1991. – P. 389–404.

42. Принципи побудування та властивості блокових симетричних IDEA подібних шифрів / Р. В. Олійников, І. Д. Горбенко, В. І. Долгов, В. І. Руженцев // Прикладная радиоэлектроника. – Харьков, 2007. – Т. 6, № 2. – С. 158–173.

43. Maurer M. A simplified and Generalized Treatment of Luby – Rackoff Pseudorandom Permutations Generator / Ueli M. Maurer // Advances in Cryptology. – Springer-Verlag, Berlin, 1993. – P. 239–255.

44. Luby M. How to construct pseudorandom permutations from pseudorandom functions / M. Luby, C. Rackoff // SIAM Journal on Computing. – 1988. – Vol. 17, № 2. – P. 373–386.

45. Maurer U. M. A simplified and generalized treatment of Luby-Rackoff pseudorandom permutation generators / U. M. Maurer // Advances in Cryptology – EUROCRYPT'92 : proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Balatonfüred, Hungary, May 24–28, 1992. – Berlin ; Heidelberg : Springer, 1993. – P. 239–255. – (Lecture Notes in Computer Science ; vol. 658).

46. Patarin J. Generic attacks on Feistel schemes / J. Patarin // Advances in Cryptology – ASIACRYPT 2001 : proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9–13, 2001. – Berlin ; Heidelberg : Springer, 2001. – P. 222–238. – (Lecture Notes in Computer Science ; vol. 2248).

47. Олейников Р.В. Уточнение эффективности различения цепи Фейстеля и случайной перестановки / Р.В. Олейников, Д.С. Кайдалов // Радиотехника. – Харьков, ХНУРЭ, 2011. – № 167. – С. 190–202.

48. Кайдалов Д.С. Различение цепи Фейстеля и случайной перестановки / Д.С. Кайдалов // Научные исследования молодежи – решению проблем европейской интергации : сб. науч. тр. XVI Междунар. науч.-практ. конф., 16 марта 2011 г. – Харьков : УБД ХИБД, 2011. – С. 25.

49. Кайдалов Д.С. Уточненная оценка верхней границы вероятности различения цепи фейстеля и случайной функции / Д.С. Кайдалов, Р.В. Олейников // Радиоэлетроника и молодежь в XXI веке : сб. науч. тр. XV Юбил. междунар. молодежного форума. – Харьков : ХНУРЭ, 2011. – С. 203–204.

50. Кайдалов Д.С. Различение цепи Фейстеля и случайной перестановки / Д.С. Кайдалов, Р.В. Олейников // Защита информации с ограниченным доступом и автоматизация ее обработки : сб. науч. тр. III Науч.-техн. конф. студентов и аспирантов, 8–9 февраля 2011 г. – Киев : НАУ,

2011. – С. 9.

51. Вентцель Е.С. Теория вероятностей / Е.С. Вентцель // Учеб. Для вузов. – 7-е изд. – Москва : Высш. шк., 2001. – С. 40–45.

52. Patarin J. New Results on Pseudorandom Permutation Generators Based on the DES Scheme / J. Patarin // Advances in Cryptology Crypto '91. – Berlin : Springer-Verlag, 1992. – P. 301–312.

53. Бондаренко М. Ф. Анализ основных атак на трёхраундовую цепь Фейстеля / М. Ф. Бондаренко, А. Б. Небывайлов // Прикладная радиоэлектроника. – ХНУРЭ, 2009. – Т. 8, С. 278–281.

54. Gorbenko I. Improvement for Distinguisher Efficiency of the 3-Round Feistel Network and a Random Permutation / I. Gorbenko, R. Oliynykov, D. Kaidalov // Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications : by materials of the 6th IEEE International Conference, 15–17 September 2011. – Prague, Czech Republic : IEEE, 2011. – P. 743–746.

55. Maurer M. A simplified and Generalized Treatment of Luby – Rackoff Pseudorandom Permutations Generator / Ueli M. Maurer // Advances in Cryptology. – Springer-Verlag, Berlin, 1993. – P. 239–255.

56. Meier W. On the security of the IDEA block cipher / W. Meier // Advances in Cryptology – Eurocrypt'93 : proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23–27, 1993. – Berlin : Springer-Verlag, 1994. – P. 371–385. – (Lecture Notes in Computer Science ; vol. 765).

57. Biham E. Preliminary Report on the Nessie Submissions: Anubis, Camellia, Khazad, IDEA, Misty1, NIMBUS, and Q [Electronic resource] / E. Biham, O. Dunkelman, V. Furman, T. Mor // Computer Science Department, Technion, Haifa, Israel. – Mode of access: https://www.cosic.esat.kuleuven.be/nessie/reports/phase1/tecwp3-011b.pdf.

58. Rompay B. A first report on CS-Cipher, Hierocrypt, Grand Cru, SAFER++ and SHACAL [Electronic resource] / B. Rompay, V. Rijmen, J. Nakahara // Belgium. – 2001. Mode of access: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.5009&rep=rep1&type=pdf.

59. Daemen J. Nessie Proposal: Noekeon [Electronic resource] / J. Daemen, M. Peeters, G. V. Assche, V. Rijmen // Mode of access: http://gro.noekeon.org/Noekeon-spec.pdf.

60. Шеннон К. Теория связи в секретных системах / К. Шеннон // Работы по теории информации и кибернетике. – Москва, 1963. – С. 333–369.

61. Gilbert H. Super-Sbox Cryptanalysis: Improved Attacks for AES-like permutations [Electronic resource] : report 2009/531 / H. Gilbert, T. Peyrin // Cryptology ePrint Archive. – 2009. – Mode of access: https://eprint.iacr.org/2009/531.pdf.

62. Federal Information Processing Standards Publication 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001 [Electronic resource]. – Mode of access: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

63. Biryukov A. Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds / A. Biryukov et al. // Advances in Cryptology – EUROCRYPT 2010 : proceedings of the 9th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010. – Berlin ; Heidelberg : Springer, 2010. – P. 299–319. – (Lecture Notes in Computer Science ; vol. 6110).

64. Bogdanov A. Biclique cryptanalysis of the full AES / A. Bogdanov, D. Khovratovich, C. Rechberger // Advances in Cryptology – ASIACRYPT 2011 : proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4–8, 2011. – Berlin ; Heidelberg : Springer, 2011. – P. 344–371. – (Lecture Notes in Computer Science ; vol. 7073).

65. Biryukov A. Distinguisher and related-key attack on the full AES-256 / A. Biryukov, D. Khovratovich, I. Nikolić // Advances in Cryptology – CRYPTO’99 : proceedings of the 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 1999. – Berlin ; Heidelberg : Springer, 1999. – P. 231–249. – (Lecture Notes in Computer Science ; vol. 1666).

66. Biryukov A. Related-key Cryptanalysis of the Full AES-192 and AES-256 / A. Biryukov, D. Khovratovich // Advances in Cryptology – ASIACRYPT 2009 : proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6–10, 2009. – Berlin ; Heidelberg : Springer, 2009. – P. 1–18. – (Lecture Notes in Computer Science ; vol. 5912).

67. Biryukov A. Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds / A. Biryukov et al. // Advances in Cryptology – EUROCRYPT 2010 : proceedings of the 9th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010. – Berlin ; Heidelberg : Springer, 2010. – P. 299–319. – (Lecture Notes in Computer Science ; vol. 6110).

68. Biryukov A. Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia, Khazad and others / A. Biryukov, I. Nikolić // Advances in Cryptology – EUROCRYPT 2010 : proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010. – Berlin ; Heidelberg : Springer, 2010. – P. 322–344. – (Lecture Notes in Computer Science ; vol. 6110).

69. Kim J. Related-key rectangle attacks on reduced AES-192 and AES-256 / J. Kim, S. Hong, B. Preneel // Fast Software Encryption. – Berlin ; Heidelberg : Springer, 2007. – P. 225–241.

70. Biham E. Related-key impossible differential attacks on 8-round AES-192 / E. Biham, O. Dunkelman, N. Keller // Topics in Cryptology – CT-RSA 2006

: proceedings of the Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13–17, 2006. – Berlin ; Heidelberg : Springer, 2006. – P. 21–33. – (Lecture Notes in Computer Science ; vol. 3860).

71. Zhang W. Improved related-key impossible differential attacks on reduced-round AES-192 / W. Zhang et al. // Selected Areas in Cryptography : proceedings of the 14th International Workshop, SAC 2007, Ottawa, Canada, August 16–17, 2007. – Berlin ; Heidelberg : Springer, 2007. – P. 15–27. – (Lecture Notes in Computer Science ; vol. 4876).

72. Zhang W. Related-key differential-linear attacks on reduced AES-192 / W. Zhang et al. // Progress in Cryptology – INDOCRYPT 2007 : proceedings of the 8th International Conference on Cryptology in India, Chennai, India, December 9–13, 2007. – Berlin ; Heidelberg : Springer, 2007. – P. 73–85. – (Lecture Notes in Computer Science ; vol. 4859).

73. Казимиров А. В. Алгебраические свойства схемы разворачивания ключей блочного симметричного шифра «Калина» / А. В. Казимиров, Р. В. Олейников // Радіоелектронні і комп'ютерні системи. – Харків : ХАІ, 2010. – № 5 (46). – С. 61–66.

74. Олейников Р. В. Анализ свойств схемы разворачивания ключей алгоритма шифрования «Калина» / Р. В. Олейников, В. И. Руженцев, В. Д. Дырявый // Безпека інформації в інформаційно-телекомунікаційних системах : матеріали XII Міжнар. наук.-практ. конф., 19–22 трав. 2009 р. – Київ, 2009. – С. 35–36.

75. Принципи побудування та властивості блокового симетричного шифру «Калина» / І. Д. Горбенко, В. І. Долгов, Р. В. Олійников та ін. // Прикладная радиоэлектроника. – 2007. – Т. 6, № 2. – С. 209–216.

76. Криптостійкість шифру «Калина» / Р. В. Олійников, І. Д. Горбенко, В. І. Долгов, В. І. Руженцев // Прикладная радиоэлектроника. – Харьков, 2007. – Т. 6, № 2. – С. 217–229.

77. Олейников Р. В. Анализ цикловых свойств перспективного блочного симметричного алгоритма шифрования «Калина-2» / Р. В.

Олейников, И. Д. Горбенко // Компьютерные науки и технологии (КНиТ-2011) : материалы 2-й Междунар. науч.-техн. конф., 3–7 окт. 2011 г., Белгород, Россия. – Белгород, 2011. – С. 496–500.

78. Методика и результаты исследования криптографической стойкости алгоритма шифрования «Калина» / Р. В. Олейников, И. Д. Горбенко, В. И. Долгов, В. И. Руженцев // Безпека інформації в інформаційно-телекомунікаційних системах : матеріали X Міжнар. наук.-практ. конф., Київ, 16–18 трав. 2007 р. – Київ, 2007. – С. 16–17.

79. Алексейчук А.Н. Оценки практической стойкости блочного шифра Калина относительно методов разностного, линейного криптоанализа и относительно алгебраических атак, основанных на гомоморфизмах / А. Н. Алексейчук, Л. В. Ковальчук, Е. В. Скрыпник, А. С. Шевцов // Прикладная радиоэлектроника. – 2008. – Т. 7, № 3. – С. 203–209.

80. Долгов В.И. Криптографические свойства уменьшенной версии шифра «Калина» / В.И. Долгов, Р.В. Олейников, А.Ю. Большаков, А.В. Григорьев, Е.В. Дроботько // Прикладная радиоэлектроника. – 2010. – Т. 9. – № 3. – С. 349–354.

81. ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення. – Введ. 01–07–2015. – Київ : Мінекономрозвитку України, 2015.

82. Oliynykov R. V. A new approach of key schedule construction for symmetric block ciphers / R. V. Oliynykov, V. I. Ruzhentsev // Известия ЮФУ. Технические науки. – Таганрог : Изд-во ТТИ ЮФУ, 2010. – № 11 (112). – С. 156–161.

83. Menezes A. J. Handbook of applied cryptography [Text] / Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. – CRC press, 2010.

84. Bit-Pattern Based Integral Attack [Text] / Muhammad Reza Z'aba, Håvard Raddum, Matt Henricksen, Ed Dawson // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland,

February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P. 363-381. – (Lecture Notes in Computer Science ; vol. 5086).

85. Biham E. New Techniques for Cryptanalysis of Hash Functions and Improved Attacks on Snefru [Text]/ Eli Biham // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P. 444-461. – (Lecture Notes in Computer Science ; vol. 5086).

86. Bing Sun. New Cryptanalysis of Block Ciphers with Low Algebraic Degree [Text] / Bing Sun, Longjiang Qu, Chao Li // Fast Software Encryption (FSE 2009) : proceedings of the 16th International Workshop, FSE 2009 Leuven, Belgium, February 22-25, 2009. – Berlin ; Heidelberg : Springer, 2009. – P. 180-192. – (Lecture Notes in Computer Science ; vol. 5665).

87. Albrecht M. Algebraic Techniques in Differential Cryptanalysis [Text]/ Martin Albrecht, Carlos Cid // Fast Software Encryption (FSE 2009) : proceedings of the 16th International Workshop, FSE 2009 Leuven, Belgium, February 22-25, 2009. – Berlin ; Heidelberg : Springer, 2009. – P. 193-208. – (Lecture Notes in Computer Science ; vol. 5665).

88. Manuel S. Collisions on SHA-0 in One Hour [Text] / Stéphane Manuel, Thomas Peyrin // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P.16-35. – (Lecture Notes in Computer Science ; vol. 5086).

89. Demirci H. A Meet-in-the-Middle Attack on 8-Round AES [Text] / Hüseyin Demirci, Ali Aydın Selçuk// Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P.116-126. – (Lecture Notes in Computer Science ; vol. 5086).

90. Steinfeld R. Cryptanalysis of LASH [Text] / Ron Steinfeld, Scott Contini, Krystian Matusiewicz, Josef Pieprzyk, Jian Guo, San Ling // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop,

Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P.207-223. – (Lecture Notes in Computer Science ; vol. 5086).

91. A (Second) Preimage Attack on the GOST Hash Function [Text] / Florian Mendel, Norbert Pramstaller, Christian Rechberger // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P.224-234. – (Lecture Notes in Computer Science ; vol. 5086).

92. Chang D. Improved Indifferentiability Security Analysis of chopMD Hash Function [Text] / Donghoon Chang, Mridul Nandi // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P. 429-443. – (Lecture Notes in Computer Science ; vol. 5086).

93. Nikolić I. Collisions for Step-Reduced SHA-256 [Text] / Ivica Nikolić, Alex Biryukov // Fast Software Encryption (FSE 2008) : proceedings of the 15th International Workshop, Lausanne, Switzerland, February 10–13, 2008. – Berlin ; Heidelberg : Springer, 2008. – P.1-15. – (Lecture Notes in Computer Science ; vol. 5086).

94. Daemen J. AES proposal: Rijndael [Text] / J. Daemen, V. Rijmen // First Advanced Encryption Standard (AES) Conference, Ventura, CA, August 20–22, 1998.

95. Biham E. Cryptanalysis of Reduced Variant of Rijndael [Electronic resource] / E. Biham, N. Keller // The Third Advanced Encryption Standard Candidate Conference, New York, USA, April 13–14, 2000. – Mode of access : www. URL: http://csrc.nist.gov/archive/aes/index.html.

96. Sugita M. Relationships among differential, truncated differential, impossible differential cryptanalyses against word-oriented block cipher like Rijndael, E2 [Electronic resource] / M. Sugita, K. Kobara // National Institute of Standards and Technology. – Mode of access : http://www.nist.gov/aes.

97. Aoki K. Practical Evaluation of Security against Generalized Interpolation Attack [Text] / K. Aoki // IEICE Transactions Fundamentals of

Electronics, Communications and Computer Sciences (Japan), Vol. E83-A, No. 1, pp. 33–38, 2000.

98. Аналіз властивостей алгоритмів блокового симетричного шифрування (за результатами міжнародного проекту NESSIE) [Текст] / І. Д. Горбенко, Г. М. Гулак, Р. В. Олійников, В. І. Руженцев та ін. // Радиотехника : Всеукр. межвед. науч.-техн. сб. – Х., 2005. – Вып. 141. – С. 7–24.

99. Порівняльний аналіз блокового симетричного шифрування (за результатами виконання проекту Nessie) [Текст] / Горбенко І.Д., Пушкарьов А.І., Олійников Р.В., Руженцев В.І. та ін. // Радиотехника : Всеукр. межвед. науч.-техн. сб. – Х., 2005. – Вып. 141. – С. 25–30.

100. Разработка требований и принципы проектирования перспективного симметричного блочного алгоритма шифрования [Текст] / И. Д. Горбенко, В. И. Долгов, Р. В. Олейников, В. И. Руженцев и др. // Известия ЮФУ. Технические науки. – Таганрог : Изд-во ТТИ ЮФУ, 2007. – № 1 (76). – С. 183–189.

101. Принципи побудування та властивості блокових симетричних IDEA подібних шифрів [Текст] / Р. В. Олійников, І. Д. Горбенко, В. І. Долгов, В. І. Руженцев // Прикладная радиоэлектроника. – Х., 2007. – Т. 6, № 2. – С. 158–173.

102. Обгрунтування вимог та розробка основних рішень з побудування та властивості перспективного БСШ «Мухомор» [Текст] / Р. В. Олійников, І. Д. Горбенко, В. І. Долгов, В.І. Руженцев [та ін.] // Прикладная радиоэлектроника. – Х., 2007. – Т. 6, № 2. –С. 174–185.

103. Принципи побудування та властивості блокового симетричного шифру «Калина» [Текст] / І. Д. Горбенко, В. І. Долгов, Р. В. Олійников, В.І. Руженцев [та ін.] // Прикладная радиоэлектроника. – 2007. – Т. 6, № 2. – С. 209–216.

104. Олейников Р.В. Построение переопределённой системы уравнений для описания алгоритма шифрования AES [Текст]/ Олейников Р.В., Руженцев В.И., Шумов А.И // VII Міжнародна науково-практична

конференція. Безпека інформації в інформаційно-телекомунікаційних системах**.** Київ,11-14 травня 2004 року. С.91

105. Долгов В.И. Проверка шифров AES и ГОСТ 28147-89 на наличие слабых и эквивалентных ключей [Текст]/ В.И. Долгов, В.И. Руженцев// Сборник научных трудов по материалам 10-й международной конференции “Теория и техника передачи, приема и обработки информации», Харьков: ХНУРЭ, 2004 г. С. 87-88.

106. Руженцев В.И. Об особенностях организации статистических корреляционных атак на блочные симметричные шифры [Текст] / В.И. Руженцев // IX Международная научно-практическая конференция «Безопасность информации в информационно-телекоммуникационных системах», 17-19 мая 2006. тезисы докладов. – К.: НИЦ «ТЕЗИС» НТУУ «КПИ», 2006. – с.34.

107. Руженцев В.И. Особенности организации интерполяционной атаки на блочные симметричные шифры [Текст] / В.И. Руженцев // I-я международная конференция "Глобальные информационные системы. Проблемы и тенденции развития".-Харьков.ХНУРЭ,2006,с.384-385.

108. Biham E. Differential cryptanalysis of DES-like cryptosystems [Text] / E. Biham, A. Shamir // J. of Cryptology. – 1991. – Vol. 4, № 1. – P. 3–72.

109. Biham E. Differential Cryptanalysis of the Data Encryption Standard [Text] / E. Biham, A. Shamir. – Berlin ; Heidelberg : Springer-Verlag, 1993. – 312p.

110. Biham E. Differential cryptanalysis of the full 16-round DES [Text] / E. Biham, A. Shamir // Advances in Cryptology – CRYPTO'92 : proceedings of the 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16–20, 1992. – Berlin ; Heidelberg : Springer, 1993. – P. 487–496. – (Lecture Notes in Computer Science ; vol. 740).

111. The cipher SHARK [Text] / V. Rijmen et al. // Fast Software Encryption. – Berlin ; Heidelberg : Springer, 1996. – P. 99–111.

112. Daemen J. The block cipher Square [Text] / J. Daemen, L.R. Knudsen, V. Rijmen // Fast Software Encryption, LNCS 1267, E. Biham, Ed., Springer-Verlag, 1997, pp. 149–165.

113. Barreto P. The Khazad legacy-level block cipher [Electronic resource] / P. Barreto, V. Rijmen // First Open NESSIE Workshop, Leuven, Belgium, November 13–14, 2000. – Mode of access : www. URL: https://www.cosic.esat.kuleuven.be/nessie/workshop/.

114. Barreto P. The Anubis block cipher [Electronic resource] / P. Barreto, V. Rijmen // First Open NESSIE Workshop, Leuven, Belgium, November 13–14, 2000. – Mode of access : https://www.cosic.esat.kuleuven.be/nessie/workshop/.

115. Beth T. On almost perfect nonlinear permutations [Text] / T. Beth, C. Ding // Advances in Cryptology – Eurocrypt'93 : proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23–27, 1993. – Berlin : Springer-Verlag, 1994. – P. 65–76. – (Lecture Notes in Computer Science ; vol. 765).

116. Nyberg K. Differentially uniform mappings for cryptography [Text] / K. Nyberg // Advances in Cryptology – Eurocrypt'93 : proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23–27, 1993. – Berlin : Springer-Verlag, 1994. – P. 55–64. – (Lecture Notes in Computer Science ; vol. 765).

117. Supporting document on E2 [Electronic resource] // Nippon Telegraph and Telephone Corporation. 1999.– Mode of access : http://info.isl.ntt.co.jp/e2/.

118. Lai X. Markov ciphers and differential cryptanalysis [Text] / X. Lai, J. L. Massey, S. Murphy // Advances in Cryptology – EUROCRYPT'91 : proceedings of theWorkshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8–11, 1991. – Berlin ; Heidelberg : Springer, 1991. – P. 17–38. – (Lecture Notes in Computer Science ; vol. 547).

119. Nyberg K. Provable Security Against a Differential Attack [Text]/ K. Nyberg, L.R. Knudsen // Journal of Cryptology, Vol. 8, No. 1, pp. 27–37, 1995.

120. Matsui M. On correlation between the order of S-boxes and the strength of DES [Text]/ Matsui M. // Advances in Cryptology – Proc. *EUROCRYPT'94*, LNCS 950, pp. 366–375, 1995. Springer–Verlag..

121. Aoki K. Best Differential Characteristic Search of FEAL [Text]/ K. Aoki, K. Kobayashi, S. Moriai // In E. Biham, editor, Fast Software Encryption – 4th International Workshop, FSE'97, Volume 1267 of Lecture Notes in Computer Science, pp. 41-53. Berlin, Heidelberg, New York, 1997. Springer–Verlag.

122. Ohta K. Improving the Search Algorithm for the Best Linear Expression [Text] / K. Ohta, S. Moriai, K. Aoki // In D. Coppersmith, editor, Advances in Cryptology –CRYPTO'95, Volume 963 of Lecture Notes in Computer Science, pp. 157-170. Springer–Verlag, Berlin, Heidelberg, New York, 1995.

123. Руженцев В.И. Методы оценки стойкости блочных симметричных шифров к дифференциальным атакам. Диссертация на соискание ученой степени кандидата технических наук по специальности 05.13.21 – системы защиты информации. Харьковский национальный университет радиоэлектроники, Харьков, 2003.

124. Practical and Provable Security against Differential and Linear Cryptanalysis for Substitution-Permutation Networks [Text] / J-S. Kang, S. Hong, S. Lee et. al. // ETRI Journal, Volume 23, Number 4, December 2001.

125. Ohkuma K. Specification and assessment of the cipher Hierocrypt [Text] / K.Ohkuma, H.Muratani, F.Sano // Technical Report of IEICE ISEC2000-7, 2000.

126. Kanda M. On the security of Feistel cipher with SPN round function against differential, linear and truncated differential cryptanalysis [Text] / M. Kanda, T. Matsumoto // IEICE Trans. Fundamentals, Vol.E-85-A, №1, January 2002.

127. Kleiman E. The XL and XSL attacks on Baby Rijndael [Electronic resource] / E. Kleiman // Thesis, 2005. Mode of access : http://orion.math.iastate.edu/dept/thesisarchive/MS/EKleimanMSSS05.pdf.

128. Руженцев В.И. О влиянии модификаций в шифре Rijndael на вероятности дифференциалов [Текст] / Руженцев В.И. // Материалы конференции Компьютерные науки и технологии "КНиТ-2011". Белгородский государственный национальный исследовательский университет. 2011. С. 513-517.

129. Олейников Р.В. Построение переопределённой системы уравнений для описания алгоритма шифрования AES [Текст]/ Олейников Р.В., Руженцев В.И., Шумов А.И // VII Міжнародна науково-практична конференція. Безпека інформації в інформаційно-телекомунікаційних системах. Київ,11-14 травня 2004 року. С.91

130. Руженцев В.И. О корректности методов оценки стойкости блочных шифров к дифференциальным атакам [Текст] / В.И. Руженцев // 2-а міжнародна наукова конференція “Сучасні інформаційні системи. Проблеми та тенденції розвитку”: Зб. матеріалів конф. – Харків:ХНУРЕ, 2007. С. 439-440.

131. Руженцев В.И. Дифференциальные свойства масштабированных моделей блочных симметричных шифров[Текст] / В.И. Руженцев, В.И. Долгов, Р.В. Олейников// 11-ая Международная научно-практическая конференция «Безопасность информации в информационно-телекоммуникационных системах», 20-23 мая 2008. Тезисы докладов. – К.: ЧП «ЕКМО» НИЦ «ТЕЗИС» НТУУ «КПИ», 2008.

132. Руженцев В.И. О стойкости уменьшенных моделей блочных шифров к линейному криптоанализу [Текст] / В.И. Руженцев, В.И. Долгов // Сб. научн. трудов 3-го межд. Радиоэлектронного форума «Прикладная радиоэлектроника» (МРФ 2008). Харьков. Том 5, 2008. С. 264-265.

133. Руженцев В.И. Сравнительный анализ криптостойкости уменьшенных моделей блочных шифров [Текст] / Руженцев В.И., Долгов В.И. // Праці міжнар. симпозіуму «Питання оптимізації обчислень» (ПОО), 24-29 вересня 2009. Київ: Інститут кібернетики ім. Глушкова НАН України, 2009. Т.1. С. 211-215.

134. Computational aspects of the expected differential probability of 4-round AES and AES-like ciphers [Text] / J. Daemen, M. Lamberger, N. Pramstaller, V. Rijmen, F. Vercauteren // Computing (2009) 85: P. 85–104.

135. Knudsen L. R. Truncated differentials of SAFER [Text] / L. R. Knudsen, T. A. Berson // In Fast Software Encryption - Third International Workshop, FSE'96, Volume 1039 of Lecture Notes in Computer Science, Berlin, Heidelberg, New York, Springer-Verlag, 1996.

136. Matsui M. Cryptanalysis of reduced version of the block cipher E2 [Text] / M. Matsui, T. Tokita // In pre-proceedings of Fast Software Encryption'99, pp. 70-79, 1999.

137. Руженцев, В. И. О методах оценки стойкости к атаке усеченных дифференциалов [Текст] / В. И. Руженцев // Радиоэлектроника и информатика. 2003. №4. С. 130-133.

138. Руженцев В.И. Об условиях отсутствия эффективных усеченных байтовых дифференциалов для блочных симметричных шифров [Текст] / Руженцев В.И. // Радиотехника: Всеукр. межвед. науч.-техн.сб., Харьков, 2014. Вып. 176. С. 55-61.

139. Moriai S. Security of E2 against Truncated Differential Cryptanalysis [Text] / S. Moriai, M. Sugita, K. Aoki // In H. Heys and C. Adams, editors, Selected Areas in Cryptography — 6th Annual International Workshop, SAC’99, Volume 1758 of Lecture Notes in Computer Science, pp. 106–117, Berlin, Heidelberg, New York, Springer-Verlag, 2000.

140. Improved Impossible Differential Cryptanalysis of Rijndael and Crypton [Text]/ Cheon, J.H., Kim, M., Kim, K., Lee, J.-Y., Kang, S. // In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 39–49. Springer, Heidelberg (2002).

141. New Impossible Differential Attacks on AES [Electronic resource] / J. Lu, O. Dunkelman, N. Keller, J. Kim // IACR Cryptology ePrint Archive 2008: 540 (2008).

142. Biham E. Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials [Text] / Biham E., Biryukov A., Shamir A. // Technion,

CS Dept, Tech Report CS0947 (1998).

143. Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1 [Electronic resource] / Lu, J., Kim, J., Keller, N., Dunkelman, O. // Mode of access : http://jiqiang.googlepages.com.

144. Wu W. Impossible differential cryptanalysis of reduced-round ARIA and Camellia [Text] / W. Wu, W. Zhang, D. Feng. // Journal of Computer Science and Technology, 22(3):449-456, 2007. Springer.

145. Impossible differential cryptanalysis for block cipher structures [Text] / J. Kim, S. Hong, J. Sung, S. Lee, J. Lim // INDOCRYPT 2003, LNCS 2904, pp. 82-96, 2003.

146. A Unified Method for Finding Impossible Differentials of Block Cipher Structures [Electronic resource] / Y. Luo, Z. Wu, X. Lai, G. Gong // IACR Cryptology ePrint Archive 2009: 627 (2009).

147. Li R. Impossible Differential Cryptanalysis of SPN Ciphers [Electronic resource] / Ruilin Li, Bing Sun, Chao Li // IACR Cryptology ePrint Archive 2010: 307 (2010).

148. Yap H. Impossible Differential Characteristics of Extended Feistel Networks with Provable Security against Differential Cryptanalysis [Text]/ H. Yap // SecTech 2008, CCIS 29, pp. 103-121, 2009.

149. Biham E. Miss in the Middle Attacks on Idea and Khufu [Text] / E. Biham, A. Biryukov, A. Shamir // Fast Software Encryption : proceedings of the 6th International Workshop, FSE'99, Rome, Italy, March 24–26, 1999. – Berlin ; Heidelberg : Springer, 1999. – P. 124–138. – (Lecture Notes in Computer Science ; vol. 1636).

150. Provable security against differential and linear cryptanalysis for the SPN structure [Text] / Hong, S., Lee, S., Lim, J., Sung, J., and Cheon, D. // Proc. Fast Software Encryption (FSE 2000), LNCS, 1978, 2001, edited by Schneier, B., (Springer), pp. 273–283.

151. Руженцев В.И. Особенности реализации наиболее эффективных криптоаналитических атак на стандарт шифрования FIPS-197 [Текст] /

В.И. Руженцев, Р.В.Олейников // Радиотехника : Всеукр. межвед. науч.-техн. сб. – Х., 2012. – Вып. 169. – С. 208–215.

152. Daemen J. Two-Round AES Differentials [Electronic resource]: report 2006/039 / J. Daemen, V. Rijmen // Cryptology ePrint Archive. – 2006. – Mode of access : http://eprint.iacr.org/2006/039.

153. Руженцев В.И. Оценка вероятностей двухцикловых дифференциалов шифра AES [Текст] / Руженцев В.И. // Прикладная радиоэлектроника. Тематический выпуск, посвященный проблемам обеспечения безопасности информации. Харьков. Том 10, №2, 2011 г. С. 116-121.

154. Руженцев В.И. Вероятности двухцикловых дифференциалов rijndael-подобного шифра с произвольными подстановками [Текст] / Руженцев В.И. // Прикладная радиоэлектроника. Тематический выпуск, посвященный проблемам обеспечения безопасности информации. Харьков. Том 13, №3, 2014 г. С. 235-238.

155. Руженцев В.И. О двухцикловых дифференциалах шифра AES / В.И. Руженцев // Сб. научн. трудов VI межд. научн.-практ. конф. «Наука и социальные проблемы общества: информатизация и информационные технологии». – Харьков: ХНУРЕ. 2011. С. 256-257.

156. Перспективний блоковий симетричний шифр «Калина» – основні положення та специфікація [Текст] / І. Д. Горбенко, В. І. Долгов, Р. В. Олійников, В.І. Руженцев [та ін.] // Прикладная радиоэлектроника. – Х., 2007. – Т. 6, № 2. – С. 195–208.

157. Державна служба спеціального зв'язку та захисту інформації України, Інститут кібернетики імені В.М. Глушкова Національної академії наук України. Положення про проведення відкритого конкурсу криптографічних алгоритмів. [Electronic resource] Mode of access : www. URL: http://www.dstszi.gov.ua/dstszi/control/uk/publish/printable_article?art_id=48383.

158. Ruzhentsev V. I. Towards provable security of Rijndael-like SPN ciphers against differential attacks [Text] / V. I. Ruzhentsev , V. I. Dolgov // Tatra Mountains Mathematical Publications. – 2012. – Vol. 53. – P. 189–199.

159. Руженцев В.И. Доказуемая стойкость rijndael-подобных шифров к атаке усеченных дифференциалов [Текст] / Руженцев В.И. // Науково-технічний журнал: Радіоелектронні і комп'ютерні системи. ХАИ. 2012. №5. С. 51-55.

160. Ruzhentsev V. I. Towards provable security of Rijndael-like SPN ciphers against differential attacks [Text]/ V.I. Ruzhentsev, V.I. Dolgov // Abstracts of papers presented at Tatracrypt 2012, The 12th Central European Conference on Cryptology, P. 40-41, Congress Center of SAS, Smolenice, Slovakia July 2-4, 2012.

161. Руженцев В.И. Анализ стойкости Rijndael-подобных шифров с большим размером блока к атаке усеченных дифференциалов [Текст] / Руженцев В.И. // ВІСНИК Національний університет кораблебудування імені адмірала Макарова №1, 2013. С. 44-51.

162. Руженцев В.И. Анализ стойкости Rijndael-подобных шифров с большим размером блока к атаке усеченных дифференциалов [Текст]/ В.И. Руженцев // Материалы всеукраинской науч.-техн. конф. с международным участием «Современные проблемы информационной безопасности в транспорте». – Николаев: НУК, 2012. С. 76-78.

163. Руженцев В.И. Об условиях отсутствия эффективных усеченных байтовых дифференциалов для блочных симметричных шифров [Текст] / Руженцев В.И. // Радиотехника: Всеукр. межвед. науч.-техн.сб., Харьков, 2014. Вып. 176. С. 55-61.

164. Ruzhentsev V. The conditions of provable security of block ciphers against truncated differential attack [Text]/ V. Ruzhentsev // Studia Scientiarum Mathematicarum Hungarica. Vol. 52 (2). Budapest. 2015. P. 176–184.

165. Ruzhentsev V. I. The conditions of provable security of block ciphers against truncated differential attack [Text] / V.I. Ruzhentsev // Pre-proceding of the

14th Central European Conference on Cryptology, P. 103-108, Alfred Renyi Institute of Mathematics, Budapest, Hungary May 21-23, 2014.

166. T. Jakobsen. The Interpolation Attack on Block Cipher. [Text] / T. Jakobsen, L. R. Knudsen // In E. Biham, editor, Fast Software Encryption — 4th International Workshop, FSE'97, Volume 1267 of Lecture Notes in Computer Science, pp. 28–40, Berlin, Heidelberg, New York, 1997. Springer-Verlag.

167. Горбенко И.Д. Алгоритм блочного симметричного шифрования «Торнадо». Спецификация преобразования [Текст] / Горбенко И.Д., Головашич С.А. // Радиотехника. 2003. № 134. С. 60-80.

169. Головашич, С. А. Спецификация алгоритма блочного симметричного шифрования «Лабиринт» [Текст] / С. А. Головашич // Прикладная радиоэлектроника. Тематический выпуск, посвященный проблемам обеспечения безопасности информации. Харьков. Том 6, №2, 2007 г. С. 230-240.

170. Руженцев В.И. О методах доказательства стойкости блочных шифров к атаке невыполнимых дифференциалов [Текст] / Руженцев В.И. // Праці міжнародної наукової конференції „Питання оптимізації обчислень (ПОО-XL)". Київ: Інститут кібернетики імені В.М. Глушкова НАН України, 2013. 292 с. С. 224-225.

171. Руженцев В.И. О методе доказательства стойкости блочных шифров к атаке невыполнимых дифференциалов [Текст] / Руженцев В.И. // Прикладная радиоэлектроника. Тематический выпуск, посвященный проблемам обеспечения безопасности информации. Харьков. Том 12, №2, 2013 г. С. 215-219.

172. Junod P. FOX: a new family of block ciphers [Text] / P. Junod, S. Vaudenay // Selected Areas in Cryptography. – Berlin ; Heidelberg : Springer, 2005. – P. 114–129.

173. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Computing 26 (1997) 1484-1509.

174. J. B. Fraleigh, A first course in abstract algebra, Pearson Education India, 1971.

175. D. R. Stinson, Cryptography: Theory and Practice, Third Edition // Chapman &Hall, Boca Rato. – 2005.

176. N. R. Wagner, M. R. Magyarik, A public key cryptosystem based on the word problem, in Advances in Cryptology – CRYPTO ’84 (G.R. Blakley, D. Chaum, eds.), Lecture Notes in Computer Science 196 (Springer, Berlin, 1985) p.19-36.

177. M. I. Gonzarlez Vasco, R. Steinwandt, A reaction attack on a public key cryptosystem based on the word problem, Applicable Algebra in Engineering, Communication and Computing 14. – 2004. – p.335-340.

178. F. Levy-dit-Vehel, L. Perret, On the Wagner–Magyarik cryptosystem, Coding and Cryptography (O. Ytrehus, ed.). – Springer. – Berlin. – 2006. – p.316–329.

179. F. Levy-dit-Vehel, L. Perret, Security analysis of word problem-based cryptosystems, Des. Codes Cryptography 54 (1) (2010) 29-41.

180. M. Dehn, UË ber unendliche diskontinuerlich gruppen, Math. Ann. 69(1911) p.116–144.

181. P. S. Novikov, On the algorithmic unsolvability of the word problem in group theory, Trudy Mat. Inst. Steklov 44 (1955) p.1–143.

182. W. W. Boone, Certain simple unsolvable problems in group theory, I–VI, Nederl. Akad. Wetensch Proc. Ser. A57,231–237, 492–497 (1954), 58, 252–256, 571–577 (1955), 60, 22–27, 227–232 (1957).

183. I. Anshel, M. Anshel, D. Goldfeld, An algebraic method for public-key cryptography, Math. Res. Lett. 6. – 1999. – p.287-291.

184. W. Diffie and M. E. Hellman, New directions in cryptography, IEEE Trans. Information Theory 22. – 1976. – p.644–654.

185. A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, 1997.

186. A. Myasnikov, V. Shpilrain, A. Ushakov, Group-based Cryptography, Advanced Courses in Mathematics CRM Barcelona. – BirkhaËuser. – Basel. – 2008.

187. R. Wernsdorf, The one-round functions of the DES generate the alternating group, Advances in Cryptology – EUROCRYPT 1992 (R.A. Rueppel, ed.), Lecture Notes in Computer Science 658 (Springer–Verlag, Berlin, 1993). – P. 99–112.

188. K. W. Campbell, M. J. Wiener, DES is not a group, Advances in Cryptology – CRYPTO ’92 (E.F. Brickell, ed.), Lecture Notes in Computer Science 740 (Springer–Verlag, Berlin, 1993). – p.512–520.

189. D. Coppersmith, The Data Encryption Standard (DES) and its strength against attacks, IBM Research Report RC 18613 (IBM, 1992).

190. R. Sparr and R. Wernsdorf, Group theoretic properties of RIJNDAEL-like ciphers, Discrete Appl. Math. 156. – 2008 – p.3139–3149.

191. R. Wernsdorf, The round functions of RIJNDAEL generate the alternating group FSE. – 2002. – p.143–148.

192. G. Zermor, Hash functions and Cayley graphs, Des. Codes Cryptography, 4(4) . – 1994. – p.381–394.

193. J.P. Tillich, G. Zermor, Hashing with SL2. Advances in Cryptology - CRYPTO ’94 (Y. Desmedt, ed.), Lecture Notes in Computer Science 839 (Springer-Verlag, 1991). – p.508–511.

194. F.A. Garside, The braid group and other groups, Quart. J. Math. Oxford 20 (1969). – p.235–254.

195. J. S. Birman, V. Gebhardt, J. Gonzarlez-Meneses, Conjugacy in Garside groups I: cycling, powers and rigidity, Groups Geom. Dynamics 1 (2007) p.221–279.

196. J. S. Birman, V. Gebhardt, J. Gonzarlez-Meneses, Conjugacy in Garside groups II: structure of the ultra-summit set, Groups Geom. Dynamics 2. – 2008. – p.13–61.

197. J. S. Birman, V. Gebhardt, J. Gonzarlez-Meneses, Conjugacy in Garside groups III: periodic braids, J. Algebra 316. – 2007. – p.746–776.

198. D. Hofheinz, R. Steinwandt, A practical attack on some braid group based cryptographic primitives, Public Key Cryptography – PKC 2003 (Y.G. Desmedt, ed.), Lecture Notes in Computer Science 2384. – Springer, Berlin, 2002. – p.176–189.

199. J. Hughes, A. Tannenbaum, Length-based attacks for certain group based encryption rewriting systems, available at http://arxiv.org/PS_cache/cs/pdf/0306/0306032v1.pdf.

200. D. Garber, S. Kaplan, M. Teicher, B. Tsaban, U. Vishne, Probabilistic solutions of equations in the braid group, Adv. Appl. Math. 35. – 2005. – p.323–334.

201. A. D. Myasnikov, A. Ushakov, Length based attack and braid groups: cryptanalysis of Anshel-Anshel-Goldfeld key exchange protocol, Public Key Cryptography – PKC 2007 (T. Okamoto, X. Wang, eds.), Lecture Notes in Computer Science 4450. – Springer, Berlin. – 2007. – p.76–88.

202. J. Hughes, A linear algebraic attack on the AAFG1 braid group cryptosystem, Lecture Notes in Computer Science 2384. – 2002. – p.176–189.

203. Sang Jin Lee and Eonkyung Lee, Potential weaknesses of the commutator key agreement protocol based on braid groups, Advances in Cryptology – EUROCRYPT 2002 (L. Knudsen, ed.), Lecture Notes in Computer Science 2332. – Springer, Berlin. – 2002. – p.14–28.

204. J. H. Cheon, B. Jun, A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem, Advances in Cryptology – CRYPTO 2003 (D. Boneh, ed.), Lecture Notes in Computer Science 2729. – Springer, Berlin. – 2003. – p.212–225.

205. T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory 31. –1985. – p.469–472.

206. G. Higman, Suzuki 2-groups. Ill. J. Math. 7. – 1963. – p.79–96.

207. B. Huppert, Endliche Gruppen I. – Springer, Berlin. – 1967.

208. B. Huppert, N. Blackburn, Finite Groups II. – Springer, Berlin. – 1982.

209. S.S. Magliveras, A cryptosystem from logarithmic signatures of finite groups. In Proceedings of the 29th Midwest Symposium on Circuits and Systems. – Elsevier, Amsterdam. – 1986. – pp.972–975.

210. S.S. Magliveras, N.D. Memon, The algebraic properties of cryptosystem PGM. J. Cryptol. 5. – 1992. – p.167–183.

211. S.S. Magliveras, D.R. Stinson, T. van Trung, New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups. J. Cryptol. 15. – 2002. – p.285–297.

212. P. Nguyen, Editor, New Trends in Cryptology, European project "STORK—Strategic Roadmap for Crypto"—IST-2002-38273. http://www.di.ens.fr/~pnguyen/pub.html#Ng03.

213. P. Shor, Polynomial time algorithms for prime factorization and discrete logarithms on quantum computers. SIAM J. Comput. 26(5) . – 1997. – p. 1484–1509.

214. S.S. Magliveras, D.R. Stinson, T. van Trung, New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups. J. Cryptol. – 2002. – Vol.15. – p. 285–297.

215. Lempken W., Magliveras S.S., Tran van Trung and Wei W., A public key cryptosystem based on non-abelian finite groups, J. of Cryptology. – 2009. – Vol.22. – p.62–74.

216. Higman G. Suzuki 2-groups. Ill. J. Math. – 1963. – №.7. –p. 79–96.

217. Hanaki A. A condition on lengths of lengths of conjugacy classes and character degrees, Osaka J. Math. – 1996. – Vol.33. – p. 207-216.

218. Google accelerates end of SHA-1 support [Электронный ресурс]. – Режим доступа http://www.zdnet.com/article/google-accelerates-end-of-sha-1-support-certificate-authorities-nervous/.

219. Windows Enforcement of Authenticode Code Signing and Timestamping. [Электронный ресурс]. – Режим доступа:

http://social.technet.microsoft.com/wiki/contents/articles/32288.windows-enforcement-of-authenticode-code-signing-and-timestamping.aspx

220. SHA-1 Depreciation Update [Электронный ресурс]. – Режим доступа: https://blogs.windows.com/msedgedev/2015/11/04/sha-1-deprecation-update/

221. Federal Register Notice published on November 2, 2007. – [Электронный ресурс]. – Режим доступа: http://csrc.nist.gov/groups/ST/hash.

222. NIST Computer security resource center. Announcing request for Candidate algorithm nominations for a new cryptographic hash algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. – 2007. [Электронный ресурс]. – Режим доступа :http://csrc.nist.gov/groups/ST/.

223.Bellare M. Keying hash functions for message authentication / M. Bellare, R. Canetti, H. Krawczyk // Advances in Cryptology, Proceedings Crypto'96, LNCS 1109, N. Koblitz, Ed., Springer-Verlag. – 1996. – P.1-15.

224. ISO/IEC 10118-1. Information technology – Security techniques – Hash-functions – Part 1: General. [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber.

225. Kabatianskii G. On the Cardinality of Systematic Authentication Codes Via Error-Correcting Codes / G. Kabatianskii, B. Smeets, T. Johansson // IEEE Transactions on Information Theory. – 1991. – Vol. IT-42. – P.583-602.

226. International Organization for Standardization, “ISO/IEC 9797-1: Information Technology – Security Techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher", 1999. [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/catalogue_detail.htm? csnumber

227. International Organization for Standardization, \ISO/IEC 9797-2: Information Technology – Security Techniques – Message Authentication Codes (MACs) – Part 2: “Mechanisms using a dedicated hash-function", 2002. [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/catalogue.

228. ДСТУ ГОСТ 28147:2009 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». [Электронный ресурс]. – Режим доступа: www.csm.kiev.ua.

229. Халимов Г. Стойкий к коллизиям алгоритм Whirpool / Г. Халимов, Е.Котух // Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні Київ. – 2005. – № 10. – С.159-165.

230. ISO/IEC 10118-3 Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions. [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm.

231. FIPS 180-2 : Secure hash standard — NIST, 2002

232. ANSI X9.30: American National Standard for Financial Institution Message Authentication (Wholesale) - American Bankers Association, 1986

233. ANSI X9.31: American National Standard for Financial Institution Message Authentication (Wholesale) - American Bankers Association, 1986

234. ISO/IEC 10118-2. Information technology – Security techniques – Hash-functions – Part 2: Hash-functions using an n-bit block cipher. [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/iso_catalogue/catalogue_tc/ catalogue_detail.htm.

235. ISO/IEC 10118-4 Information technology – Security techniques – Hash-functions. – Part 4: Hash-functions using modular arithmetic. [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/iso_catalogue/catalogue_tc/ catalogue_ detail.htm.

236. Стандарты хеш-функций. Основные требования. – [Электронный ресурс]. – Режим доступа: http://vunivere.ru/work23768/page4

237. Bierbrauer J. Universal hashing and geometric codes, to appear in Designs / J. Bierbrauer // Designs, Codes and Cryptography. – 1997. – N.11. – P.207-221.

238. Горбенко И.Д. TWO-TRACK-MAC алгоритм высокой защиты / И.Д. Горбенко, О.Г. Халимов // Радиотехника. – 2005. – Вып. 141. – С. 161-167.

239. Котух Е. Скоростное универсальное хеширование на основе многопотоковых вычислений / Е. Котух, В. Карташов, Д. Цапко, О. Халимов, А. Самойлова // Захист інформації. – 2015. – Т.17. – №2. – 9с.

240. Корченко А.Г. Построение систем защиты информации на нечетких множествах / А.Г. Корченко. – К. : МК-Пресс, 2006. – 320 с.

241. Simmons G.J. Authentication theory/coding theory, in Advances in Cryptology / G.J. Simmons // Proceedings of Crypto 84, Lecture Notes in Computer Science. – 1985. – V.196. – P.411-431.

242. Fleischmann E. Classification of the SHA-3 Candidates / E.Fleischmann, C.Forler, M.Gorski. – 2008. [Электронный ресурс] – Режим доступа: http://eprint.iacr.org/2008/511.pdf.

243. Imad Fakhri Alshaikhli, Comparison and analysis study of SHA-3 finalists/ Imad Fakhri Alshaikhli, Mohammad A. Alahmad Khanssaa Munthir // 2012 International Conference on Advanced Computer Science Applications and Technologies, Proceedings 2012 p.366-371

244. Халимов Г.З. Алгоритм универсального хеширования по кривой Судзуки / Г.З. Халимов, Е.В. Котух // Восточно-Европейский журнал передовых технологий. – 2011. – № 3/9 (51). – С. 10-16.

245. Горбенко И.Д. Анализ безопасности международного стандарта ISO/IEC 9797-1 / И.Д. Горбенко, Г.З. Халимов, А.В. Потий, Е.В. Котух, А.В. Дунь // Прикладная радиоэлектроника. – 2007. – Т. 6, №2. – С.250-256.

246. Котух Е.В. Универсальное хеширование с ограничением функционального поля алгебраических кривых // Радиотехника. – 2012. – Вып. 171. – С. 109-115.

247. Blake submission package [Електронний ресурс] - Режим доступу: http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/documents/Blake_FinalRnd.zip

248. SHA-3 Proposal Blake. / Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan; NIST. Candidate to the NIST Hash Competition. 2008.

249. Groestl a SHA-3 candidate / Soren Steffen Thomsen, Martin Schlaffer, Christian Rechberger, Florian Mendel, Krystian Matusiewicz, Lars R. Knudsen, Praveen Gauravaram; NIST. Candidate to the NIST Hash Competition. - 2011.

250. The Hash Function JH / Hongjun Wu; NIST. Candidate to the NIST Hash Competition. 2008.

251. The Skein Hash Function Family / Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, Jesse Walker; NIST. Candidate to the NIST Hash Competition. 2010.

252. Skein submission package [Електронний ресурс] - Режим доступу: http://csrc.nist.gov/groups/ST/hash/sha-3/Round3/documents/Skein_FinalRnd.zip

253. The Keccak SHA-3 submission/Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche; NIST. Candidate to the NIST Hash Competition. 2011.

254. Pierre-Louis Cayrel, Gerhard Hoffmann, Michael Schneider. GPU Implementation of the Keccak Hash Function Family. Interntaional Journal of Security and Its Applications. Vol 5. No. 4, October 2011

255. Carter J. L. Universal classes of hash functions / J. L. Carter, M.N. Wegman // Journal of Computer and Systems Science. – 1979. – V.18. – P.143-154.

256. Wegman. M. N. New hash functions and their use in authentication and set equality / M.N. Wegman, J.L. Carter // Journal of Computer and Systems Science. – 1981. – V. 22. – P. 265–279.

257. Халимов Г.З. Асимптотические границы вероятности коллизии для MAC с алгебраическим кодированием / Г.З. Халимов, А.В. Дунь // Восточно-европейский журнал передовых технологий. – Х., 2007. – Вып. 5/2(29). – С.23-26.

258. Stinson D.R. Combinatorial techniques for universal hashing / D.R. Stinson // Journal of Computer and Systems Science. – 1994. – V. 48. – P. 337-346.

259. Stinson D.R. Universal hashing and authentication codes / D.R. Stinson // Designs, Codes and Cryptography. – 1994. – N.4. – P.369–380.

260. Халимов Г.З. Аутентификация и универсальное хеширование/ Г.З. Халимов, А.А. Кузнецов // Радиотехника. – 2001. – Вып. 119. – С. 88-94.

261. Black, J. UMAC: Fast and secure message authentication / J. Black, S. Halevi, H. Krawczyk, T. Krovetz and P. Rogaway // In Advances in Cryptology ‘CRYPTO '99, of Lecture Notes in Computer Science, Springer-Verlag. – 1999. – vol. 1666. – P. 216-233.

262. Krovetz T. UMAC / T. Krovetz, J. Black, S. Halevi, A. Hevia, H. Krawczyk and P. Rogaway // Primitive submitted to NESSIE. – 2000. – Sept. – P. 157-160.

263. Bierbrauer J. On families of hash functions via geometric codes and concatenation. / J. Bierbrauer, T. Johansson,G. Kabatianskii, B. Smeets // Advances in Cryptology-CRYPTO ’93 Proceedings, Springer-Verlag. – 1994. – P. 331-342.

264. Stinson D.R. On the connections between universal hashing, combinatorial designes and error-correcting codes / D.R. Stinson // Congressus Numerantium. – 1996. – V.114. – P. 7-27.

265. Халимов Г.З. Высокоскоростной UMAC алгоритм / Г.З. Халимов // Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні. – НТУУ “КПІ” МОНУ, ДСТСЗІСБУ. – 2005. – Вип.11. – С.167-173.

266. Гоппа В.Д. Коды на алгебраических кривых / В.Д. Гоппа // Докл. АН СССР. – 1981. – Т.259, № 6. – С. 1289-1290.

267. Цфасман М.А. Коды Гоппы, лежащие выше границы Варшамова – Гилберта / М.А. Цфасман // Проблемы передачи информации. – 1982. – Т.18, №3. – С. 3-6.

268. Халимов Г.З. Криптоанализ прямой атаки на универсальное семейство хеш-функций с алгебраическим кодированием / Г.З. Халимов, А.Ю. Иохов, А.В. Северинов // Системи обробки інформації. – Харьков : ХВУ, 2004. – Вып. 12(40). – С. 238-247.

269. Халимов Г. З. Методы и средства аутентификации многоадресного источника данных / Г.З. Халимов, А.В. Дунь // Прикладная радиоэлектроника. -2007. – №3. – С.377-384.

270. Kurosawa K. Almost k-wise independent sample spaces and their cryptologic applications / K. Kurosawa, T. Johansson, D. Stinson // Lecture Notes in Computer Science. – 1997. – N. 1233. – P.409-421.

271. Alon N. Simple constructions of almost k-wise independent random variables / N. Alon, O. Goldreich, J. Hastad, R. Peralta // Random Structures and Algorithms. – 1992. – N.3. – P. 289-304.

272. Халимов Г.З. Оценка имитостойкости систем с композиционной схемой аутентификации / Г.З. Халимов, А.А. Кузнецов, А.В. Северинов, А.Д. Буханцов // Зб. наук. праць ХВУ. – Харків : ХВУ. – 2001. – Вип. 5(35). – С.113-115.

273. Bierbrauer J. Weakly biased arrays, almost independent arrays and error-correcting codes / J. Bierbrauer, H. Schellwat // Publication in Proceedings of AMS-DIMACS, 2000. – P.33.

274. Халимов Г.З. Безусловная аутентификация с использованием слабосмещенных массивов / Г.З. Халимов // Радиотехника. – 2003. – №134. – С.165-171.

275. Torres F. Notes on Goppa codes / F. Torres // Report Series: IMECC-UNICAMP, Cx. P. 6065, Campinas, 13083-970-Sp-Brazil. – 2000. – P.17.

276. Халимов Г.З. Коллизионные оценки универсального хеширования на основе схем с алгебраическими кодами / Г.З. Халимов // Прикладная радиоэлектроника. – 2009. – Т. 8, вып. 3. – С.338-342.

277. Халимов Г.З. Аутентификация с применением алгеброгеометрических кодов / Г.З. Халимов, А.А. Кузнецов // Радиотехника. – 2001. – Вып. 119. – С.103-109.

278. Халимов Г.З. Аутентификация с применением эрмитовых кодов / Г.З. Халимов, А.Ю. Иохов // Вестник ХПИ. – Х. : НТУ „ХПИ”, 2005. – Вып. 9. – С. 26-32.

279. Халимов Г.З. Оценка параметров кривых Ферма для универсального хеширования в простом поле / Г.З. Халимов // Материалы науч.-техн. конф. с международным участием. Компьютерное моделирование в наукоемких технологиях (Ч. 2). КМНТ. – Харьков, 18–21 мая 2010. – С.266.

280. Халимов Г.З. Максимальные кривые Гурвица для целей универсального хеширования / Г.З. Халимов // Материалы XI Междунар. науч.-практ. конф. «Информационная безопасность» (Таганрог, Россия, 23–25 июня 2010), ТТИ ЮФУ. – 2010. – Ч. 3. – С. 144-146.

281. Халимов Г.З. Универсальное хеширование по максимальным кривым / Г.З. Халимов // XIII Междунар. науч.-практ. конф. «Безопасность информации в информационно-телекомуникационных системах», Киев, 18–21 мая 2010г. ; тезисы докладов. – С.53.

282. Weil A. Courbes alg´ebriques et vari´et´es abeliennes / A. Weil // Hermann, Paris, 1971. – P.301.

283. Ihara Y. Some remarks on the number of rational points of algebraic curves over finite fields / Y. Ihara // J. Fac. Science. Tokio. – 1981. – N.28. – P. 721-724.

284. Carlitz L. Bounds for exponential sums / L. Carlitz, S. Uchiyama // Duke Mathematical Journal. – 1957. – N.24. – P.37-41.

285. Vladut S.G. Number of points of an algebraic curve / S.G. Vladut , V.G.Drinfeld // Function Analysis. – 1983. – N.17(1). – P.68–69.

286. Giulietti M. A new family of Fq2-maximal curves / M. Giulietti, G. Korchmaros // preprint [Электронный ресурс]. – 2007. – Режим доступа:www.math.u-bordeaux.fr/.../rocher09.pdf

287. Krovetz T. Fast universal hashing with small keys and no preprocessing: The PolyR construction. / T. Krovetz, P. Rogaway // In Information Security and Cryptology – ICICS 2000. Springer-Verlag. – 2000. – P.73–89.

288. Hansen, J.P. Deligne-Lusztig varieties and group codes / J.P. Hansen // Lecture Notes of Mathematics. – 1992. – V.1518. – P.63-81.

289. Ruck H.G. A characterization of Hermitian function fields over finite fields / H.G. Ruck, H. Stichtenoth // J. reine angew. Mathematics. – 1994. – V.457. – P.185-188.

290. Fuhrmann R. The genus of curves over finite fields with many rational points / R. Fuhrmann, F. Torres // Manuscripta Mathematics. – 1996. – N.89. – P.103-106.

291. Torres F. The Deligne-Lusztig curve associated to the Suzuki group / F. Torres [Электронный ресурс] arXiv:alg-geom/9706012v1 26Jun 1997.

292. Hoholdt N. Algebraic geometry codes. In the Handbook of Coding Theory / N. Hoholdt, J.H.van Lint and R. Pellican // V.S. Pless, W.C. Huffman and R.A. Brualdi Eds., Elsevier, Amsterdam. – 1998. – V.1. – P.871-961.

293. Korchmaros G. On the genus of a maximal curve / G. Korchmaros, F. Torres // Mathematics Annatatsiya. – 2002. – V.323(3). – P.589-608.

294. Lauter K. Improved upper bounds for the number of rational points on algebraic curves over finite fields / K. Lauter // C.R. Academy Science Paris. – 1999. – V.328(12), Serie I. – P.1181-1185.

295. Garcia A. On subfields of the Hermitian function field / A. Garcia, H. Stichtenoth and C.P. Xing // Composition Mathematics. – 2000. – V.120. – P.137-170.

296. Deligne P. Representations of reductive groups over finite fields / P. Deligne, Lusztig // Annatatsiya of Mathematics. – 1976. – N.103. – P.103-161.

297. Халимов Г.З. Оценка параметров кривых Ферма в расширенном поле для универсального хеширования / Г.З. Халимов, А.В. Леншин // Защита информация. Сб. науч. тр. НАУ, Киев. – 2010. – Вып.17. – С.116-120.

298. Халимов Г.З. Оценка параметров кривых Ферма в расширенном поле для универсального хеширования / Г.З. Халимов, А.В. Леншин // Наук.-практ. конф. «Захист інформації в інформаційно-комунікаційних системах», 24–26.05.2010, Київ. Зб. тез доповідей. – 2010. – С.102.

299. Халимов Г.З. Кривые Ферма с большим числом точек в расширенных конечных полях / Г.З. Халимов // Системи обробки інформації. МО України. Харк. ун-т Повітряних Сил ім. Івана Кожедуба. – 2011. – Вип. 4(94). – С.146-150.

300. Халимов Г.З. Двухкаскадное универсальное хеширование с использованием АГ кодов / Г.З. Халимов, А.Ю. Иохов // Восточно-европейский журнал передовых технологий. – 2005. – Вып. 2/2 (14). – С. 115-119.

301. Котух Е.В. Универсальное хеширование по кривым, ассоциированным с группой Судзуки / Е.В. Котух, Г.З. Халимов // Прикладная радиоэлектроника. – 2015. – Т.14, №4. – С.361 – 365.

302. Beelen P. The Newton polygon of plane curves with many rational points. / P. Beelen, R. Pellikan // Designs, Codes and Cryptography. – 2000. – N.21. – P.41-67.

303. Pedersen J.P. A function field related to the Ree group / J.P. Pedersen // Lecture Notes Mathematics. – 1992. – V.1518. – P.122-131.

304. Халимов Г.З. Универсальное хеширование по кривым Сузуки / Г.З. Халимов, Е.В. Котух // Прикладная радиоэлектроника. – 2011. – Т.10, № 2. –С.164-170.

305. Халимов Г.З. Функциональное поле кривой Судзуки для универсального хеширования / Г.З. Халимов, Е.В. Котух // Междунар. науч.-практ. конф. «Застосування інформаційніх технологій у підготовці та діяльності сил охорони правопорядку». Академія внутрішніх військ МВС України 17 – 18.03.2011 : тези доповідей. – 2011. – С.45-48.

306. Халимов Г.З. Универсальное хеширование по рациональным функциям максимальной кривой третьего рода / Г.З. Халимов // Радиотехника. – 2011. – Вып. 165. – С.218-224.

307. Eid A. Suzuki-invariant codes from the Suzuki curve. / A.Eid, H.Hasson, A.Ksir, J.Peachey // arXiv: 1411.6215v2[math.AG] 25 Nov 2014

308. Влэдуц С.Г. Алгеброгеометрические коды. Основные понятия / С.Г. Влэдуц, Д.Ю. Ногин, М.А. Цфасман. – М. : МЦНМО, 2003. – 504с.

309. Torres F. Plan maximal curves / F. Torres // Acta Arithmetic. – 2001. – Vol. 98, No. 2. – P. 165-179.

310. Халимов Г.З. Асимптотические оценки максимальных кривых для универсального хеширования / Г.З. Халимов // Междунар. науч.-практ. конф. «Застосування інформаційніх технологій у підготовці та діяльності сил охорони правопорядку» ; Академія внутрішніх військ МВС України 17–18.03.2011. Зб. тез. доповідей. – 2011. – С.43-44.

311. Халимов Г.З. Универсальное хеширование по алгебраическим кривым в простом поле / Г.З. Халимов // XIV Междунар. науч.-практ. конф. ”Безопасность информации в информационно-телекомуникационных системах”, Киев, 18 – 21 мая 2010. Тезисы докладов. – 2011. – С.22-23.

312. Халимов Г.З. Оценка параметров кривых Ферма для универсального хеширования / Г.З. Халимов // Радіоелектроніка, інформатика, управління. – Запоріжжя : Запоріжський національний університет, 2011. – №1(24). – С.82-86.

313. Халимов Г.З. Универсальное хеширование хеширование по максимальным кривым Ферма в квадратичном поле / Г.З. Халимов // XII Міжнар. наук.-практ. конф. «Безпека інформації в інформаційно-телекомунікаційних системах», Київ, 19–22.05.2009. Зб. тез доповідей. – 2009. – С.32.