НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичних методів захисту інформації

«До захисту допущено»
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності: 113 Прикладна математика
на тему: **«Побудова загальних атак відновлення раундової функції на Фейстель-подібні шифри з малою областю визначення»**

Виконав: студент 4 курсу, групи ФІ-83
Мельниченко Олексій Сергійович

Керівник: к.ф.-м.н., ст. викладач Фесенко А.В. _____

Консультант: _ _____

Рецензент: к.т.н., доцент кафедри ІБ Стьопочкіна І.В. _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2022 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Мельниченко Олексій Сергійович

1. Тема роботи: «*Побудова загальних атак відновлення раундової функції на Фейстель-подібні шифри з малою областю визначення*»,

керівник: к.ф.-м.н., ст. викладач Фесенко А.В.,

затверджені наказом по університету №__ від «__» _____ 2022 р.

2. Термін подання студентом роботи: «__» _____ 2022 р.

3. Вихідні дані до роботи: *обчислення оцінок часової складності, складності за пам'яттю та кількості необхідних пар відкритих текстів та шифротекстів для проведення успішної атаки відновлення раундової функції.*

4. Зміст роботи: *огляд атаки повного перебору та атаки Meet-In-The-Middle на шифри на основі схеми Фейстеля, які дозволяють відновити раундові функції; побудова аналогічних атак відновлення раундової функції на шифри на основі узагальнених схем Фейстеля; обчислення часової складності, складності за пам'яттю та кількості необхідних пар відкритих текстів та шифротекстів для проведення успішної атаки відновлення раундової функції.*

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): *Презентація доповіді*

6. Дата видачі завдання: 10 вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2021 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2021 р.	Виконано
3	Дослідження шифрів зі збереженням формату	Листопад-грудень 2021 р.	Виконано
4	Дослідження атак відновлення раундової функції	січень-лютий 2022 р.	Виконано
5	Отримання власних результатів та їх доведення	березень-травень 2022 р.	Виконано

Студент _____ Мельниченко О.С.

Керівник _____ Фесенко А.В.

РЕФЕРАТ

Кваліфікаційна робота містить: 38 сторінок, 10 рисунків, 1 таблицю, 16 джерел.

Метою роботи є огляд наявних стандартів шифрування зі збереженням формату, огляд атак відновлення раундової функції та знаходження оцінок складності для цих атак.

У ході написання роботи було проведено огляд наявних стандартів шифрування зі збереженням формату, побудованих на основі схеми Фейстеля, та атак відновлення раундових функцій. Було побудовано Meet-In-The-Middle атаки відновлення раундових функцій на декілька Фейстель-подібних шифрів зі збереженням формату. Також було обчисленно оцінки часової складності та складності за пам'яттю для атаки на кожен шифр.

СХЕМА ФЕЙСТЕЛЯ, АТАКИ ВІДНОВЛЕННЯ РАУНДОВОЇ ФУНКЦІЇ, АТАКА MEET-IN-THE-MIDDLE

ABSTRACT

Qualification work contains: 38 pages, 10 figures, 1 table and 16 references.

The aim of this paper is to review the existing format-preserving encryption standards, round function recovery attacks, and calculating the complexity estimates for those attacks.

In the course of writing the qualification work, a review of published sources on the Feistel-based format-preserving encryption standards and the round function recovery attacks was conducted. The Meet-In-The-Middle round function recovery attack was implemented for several Feistel-based format-preserving ciphers. The time and memory complexity of the attack was calculated for each cipher.

FEISTEL NETWORK, ROUND FUNCTION RECOVERY ATTACK, MEET-IN-THE-MIDDLE ATTACK

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

FPE — шифрування зі збереження формату

$\oplus$ — операція побітового додавання

ВСТУП

Актуальність дослідження. Схема Фейстеля дуже часто використовується для шифрів зі збереженням формату з малою областю визначення, зокрема шифри які входять у стандарти FF1 та FF3-1 побудовані саме з використанням схеми Фейстеля.

Найбільш вдалими атаками на шифри зі збереженням формату є атаки відновлення раундової функції. У роботі розглянуті атака повного перебору раундових функцій та атака Meet-In-The-Middle. Також запропонована побудова атаки Meet-In-The-Middle на узагальнені схеми Фейстеля, а саме: незбалансовану, альтернаційну, схеми типу I, типу II та типу III. Для побудованих атак обчислені часові складності, складності за пам'яттю та кількості необхідних пар відкритих текстів та шифротекстів для проведення успішної атаки відновлення раундової функції.

Об'єктом дослідження є стандарти шифрування зі збереженням формату даних та побудова атак відновлення раундової функції на них.
Предметом дослідження є стійкість шифрів та стандартів шифрування зі збереженням формату до атак відновлення раундової функції.
Метою дослідження є огляд наявних стандартів шифрування зі збереженням формату, огляд атак відновлення раундової функції та знаходження оцінок складності для цих атак.

У процесі дослідження було поставлено такі задачі: 1) Розглянути шифри зі збереженням формату з малою областю визначення;

2) Розглянути атаки відновлення раундової функції, а саме атаку Meet-In-The-Middle;

3) Обчислити оцінки часової складності для атаки Meet-In-The-Middle побудованої на узагальнених схемах Фейстеля;

4) Обчислити оцінки складності за пам'яттю для атаки Meet-In-The-Middle побудованої на узагальнених схемах Фейстеля;

5) Обчислити кількість необхідних пар відкритих текстів та шифротекстів для проведення успішної атаки відновлення раундової функції для узагальнених схем Фейстеля.

Наукова новизна отриманих результатів полягає в тому, що вперше побудовано та обчислено складності атаки Meet-In-The-Middle побудованої до узагальнених схем Фейстеля, а саме: незбалансовану, альтернаційну, типу I, типу II та типу III.

Практичне значення результатів полягає у використанні отриманих оцінок складності атаки відновлення раундової функції для подальшого дослідження стійкості шифрів зі збереженням формату та покращення їх алгоритмів та структури.

Апробація результатів та публікації. Результати дослідження, описані в даній роботі, частково були висвітлені в доповіді на XX Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих учених «Теоретичні та прикладні проблеми фізики, математики й інформатики» (місто Київ, 15-17 червня 2022 року).

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

У цьому розділі буде розглянуто базові поняття потрібні для шифрування зі збереженням формату (FPE), історію створення FPE, проблеми та способи їх вирішення, а також атаки які застосовуються до цих видів шифрування.

1.1 Загальні відомості з захисту інформації

З початку розвитку людства проблеми конфіденційної передачі прихованої інформації не давали спокою вченим. Раніше не було потреб шифрувати невеликі повідомлення так часто, але з плином часу така необхідність з'явилася.

На даному етапі введемо поняття алгебраїчної системи запропоноване у середині XX століття американським криптографом Клодом Шенноном. Це визначення поклало початок сучасної криптографії.

Означення 1.1. Алгебраїчною моделю шифру ($\mathbb{A}$) *(криптосистемою з секретним ключем або симетричною криптосистемою)* називають універсальну алгебру $\mathbb{A} \equiv < \mathbb{M}, \mathbb{K}, \mathbb{C}, \mathbb{E}, \mathbb{D} >$, де $\mathbb{M}$(messages) — множина повідомлень (множина відкритих текстів),
$\mathbb{K}$(keys) — множина ключів,
$\mathbb{C}$(cipher text) — множина шифротекстів,
$\mathbb{E}$(encryption) — множина функцій зашифрування,
$\mathbb{D}$(decryption) — множина функцій розшифрування.

Означення 1.2. Функція зашифрування — це таке шифруюче перетворення $E_k \in \mathbb{E} : E_k : \mathbb{M} \times \mathbb{K} \to \mathbb{C}$.

Означення 1.3. Функція розшифрування — це таке розшифруюче перетворення $D_k \in \mathbb{D} : D_k : \mathbb{C} \times \mathbb{K} \to \mathbb{M}$.

Однак час минав, відбулася індустріалізація, потім діджиталізація і

цифровізація. На сьогоднішній день майже кожен користується банками, медичними установами, страховими компаніями тощо. В усіх цих місцях кожному клієнту належать його унікальні ідентифікатори, що підтверджують справжність особи. Для всіх цих ідентифікаторів треба забезпечити приватність, їх цілісність і секретність. Через це стали конче необхідні алгоритми шифрування і розшифрування даних.

Означення 1.4. Блочний шифр — детермінований примітив, сімейство перестановок, які індексовані секретним ключем.

$$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n,$$

де P(plain text) — відкритий текст,
C(cipher text) — зашифрований текст,
E(encryption) — зашифрування,
K(key) — ключ.

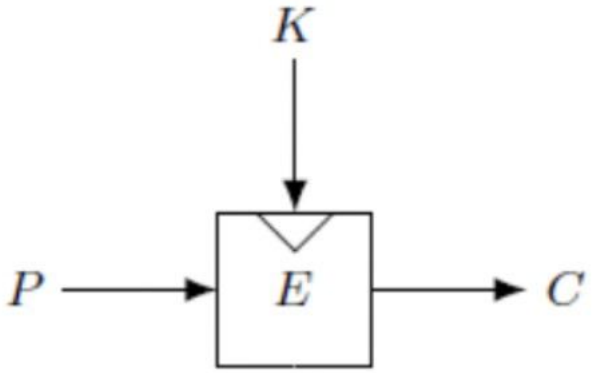

Рисунок 1.1 – Схема звичайного шифрування.

Для блочних шифрів не використовують ключі маленької довжини, бо це є загрозою легкого зламу методом перебору всіх ключів. Також для побудови блочного шифру необхідні: режими роботи, схеми заповнення та схема шифрування.

Означення 1.5. Режим роботи блочного шифру (або просто режим) — це алгоритм криптографії для перетворення даних на основі блочного шифру.

Раніше затверджені режими для шифрування полягали в перетворені двійкових даних, тобто входи та виходи режимів є бітовими рядками (послідовності нулів та одиниць).

Для послідовностей небінарних символів, немає загального способу, для раніше затверджених режимів, створення зашифрованих даних, які мають той самий формат.

Наприклад, номер банківської карти складається з 16-ти десяткових цифр — ціле число, яке менше 10^{16}. Це ціле число можна перетворити на бітовий рядок, як вхідні дані для раніше затвердженого режиму, але коли вихідний (бітовий) рядок перетворюється назад у ціле число, він може бути більше за 10^{16}, що було б занадто довго для номеру банківської карти. Виходячи з цього з'явилася потреба зробити алгоритми шифрування зі збереженням формату (FPE).

1.2 Шифрування зі збереженням формату

Означення 1.6. Шифрування зі збереженням формату (англ. format-preserving encryption (FPE)) — таке криптографічне шифрування, у якого вихідний шифротекст має той самий формат, що і вхідний відкритий текст.

У цьому визначенні ключове слово «формат». В залежності від складу відкритого тексту формат може бути різним. Найчастіше використовують:

– скінчені числові набори (шифрування 16-ти цифрового номеру банківської картки в інший 16-ти цифровий набір)

– скінчені алфавітні набори (слово з символів певного алфавіту в інше слово з символів цього алфавіту)

– скінчені буквенно-числові набори (певна комбінація з числових і алфавітних наборів символів в іншу комбінацію такого формату).

FPE розроблено для даних які не обов'язково є двійковими, воно працює з кінцевими послідовностями символів, при тому вхідний і вихідний текст має однакову довжину і сам набір символів (той самий формат).

Звичайні блочні шифри не можуть забезпечити збереження формату тому, що:

– у блочному шифруванні втрачається формат тексту,

– іде розширення тексту і текст змінює свій розмір,

– як казали Брайтвелл і Сміт у своїй роботі 1997 року: "Шифрований текст... приблизно так само схожий на відкритий текст... як гамбургер до стейка на кістці".

Ідентифікатори зазвичай мають однаковий формат: номер картки це 16 цифр від 0 до 9, номер страхування SSN складається з 9 десяткових цифр, номер ID-картки — 9 десяткових цифр.

Також, дуже важливим є коректне шифрування бази даних. Бажано, щоб ці дані при передачі зберігали свій формат, але були зашифровані. Звичайна алгебраїчна модель шифру не підходить для відкритого тексту малої довжини. Для шифрування такого тексту треба ввести поняття алгебраїчна модель шифру з налаштуванням.

Означення 1.7. Шифрування з налаштуванням (англ. tweakable encryption) — алгебраїчна модель шифру, де окремим параметром задається налаштування (англ. tweak). $\mathbb{A} \equiv < \mathbb{M}, \mathbb{K}, \mathbb{T}, \mathbb{C}, \mathbb{E}, \mathbb{D} >$,

де $\mathbb{M}$(messages) - множина повідомлень (множина відкритих текстів),

$\mathbb{K}$(keys) - множина ключів,

$\mathbb{T}$(tweaks) - множина налаштувань,

$\mathbb{C}$(cipher text) - множина шифротекстів,

$\mathbb{E}$(encryption) - множина функцій зашифрування,

$\mathbb{D}$(decryption) - множина функцій розшифрування.

$$E : \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^n$$

Означення 1.8. Функція зашифрування з налаштуванням — це таке шифруюче перетворення $E_k^t \in \mathbb{E} : E_k^t : \mathbb{M} \times \mathbb{K} \times \mathbb{T} \to \mathbb{C}$.

Означення 1.9. Функція розшифрування з налаштуванням — це таке розшифруюче перетворення $D_k^t \in \mathbb{D} : D_k^t : \mathbb{C} \times \mathbb{K} \times \mathbb{T} \to \mathbb{M}$.

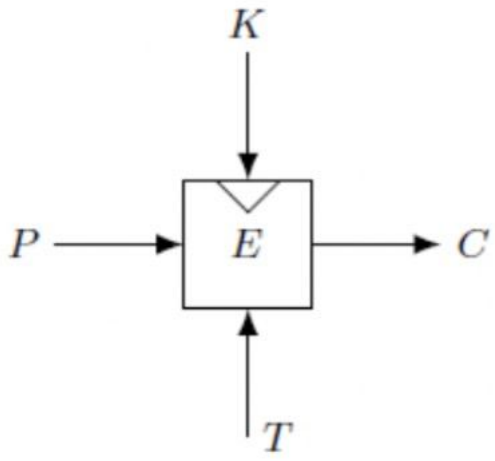

Рисунок 1.2 – Схема шифрування з налаштуванням.

Це шифрування було формалізоване Лісковим, Рівестом, Вагнером у їх роботі 2002 року [1]. Тепер можна показати на прикладі чим шифрування з налаштуванням краще звичайного шифування для даних невеликої довжини. Спочатку буде приклад зі звичайним шифруванням, а потім шифруванням з налаштуванням.

Приклад 1.1. Розглянемо номер банківської карти (англ. credit card number (CCN)). Це 16 десяткових цифр.
Візьмемо для прикладу: 4385 8220 5611 0982.
Перші 6 цифр вказують банківський ідентифікатор.
В нашому випадку 4385 82.
Останні 4 цифри це перевірочні символи, їх показують абоненту кожен раз перед підтвердженням оплати.

В нашому випадку 0982.

І тільки цифри характерні абоненту - це цифри з сьомої по дванадцяту, вони вказують на номер рахунку.

В нашому випадку 6 цифр 20 5611.

Отже, коли ми приступимо до шифрування ми будемо шифрувати тільки ці шість цифр посередині.

Припустимо, що ми дізналися номер банківської картки абонента з іншого банку, і номер рахунку абонента з другого банку збігається з першим.

Наприклад картка з номером: 4485 3920 5611 2836.

Бачимо, що 6 цифр посередині у двох карток однакові:

438582 205611 0982

448539 205611 2836

Тепер спробуємо зашифрувати номер рахунку абонента:

438582 $E_k(205611)$ 0982

448539 $E_k(205611)$ 2836

В цем момент і виникає колізія, бо в результаті отримаємо два однакових шифротекста.

438582 849682 0982

448539 849682 2836

Проблема в тому, що якщо ми отримаємо номер іншої картки, зашифруємо тільки 6 цифр і вони однакові, то ми отримаємо два шифротексти з однаковими шифротекстами посередині. Це нам дозволить використати атаку перебору за словником. Але знаючи тільки номер кредитної карти і вихідний шифротекст, можемо зберігати їх в таблицю, і кожен раз коли 6 цифр збігаються ми можемо розшифрувати новий номер кредитної карти, шифрування котрої ми не знали. Щоб ми не могли провести атаку перебору за словником, треба щоб ті хто шифрують номери використовували різні ключі.

Означення 1.10. Перебір за словником (англ. dictionary attack) — атака на систему захисту, що використовує метод повного послідовного перебору всіх символів використовуваних для аутентифікації.

Ми бачимо, що не можемо просто так вибрати ключ, який би був пов'язаний з повідомленням. Нам треба якось змінити функцію шифрування, зробити в ній деякі налаштування. Спробуємо зашифрувати через шифрування з налаштуванням. Саме налаштування ми будемо брати використавши відкриту відому інформацію, наприклад номер банківського ідентифікатора або контрольні цифри.
Мета цього не використовувати один випадковий закритий ключ, а замість нього використати дві різні перестановки. Таким чином налаштування нам дають сімейство псевдовипадкових перестановок.

Означення 1.11. Псевдовипадкова перестановка (англ. pseudorandom permutation (PRP)) — це функція, яку неможливо за допомогою розумних зусиль відрізнити від випадкової перестановки (тобто від перестановки обраної випадково й однорідно з сім'ї всіх перестановок на домені функції).

Вибираючи один ключ ви отримуєте, замість однієї випадкової перестановки, ви отримуєте дуже багато перестановок в залежності від налаштувань, які ви використовуєте. Однак налаштування дуже відрізняються від ключів. Налаштування не потрібно приховувати, ми використовуємо загальновідому інформацію, це додає варіативності.

Приклад 1.2. Розглянемо приклад з номерами карток з прикладу 1, тільки тепер будемо використовувати шифрування з налаштуваннями.
Номери до шифрування:
438582 205611 0982
448539 205611 2836
Використаємо шифрування з налаштуванням:
438582 $E_k(205611)^{4385820982}$ 0982
448539 $E_k(205611)^{4485392836}$ 2836
В результаті вийшло:
438582 237849 0982
448539 967395 2836

1.3 Стандартизація шифрів зі збереженням формату

Коли йде мова про нове шифрування, то полягає необхідність в нових стандартах. Стандартизація FPE описана у проекті NIST SP 800-38G (оновлена у лютому 2019р.)[2].

Головні вимоги визначенні для стандартів:

– повинен підримуватися будь-який формат, який визначає користувач;

– не дозволяється розширення зашифрованого тексту.

Дуже важливо розуміти у що ми хочемо зашифрувати наш текст. Який формат буде мати вихідний текст. Наприклад, ми хочемо зашифрувати цифру 2. Треба розуміти який її формат, це число від 1 до 3, чи від 1 до 5, чи від 1 до 9. Просто побачивши повідомлення ми не знаємо його формат. Отже, якщо ми зацікавлені в зашифруванні, ми не зрощуємо по повідомленню що це за формат, нам треба обов'язково вказати формат по відношенню до якого ми будемо шифрувати. Визначаються такі поняття безпеки в FPE:

– псевдовипадкова перестановка (англ. pseudo-random permutation (PRP))

– одноточкова нерозрізненість (англ. single point indistinguishability (SPI))

– приватність повідомлень (англ. message privacy (MP))

– відновлення повідомлень (англ. message recovery (MR))

$PRP \Longleftrightarrow SPI \Longrightarrow MP \Longrightarrow MR$

1.4 FPE конструкції

Вперше реалізація FPE була використана в роботі Джона Блека та Філіпа Рогавея CT-RSA 2002. Також було доведено, що кожен з методів має таку ж надійність, як і сам блочний шифр, який використовується для їх побудови. Це означало, що зловмисник, який здатний розшифрувати FPE, здатний розшифрувати і алгоритм блочного шифру.

Загальні схеми FPE від Блека і Рогавея:

– префіксний шифр (prefix cipher)

– циклове ходіння (cycle-walking)

– узагальнений шифр Фейстеля(generalized-Feistel cipher)

Інші спеціальні конструкції:

– FFSEM (Спайс) пізніше замінений на FFX [3];

– FFX (FF1, FF2). (Беллар , Рогавей, Спайс, поданий до NIST в лютому 2010) [4];

– BPS (renamed FF3 by NIST). (Браєр, Пайрін, Стерн, поданий до NIST в березні 2010) [5];

– FEA-1 and FEA-2 (Лі, Ку, Ро, Кім, Квон, корейські FPE стандарти)[6].

1.5 Узагальнена схема Фейстеля

В 1973 році Горст Файстель в журналі Scientific American опублікував статтю «Криптографія і комп'ютерна безпека» («Cryptography and Computer Privacy»), в якій розкрив деякі важливі аспекти шифрування, а також ввів схему ітеративного блочного шифру, названу пізніше мережею Фейстеля.

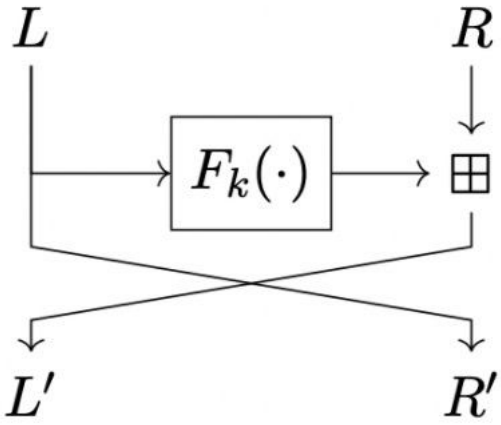

Рисунок 1.3 – Схема шифрування на основі мережі Фейстеля.

Висновки до розділу 1

У даному розділі були розглянуті матеріали за темою дослідження. Шифрування зі збереженням формату полягає в алгоритмах і схемах, які забезпечуюють цілісність, приватність, конфіденційність, певну складність до зламу.

У розділі подані найсучасніші стандарти, які затверджені і активно використовуються у світовій практиці при шифруванні даних певної фіксованої довжини та формату.

2 ОГЛЯД СТАНДАРТІВ ШИФРУВАННЯ ЗІ ЗБЕРЕЖЕННЯМ ФОРМАТУ ТА ПОБУДОВА АТАК НА НИХ

У даному розділі будуть розглянуті найвідоміші стандарти шифрування зі збереженням формату побудовані саме з використанням схеми Фейстеля, а саме стандарт FF1 та FF3-1. Також побудовані атака повного перебору та атака Meet-In-The-Middle на узагальнені схеми Фейстеля, а саме: незбалансовану, альтернаційну, схеми типу I, типу II та типу III. Для побудованих атак обчислені часові складності, складності за пам'яттю та кількості необхідних пар відкритих текстів та шифротекстів для проведення успішної атаки відновлення раундової функції.

2.1 Стандарт FF1

Стандарт FF1 є Фейстеле-подібною структурою, і для шифрування використовує раундову функцію на основі алгоритму AES-128. Також для шифрування знадобиться відкритий текст, 128-бітний ключ AES та певне налаштування.

Спочатку дані розбиваються на дві частини, потім одна частина обчислюється за допомогою n (довжини всього повідомлення), T (нашого налаштування) та r номеру раунду у раундовій функції (F_K); і змінюється з іншою місцями. Цей процес повторюється для r раундів, де для FF1 це 10 раундів.

FF1 — це шифрування, що зберігає формат, заснований на FFX режимі. Режим FFX обмежує довжину повідомлення до 128 біт, однак FF1 може мати мінімальну довжину в 2 біти,а максимальну в 2^{32} біти і довжина налаштування 2^{32} біта.

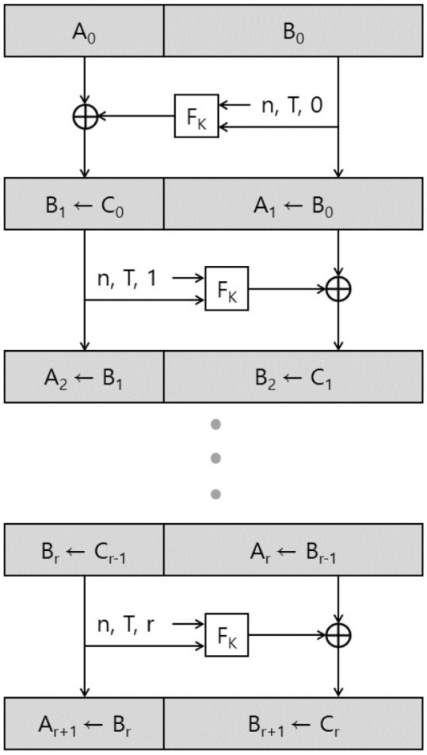

Рисунок 2.1 – Схема шифрування FF1 на основі мережі Фейстеля.

2.2 Стандарт FF3 та його оновлення FF3-1

Браєр, Пейрін і Штерн спеціально для NIST розробили алгоритм шифрування даних зі збереженням формату, який був названий в честь них - BPS. Однак, потім NIST[7] прийняв їх схему, як стандарт шифрування і перейменував на FF3.

BPS був розроблений для використання стандартизованих алгоритмів у раундовій функції, такі як TDEA, AES і SHA-2, але був модифікований для використання AES [8].

Дизайном шифрування FF3 є схема Файстеля, де в раундову функцію подається номер раунду та певне налаштування. Раундовою функцією виступає алгоритм AES-128.

Спочатку вхідні дані розбиваються на дві частини. Потім одна частина обчислюється за допомогою n (довжини всього повідомлення), T (налаштування) і r (номеру раунду) - все це використовується в раундовій функції (F_K). Потім ця частина змінюється з іншою місцями.

Цей процес повторюється 8 раундів. [2]

а) б)

Рисунок 2.2 – а) мережа Файстеля; б) схема шифрування FF3

У своїй роботі Дюрак і Водено [9] показали, що FF3 є недостатньо безпечним і не може гарантувати стійскість при 128-бітному просторі вхідних даних при $7 \leqslant N \leqslant 10$, де N - кількість всіх можливих входів раундової функції мережі Файстеля всередині алгоритму. Також у цій роботі було запропоновано зміни до стандарту FF3, які повинні були його покращити.

У квітні 2017 року NIST[10] прийняв до уваги їх роботу та оголосив про свій намір або переглянути стандарт FF3, зменшивши розмір налаштування з 64 бітів до 48 (як радили автори у статті [11]), або зовсім відмовитись від цього стандарту.

Натомість був обраний варіант зменшити розмір налаштування до 56 бітів, це було розроблено спільно з авторами статті. Пізніше цю зміну стандарту опубліковано у новій ревізії стандартів [2], а новий покращений стандарт отримав назву FF3-1.

Введемо деякі позначення для функцій:

STR_s^b - функція, яка представляє ціле число x, де $0 \leqslant x < s^b$, у вигляді рядка довжини b за основою s із старшим символом(бітом)

спочатку. Наприклад, $STR_{12}^{4}(554) = 03A2$.

NUM_s - функція, яка перетворює рядок X у ціле число x так, що $STR_s^b(x) = X$. Наприклад, $NUM_s(00011010) = 26$.

$REV(X)$ - функція, яка змінює порядок символів рядка X на зворотний.

Розглянемо алгоритм шифрування FF3-1[12]. Схема алгоритму запозичена зі статті Янга та Лі [13] .

Алгоритм шифрування FF3-1:

1) Нехай $u = \lceil n/2 \rceil$; $v = n - u$.

2) Нехай $A = X[1..u]$; $B = X[u + 1..n]$.

3) Нехай $T_L = T[0..27] \parallel 0^4$ та $T_R = T[32..55] \parallel T[28..31] \parallel 0^4$.

4) Для i від 0 до 7:

а) Якщо i парне, то нехай $m = u$ та $W = T_R$, інакше нехай $m = v$ та $W = T_L$.

б) $P = W \oplus [i]^4 \parallel [NUM_{radix}(REV(B))]^{12}$.

в) $S = REVB(CIPH_{REVB(K)}REVB(P))$.

г) $y = NUM(S)$.

д) $c = (NUM_{radix}(REV(A)) + y) \mod radix^m$.

е) $C = REV(STR_radix^m(c))$.

ж) $A = B$.

з) $B = C$.

5) Подаємо на вихід $A \parallel B$.

Розбиваємо відкритий текст X на дві частини A і B. Довжина налаштування T в алгоритмі є фіксованою і дорівнює 54 бітам. Далі налаштування T розбивається на дві частини T_L та T_R, як показано на третьому кроці алгоритму.

Потім починається цикл в якому i виступає позначенням номеру раунду. Якщо i парне, то $m = u$, $W = T_R$, якщо i непарне, то $m = v$, $W = T_L$ (u та v були обраховані на першому кроці алгоритму).

4-байтове значення створюється операцією $\oplus$ (XOR) між W та i

(доповненим на початку нулями до розміру 4 байти). 12-байтове значення створюється наступним чином: число B, записане в системі числення з основою $radix$, записується у зворотньому порядку та переводиться у десяткову систему, далі доповнюється нулями на початку до розміру 12 байт. Далі 4-байтове на 12-байтове значення конкатенуються.

Для обчислення S, порядок байтів у числі P змінюється на зворотній, далі отримане число зашифровується алгоритмом AES-128 з ключем отриманим операцією реверсу байтів ключа K. Байти отриманого числа знову записуються в зворотньому порядку. Далі за допомогою функцій NUM_{radix}, STR_{radix} та REV отримуємо значення y, c та C. B стає на місце A, C стає на місце B.

Цикл повторюється 8 раундів і потім, в результаті конкатенації кінцевих A та B, отримуємо шифротекст.

Алгоритм розшифрування є оберненим до алгоритму зашифрування.

2.3 Атаки відновлення раундової функції

Для шифрів з малою областю визначення є можливим підібрати не самі ключі до раундових функцій, а саму раундову функцію. Через те що область визначення раундової функції є невеликого розміру може скластися так, що перебирати всі раундові функції буде менше, ніж перебирати всі ключі до них. Крім того, в раундові функції алгоритмів шифрування зі збереженням формату, зокрема шифри FF1 та FF3-1, подається не тільки ключ, а ще й деяке налаштування, що ще більше ускладнює перебір всіх можливих ключів.

2.4 Атака повного перебору

Розглянемо найпростішу атаку на основі пар відкритого тексту та шифротексту методом повного перебору. Для проведення цієї атаки

необхідно перебрати всі можливі раундові функції на всіх раундах для всіх відкритих текстів.

Зламом схеми буде знаходження відповідностей між зашифрованими відкритими текстами і початковими шифротекстами (буває декілька варіантів тому з великою ймовірністю варіант їх збігу можливий тільки за умови підбору правильних раундових функцій).

Нехай схема складається з r раундів, $2n$ є довжиною відкритого тексту та шифротексту, $N = 2^n$ — кількість можливих входів у раундову функцію, k — довжина ключа. Оцінки складності за часом і пам'яттю для атаки на основі повного перебору такі:

$$S = 2n2^{2n}(2^k)^r,$$

$$T = (N^N)^r.$$

Однак повний перебір явно не може дати бажаних результатів у ефективності роботи, тому у 1977 році Діффі та Геллман [14] запропонували нову атаку під назвою "Meet-In-The-Middle" ("зустріч посередині").

2.5 Атака Meet-In-The-Middle

Це атака на основі відомих пар відкритих текстів і шифротекстів. Алгоритмом атаки Meet-In-The-Middle є не перебір всіх раундових функцій для всіх r раундів, а навпаки, вибір раунду приблизно посередині (назвемо його u-й раунд, де найчастіше $u = \lfloor \frac{r}{2} \rfloor$) і часткове зашифровування відкритих текстів з першого до u-го раунду, а також часткове розшифрування шифротекстів з r-го раунду теж до u-го раунду. Таким чином, на цьому раунді u будемо шукати збіги серед частково зашифрованих відкритих текстів і частково розшифрованих шифротекстів.

2.6 Побудова атаки Meet-In-The-Middle

Для перебору всіх раундових функцій з довжиною ключа k для одного раунду необхідно перебрати 2^k варіантів ключів, таким чином для u раундів треба 2^{uk} . Так як кожен відкритий текст та шифротекст має довжину $2n$, то для q пар відкритих текстів та шифротекстів знадобиться $2nq$ варіантів для опису всіх підібраних шифротекстів. Таким чином, отримуємо складність за пам'яттю:

$$S = 2nq(2^k)^u$$

Використовуємо впорядкованість вхідних даних функцій, щоб не брати до уваги їх перестановки. Складність за часом розглянемо як перебір всіх варіантів раундових функцій 2^k для $r-u$ раундів. Таким чином, складність по часу буде:

$$T = (2^k)^{r-u},$$

$k = n2^n = Nlog_2N$, де $N = 2^n$ — кількість способів описати раундову функцію, бо описуємо всі варіанти входів і виходів раундової функції. Відзначимо, що $r - u = \lceil \frac{r}{2} \rceil$

З цього випливає оцінка складності по часу:

$$T = N^{N\lceil \frac{r}{2} \rceil}$$

Для відновлення однієї раундової функції необхідно $n2^n$ бітів, отже, для r-раундової схеми Фейстеля вже необхідно буде $rn2^n$ бітів.

З цього випливає, що необхідний мінімум наявних пар відомих відкритих текстів та шифротекстів, повинен бути $q = \frac{rN}{2}$, що доведено в роботі Патаріна [15].

2.7 Обчислення оцінок складності атак відновлення раундових функцій для узагальнених схем Фейстеля

Розглянемо узагальнені схеми Фейстеля, а саме: незбалансовану, альтернаційну, схему типу I, типу II та типу III. Усі вони описані у роботі Гоанга та Рогавея [16].

Незбалансована схема Фейстеля

Спочатку розглянемо незбалансовану схему Фейстеля зображену на рис. 2.3.

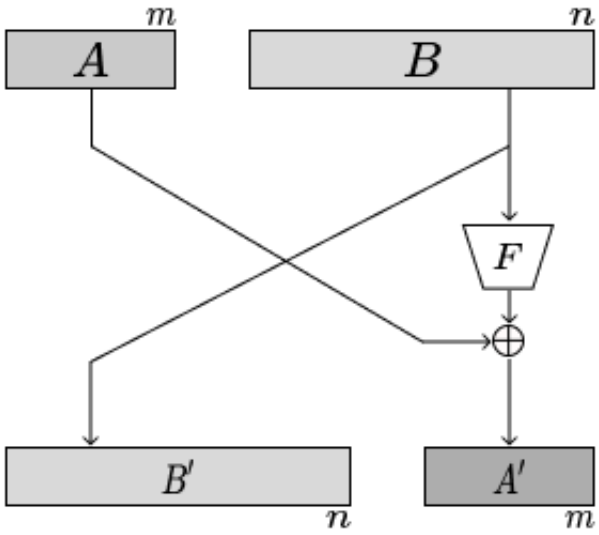

Рисунок 2.3 – Незбалансована схема Фейстеля

У незбалансованій схемі Фейстеля раундова функція є стискаючою, бо довжина входу — n бітів, а довжина виходу — m бітів ($n > m$).

Будемо позначати буквами N та M, де $N = 2^n$ та $M = 2^m$, кількість різних варіантів вхідних даних раундової функції (n або m позначають довжину входу у бітах).

Таким чином, можна обчислити кількість бітів для відновлення раундової функції, яка дорівнює $k = 2^n m = N log_2 M$, а довжина вхідного відкритого тексту дорівнює $(n + m)$ бітів.

З цього випливає, що складність за пам'яттю і часова складність,

а також необхідна кількість пар відкритих текстів та шифротекстів для відновлення раундових функцій обчислюється, як при проведенні атаки Meet-In-The-Middle:

$$S = (n+m)q(2^k)^u,$$

$$T = M^{N\lceil\frac{r}{2}\rceil},$$

$$q = \frac{rm2^n}{n+m}.$$

Альтернаційна схема Фейстеля

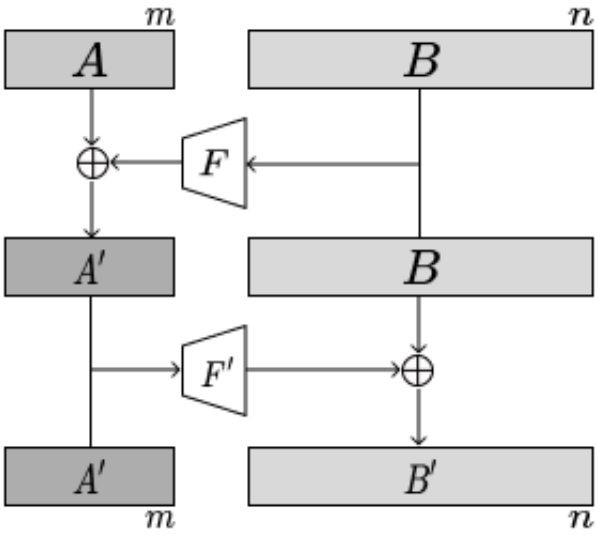

Рисунок 2.4 – Альтернаційна схема Фейстеля

Альтернаційна схема Фейстеля, зображена на рис. 2.4, має два види раундових функцій: стискаючу, бо довжина входу — n бітів, а довжина виходу — m бітів, та розширюючу, бо довжина входу — m бітів, а довжина виходу — n бітів.

Таким чином, загальну кількість різних варіантів двох видів раундових функцій позначимо, як k_1 та k_2, де $k_1 = 2^n m = N log_2 M$ та $k_2 = 2^m n = M log_2 N$.

З цього випливає, що складність за пам'яттю і часова складність, а також необхідна кількість пар відкритих текстів та шифротекстів для

відновлення раундових функцій при проведенні атаки Meet-In-The-Middle обчислюється, як:

$$S = (n+m)q(2^k)^u,$$

$$T = M^{N\lceil\frac{r}{4}\rceil} \cdot N^{M\lceil\frac{r}{4}\rceil},$$

$$q = \frac{r(mN+nM)}{2(n+m)}.$$

Схема Фейстеля тип I

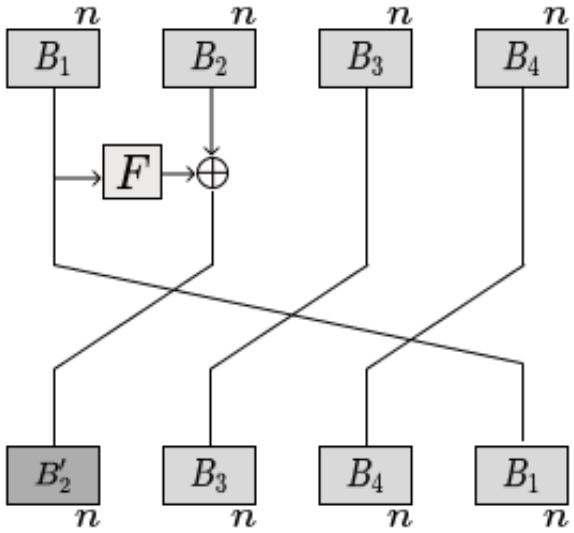

Рисунок 2.5 – Схема Фейстеля тип I

Тепер розглянемо узагальнену схему Фейстеля типу I, зображену на рис. 2.5.

Її особливістю є те, що вона розбиває вхідний текст на 4 частини, кожна з яких має довжину n.

У схемі типу I є одна звичайна раундова функція. Функція приймає вхідні дані довжиною n бітів та в результаті повертає рядок довжиною n бітів.

З цього випливає, що $k = n2^n = Nlog_2N$.

У результаті проведення атаки Meet-In-The-Middle на узагальнену схему Фейстеля типу I, отримано такі результати складності за пам'яттю,

часової складності та необхідної кількості пар відкритих текстів та шифротекстів для відновлення раундових функцій:

$$S = 4nq(2^k)^u,$$

$$T = N^{N\lceil\frac{r}{2}\rceil},$$

$$q = r2^{n-2}.$$

Схема Фейстеля тип II

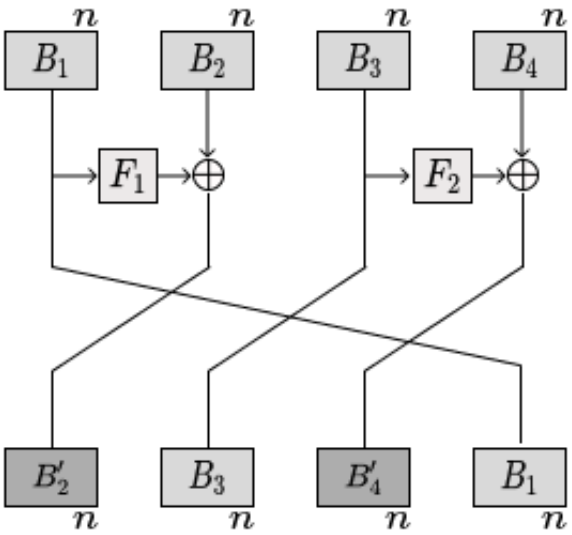

Рисунок 2.6 – Схема Фейстеля тип II

Розглянемо узагальнену схему Фейстеля типу II, зображену на рис. 2.6.

Її особливістю є те, що вона розбиває вхідний текст на 4 частини, кожна з яких має довжину n.

У схемі типу II є дві звичайні раундові функції. Функції приймають вхідні дані довжиною n бітів та в результаті кожна повертає рядок довжиною n бітів.

З цього випливає, що $k = n2^n = Nlog_2N$.

У результаті проведення атаки Meet-In-The-Middle на узагальнену схему Фейстеля типу II, отримано такі результати складності за

пам'яттю, часової складності та необхідної кількості пар відкритих текстів та шифротекстів для відновлення раундових функцій:

$$S = 4nq(2^k)^u,$$

$$T = N^{2N\lceil\frac{r}{2}\rceil},$$

$$q = r2^{n-1}.$$

Схема Фейстеля тип III

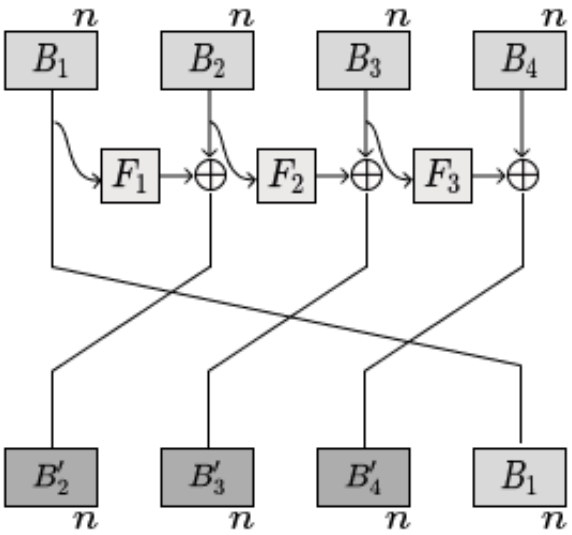

Рисунок 2.7 – Схема Фейстеля тип III

Останньою розглянемо узагальнену схему Фейстеля типу III, зображену на рис. 2.7.

Її особливістю є те, що вона розбиває вхідний текст на 4 частини, кожна з яких має довжину n.

У схемі типу III є три звичайні раундові функції. Функції приймають вхідні дані довжиною n бітів та в результаті кожна з них повертає рядок довжиною n бітів.

З цього випливає, що $k = n2^n = Nlog_2N$.

У результаті проведення атаки Meet-In-The-Middle на узагальнену схему Фейстеля типу III, отримано такі результати складності за

пам'яттю, часової складності та необхідної кількості пар відкритих текстів та шифротекстів для відновлення раундових функцій:

$$S = 4nq(2^k)^u,$$

$$T = N^{3N\lceil\frac{r}{2}\rceil},$$

$$q = 3r2^{n-2}.$$

2.8 Експериментальні результати

Побудуємо таблиці експериментальних результатів для атаки повного перебору та атаки Meet-In-The-Middle застосованих до шифрів зі збереженням формату, а саме з параметрами стандартів FF1 та FF3-1.

Обчислимо чисельні оцінки складності для вхідного тексту довжиною 12 бітів. Будемо припускати, що маємо 10 раундів схеми Фейстеля, як в алгоритмі шифрування FF1. Для незбалансованих схем припустимо, що поділ буде йти на дві частини по 4 бітів та 8 бітів.

Таблиця 2.1 – чисельні оцінки складності атак для вхідного тексту довжиною 12 бітів

Схема Фейстеля	Складність за пам'яттю	Часова складність	к-ть пар ВТ та ШТ
Стандартна для повного перебору	2^{74}	2^{2560}	-
Стандартна для MITM	2^{41}	2^{1920}	320
Незбалансована для MITM	2^{56}	2^{2520}	1024
Альтернаційна для MITM	2^{53}	2^{3456}	512
Тип I для MITM	2^{19}	2^{360}	20
Тип II для MITM	2^{19}	2^{720}	40
Тип III для MITM	2^{19}	2^{1080}	60

Висновки до розділу 2

У даному розділі було досліджено алгоритми шифрування FF1 та FF3-1. Побудовано атаку повного перебору та атаку Meet-In-The-Middle на узагальнені схеми Фейстеля, а саме: незбалансовану, альтернаційну, типу I, типу II та типу III. Було обчислено оцінки складності для цих атак побудованих на узагальнені схеми Фейстеля.

В результаті, атака Meet-In-The-Middle є значно ефективнішою за атаку повного перебору. Серед узагальнених схем Фейстеля найбільш вразливою до атаки Meet-In-The-Middle виявилась схема Фейстеля типу I.

ВИСНОВКИ

У оглядовій частині даної роботи було проведено огляд опублікованих джерел за тематикою шифоування зі збереженням формату. Розглянуто необхідні теоретичні відомості для загального розуміння цих шифрів. Наведено вимоги до шифрів та розглянуто шифри, які були включені у стандарти NIST. Більшість з них побудовано на основі схем Фейстеля.

У другому розділі було досліджено алгоритми шифрування FF1 та FF3-1. Побудовано атаку повного перебору та атаку Meet-In-The-Middle на узагальнені схеми Фейстеля, а саме: незбалансовану, альтернаційну, типу I, типу II та типу III. Було обчислено оцінки складності для цих атак побудованих на узагальнені схеми Фейстеля.

В результаті, атака Meet-In-The-Middle є значно ефективнішою за атаку повного перебору. Серед узагальнених схем Фейстеля найбільш вразливою до атаки Meet-In-The-Middle виявилась схема Фейстеля типу I.

Подальші дослідження можуть стосуватися модифікацій стандартів шифрування за допомогою інших узагальнених схем Фейстеля. Дослідження можуть розглянути стійкість цих схем до різних видів атак, а також сумісність з різними форматами текстів.

ПЕРЕЛІК ПОСИЛАНЬ

1. M. Liskov, R. Rivest, D. Wagner. Tweakable Block Ciphers. — 2002. — Access mode: https://people.csail.mit.edu/rivest/pubs/LRW02.pdf.

2. M. Dworkin. Recommendation for Block Cipher Modes of Operation: Methods for Format-preserving encryption. — 2019. — Access mode: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38Gr1-draft.pdf.

3. T. Spies. Feistel Finite Set Encryption Mode. — 2008. — Access mode: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/proposed-modes/ffsem/ffsem-spec.pdf.

4. M. Bellare, P. Rogaway, T. Spies. The FFX Mode of Operation for Format-Preserving Encryption. — 2010. — Access mode: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/proposed-modes/ffx/ffx-spec.pdf.

5. E. Brier, T. Peyrin, J. Stern. BPS: a Format-Preserving Encryption Proposal. — 2010. — Access mode: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.694.918&rep=rep1&type=pdf.

6. Format-Preserving Encryption Algorithms Using Families of Tweakable Blockciphers / J.-K. Lee, B. Koo, W.-H. Kim, and D. Kwon. — 2014. — Access mode: https://www.semanticscholar.org/paper/Format-Preserving-Encryption-Algorithms-Using-of-Lee-Koo/ee68cb7a5fecac58bb8f0053ada0cb4ac21458ad.

7. M. Dworkin. Recommendation for Block Cipher. Modes of Operation:. — 2016. — Access mode: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf.

8. ADVANCED ENCRYPTION STANDARD (AES). — 2001. — Access mode: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

9. B. Durak, S. Vaudenay. Generic Round-Function-Recovery Attacks for Feistel Networks over Small Domains. — 2018. — Access mode: https://eprint.iacr.org/2018/108.

10. Recent Cryptanalysis of FF3. — 2017. — Access mode: https://csrc.nist.gov/news/2017/recent-cryptanalysis-of-ff3.

11. B. Durak, S. Vaudenay. Breaking the FF3 FormatPreserving Encryption.Proceedings of ESC 2017. — 2017. — Access mode: :https://www.cryptolux.org/mediawiki-esc2017/images/8/83/Proceedings_esc2017.pdf.

12. T. Beyne. Linear Cryptanalysis of FF3-1 and FEA. — 2021. — Access mode: https://iacr.org/submit/files/slides/2021/crypto/crypto2021/83/slides.pdf.

13. W. Jang, S.-Y. Lee. A Format-preserving encryption FF1, FF3-1. — 2020. — Access mode: https://dl.acm.org/doi/pdf/10.1145/3341105.3373953.

14. Diffie W., Hellman M. E. Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard. — 1977.

15. J. Patarin. Security of Balanced and Unbalanced Feistel Schemes with NonLinear Equalities. — 2010. — Access mode: http://eprint.iacr.org/2010/293, 2010.

16. Hoang V. T., Rogaway P. On Generalized Feistel Networks. — 2010. — Access mode: https://eprint.iacr.org/2010/301.pdf.