Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему

Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів

Назва теми

Галузь знань 12 – Інформаційні технології»
Шифр. назва

Спеціальність 125 – Кібербезпека
Шифр. назва

КРМКБ.180125.22.01.01 ПЗ

Виконав: студент 2 курсу, група КБм-22-1 _____ Анікін В.А.
Підпис Ініціали, прізвище

Керівник к.т.н., доцент кафедри КБ _____ Муляр І.В.
Підпис Ініціали, прізвище

Нормоконтролер ст. викладач кафедри КБ _____ Мостовий С.В.
Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри кібербезпеки _____ Кльоц Ю.П.
Підпис, дата Ініціали, прізвище

15 грудня 2023 р.

Хмельницький, 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КІБЕРБЕЗПЕКИ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Спеціальність 125 КІБЕРБЕЗПЕКА

Освітня програма КІБЕРБЕЗПЕКА

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"30" 08 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Анікіну Володимиру Андрійовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів
Керівник роботи Муляр Ігор Володимирович
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання
кандидат технічних наук, доцент
Затверджена наказом № 30 ректора університету, додаток №25 від 15.08.2023

2. Строк подання студентом проєкту (роботи) на кафедру 17.11.2023

3. Вихідні дані до проекту (роботи) Криптографічний захист інформації, метод розгортання криптографічного ключа, симетричне шифрування

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Дослідження теоретичної бази предметної області, та актуальних наукових досягнень у сфері сучасної криптографії. Постановка задачі дослідження. Математична модель процесу розгортання криптографічного ключа у нелінійних криптосистемах. Метод розгортання криптографічного ключа для використання у нелінійних криптосистемах. Прикладне застосування методу розгортання криптографічного ключа. Висновки.

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

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали і посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В. Старший викладач кафедри кібербезпеки		

7. Дата видачі завдання «01» вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Вибір напряму дослідження і узгодження тематики КРМ з керівником	01.06.2023	
2	Ознайомлення з предметною областю; формулювання мети і задач дослідження; визначення об'єкта і предмета дослідження	12.09.2023	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	24.09.2023	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	06.10.2023	
5	Робота над розділом 3 – розробка алгоритмів і технологій, їх аналіз	17.10.2023	
6	Робота над розділом 4 – апробація запропонованих рішень	13.11.2023	
7	Узгодження отриманих результатів, оформлення пояснювальної записки згідно вимог	15.11.2023	
8	Попередній захист роботи	17.11.2023	
9	Захист роботи на засіданні ЕК	06.12.2023	

Студент _____ В.А. Анікін
Підпис Ініціали, прізвище

Керівник проекту (роботи) _____ І.В. Муляр
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів

Автор роботи: Анікін Володимир Андрійович

Керівник роботи: к.т.н., доц. Муляр Ігор Володимирович

Загальний обсяг роботи: 79 сторінок, 31 рисунки, 3 таблиця, 2 додатки, 54 посилань.

Ключові слова: криптографія, розгортання ключа, нелінійна криптографія.

Метою даної роботи є розробка методу розгортання криптографічного ключа для нелінійних симетричних криптосистем, задля усунення недоліків криптосистем на основі нелінійних криптографічних примітивів.

В роботі запропоновано метод розгортання криптографічного ключа, для нелінійних криптосистем, що складається з шести кроків. Даний метод дозволяє безпечно генерувати модифікатори для нелінійних криптографічних примітивів. Побудовано математичну модель процесу розгортання ключів для нелінійних криптосистем.

01.12.2023 р.

ANNOTATION

Theme of qualification work: Cryptographic key deployment method for use in cryptosystems based on nonlinear cryptographic primitives

Author of the work: Anikin Volodymyr Andriiovych

Mentor: Ph.D. Muliar Ihor Volodymyrovych

Total volume of work: 79 pages, 31 figures, 3 tables, 2 appendices, 54 links

Keywords: cryptography, key expansion, nonlinear cryptography.

The purpose of this work is to develop a method of deploying a cryptographic key for nonlinear symmetric cryptosystems to eliminate the shortcomings of cryptosystems based on nonlinear cryptographic primitives.

The paper proposes a method of deploying a cryptographic key for nonlinear cryptosystems consisting of six steps. This method allows you to safely generate modifiers for non-linear cryptographic primitives. A mathematical model of the key deployment process for non-linear cryptosystems has been built.

01.12.2023 p.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

АГВП – Алгоритм генерації вектору повідомлення

АГК – Алгоритм генерації ключа

АПЗ – Апаратно-програмне забезпечення

АКП – Алгоритм кінцевого перетворення

ЕЦП – Електронний цифровий підпис

ДСТУ – Державний стандарт України

ІТ – Інформаційні технології

ІКС – Інформаційно-комунікаційна система

МФ – Мережа Фейстеля

ОС – Операційна система

КСЗІ – Комплексна система захисту інформації

ПЗ – Програмне забезпечення

AES – Advanced Encryption Standard

DES – Data Encryption Standard

IDEA – International Data Encryption Algorithm

PES – Proposed Encryption Standard

TEA – Tiny Encryption Algorithm

ВСТУП

Сучасна криптологія є важливою науковою дисципліною із широкою сферою практичного використання. Сучасна криптографія є основою безпечного зв'язку, електронного банкінгу, корпоративних комунікацій тощо. На їх основі будуються багаторівневі криптографічні протоколи, що забезпечують високий рівень захисту інформації.

З іншого боку, розвиток криптографії також призводить до розвитку криптоаналізу, оскільки захист та приховування деяких даних обов'язково викликає до себе інтерес. Щосекунди безліч об'єктів мережевої інфраструктури, сховищ даних та комп'ютерних систем піддаються різноманітним хакерським атакам, де можливість взлому криптографічних засобів захисту є одним із ключових факторів їх успішності.

Сучасні блочні та потокові алгоритми шифрування є результатом еволюції класичної криптографії, доповненої стрімким ростом обчислювальних потужностей комп'ютерної техніки, які з одного боку дають можливість створювати більш складні та ресурсоємні алгоритми шифрування, а з іншого – змушують постійно враховувати можливість складних математичних, статистичних досліджень, атак грубої сили та інших інструментів криптоаналізу із використанням тих ж технічних потужностей.

На рівні із криптографією, стеганографія також із давніх часів є важливою складовою у захисті інформації, оскільки маскування самого факту наявності деякої конфіденційної інформації може убезпечити її від атак. Комплексне систематизоване поєднання сучасної криптографії та стеганографії в рамках криптографічних систем та протоколів здатне забезпечити високий рівень захисту інформації.

Криптографічний ключ є невід'ємним елементом криптосистеми, що притаманний як усім, без виключення, сучасним системам, так і класичній криптографії і навіть старим докласичним протокриптографічним засобам захисту інформації.

Основними вимогами до нього є відносно невеликий розмір, неможливість штучного відновлення, відповідність вимогам криптосистеми, в якій він застосовується. Наявність слабких ключів є серйозною проблемою у багатьох сучасних криптосистемах.

Деякі криптографічні системи, зокрема нелінійні, мають специфічні вимоги до криптографічних ключів, зокрема у питаннях генерації раундових ключів, модифікаторів тощо.

Можливість ускладнення процесу генерації та застосування криптографічного ключа, шляхом створення особливих алгоритмів його розгортання із систематизацією їх використання, в рамках деякої криптосистеми чи криптографічного протоколу, дозволить в цілому покращити їх криптографічну стійкість.

Таким чином, дослідження у сфері розгортання криптографічного ключа для використання у сучасних блочних симетричних криптографічних системах є актуальними, оскільки вони дозволять підвищити стійкість даних криптосистем та протоколів шифрування, ускладнять криптоаналіз даних. Створений метод розгортання криптографічного ключа ідеально поєднається із криптосистемами на основі нелінійних криптографічних примітивів, оскільки включатиме в себе також можливість генерації модифікаторів нелінійності для даних систем. Це, в свою чергу, усуне серйозний недолік нелінійних криптосистем.

Мета кваліфікаційної роботи: розробити метод розгортання криптографічного ключа для нелінійних симетричних криптосистем, задля усунення недоліків криптосистем на основі нелінійних криптографічних примітивів.

Для досягнення мети кваліфікаційної роботи вирішено наступний перелік завдань:

1. Досліджено предметну область за напрямком сучасної симетричної нелінійної криптографії.

2. Сформовано перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах.

3. Побудовано математичну модель процесу розгортання криптографічного ключа у нелінійній криптосистемі.

4. Синтезовано існуючі методи розгортання криптографічного ключа та криптостійких генераторів псевдо-випадкових послідовностей.

5. Створено метод розгортання криптографічного ключа, відповідно сформованих вимог та на основі результатів синтезу існуючих методів.

6. Апробовано створений метод розгортання криптографічного ключа у симетричних криптосистемах на основі нелінійних криптографічних примітивів.

7. Апробовано метод розгортання криптографічного ключа у криптографічних протоколах.

8. Підведено підсумки дослідження.

Об’єктом дослідження є процес розгортання криптографічних ключів в рамках процесу симетричного шифрування інформації в інформаційно-комунікаційних системах.

Предметом дослідження є методи та алгоритми розгортання криптографічного ключа у криптографічних засобах захисту інформації, зокрема побудованих на основі нелінійних криптографічних примітивів.

Наукова новизна:

1. Вдосконалено математичну модель процесу розгортання криптографічного ключа, адаптовано її до схеми роботи криптосистем на основі нелінійних криптографічних примітивів.

2. Створено метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів.

Методи дослідження: порівняння, синтез, апарат теорії алгоритмів, теорії захисту інформації, комп’ютерного та математичного моделювання, системного аналізу, експеримент.

Практична цінність: результати дослідження дають можливість усунути значні недоліки нелінійних симетричних криптосистем, що є логічним продовженням теми нелінійної криптографії, висвітленої у попередніх публікаціях.

Розроблений метод полегшить практичне використання даних криптосистем, а також може бути використаний для покращення інших сучасних засобів криптографічного захисту, або при розробці нових криптосистем, або протоколів шифрування.

За напрямком кваліфікаційної роботи опубліковано:

– 1 фахова стаття;
– 1 стаття Scopus (очікує на індексацію);
– 3 тези доповідей конференцій;
– 1 конкурсна студентська наукова робота.

1 ДОСЛІДЖЕННЯ ТЕОРЕТИЧНОЇ БАЗИ ПРЕДМЕТНОЇ ОБЛАСТІ, ТА АКТУАЛЬНИХ НАУКОВИХ ДОСЯГНЕНЬ У СФЕРІ СУЧАСНОЇ КРИПТОГРАФІЇ

1.1 Розвиток класичної симетричної криптографії

Перед початком дослідження дамо визначення основним термінам та поняттям, що будуть використовуватися в даній роботі.

Криптологія – це наука, що вивчає методи криптографічного захисту інформації: шифрування, дешифрування, криптографічне хешування, а також методи дослідження зашифрованих даних з метою зламу. Сам термін «криптологія» вперше виник у вжитку у XVII столітті, а в сучасному значенні був введений американським криптоаналітиком Вільямом Фрідманом на початку XIX століття [1].

Криптологія вивчає криптографічний захист інформації в цілому, можливості його обходу, зламу, з використанням різних методів аналізу та дослідження шифрованих повідомлень, зокрема з використанням електронно-обчислювальної, механічної та іншої техніки [1-4].

Відповідно до цього, криптологія поділяється на два основних розділи: криптографію та криптоаналіз. Обидві дисципліни зародились близько чотирьох тисяч років тому та розвивалися в тандемі: революційні відкриття в одній із них невідворотно змушували розвиватись іншу. Порушення балансу між ними, швидше за все, призвело б до зникнення, або докорінної перебудови їх обох [1].

Криптографія – це розділ криптології, що вивчає методи шифрування, дешифрування, хешування, автентифікації та перевірки цілісності інформації. Завданням криптографії зокрема є організація безпечного обміну інформацією, навіть за умов публічності та незахищеності каналу зв’язку між його вузлами із забезпеченням автентифікації та верифікації сторін комунікації та із гарантуванням цілісності надісланих повідомлень [3, 5].

Історично криптографія зародилась на початку першого тисячоліття до н.е., перші шифровані письмові джерела датуються близько 1900 року до н.е.

Серед найдавніших тайнописів виділяють давньоєгипетські, що використовували спеціальні ієрогліфи, та піктографії, давньогрецькі, зокрема відомі Скитала, диск Енея, квадрат Полібія, давньоримські, серед яких, напевно один із найвідоміших, шифр Цезаря, давньоіудейський Атбаш, давньоарабські та давньоазійські тощо [6].

Найдавніші українські шифровані письмові джерела датуються XII – XIII століттям та представлені простою та мудрою літореями, шифрами простої заміни тощо [1, 7].

Загалом криптографія давніх часів представляла собою мистецтво тайнопису, в якому філософи, воєначальники, або просто ентузіасти змагалися у винахідливості, вигадуючи різноманітні системи тайнописів, на основі деяких простих криптографічних примітивів, серед яких найбільш поширеними були проста перестановка, підстановка, заміна, моноалфавітні перетворення тощо.

В Середньовіччі здебільшого використовували здобутки попередніх епох у сфері криптографії: дипломати, купці, військові як правило використовували прості шифри заміни.

Одним із нововведень тих часів були омофонічні шифри, метою яких було заплутування розповсюдженого вже тоді методу криптоаналізу шляхом частотного аналізу текстів [7-8].

Виключенням в епоху Середньовіччя був арабський світ, який активно розвивався. Арабські вчені періоду IX – XV століття у своїх роботах провели достатньо серйозну роботу над систематизацією відомих на той час шифрів, та методів їх зламу. Можна навіть сказати що їх фундаментальні праці заклали основу криптографії та криптології в цілому, як повноцінної комплексної науки з систематизованим підходом.

В майбутньому все це стало підґрунтям класичної криптографії, що розвивалась у XV – п.п. XX століттях, тобто з часів Відродження до Нового часу,

коли розвивалась механіка, з’явились телеграф, радіо, а згодом і обчислювальні машини.

Одним із здобутків криптографії у цей час став перехід до поліалфавітної схеми шифрування, що відзначився великою кількістю подібних систем шифрування [7, 9].

Найвідоміші серед них:

– диск Альберті;

– шифр Віженера;

– шифр Гронсфельда;

– шифр Вернама.

Одним з найбільш значущих науковців тих часів був Леон-Баттіста Альберті, що написав одну із фундаментальних праць з криптології під назвою «Трактат про шифри», де була запропонована концепція поліалфавітного шифрування. Фактично ця концепція задала вектор розвитку криптографії на сотні років вперед.

Продовжив та вдосконалив ідею Альберті Шотландський абат Йогенс Трітеміус. Він запропонував концепцію шифрування де кожна наступна літера шифротексту замінюється на шифровану літеру з наступного шифрувального алфавіту, на відміну від Альберті, який пропонував шифрувати кілька слів одним алфавітом, наступні кілька – іншим.

В одній зі своїх праць Трітеміус запропонував систему зсуву алфавітів у вигляді квадратної таблиці, в першому рядку якої знаходиться звичайний латинський алфавіт, а кожен наступний донизу алфавіт зміщений на 1 позицію вліво.

На сьогоднішній день таку систему шифрування прийнято називати шифром Віженера, на ім’я французького дипломата та криптографа Блеза де Віженера [7].

Віженер активно цікавився роботами Трітеміуса та Джованні Баттісти Белласо, перейняв їх ідеї та популяризував у своїх працях. Саме тому і квадрат

Тритеміуса і система шифрування Тритеміуса–Белласо сьогодні відома на загал під іменем Віженера.

В класичному вигляді шифр Віженера представляє собою квадратну таблицю зсувів деякого звичайного або перемішаного алфавіту, за допомогою якої здійснюється шифрування. Для вибору зсуву, з яким буде шифруватися окрема літера відкритого тексту вводиться деяке секретне слово, фраза, речення або навіть цілий текст, літери якого співвідносяться із літерами відкритого тексту. Фактично це таємне слово, або група слів є криптографічним ключем даного алгоритму шифрування. Якщо довжини ключа не вистачає – ключове слово просто дублюється стільки раз підряд, щоб заповнити всю довжину повідомлення, яке слід зашифрувати. Далі літера відкритого повідомлення замінюється на відповідну їй літеру із таблиці з рядка, що починається із відповідній літері ключа (рис. 1.1).

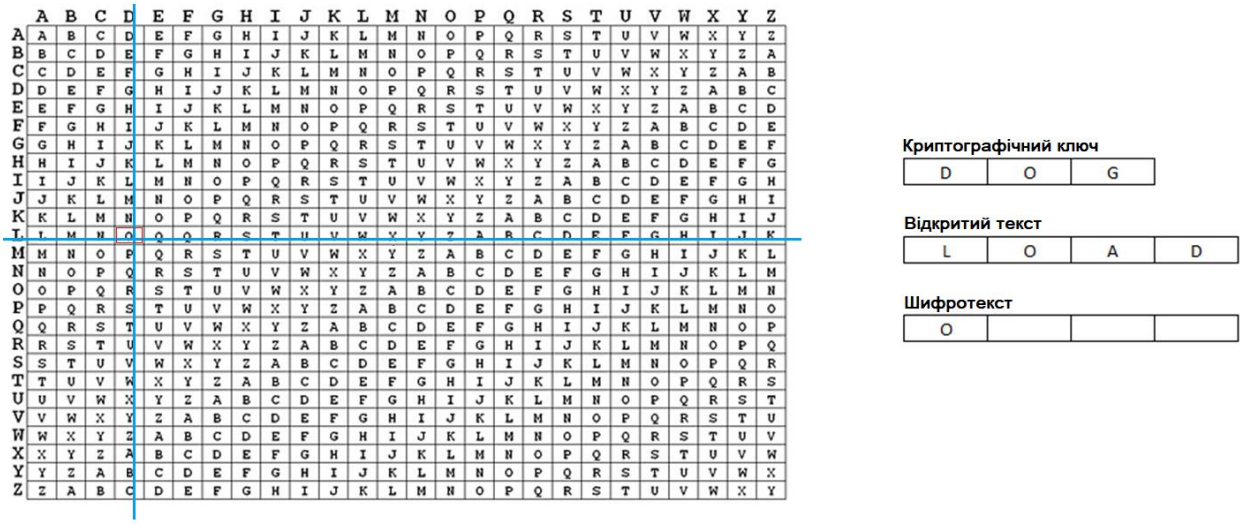


Рисунок 1.1 – Схема роботи шифру Віженера

Проте сам Віженер розвивав ідею більш складної системи генерації криптографічного ключа даного алгоритму, ніж просте повторення ключового слова.

Одним із запропонованих ним варіантів використання даного алгоритму був варіант шифрування із автоматичним вибором ключа, де пропонувалось зокрема

створювати нові ключі в процесі шифрування, на основі щойно зашифрованих слів. Набагато пізніше цю ідею розвине Хорст Фейстель, у своїй системі шифрування [7, 9-11].

Іншим напрямком розвитку шифру було збільшення розміру криптографічного ключа. Сам Віженер пропонував використовувати деякі великі сторонні тексти для шифрування. Також пізніше американський математик та криптоаналітик Клод Шенон у своїй праці «Математична теорія криптографії» довів абсолютну криптографічну стійкість шифру Віженера за умови що довжина криптографічного ключа без повторів рівна, або більша довжині відкритого повідомлення. В такому випадку криптоаналітик отримавши виключно шифрований текст, не зможе отримати жодної інформації про зашифроване повідомлення [12].

Саме така схема шифрування була запропонована Гілбертом Вернамом у 1917 році, для телеграфного зв'язку [13]. Вона отримала назву «Шифр Вернама» (рис. 1.2).

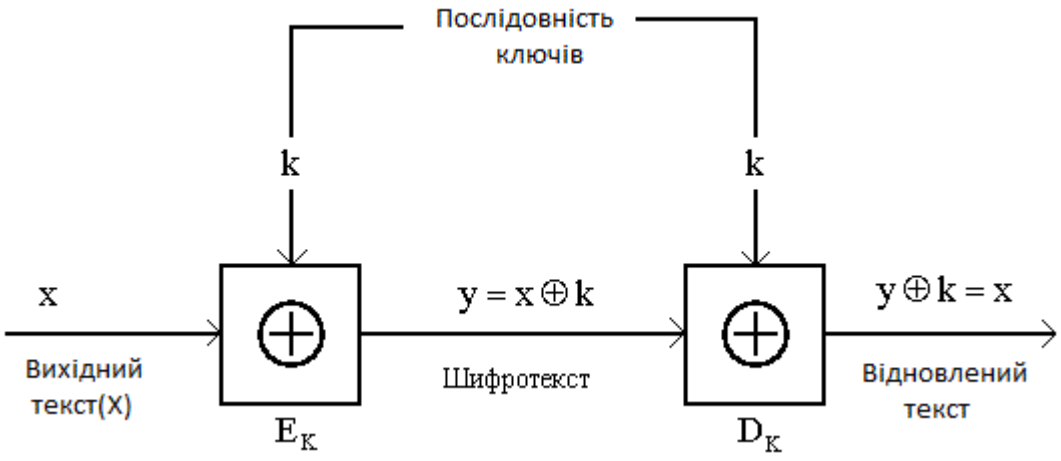


Рисунок 1.2 – Схема роботи шифру Вернама

В сучасній криптографії такий метод шифрування називається гамування. Дана криптосистема є прикладом криптосистеми з абсолютною криптостійкістю, проте практичність її використання викликає багато складнощів. Серед головних її недоліків:

– необхідність істино випадкового генератора послідовностей;

– необхідність таємної передачі криптографічного ключа, розмір якого рівний розміру таємного повідомлення;

– висока чутливість до порушень правил шифрування, особливо щодо повторного використання гами.

Гамування використовується як складова частина симетричних потокових алгоритмів шифрування й до сьогодні, однак для дотримання належного рівня безпеки та криптографічної стійкості варто врахувати усі перелічені вище фактори [12, 14-16].

В період активного розвитку механіки та першої електротехніки великого поширення здобули роторні шифрувальні машини. Найвідомішою серед них, безумовно, була німецька Енігма, хоча й, звісно, була вона далеко не єдиною шифрувальною машиною такого типу. Подібною до Енігми була також британська шифрувальна машина «Турех», із рядом вдосконалень, відносно німецького аналогу. Так наприклад Енігма мала одну серйозну вразливість: в зашифрованому нею тексті літера не могла бути такою ж, як літера відкритого тексту. Тобто літера «А» у відкритому тексті ніколи не могла залишитись літерою «А» у шифрограмі, що порушувало розподіл літер у шифрованому тексті та в решті решт призвело до зламу методу шифрування Енігми [11, 17-19].

В цілому, абсолютна більшість роторних шифрувальних машин мали подібну конструкцію та принцип дії. В їх корпусі, паралельно один до одного, знаходились циліндри (ротори), кількість яких в різних машинах могла відрізнятися. Нижче знаходилась механічна клавіатура з символами алфавіту, а також деякій пристрій виводу, наприклад лампи з літерами, як у Енігми, або пристрій для перфорування паперової ленти, як у радянської М-125 «Фіалка» тощо [20].

Ротори мали деяку кількість наскрізних контактів, праворуч та ліворуч, проте всередині циліндра праві та ліві контакти були з’єднані заплутаним чином за допомогою провідників. Крайній циліндр мав контакти лише з одного боку, які замикались всередині нього один з одним. Таким чином подача електричного

струму на один із контактів крайнього циліндра замикала електричне коло з одним із інших контактів цього ж циліндра (рис. 1.3).

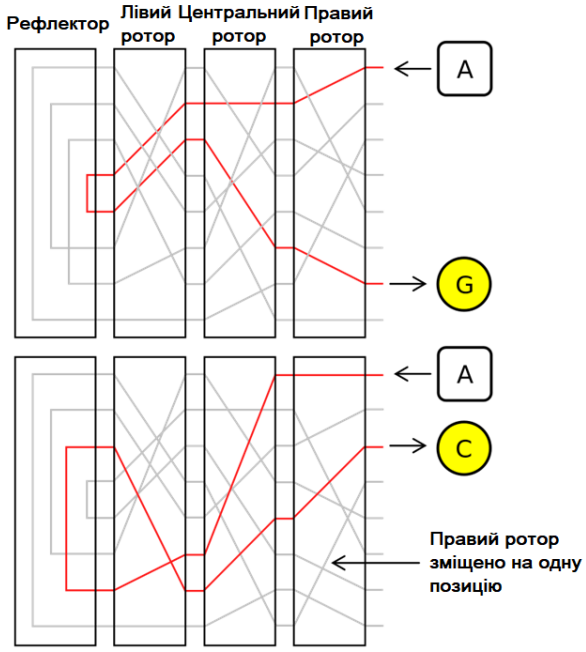


Рисунок 1.3 – Типова конструкція роторних шифрувальних машин

Ротори прокручувались певним чином після кожного введеного символу.

Якщо записати цей процес у вигляді схеми, то ми отримаємо класичну поліалфавітну схему шифрування, вкрай подібну до шифру Віженера, а вся система з роторами у ній, по-суті, виступає псевдовипадковим генератором гамми. Для статичного положення роторів в момент часу перед натисканням клавіші, для кожної із літер існує єдиний варіант заміни, тобто ми говоримо про звичайну моноалфавітну заміну. Прокручування роторів змінює «алфавіт», а оскільки це відбувається після кожної літери – ми й можемо говорити про велику подібність із шифром Віженера, а точніше про принципи шифрування, запропоновані Тритеміусом ще на початку XVI сторіччя.

Проте вже до закінчення Другої Світової Війни подібні системи шифрування показали себе не достатньо надійними і хоча ними продовжували користуватись до кінця XX століття, вже у 50-ті роки постала гостра проблема у принципово нових підходах до криптографічного захисту, що лише

підсилювалась стрімким розвитком електронно-обчислювальної, а згодом і набагато продуктивнішої комп’ютерної техніки [21-22].

Саме тоді були розроблені криптографічні концепції математичної криптографії, які стали основою всієї сучасної симетричної криптографії, були створені перші блочні шифри.

На відміну від поліалфавітних шифрів по типу шифру Віженера, які шифрують окремі символи, або числа, блочні алгоритми шифрування працюють з блоками даних певної довжини, над якими певну кількість разів виконуються певні математичні операції. Таким чином зміна навіть одного елемента всередині блоку може повністю змінити весь вихідний блок даних після шифрування [23, 24].

Реалізація блочних шифрів здійснювалась вже з використанням електронно-обчислювальної техніки, оскільки вони потребували значно вищої обчислювальної потужності ніж їх попередники і ручне шифрування, або шифрування за допомогою електронно-механічних приладів вже було на практичним.

Однією із нових криптосистем, розроблених саме в рамках даної концепції була криптосистема DES, що стала фундаментальним рішенням, яке залишалось актуальним не одне десятиліття [23-26].

DES – це блочний симетричний алгоритм шифрування, що працює на основі мережі Фейстеля, названої іменем її розробника – Хорста Фейстеля. Даний алгоритм був переможцем конкурсу Національного бюро стандартів США у 1974 році та був затверджений як стандарт шифрування, аж до 2000-го року, коли новим стандартом шифрування став AES [24, 27-28].

Мережа Фейстеля – це симетрична блокова схема шифрування цифрових даних, що передбачає ітеративне шифрування двох бітових підблоків однакового розміру шляхом гамування їх за певною закономірністю один з одним, а також з криптографічним ключем, з використанням деякої криптографічної функції (рис. 1.4).

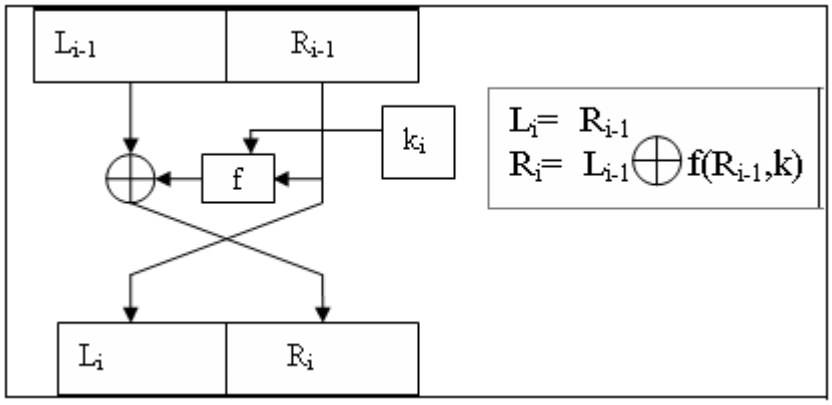


Рисунок 1.4 – Раунд шифрування мережі Фейстеля

На основі МФ розроблено величезну кількість сучасних алгоритмів шифрування, серед яких і вже згаданий DES, з різноманітними модифікаціями, а також такі алгоритми як IDEA, Blowfish, Twofish, TEA, ДСТУ ГОСТ 28147:2009 та інші [24, 29].

Алгоритм DES повністю базується на МФ, а його криптографічна функція виконує операції підстановки та перестановки, на основі SP-блоків (рис. 1.5).

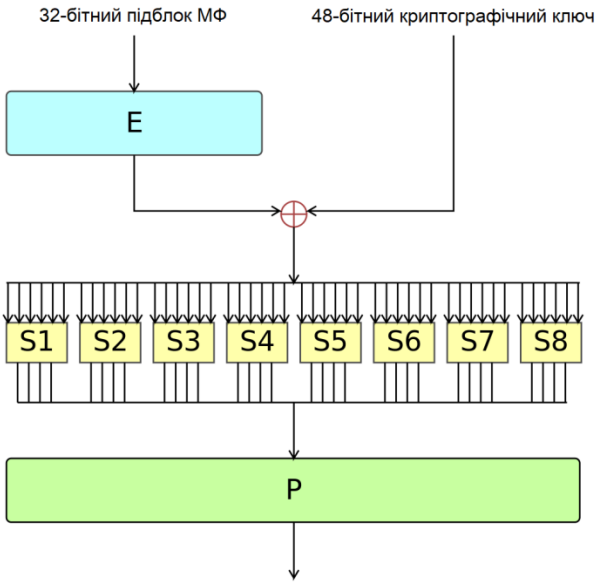


Рисунок 1.5 – Криптографічна функція DES

Криптосистема DES була революційною для свого часу та забезпечувала високий рівень захисту даних, проте комп’ютерна техніка не припиняла стрімко

розвиватися та вже до кінця 1980-х стало очевидною необхідність створення нових більш стійких алгоритмів шифрування. Існує багато симетричних блокових алгоритмів шифрування на основі МФ, що залишаються криптографічно стійкими й до сьогодні, але алгоритм шифрування, який було обрано як новий стандарт на заміну DES відійшов від цієї парадигми та використав новий підхід симетричного блокового шифрування.

AES – новий стандарт шифрування, що переміг у конкурсі алгоритмів шифрування США та у 2002 році замінив DES як основний стандарт шифрування, дозволений навіть для шифрування таємних документів [24, 30-31].

Цікавим є те що в своїй конструкції даний алгоритм не використовує МФ, а є виключно SP-мережею, що перегукується з методами шифрування класичної криптографії, доповненої сучасними криптографічними підходами (рис. 1.6).

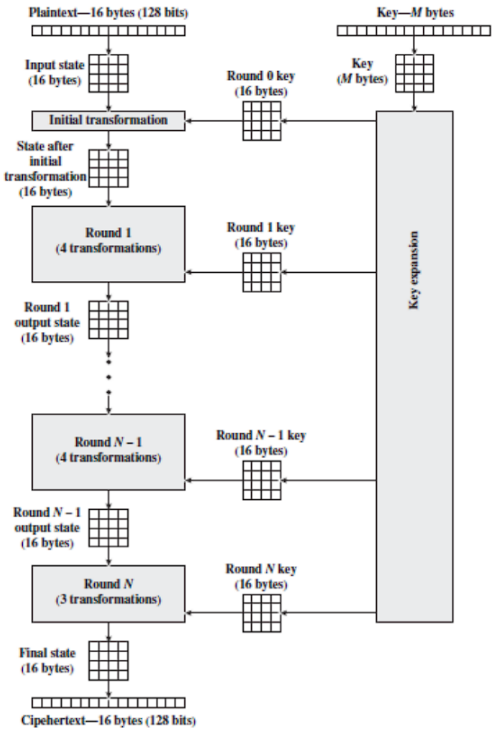


Рисунок 1.6 – Схема роботи шифру AES

Шифрування у алгоритмі AES відбувається за допомогою квадратної матриці 4x4, та складається з таких кроків:

– нелінійна заміна бітів матриці за допомогою S-блоку;

– зсув рядків матриці;

– перемішування стовпців матриці;

– XOR бітів матриці з раундовим ключем.

Даний алгоритм шифрування залишається стандартом у США та багатьох передових країнах світу до сьогодні [32].

Таким чином, ми проаналізували шлях розвитку криптографії від давніх часів до сьогодні та виявили ряд закономірностей. Зокрема, можна стверджувати що кожен наступний етап розвитку криптографії нерозривно пов’язаний із попереднім та, як правило є його вдосконаленням, з усуненням тих чи інших недоліків. Прикладом цього є «Енігма», яка по-суті була електро-механічною реалізацією шифру Віженера, хоча різниця між ними складала майже чотири сторіччя. Також сучасні симетричні блокові системи шифрування використовують досвід класичної криптографії, доповнюючи її досягненнями та потужностями комп’ютерної техніки, з математичним обґрунтуванням.

1.2 Сучасний криптоаналіз та типи криптографічних атак

Криптоаналіз - це розділ криптології, що вивчає методи дослідження шифрований повідомлень та самих методів криптографічного захисту інформації, з метою їх зламу, дешифровки повідомлень без криптографічного ключа, відновлення криптографічного ключа, компрометації ЕЦП тощо.

Розвиток криптоаналізу завжди був взаємозалежним до розвитку криптографії. Сам термін «криптоаналіз» було введено Вільямом Фрідманом у 1920 році [1-10, 33].

Основою криптоаналізу в давні часи був частотний аналіз, який точно був відомий на IX століття, оскільки був описаний у праці «Манускрипт про дешифрування криптографічних повідомлень» арабського філософа Аль-Кінді.

Розвиток даного методу криптоаналізу зробив вразливими усі моноалфавітні системи шифрування, а в комплексному застосуванні із тестами Казіскі – навіть на деякі поліалфавітні шифри. [33-35]

Сучасний криптографічний аналіз, та криптологія в цілому, ґрунтується на принципі, запровадженим англійським криптографом Огюстом Керкгоффсом, згідно якого передбачається що вся інформація про довільну криптосистему може бути публічною, включно із принципом роботи усіх її алгоритмів, та нюансами експлуатації, які криптоаналітик досконало знає в процесі своєї роботи. Криптографічна стійкість будь-якої криптосистеми повинна ґрунтуватись виключно на складності криптографічного ключа, який є таємним, коли все інше, окрім нього є добре відомим [36-38].

Згідно цього принципу, будь-яка криптосистема не повинна представляти собою чорний ящик, адже, по-перше, практично будь-який механізм, шифрувальний пристрій, шифрувальне ПЗ або АПЗ, може бути викрадено чи захоплено, після чого, методами реверсивної інженерії, можуть бути досконало встановлені принципи його роботи, що автоматично скомпрометує усі екземпляри даного криптографічного забезпечення.

По-друге, таємність криптосистеми може призвести до її недосконалої перевірки, адже усі відомі та популярні криптосистеми щодня піддаються новим і новим атакам, що ґрунтуються на передових теоріях та технологіях. Успішні атаки швидко стають відомі, а їх відсутність – важливе побічне підтвердження надійності системи шифрування.

В цілому виділяють чотири основних типи атак на симетричні криптосистеми:

- атаки виключно на шифрований текст;
- атаки на пари відкритого та шифрованого тексту;
- атаки на обраний відкритий текст;
- атаки на обраний шифротекст.

Атака на основі лише зашифрованого тексту є найскладнішою для криптоаналітика, оскільки передбачається що він немає жодної додаткової інформації, окрім самого шифрованого повідомлення (рис. 1.7).

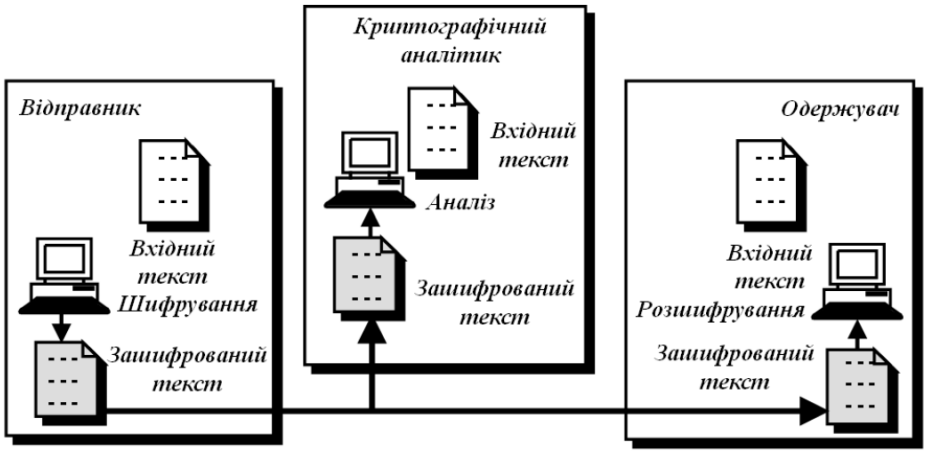


Рисунок 1.7 – Схема криптографічної атаки на основі виключно шифрованого тексту

Атака на базі пар «відкритий текст – шифротекст» є більш сприятливою для криптоаналітика. Вона можлива у випадку коли аналітик може отримувати пари відкритого тексту та відповідного йому шифротексту. При цьому передбачається що він також досконало знає алгоритми криптосистеми, невідомий йому виключно криптографічний ключ (рис. 1.8).

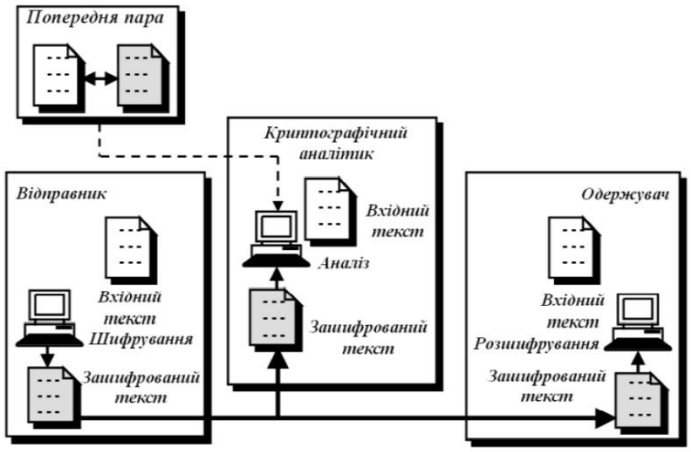


Рисунок 1.8 - Схема криптографічної атаки на основі пар відкритого та шифрованого тексту

Атаки на основі обраного відкритого та шифрованого тексту передбачають що аналітик прямо, чи опосередковано має можливість створювати шифротекст, відповідний конкретному відкритому тексту та навпаки, на власний розсуд, це є найбільш сприятливий тип атаки.

В усіх із перелічених типів також передбачається що аналітик не має можливості модифікувати повідомлення при передачі, чи запобігати їх отриманню. Він може лише їх читати та зберігати копію.

З розвитком математичної криптографії з'являлися й нові види математичного криптоаналізу, зокрема розвивався лінійний та диференціальний криптоаналіз [39-41].

Лінійний криптоаналіз – це метод криптоаналізу, що використовує лінійні наближення для побудови математичної моделі процесу шифрування криптосистеми. Лінійний криптоаналіз проводиться в два кроки:

– виявлення та систематизація відношень між відкритим текстом, шифротекстом та криптографічним ключем;

– використання виявлених відношень для відновлення криптографічного ключа, на основі пар відкритого та шифрованого тексту.

Теоретично більшість алгоритмів шифрування піддаються лінійному аналізу, за умови накопичення великої кількості пар відкритого та шифрованого тексту. Саме велика даних для аналізу є ключовим фактором для успішності даного методу і чим їх більше – тим краще.

Криптосистема DES може бути успішно зламана, на основі 2^{39} – 2^{41} пар текстів. При відновленні деяких бітів криптографічного ключа, паралельно із лінійним аналізом може бути запущена атака грубої сили на простий перебір бітів ключа що залишилися [42].

Диференціальний криптографічний аналіз – це метод криптоаналізу, що базується на основі вивчення різниці між змінами у відкритому тексті та відповідних їм змінам шифротексту [43].

Зазвичай диференціальний аналіз використовується в рамках атак на основі обраного відкритого тексту, хоча й існують випадки, за яких даний метод може

бути використаний і з атаками на основі відомого відкритого тексту і навіть на основі виключно шифротексту, проте це рідкість.

Між парами відкритого та шифрованого тексту знаходиться різниця, як правило за допомогою операції XOR. Ця різниця називається диференціалом. Далі аналітик проводить статистичні дослідження диференціалів з метою виявити залежності та закономірності розподілу.

Це доведений теоретичний метод криптоаналізу, проте практичне використання диференційного аналізу є складним завданням, оскільки він вимагає багато часу і великий об’єм даних [44-45].

Отже, ми розглянули основні типи криптографічних атак та методів сучасного криптоаналізу. Проектування будь-яких криптосистем, чи їх елементів вимагає знання актуальних можливостей криптоаналізу, з метою організації запобігання їм.

1.3 Криптосистеми на основі нелінійних криптографічних примітивів, їх переваги та недоліки

Для початку варто визначити термінологію, адже в різних джерелах термін «нелінійна криптосистема» може трактуватися по-різному. Щоб дати визначення поняттю «нелінійності», розглянемо протилежне йому.

Лінійна криптосистема - це система, побудована виключно з використанням лінійних криптографічних примітивів. Така криптосистема має лінійну залежність між відкритими даними та відповідним їм шифротексту. У випадку з лінійними блоковими симетричними криптосистемами для відкритого блоку A зашифрованого криптографічним ключем K існуватиме єдиний варіант шифрованого блоку A`. Справедливим в даному випадку буде й зворотне твердження, що для шифротексту A`, розшифрованого ключем K існує єдиний результат відкритого тексту A [46-50].

Лінійний криптографічний примітив - це певний криптографічний алгоритм, що є складовою частиною криптосистеми та який повертає передбачуване значення на виході при однакових вхідних даних (рис. 1.9).

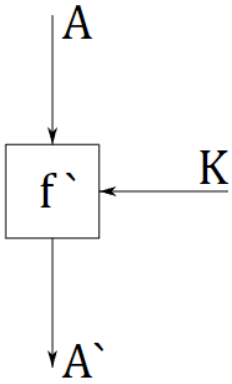


Рисунок 1.9 – Лінійний криптографічний примітив

Нелінійний криптографічний примітив – це певний криптографічний алгоритм, що є складовою частиною криптосистеми та окрім визначених параметрів має деякий модифікатор M, який також впливає на вихідне значення.

Нелінійність в даному випадку досягається за рахунок того що при ідентичних вхідних значеннях блоку A та ключа K ми матимемо не одне прогнозоване вихідне значення, а множину ймовірних значень, одне серед яких визначить саме модифікатор (рис. 1.10).

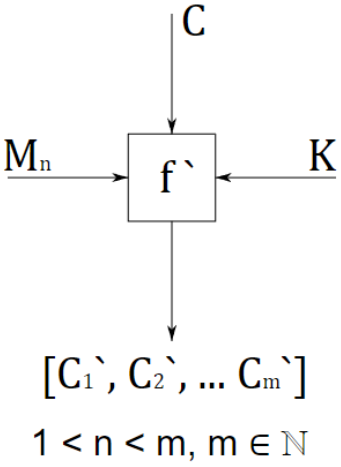


Рисунок 1.10 – Лінійний криптографічний примітив

В такій функції модифікатор може бути або певним стороннім значенням, яке передається ззовні, вихідним значенням деякої іншої функції з параметрами, чи взагалі випадковим значенням із заданого діапазону.

Нелінійна криптосистема – це система шифрування, у складі якої є хоча б один нелінійний криптографічний примітив, оскільки навіть один нелінійний криптографічний примітив в складі криптосистеми здатен створити розгалуження у вихідному шифротексті. Впровадження додаткових параметрів у криптосистему істотно розширює варіативність шифрування, додаючи стійкості зокрема перед лінійним криптоаналізом.

З одного боку це збільшує варіативність у процесі шифрування, а з іншого – дає можливість опосередковано впливати на отриманий шифротекст шляхом зміни модифікатора [46-50].

Перевагами подібних нелінійних криптосистем є:

– підвищення криптографічної стійкості, за рахунок того що кожен із вихідних блоків є не єдиним можливим варіантом, а одним із множини можливих, обраних модифікатором;

– ускладнення встановлення відповідностей між відкритим та шифрованим текстом;

– можливість додавати будь-яку кількість модифікаторів для криптосистеми, чи криптографічного протоколу;

– можливість управління вихідним шифротекстом, шляхом вибору найбільш підходящих модифікаторів, що відкриває зокрема стеганографічний потенціал використання.

Недоліками криптосистем, з використанням нелінійних криптографічних примітивів є:

– висока залежність від надійності генератора випадкових чисел, за умови використання випадкових модифікаторів;

– збільшення розміру вихідного шифротексту, за умови додавання модифікаторів до шифрованих блоків;

– суттєве збільшення розміру ключа, у випадку якщо модифікатори передаються як параметр ключа;

– збільшення обчислювального навантаження при шифруванні.

Таким чином, використання нелінійних криптографічних примітивів у складі криптосистеми, або функції шифрування створює варіативність вихідного шифротексту та в цілому підвищує криптографічну стійкість шифрування. При цьому для практичного використання подібних елементів необхідно мати просту, зручну та надійну систему генерації модифікаторів, яка б не розширювала криптографічний ключ та шифротекст багатократно, та дозволяла б ефективно використовувати нелінійні криптографічні примітиви.

1.4 Постановка задачі

Отже, ми проаналізували шлях розвитку криптографії від давніх часів до сьогодні та встановили що кожен наступний етап розвитку криптографії нерозривно пов’язаний із попереднім та є його вдосконаленням, з усуненням тих чи інших недоліків. Сучасні симетричні блокові системи шифрування використовують досвід класичної криптографії, доповнюючи її досягненнями та потужностями комп’ютерної техніки, з математичним обґрунтуванням.

Використання нелінійних криптографічних примітивів у криптосистемах чи криптопротоколах дозволяє в цілому покращити криптостійкійсть, а також надати нових властивостей, пов’язаних з розгалуженістю шифротексту.

Проте, разом з цим, використання нелінійних криптографічних примітивів є не дуже зручним та практичним через відсутність систематичного зручного та надійного підходу для генерації модифікаторів нелінійності, який, з одного боку, значно не розширював би криптографічний ключ та шифротексту, а з іншого був би ефективним та криптографічно стійким.

Відповідно до цього метою даної роботи було поставлено розробку методу розгортання криптографічного ключа для використання у криптосистемах, на

основі нелінійних криптографічних примітивів, який би дозволив вирішити перелічені недоліки.

Для досягнення поставленої мети необхідно вирішити наступний перелік завдань:

1. Дослідити предметну область за напрямком сучасної симетричної нелінійної криптографії.
2. Сформувати перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах.
3. Побудувати математичну модель процесу розгортання криптографічного ключа у нелінійній криптосистемі.
4. Синтезувати існуючі методи розгортання криптографічного ключа та криптостійких генераторів псевдо-випадкових послідовностей.
5. Створити метод розгортання криптографічного ключа, відповідно сформованих вимог та на основі результатів синтезу існуючих методів.
6. Апробувати створений метод розгортання криптографічного ключа у симетричних криптосистемах на основі нелінійних криптографічних примітивів.
7. Апробувати метод розгортання криптографічного ключа у криптографічних протоколах.
8. Підвести підсумки дослідження.

2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ РОЗГОРТАННЯ КРИПТОГРАФІЧНОГО КЛЮЧА У НЕЛІНІЙНИХ КРИПТОСИСТЕМАХ

2.1 Схема роботи криптосистем на основі нелінійних криптографічних примітивів

Складемо математичний опис криптосистеми на основі нелінійних криптографічних примітивів.

Загальна схема криптосистеми із нелінійним криптографічним примітивом складається із кількох аналогів криптоперетворення, що по-суті є лінійними криптографічними примітивами, та деяким перемикачем між ними, що працює на основі модифікатора m.

Усі можливі n криптоперетворень утворюють собою множину F:

$$F = \{f_1, f_2, \dots, f_n\},\ \ n \in \mathbb{N} \tag{2.1}$$

При чому кожне із криптоперетворень f_x повинно мати однакові за типом та розміром вхідні параметри, як правило це буде блок відкритих даних c, та раундовий криптографічний ключ k, та повертати однаковий за розміром шифрований блок даних c`. При цьому шифротексти c` також утворюють собою множину $\bar{C}$:

$$c` = f(c, k),\ f \in F, \tag{2.2}$$

$$\overline{C} = \{c_1,\ c_2, \dots, c_n\},\ n \in \mathbb{N} \tag{2.3}$$

Криптографічний примітив не може мати ідентичних криптоперетворень, а за умови випадкової генерації модифікаторів, ймовірність випадання кожного з них має бути рівною:

$$f_i \neq f_j,\ 1 \leq i,j \leq n,\quad i,j,n \in \mathbb{N},\ \ f_i, f_j \in F \tag{2.4}$$

Також слід зазначити, що при недопустимості однакових криптоперетворень, однакові шифротексти для двох різних функцій допускаються і не є проблемою, оскільки сам по собі шифротекст не повинен давати криптоаналітику жодної інформації про використане криптоперетворення, так само як ідентичність шифротекстів для однакових вхідних значень c та k, не повинна давати підстав вважати що однозначно було використано одне і те ж криптоперетворення f, що потенційно повинно ускладнити криптоаналіз на основі пар відкритого та шифрованого тексту, а також на основі вибраного відкритого тексту.

Криптоперетворення повинні бути повністю взаємозамінні між собою, а утворені ними шифровані блоки даних не повинні містити жодних ознак, які б вказували на те яким саме алгоритмом вони були виконані. Таким чином для аналітика не повинно бути створено можливості визначити який саме модифікатор було використано в тому чи іншому нелінійному криптографічному примітиві.

Загальна структура криптосистеми, побудованої на основі нелінійних криптографічних алгоритмів може включати в себе один, або кілька таких примітивів, модифікатори яких можуть братися або одні і ті ж самі, або різні (рис. 2.1).

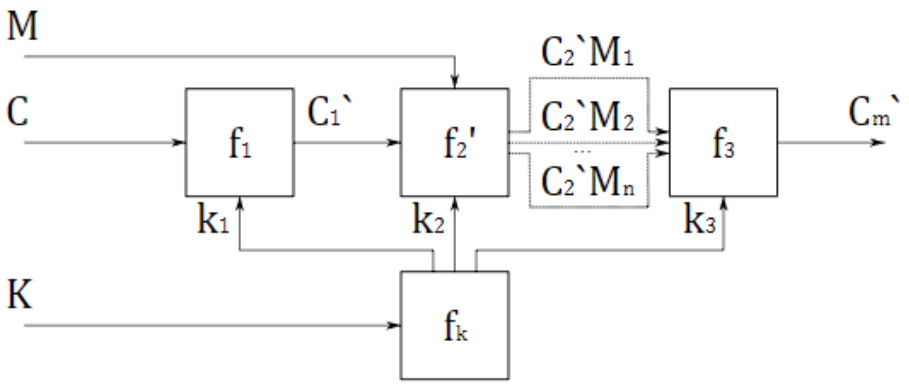


Рисунок 2.1 – Загальна схема криптосистеми з нелінійністю

Імплементація криптосистеми, побудованої на основі нелінійних криптографічних примітивів, також представлятиме собою точку входу із подальшим розгалуженням алгоритму на гілки. Гілка, за якою буде виконане шифрування, обирається перемикачем, на основі модифікатора. Після виконання криптографічного перетворення, гілки алгоритму сходяться в одну точку та повертають результат в однаковому розмірі та форматі (рис. 2.2).

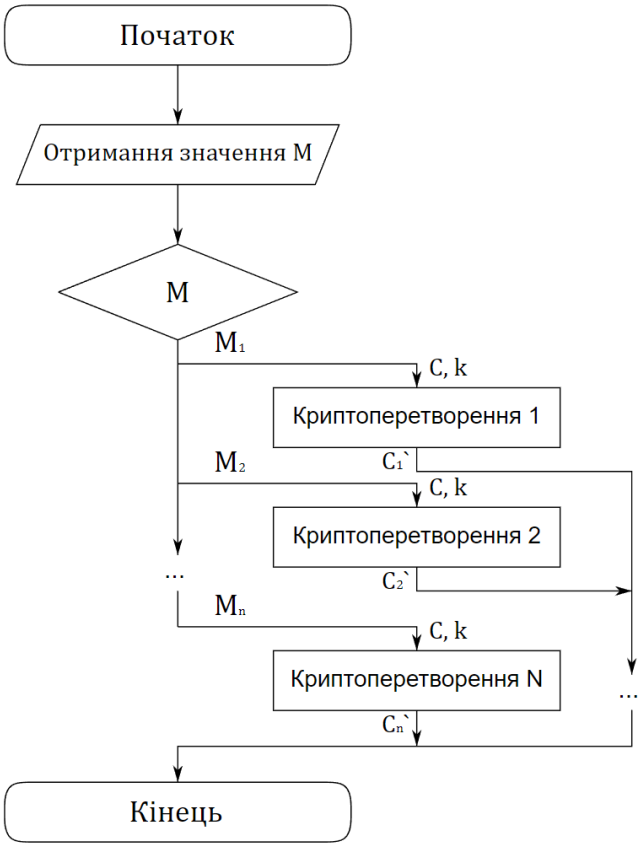


Рисунок 2.2 – Блок-схема будови нелінійного криптографічного алгоритму

Важливо зазначити що жоден елемент структури подібної криптосистеми не є таємним, у відповідності до принципу Керкгоффса. Криптографічна стійкість криптосистеми на основі нелінійних криптографічних примітивів базується на стійкості криптоперетворень, що доповнена їх псевдовипадковим застосуванням

Також важливим аспектом криптосистем на основі нелінійних криптографічних примітивів є спосіб генерації модифікатора, а також його

передача, оскільки ідентичну послідовність модифікаторів необхідно мати як на етапі шифрування повідомлення, так і на етапі його розшифровки.

Серед основних способів утворення модифікатора є:

1. Випадкова генерація модифікатора в процесі шифрування, збереження та передача через деякий сторонній захищений канал зв’язку;

2. Випадкова генерація модифікатора в процесі шифрування, злиття його із шифротекстом, за допомогою деякого АКП;

3. Передача набору модифікаторів як параметра криптографічного ключа;

4. Генерація модифікаторів алгоритмом розгортання ключа, на основі деякого сіду.

Перший спосіб є найбільш стійким із перелічених, проте найменш практичним, оскільки за наявності деякого достатньо захищеного каналу зв’язку, через який можна було б безпечно передати масив модифікаторів, то через нього, швидше за все, можна також безпечно передати саме повідомлення. Можливим винятком, за якого цей спосіб міг би бути оптимальним – це випадок коли безпечний канал зв’язку має обмеження щодо розміру даних, які він здатний пропустити, оскільки масив модифікаторів завжди буде суттєво менший за розміром, ніж шифроване повідомлення.

Другий спосіб є допустимим, проте він має цілий ряд недоліків, серед яких зокрема:

– суттєве розширення шифротексту, відносно відкритих даних, оскільки модифікатор необхідно дописувати для кожного блоку;

– передача модифікаторів разом із шифротекстом, навіть за умови їх «затирання» АКП, робить їх потенційно значно більш вразливими перед методами криптоаналізу;

– коефіцієнт розширення шифрованого тексту відносно відкритого, разом із знанням розміру звичайного блоку алгоритму шифрування, дає можливість встановити розмір модифікатора.

В цілому, використання даного способу утворення та передачі модифікаторів є прийнятним, проте потребує комплексного підходу, з урахуванням усіх можливих нюансів (рис. 2.3).

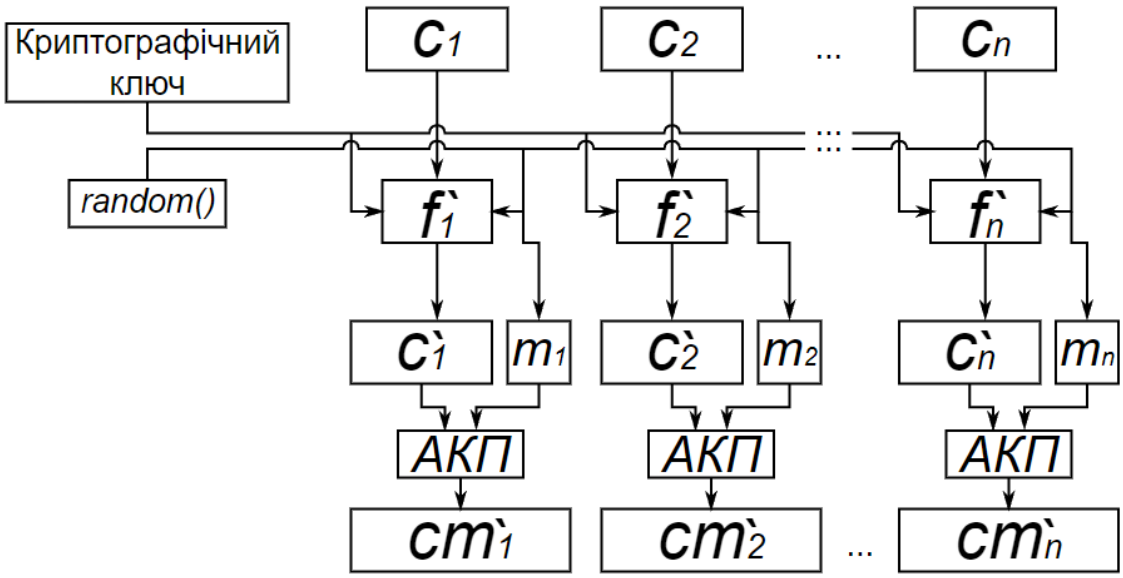


Рисунок 2.3 – Спосіб випадкового утворення модифікаторів та передачі їх разом із блоками шифрованих даних

Третій спосіб є достатньо надійним, з точки зору того, що криптоаналітик не зможе встановити модифікатори із шифрованого або відкритого тексту, проте має одну суттєву вразливість: усі повідомлення, зашифровані одним ключем, матимуть однакову послідовність модифікаторів, що робить їх вразливими перед, зокрема, лінійним аналізом. При застосуванні даного способу аналітик зможе достеменно заявити що $n-$ні перші блоки усіх повідомлень, зашифрованих однаковим ключем, завжди зашифровані одним і тим ж модифікатором.

Четвертий спосіб є найбільш оптимальним, оскільки він не розширяє шифротекст, та при цьому не дає аналітику можливості визначити послідовність використаних модифікаторів. Цей спосіб подібний до того як у потоковому симетричному алгоритмі шифрування розгортається псевдовипадкова гамма.

Саме цей спосіб утворення модифікатора найбільше цікавить нас в рамках даної роботи та є пріоритетним для дослідження.

Таким чином ми склали математичний опис криптосистеми на основі нелінійних криптографічних примітивів та розглянули основні способи утворення та передачі модифікаторів, серед яких найбільш оптимальним є генерація модифікаторів алгоритмом розгортання ключа, на основі сіду.

2.2 Вимоги до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах

Згідно складеного опису криптосистем на основі нелінійних криптографічних примітивів, складемо перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів, якими користуватимемось надалі.

Вимоги до процесу розгортання криптографічного ключа ми згрупуємо за наступними категоріями:

- загальні вимоги щодо архітектури процесу;
- криптографічні вимоги;
- вимоги швидкодії та оптимізації.

Визначені вимоги необхідні для побудови математичної моделі процесу розгортання криптографічного ключа у нелінійних криптосистемах та для розробки методу розгортання ключа, на її основі.

Загальні вимоги щодо архітектури процесу розгортання ключа у криптосистемах з нелінійністю безпосередньо пов’язані із схемою роботи подібних криптосистем, що описана у попередньому розділі. Дана категорія вимог описує концептивно-технічну сторону процесу розгортання та встановлює необхідні правила взаємодії даного процесу із процесом шифрування, в рамках довільної криптосистеми, із використанням нелінійних криптографічних примітивів.

Перелік загальних вимог щодо архітектури процесу розгортання ключа оформлено у вигляді таблиці (табл. 2.1).

Таблиця 2.1 – Загальні вимоги щодо архітектури процесу

№	Вимога	Пояснення
1	Розгортання має проводитись на основі загального криптографічного ключа, або його параметра	Процес розгортання має базуватись на основі деякого секретного параметру. В будь-якій криптосистемі єдиним таким параметром має бути криптографічний ключ. Задля запобігання компрометації ключа, розгортання може проводитись на основі деякого параметра в його складі. Це не протирічить принципу Керкгоффса.
2	В результаті процесу розгортання повинен утворюватися модифікатор для конкретного блоку	Оскільки дане розгортання має використовуватись як при шифруванні, так і при дешифруванні, для різних режимів шифрування (таких як CBC, ECB, PCBC тощо), процес розгортання повинен повертати модифікатор вказаного розміру для довільного блоку.
3	Процес розгортання повинен розрізнятися для кожного повідомлення	Якщо процес розгортання братиме за основу виключно параметр загального ключа, то для всіх повідомлень, зашифрованих одним ключем, послідовність модифікаторів буде однаковою, що створює вже згадані раніше вразливості. Отже, процес розгортання повинен вносити деякі відмінності для кожного повідомлення.
4	Процес розгортання повинен працювати із різними системами шифрування	Отже, розміри вхідних та вихідних даних можуть відрізнятися, при цьому процес розгортання повинен виконуватись універсально.

Криптографічні вимоги розгортання криптографічного ключа стосуються стійкості та безпекових характеристик даного процесу. Їх дотримання є критично важливим, оскільки будь-які порушення вимог цієї категорії можуть не тільки нівелювати переваги та позитивні сторони нелінійних криптографічних примітивів, але й скомпрометувати криптографічний захист в цілому, створивши явні чи приховані вразливості у криптосистемі.

Для опису даної категорії вимог нам необхідні наступні формули:

$$f_{\text{розгорт.}}(k,i) \rightarrow m_i, \quad f(m_i) \nrightarrow k, \tag{2.5}$$

$$P(m_i) = P(m_j) = \frac{1}{n},\ 1 < i,j < n,\ \ i,j,n \in \mathbb{N} \tag{2.6}$$

Криптографічні вимоги процесу розгортання ключа також оформимо у вигляді таблиці (табл. 2.2).

Таблиця 2.2 – Криптографічні вимоги щодо архітектури процесу

№	Вимога	Пояснення
1	2	3
1	Процес розгортання не повинен компрометувати жоден модуль криптосистеми	Даний процес не повинен створювати вразливостей, які б призвели до зниження, чи нівелювання криптографічної стійкості процесу шифрування. Рекомендується сепарувати ці два процеси, з метою уникнення непередбачуваних наслідків їх впливу один на одного.
2	Розгортання повинно бути одностороннім	Маючи номер блоку та криптографічний ключ, ми повинні з легкістю отримати модифікатор для даного блоку, однак маючи модифікатор для конкретного блоку, ми жодним чином не повинні мати можливості отримати криптографічний ключ, або будь-яку іншу інформацію про процес розгортання, згідно формули (2.5).
3	Усі модифікатори, із множини можливих, мають бути рівноймовірними	Не залежно від кількості елементів у множині F, ймовірність обрання кожного із них повинна бути однаковою. Відповідно до цього, ймовірність кожного модифікатора m в процесі розгортання ключа має бути рівною, без найменших викривлень на користь якогось із них, згідно формули (2.6).

Кінець таблиці 2.2

1	2	3
4	Процес розгортання повинен бути стійким до зламу та прогнозування результатів	Єдиним способом отримати модифікатор m для довільного блоку в процесі шифрування має бути процес розгортання, на основі криптографічного ключа. Отримання модифікатора жодним іншим чином, в обхід даного процесу, шляхом прогнозування на основі відомих попередніх модифікаторів, чи будь-яким іншим чином, є недопустимим, оскільки одразу практично нівелює нелінійність криптосистеми.
5	Повторні результати процесу розгортання ключа повинні бути ідентичними	При багатократному запуску процесу розгортання ключа для одного і того ж повідомлення, з одним і тим ж криптографічним ключем, модифікатори для кожного блоку даних повинні бути ідентичними щоразу, незалежно від жодних інших чинників.

Вимоги швидкодії та оптимізації процесу розгортання ключа для криптосистем на основі нелінійних криптографічних примітивів зводяться до трьох ключових:

1. Процес розгортання не повинен бути перенавантаженим зайвою логікою, та будь-якими іншими надлишковими субпроцесами, які б при проектуванні програмних алгоритмів на його основі створили б надмірне навантаження на електронно-обчислювальну систему.

2. Процес розгортання повинен складатись з мінімально-можливої кількості субпроцесів, при збереженні повноцінного функціонування. Наявні субпроцеси повинні бути чітко та логічно структуровані, а за необхідності розбиті на ще менші етапи, що в майбутньому відобразиться в чіткій структурі алгоритмів, на етапі проектування методу та подальшої його імплементації.

3. У випадку, якщо нелінійна криптосистема створювалась на основі іншої, самостійної криптосистеми, швидкодія процесу шифрування із усіма

модифікаціями, включно із процесом розгортання ключа, не повинна бути порядково повільніша, ніж швидкодія шифрування без модифікацій.

Варто також враховувати що процес розгортання ключа, задля отримання модифікатора для нелінійного криптографічного примітиву, викликатиметься велику кількість разів, в ході шифрування. Кількість викликів даного процесу $n_{\text{викл.}}$ в рамках процесу шифрування можна обрахувати із довжини повідомлення у бітах w_c та розміру одного блоку даних b_c, за формулою:

$$n_{\text{викл.}} = \frac{w_c}{b_c},\ n_{\text{викл.}} \in \mathbb{N} \tag{2.7}$$

При чому дане число завжди округлюється в більшу сторону. Тобто при шифруванні середньостатистичного зображення, розміром 2 мегабайта, що рівне 2097152 байтам та відповідно 16777216 біт, нелінійною криптосистемою, на основі шифру DES, розмір блоку якого складає 64 біт, згідно формули (2.7), кількість викликів процесу розгортання ключа складатиме 262144. При шифруванні об’єктів подібного, або й значно більшого, розміру навіть найменші логічні надлишковості призведуть до суттєвих та помітних затримок в роботі, що значно уповільнить процес шифрування, по відношенню до першопочаткового.

Таким чином, ми склали перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах за трьома категоріями: загальні вимоги щодо архітектури процесу, криптографічні вимоги та вимоги швидкодії та оптимізації.

2.3 Побудова математичної моделі процесу розгортання криптографічного ключа у нелінійній криптосистемі

Для початку наведемо типову модель розгортання криптографічного ключа на основі відомих рішень.

Процес розгортання ключа – це складова частина процесу симетричного шифрування: як блокового, так і потокового.

У потокових системах шифрування процес розгортання ключа як правило використовується для генерування гамми, яка практично в реальному часі накладається на потік відкритих даних за допомогою операції XOR. Прикладом такої потокової криптосистеми є шифр RC4, в складі якого є генератор гамми.

Типова логіка потокового процесу шифрування та розшифрування будується на двох синхронізованих генераторах гамми, які працюють автономно, та при цьому генерують однакові псевдо-випадкові послідовності, на основі деякого сід-ключа (рис. 2.4).

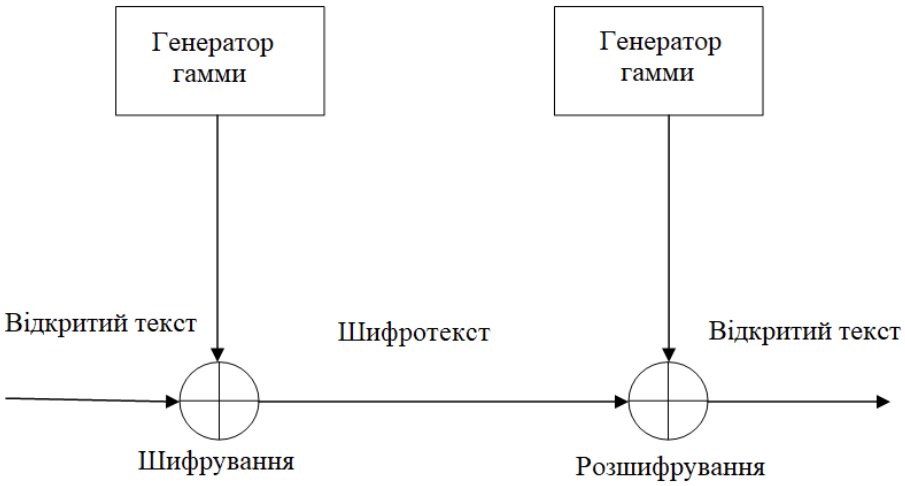


Рисунок 2.4 – Шифрування та розшифрування у потокових алгоритмах

У будь яких систем такого типу критично важливим чинником їх захищеності є стійкість генератора псевдо-випадкових чисел. Він має генерувати послідовності, які жоден ефективний алгоритм не повинен мати можливість відрізнити від повністю випадкових послідовностей за практичний час. Тобто жодне статистичне дослідження не повинне відрізняти отриману послідовність псевдовипадкових чисел від дійсно випадкової послідовності.

В блокових системах шифрування процеси розгортання ключів часта використовуються для отримання раундових підключів, або інших параметрів в

процесі шифрування. При цьому варто зазначити, що використання подібних блоків несе й певну загрозу.

Одним із рішень, що часто застосовуються для вирішення подібних задач є використання криптографічного хешування, яке є односторонньо-направленим та не дозволяє відновити вхідні параметри, на основі вихідних.

Для побудови математичної моделі процесу розгортання криптографічного ключа у нелінійній криптосистемі ми використаємо перелічені підходи, що вже використовуються у існуючих криптосистемах.

В процесі побудови слід перевіряти використані рішення на відповідність вимог, описаних у попередньому пункті.

Основою процесу розгортання ключа буде криптостійкий генератор псевдовипадкових чисел, який створюватиме послідовність значень, на основі сід-параметру ключа, та вектору повідомлення що шифрується. Вектором виступатиме деяке випадкове значення v_c, що створюється перед початком шифрування повідомлення.

Даний параметр необхідний, відповідно до вимоги про різницю генерації для кожного повідомлення.

За результатами роботи генератора псевдовипадкових послідовностей ми отримаємо проміжну криптографічну сіль $r`$:

$$r` = f_{rand}(k, v_c) \tag{2.8}$$

Криптографічна сіль – це деяке псевдо-випадкове значення, що використовуватиметься як основа для подальших операцій, зокрема хешування. До нього додаватимуться інші необхідні параметри, зокрема ідентифікатор блоку, в якості якого може використовуватись просто його порядковий номер в чистому вигляді, або з додаванням деякої константи задля підвищення ентропії.

З метою дотримання вимог оптимізації, криптографічна сіль може використовуватись для кількох блоків, а не перестворюватись щоразу.

Хеш-функція створюватиме хеш-зліпок фіксованого розміру, на основі криптографічної солі та ідентифікатора блоку. Даний елемент забезпечуватиме вимогу одностороннього перетворення в процесі розгортання ключа. Таким чином отримані результати буде вкрай важко реверсувати до використаних початкових значень, відповідно до формули (2.5).

Після створення хеш-зліпку на основі криптографічної солі та ідентифікатору блоку, необхідно привести отримане значення до розміру модифікатору, причому обов'язково враховуючи вимогу рівноймовірності усіх можливих ідентифікаторів, згідно формули (2.6).

Поєднавши усі описані елементи, отримаємо модель розгортання криптографічного ключа для нелінійних криптосистем (рис. 2.5).

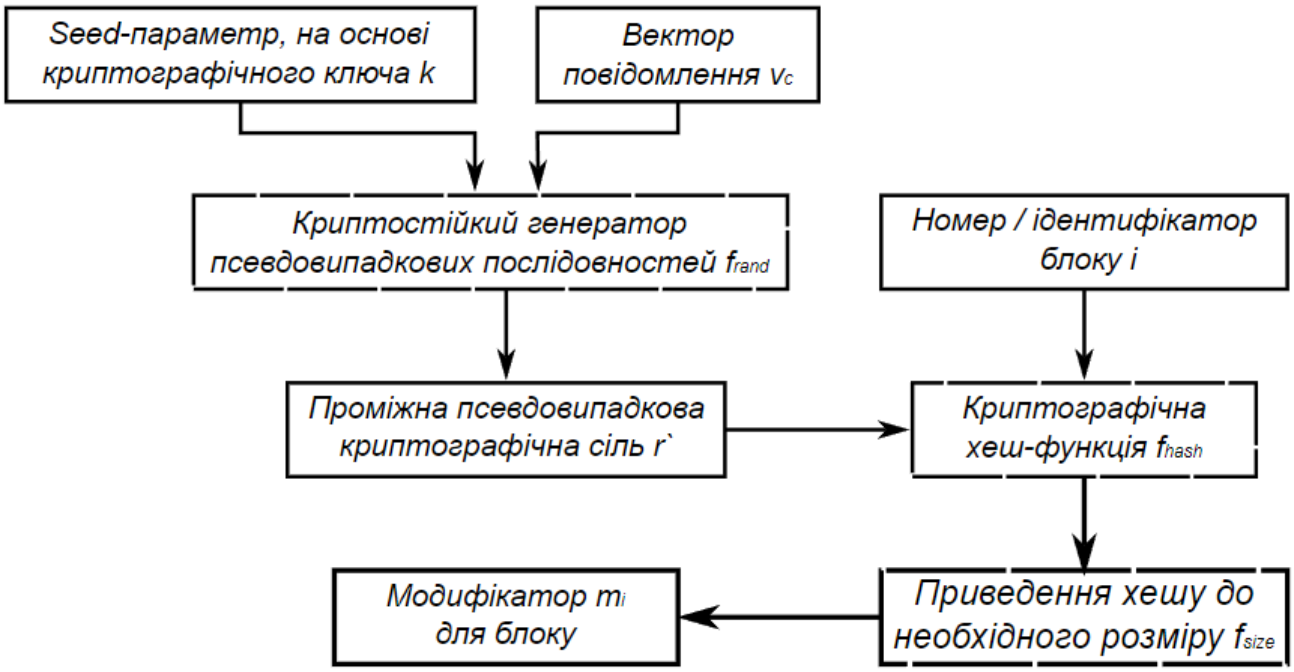


Рисунок 2.5 – Модель процесу розгортання криптографічного ключа

Вихідним результатом процесу розгортання ключа, згідно даної моделі буде модифікатор m_i, для блоку даних i, повідомлення c, з вектором v_c зашифрованого криптографічним ключем, параметром якого є сід k. Максимальний розмір модифікатора обмежений лише максимальним розміром зліпка хеш-функції, що

використовується, як правило це 128, 256, або 512 біт для найбільш поширених безпечних криптографічних хеш-функцій.

Генерацію модифікатора m_i можна записати математично наступним чином:

$$m_i = f_{size}\big(f_{hash}(r`, i)\big), \;\; i \in \mathbb{N} \tag{2.9}$$

Така схема роботи є відносно простою, з точки зору затрат електронно-обчислювальної потужності, має можливості для оптимізації, є криптографічно стійкою та відповідає визначеним вимогам.

Окремо варто відзначити вектор v_c, що передаватиметься публічно з повідомленням. Це повинно бути випадкове значення сталої довжини. Вектори не є секретними, проте вони не повинні повторюватись, оскільки, фактично, повідомлення з однаковими векторами, що зашифровані одним криптографічним ключем, матимуть однакову послідовність модифікаторів, що може створити криптографічну вразливість. Криптоаналік, навіть не знаючи самої послідовності, зможе згрупувати блоки з різних повідомлень з однаковими векторами та криптографічними ключами, що знаходяться на однаковій позиції i, та зможе однозначно стверджувати що блоки даних всередині групи зашифровані одним криптоперетворенням.

Оскільки вектори мають бути різними для кожного повідомлення – логічно припустити, що вони можуть утворюватись на основі дати та часу, перетворених за деяким алгоритмом, адже в такому випадку вони ніколи не будуть повторюватись.

Функціонування процесу розгортання криптографічного ключа не має сенсу саме по собі, у відриві від інших криптографічних процесів. В абсолютній більшості випадків воно буде виступати в ролі підпроцесу в рамках загального процесу шифрування, з використанням нелінійності.

В даному форматі процес розгортання криптографічного ключа розпочинатиметься перед початком шифрування блоків даних, на основі параметрів загального криптографічного ключа та вектору повідомлення. В ході шифрування кожного блоку даних, процес розгортання повертатиме модифікатор для поточного блоку (рис. 2.6).

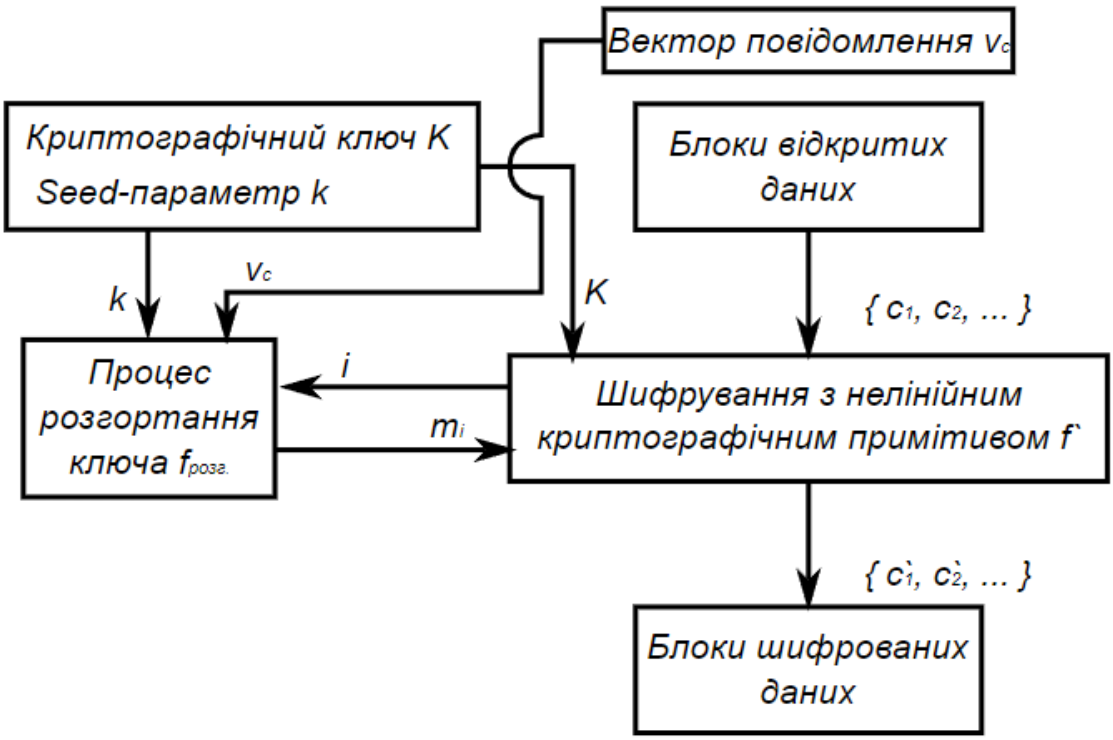


Рисунок 2.6 – Модель процесу нелінійного шифрування з процесом розгортання криптографічного ключа

В процесі розшифрування схема буде абсолютно ідентичною, за винятком того що ідентифікатори блоків подаватимуться у зворотному порядку, відносно шифрування, а процес генерації випадкових послідовностей у складі процесу розгортання повинен при ініціації відразу виконати стільки ітерацій, скільки він виконував при шифруванні цього повідомлення.

Дана кількість залежить від того чи генерується криптографічна сіль r` нова для кожного блоку, чи рідше. Для виконання розшифрування стек згенерованих криптографічних солей має бути збережений, та вираховуватися в зворотному порядку, у ході розшифрування блоків шифротексту.

Побудовані моделі у повній мірі відповідають принципу Керкгоффса, оскільки таємним в них є виключно криптографічний ключ, а всі інші складові процесів шифрування та розгортання криптографічного ключа є публічними та детально описаними.

Таким чином, ми побудували математичну модель процесу розгортання криптографічного ключа у криптосистемах, на основі нелінійних криптографічних примітивів, на основі відомих рішень, вдосконаливши їх. Побудована модель у повній мірі відповідає переліку вимог, що був сформований у попередньому розділі.

Використовуючи побудовану математичну модель методу розгортання криптографічного ключа, ми готові приступити до початку створення методу розгортання криптографічного ключа, на її основі.

2.4 Висновок

У цьому розділі ми склали загальний математичний опис криптосистем на основі нелінійних криптографічних примітивів, розглянули схему їх роботи та ключові моменти їх функціонування, зокрема найбільш оптимальні способи утворення та передачі модифікаторів.

Також ми склали перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах за трьома категоріями, серед яких: загальні вимоги щодо архітектури процесу, криптографічні вимоги та вимоги швидкодії та оптимізації, а також обґрунтували їх важливість та вкрай негативні наслідки різного характеру у випадку порушення будь-якої із них.

На основі схеми роботи криптосистем із нелінійними криптографічними примітивами та з урахуванням сформованого переліку вимог до процесу розгортання ключа та генерації модифікаторів у нелінійних криптосистемах, було створено математичну модель процесу розгортання ключа у нелінійних

криптосистемах, шляхом вдосконалення математичних моделей існуючих аналогів.

Побудована математична модель стане основою у створенні методу розгортання криптографічного ключа для використання у нелінійних криптосистемах.

Таким чином, ми виконали перше, друге та третє завдання, в рамках поставленої задачі, та готові приступати до створення методу розгортання криптографічного ключа.

3 МЕТОД РОЗГОРТАННЯ КРИПТОГРАФІЧНОГО КЛЮЧА ДЛЯ ВИКОРИСТАННЯ У НЕЛІНІЙНИХ КРИПТОСИСТЕМАХ

3.1 Синтез методів розгортання ключів у потокових криптосистемах та методів генерації псевдовипадкових послідовностей

Для створення методу розгортання криптографічного ключа для використання у криптосистемах, на основі нелінійних криптографічних примітивів, відповідно до побудованої математичної моделі процесу розгортання криптографічного ключа, проведемо аналіз та синтез існуючих рішень, методів та алгоритмів розгортання ключів та інших об’єктів, що можуть бути цікавими, в рамках даного дослідження.

Спочатку проаналізуємо криптографічно стійкі алгоритми генерації випадкових чисел, оскільки даний тип алгоритмів є основою для абсолютної більшості методів розгортання криптографічних ключів, особливо у складі генератора гамми потокових симетричних систем шифрування.

Одним з найпоширеніших є генератор псевдовипадкових чисел «Blum Blum Shub» [51]. Він ґрунтується на складності факторизації великих чисел і є криптографічно стійким за умови вибору достатньо великого стартового значення. Даний метод описується формулою, де p, q – великі прості числа:

$$x_{n+1} = x_n^2 \; mod \; M, \tag{3.1}$$

$$M = pq, \tag{3.2}$$

Числа p та q повинні бути конгруентні з 3 по модулю 4. Особливістю цього алгоритму є те що n-на ітерація випадкового числа, може бути обчислена напряму, без необхідності виконання усіх ітерацій.

Недоліками цього методу є відносно низька швидкість.

Ще один популярний генератор – «Fortuna». Він використовує декілька джерел ентропії, зокрема апаратні шумові генератори. Криптостійкість досягається за рахунок регулярного перезатирання внутрішнього стану.

Даний алгоритм включає в себе генератор псевдовипадкових чисел, що працює на основі довільного алгоритму шифрування, як наприклад AES, Twofish тощо.

Типовим криптографічно стійким генератором чисел є використання звичайного блокового алгоритму шифрування в режимі лічильника. Послідовність натуральних чисел шифрується деяким таємним ключем, а зашифровані дані і виступають у якості випадкових послідовностей.

Цей метод генерації псевдовипадкових послідовностей є простим та швидким, при цьому він має достатню криптографічну стійкість, за умови секретності ключа шифрування.

Прикладом такого генератора є SP800-90 AES CTR DRBG [52]. Даний алгоритм має два основних параметри: ключ K та вектор V (рис. 3.1).

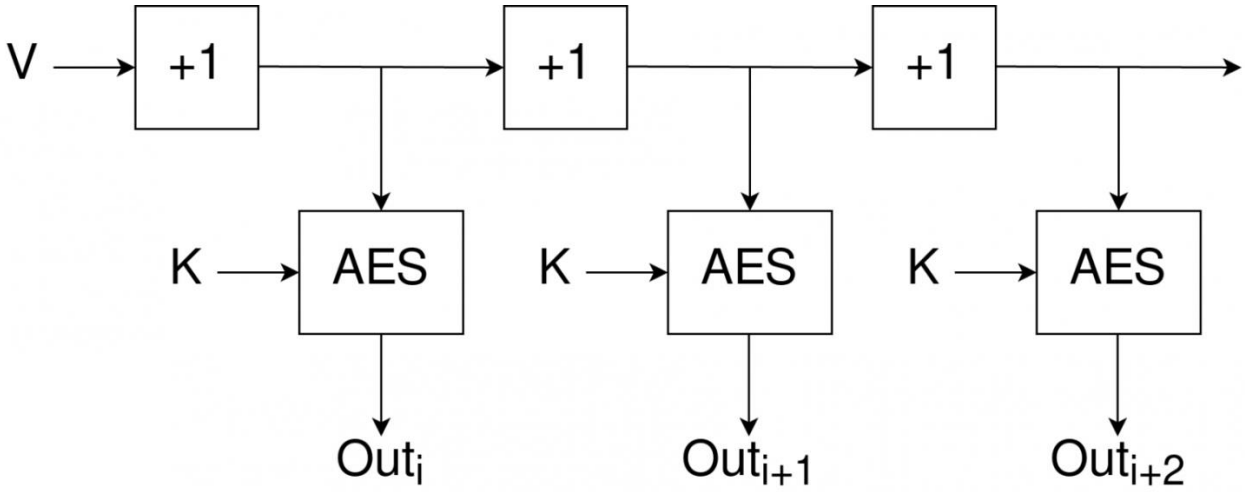


Рисунок 3.1 – Алгоритм генерації випадкових чисел AES CTR

Метод генерації складається з двох кроків:

1. Створення ключа K, та початкового вектору V, що є лічильником.

2. Генерація псевдовипадкових чисел, шляхом шифрування вектору з використанням 256-бітного AES.

Дана категорія алгоритмів добре підходить для методу розгортання криптографічного ключа у нелінійних криптосистемах.

Іншою технологією, що може бути цікавою в контексті даного дослідження є алгоритми генерації одноразових паролів. Прикладом цього є відомий додаток Google Authenticator, що базується на алгоритмі TOTP [53-54].

Алгоритм TTOP – це алгоритм генерації одноразових паролів, на основі часу. Даний алгоритм базується на іншому подібному алгоритмі – HOTP. Різниця між ними в тому, що HOTP генерує одноразові паролі на основі деякого таємного сіда, а TTOP, окрім нього використовується зліпок часу, як правило, з проміжками у 30 секунд (рис. 3.2).

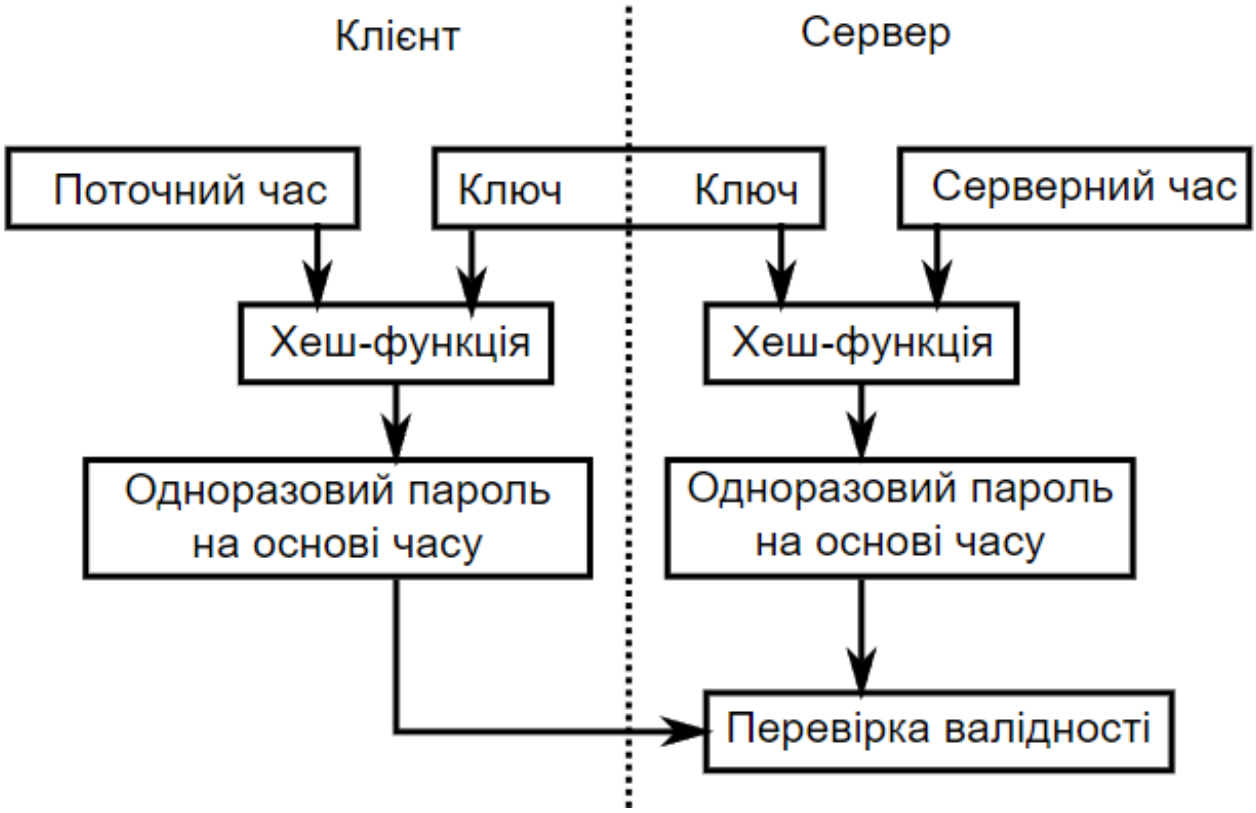


Рисунок 3.2 – Схема роботи алгоритму TTOP

HOTP використовує у своїй структурі механізм валідації HMAC, на основі хеш-функції SHA-1. Вхідними даними алгоритму є секретний ключ K, лічильник C та функція хешування HMAC (SHA-1).

Обчислюється значення хеш-функції від параметрів K та C. Далі з результату беруться 4 молодші байти. Отримані байти перетворюються у ціле число. Отримане число береться за модулем 10^6. Результатом цих дій є одноразовий пароль для лічильника C.

Такий алгоритм дозволяє генерувати різні паролі для різних значень лічильника, не даючи можливості передбачити наступні паролі на основі попередніх.

Криптостійкість алгоритму HOTP забезпечується за рахунок стійкості функції хешування та секретного ключа K.

Елементи даного алгоритму також можуть бути використані в методі розгортання криптографічного ключа для криптосистем, на основі нелінійних примітивів.

Для створення власного методу ми можемо синтезувати два описаних підходи до генерації значень. Візьмемо елементи генератора випадкових чисел типу AES CTR DRBG та поєднаємо їх з підходами, що використовуються в алгоритмі HOTP.

В методі розгортання ключа ми використаємо генератор випадкових чисел, результат якого буде проміжним значенням, та використовуватиметься як сіль для наступного етапу.

Другий етап включатиме в себе змішування псевдовипадкової солі та ідентифікатора блоку, що будуть параметрами хеш-функції. Результат хешування братиметься за деяким визначеним модулем, по аналогії з алгоритмом HOTP.

Такий підхід дозволить використовувати переваги обидвох алгоритмів, а також створить простір для оптимізації реалізацій даного методу.

Отже, за результатами дослідження методів розгортання ключа та іншими суміжними алгоритмами ми розглянули можливість синтезу деяких із цих підходів, наприклад поєднання AES CTR DRBG та HOTP. Найбільш підходящі для нас рішення, що використовуються в описаних методах будуть використані при створенні власного методу розгортання ключа.

3.2 Створення методу розгортання ключа для криптосистем на основі нелінійних криптографічних примітивів

Приступимо до створення власного методу розгортання ключа, на основі побудованої математичної моделі, а також з урахуванням моделі та конструктивних особливостей криптосистем на основі нелінійних криптографічних примітивів.

В процесі створення даного методу обов’язково слід враховувати перелік вимог, описаних у попередньому розділі.

Створення методу розгортання криптографічного ключа здійснюватимемо шляхом поступового приведення елементів математичної моделі до конкретних алгоритмів, або їх варіацій, із схемою їх комплексного застосування, конкретизації параметрів та діапазонів їх величин.

Розпочнемо з генерації криптографічного ключа для нелінійної криптосистеми.

В цілому ніщо не перешкоджає використовувати сам криптографічний ключ як сід генератора випадкових послідовностей. Проте в попередньому, серед вимог, було зазначено що задля недопущення створення будь-яких вразливостей в базовій криптосистемі рекомендується сепарувати усі елементи даного методу з метою усунення будь-якого їх впливу. Керуючись цим, розділимо ключ криптосистеми та сід-ключ для ініціації генератора випадкових послідовностей.

Що стосується розміру сід-ключа, то для високого рівня захисту рекомендується використовувати 384-бітний сід (рис. 3.3.).

```
{
    "key": {
        "base_key": "66455748595902A9F87A598E68401210BB9010146A4465849FD0464521412254",
        "seed_key": "A0EDB7F0E29D1692524B3E565C90C25E4CAB6CCB75269B16D626DEE35CF572B048F2634E0042F27D81A18B0621098DEE"
    }
}
```


Рисунок 3.3 – Приклад структури ключа методу

З одного боку це збільшує розмір криптографічного ключа, проте з іншого – забезпечує високий рівень стійкості генератора псевдовипадкових послідовностей. Проте навіть так розширення ключа на 48 байт, не буде помітним для сучасної комп’ютерної техніки.

У випадку якщо розмір ключа є принциповим, а його збільшення недопустиме – можливо використовувати ключ як сід, порушуючи сепарацію, та знижуючи стійкість генератора. Проте безпечність такого використання ключа потребує додаткових досліджень.

Генерацію значень параметрів ключа ми не розглядаємо в рамках даного методу, оскільки вважаємо їх вхідними параметрами, згенерованими заздалегідь.

Далі розберемось з вектором повідомлення: це випадкове значення яке задає різницю при генерації модифікаторів для конкретного повідомлення. Як було сказано раніше, вектори не повинні повторюватись, оскільки два, чи більше, повідомлень, що зашифровані за однаковими векторами, матимуть однакову послідовність модифікаторів в процесі шифрування.

Для генерації вектору повідомлення створимо алгоритм, на основі поточного часу в момент шифрування, щоб запобігти повторенням.

Оскільки відкрита інформація про точний час шифрування повідомлення може створити вразливості та додаткові можливості для статистичного дослідження, в складі алгоритму генерації вектору повідомлення використаємо хеш-функцію SHA-2. Окрім часу, використовуватимемо криптографічну сіль, з довільного криптостійкого генератора чисел. Розмір солі встановимо як 256-біт, розмір вектору повідомлення – також 256 біт.

АГВП складатиметься з таких етапів:

– генерація 256-бітної криптографічної солі;

– поєднання криптографічної солі з датою та часом у форматі: «20231201-1845-432»;

– створення хешу криптографічної солі та дати алгоритмом SHA-256.

За результатами цього отримаємо 256-бітний вектор повідомлення, який передаватимемо публічно разом із зашифрованими даними (рис.3.4).

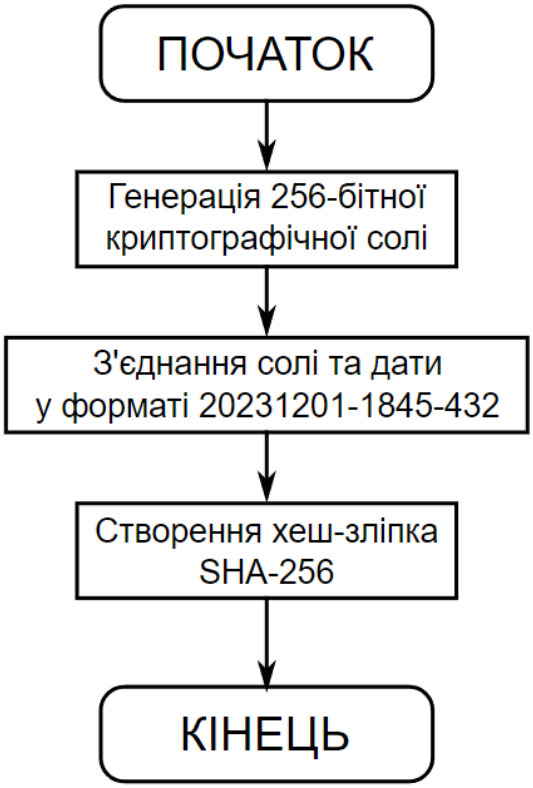


Рисунок 3.4 – Алгоритм генерації вектору повідомлення

Далі перейдемо до криптостійкого генератора псевдовипадкових послідовностей. Це найбільш складний елемент із наведених.

Генератор псевдовипадкових послідовностей зробимо, використовуючи елементи алгоритму AES CTR DRBG, описаного в попередньому пункті. Початковими параметрами алгоритму будуть 256-бітний вектор повідомлення та 384-бітний сід-параметРРр криптографічного ключа. Алгоритм включатиме:

1. поділ сіда на дві частини по 128 та 256 біт: s_1, s_2;

2. об’єднання вектору повідомлення та другої частини s_2 за допомогою операції XOR;

3. шифрування s_1 криптосистемою AES-256, за допомогою об’єднаного ключа s_2;

4. повторення попереднього етапу ще двічі, з додаванням одиниці до блоку, який шифрується;

5. за необхідності згенерувати нову псевдовипадкову послідовність, об’єднуємо результати трьох блоків шифрування у 384-бітну послідовність та повторюємо дії, за винятком етапу 2.

Таким чином ми зможемо згенерувати довільну кількість псевдовипадкових значень, на основі криптографічного ключа та вектору повідомлення (рис.3.5).

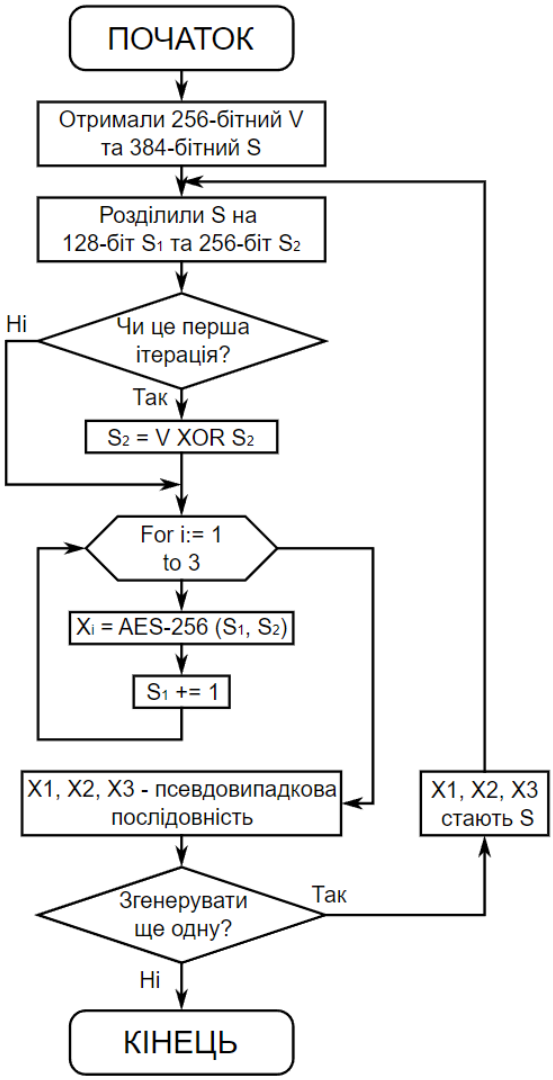


Рисунок 3.5 – Алгоритм генерації псевдовипадкової послідовності

На виході даний алгоритм повертає 384-бітну криптографічну сіль, яка використовуватиметься в подальших кроках методу.

Наведені розміри блоків даних можуть корегуватися при конкретних імплементаціях даного алгоритму, проте не рекомендується їх зменшувати,

оскільки це може призвести до зниження криптографічної стійкості алгоритму генерації.

Даний алгоритм генерації псевдовипадкової послідовності виконуватиметься 1 раз для 100 блоків даних, з метою оптимізації та підвищення швидкодії його імплементацій.

Наступним кроком йде хеш-функція, для формування псевдовипадкової послідовності для конкретного блоку даних. Для цього ми будемо використовувати звичайну функцію SHA-256, оскільки вона є криптографічно стійкою та повністю відповідає вимогам даного елементу, згідно математичної моделі.

На вхід функція отримуватиме два параметри:

– 384-бітну криптографічну сіль;

– ідентифікатор поточного блоку даних.

Криптографічна сіль є результатом роботи попереднього алгоритму генерації псевдовипадкових послідовностей.

Ідентифікатор блоку братиметься як простий порядковий номер поточного блоку, починаючи з нульового.

Останнім елементом математичної моделі залишилось приведення хешу до необхідного розміру. Є багато різних способів якими можна було б його реалізувати. Одним з найпростіших способів є приведення хеш-зліпку до десяткового числа та взяття його за модулем n, де n – це загальна кількість можливих модифікаторів.

Для оптимізації даного алгоритму, за умови що в нелінійній криптосистемі існує не велика кількість криптографічних перетворень та відповідно модифікатор – це не велике число, можемо приводити до числа не весь хеш-зліпок, а лише певну кількість його символів, наприклад останніх вісім шістнадцяткових знаків, тобто 4 байти.

За результатами цього кроку ми отримаємо модифікатор, готовий до використання у нелінійній криптосистемі.

Важливою вимогою до модифікатора є рівна ймовірність його випадання, з множини усіх можливих модифікаторів. Перевіримо це експериментально: запустимо програмний прототип описаних алгоритмів. Для тестування кількість можливих модифікаторів в алгоритмі приведення хешу до необхідного розміру задана як 10. Кількість тестових ітерацій було задано як одну тисячу. Результати виконання тестової генерації занесено до таблиці (табл. 3.1).

Таблиця 3.1 – Результати тестової генерації модифікаторів

№	Модифікатор	Кількість випадінь	Відсоткове відношення
1	«1»	98	9,8%
2	«2»	100	10%
3	«3»	100	10%
4	«4»	99	9,9%
5	«5»	101	10,1%
6	«6»	99	9,9%
7	«7»	100	10%
8	«8»	102	10,2%
9	«9»	100	10%
10	«10»	101	10,1%

За результатами розподілу видно що ймовірності випадіння модифікаторів практично рівні, в межах статистичної похибки, а отже ми можемо говорити про успішність експерименту.

Зашифроване повідомлення передається разом із вектором по незахищеному каналу зв’язку, оскільки, як говорилось раніше, згідно принципу Керкгоффса, таємним у даному методі залишається виключно криптографічний ключ, коли усі інші параметри є загальновідомими, в тому числі для криптоаналітика, який також досконало знає принцип роботи даного методу (рис.3.6).

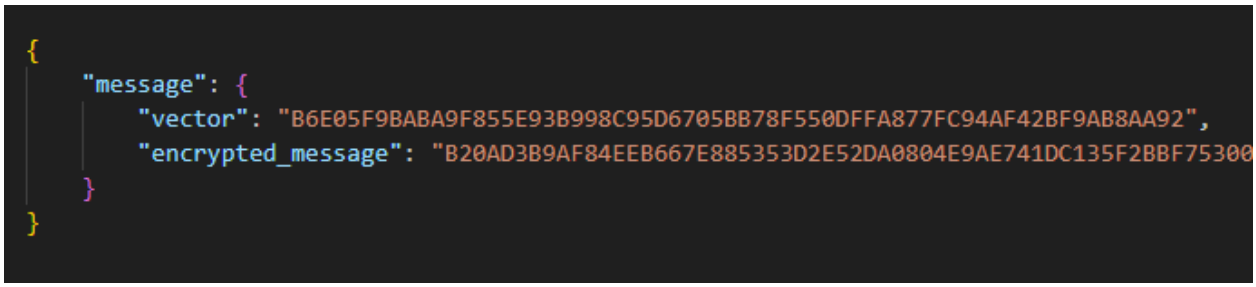


Рисунок 3.6 – Приклад структури зашифрованого повідомлення

На зразках продемонстровано структуру повідомлень у форматі JSON, оскільки вона є більш наглядною.

На практиці ж як криптографічний ключ, так і зашифроване повідомлення можуть передаватися у бінарному вигляді, без деталізації, можливо з використанням програмних засобів архівації та стискання файлів, з метою економії ресурсів.

Таким чином, запропонований метод включає в себе 6 кроків, серед яких:

1. Генерація криптографічного ключа з двома параметрами: стандартного ключа криптосистеми та 384-бітного сід-ключа для ініціації генератора псевдовипадкових послідовностей.

2. Генерація 256-бітного вектору повідомлення, за допомогою АГВП.

3. Ініціація криптостійкого генератора псевдовипадкових повідомлень, з початковими параметрами: 384-бітним сідом та 256-бітним вектором повідомлення.

4. Виконання шифрування блоків повідомлення, із генерацією модифікатора для кожного із них за допомогою хеш-функції SHA-256 з параметрами у вигляді ідентифікатора блоку та криптографічної солі, створеної генератором псевдовипадкових послідовностей.

5. Передача зашифрованого повідомлення, разом із вектором.

6. Виконання розшифрування у зворотній послідовності.

Структура методу відповідає математичній моделі та переліку вимог, поставлених перед ним (рис. 3.7).

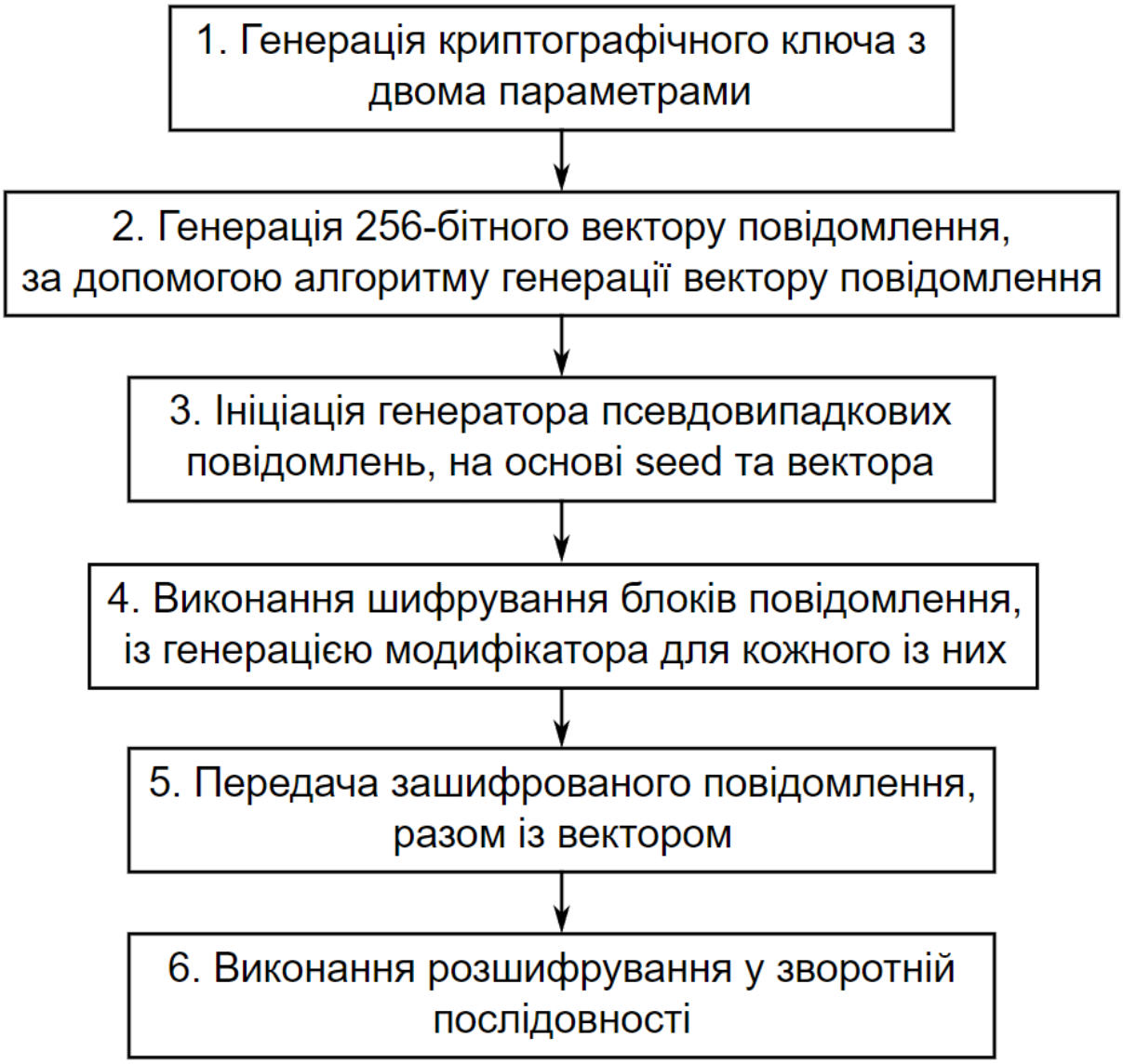


Рисунок 3.7 – Метод розгортання криптографічного ключа для використання у нелінійних криптосистемах

Таким чином, ми створили метод розгортання криптографічного ключа для використання у нелінійних криптосистемах у строгій відповідності до математичної моделі процесу розгортання ключа та із дотриманням встановленого переліку вимог. В процесі розробки ми використовували елементи існуючих рішень, проаналізувавши та синтезувавши їх.

Створений метод відповідає першочерговому завданню та меті даного дослідження.

3.3 Застосування створеного методу у криптосистемах та криптографічних протоколах

Створений метод є дуже гнучкий у застосуванні, він може бути використаний у будь-якій криптосистемі з використанням нелінійних криптографічних примітивів, де необхідне розгортання ключа з метою отримання модифікаторів.

Серед основних можливих напрямків застосування створеного методу можна виділити:

– модифікація існуючих перевірених криптографічних систем, з впровадженням в них нелінійних криптографічних примітивів, разом з методом розгортання криптографічного ключа;

– створення нових криптосистем на основі нелінійних криптографічних примітивів, разом з методом розгортання криптографічного ключа;

– створення нелінійних криптографічних протоколів, з використанням нелінійних криптографічних алгоритмів, без жодних модифікацій існуючих криптосистем.

Відповідно до перелічених вище напрямків, надалі, в практичній частині, буде створено експериментальні прототипи програмного забезпечення, за кожним із них.

Розберемо детальніше кожен із цих способів застосування.

Модифікація існуючих криптосистем, що є перевіреними та сертифікованими, дає можливість дослідити та практично застосувати нелінійність на основі надійних криптосистем. Даний напрямок передбачає аналіз деякої існуючої блокової симетричної системи шифрування, такої як DES, AES, IDEA тощо, та впровадження до її структури нелінійного криптографічного примітиву, разом із створеним методом розгортання криптографічного ключа для нього.

Перевагами даного застосування є:

– використання завідомо надійної перевіреної системи шифрування;

– наявність в публічній площині великої кількості різнобічних досліджень, що розкривають зокрема й можливі вразливості криптосистеми, на які варто звернути увагу;

– наявність спільноти зацікавлених дослідників.

Серед недоліків даного застосування:

– можливість своєю модифікацією створити явні або приховані вразливості, що можуть проявитися лише через значний проміжок часу;

– порушення сертифікованого стандарту шифрування;

– захищеність патентами та авторськими правами деяких криптосистем.

Створення нової криптосистеми на основі нелінійних криптографічних примітивів, разом з методом розгортання криптографічного ключа є цікавим варіантом, який дозволяє в повній мірі реалізувати потенціал нелінійної криптографії, розробивши криптографічну систему на них, можливо навіть з більш розгалуженою системою шифрування. Проте даний спосіб також має багато недоліків, серед яких:

– складність проектування нової криптосистеми, з необхідністю враховувати усі фактори;

– необхідність багатостороннього тестування, в тому числі третіми сторонами;

– необхідність сертифікації, перед будь-яким реальним практичним застосуванням;

– відсутність поінформованої спільноти.

Третім можливим напрямком застосування є створення нелінійного криптографічного протоколу, з використанням нелінійних криптографічних алгоритмів, без жодних модифікацій існуючих криптосистем.

Даний варіант є цікавим, оскільки фактично об’єднує в собі попередніх два: з одного боку дає можливість реалізувати своє бачення логіки нелінійного шифрування, з використанням запропонованого методу, а з іншого – передбачає

використання перевірених криптографічних систем та усіх згаданих напрацювань по них.

Серед переваг даного застосування є:

– використання готових перевірених криптосистем;

– відсутність будь-яких модифікацій всередині них, а отже вони залишаються відповідними стандарту;

– широка поінформованість щодо існуючих систем та накопичена наукова база;

– можливість побудови власної логіки нелінійної взаємодії криптосистем з мінімізованими ризиками порушення безпеки.

Серед недоліків даного застосування є:

– залишається необхідність у незалежних дослідженнях та сертифікації перед будь якими спробами реального використання;

– залишається можливість створення нових вразливостей, хоча й вона зменшена.

Створений метод розгортання ключа може бути однаково легко застосований до будь-якого із наведених напрямків застосування, оскільки він не залежить від конкретної імплементації нелінійних алгоритмів. Дотримання вимоги сепарації дає можливість уникнути будь-яких проблем чи вразливостей, пов’язаних саме з методом розгортання криптографічного ключа для криптосистем.

Таким чином, ми розглянули можливі напрямки застосування нелінійних криптографічних систем разом із створеним методом розгортання криптографічного ключа. Кожен із них має свої переваги та недоліки. Одним із найцікавіших серед них є створення нелінійного криптографічного протоколу, з використанням нелінійних криптографічних алгоритмів, без жодних модифікацій існуючих криптосистем, оскільки він є найбільш безпечним із перелічених.

За усіма наведеними напрямками надалі буде розроблено експериментальні прототипи програмного забезпечення з метою практичного тестування створеного

методу розгортання ключа в різних умовах та дослідження характеристик прототипів.

3.4 Висновок

У даному розділі ми розглянули можливість синтезу деяких із підходів, що використовуються у методах розгортання ключів у потокових криптосистемах та методах генерації псевдовипадкових послідовностей, зокрема на прикладі AES CTR DRBG та HOTP. Найбільш підходящі для нас рішення, що використовувались в описаних методах були використані при створенні власного методу розгортання ключа.

На основі цього, а також з використанням математичних моделей, побудованих у попередніх розділах, ми створили метод розгортання ключа для криптосистем на основі нелінійних криптографічних примітивів та розглянули напрямки його можливого використання у складі криптографічних систем та протоколів.

4 ПРИКЛАДНЕ ЗАСТОСУВАННЯ МЕТОДУ РОЗГОРТАННЯ КРИПТОГРАФІЧНОГО КЛЮЧА

4.1 Модифікація алгоритму DES на основі створеного методу

Перед початком роботи над програмними реалізаціями прототипів, викладемо інструменти, необхідні для успішного виконання завдання, а також опишемо загальні підходи, що застосовуватимуться.

Оскільки програмне забезпечення розроблятиметься під ОС Windows 10, то найбільш доречним та зручним середовищем розробки є Microsoft Visual Studio.

Це програмний комплекс, що надає можливості для створення, налагодження та експлуатації як консольних застосунків, так і програм з графічним інтерфейсом, зокрема з підтримкою Windows Forms. Також дозволяє розробляти веб-застосунки та сайти для всіх платформ ОС Microsoft Windows.

Розробка вестиметься мовою програмування C#, з використанням можливостей фреймворку .NET Framework та програмного інтерфейсу Windows Forms. Цей набір інструментів чудово підходить для успішного виконання поставлених завдань та добре підтримується у середовищі Microsoft Visual Studio.

Мова C# є об'єктно-орієнтованою. Її код компілюється у виконувані файли формату DLL, що забезпечує швидкодію та ефективне використання ресурсів.

Розробка вестиметься з дотриманням принципів об'єктно-орієнтованого програмування та загальноприйнятих стандартів. Це дозволить іншим розробникам працювати над проектом та уникнути дублювання коду.

Алгоритми реалізовуватимуться у вигляді динамічних бібліотек DLL, щоб їх можна було використовувати в інших проектах, підключивши бібліотеку та отримавши доступ до публічного API.

Крім того, у проекті застосовуватиметься система керування версіями Git. Вона дозволяє розподіляти розробку на гілки, створювати контрольні точки (коміти), стежити за загальним прогресом та повертатися до попередніх версій.

Також є можливість синхронізуватися з віддаленим репозитарієм та працювати над проектом з різних локацій.

Адміністрування версій здійснюватиметься за допомогою веб-платформи GitHub. Цей сервіс надає можливість безкоштовно створювати публічні та приватні репозитарії, керувати правами доступу та іншими налаштуваннями. Використання GitHub дасть змогу у майбутньому опублікувати відкритий код проекту.

Отже, спробуємо модифікувати існуючий симетричний блоковий шифр за допомогою нелінійних криптографічних примітивів та з використанням створеного методу розгортання криптографічного ключа.

Як приклад оберемо відомий алгоритм DES, керуючись кількома міркуваннями:

– алгоритм добре вивчений;

– алгоритм має відносно просту схему шифрування на основі класичної мережі Фейстеля, яку нескладно модифікувати;

– алгоритм є застарілим та вразливим до деяких сучасних атак, модифікація дозволить усунути ці недоліки.

Здійснимо модифікацію шляхом заміни статичної лінійної раундової функції на динамічну нелінійну з використанням описаного раніше підходу.

Для цього розробимо декілька аналогів криптофункції, що зберігатимуть її сигнатуру. Всі створені аналоги матимуть ідентичні сигнатури.

Існує чимало способів зміни функції без зміни її структури. Наприклад, можемо розглянути лінійне криптоперетворення накшталт початкового розширення блоку, що є у складі даної функції.

У класичній функції алгоритму DES воно завжди виконується однаковим чином: 32-бітний блок здійснює розширення до 48-бітного, шляхом поділу вхідного блоку на малі групи по 4 біта та «запозиченням» двох бітів з сусідніх груп.

Не змінюючи дану логіку роботи і принципову роль даного криптографічного перетворення в функції, ми маємо можливість змінити позиції, на які вставляються «запозичені» біти.

Зробити це можна кількома способами. Така незначна зміна не спричинить глобальну зміну логіки функціонування криптосистеми, але змінить вихідний шифротекст (рис. 4.1).

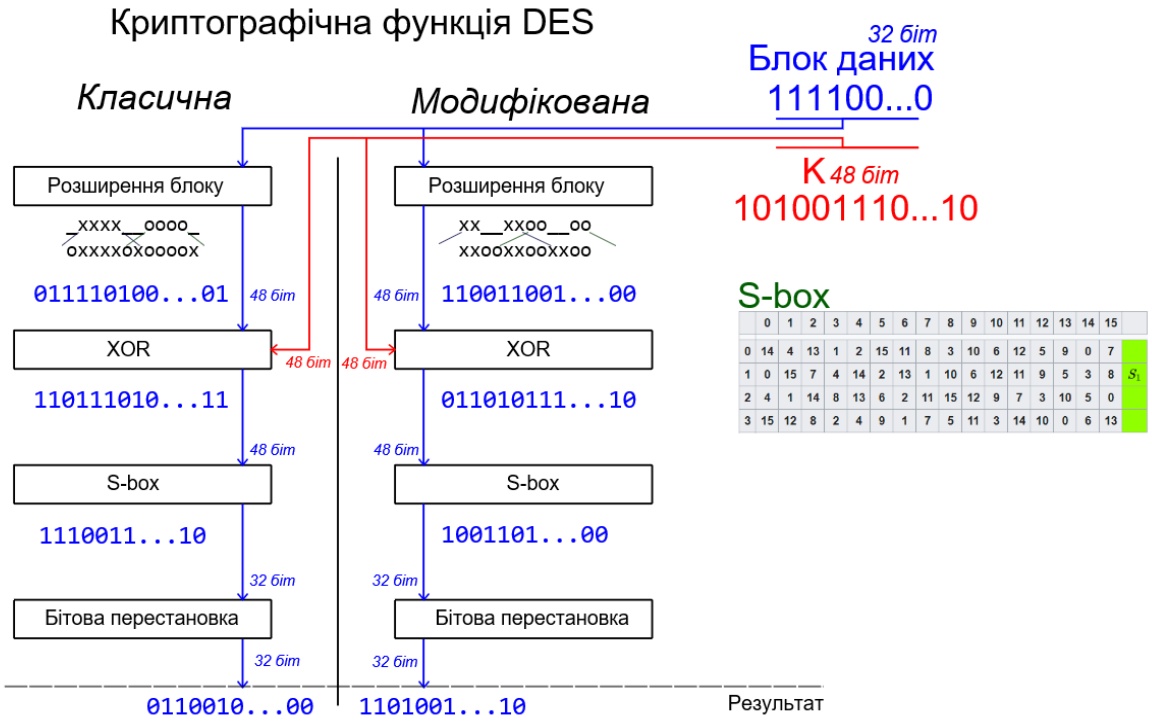


Рисунок 4.1 – Класична та варіант модифікованої криптографічної функції алгоритму DES

Отже, ми модифікували криптографічну функцію алгоритму DES, неістотно змінивши лише один лінійний криптопримітив у її складі.

Звичайно, існує безліч інших способів подібної модифікації складових цієї функції. В будь-якому разі, вносячи зміни до стандартних елементів, треба уважно стежити, аби це не спричинило появу вразливостей у криптосистемі.

Використаємо створений метод розгортання криптографічного ключа для генерації модифікаторів в процесі шифрування.

Створимо класс DesRound та зробимо структуру методів (рис. 4.2).

```
public class DesRound
{
    private const int BlockSize = 64;

    private readonly UInt32[] _keySchedule;

    public DesRound(UInt32[] keySchedule)
    {
        _keySchedule = keySchedule;
    }

    public UInt64 Encrypt(UInt64 block, int round)
    {
        UInt32 left = (UInt32)(block >> 32);
        UInt32 right = (UInt32)block;

        UInt32 f = FeistelFunction(right, round);

        UInt32 newRight = left ^ f;
        UInt32 newLeft = right;

        UInt64 output = ((UInt64)newLeft << 32) | newRight;
        return output;
    }

    private UInt32 FeistelFunction(UInt32 block, int round)
    {
        UInt32 expanded = Expand(block);
        UInt32 keyed = Xor(expanded, _keySchedule[round]);
        UInt32 substituted = Substitute(keyed);
```

Рисунок 4.2 – Об’єктно-орієнтована структура класу DesRound

Реалізуємо алгоритми, відповідно блок-схем, наведених у попередньому розділі. Відповідно до цього реалізуємо в динамічній бібліотеці наступний перелік класів:

– ModifierGenerator;

– MessageVector;

– ExtendedKey.

Всередині класів реалізуємо відповідну логіку, зокрема клас «ModifierGenerator» відповідатиме за загальну логіку роботи алгоритмів методу. Він міститиме методи:

– getModifierForBlock;

– getNextRandomSalt;

– subtractModifierFromHash;

– makeHash.

Класи «MessageVector» та «ExtendedKey» будуть транспортними об’єктами, які мітитимуть в собі необхідні інкапсульовані параметри.

Імплементуємо усі не обхідні методи, відповідно до їх опису та реалізуємо простий візуальний інтерфейс (рис.4.3).

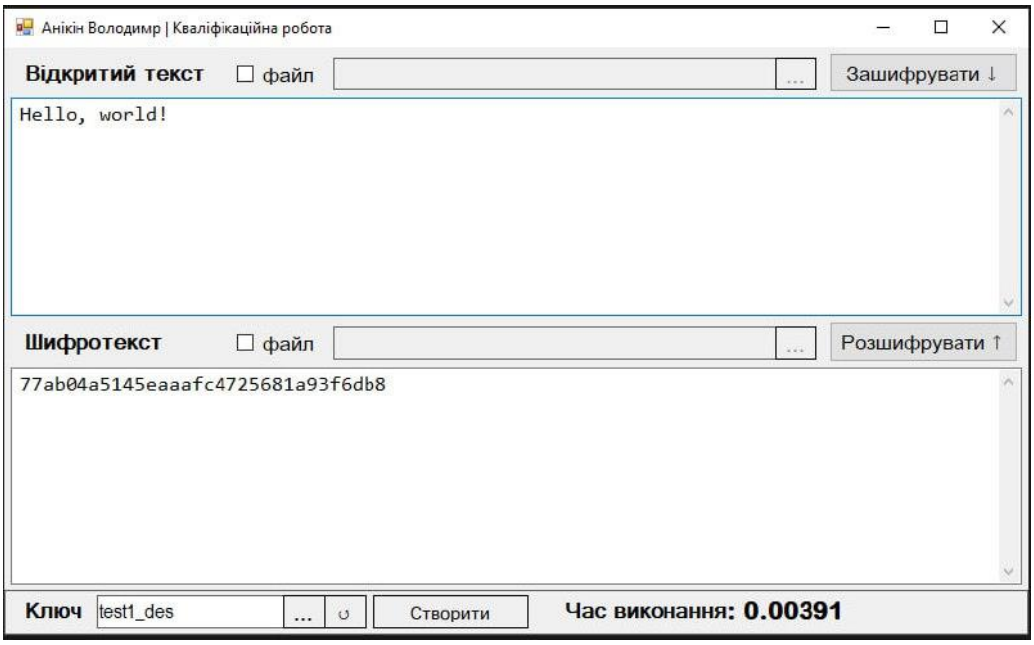


Рисунок 4.3 – Реалізація нелінійної криптосистеми, на основі DES, з методом розгортання криптографічного ключа

За результатами тестування програмного прототипу різниця швидкодії між звичайним алгоритмом DES та модифікованим складає не більше 15-20 відсотків.

Характерною позитивною рисою є те, що взагалі відсутнє систематичне розширення шифрованого тексту, відносно відкритого.

Таким чином, ми створили програмний прототип нелінійної криптосистеми на основі алгоритму DES із впровадженням створеного методу розгортання криптографічного ключа.

4.2 Модифікація алгоритму AES на основі створеного методу

Для створення програмного прототипу модифікованої нелінійної криптосистеми на базі шифру AES, виконаємо ті ж кроки, що й для DES.

Спочатку визначимось на якому саме етапі впроваджуватимемо нелінійний криптографічний примітив.

Для запобігання пошкодження криптосистеми, або створенню в ній якихось вразливостей, ми не будемо чіпати її стандартні методи. Додамо власний нелінійний елемент туди, де він з найбільшою ймовірністю не викличе жодних проблем – до відкритого тексту.

Ми введемо елементарне нелінійне розгалуження: додамо циклічний зсув бітів для блоків відкритих даних з такими варіантами:

– для модифікатора «0» – зсув на 2 біта ліворуч;

– для модифікатора «1» – зсув на 4 біта ліворуч;

– для модифікатора «2» – зсув на 2 біта праворуч;

– для модифікатора «3» – зсув на 4 біта праворуч.

Таким чином, ми маємо 4 варіанти нелінійного криптографічного перетворення. Реалізуємо це в програмному коді (рис. 4.4).

```
134     return state;
135 }
136
137 private void SubBytes()
138 {
139     for(int i = 0; i < 4; i++)
140     {
141         for(int j = 0; j < 4; j++)
142         {
143             byte s = state[i,j];
144             byte substitution = SBox[s];
145             state[i,j] = substitution;
146         }
147     }
148 }
149
150 private void ShiftRows()
151 {
152     byte[,] shifted = new byte[4,4];
153
154     for(int i = 0; i < 4; i++) {
155         shifted[0, i] = state[0, i];
156     }
157
158     int[] offsets = {1, 2, 3};
159
160     for(int row = 1; row < 4; row++) {
161         for(int col = 0; col < 4; col++) {
162             int index = (col + offsets[row-1]) % 4;
163             shifted[row, col] = state[row, index];
164         }
165     }
166
167     state = shifted;
168 }
```

Рисунок 4.4 – Реалізація простої нелінійної криптосистеми, на основі AES, з методом розгортання криптографічного ключа

Використання динамічної бібліотеки дає можливість використовувати ту ж структуру класів, що й для реалізації криптосистеми DES, уникаючи дублікатів

коду. Ми можемо відразу використовувати вже реалізовані методи, просто підключивши файл бібліотеки з розширенням «.dll» (рис. 4.5).

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using NonlinearKeyExtends;

namespace шифратор
{
    Ссылок: 4
    public partial class CreateKeyForm : Form
```

Рисунок 4.5 – Використання динамічної бібліотеки

Ми використаємо то й ж самий UI для, на основі елементів Windows Forms, для тестування програмного прототипу (рис. 4.6).

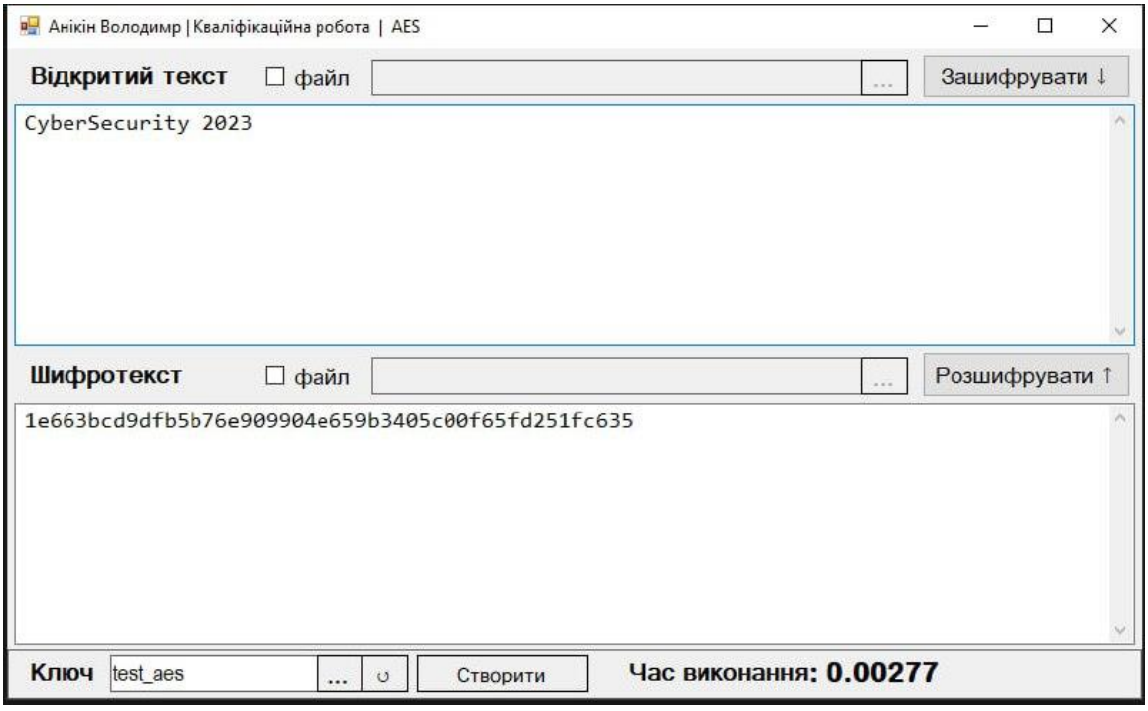


Рисунок 4.6 – Графічний інтерфейс програми-шифратора з методом розгортання ключа

Зменшення швидкодії між звичайним шифруванням криптосистемою AES та модифікованою нелінійною криптографічною системою на його ж основі, є ще менш значущим, ніж у випадку з DES: в середньому всього 8-12 відсотків.

Метод розгортання криптографічного ключа для даної системи також працює без будь-яких проблем та не викликає жодних складнощів чи незручностей в процесі експлуатації.

Таким чином, ми створили програмний прототип нелінійної криптосистеми на основі алгоритму AES із впровадженням створеного методу розгортання криптографічного ключа.

4.3 Використання методу розгортання ключа у криптографічних протоколах

Створимо криптографічний протокол, з нелінійною структурою шифрування.

Суттєвою відмінністю між цією реалізацією та двома попередніми є те, що в даному випадку ми взагалі не змінюватимемо жодних криптосистем, що є перевагою.

Єдине що ми зробимо в рамках даного підрозділу – налаштуємо правила шифрування стандартними сертифікованими криптосистемами. Використаємо для цього ті ж криптосистеми що й раніше, а саме DES та AES.

Загальна логіка даного криптографічного протоколу буде наступна:

– наявні два тільки модифікатора: «0» та «1»;

– якщо приходить модифікатор «0» – поточний блок шифрується за допомогою криптосистеми DES;

– якщо приходить модифікатор «1» – поточний блок шифрується за допомогою криптосистеми AES;

– розмір базового криптографічного ключа береться по криптосистемі з більшою довжиною ключа, в даному випадку – це AES, у 256 біт;

– ключ для криптосистеми з меншим стандартним розміром ключа береться просто шляхом вирізання відрізку необхідної довжини з масиву біт базового ключа.

Таким чином, використання нелінійності у даному криптографічному протоколі, разом із методом розгортання криптографічного ключа для неї, дозволяє створити «мультикриптосистему», деякі блоки даних шифруються одним алгоритмом, а решта – іншим.Причому криптоаналітик не зможе на основі виключно шифротексту встановити які блоки чим зашифровані, оскільки обидва алгоритми шифрування на виході дають просто набір біт, без яких-небуть знаків розрізнення.

Для успішного аналізу такого протоколу, криптоаналітик спочатку повинен взломати метод розгортання криптографічного ключа, а лише тоді намагатись досліджувати звичайне шифрування.

Також слід зауважити, що криптографічна стійкість базових криптосистем ніяким чином не порушена, тому навіть за умови компрометації методу розгортання криптографічних ключів, криптоаналітику доведеться подолати звичайну стійкість алгоритму шифрування.

Нелінійність в даному випадку виглядає як додатковий шар захисту на шляху до самих алгоритмів.

Приступимо до реалізації даного криптографічного протоколу. В даній реалізації, як і в попередній, ми використовуватимемо бібліотеку NonlinearKeyExtends, яка імплементує весь функціонал для роботи з алгоритмами створеного методу.

Імплементуємо метод шифрування з потоковим перемикачем алгоритму шифрування, на основі модифікатору (рис.4.7).

```
134     return state;
135 }
136
137 public delegate int DesEncrypt(byte[] key, byte[][] blocks);
138
139 public delegate int AesEncrypt(byte[] key, byte[][] blocks);
140
141 private byte[] Encrypt (byte[] key, byte[][] blocks, int modifier) {
142     for(int i = 0; i < blocks.Length; i++) {
143         if(modifier === 1) {
144             return AesEncrypt(key, blocks);
145         } else {
146             return DesEncrypt(key, blocks);
147         }
148     }
149 }
150
151
152 private void SubBytes()
```

Рисунок 4.7 – Імплементація криптографічного протоколу, на основі нелінійності

Виконаємо шифрування файлу за допомогою даного криптографічного протоколу.

Відкривши зашифрований файл за допомогою шістнадцяткового редактору, ми побачимо просто рівномірний набір байтів. При цьому криптографічний протокол успішно працює, оскільки деякі блоки, зокрема перший, другий, четвертий і т.д., зашифровані за допомогою криптосистеми AES, а третій, п’ятий і т.д. – криптосистемою DES (рис.4.8).

```
0000000  15 06 13 AA 77 A7 4C FE 52 93 84 F7 0E D9 A0 76   ....w.L.R......v
0000010  19 B3 AC F6 57 7A 01 2C 77 1F D8 13 75 C4 F4 EE   ....Wz.,w...u...
0000020  A9 2C 4D 74 B2 EA CC F1 06 2F 83 8D B1 65 B5 8D   .,Mt...../...e..
0000030  CB D3 12 C4 15 3E CB 5F B0 F8 68 90 A9 A8 CC 36   .....>._..h....6
0000040  7A 9B EA ED 22 62 0D 8C 7F 4C F7 3B 65 C8 4E 04   z..."b.. L.;e.N.
0000050  84 E0 0B E0 AA 9E E8 17 D0 FC 18 49 5D C4 EF 41   ...........I]..A
0000060  C0 BA 08 6F 60 A9 80 24 E5 10 41 D5 E7 F9 AC F1   ...o`..$..A.....
0000070  6C 55 EF F3 22 FA BE 46 5A 0C F0 42 D4 B1 0C 11   lU.."..FZ..B....
0000080  4F 1D 94 1E DD EB 4F 42 B0 6E EF A3 D2 84 4C 84   O.....OB.n....L.
0000090  D3 C9 28 E7 DD 9A 87 58 1F 9F CF FF EA BE 91 BA   ..(....X........
00000A0  6B A0 A7 29 F3 74 19 B9 CC 91 5B 21 0F 5A 7F 21   k..).t....[!.Z !
00000B0  20 70 DB 9B 03 38 57 8D 69 9E FC 8D D0 08 4F A7   .p...8W.i.....O.
00000C0  BD F5 35 5B 01 18 24 92 38 23 13 EA CE FF EB FB   ..5[..$.8#......
00000D0  83 FC D8 23 CE 92 E0 E7 4A 9F 41 8E 2A AA 83 12   ...#....J.A.*...
00000E0  0D 0F DE 3E 94 78 83 D7 81 C8 FF 57 AC B7 D6 6C   ...>.x.....W...l
00000F0  B6 79 60 BF 8E B5 FF E3 FD FB EB D3 B1 FC B7 F5   .y`.............
0000100  83 FC 9E 9E E4 60 1B E1 03 E6 31 03 0A 0A 94 1A   .....`....1.....
0000110  21 AD 69 DF 54 08 3A CD 2B 1E A9 4D F0 17 8D 22   !.i.T.:.+..M..."
0000120  79 C8 52 9B FA 16 B7 AF DF 3D BF A6 2F 02 1B C7   y.R......=../...
```

Рисунок 4.8 – Результат роботи нелінійного криптографічного протоколу

Таким чином, ми створили криптографічний протокол, з нелінійною структурою шифрування, який вибірково, на основі створеного методу розгортання ключа, шифрує деякі блоки шифром AES, а деякі DES. Архітектура даного протоколу не дає можливості по шифротексту зрозуміти яким алгоритмом було зашифровано той чи інший блок.

Отже, ми успішно виконали практичну частину дослідження.

4.4 Висновок

У цьому розділі ми створили програмні прототипи нелінійних криптосистем, з використанням створеного методу розгортання криптографічного ключа, що базувались на теоретичному підґрунті, дослідженому в попередніх розділах, зокрема, у ході розробки було використано наведені раніше моделі, блок-схеми та інші матеріали.

Усі зазначені програмні прототипи були створені на основі технології динамічних бібліотек і були під’єднані до простої реалізації графічного інтерфейсу з можливістю їх зручної експлуатації.

Розроблене ПЗ було протестовано та добре показало себе в роботі. Модифіковані криптосистеми програли у швидкодії не більше ніж на 20% відносно базових алгоритмів, що говорить про достатній рівень оптимізації. За результатами тестувань не було виявлено жодних суттєвих недоліків.

Таким чином ми в повній мірі виконали усі поставлені завдання даної кваліфікаційної роботи.

ВИСНОВКИ

За результатом дослідження, згідно мети кваліфікаційної роботи, було повністю виконано перелік завдань, зокрема:

1. Досліджено предметну область за напрямком сучасної симетричної нелінійної криптографії.

2. Сформовано перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах.

3. Побудовано математичну модель процесу розгортання криптографічного ключа у нелінійній криптосистемі.

4. Синтезовано існуючі методи розгортання криптографічного ключа та криптостійких генераторів псевдо-випадкових послідовностей.

5. Створено метод розгортання криптографічного ключа, відповідно сформованих вимог та на основі результатів синтезу існуючих методів.

6. Апробовано створений метод розгортання криптографічного ключа у симетричних криптосистемах на основі нелінійних криптографічних примітивів.

7. Апробовано метод розгортання криптографічного ключа у криптографічних протоколах.

8. Підведено підсумки дослідження.

Метою кваліфікаційної роботи було розробити метод розгортання криптографічного ключа для нелінійних симетричних криптосистем, задля усунення недоліків криптосистем на основі нелінійних криптографічних примітивів.

За результатами досліджень, можна сказати що мета роботи досягнута в повному обсязі.

В ході роботи ми проаналізували шлях розвитку криптографії від давніх часів до сьогодні та встановили що кожен наступний етап розвитку криптографії нерозривно пов’язаний із попереднім та є його вдосконаленням, з усуненням тих чи інших недоліків. Також склали загальний математичний опис криптосистем на основі нелінійних криптографічних примітивів, розглянули схему їх роботи та

ключові моменти їх функціонування, зокрема найбільш оптимальні способи утворення та передачі модифікаторів, склали перелік вимог до процесу розгортання криптографічного ключа та генерації модифікаторів у нелінійних криптосистемах за трьома категоріями, серед яких: загальні вимоги щодо архітектури процесу, криптографічні вимоги та вимоги швидкодії та оптимізації, а також обґрунтували їх важливість та вкрай негативні наслідки різного характеру у випадку порушення будь-якої із них, побудували математичну модель процесу розгортання криптографічного ключа.

Після цього ми розглянули можливі напрямки застосування нелінійних криптографічних систем разом із створеним методом розгортання криптографічного ключа, кожен із них мав свої переваги та недоліки, створили метод розгортання ключа для криптосистем на основі нелінійних криптографічних примітивів та розглянули напрямки його можливого використання у складі криптографічних систем та протоколів.

На основі теоретичної бази створили програмні прототипи нелінійних криптосистем, з використанням створеного методу розгортання криптографічного ключа.

Розроблене ПЗ було протестовано та добре показало себе в роботі. Модифіковані криптосистеми програли у швидкодії не більше ніж на 20% відносно базових алгоритмів, що говорить про достатній рівень оптимізації. За результатами тестувань не було виявлено жодних суттєвих недоліків.

Висвітлена в кваліфікаційній роботі тематика має перспективу подальших досліджень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Повідайчик, Михайло Михайлович, Іван Ярославович Шпонтак. "Класичні методи криптології: методичні рекомендації для здобувачів спеціальностей «Прикладна математика» та «Системний аналіз»." (2020).

2. KESSLER, Gary C. An overview of cryptography, 2020 [Електронний ресурс]. – режим доступу: http://www.garykessler.net/library/crypto.html

3. Войцехівська І.Н. КРИПТОГРАФІЯ [Електронний ресурс] // Енциклопедія історії України: Т. 5: Кон - Кю / Редкол.: В. А. Смолій (голова) та ін. НАН України. Інститут історії України. - К.: В-во "Наукова думка", 2008. - 568 с.: іл.. – Режим доступу: http://www.history.org.ua/?termin=Kriptografiya (останній перегляд: 15.11.2023)

4. Прикладна криптологія: системи шифрування : підручник / О.Г. Корченко, В.П. Сіденко, Ю.О. Дрейс. – К. : ДУТ, 2014. – 448 с.

5. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби. / В.Л. Бурячок, С.В. Толюпа, В.В. Семко, Л.В. Бурячок, П.М. Складанний, Н.В. Лукова-Чуйко – К.: ДУТ - КНУ, 2016. – 178 с.

6. Прикладна криптологія: системи шифрування : підручник / О.Г. Корченко, В.П. Сіденко, Ю.О. Дрейс. – К. : ДУТ, 2014. – 448 с.

7. Гребенніков, Вадим Вікторович. "Історія криптології & секретного зв'язку." (2012).

8. Мудрак, Л. П. "З історії виникнення таємної писемності." ВІСНИК Житомирського державного університету імені Івана Франка 45 (2009): 208-212.

9. Qowi, Z., and N. Hudallah. "Combining caesar cipher and hill cipher in the generating encryption key on the vigenere cipher algorithm." Journal of Physics: Conference Series. Vol. 1918. No. 4. IOP Publishing, 2021.

10. Gençoğlu, Muharrem Tuncay. "Importance of cryptography in information security." IOSR J. Comput. Eng 21.1 (2019): 65-68.

11. Водніцька, О. С. "ШИФРУВАЛЬНА МАШИНА «ЕНІГМА»." МАТЕМАТИКА, ЩО НАС ОТОЧУЄ: МИНУЛЕ, СУЧАСНЕ, МАЙБУТНЄ (2023): 44.

12. Shannon, Claude Elwood. "A mathematical theory of communication." The Bell system technical journal 27.3 (1948): 379-423.

13. Tornea, O., "DNA Vernam cipher." 2011 E-Health and Bioengineering Conference (EHB). IEEE, 2011.

14. Яковлєв С. В. Конспект лекцій з дисципліни «Спеціальні розділи криптографії». — 2017.

15. Кузнецов, О. О., "Статистичні дослідження сучасних потокових шифрів." Прикладная радиоэлектроника 15,№ 3 (2016): 167-178.

16. Гапак, Оксана Михайлівна. "Криптоаналіз. Криптографічні протоколи." (2021).

17. Бабяк, Євгенія Олексіївна, Олена Олександрівна Масальська. "Проблеми криптозахисту шифрувальної машинки «ЕНІГМА»." (2015).

18. Chang, Kelly, Richard M. Low, and Mark Stamp. "Cryptanalysis of Typex." Cryptologia 38.2 (2014): 116-132.

19. Miller, A. Ray. "The cryptographic mathematics of enigma." Cryptologia 19.1 (1995): 65-80.

20. Bauer, Friedrich L. "Rotor machines and bombes." The History of Information Security. Elsevier Science BV, 2007. 381-446.

21. Фільштінський, Вадим, and Андрій Володимирович Бережний. "Математичні основи криптографії." (2011).

22. Бессалов, Анатолій Володимирович. "Математичні методи криптографії." (2019).

23. Biryukov, Alex. "Block ciphers and stream ciphers: The state of the art." Cryptology EPrint Archive (2004).

24. Walia, Anjula Gupta1 Navpreet Kaur. "Cryptography Algorithms: A Review." International Journal of Engineering Development and Research 146 (2014).

25. Thakur, Jawahar, and Nagesh Kumar. "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis." International journal of emerging technology and advanced engineering 1.2 (2011): 6-12.

26. Adhie, Roy Pramono, "Implementation cryptography data encryption standard (DES) and triple data encryption standard (3DES) method in communication system based near field communication (NFC)." Journal of Physics: Conference Series. Vol. 954. No. 1. IOP Publishing, 2018.

27. Hoang, Viet Tung, and Phillip Rogaway. "On generalized Feistel networks." Annual Cryptology Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

28. Драга, В. С. "Звязок між методами формального представлення фейстель-подібних мереж." (2021).

29. Система обробки інформації. Криптографічний захист інформації. Алгоритм криптографічного перетворення (ГОСТ 28147-89) : державний стандарт України, 2009. URL: https://usts.kiev.ua/wp-content/uploads/2020/07/dstu-host-28147-2009.pdf (дата звернення: 15.11.2023).

30. Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

31. Бганцов, Євгеній. "Шифрування даних. Опис та структура алгоритму симетричного шифрування AES." (2023).

32. Dworkin, Morris J. "Advanced Encryption Standard (AES)." (2023).

33. Benamira, Adrien, "A deeper look at machine learning-based cryptanalysis." Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40. Springer International Publishing, 2021.

34. Гапак, Оксана Михайлівна. "Криптоаналіз. Криптографічні протоколи." (2021).

35. Цубера, Василь Васильович. Криптоаналіз історичних шифрів заміни. BS thesis. 2021.

36. Smirnova, Tetiana, "ДОСЛІДЖЕННЯ СТІЙКОСТІ ДО ЛІНІЙНОГО КРИПТОАНАЛІЗУ ЗАПРОПОНОВАНОЇ ФУНКЦІЇ ГЕШУВАННЯ УДОСКОНАЛЕНОГО МОДУЛЯ КРИПТОГРАФІЧНОГО ЗАХИСТУ В

ІНФОРМАЦІЙНО КОМУНІКАЦІЙНИХ СИСТЕМАХ." Системи управління, навігації та зв'язку. Збірник наукових праць 1.67 (2022): 84-89.

37. Cryptology and communication security [Електронний ресурс]/ Shri Kant. – Defense Science Journal. – Vol. 62. - №1. – режим доступу: https://core.ac.uk/reader/333719963

38. Гнатюк, Сергій, Василь Кінзерявий, and Андрій Охріменко. "Особливості криптографічного захисту державних інформаційних ресурсів." Безпека інформації 1 (2012): 68-8.

39. Alekseychuk, A. N. "Узагальнений диференціально-лінійний криптоаналіз блокових шифрів." Radiotekhnika 204 (2021): 5-15.

40. Тафтай, Анастасія, Сергій Яковлєв, and Ганна Южакова. "Методика оцінювання стійкості SP-мереж до узагальненого лінійного криптоаналізу." " Інформаційні технології та комп'ютерне моделювання"; матеріали статей Міжнародної науково-практичної конференції, м. Івано-Франківськ, 15-16 грудня 2022 року.–Івано-Франківськ: п. Голіней ОМ, 2022.–с. (2022): 104.

41. Heys, Howard M. "A tutorial on linear and differential cryptanalysis." Cryptologia 26.3 (2002): 189-221.

42. Matsui, Mitsuru. "Linear cryptanalysis method for DES cipher." Workshop on the Theory and Application of of Cryptographic Techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993.

43. Biham, Eli, and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems." Journal of CRYPTOLOGY 4 (1991): 3-72.

44. Лупол, А. А., and П. К. Ніколюк. "Дослідження різних методів криптоаналізу та їх ефективності в розшифруванні зашифрованих повідомлень." Прикладні інформаційні технології (2023): 226-229.

45. Bar-On, Achiya, et al. "DLCT: a new tool for differential-linear cryptanalysis." Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38. Springer International Publishing, 2019.

46. Анікін В.А. Симетрична криптосистема з нелінійним шифруванням та можливістю контролю шифротексту з метою маскування / В. А. Анікін, В. М. Джулій, І.В. Муляр, В.С. Орленко, В.Ю. Тітова // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 6. – С. 33-38.

47. Анікін В.А. Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування: наукова робота студента / Анікін Володимир Андрійович. - ХНУ, 2021. - 47 с.

48. Анікін В.А. Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування / В.А. Анікін, І.В. Муляр // «Інтелектуальний потенціал – 2020» – збірник наукових праць молодих науковців і студентів / Колектив авторів – Хмельницький: ПВНЗ УЕП, 2020. – Ч.2. – С. 93-97.

49. Анікін В. А. Симетрична поліалфавітна криптосистема на основі випадкової генерації коефіцієнту перестановки / В. А. Анікін , В. О. Бойчук // Тези доповідей XVI Міжнародної науково - практичної конференції " Військова освіта і наука : сьогодення та майбутнє ", 27 листоп. 2020 р. – Київ : ВІКНУ , 2020. – Т. 1. – С. 20

50. Анікін В. А. Побудова симетричної криптосистеми з нелінійним шифруванням / В. А. Анікін , А. О. Рамський, О. В. Мірошніченко, С. Ф. Стремецький // Тези доповідей Всеукраїнської науково-практичної конференції молодих вчених, ад’юнктів, слухачів, курсантів і студентів "Молодіжна військова наука у Київському національному університеті імені Тараса Шевченка", 23 квітня 2021 р. – Київ : ВІКНУ , 2021. – Т. 1. – С. 100

51. Yu, Shanshan, et al. "Development of Modified Blum-Blum-Shub Pseudorandom Sequence Generator and its Use in Education." Measurement Science Review 22.3 (2022): 143-151.

52. Selvakumar, David, et al. "Formal verification and analysis of a pseudo random number generator." 2021 25th International Symposium on VLSI Design and Test (VDAT). IEEE, 2021.

53. Dobreva, Jovana, et al. "A Comparative Analysis of HOTP and TOTP Authentication Algorithms. Which one to choose?." Security & Future (2021).

54. Власов, Владислав Миколайович. "Метод захисту Web-сторінок Інтернет магазину." (2021).

ДОДАТОК А

Копії наукових публікацій

Технічні науки *ISSN 2307-5732*

DOI 10.31891/2307-5732-2020-291-6-33-38
УДК 004.891

В.М. ДЖУЛІЙ, І.В. МУЛЯР, В.С. ОРЛЕНКО, В.Ю. ТІТОВА, В.А. АНІКІН
Хмельницький національний університет

СИМЕТРИЧНА КРИПТОСИСТЕМА З НЕЛІНІЙНИМ ШИФРУВАННЯМ ТА МОЖЛИВІСТЮ КОНТРОЛЮ ШИФРОТЕКСТУ З МЕТОЮ МАСКУВАННЯ

В статті запропонована проста симетрична перестановочна криптосистема, перестановки в якій відбуваються на основі потокової генерації випадкових чисел. Наведений криптографічний алгоритм, в зв'язку з нелінійністю процесу шифрування, здатний шифрувати одні і ті ж блоки інформації по-різному. Дана властивість може бути використана для формування шифротексту конкретного вигляду та, як наслідок, для прихованої передачі шифрованих повідомлень і повного, або часткового маскування факту криптографічного впливу.

Підстановочно-перестановочні алгоритми достатньо розповсюджені на сьогоднішній день. Вони зарекомендували себе як надійні криптостійкі шифри, що склали серйозну конкуренцію ітеративним математичним шифрам, на базі мережі Фейстеля, та поступово все більше та більше витісняють їх, займаючи їх місце.

В статті запропонований алгоритм шифрування, який має просту структуру та легкий в реалізації, але при цьому, прогнозовано, він є достатньо стійким для криптоатак, має високу швидкість роботи та використовує всі переваги сучасних алгоритмів шифрування та має потенціал для маскування шифрованих даних.

Розроблено схему шифрування, в якій перестановки відбуваються у випадковому порядку, та, як наслідок, стає можливою ситуація, коли одному відкритому тексту при одному і тому ж ключі відповідає безліч шифротекстів, що серйозно ускладнює роботу криптоаналітиків, ускладнюючи аналіз та злом шифру, а також суттєво збільшує область прикладного використання даної криптосистеми. Також дана криптосистема має стеганографічний потенціал, оскільки нелінійність, яка породжує варіативність шифротекстів, дозволяє, фактично, один і той ж самий байт зашифрувати десятками, або навіть сотнями комбінацій.

Запропоновано новий підхід до створення шифрувальної варіативності та випадкового вибору, що дає можливість модифікації вже існуючих алгоритмів шифрування.

Ключові слова: криптографія, стеганографія, симетрична криптосистема, перестановочний шифр, випадкова генерація, захист інформації.

V.M. DZHULIY, I. V. MULYAR, V.S. ORLENKO, V. YU. TITOVA, V. A. ANIKIN
Khmelnytsky National University, Khmelnytsky, Ukraine

SYMMETRIC CRYPTOSYSTEM WITH NONLINEAR ENCRYPTION AND THE POSSIBILITY OF CONTROL OF CIPHERTEXT FOR CONCEALMENT

The article proposes a simple symmetric permutation cryptosystem, in which permutations occur based on streaming generation of random numbers.

Substitution-permutation algorithms are quite common today. They have proven to be reliable crypto-strong ciphers that have seriously competed with iterative mathematical ciphers, based on the Feistel network, and are gradually displacing them more and more, taking their place.

The aim of the article is to propose an encryption algorithm that has a simple structure and is easy to implement. At the same time, it is estimated to be quite resistant to crypto attacks. This encryption algorithm has a high speed and uses all the advantages of permutation encryption algorithms.

Developed an encryption scheme in which permutations occur randomly, and as a result, a situation becomes possible when one plaintext with the same key corresponds to many ciphertexts, which seriously complicates the work of cryptanalysts, complicating the analysis and cracking of the cipher, and significantly increases the scope of application of this cryptosystem. Also, this cryptosystem has steganographic potential, because the nonlinearity that creates the variability of ciphertexts, allows the same byte to encrypt dozens or even hundreds of combinations.

This paper has clearly shown a new approach to the creation of encryption variability and random selection is proposed. It makes possible to modify existing encryption algorithms.

Key words: cryptography, steganography, symmetric cryptosystem, permutation cipher, random generation, information protection.

Вступ

У всі часи люди намагалися захищати інформацію, яку вважають важливою, постійно розвиваючи та удосконалюючи засоби для її захисту. Стислий Оксфордський словник англійської мови визначає криптографію як «мистецтво написання або розв'язування кодів". Це історично точно, але воно не враховує сучасної розповсюдженості галузі та її сьогоденних наукових основ [1]. На сьогоднішній день криптографія має вкрай широкий спектр застосувань, починаючи від класичного захисту інформації при зберіганні, чи передачі, закінчуючи контролем цілісності повідомлень та цифровими підписами, для верифікації даних.

Також здавна люди намагались приховувати ту чи іншу інформацію, різноманітними способами маскуючи її. Такі починання згодом переросли в науку під назвою стеганографія, що на сьогоднішній день широко використовується в сфері захисту інформації.

У розрізі цього, не видається дивним що здавна і до сьогодні люди працюють над розробкою нових і нових криптосистем: шифр Цезаря змінили криптосистеми простої заміни, їх, в свою чергу, витіснили поліалфавітні системи, на кшталт шифрів Гронсфельда та Віженера [2], а їх, пізніше, – ітеративні блочні шифри, на основі мережі Фейстеля, яскравим прикладом яких є DES [3], різноманітні модифікації якого широко використовуються до сьогоднішнього дня.

Разом із розвитком криптосистем, невпинно розвивався і криптоаналіз, накопичений інструментарій якого [4], у поєднанні з технічними можливостями сучасної обчислювальної техніки, задає вкрай високу планку вимог для сучасної криптографії вцілому та безпосередньо для використовуваних криптосистем.

На сьогоднішній день одними з найбільш захищених криптосистем у світі є шифри, на основі підстановочно–перестановочних систем, наприклад SQUARE та Rijndael, останній відомий на широкий загал як Advanced Encryption Standard (AES) [3, 5].

З огляду на це, можна стверджувати що підстановочно–перестановочні системи, спроектовані таким чином, щоб максимально використовувати сучасні обчислювальні потужності та досягати високої ентропії при шифруванні, здатні надійно захищати інформацію та є стійкими до сучасних криптоатак.

Також, безумовно, важливою складовою захисту інформації є непомітність. Логічно, що інформацію, яка виглядає цілісно, осмислено, без слідів того що в ній щось приховано, або зашифровано, навряд чи хто буде досліджувати без суттєвих на те підстав. Саме через це в сучасному світі, у сфері кібербезпеки, все частіше використовується стеганографія – наука, що до надає інструментарій для приховування однієї, умовно «таємної», інформації всередині іншої – публічної, тієї що не представляє для сторонніх осіб жодного інтересу. В зв'язку з цим, виникає великий попит на алгоритми, які підходять до захисту комплексно: вони здатні як захистити інформацію криптографічно, так і, повністю, або частково, замаскувати сам факт того що деяка інформація була прихована.

Огляд літературних джерел

Симетричні криптосистеми давно відомі науці та є в достатній мірі вивчені та проаналізовані. В зв'язку з цим, дослідження симетричних систем шифрування знайшло відбиток у великій кількості наукових статей та літературі. Дані напрацювання обов'язково слід враховувати при розробці криптосистеми.

Симетричні криптосистеми на протязі багатьох століть використовувались для захисту інформації та «закриття» важливої інформації від несанкціонованого використання [6, 7, 8]

Симетричні алгоритми шифрування перетворюють деяку конфіденційну інформацію, на основі криптографічного ключа, у шифровану форму. Зворотне перетворення, яке відновлює вхідну інформацію, відбувається при зворотному виконанні алгоритму, за умови, що у вас є такий цей ж самий ключ [8]. Сучасні симетричні криптосистеми, хоч і є більш витонченими, математично та комбінаторно обґрунтованими, все одно, як правило дотримуються таких принципів.

Також важливою складовою при розробці криптосистеми є оцінка інструментарію криптоаналізу, та сучасні можливості у сфері «взлому» систем шифрування. Для створення криптостійкого алгоритму слід на етапі проектування враховувати те, яким чином він буде поводитись під час криптоатаки та забезпечити достатній запас міцності та стійкості [9 - 11].

Постановка задачі

Необхідно створити простий у реалізації криптостійкий перестановочний симетричний алгоритм шифрування, який би відповідав сучасним вимогам безпеки та значно ускладнював існуючі способи криптоаналізу, при цьому з можливістю корегування вихідного шифротексту.

Основна ідея даного алгоритму полягає в тому, що P-блок, за яким відбуватиметься перестановка, для кожного байту обирається випадково, на основі генератора випадкових чисел. Такий підхід значно ускладнить будь-який статистичний криптоаналіз, оскільки будь-які закономірності між відкритими байтами та зашифрованими будуть відсутні. Потокова випадкова генерація коефіцієнта перестановки дозволить досягти неоднорідності шифрування та відсутності закономірностей у ньому.

Також, через відсутність складної ітеративності, даний алгоритм повинен мати високу швидкість роботи, порівняно з аналогами.

Основна частина

Відповідно до поставленої задачі, криптосистема повинна відповідати сучасним вимогам безпеки, бути надійною та придатною для практичного використання, забезпечуючи високу надійність, а отже, даний шифр повинен бути стійким перед інструментарем сучасного криптоаналізу.

Одною з головних потужностей криптоаналізу на сьогодні є обчислювальна потужність комп'ютерної техніки, яка серйозно збільшується з кожним новим поколінням. Фактично будь-який симетричний шифр можливо взламати атакою «грубої сили», тобто повним перебором всіх можливих криптографічних ключів, та почерговими спробами розшифрування кожним з них. До сьогоднішнього дня даний метод – найбільш універсальний серед всіх існуючих, за умови що нам відомо тип алгоритму та принцип його роботи, при цьому не важко здогадатись, що ККД даного методу є вкрай низьким.

Для оцінки реальності загрози атаки грубої сили вводиться термін «практичний час», як деяка відповідність теоретичному часу, за який машина прогнозовано зможе підібрати ключ, до часу збереження актуальності та цінності прихованої інформації. Простіше кажучи, якщо умовна Єва перехопила зашифроване повідомлення компанії конкурента, і на атаку грубої сили, прогнозовано буде затрачено 50 років, то такий час можна вважати непрактичним, адже за 50 років розшифроване повідомлення дасть Єві мінімум користі. При цьому обов'язково слід врахувати той факт, що якщо на сьогоднішній день прогнозований час атаки й складатиме 50 років, то абсолютно не факт, що через кілька років, з урахуванням нових технологій та потужностей, цей час не складатиме 50 годин, чи навіть хвилин. Окрім цього на сьогоднішній день все більшої популярності набувають розподілені обчислення, що переганяють за швидкістю роботи навіть передові суперкомп'ютери [12].

Таким чином запас міцності, в контексті стійкості криптосистеми до атаки грубої сили, як одна з базових характеристик надійності шифру повинна враховувати не лише обчислювальні потужності, існуючі сьогодні, а і їх прогнозоване зростання, яке є неминучим.

Слід враховувати й безліч інших існуючих криптоатак, такий як, наприклад, атака на основі перехопленого відкритого тексту, атака на основі підібраного відкритого тексту, диференціальний криптоаналіз та інші існуючі підходи.

Головна сила запропонованої криптосистеми – нелінійність та рандомізація алгоритму, саме це і повинно забезпечити його високу криптостійкість. В цілому, ідея полягає у створенні кількох можливих варіантів роботи шифру на тому, чи іншому етапі його виконання, серед яких випадковим чином обирається один. Для можливості дешифрування, позначається лише індекс обраного варіанту, який не дає криптоаналітику жодної додаткової інформації про те, що за ним стоїть.

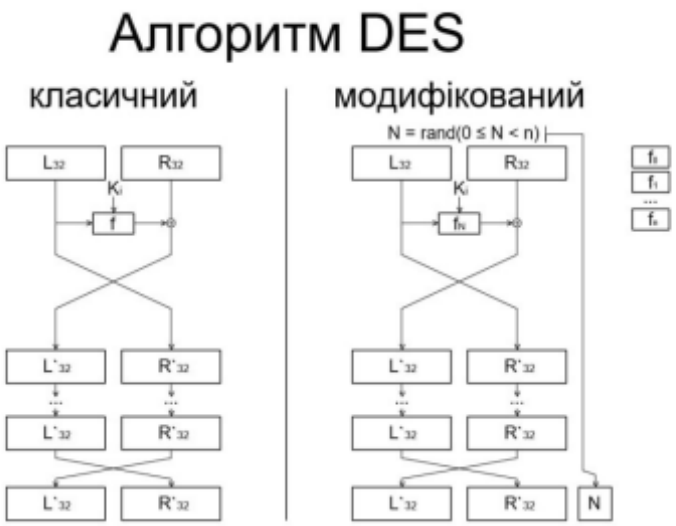


Рис. 1. Схема звичайного та модифікованого алгоритму DES

Даний підхід можна продемонструвати на прикладі алгоритму DES (Рис. 1).

В класичному представлені даного алгоритму у нас є одна функція, яка на основі раундового ключа та першого підблоку мережі Фейстеля, розміром 48 та 32 біта, відповідно, та повертає новий 32-бітний блок даних. Проте давайте змоделюємо ситуацію, при якій в даному алгоритмі у нас не одна функція, а, до прикладу, їх вісім. Кожна з цих функцій також прийматиме на вхід два блока у 48 та 32 біт та повертатиме, на їх основі, новий 32-бітний блок, при цьому кожна з них буде робити це по-різному. Якщо точніше, вони будуть і далі працювати за схемою класичної функції DES, проте, кожна з цих 8 функцій матиме деякі особливості: одна, наприклад, по-іншому розширюватиме вхідний блок, друга – перед розширенням проводитиме циклічний зсув вхідного блоку на деяке число, третя – проводитиме аналогічний зсув, але вже після розширення, четверта – залишиться у класичному вигляді, без змін, а п'ята – додаватиме 27, за модулем 256, до кожного парного байту після розширення вхідного блоку і так далі. Як результат – кожна функція, в цілому, подібна до інших та кожну з них можна без проблем замінити іншою, проте 32-бітні блоки які буде повертати та, чи інша функція, як правило, будуть різні.

Тепер нам залишається лише незначно модифікувати загальний алгоритм шифрування: при генерації ключа, додатково, випадковим чином цим функціям буде присвоєно індекси від 0 до 9. Інформація про дані індекси та їх відповідності буде частиною секретного ключа. А при шифруванні, перед першим раундом кожного блоку, ми обиратимемо випадкове ціле число в діапазоні $[0; 8)$, яке і вказуватиме яку функцію з переліку ми використовуватимемо у наступних 16 раундах. Інформацію про те яка функція була використана можемо записати у вигляді 3-бітної комбінації в кінці, після зашифрованого 64-бітного блоку.

При такій модифікації алгоритму шифрований текст розшириться на n біт, де $n = \frac{N_{\text{повідомлення}} \times 67}{64}$, тобто, приблизно, на 4,5%.

В описаному випадку, при перехопленні шифрованого повідомлення, у криптоаналітика будуть на руках самі шифроблоки, а також індекси використаних функцій, без будь-якої важливої інформації, звичайно, якщо криптоаналітик знає принцип роботи даної модифікації. Проте в даній ситуації спеціаліст зможе однозначно сказати чи однакові функції були використані, в тому, чи іншому випадку, чи ні, елементарно зрівнявши ці самі індекси вкінці блоків. На основі даної інформації він зможе згрупувати блоки, в яких були використані однакові функції та досліджувати їх окремо.

Для того щоб позбавити криптоаналітика такої можливості, введемо ще одну зміну: індексів буде більше ніж самих функцій. На описаному вище прикладі, це можна зробити двома шляхами: або зменшити кількість функцій, тоді в нас буде, наприклад, 4 функції та 8 індексів, або збільшити кількість індексів, тоді при 8 функціях ми матимемо 16 індексів.

При генерації нового ключа, як і раніше, ми випадково генеруємо та записуємо відповідності між функціями та індексами, проте вже не у відношенні 1:1, а у відношенні 1:m, де одній функції відповідатиме кілька індексів. При шифруванні тепер ми кидатимемо монетку двічі: перший раз – щоб вибрати функцію, другий – щоб обрати один індекс, серед всіх, пов'язаних з даною функцією.

Слід зазначити що дані числа краще обирати рівним 2^n, а кількість функцій, рекомендовано, має бути кратною кількості індексів, при чому вони повинні бути рівномірно розподілені між функціями. Це дозволить в повній мірі, без надлишковості, використовувати діапазон значень, при шифруванні, а також забезпечить однакову ймовірність випадання функцій.

Тепер, перехопивши повідомлення, криптоаналітик не зможе однозначно сказати чи були використані однакові функції в блоках з різними індексами, адже це може бути як інший ідентифікатор тієї ж самої функції, так й ідентифікатор іншої функції.

Також звідси витікає цікава властивість даного алгоритму: зашифрувавши одне і те ж саме повідомлення, використовуючи один і той ж самий ключ, ми отримаємо різні шифротексти, оскільки тепер те як буде зашифровано той чи інший блок, залежатиме від випадкового показника, при шифруванні. Отже, криптоаналітик, перехопивший n зашифрованих повідомлень, не зможе однозначно виявити які серед них будуть ідентичними після дешифровки.

Приклад, наведений на основі криптографічного алгоритму DES, наглядно та в повній мірі демонструє принципи роботи, на базі яких пропонується створити окрему криптосистему.

Навіть у вигляді модифікації до вже існуючого алгоритму, така нелінійність, наявність кількох «сценаріїв» роботи, серед яких обирається випадковий, дозволяє значно ускладнити роботу криптоаналітиків та понизити ефективність машинного криптоаналізу. Причому слід зазначити, що в описаній модифікації зазначена варіативність була додана лише для раундових функцій, хоча це зовсім не єдиний аспект, де даний підхід можна було б застосувати, навіть у випадку шифру DES.

Таким чином, аналогічно до наведеного прикладу ми можемо спроектувати окремий алгоритм, в якому описаний принцип нелінійності буде не модифікацією, а основою його роботи.

Логікою шифрування в даному алгоритмі буде заміна одного байту на інший, відповідно до деякої поліалфавітної таблиці перестановок. Дана таблиця міститиме в собі n рядків та 256 стовпців, якщо за умовний блок даних ми візьмемо один байт. В кожному рядку буде послідовність від 0 до 255, перемішана випадковим чином. Кількість рядків, а відповідно «алфавітів перестановки», може бути якою завгодно.

Кожному рядку, точно як і функції, у прикладі з DES, присвоюється m випадкових двійкових ідентифікаторів. Розрядність цих ідентифікаторів також може бути обрана різна, проте вона повинна бути однаковою для всіх ідентифікаторів. Від обраної розрядності залежатиме, по-перше, кількість алфавітів, яку ми можемо визначити, наприклад якщо розрядність дорівнюватиме двом, то максимум у нас може бути 4 алфавіти, а з урахуванням рекомендації використовувати не менше двох ідентифікаторів на один алфавіт – всього 2. По-друге, від розрядності залежатиме те, наскільки збільшиться шифрований текст, порівняно з вхідним повідомленням. Якщо розрядність ідентифікатора буде рівною розрядності нашого блоку даних, а у нашому випадку – це 8 біт, то шифроване повідомлення буде вдвічі більше за вхідне. Чим менша розрядність ідентифікатора, по відношенню до розміру блоку даних, тим менше буде збільшення вихідного повідомлення при шифруванні і навпаки.

Дана таблиця замін, разом із відповідними ідентифікаторами, складатиме собою секретний ключ. Для її компактного запису можна використовувати різноманітні технології стиснення.

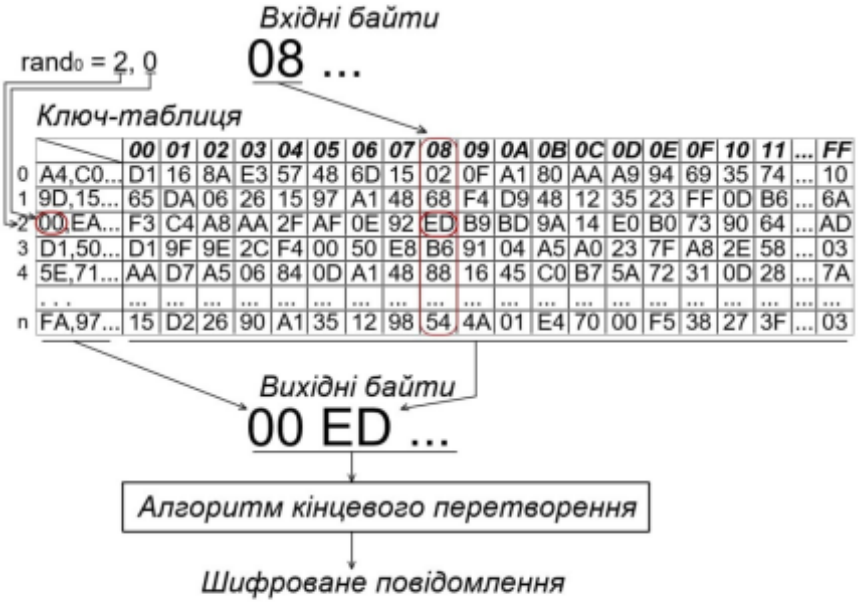


Рис. 2. Схема шифрування

Схема шифрування даним методом показана на рис. 2 та в цілому подібна до описаної на прикладі DES: спочатку ми обираємо два випадкових числа, з яких перше – в діапазоні $[0; n)$, де n – кількість алфавітів, а друге – в діапазоні $[0; m)$, де m – кількість ідентифікаторів, відповідних даному алфавіту. Після цього ми заміняємо вхідний блок даних на вираз, що складається з обраного ідентифікатора та числа, що знаходиться в обраному алфавіті на позиції, номер якої відповідає числу, утвореного з блока вхідних даних. Тобто, якщо ми шифруємо 8-бітні блоки, і на вході, як приклад, отримали блок «10001010», що у десятковій формі відповідає числу 138, а випадкові числа випали 2 і 0, за умови, що вони відповідатимуть межам зазначених діапазонів, то шифрований вираз для даного блоку складатиметься з 0-го ідентифікатора 2-го алфавіту перестановок та числа, що знаходиться на 138-й позиції у 2-му алфавіті. Такі перетворення почергово проводяться для кожного блоку даних, до кінця відкритого повідомлення.

В прикладі з DES на цьому ми закінчували, проте в даному випадку не слід залишати шифроване повідомлення в такому вигляді, оскільки у обізнаного криптоаналітика, при перехопленні буде можливість проаналізувати однакові ідентифікатори та відповідно, на основі великої кількості повідомлень скласти деяку кореляцію. Щоб позбавити його такої можливості, ми використаємо два прийоми.

По-перше, ми додамо варіативності до принципу зчеплення ідентифікатора та блоку заміни. Це дозволить ускладнити розпізнавання ідентифікаторів та ускладнить криптоаналіз. Приклади зчеплень елементів продемонстровано на рис. 3.

По-друге, після формування вихідного масиву байт, після завершення замін, ми «змішаємо» його за допомогою алгоритму кінцевого перетворення. Найпростішим варіантом даного алгоритму може бути поблокове додавання за модулем 2, в режимі зчеплення блоків, де ключ та перший блок деяким чином формуються на основі таблиці. Даний алгоритм може бути реалізовано на основі мережі Фейстеля з невеликою кількістю раундів, на основі матричних перестановок, чи на основі будь-якого іншого існуючого алгоритму, що забезпечує надійне «розмиття».

Таким чином даний алгоритм буде стійким до атаки грубою силою, оскільки простим перебором відновити таблицю замін не можливо за практичний час, а за необхідності, розмір вхідного блока даних можна додатково збільшити. Кількість можливих варіантів перестановок для одного 8-бітного алфавіту складає 256!, а за умови що алфавітів багато, кількість можливих комбінацій забезпечує надійність ключа навіть з урахуванням розвитку технологій.

Слід зауважити, що при абсолютно випадковій генерації ключа, шифротекс, звісно, також вийде випадковий. При цьому, ми можемо піти від зворотнього: припустимо що в нас вже є шифротекст, який ми маємо отримати. Ми можемо згенерувати такий ключ, щоб при шифруванні конкретним способом, без викидання випадкових чисел, ми отримали необхідний нам шифротекст. При даному підході також з'являються додаткові вимоги до алгоритму кінцевого перетворення: він повинен мати змогу розшифрувати будь-який випадковий масив байт.

Рис. 3 Приклади зчеплення ідентифікатора та блоку заміни

Нелінійність алгоритму ускладнить машинний та людський криптоаналіз та затруднить статистичні дослідження.

Даний алгоритм також буде стійким на до атаки на основі відкритого тексту, оскільки, маючи на руках відповідне шифроване повідомлення, криптоаналітик матиме лише один з можливих варіантів шифротексту, при цьому криптоаналітик не матиме доступу до всіх інших можливих варіантів шифрування.

В цілому запропонований алгоритм представляє собою цікавий та перспективний метод криптографічного захисту, безумовно потребуючи розвитку та подальших досліджень. Окрім криптографічного, даний алгоритм представляє також стеганографічний інтерес, оскільки нелінійність, яка породжує варіативність шифротекстів, дозволяє, фактично, один і той ж самий байт зашифрувати десятками, або навіть сотнями комбінацій. Виходячи з цього, теоретично, можливо створити такий ключ, при якому деякий конфіденційний текст в зашифрованому вигляді представлятиме інший осмислений текст, що не викликатиме підозр. Даний аспект також потребує додаткових досліджень.

Висновки

Таким чином, алгоритми шифрування, що мають в собі елементи нелінійності, включають рівноможливі варіанти виконання, серед яких в процесі виконання обирається один, значно підвищують його стійкість та захищенність.

Запропонований алгоритм, який в повній мірі відповідає описаним принципам, проявляє цікаві властивості, а проведенні дослідження говорять про його достатню захищеність та стійкість.

Безумовно, наведені алгоритми потребують додаткового вивчення, проте вже на даному етапі можна говорити про те, що навіть модифікація вже існуючих алгоритмів, введення в них варіативних елементів, є легким способом підвищення їх надійності.

Окрім зазначеного, даний алгоритм може знайти розвиток у сфері стеганографії.

Літертура

1. Інформаційна безпека: навчальний посібник / [Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв та ін.]; за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого. – Львів: Видавництво Львівської політехніки, 2019. – 580 с.

2. Прикладна криптологія: системи шифрування : підручник / О. Г. Корченко, В. П. Сіденко, Ю. О. Дрейс. – К. : ДУТ, 2014. – 448 с.

3. Основы криптографии/ Г.В. Басалова. – М.: ИНТУИТ, 2016. - 283 с.

4. Cryptology and information security - past,present, and future role in society/ S. Bhattacharya. - International Journal on Cryptography and Information Security (IJCIS). – Vol. 9, No.1/2, 2019. – P. 13-36.

5. Муляр І.В. Ітераційно-геометричний метод для стійкого перцептуального хешування зображення / В. М. Джулій, Ю. П. Кльоц, І. В. Муляр, В. М. Чешун // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 1. – С. 76–79

6. Hybrid cryptographic technique using rsa algorithm and scheduling concepts/ M. Shankar, P. Akshaya// International Journal of Network Security & Its Applications (IJNSA)/ Vol.6, No.6. – 2014. - p. 39-48

7. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних: підручник. / Г. Ф. Конахович, Д. О. Прогонов, О. Ю. Пузиренко. – Київ: «Центр учбової літератури», 2018. – 558 с.

8. Digital Watermarking and Steganography: Fundamentals and Techniques / Edited by Frank Y. Shih. – Taylor & Francis Group, 2017. – 293 p

9. The Evolution of Cryptology [Електронний ресурс]/ G. R. Souza. – California State University, San Bernardino, 2016. – режим доступу: https://core.ac.uk/reader/55336770

10. Cryptology and communication security [Електронний ресурс]/ Shri Kant. – Defense Science Journal. – Vol. 62. - №1. – режим доступу: https://core.ac.uk/reader/333719963

11. Криптоанализ шифрсистемы ACBF/ И. В. Боровкова, И. А. Панкратова. – Прикладная дискретная математика. Приложение, Томск. – 2019. – С. 90-93.

12. Захист інформаційних ресурсів: криптографічні та стеганографічні методи захисту даних. Посібник для викладачів, вчителів та студентів інформатичних спеціальностей./ В.М. Франчук – К.: НПУ імені М.П. Драгоманова, 2014. – 120 с.

References

1. Informatsiina bezpeka: navchalnyi posibnyk / [Iu. Ya. Bobalo, I. V. Horbatyi, M. D. Kiselychnyk, A. P. Bondariev ta in.]; za zah. red. d-ra tekhn. nauk, prof. Yu. Ya. Bobala ta d-ra tekhn. nauk, dots. I. V. Horbatoho. – Lviv: Vydavnytstvo Lvivskoi politekhniky, 2019. – 580 s.

2. Prykladna kryptolohiia: systemy shyfruvannia : pidruchnyk / O. H. Korchenko, V. P. Sidenko, Yu. O. Dreis. – K. : DUT, 2014. – 448 s.

3. Osnovы kryptohrafyy/ H.V. Basalova. – M.: YNTUYT, 2016. - 283 s.

4. Cryptology and information security - past,present, and future role in society/ S. Bhattacharya. - International Journal on Cryptography and Information Security (IJCIS). – Vol. 9, No.1/2, 2019. – P. 13-36.

5. Muliar I.V. Iteratsiino-heometrychnyi metod dlia stiikoho pertseptualnoho kheshuvannia zobrazhennia / V. M. Dzhulii, Yu. P. Klots, I. V. Muliar, V. M. Cheshun // Visnyk Khmelnytskoho natsionalnoho universytetu. Tekhnichni nauky. – 2020. – № 1. – S. 76–79

6. Hybrid cryptographic technique using rsa algorithm and scheduling concepts/ M. Shankar, P. Akshaya// International Journal of Network Security & Its Applications (IJNSA)/ Vol.6, No.6. – 2014. - p. 39-48

7. Kompiuterna stehanohrafichna obrobka y analiz multymediinykh danykh: pidruchnyk. / H. F. Konakhovych, D. O. Prohonov, O. Yu. Puzyrenko. – Kyiv: «Tsentr uchbovoi literatury», 2018. – 558 s.

8. Digital Watermarking and Steganography: Fundamentals and Techniques / Edited by Frank Y. Shih. – Taylor & Francis Group, 2017. – 293 p

9. The Evolution of Cryptology [Elektronnyi resurs]/ G. R. Souza. – California State University, San Bernardino, 2016. – rezhym dostupu: https://core.ac.uk/reader/55336770

10. Cryptology and communication security [Elektronnyi resurs]/ Shri Kant. – Defense Science Journal. – Vol. 62. - №1. – rezhym dostupu: https://core.ac.uk/reader/333719963

11. Kryptoanalyz shyfrsystemы ACBF/ Y. V. Borovkova, Y. A. Pankratova. – Prykladnaia dyskretnaia matematyka. Prylozhenye, Tomsk. – 2019. – S. 90-93.

12. Zakhyst informatsiinykh resursiv: kryptohrafichni ta stehanohrafichni metody zakhystu danykh. Posibnyk dlia vykladachiv, vchyteliv ta studentiv informatychnykh spetsialnostei./ V.M. Franchuk – K.: NPU imeni M.P. Drahomanova, 2014. – 120 s.

Надійшла / Paper received : 12.11.2020 р. Надрукована/Printed :04.01.2021 р.

Анікін В.А. (ХмНУ)
к.ф.-м.н., доц. Рамський А.О. (ХмНУ)
к.т.н., с.н.с. Мірошніченко О.В. (ВІКНУ)
Стремецький С.Ф. (ХмСЗОШ №29)

Побудова симетричної криптосистеми з нелінійним шифруванням

Разом із розвитком криптосистем, невпинно розвивався і криптоаналіз, накопичений інструментарій якого, у поєднанні з технічними можливостями сучасної обчислювальної техніки, задає вкрай високу планку вимог для сучасної криптографії в цілому та безпосередньо для використовуваних криптосистем.

На сьогоднішній день одними з найбільш захищених криптосистем у світі є шифри, на основі підстановочно–перестановочних систем, наприклад SQUARE та Rijndael, останній відомий на широкий загал як Advanced Encryption Standard (AES).

З огляду на це, можна стверджувати що підстановочно–перестановочні системи, спроектовані таким чином, щоб максимально використовувати сучасні обчислювальні потужності та досягати високої ентропії при шифруванні, здатні надійно захищати інформацію та є стійкими до сучасних криптоатак.

Симетричні криптосистеми давно відомі науці та є в достатній мірі вивчені та проаналізовані. В зв'язку з цим, дослідження симетричних систем шифрування знайшло відбиток у великій кількості наукових статей та літературі. Дані напрацювання обов'язково слід враховувати при розробці криптосистеми.

Симетричні криптосистеми на протязі багатьох століть використовувались для захисту інформації та «закриття» важливої інформації від несанкціонованого використання.

Необхідно створити простий у реалізації криптостійкий перестановочний симетричний алгоритм шифрування, який би відповідав сучасним вимогам безпеки та значно ускладнював існуючі способи криптоаналізу, при цьому з можливістю корегування вихідного шифротексту.

Основна ідея даного алгоритму полягає в тому, що P-блок, за яким відбуватиметься перестановка, для кожного байту обирається випадково, на основі генератора випадкових чисел. Такий підхід значно ускладнить будь-який статистичний криптоаналіз, оскільки будь-які закономірності між відкритими байтами та зашифрованими будуть відсутні. Потокова випадкова генерація коефіцієнта перестановки дозволить досягти неоднорідності шифрування та відсутності закономірностей у ньому.

Також, через відсутність складної ітеративності, даний алгоритм повинен мати високу швидкість роботи, порівняно з аналогами.

The advantages of nonlinear cryptographic primitives and their areas of use

Serhii Lienkov[1], Ihor Muliar [2*], Volodymyr Anikin [3], Volodymyr Dzhulii [4], Evgeny Lenkov [5], Tarhonskyi Vitalii[6]

[1] *Military Institute of Taras Shevchenko National University, Lomonosova, 81, 03189, Ukraine*
[2-4] *Khmelnytskyi National University, Instytutska str. 11, Khmelnytskyi, 19016, Ukraine*
[5,6] *Central Research Institute of the Armed Forces of Ukraine, Povitroflotsky ave. 28, 03049, Ukraine*

Abstract

Cryptography and cryptanalysis are constantly competing with each other. This stimulates the constant mutual development of these industries. Modern cryptography and methods of cryptographic protection of information are the subjects of many scientific studies, the purpose of which is to obtain practical results that would improve cryptographic security. One of the promising areas of modern cryptography is nonlinear cryptography. The article describes the main advantages of using non-linear cryptographic primitives, their types, and their main characteristics. Variants of modification of non-linear cryptographic primitives are also proposed and various areas of their application are considered. A cryptographic protocol based on a non-linear cryptographic primitive was developed and its software prototype was created.

Keywords

nonlinear cryptography, cryptographic primitive, cryptology, cipher, cybersecurity.

International Scientific and Practical Forum
«Digital Reality» 2023: The advantages of nonlinear cryptographic primitives and their areas of use
March 1-2, 2023, Odesa - Kharkiv, Ukraine
EMAIL: lenkov_s@ukr.net (A.1); muliariv@khmnu.edu.ua (A.2); vladimiranikin01@gmail.com (A.3); dzhuliivm@khmnu.edu.ua (A.4); torwer007@gmail.com (A.5); V_tarhonskiy@ukr.net (A.6)
ORCID: 0000-0001-7689-239X (A. 1); 0000-0002-6659-605X (A. 2); (A. 2); 0000-0003-3395-2764 (A. 3); 0000-0003-1878-4301 (A. 4); 0000-0001-5819-2656 (A. 5); 0000-0001-1313-7666 (A. 6)

1. Introduction

Modern cryptography is an important component of cyber security and information security today.

This is one of the oldest sciences, which in the modern world is classified as computer science. Since ancient times, people have invented various methods of cryptographic transformation of data in order to protect their secrets and other important information. To date, cryptography has changed significantly since ancient times, having acquired a significant mathematical basis in the course of evolution.

Today, cryptography has an extremely wide range of applications, starting from the classic protection of information during storage or transmission, and ending with validation of the integrity of messages and digital signatures, for data verification [1].

Another area that developed in parallel with cryptography was steganography. People tried to hide confidential information, disguising it as ordinary things that do not arouse interest in outsiders.

At the same time when cryptography was born, the opposite field of activity appeared, which was called cryptanalysis. Modern cryptanalysis has sufficiently effective tools for cracking various cryptosystems. In particular, it uses advanced information and computer technologies to research encryption systems [1, 2].

The rapid development of cryptanalysis requires changes in cryptography because they are interconnected. On the other hand, significant innovations in cryptography cause the intensification of research in the field of cryptanalysis. It is an ongoing process that never stops.

It is because of this that the improvement of existing and creation of new methods of cryptographic protection of information, which would be resistant to the tools of modern cryptanalysis, is an actual task now [1-5].

One of the possible ways to increase the cryptographic strength of encryption systems is to ensure the nonlinearity of their functioning, which greatly complicates their analysis. The nonlinearity of the cryptosystem is achieved due to the nonlinear cryptographic primitives in their composition.

We will consider in more detail nonlinear cryptographic primitives, methods of their creation, and properties in this article.

1.1. Review of literary sources. Methods

Modern cryptography is the subject of many studies, scientific articles, and publications.

Some of them consider classical cryptography and historical context, ways of evolution of cryptology, and related fields [1-3].

Much of the research is devoted to modern symmetric cryptosystems, in particular block ciphers such as DES, AES, IDEA, etc [2-4, 6].

Among the latest trends, one can notice many scientific publications on the topic of various blockchain systems that use cryptographic primitives as the basis of their protocols [7, 8]. Also, a lot of attention is paid to the public key of cryptography and the possibilities of its development [9, 10].

All this shows the diversity and multi-vector nature of modern cryptography.

In addition, I analyzed articles on the topic of nonlinear cryptography, which are directly related to the topic of this article [11-14].

Analysis and synthesis of literary sources, comparisons, including on the basis of own publications on related topics, were carried out [11, 13, 14].

The characteristics of nonlinear cryptosystems were tested experimentally. Software prototypes have been created [13, 14].

1.2. Formulation of the problem

The task of this article is to implement a prototype cryptographic protocol with a nonlinearity factor.

To do this, we will rely on the theoretical background covered in this article, as well as on previous publications on related topics [13, 14].

At the same time, the software prototype of the nonlinear cryptographic protocol will be based on reliable certified encryption algorithms and will not change the logic of their operation.

The basis of the cryptographic protocol will be the scheme of a non-linear cryptographic

primitive, and the final result is intended to visually demonstrate all its characteristics.

2. The concept of nonlinear cryptographic primitives

2.1. Modifier in nonlinear cryptographic primitives

To study the concept of a nonlinear cryptographic primitive, a working terminology should be defined.

A cryptographic primitive is a defined low-level cryptographic algorithm. In the frames of this article, we will talk about cryptographic primitives in the context of symmetric key cryptography, omitting another field of application of this term.

As a linear cryptographic primitive, we will consider a cryptographic algorithm of the form $C' = f(C, K)$, in which the original (encrypted) result C' depends on the input (open) data C and cryptographic key K. This is shown in Figure 1.

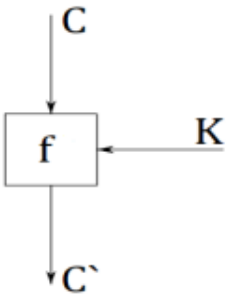


Figure 1: Linear cryptographic primitive

The linearity, in this case, consists of the fact that for the same values of C and K, the value of C' will always be the same.

In contrast, by the concept of a nonlinear cryptographic primitive, we will consider a cryptographic algorithm of the form $C' = f(C, K, M)$, where, in addition to the cryptographic key and open data, the encrypted result will be influenced by some modifier M. Due to this, the nonlinearity of encryption is achieved since when using the specified approach, the direct linear dependence between the cryptographic key and the ciphertext is lost, even if the input data is equal. This is shown in Figure 2.

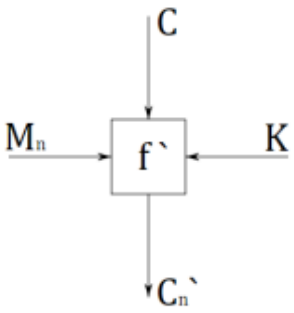


Figure 2: Nonlinear cryptographic primitive

The main feature of the functioning of nonlinear cryptographic primitives is the variability of the ciphertext, which is achieved due to various modifiers. We have the opportunity to "manage" the ciphertext, using various modifiers, and choose the option in which there is a need [13, 14]. This in particular opens up the steganographic scope of this concept, since under ideal conditions we can even reproduce some third-party data as ciphertext by selecting the necessary modifiers.

Of course, not every cryptosystem will provide an opportunity to do this, and the efficiency of such encryption may be small. However, even the theoretical possibility of such use is interesting for further study.

From another point of view, the use of random or pseudo-random modifiers in the proposed scheme enables "random encryption". That is, the encrypted message will be different for the same cryptographic key and input data.

This approach is similar to the common practice today of filling input data with a random cryptographic salt, to complicate cryptanalysis and statistical studies of encrypted messages. The use of nonlinear cryptographic primitives based on random modifiers makes it possible to qualitatively eliminate the need to use cryptographic salt since such randomization will be present by default in these algorithms.

The proposed scheme also complements well the paradigm proposed by J. Nash, an American mathematician.

According to this, the complexity of calculations for successful cryptanalysis is exponentially dependent on the length of the key, and an increase in the number of key parameters has a positive effect on crypto resistance as a whole.

The modifier in the scheme of the nonlinear cryptographic primitive can also be considered as an additional parameter of the key. Also, a

cryptographic key can contain a seed for deploying a pseudo-random sequence from which modifiers will be chosen according to some regularity.

Thus, we can distinguish three main ways of forming the modifier M in a nonlinear cryptographic primitive:

- modifier generation is absolutely random (or pseudo-random, without any obvious patterns);
- generating a modifier based on a crypto-resistant pseudo-random sequence expanded based on a seed that is a key parameter or based on the key itself;
- generation of a functional modifier based on any iteration data in the encryption process (in this case, the "random encryption" property will be missing);
- other ways to get the modifier.

At the same time, the choice of the necessary method of getting the modifier depends on the tasks standing next to the cryptosystem and can be specified and changed at the level of the standard or cryptographic protocol. In some cases, random generation can be used to improve cryptographically strong, in other cases, a more flexible selection of required modifiers for specific tasks can be used.

Also, the idea of using a nonlinear cryptographic primitive with a completely random modifier in combination with a synchronous crypto-resistant generator of pseudo-random numbers, similar to that used in two-factor authentication protocols, seems extremely interesting and promising.

The modifier is one of the most important components of a non-linear cryptographic primitive. The method of getting and using it determines the characteristics of the cryptosystem.

2.2. The functional part of the nonlinear cryptographic primitive

We have dealt with the meaning and properties of the modifier, but the characteristics of the functional part of the nonlinear cryptographic primitive are no less important.

Of course, we can design and implement some symmetric-key cryptosystem, which would immediately have a modifier of some kind as an input parameter, together with a cryptographic key and input data. And such work has already been done [13, 14].

However, this is a difficult path that requires a huge amount of time and resources, because the development of a new cryptosystem is an extremely difficult and responsible process. In addition to the fact that the created cryptosystem must be reliable and cryptographically strong, it must also withstand a large number of studies and certifications before practical use. And this is definitely not the path of those who want to simply and quickly ensure information security, although such a development may have good prospects in the future.

In the case when designing a new separate non-linear cryptosystem is impractical, we can use other tools to getting a nonlinear cryptographic primitive, or a cryptographic protocol with a factor of nonlinearity. For this, we will use ready-made certified symmetric-key cryptosystems, which are recognized as cryptographically strong and widely distributed. It can be AES, DES, IDEA, or any other linear symmetric cryptosystem of your choice.

Let's consider simple ways to transform them into non-linear cryptographic primitives without changing their characteristics:

- modification of the cryptographic key, based on the modifier;
- modification of input data blocks based on the modifier;
- modification of the functional parts of the cryptographic algorithm based on the modifier.

Modifying a cryptographic key is one of the easiest and most secure ways to convert a linear cryptographic primitive into a non-linear one.

This method provides the ability to modify the cryptographic key based on the modifier, using a simple bitwise shift, bitwise permutation, XOR cipher, hashing, or anything else. This method is safe because the primary cryptosystem is not changed in any way, only the cryptographic key is modified, which in its modified form is used as a standard parameter of the cryptosystem.

It is possible to change both the general cryptographic key and the keys on individual iterations. So, for example, the modifier can affect the key differently for each even and odd pair of input blocks or take into account some parameters of previously encrypted blocks, similar to the CBC mode.

The only caveat to using this method to impart nonlinearity to a cryptographic primitive is that some symmetric encryption algorithms have arrays of weak keys that should be avoided.

Another method of creating nonlinearity is to modify the input data based on a modifier.

It is also completely secure, as it does not change the original encryption algorithm in any way, but only changes the format of the input data.

In general, this method is similar to the previous one. We can perform bitwise shifting, permutation, different modes of splitting into blocks, depending on the modifier, etc.

The combination of the two previous methods seems to be extremely effective.

Moreover, the essence of the manipulations should not be secret, as this would somehow contradict Kerckhoffs's principle. The complicating factor in the above examples is primarily a large number of possible variations of the cryptosystem, which depend on the modifier, and not their secrecy.

The last mentioned method of creating nonlinear cryptographic primitives consists of the analysis of encryption algorithms and the selection of modules that can be replaced by analogs without changing the basic logic of work.

This method is much more complicated than the previous ones and more specific. It is strongly not recommended to interfere with the operation of the cryptosystem without having the proper knowledge and principles of its operation, as this can cause serious vulnerabilities and failures.

The essence of this method can be explained most simply on the example of the Feistel network, or on any algorithms derived from it, in particular DES. In the structure of these cryptosystems, there is some cryptographic function with certain requirements. We can create some set of similar functions, with minimal differences in operation, that will return different values at the output, Figure 3.

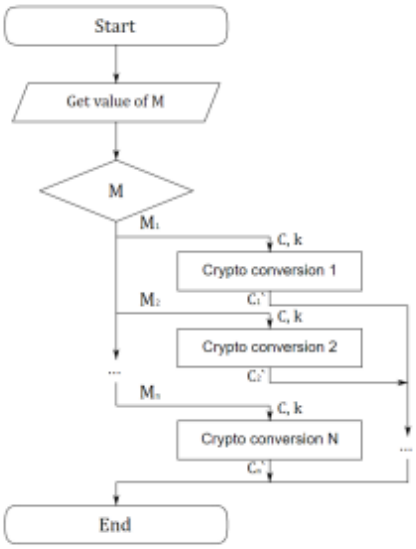


Figure 3: Nonlinear cryptographic algorithm

Having defined a certain number of these functions, based on the modifier, we will determine which of them will be used in a particular iteration.

In various symmetric encryption algorithms, similar functional blocks can be identified, which can be replaced by analogs.

However, one should be extremely careful with this method, as the modified cryptosystem requires serious research for new vulnerabilities.

Therefore, the described methods of nonlinear modification of linear cryptographic primitives make it possible to create cryptographic protocols with a factor of nonlinearity with a high level of cryptographic strong and based on secure certified cryptosystems.

2.3. Practical development of a software prototype of a cryptographic protocol

We will develop a software implementation of a cryptographic protocol based on a nonlinear cryptographic primitive.

One of the possible ways, as described earlier, is to create a new cryptosystem, which by default will have the modifier as one of the required parameters.

An example of such an algorithm was developed and implemented in software [13, 14].

It is based on the simple principle of bit permutation, while the pattern of this permutation depends on the received modifier. It

is quite simple in its structure, while it is quite reliable in operation, Figure 4.

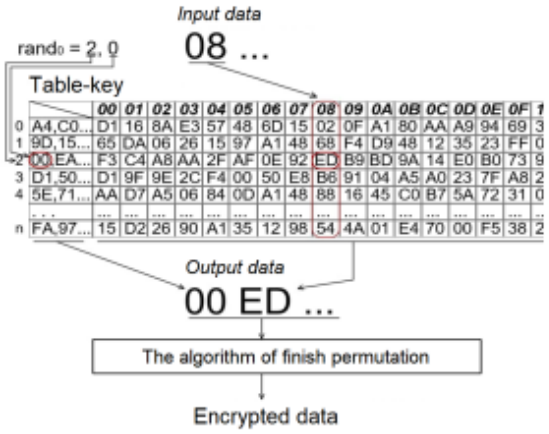


Figure 4: Algorithm of nonlinear encryption

The problem of transmitting the generated modifiers in this algorithm is solved by the fact that these modifiers are combined with encrypted data. Their relation to the key is random, so it is difficult for a cryptanalyst to compare them. In addition, a large number of modifier mappings are used for each pattern, which further complicates cryptanalysis. The algorithm of finish permutation blurs modifiers and encrypted data, which makes it possible to further complicate the analysis of encrypted messages.

The advantages of this algorithm are high speed and resistance to mathematical analysis provided that the random number generator is cryptographically strong.

Among the disadvantages are the large size of the key and the expansion of encrypted data compared to open data.

An interesting property of this algorithm is the possibility of its steganographic use, which was mentioned earlier.

Its simplicity allows us to generate such a cryptographic key, encrypting some text allows us to obtain another text.

That is, we can generate a standard cryptographic key for this algorithm, which can be used in the usual mode of operation for data encryption and decryption. But at the same time, after deciphering some non-secret public text or data with the help of this cryptographic key, we will receive other secret data.

At the same time, the only problem will be the algorithm of finish permutation, which must be adapted for steganographic use.

The principle of operation of this is shown in Figure 5.

Figure 5: Steganographic algorithm model

However, as we have already noted above, this is a difficult path that requires spending significant resources and time.

In addition, symmetric cryptosystems created in this way cannot be recommended for practical use without proper checks and certification. Therefore, we will use another method.

We will use a ready-made proven encryption algorithm and create a modified cryptographic protocol based on it. For example, we can take the DES cryptosystem or any of its modifications. Based on our theory, we will consider it a linear cryptographic primitive.

We can use one of the previously listed modification methods to turn it into a nonlinear cryptographic algorithm. One of the proposed options for modifying the linear cryptographic primitive was to replace the functional parts of the algorithm with analogs, without fundamentally changing the logic.

This can be easily done in the DES cryptosystem since it is based on the Feistel network and has a function in its structure, the modification of which will not cause problems.

For example, we can change the initial permutation pattern without changing its overall logic, or we can change the block expansion format. We can also add a start or end bit offset, block of data, or round`s cryptographic key.

In addition, we can add new intermediate operations, such as an initial bit shift by some value, based on a modifier or parameter of the cryptographic key.

The changes in the function will not be significant, but they will change the data at the output, Figure 6.

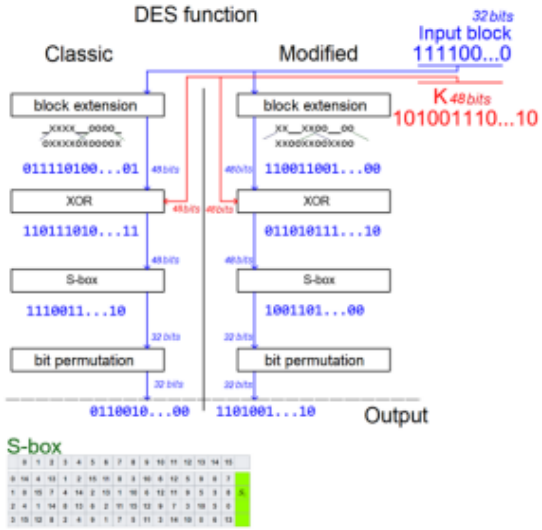


Figure 6: Modification of the DES function

In this way, we can create a large number of analogs of the cryptographic function. However, they will all have a single format and logic of operation. The very structure of the DES algorithm will also remain unchanged. The only change will be that the cryptographic function will be chosen based on a modifier from a set of analogs.

We will get the modifier from a synchronous cryptographically strong random number generator. The modifier will be separate for each pair of blocks, Figure 7.

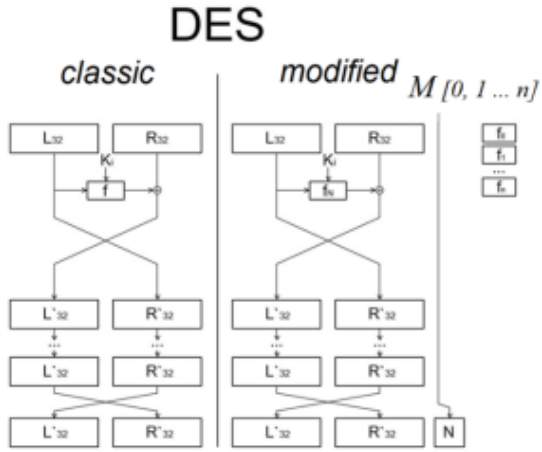


Figure 7: Modified nonlinear DES

In this way, we will get a non-linear modification of the DES algorithm, which on the one hand will be reliable and cryptographically strong, and on the other hand, it will have all the advantages of a non-linear cryptosystem. In particular, it will output random encrypted data, despite the fact that we did not use a cryptographic salt.

Another way of modifying this algorithm is also possible. For example, we can refuse to generate random modifiers every time, and take some hash sum of the previous block as the modifier's value. To increase cryptographic stability, this hash sum can mutate based on some additional key parameter.

In this case, we will not get a "random ciphertext", but thanks to the nonlinearity of the encryption, we will increase the cryptographic strength of the cryptosystem, in particular, reduce its vulnerability to crypto attacks based on statistical studies of the ciphertext. And in the case of using an additional key parameter, we will also increase resistance to brute force attacks. Moreover, this method of increasing the size of the key is also more effective than simple linear key expansion.

We can also establish a random correspondence between the value of a modifier and the cryptographic function it points to by generating arbitrary relations and writing them as cryptographic key parameters.

This method of generating modifiers is well suited in the case of building a cryptographic protocol, in which there is a need to compare and validate encrypted data without decrypting it. In particular, it can be useful when designing cryptographic hash functions based on non-linear elements.

However, in our case, we will use the random generation of modifiers, since our goal is to complicate the analysis and comparison of encrypted data.

The crypto-resistant random number generator in this cryptographic protocol will be used from ready-made solutions from open sources.

The software implementation of the cryptographic protocol prototype will be done in the C# programming language with .NET 4.7 framework.

The graphic part will be implemented using the Windows Forms framework. The software prototype has a bandwidth of about 3 Mb/s and works with any input data without any bugs.

Thus, in practice, we developed a cryptographic protocol based on a nonlinear cryptographic primitive and created a software prototype, Figure 8.

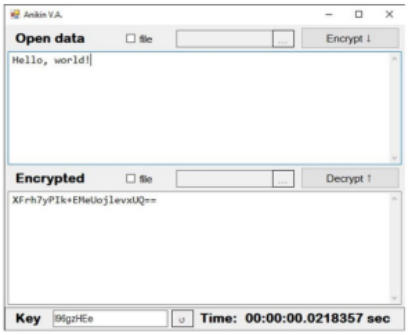


Figure 8: Software implementation of a nonlinear cryptographic protocol

There is many possible modifications of this protocol, depending on the tasks.

The described modification can also function without problems in triple-DES mode.

Similarly, non-linear modifications of any other cryptosystems, including AES, IDEA, Blowfish, and others, can be created.

Such modifications can be practically used in information protection systems, as they are based on proven cryptosystems, effectively complementing them.

The use of non-linear cryptographic primitives can be used to solve specific tasks, as demonstrated by the example of steganographic modification.

3. Conclusions

Nonlinearity is one of the important concepts of all modern science, in particular, it also applies to modern cryptography. Nonlinearity in cryptography makes it possible to significantly complicate the analysis by a cryptanalyst, spending a minimum of resources for this.

The advantages of using non-linear cryptographic primitives, their features, and their characteristics were highlighted in this article.

The modifier generation methods described above and the possibilities of their use can help in the development of cryptographic protocols with a nonlinearity factor and provide a high level of security, and cryptographic strong, and provide a number of unique characteristics that complicate the analysis of encrypted data.

Further research in the field of nonlinear cryptography is quite promising from a practical point of view since cryptosystems designed or modified on the basis of the described schemes can significantly complicate their cryptanalysis.

The specific properties of nonlinear cryptographic primitives allow expanding the scope of their use. In particular, they have steganographic capabilities.

We developed a cryptographic protocol based on a non-linear modification of the DES algorithm and created its software implementation. The created cryptographic protocol is only one of the possible construction options based on non-linear cryptographic primitives and uses the generation of random modifiers in its work.

The designed non-linear modification of the algorithm protects it from a number of popular attacks and greatly complicates the analysis of encrypted messages and statistical research.

The software prototype, having a bandwidth of about 3 Mb/s, is ready for use in information protection systems.

4. References

[1] Bellare, Mihir, and Phillip Rogaway. "Introduction to modern cryptography." Ucsd Cse 207 (2005): 207.

[2] O.H. Korchenko, V.P. Sidenko, Yu.O. Dreis, Prykladna kryptolohiia: systemy shyfruvannia : pidruchnyk, DUT, Kyiv, K, 2014.

[3] Lenkov, S., Kubyavka, M., Kubiavka, L., Lenkov, Y., Shevchuk, V. Reflex Intellectual text processing systems: Natural language text addressing (2019) CEUR Workshor Proceedings, ISSN: 16130073 . - 2386, pp. 85-95.

[4] S.V. Lenkov, D.A. Perehudov, V.A. Khoroshko, Metody i sredstva zashity informacii, Aryi, Kyev, K, 2008.

[5] V.L. Buriachok, S.V. Toliupa, V.V. Semko, L.V. Buriachok, P.M. Skladannyi, N.V. Lukova-Chuiko Informatsiinyi ta kiberprostory: problemy bezpeky, metody ta zasoby borotby, DUT, Kyiv, K, 2016

[6] Qadir, Abdalbasit Mohammed, and Nurhayat Varol. "A review paper on cryptography." 2019 7th international symposium on digital forensics and security (ISDFS). IEEE, 2019.

[7] Gadekallu, Thippa Reddy, et al. "Blockchain for the metaverse: A review." arXiv preprint arXiv:2203.09738 (2022).

[8] Safe Decentralized Applications Development Using Blockchain Technologies / Viktor Cheshun, Ihor Muliar, Vasyl Yatskiv, Ruslan Shevchuk, Serhii Kulyna, Taras Tsavolyk // 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), 16-18 Sept. 2020, Deggendorf, Germany. – Publisher: IEEE, 2020. – P. 800-805.

[9] Imam, Raza, et al. "Systematic and critical review of rsa based public key cryptographic schemes: Past and present status." IEEE Access 9 (2021): 155949-155976.

[10] Method of Multi-Bit Numbers Multiplication in Residue Number System for Asymmetric Cryptosystems Kasianchuk, M., Yakymenko, I., Yatskiv, V., Karpinski, M., Yatskiv, S. CEUR Workshop Proceedingsthis link is disabled, 2022, 3156, pp. 365–377

[11] Gao, Wei, et al. "Construction of nonlinear component of block cipher by action of modular group PSL (2, Z) on projective line PL (GF (2 8))." IEEE Access 8 (2020): 136736-136749.

[12] Tang, Deng, Claude Carlet, and Xiaohu Tang. "Highly nonlinear Boolean functions with optimal algebraic immunity and good behavior against fast algebraic attacks." IEEE transactions on information theory 59.1 (2012): 653-664.

[13] V. A. Anikin, V. M. Dzhulii, I.V. Muliar, V.S. Orlenko, V.Iu. Titova Symetrychna kryptosystema z neliniinym shyfruvanniam ta mozhlyvistiu kontroliu shyfrotekstu z metoiu maskuvannia , (2020) 12-19.

[14] Anikin V.A. Symetrychnyi alhorytm neliniinoho shyfruvannia z mozhlyvistiu stehanohrafichnoho zastosuvannia: naukova robota studenta, Khmelnytskyi, KhNU, 2021.

к.т.н. Бойчук В.О. (ХмНУ)
Анікін В.А. (ХмНУ)

Симетрична поліалфавітна криптосистема на основі випадкової генерації коефіцієнту перестановки

Криптографія - це прикладна наука, яка розробляє і впроваджує системи захисту інформації шляхом перетворення вихідних осмислених повідомлень в зашифровані повідомлення, які неможливо розшифрувати без ключа дешифрування, хоча вони уразливі для криптоаналітичних атак.

В будь-які часи люди намагалися захистити різноманітну важливу інформацію, ховаючи, маскуючи, або видозмінюючи її. Класична криптографія грунтується на припущенні, що ніхто не може вирішити певну складну задачку за реалістичний проміжок часу, або покладатися на аргументи теорії інформації.

Пропонується метод шифрування оснований на принципі поліалфавітної байтової перестановки, де коефіцієнт перестановки для кожного байта обирається випадково. Дешифрування можливе завдяки присвоєнню унікальних ідентифікаторів для кожного байту. Кожен алфавіт може мати безліч ідентифікаторів, проте будь-який ідентифікатор може належати лише одному алфавіту заміни.

Криптографічним ключем у даній криптосистемі є таблиця з довільною кількістю випадково перемішаних відкритих алфавітів (де відкритий алфавіт – це послідовність байт від 0 до 256), кожному з яких належить набір ідентифікаторів.

Алгоритм шифрування складається з наступних етапів: представлення відкритої інформації у байтовому вигляді, генерація випадкового коефіцієнту перестановки (номеру алфавіту та ідентифікатора) для кожного байту, перестановка відкритих байт з байтами випадково обраних алфавітів, формування вихідного шифротексту у форматі «ідентифікатор-байт», кінцеве «забілення».

Алгоритм дешифрування є протилежним до алгоритму шифрування.

Таким чином досягається висока ентропія зашифрованої інформації, а випадковість та відсутність закономірностей у перестановках унеможливлює та серйозно ускладнює різні види криптоаналізу. Надійність даного алгоритма напряму залежить від стійкості генератора випадкових чисел.

Список використаних джерел:

1. Jara Vera V, Sánchez Ávila C (2019) Graphemic-phonetic diachronic linguistic invariance of the frequency and of the Index of Coincidence as cryptanalytic tools. PLoS ONE 14(3): e0213710 10.1371/journal.pone.0213710.

Симетричний алгоритм нелінійного шифрування з можливістю стеганографічного застосування

Анікін В.А.
Науковий керівник – к.т.н.,доц. Муляр І.В.
Хмельницький національний університет

На сьогоднішній день захист інформації є одним з найбільш актуальних напрямків кібербезпеки, враховуючи стрімкий ріст інформаційних технологій та збільшення цінності інформації як ресурсу. Вже давно як рядові користувачі, так і великі корпорації діджиталізують особисті дані, серед яких велика кількість персональної, конфіденційної інформації, даних що становлять корпоративну, лікарську, чи навіть державну таємницю.

Більше того, дана інформація пересилається через публічні мережі, зберігається на онлайн ресурсах, що окрім комфорту та зручності створює серйозні безпекові загрози.

Для усунення, чи мінімізації цих загроз, для збереження цілісності, доступності та конфіденційності – трьох базових аспектів захисту інформації, впродовж багатьох століть люди використовують шифрування, яке на сьогоднішній день вивчає наука під назвою криптографія.

Криптографія – це прикладна наука, яка розробляє і впроваджує системи захисту інформації шляхом перетворення вихідних осмислених повідомлень в зашифровані повідомлення, які неможливо, або вкрай важко розшифрувати без криптографічного ключа, хоча вони й вразливі для криптоаналітичних атак [1].

В будь-які часи люди намагалися захистити різноманітну важливу інформацію, ховаючи, маскуючи, або видозмінюючи її. Класична криптографія ґрунтується на припущенні, що ніхто не може вирішити певну складну задачку за реалістичний проміжок часу, або покладатися на аргументи теорії інформації [2].

Сучасна криптографія є математично-комбінаторною наукою, що вивчає способи перетворення інформації з метою її захисту від несанкціонованого використання.

Викликом перед сучасною криптографією є значний прорив у області комп'ютерних електронно-обчислювальних машин, в силу чого незламні до цього криптосистеми стали вразливими. Через це в сучасній криптографії, починаючи з другої половини минулого століття стійкість криптосистеми оцінюється на основі складності обчислень, необхідних для успішного проведення криптоатаки, з використанням усіх можливих ресурсів. Даний підхід запропонував криптолог Джон Неш і він залишається актуальним до сьогодні [3].

Стеганографія також є невід'ємною складовою сучасного захисту інформації, оскільки криптографічне повідомлення, яке представляє собою неосмислений набір байт, неодмінно приверне до себе увагу, в той час як стеганографічне повідомлення у вигляді зображення, чи будь-яких інших даних, що самі по собі жодної цінності не представляють, з великою ймовірністю залишиться не поміченим.

Комплексні алгоритми захисту, що включають в себе як криптографічний так і стеганографічний аспект здатні забезпечити надійний захист важливої інформації.

На підставі цього пропонується використання нелінійності у криптографічних алгоритмах яка з одного боку забезпечить додаткову складність обчислень, необхідних для криптоаналізу шифрованих повідомлень, а з іншого – дасть можливість отримати множину можливих

шифротекстів, при одному і тому ж вхідному повідомленні та ключі, що дасть підґрунтя для стеганографічного використання даного алгоритму.

Пропонується метод шифрування оснований на принципі поліалфавітної блочної підстановки, де система (алфавіт) підстановки для кожного блоку обирається на основі випадкових значень, що генеруються в процесі шифрування. Дешифрування можливе завдяки присвоєнню унікальних ідентифікаторів для кожної підстановочної системи. Кожен алфавіт може мати безліч ідентифікаторів, проте будь-який ідентифікатор може належати лише одному алфавіту заміни.

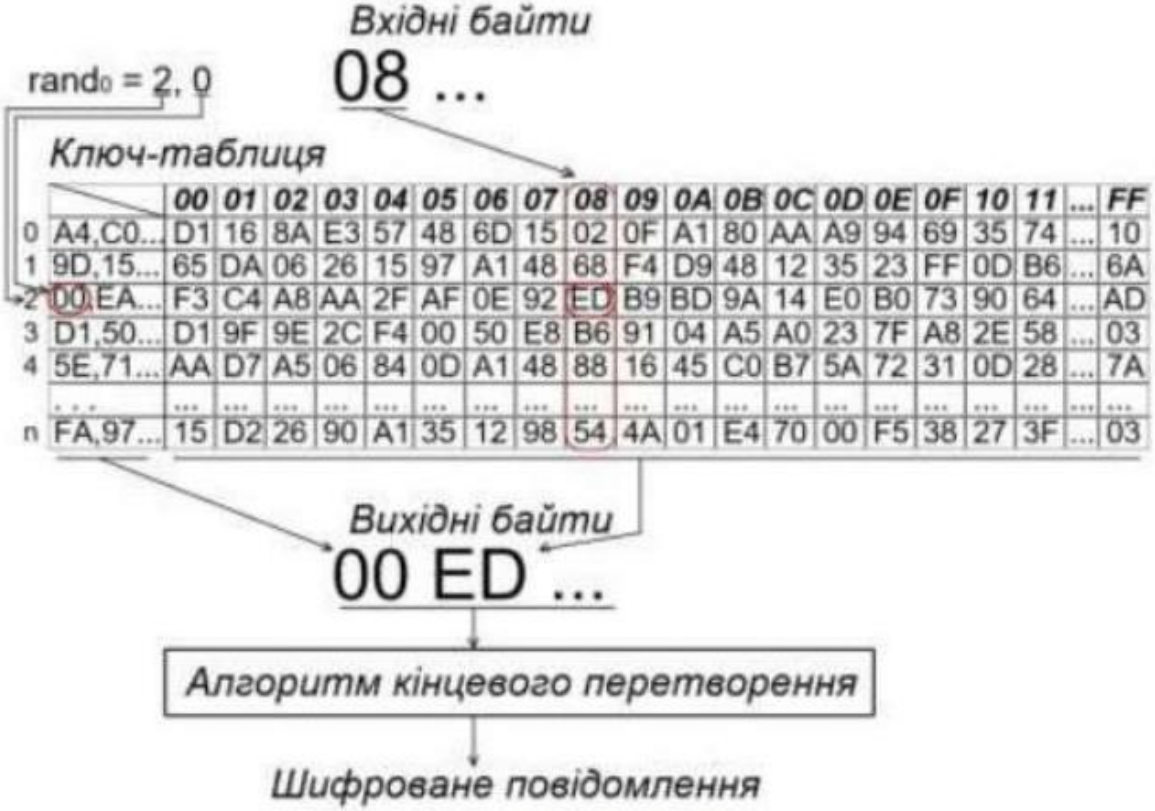
Логікою шифрування в даному алгоритмі буде заміна одного байту на інший, відповідно до деякої поліалфавітної таблиці перестановок. Дана таблиця міститиме в собі n рядків та 256 стовпців, якщо за умовний блок даних ми візьмемо один байт. В кожному рядку буде послідовність від 0 до 255, перемішана випадковим чином. Кількість рядків, а відповідно «алфавітів перестановки», може бути якою завгодно.

Кожному рядку присвоюється m випадкових двійкових ідентифікаторів. Розрядність цих ідентифікаторів також може бути обрана різна, проте вона повинна бути однаковою для всіх ідентифікаторів. Від обраної розрядності залежатиме, по-перше, кількість алфавітів, яку ми можемо визначити, наприклад якщо розрядність дорівнюватиме двом, то максимум у нас може бути 4 алфавіти, а з урахуванням рекомендації використовувати не менше двох ідентифікаторів на один алфавіт – всього 2. По-друге, від розрядності залежатиме те, наскільки збільшиться шифрований текст, порівняно з вхідним повідомленням. Якщо розрядність ідентифікатора буде рівною розрядності нашого блоку даних, а у нашому випадку – це 8 біт, то шифроване повідомлення буде вдвічі більше за вхідне. Чим менша розрядність ідентифікатора, по відношенню до розміру блоку даних, тим менше буде збільшення вихідного повідомлення при шифруванні і навпаки.

Дана таблиця замін, разом із відповідними ідентифікаторами, складатиме собою секретний ключ. Для її компактного запису можна використовувати різноманітні технології стиснення.

Схема шифрування даним методом показана на рис. 1: спочатку ми обираємо два випадкових числа, з яких перше – в діапазоні $[0; n)$, де n – кількість алфавітів, а друге – в діапазоні $[0; m)$, де m – кількість ідентифікаторів, відповідних даному алфавіту. Після цього ми заміняємо вхідний блок даних на вираз, що складається з обраного ідентифікатора та числа, що знаходиться в обраному алфавіті на позиції, номер якої відповідає числу, утвореного з блока вхідних даних. Тобто, якщо ми шифруємо 8-бітні блоки, і на вході, як приклад, отримали блок «10001010», що у десятковій формі відповідає числу 138, а випадкові числа випали 2 і 0, за умови, що вони відповідатимуть межам зазначених діапазонів, то шифрований вираз для даного блоку складатиметься з 0-го ідентифікатора 2-го алфавіту

перестановок та числа, що знаходиться на 138-й позиції у 2-му алфавіті. Такі перетворення почергово проводяться для кожного блоку даних, до кінця відкритого повідомлення.



	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	FF
0	A4,C0...	D1	16	8A	E3	57	48	6D	15	02	0F	A1	80	AA	A9	94	69	35	74	...	10
1	9D,15...	65	DA	06	26	15	97	A1	48	68	F4	D9	48	12	35	23	FF	0D	B6	...	6A
2	00,EA...	F3	C4	A8	AA	2F	AF	0E	92	ED	B9	BD	9A	14	E0	B0	73	90	64	...	AD
3	D1,50...	D1	9F	9E	2C	F4	00	50	E8	B6	91	04	A5	A0	23	7F	A8	2E	58	...	03
4	5E,71...	AA	D7	A5	06	84	0D	A1	48	88	16	45	C0	B7	5A	72	31	0D	28	...	7A
. . .	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
n	FA,97...	15	D2	26	90	A1	35	12	98	54	4A	01	E4	70	00	F5	38	27	3F	...	03

Рисунок 1 – Схема шифрування

Криптографічним ключем у даній криптосистемі є таблиця з довільною кількістю випадково перемішаних підстановочних алфавітів, кожному з яких належить набір ідентифікаторів.

Алгоритм шифрування складається з наступних етапів: представлення відкритої інформації у бітовому (двійковому) вигляді, генерація випадкових значень (номеру алфавіту та ідентифікатора) для кожного n-бітного вхідного блоку, підстановка вхідних блоків за випадково обраним алфавітом, зчеплення підставленого блоку та ідентифікатора використаної підстановочної системи, кінцеве перетворення з метою «затирання» ідентифікаторів.

Алгоритм дешифрування є протилежним до алгоритму шифрування. Однозначне дешифрування у даному алгоритмі можливе завдяки ідентифікаторам, які однозначно вказують на те який підстановочний алфавіт було використано для конкретного блоку даних.

Відсутність складної ітеративності забезпечує високу швидкість роботи даного алгоритму.

Таким чином досягається висока ентропія зашифрованої інформації, а випадковість та відсутність закономірностей у перестановках унеможливлює та серйозно ускладнює різні види криптоаналізу. Надійність даного алгоритму напряму залежить від стійкості генератора випадкових чисел.

Стеганографічне функціонування алгоритму базується на припущенні: «Якщо з використанням нелінійного шифрування кожен n-бітний блок може бути замінено багатьма способами, в залежності від того яку підстановочну систему було обрано, а самі підстановочні системи генеруються без будь-яких закономірностей, то, фактично, будь-який вхідний блок після зашифрування може бути перетворений на будь-який вихідний блок, при чому ця відповідність є контрольованою».

Інакше кажучи, даний алгоритм дає можливість створити такий ключ, при якому шифротекст матиме деякий конкретний вигляд, а контрольована генерація ключа дасть можливість контролювати вихідний шифротекст.

На основі цього можна припустити, що можливо створити такий ключ, при якому деяке повідомлення, зашифроване одним із можливих способів на виході дасть шифротест, що представляє собою деяке інше осмислене повідомлення.

Перелік посилань

1. Jara Vera V, Sánchez Ávila C (2019) Graphemic-phonetic diachronic linguistic invariance of the frequency and of the Index of Coincidence as cryptanalytic tools. PLoS ONE 14(3): e0213710 10.1371/journal.pone.0213710
2. Cavaliere, Fabio, John Mattsson, and Ben Smeets. «The security implications of quantum cryptography and quantum computing.» Network Security 2020.9 (2020): 9-15.
3. Прикладна криптологія: системи шифрування : підручник / О.Г. Корченко, В.П. Сіденко, Ю.О. Дрейс. – К. : ДУТ, 2014. – 448 с.

ДОДАТОК Б

Презентація кваліфікаційної роботи

Кваліфікаційна робота на тему

Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів

Виконав студент гр. КБм-22-1

Анікін Володимир

Керівник:

к.т.н., доц. кафедри КБ
Муляр І.В.

Мета кваліфікаційної роботи

Створення методу розгортання криптографічного ключа для нелінійних симетричних криптосистем, задля усунення недоліків криптосистем на основі нелінійних криптографічних примітивів.

Об’єкт дослідження

Процес розгортання криптографічних ключів в рамках процесу симетричного шифрування інформації в інформаційно-комунікаційних системах

Предмет дослідження

Методи та алгоритми розгортання криптографічного ключа у криптографічних засобах захисту інформації, зокрема побудованих на основі нелінійних криптографічних примітивів

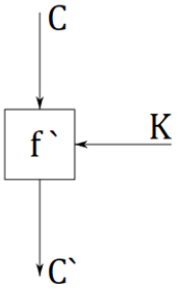
Наукова новизна

1. Вдосконалено математичну модель процесу розгортання криптографічного ключа, адаптовано її до схеми роботи криптосистем на основі нелінійних криптографічних примітивів.
2. Створено метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів

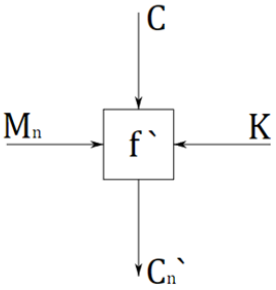
Наукові публікації

- 1 фахова стаття
- 1 стаття Scopus (очікує на індексацію)
- 3 тези доповідей конференцій
- 1 конкурсна студентська наукова робота

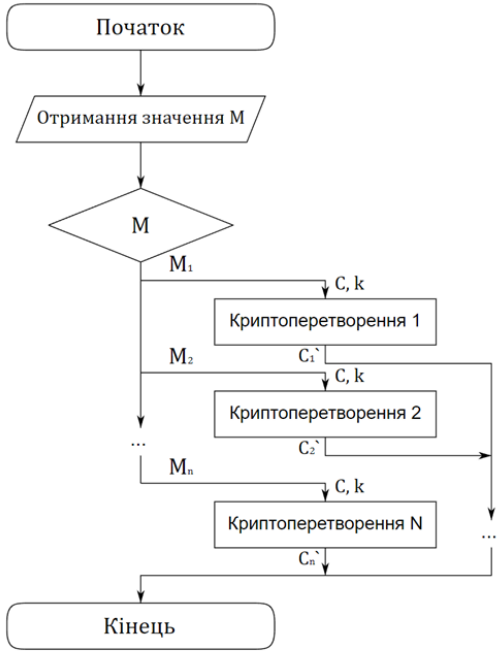
Модель криптосистеми, на основі нелінійних криптографічних примітивів



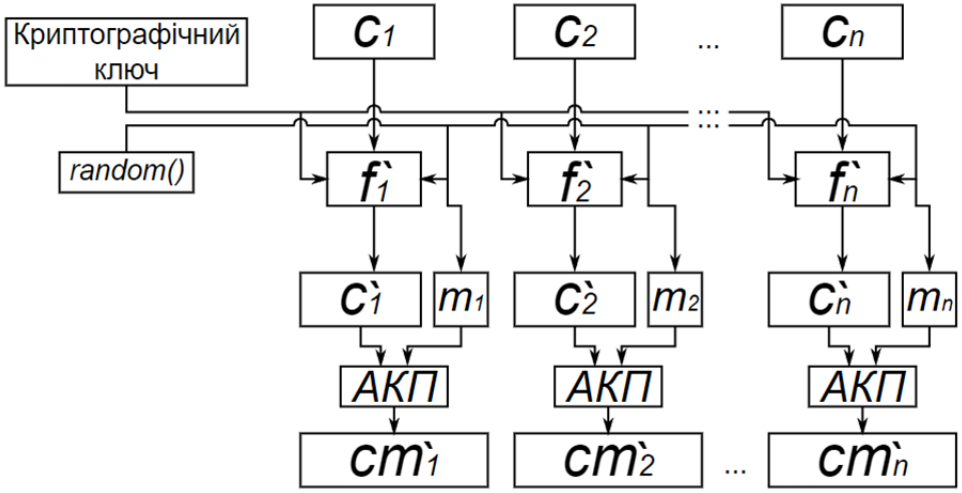
Лінійний криптографічний примітив

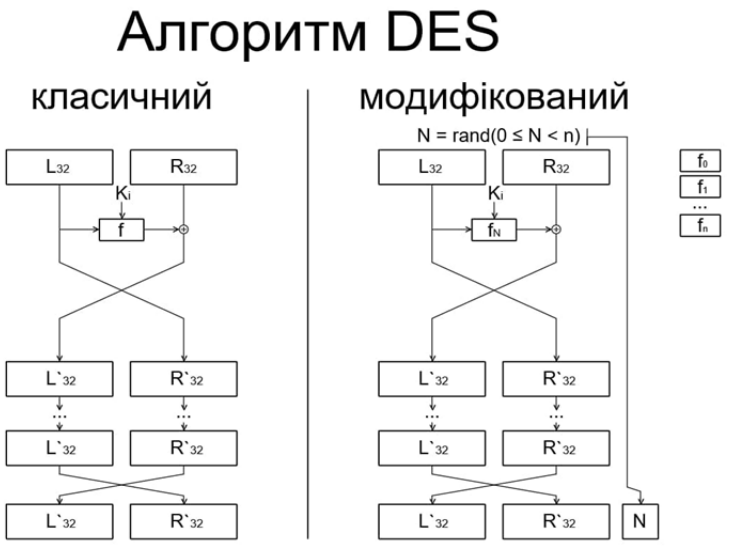


Нелінійний криптографічний примітив

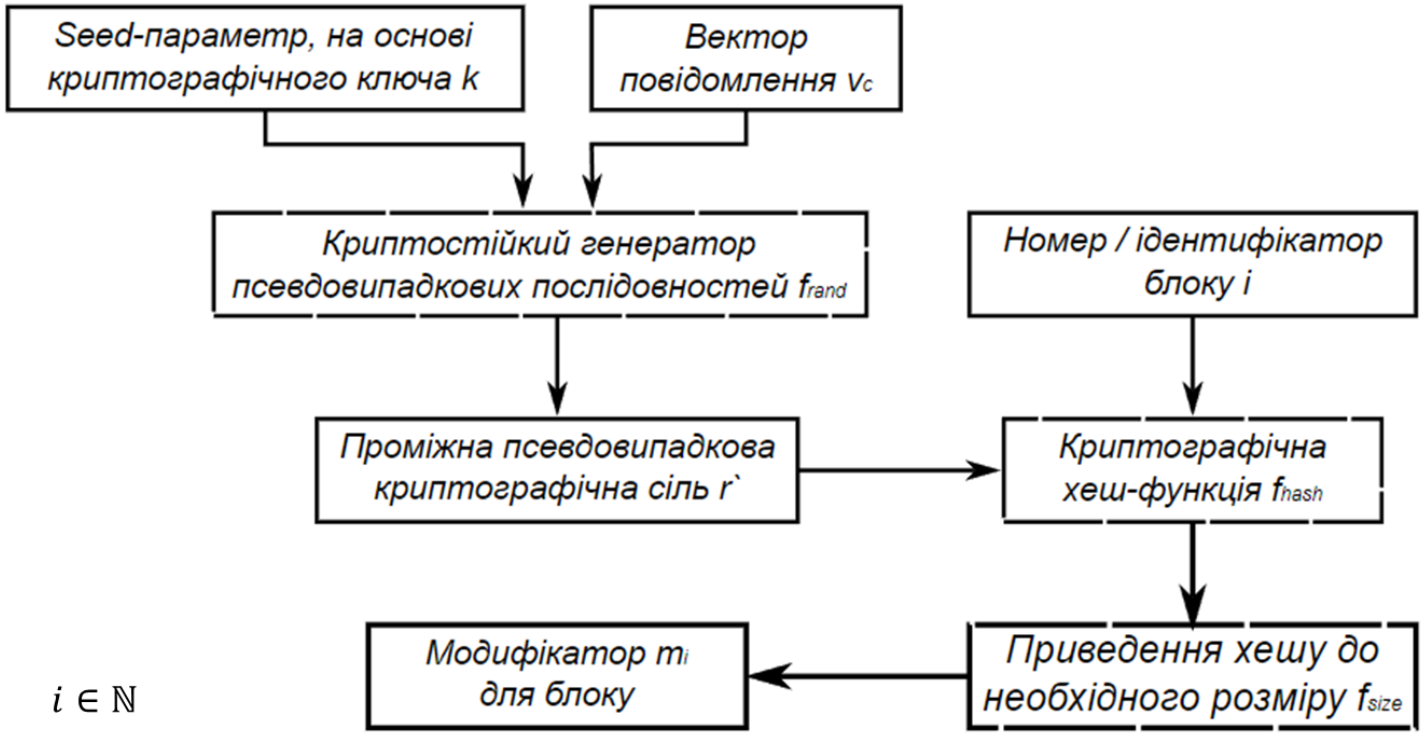


Проблема криптосистем на основі НКП





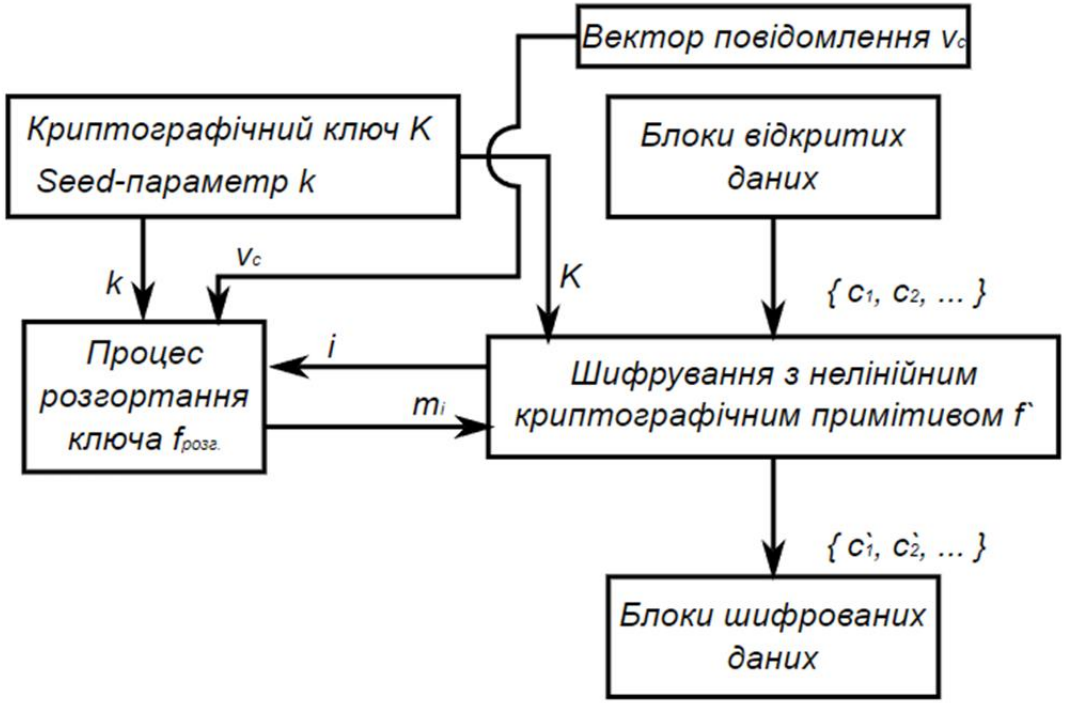
Модель процесу розгортання ключа



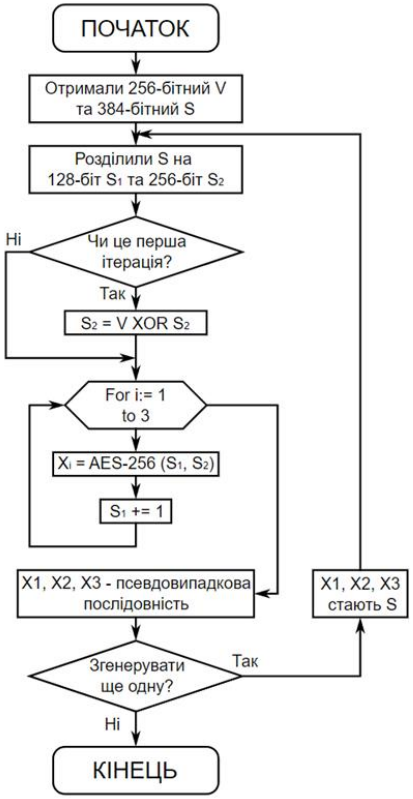
$$r` = f_{rand}(k, v_c)$$

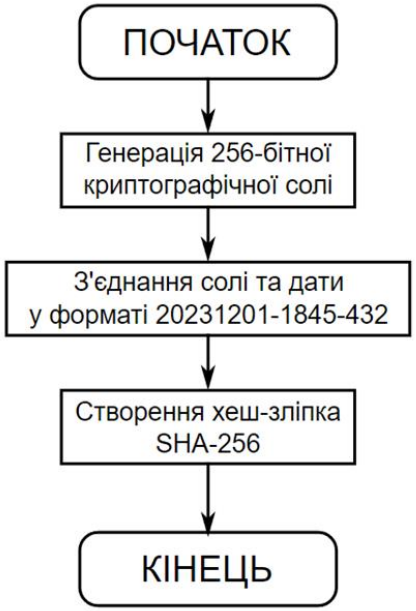
$$m_i = f_{size}(f_{hash}(r`, i)), \qquad i \in \mathbb{N}$$

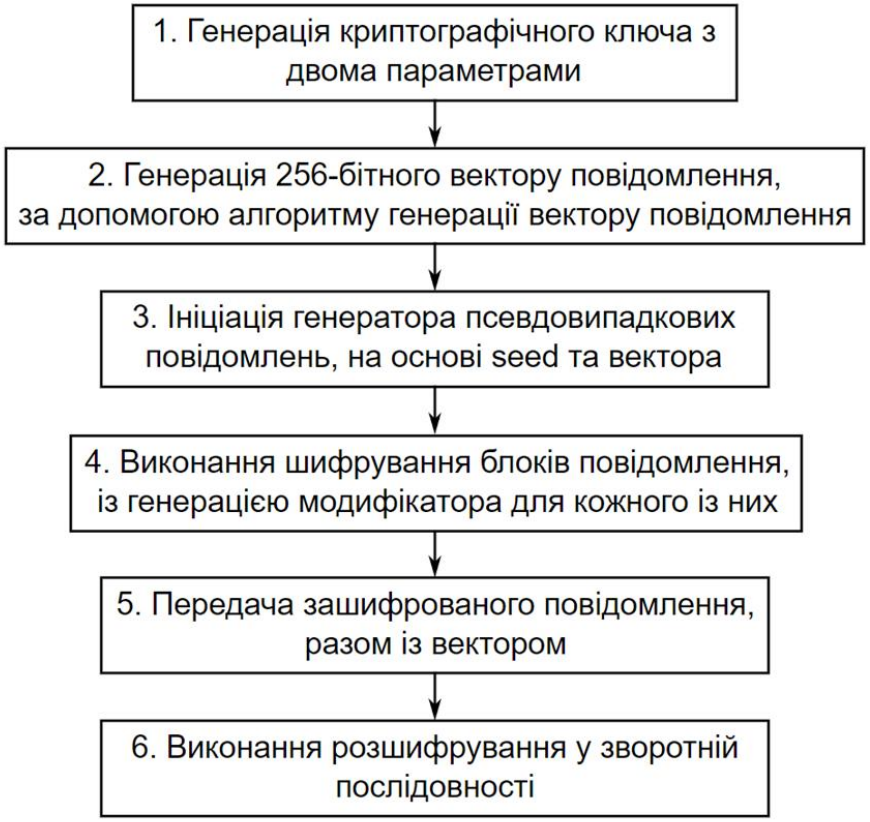
Процес розгортання ключа, як частина процесу шифрування нелінійної криптосистеми



Метод розгортання ключа







Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Анікін В.А.
ПІБ здобувача вищої освіти

Студента ФІТ, 2 курсу, групи КБм-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

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

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

11.12.2023
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 12%**

	Документ		Сумарний збіг по Базі Даних	
ID: 123288 Назва: Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів Додано в БД: 2023-12-14 Автора: Анікін В.А. Керівники: Муляр І.В. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	80193	1188	672 (1%)	17 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016007140

Дата перевірки:
14.12.2023 20:12:01 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
14.12.2023 20:13:50 EET

ID користувача:
100008300

Назва документа: **Анікін**

Кількість сторінок: **73** Кількість слів: **11769** Кількість символів: **93366** Розмір файлу: **3.05 MB** ID файлу: **1015691616**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.84%
Схожість

Найбільша схожість: **1.52%** з джерелом з Бібліотеки (ID файлу: **1011379842**)

1.64% Джерела з Інтернету	128	Сторінка 75
1.83% Джерела з Бібліотеки	137	Сторінка 75

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи	9
Підозріле форматування	11 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів

Автор: Анікін Володимир Андрійович

Науковий керівник: Муляр Ігор Володимирович, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 98,16%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 100%.

Згідно з правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100 %, визнається роботою з високою унікальністю тексту і допускається до захисту.

Керівник роботи І.В. Муляр

Гарант ОП В.Ю. Тітова

Завідувач кафедри кібербезпеки Ю.П. Кльоц

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «магістр»

Магістр: Анікін В.А.

Тема: Метод розгортання криптографічного ключа для використання у криптосистемах на основі нелінійних криптографічних примітивів

Галузь знань: 12 – Інформаційні технології

Спеціальність: 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість сторінок записки 79

1. Короткий зміст кваліфікаційної роботи та прийнятих рішень: в рамках роботи запропоновано метод розгортання криптографічних ключів для використання у криптосистемах на основі нелінійних криптографічних примітивів. Також створені програмні прототипи, із реалізаціями алгоритмів запропонованого методу розгортання криптографічних ключів.

2. Висновок про відповідність кваліфікаційної роботи завданню: Кваліфікаційна робота магістра у повній мірі відповідає поставленому завданню у теоретичній, та практичній частинах.

3. Характеристика виконання виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: Вступ кваліфікаційної роботи висвітлює актуальність теми роботи, декларує мету, цілі і завдання дослідження, описує наукову новизну та практичну цінність результатів дослідження. У першому розділі проводиться дослідження предметної області, розвиток та поточний стан справ сучасної криптології та стеганографії. У другому розділі складається математична модель запропонованого методу, на основі перечислених вимог. Третій розділ містить опис створення методу розгортання криптографічних ключів, шляхом синтезу, модифікації та доповнення існуючих рішень. Четвертий розділ присвячено практичним дослідженням запропонованого методу, апробації його на різних криптосистемах.

4. Позитивні сторони роботи: Кваліфікаційна робота є послідовним продовженням тематики симетричних систем нелінійного шифрування, висвітлених у попередніх публікаціях студента. Дана робота містить ряд рішень, що являють собою наукову новизну, а також вирішують суттєві недоліки криптографічних рішень, наведених у попередніх публікаціях, зокрема поєднання процесу генерації нелінійного модифікатора із процесом генерації криптографічного ключа.

перед початком його практичного використання.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Графічне оформлення виконане якісно та у відповідності до теми кваліфікаційної роботи з дотриманням стандартів. Пояснювальна записка відповідає нормам оформлення текстових документів університету.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота студента заслуговує позитивної оцінки. Матеріал дипломної роботи релевантний, структурований та чіткий. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Наукова новизна та практична цінність роботи є послідовним вирішенням більш глобальних завдань, висвітлених у систематичних публікаціях студента.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи: Представлена кваліфікаційна робота, за сумою позитивних та негативних сторін, науковою новизною, актуальністю та практичною цінністю заслуговує на оцінку «відмінно»

РЕЦЕНЗЕНТ (прізвище, ім`я, по батькові, посада, місце роботи) Завідувач кафедри ТМІТ д.т.н проф. Підченко С.К.

«11» грудня 2023. _____ (підпис)

ХНУ. Зам. 54, тир. 6000, 2007.