Хмельницький національний університет
Факультет інформаційних технологій

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Криптографічна система нелінійного шифрування з можливістю стеганографічного застосування
Назва теми

КРКБ.180125.18.01.01 ПЗ
Шифр

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *125 «Кібербезпека»*
Шифр, назва

Освітня програма *«Кібербезпека»*
Шифр, назва

Виконав: студент 4 курсу, група КБ-18-1 _____ 06 ЧЕР 2022 Володимир АНІКІН
Підпис, дата Ініціали, прізвище

Керівник _____ 06 ЧЕР 2022 Ігор МУЛЯР
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ 03.06.22 Сергій МОСТОВИЙ
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри кібербезпеки _____ Юрій КЛЬОЦ
Підпис, дата Ініціали, прізвище

8 06 2022 р.

Хмельницький, 2022

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормаконтроль	Мостовий С.В.	–	
Антиплагіат	Мостовий С.В.	–	

7. Дата видачі завдання 30 СІЧ 2022

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапу (розділу) кваліфікаційної роботи	Строк виконання етапу роботи	Примітка
1	Вибір та затвердження теми кваліфікаційної роботи.	Січень	–
2	Аналіз об'єкта дослідження.	Січень–лютий	–
3	Проектування та розробка загальної архітектури і структури системи.	Лютий–березень	–
4	Програмна реалізація запропонованого рішення та тестування системи; аналіз результатів і оцінювання прийнятих рішень.	Квітень	–
5	Написання тексту пояснювальної записки та розробка графічних матеріалів.	Травень	–
6	Остаточне коригування кваліфікаційної роботи з урахуванням зауважень керівника.		–
7	Оформлення кваліфікаційної роботи як документа відповідно до вимог.		–
8	Отримання супровідних документів. Нормоконтроль.	Червень	–
9	Підготовка до захисту та захист кваліфікаційної роботи.		–

Студент _____ Володимир АНІКІН
Підпис Ініціали, прізвище

Керівник роботи _____ Ігор МУЛЯР
Підпис Ініціали, прізвище

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормаконтроль	Мостовий С.В.	–	
Антиплагіат	Мостовий С.В.	–	

7. Дата видачі завдання 30 СІЧ 2022

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапу (розділу) кваліфікаційної роботи	Строк виконання етапу роботи	Примітка
1	Вибір та затвердження теми кваліфікаційної роботи.	Січень	–
2	Аналіз об'єкта дослідження.	Січень–лютий	–
3	Проектування та розробка загальної архітектури і структури системи.	Лютий–березень	–
4	Програмна реалізація запропонованого рішення та тестування системи; аналіз результатів і оцінювання прийнятих рішень.	Квітень	–
5	Написання тексту пояснювальної записки та розробка графічних матеріалів.	Травень	–
6	Остаточне коригування кваліфікаційної роботи з урахуванням зауважень керівника.		–
7	Оформлення кваліфікаційної роботи як документа відповідно до вимог.		–
8	Отримання супровідних документів. Нормоконтроль.	Червень	–
9	Підготовка до захисту та захист кваліфікаційної роботи.		–

Студент _____ Володимир АНІКІН
Підпис Ініціали, прізвище

Керівник роботи _____ Ігор МУЛЯР
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Криптографічна система нелінійного шифрування з можливістю стеганографічного застосування».

Автор роботи: Анікін Володимир Андрійович.

Керівник роботи: Муляр Ігор Володимирович.

Пояснювальна записка: 67 с., 30 рис., 1 табл., 2 дод., 20 джерел.

Графічна частина: 5 презентаційних слайдів.

НЕЛІНІЙНЕ ШИФРУВАННЯ, СТЕГАНОГРАФІЯ, КРИПТОГРАФІЧНИЙ ЗАХИСТ, АЛГОРИТМ ШИФРУВАННЯ, DES, КРИПТОГРАФІЯ

Метою кваліфікаційної роботи є створення криптостійкого симетричного нелінійного алгоритму шифрування з можливістю комбінованого криптографічно-стеганографічного використання.

У кваліфікаційній роботі було проведено аналіз існуючих симетричних, досліджено вплив та особливості нелінійного шифрування на крипторграфічні системи, розроблено систему модифікації існуючих криптосистем, з використанням нелінійних криптографічних примітивів.

Також спроектовано та програмно реалізовано окрему нелінійну симетричну криптосистему з можливістю як криптографічного, так і стеганографічного режиму роботи.

Підпис студента

02 ЧЕР 2022

Дата

№ ряда	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1	А4	КРКБ.180125.18.01.01 ПЗ	Пояснювальна записка	67		
			Графічні матеріали			
2	А4	КРКБ.180125.18.01.01 Е8	Актуальні блокові шифри	1		
3	А4	КРКБ.180125.18.01.01 Е8	Лінійні та нелінійні криптопримітиви	1		
4	А4	КРКБ.180125.18.01.01 Е8	Модифікація DES	1		
5	А4	КРКБ.180125.18.01.01 Е8	Алгоритм нелінійного шифрування	1		
6	А4	КРКБ.180125.18.01.01 Е8	Стеганографічна версія алгоритму	1		

КРКБ.180125.18.01.01 ВП

Зм	Арк	№ док	Підпис	Дата
Розробив		Анікін В.А.		
Перевір.		Муляр І.В.		
Н. контр.		Мостовий С.В.		08.06.22
Затв.		Кльоц Ю.П.		8.06.22

Криптографічна система нелінійного шифрування з можливістю стеганографічного застосування
Відомість роботи

Літера	Аркуш	Аркушів
У	1	1

ХНУ, КБ-18-1

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

АГК – Алгоритм генерації ключа

АКП – Алгоритм кінцевого перетворення

ДСТУ – Державний стандарт України

ІТ – Інформаційні технології

ІКС – Інформаційно-комунікаційна система

ОС – Операційна система

КСЗІ – Комплексна система захисту інформації

ПЗ – Програмне забезпечення

ЦВЗ – Цифровий водяний знак

AES – Advanced Encryption Standard

DES – Data Encryption Standard

IDEA – International Data Encryption Algorithm

PES – Proposed Encryption Standard

PKC – Public-key cryptography

ВСТУП

Важливою складовою кібербезпеки є сучасна криптографія. Це одна з найстаріших наук, яку у сучасному світі відносять до наук про інформацію. Від давна люди вигадували різноманітні методи криптографічних перетворень даних, з метою захисту своїх таємниць та іншої важливої інформації.

На сьогоднішній день криптографія суттєво змінилась з прадавніх часів та, отримавши в ході еволюції суттєве математичне підґрунтя, є невід'ємною частиною сучасного світу. Криптографія є основою для незліченної кількості новітніх технологій, охоплюючи практично всю сферу IT: від безпосереднього захисту різного виду комунікацій, у чому її призначення не змінилось від стародавності, до забезпечення недоторканості критичних інформаційних процесів, створення надійних сертифікатів та електронних сигнатур.

Здобутками у даній сфері користуються найрізноманітніші категорії користувачів. Це і військові, правоохоронні державні та інші спеціальні органи, які вже століттями зацікавлені у безпечній внутрішній комунікації, безпечному збереженню внутрішньої інформації з обмеженим доступом та в цілому – в надійному захисті своїх секретів. Також це і комерційні, фінансові, банківські структури, в яких безпека комерційної таємниці, а також можливість гарантованої ідентифікації та захищеності транзакцій, сторони яких можуть знаходитись у протилежних точках земної кулі, є життєво-важливою умовою. І, звичайно, здобутками криптографії користується практично кожен користувач в мережі Інтернет, який відвідує веб-сайти, користуючись протоколом HTTPS, чи іншими захищеними протоколами, авторизується за допомогою логіна та паролю, переписується у месенджерах та соціальних мережах, здійснює покупки в інтернеті тощо.

Саме через це надійність сучасних криптографічних методів захисту є вкрай важливою, а її порушення може спричинити вкрай серйозні, подекуди невідворотні наслідки.

Іншою, проте не менш важливою, складовою сучасного захисту інформації є стеганографія, адже класичне криптографічне повідомлення, яке представляє собою неосмислений набір символів чи байт, достатньо великого розміру, неодмінно приверне до себе надмірну увагу у деяких випадках. В цей час стеганографія дає можливість замаскувати зашифроване, або будь-яке інше важливе повідомлення, під медіа-файл, чи інші дані, що самі по собі жодної цінності не становлять.

Наведені приклади дозволяють нам зробити висновки про високу актуальність досліджень у сфері підвищення надійності та захищеності криптографічних систем, а також досліджень маскувальних стеганографічних алгоритмів.

Криптографія є математично-комбінаторною наукою, яка вивчає способи перетворення даних з метою їх захисту від використання сторонніми особами та з можливістю зворотної дешифровки перетвореної інформації назад у початковий вигляд, зі збереженням ідентичності та цілісності вхідної та вихідної інформації.

Проблемою сучасної криптографії є значний прорив у сфері електронних обчислень, в зв’язку з чим незламні до цього криптосистеми стали вразливими, яскравим історичним прикладом чого стала німецька шифрувальна машина Енігма.

Через це головною вимогою до сучасних криптосистем є стійкість до автоматизованого криптографічного аналізу, з використанням комп’ютеризованих обчислень, фактична потужність яких зростає мало не щодня.

Комбінація криптографічної та стеганографічної складової може забезпечити високий рівень захисту інформації, а розробка алгоритмів, що матимуть в собі обидві цих складові, є вкрай актуальною задачею.

Існуючі симетричні алгоритми шифрування, такі як DES в його останніх модифікаціях, IDEA, AES та інші, досить надійно захищають дані, відносно поточного рівня розвитку електронно-обчислювальної техніки, але і вони мають

ряд недоліків, що дають можливість спроб їх криптоаналізу. Також ці та інші симетричні алгоритми шифрування не мають можливості стеганографічного використання, оскільки вихідний шифротекст у цих алгоритмах є важкопрогнозованим. Розробка криптосистеми, з можливістю впливати на шифротекст, зробить можливим його змішане криптографічно-стеганографічне використання.

Метою кваліфікаційної роботи є створення криптостійкого симетричного нелінійного алгоритму шифрування з можливістю комбінованого криптографічно-стеганографічного використання.

У відповідності до сформованої мети, завданнями кваліфікаційної роботи є:

- проведення аналізу існуючих симетричних криптосистем, виявлення їх переваг та недоліків;
- дослідження впливу нелінійного шифрування, порівняння результатів з аналогами;
- розробка нелінійних модифікацій для існуючих алгоритмів шифрування з підтвердженою надійністю та вивчення їх нових властивостей;
- проектування окремої нелінійної симетричної криптосистеми з можливістю як криптографічного, так і стеганографічного застосування;
- програмна реалізація спроектованих алгоритмів, тестування та апробація.

Написання даної роботи та виконання сформованої мети буде логічним продовженням тематики, висвітленої мною в 1 науковій роботі, 1 фаховій статті та в 2 тезах доповідей на всеукраїнській та міжнародній конференціях.

1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ СУМІЖНОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення та історична еволюція симетричних криптосистем

Розпочинаючи роботу, дамо визначення базовим поняттям, якими у подальшому ми будемо оперувати. Основною наукою, в предметна область якої суміжна до теми кваліфікаційної роботи є криптографія.

Перше задокументоване використання криптографії у письмовій формі відноситься приблизно до 1900 до н.е. коли єгипетський писар використовував у написі нестандартні ієрогліфи. Серед письмових джерел з історії України також містяться закодовані записи. Найдавніші з них містяться в рукописах 12–13 ст.

Деякі експерти стверджують, що криптографія виникла спонтанно через деякий час після того, як було винайдено писемність, і її застосування варіювалося від дипломатичних послань до військових планів. Тому не дивно, що невдовзі після розвитку комп'ютерних комунікацій з'явилися нові форми криптографії. У сфері інформації та телекомунікацій криптографія необхідна під час обміну даними через будь-яке ненадійне середовище, включаючи практично будь-яку мережу, особливо мережу Інтернет [1, 2].

Криптографія – це наука, що вивчає різноманітні методи безпечного обміну даними за наявності ворожої поведінки (порушника). У загальному розумінні криптографія займається вивченням та розробкою протоколів шифрування, які не дозволяють третім особам або широкому загалу читати особисті повідомлення [3].

Криптографія включає в себе різні аспекти інформаційної безпеки, зокрема такі як цілісність даних, конфіденційність, автентифікацію та невідмовність.

Сучасна криптографія існує на перетині багатьох наукових дисциплін, зокрема таких як математика, інформатика, електротехніка, наука про зв'язок, фізика тощо.

Прикладне застосування сучасної криптографії також є вкрай широким, розгалуженим та включає в себе великий спектр речей, серед яких електронна комерція, платіжні картки на основі електронних чіпів, цифрові валюти, комп'ютерні паролі, різні види зв'язку тощо.

Окрім цього криптографія є складовою частиною більш глобального терміну – криптології.

Криптологія – це розділ інформаційних наук, що вивчає методи шифрування і дешифрування, в тому числі й методом зламу, інформації. Криптологія поділяється на два підрозділи: криптографію та криптоаналіз.

Сучасна криптографія вивчає різноманітні криптографічні примітиви. Подекуди такі криптографічні примітиви об'єднують у криптосистеми.

Криптосистема — це набір криптографічних алгоритмів, необхідних реалізації конкретного протоколу шифрування.

Зазвичай криптосистема складається із трьох алгоритмів: генерація ключа, шифрування та дешифрування. Іноді використовують термін «шифр». Частіше всього його використовують для позначення пари алгоритмів: шифрування та дешифрування. Тому термін криптосистема найчастіше використовується, коли важливим є також алгоритм генерації ключа.

Перейдемо до класифікації криптографії. Існує декілька різних систем класифікації криптографії за різними критеріями.

В контексті даної кваліфікаційної роботи найбільш актуальним буде поділ за типом функціонування криптосистем. Відповідно до цього виділяють симетричні та асиметричні системи. В іноземних джерелах асиметричні криптосистеми часто називають криптосистеми з відкритим ключем, або PKC [4].

Асиметрична криптографія - це криптосистеми, що використовують декілька, як правило пару, ключі. Кожна пара складається з відкритого ключа, який по-замовчуванню є публічним та може бути відомим широкому загалу, а також закритого ключа, що тримається в таємниці та має бути відомий лише власнику. Генерація пари закритого та відкритого ключів залежить від

конкретного криптографічного алгоритму. Захищеність подібних асиметричних криптосистем ґрунтуються на математичних завданнях, що називають односторонніми функціями: прості та швидкі для одностороннього виконання, але вкрай важкі для виконання в зворотній бік (рис. 1.1).

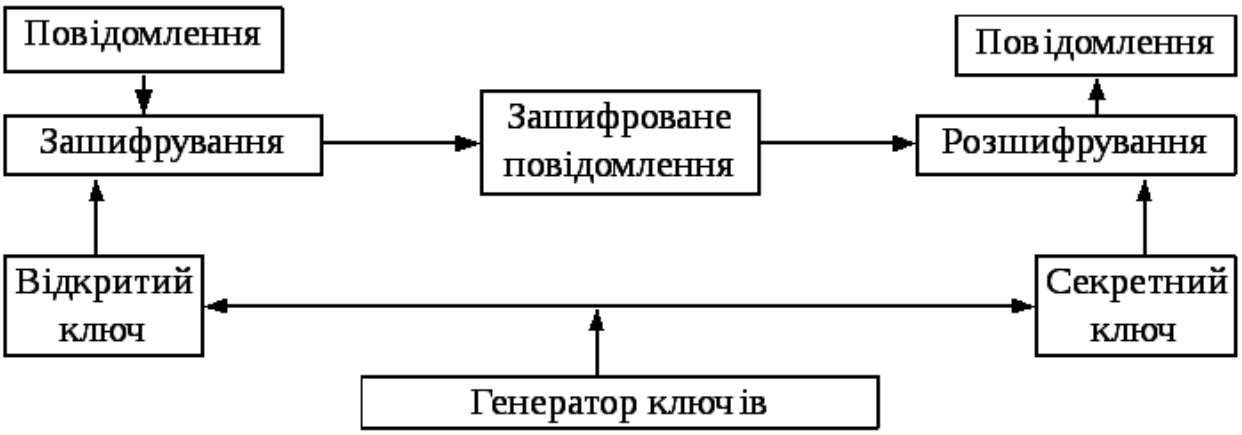


Рисунок 1.1 – Логіка роботи асиметричного алгоритму шифрування

Проте, в контексті даної роботи, основну увагу буде зосереджено на інший тип криптосистем – симетричні.

Симетричні криптосистеми – криптосистеми, в яких для шифрування та дешифрування використовується один і той самий криптографічний ключ. При цьому даний ключ повинен зберігатися в таємниці обома сторонами та не може передаватися незахищеними каналами зв’язку. Симетричне шифрування є значно старшим за асиметричне та до винаходу схеми шифрування з публічним ключем було єдиним існуючим. Вся класична криптографія фундаментально базується саме на симетричному шифруванню.

Симетричні алгоритми шифрування та дешифрування даних широко використовуються у КСЗІ, в комп'ютерній техніці в системах приховування конфіденційної та комерційної інформації від не коректного використання порушниками. Головним недоліком симетричних криптосистем є те, що обидві сторони заздалегідь повинні мати секретний ключ. З іншого боку симетричні алгоритми шифрування є значно швидшими та простими для апаратних реалізацій ніж асиметричні [4-6].

Фактично будь-яка криптосистема базується на простоті одностороннього перетворення: на основі криптографічного ключа можна легко та швидко перетворити, тобто зашифрувати, повідомлення та, у симетричних криптосистемах, маючи той ж самий ключ, провести зворотне перетворення, тобто дешифрувати, шифрованого повідомлення (рис. 1.2).

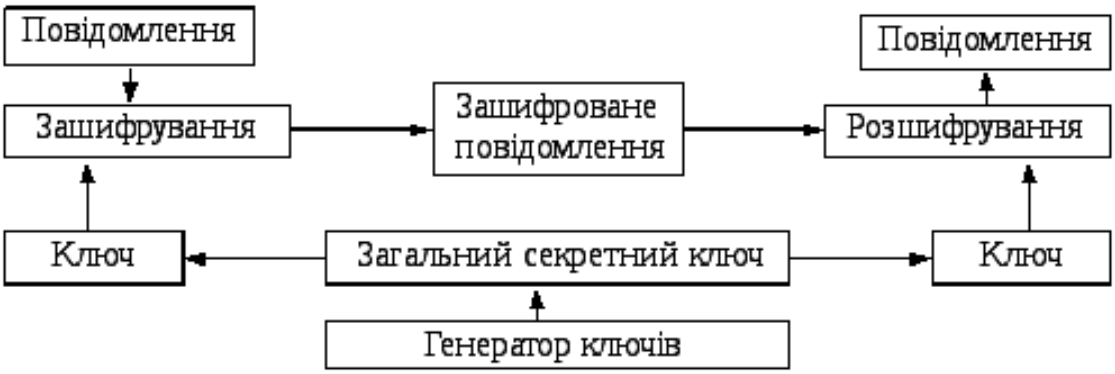


Рисунок 1.2 – Логіка роботи симетричного алгоритму шифрування

При цьому перед криптоаналітиком, який немає зазначеного ключа, а має лише саме шифроване повідомлення, стоїть суттєво важче завдання, оскільки існує вкрай велика кількість способів якими це повідомлення можна розшифрувати, затративши при цьому непрактичний час, але існує лише один, в результаті якого можливо отримати початкове повідомлення [4, 6].

Однією з фундаментальних основ криптографії в цілому та симетричної криптографії зокрема є принцип Керкхоффса. Даний принцип є важливою складовою надійності будь-якого шифру. У вільній інтерпретації звучить він таким чином: «Надійність та зламостійкість криптосистеми повинна ґрунтуватись не на знанні принципів шифрування, а виключно на знанні криптографічного ключа». Інакше кажучи, безпека алгоритму не повинна базуватися на тому що криптоаналітик не знає алгоритм шифрування, натомість вона повинна базуватись виключно на знанні криптографічного ключа.

Це правило не є даремним і безліч разів його істинність підтверджувалась, оскільки реверсивна інженерія часто дає можливість виокремити алгоритм

роботи шифратора, будь то програмне забезпечення, чи апаратний мікроконтроллер.

Багатовіковий досвід експлуатації симетричних криптосистем показав ще ряд вимог, які криптосистема в обов'язковому порядку повинна виконувати, задля забезпечення стійкості. Серед них:

- неможливість отримання криптографічного ключа на основі шифротексту;
- неможливість отримання криптографічного ключа на основі пари відкритий текст – шифротекст;
- стійкість шифротексту до статистичних досліджень та диференціювання;
- відсутність, або чіткий перелік «слабких» ключів.

В сучасних криптосистемах основою криптографічної стійкості алгоритму являється складність електронних обчислень, відповідно до якої і проводиться оцінка надійності шифру. Ця ідея була висунута американським математиком Джоном Нешем, який, окрім цього, також висунув гіпотезу, що складність обчислень, необхідних для успішного криптоаналізу, перебуває у експоненційній залежності від довжини криптографічного ключа.

Таким чином, ми дали визначення основним робочим поняттям, розібрали історичні складові криптографії та проаналізували суміжну предметну область.

1.2 Актуальні методи симетричного шифрування та їх поширеність

В продовження висвітленої в попередньому підрозділі інформації, проаналізуємо найпопулярніші методи симетричного шифрування.

На сьогоднішній день абсолютна більшість поширених симетричних алгоритмів шифрування є блочними. Багато з них базується на основі мережі Фейстеля: класичної, або модифікованої.

Одним з найбільш поширених стандартизованих алгоритмів шифрування є DES. Це застарілий стандарт шифрування, побудований на основі класичної мережі Фейстеля [6, 7].

DES є найвідомішою і однією з найбільш широко використовуваною криптосистемою, не зважаючи на умовну стійкість. Він був розроблений IBM і прийнятий Національним бюро стандартів США у середині 1970-х (рис. 1.3).

Рисунок 1.3 – Схема роботи шифру DES

Іншим популярним симетричним алгоритмом шифрування на основі модифікованої мережі Фейстеля є IDEA.

IDEA — це симетричний блочний шифр, розроблений Джеймсом Мессі та Сюецзя Лай і вперше описаний в 1991 році. Даний алгоритм був створений як заміна стандарту DES. IDEA є незначною редакцією попереднього шифру PES.

Криптосистема була запатентована у низці країн, проте все одно знаходився у відкритому доступі для некомерційного використання. Термін дії усіх патентів остаточно закінчився в 2012 році, і IDEA тепер не є запатентованою. Таким чином, даний алгоритм є цілком безкоштовний для будь-якого використання.

Алгоритм IDEA шифрує дані блоками по 64 біти, а ключ шифрування алгоритму має розмір 128 біт. Блок зашифрованих даних розділений на чотири 16-бітові підблоки на яких виконується вісім раундів перетворення. Після цього

два внутрішні підблоки міняються місцями - у всіх раундах, крім останнього. Після восьми раундів виконуються додаткові перетворення, які деколи називають дев'ятим раундом (рис. 1.4).

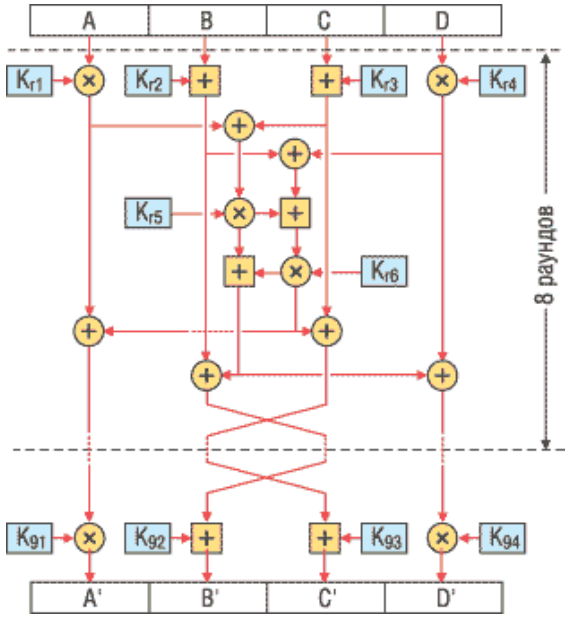


Рисунок 1.4 – Схема роботи шифру IDEA

Ще одним блочним симетричним алгоритмом шифрування, що вважається одним із найбільш захищеним на сьогоднішній день у світі є AES.

AES, також відомий як метод шифрування Rijndael у криптографії – це стандарт блокового шифрування, прийнятий федеральним урядом США. Він офіційно використовується для заміни оригінального DES. Ця криптосистема була глибоко проаналізована багатьма країнами і широко використовується у всьому світі. Після п'ятирічного відбору криптосистема AES була опублікована Національним інститутом стандартів і технологій на FIPS PUB 197 26 листопада 2001 р. і стала чинним стандартом 26 травня 2002 р.

У 2006 році AES став одним із найпопулярніших алгоритмів шифрування із симетричним ключем.

Алгоритм AES - це симетричний блочним алгоритм шифрування. Існує три типи алгоритму шифрування AES, а саме AES-128, AES-192 і AES-256. Відмінності між ними показані в таблиці (Табл. 1.1).

Таблиця 1.1 – Порівняльні характеристики алгоритмів AES

Алгоритм	Розмір ключа, біт	Розмір блоку, біт	Кількість раундів
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

Функція шифрування AES, на відміну від попередніх криптосистем, не має в своїй основі мережі Фейстеля, а представляє собою мережу заміщення-перестановки. Алгоритм шифрування для 128-бітного ключа передбачає послідовне виконання наступних операцій:

- нелінійна заміна байт;
- зміщення рядків у матриці розміром 4 на 4;
- перемішування колонок (окрім 10-го раунду);
- функція XOR блоку з раундовим ключем.

Дешифрування відбувається у зворотному порядку (рис. 1.5).

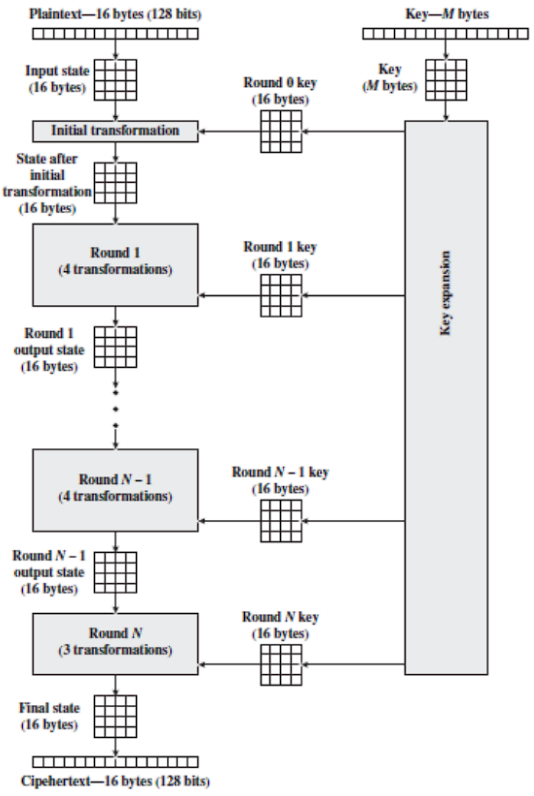


Рисунок 1.5 – Схема роботи шифру AES

Також слід згадати про блоковий алгоритм шифрування, закріплений в державному стандарті України – ДСТУ ГОСТ 28147:2009 [8].

Це симетричний блоковий алгоритм шифрування що дістався Україні в спадок від СРСР. Його було рекомендовано використовувати для захисту будь-яких електронних даних, хоча й не забороняються інші методи шифрування.

Вказаний стандарт створювався з урахуванням досвіду інших країн, й зокрема, було взято до уваги недоліки й нереалізовані можливості алгоритму DES.

На різних кроках роботи шифру ДСТУ ГОСТ 28147:2009 дані використовуються й інтерпретуються різним чином. В одних випадках елементи даних обробляються як масиви незалежних бітів, в інших випадках вони враховуються як ціле число без знаку, в третіх – як складна структура, що складається з кількох простіших елементів (рис. 1.6).

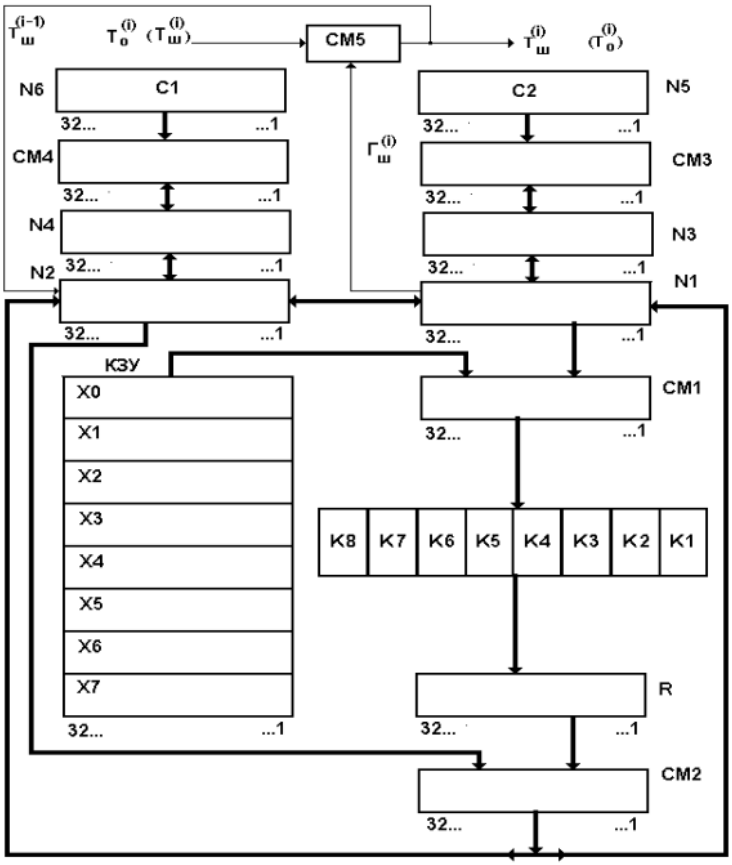


Рисунок 1.6 – Схема роботи шифру ДСТУ ГОСТ 28147:2009

Таким чином ми розглянули найбільш актуальні симетричні блокові криптосистеми. Можемо проаналізувати що вони мають в своїй основі різні методи та механізми роботи. Деякі з розглянутих алгоритмі вже є частково застарілими, а нові та більш захищені алгоритми мають інструменти забезпечення нелінійного шифрування в своїй основі.

1.3 Сучасна стеганографія та її використання

Завдання захисту інформації від несанкціонованого доступу вирішувалося з найдавнішої історії людства. Як говорилося раніше, ще з часів Стародавнього світу виділилося дві основні гілки вирішення цього завдання, які дійшли і до нашого часу: криптографія та стеганографія.

На відміну від криптографії, стеганографічний метод захисту приховує сам факт існування секретного повідомлення.

Саме слово «стеганографія» пішло з давньогрецької та дослівно означає «тайнопис». Так склалося що цей напрямок з'явився раніше, проте згодом багато у чому був витіснений криптографією.

Існує безліч способів здійснення тайнопису. Загальною його рисою є те, що секретне повідомлення, яке приховується, вбудовується в деякий неважливий, непримітний об'єкт, який пізніше відкрито передається адресату. В криптографічному методі сама наявність шифрованого повідомлення привертає до себе зайву увагу порушників, у випадку ж стеганографії факт прихованого зв'язку залишається непомітним, або малопомітним [9, 10].

Стеганографія – це спосіб замаскованої передачі інформаційного повідомлення, яке імплантується в інший елемент.

Стеганографія характеризується як процедура маскування повідомлень за допомогою деякого таємного методу, про який не відомо порушнику. Дана методика використовується з метою заховати таємне повідомлення в різні неважливі повідомлення, файли, об’єкти.

Стеганографія добре підходить для приховування таємної та чутливої інформації в точних даних контейнера. Стеганографія має важливе значення для охоплення інформації в таких медіа, як моделі, зображення, звук, відео, текст тощо.

Сучасна стеганографія, окрім прихованої передачі даних успішно вирішує й інший перелік завдань (рис. 1.7).

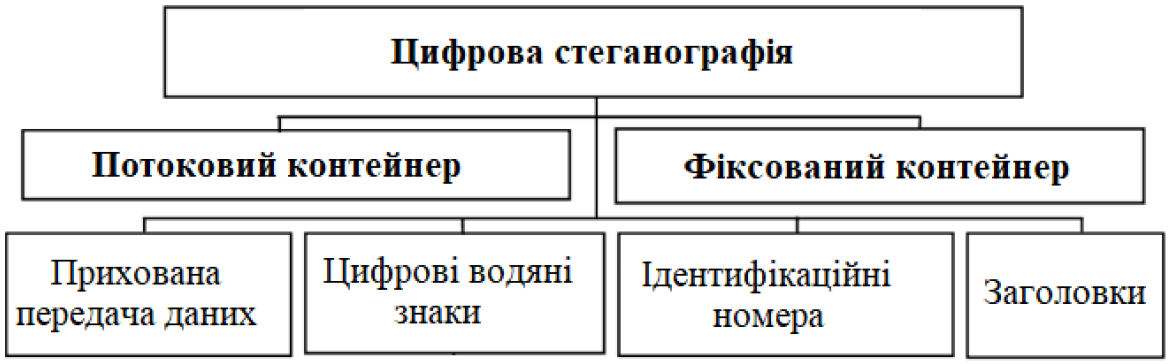


Рисунок 1.7 – Класифікація стеганографічних напрямків

Цифрова стеганографія також може використовуватися для забезпечення авторського права для маркування медіа-файлів, а також для ідентифікації різноманітних файлів та документів. Наявність ЦВЗ на електронних об’єктах дозволяє виявити факт крадіжки та плагіату. Порушнику при цьому достатньо важко виявити замасковані ЦВЗ та знищити їх [10, 11].

Деякі сучасні стеганографічні алгоритми по-замовчуванню мають криптографічний захист, використовуючи деякий секретний ключ при записі повідомлення, інші ж маскують незахищені дані.

Поєднання стеганографічних та криптографічних методів захисту інформації сприяє високому рівню її захищеності, а також дозволяє уникнути статистичних досліджень криптоаналізу та будь-яких інших маніпуляцій з зашифрованим повідомленням, оскільки криптоаналітику буде проблематично навіть виділити сам шифротекст з масиву повідомлень [12].

Окрім цього, при комплексному використанні криптографії та стеганографії, тобто при використанні стеганографічних методів маскування з

метою приховування деякого шифрованого повідомлення, відбувається ефект «накладки», за якого стегоаналіз значно ускладняється тим що аналітика не буде критерію успішності виокремлення шифрованого повідомлення, оскільки шифротекст сам по собі є неосмисленим набором символів, чи байт.

Стеганографія може використовуватись для автентифікації, наприклад, камер відеоспостереження, які накладатимуть на медіа ЦВЗ, що є вкрай малопомітним.

Для практичного вирішення задач стеганографії мають важливе значення стеганографічні протоколи.

Стеганографічний протокол - це певний порядок дій, який виконується двома, чи більше сторонами, та який призначений для вирішення певного завдання. Найкращий та найефективніший стеганографічний алгоритм через його неправильне застосування може не досягти поставленої перед ним мети.

Протокол та алгоритм є подібними термінами, оскільки обидва з них – це деяка послідовність дій. Різниця між ними в тому, що до протокол передбачає залучення двох або більше сторін та, як наслідок, їх ефективну та скоординовану взаємодію. Подібно до алгоритму, протокол складається із кроків, на кожному з яких виконуються ті, чи інші дії.

Як зазначалось раніше, стеганографія також може бути з потоковим, або фіксованим контейнером. У даній класифікації стеганографія є подібною до криптографії.

До потокової стеганографії можна віднести використання ЦВЗ на відеоряді камери спостереження, або на тих чи інших датчиках, що працюють в реальному часі.

Стеганографія з фіксованим контейнером передбачає використання кінцевого контейнера з визначеним розміром, вміст якого може бути малопомітно модифіковано. Маючи фіксований розмір контейнера, різні протоколи можуть цим користуватися. До прикладу, якщо мова йде про деякий файл-контейнер, то стеганографічне повідомлення може бути заховано з певною

закономірністю від початку, чи від кінця файлу-контейнера. Фіксований розмір контейнера дозволяє зчитати дане повідомлення за такою ж логікою.

Також варто зазначити що використання терміну «малопомітно» передбачає взаємодію з людиною та підвищення ймовірності саме людської помилки з боку аналітика.

Отже, ми розглянули поняття сучасної стеганографії, її види, класифікацію та сфери застосування. Стеганографія, при правильному використанні, є надійним та перспективним засобом захисту інформації.

1.4 Криптоаналіз та критерії надійності сучасних симетричних криптосистем

Будь-яка сучасна криптосистема перед запровадженням та безпосередньою експлуатацією повинна пройти перевірку стійкості та надійності.

Стійкістю в криптографії вважається можливість криптосистеми протистояти всім видам криптоатак, що можуть бути спрямовані на неї безпосередньо, на вихідний шифротекст, на пари відкритого та шифрованого тексту тощо.

Ідеальною, фактично, можна вважати таку криптосистему в якій отримання відкритого тексту з шифрованого і навпаки можливе виключно на основі наявного криптографічного ключа і ніяким іншим чином, причому що сам криптографічний ключ не повинен ніяким способом вираховуватись на основі відкритого, або шифрованого тексту, чи їх порівняння та бажано щоб він при цьому був досить компактним.

В зв’язку з суттєвим стрибком комп’ютерних технологій та, як наслідок, з гігантським ростом електронно-обчислювальної потужності техніки, абсолютна більшість криптосистем перестали вважатися незламними, адже неймовірно велика швидкість обчислень зробила їх вразливими навіть перед звичайною атакою грубої сили – простим перебором криптографічних ключів, який раніше

здавався неможливим. Цей факт накладає ще більш серйозні вимоги до перевірки надійності алгоритмів шифрування [6, 12].

Наука яка займається дослідженням можливостей злому криптосистем називається криптоаналіз, та на рівні з криптографією є підрозділом науки криптології [13].

Криптоатаки на симетричні криптосистеми бувають:

- на шифрований текст;
- на основі пар «відкритий-шифрований текст»;
- на обраний відкритий текст;
- на обраний зашифрований текст.

Атаки на шифрований текст в свою чергу поділяються на категорії:

- атака грубої сили;
- атака за маскою (за зразком);
- статистична атака;

Атака на шифрований текст є найбільш складною для криптоаналітика, оскільки вона передбачає що окрім самого шифрованого повідомлення у аналітика нічого немає (рис. 1.8).

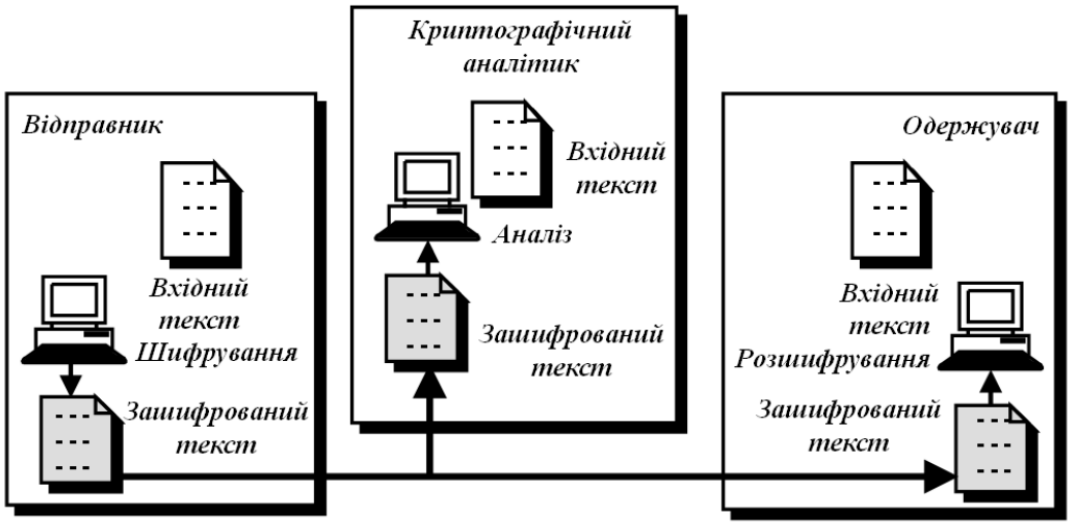


Рисунок 1.8 – Атака на шифрований текст

Варто нагадати, що згідно принципу Керкхоффса криптоаналітик досконало знає принцип роботи того чи іншого алгоритму та всі нюанси його роботи, а єдиний невідомий для нього параметр – це криптографічний ключ [6, 12].

Також передбачається що криптоаналітик може лише перехоплювати шифровані повідомлення, зберігати та не може їх модифікувати.

Атака на шифротекст можлива у випадку якщо:

- алгоритм шифрування має малу кількість криптографічних ключів, які можливо перебрати автоматизовано за практичний час;
- алгоритм є потоковим, і/або в ньому зберігається пряма лінійна чи інша закономірність між вхідним та шифрованим текстом, яку можливо виявити та експлуатувати;
- алгоритм має інші приховані недоліки в роботі, та непродуману математичну основу, за якої відкритий текст можливо отримати з шифротексту без криптографічного ключа.

Атака на основі пар відкритого та шифрованого тексту є більш сприятливою для аналітика. Передбачається що аналітик має можливість отримувати пари відкритого тексту та шифрованого тексту, що йому відповідає. При цьому криптоаналітик також досконало знає алгоритми криптосистеми, невідомий йому виключно ключ (рис. 1.9).

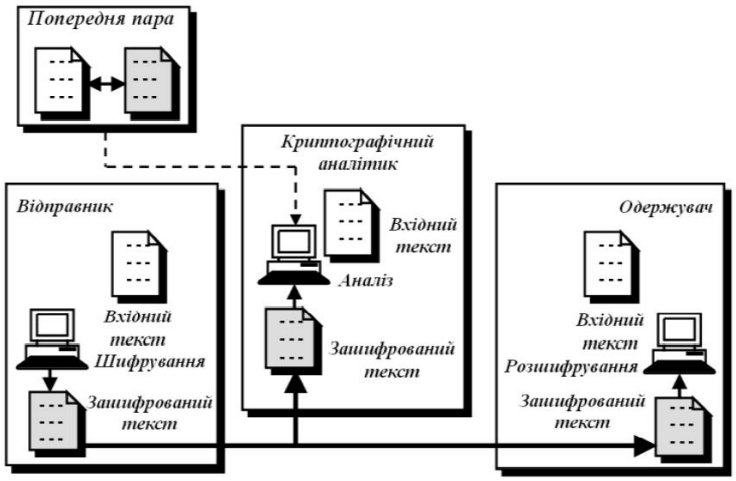


Рисунок 1.9 – Атака на відомий вхідний текст

Для реалізації даної атаки можна використати ті ж методи, що й використовуються в атаці за шифротекстом, проте цю атаку простіше здійснити, оскільки криптоаналітик має більше інформації для аналізу [13, 14].

Атака на обраний відкритий текст та атака на обраний шифрований текст моделюють ситуацію коли криптоаналітик напряму, чи опосередковано має доступ до комп'ютерів сторін, чи іншим чином може вільно створювати шифротекст, відповідний конкретному відкритому тексту та навпаки, на власний розсуд (рис. 1.10).

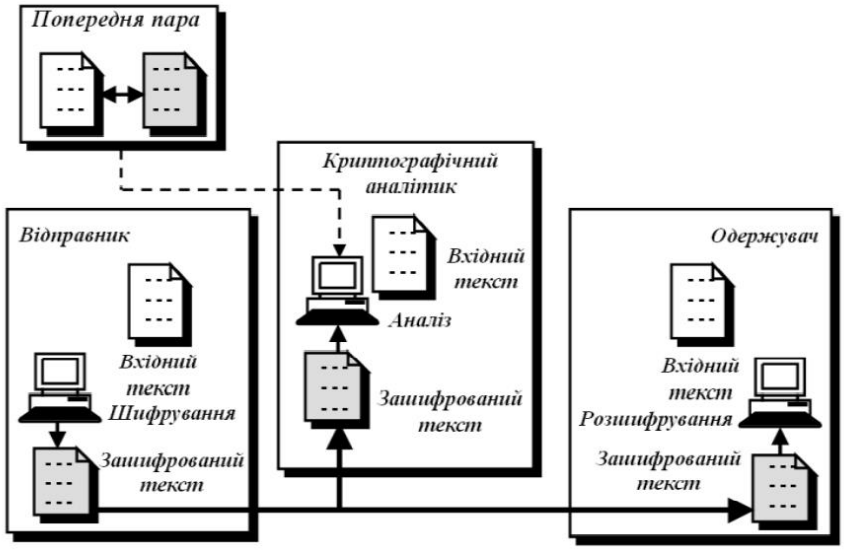


Рисунок 1.10 – Атака на обраний зашифрований текст

Такого типу атаки значно простіше здійснити, оскільки вони є найбільш сприятливими для криптографічного аналітика, проте вони є малоймовірними в реальності, оскільки передбачають занадто багато допущень.

Наведені види криптографічних атак можуть також використовуватися для зламу сучасних блокових шифрів, проте актуальні блокові шифри часто успішно їм протистоять.

Але відносно нещодавно були винайдені нові види атак, спрямовані на блокові шифри, що базуються на структурах сучасних блокових шифрів.

Дані атаки використовують методи лінійного й диференціального аналізу.

Будь яка сучасна криптосистема повинна успішно боротися з наведеними методами аналізу, повністю унеможливлювати їх, або робити їх використання неприйнятним через нереалістичну кількість необхідного часу чи ресурсів для її реалізації. Також слід враховувати постійний розвиток нових технологій та невпинний ріст обчислювальних можливостей. інформації.

1.5 Висновок

У цьому розділі ми дали визначення основним робочим поняттям, розібрали історичні складові, еволюцію та сучасний стан криптографії, дослідили найбільш відомі, популярні та актуальні методи симетричного шифрування, вивчили актуальний стан стеганографії як науки та ніші, яку вона займає.

Окрім цього були розглянуті методики криптоаналізу та критерії надійності сучасних симетричних криптосистем, а також в цілому досліджена предметна область, суміжна з темою даної кваліфікаційної роботи.

Проаналізована в першому розділі теоретична інформація стане основою для всіх подальших досліджень, буде використовуватись та прийматись до уваги надалі.

2 ОБҐРУНТУВАННЯ ВИКОРИСТАННЯ НЕЛІНІЙНОГО ШИФРУВАННЯ ЗАДЛЯ УСУНЕННЯ НЕДОЛІКІВ ІСНУЮЧИХ СИМЕТРИЧНИХ КРИПТОСИСТЕМ

2.1 Оцінка впливу нелінійності шифрування на загальні характеристики криптосистеми

На основі інформації, дослідженої та проаналізованої у попередньому розділі, проведемо оцінку впливу нелінійності шифрування на загальні характеристики криптосистем.

Для початку слід визначитись з термінологією, оскільки у різних джерелах даний термін використовують у різному розумінні. Даючи визначення поняттю «нелінійність», підемо від зворотного.

Лінійна криптосистема – це криптосистема побудована виключно на лінійних криптографічних примітивах.

Лінійний криптографічний примітив, в рамках даної роботи – це деякий криптографічний алгоритм низького рівня, що повертає визначене прогнозоване значення на виході, при відомому значенні вхідних параметрів, назвемо їх C та K, хоча їх може бути й більше (рис. 2.1).

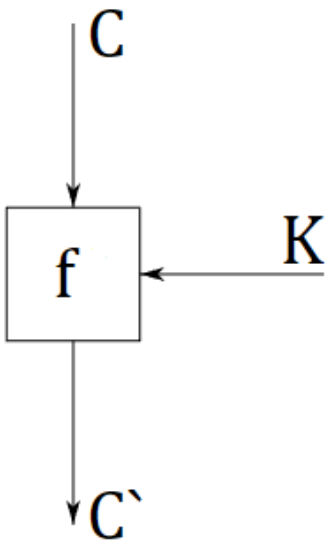


Рисунок 2.1 – Лінійний криптографічний примітив

Відповідно до рисунка, примітив f ми вважатимемо лінійним, якщо приймаючи визначену кількість параметрів він повертає одне прогнозоване значення C'.

Виходячи з цього, нелінійний криптографічний примітив – це деякий криптографічний алгоритм низького рівня, що окрім визначених параметрів має деякий модифікатор функціонування M, який впливає на вихідне значення. Нелінійність в такому випадку досягається за рахунок того що при однакових вхідних значеннях C та K ми матимемо не одне прогнозоване вихідне значення, а масив ймовірних значень, одне серед яких визначить саме модифікатор (рис. 2.2).

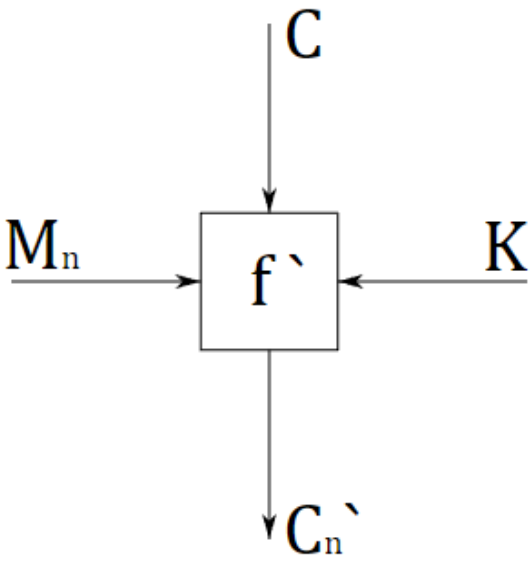


Рисунок 2.2 – Нелінійний криптографічний примітив

Модифікатор в такому випадку може бути або стороннім значенням, що передається ззовні, або вихідним значенням іншого примітиву з наявними параметрами, або ж випадковим значенням із заданого діапазону.

Відповідно нелінійна криптосистема – це криптосистема, що включає в себе один, або декілька нелінійних криптографічних примітивів.

Нелінійність криптографічних перетворень характерна для ряду сучасних криптосистем, зокрема, частково даний принцип реалізовано у алгоритмах AES

– офіційно затвердженому американському стандарті. В даному алгоритмі деколи вводять поняття вектору шифрування [6, 12, 15].

Нелінійні криптосистеми мають характерну рису – розгалуження на гілки в процесі функціонування на тому чи іншому етапі, серед яких спрацьовує лише одна (рис. 2.3).

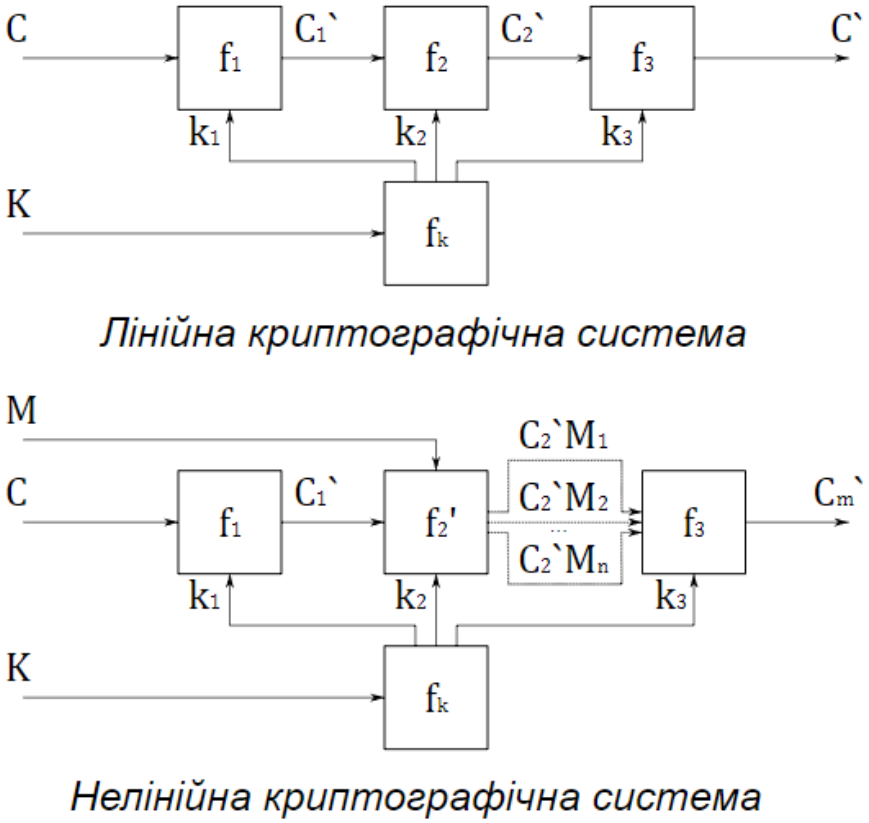


Рисунок 2.3 – Лінійна та нелінійна криптографічні системи

Введення нових параметрів значно збільшує варіативність шифрування, адже додавши хоча б один нелінійний криптографічний примітив у криптосистему, модифікатор даного примітиву впливатиме на кінцевий шифротекст.

Таким чином ми з одного боку збільшили варіативність шифрування, а з іншого створили можливість частково впливати на кінцевий шифротекст, змінюючи модифікатор, оскільки для криптосистеми з одним нелінійним примітивом кількість можливих шифротекстів для одного вхідного блока даних

буде рівна N, де N – це водночас кількість можливих модифікаторів нелінійного криптографічного примітиву.

При такій організації алгоритму шифрування ускладниться і криптоаналіз, причому ускладниться він зразу за декількома критеріями:

- шифрований текст важче піддаватиметься статистичному дослідженню;
- встановлення відповідності між відкритим та шифрованим текстом ускладниться [16, 17];
- успішна реалізація атаки грубої сили потребуватиме значно більше обчислювальної потужності.

Перелічені тези є актуальними у випадку якщо криптоаналітик не знає значення модифікатора, а отже даний параметр повинен або генеруватися за деяким алгоритмом на основі загального криптографічного ключа, чи раундового ключа, або передаватися таємно разом із ключем, або передаватися іншим можливим чином, зі збереженням таємності.

Особливо цікавим є випадок коли модифікатор генерується випадковим чином. Звісно в такому випадку з'являється необхідність у криптостійкому генераторі випадкових чисел, що може ускладнити практичну реалізацію такої криптосистеми, проте у разі вирішення цієї проблеми ми отримаємо алгоритм шифрування, який при послідовному шифрування одного і того ж самого відкритого тексту одним і тим ж ключем видаватиме різні шифровані дані, причому ці шифротексти також будуть випадкові [6, 18, 19].

Серед недоліків нелінійного шифрування можна виділити наступне:

- дані алгоритми важко, або в принципі неможливо використовувати як основу хеш-функцій, чи для порівняння двох шифрованих текстів;
- нелінійний примітив створює більше навантаження на шифрувальний пристрій, шифрування може виконуватись довше, хоча й різниця може бути не значна.

Отже, підсумовуючи сказане, можемо виділити в цілому позитивний вплив використання нелінійних криптографічних примітивів. При вдалому

застосуванні вони суттєво ускладнюють криптоаналіз, додають варіативність шифрування, в наслідок чого з’являється ряд нових цікавих особливостей, що можуть знайти своє прикладне використання, зокрема стеганографічне. За підсумком, нелінійні криптосистеми є перспективним напрямком розвитку криптографії, що безумовно потребує подальших досліджень.

2.2 Постановка задачі проектування кваліфікаційної роботи

Перед переходом до безпосередньої розробки вдосконаленого алгоритму нелінійного шифрування з можливістю стеганографічного використання, підведемо підсумки дослідженої теоретичної інформації та проведеного оцінювання, а також сформуємо задачу проектування.

Зі знайдених даних стає зрозуміло, що основними вимогами до сучасних криптосистем є стійкість до криптографічного аналізу, швидкодія та підтверджена ефективність роботи.

Блочна структура алгоритму шифрування позитивно впливає на криптостійкість та є характерною для практично усіх сучасних криптографічних систем.

Важливе місце, окрім криптографії, також займає стеганографічний захист інформації. Це перспективна галузь інформаційної безпеки, яка стрімко розвивається та вже на сьогодні має широкий спектр прикладного застосування, зокрема у ЦВЗ та прихованій передачі повідомлень, які не викликають підозри у потенційних порушників.

Криптостійкість алгоритму має базуватись на непрактичній складності обчислень, необхідних для успішної реалізації криптоатаки.

Використання нелінійних криптографічних примітивів, прогнозовано, позитивно вплине на криптосистему, за умови продуманого їх використання та мінімізації негативних особливостей.

На основі цього можемо сформувати задачу проектування кваліфікаційної роботи.

Відповідно до мети кваліфікаційної роботи, задачею проектування є розробка та програмна реалізація криптостійкого симетричного нелінійного алгоритму шифрування. Даний алгоритм повинен відповідати наступним характеристикам:

- варіант даного алгоритму має бути розроблений як модифікація до існуючого блочного шифру, з метою дослідження його властивостей;
- алгоритм повинен бути спроектований, розроблений та програмно реалізований у вигляді окремої нелінійної симетричної криптосистеми;
- окремий варіант алгоритму має бути розроблений для стеганографічного використання;
- програмна реалізація спроектованих алгоритмів, повинна пройти тестування та апробацію.

Відповідно до мети кваліфікаційної роботи, задачею проектування є розробка та реалізація криптостійкого

2.3 Висновок

У даному розділі ми дали визначення лінійному та нелінійному криптографічним примітивам, провели оцінку впливу нелінійності на криптосистеми, де вони використовуються, а також сформували задачу проектування кваліфікаційної роботи.

За результатами оцінки ми можемо виділити в цілому позитивний вплив використання нелінійних криптографічних примітивів. При вдалому застосуванні вони суттєво ускладнюють криптоаналіз, додають варіативність шифрування, в наслідок чого з’являється ряд нових цікавих особливостей, що можуть знайти своє прикладне використання, зокрема стеганографічне. Були наведені схеми лінійних та нелінійних криптографічних примітивів та описані методи їх трансформації.

Нелінійні криптосистеми є перспективним напрямком розвитку криптографії, що безумовно потребує подальших досліджень у даному

напрямку, оскільки вони мають перелік особливостей, не характерних іншим видам криптосистем.

На основі таких результатів, була поставлена задача проектування кваліфікаційної роботи, яка полягає у розробці та програмній реалізації криптостійкого симетричного нелінійного алгоритму шифрування, що відповідатиме переліченим характеристикам, у відповідності до мети кваліфікаційної роботи.

Таким чином ми обґрунтували використання нелінійних криптографічних примітивів у подальшому проектуванні, розробці та програмній реалізації алгоритму шифрування, а також можливості використання подібних криптографічних примітивів з метою модифікації існуючих симетричних криптосистем.

3 РОЗРОБКА ВДОСКОНАЛЕНОГО АЛГОРИТМУ НЕЛІНІЙНОГО ШИФРУВАННЯ З МОЖЛИВІСТЮ СТЕГАНОГРАФІЧНОГО ВИКОРИСТАННЯ

3.1 Загальні підходи до формування нелінійності у криптосистемі

На основі даних з попередніх розділів відомо що для побудови нелінійної криптографічної системи слід використовувати в її структурі нелінійні криптографічні примітиви з модифікатором функціонування.

Такі криптографічні примітиви представляють з себе декілька звичайних лінійних примітивів, з'єднаних в одну структуру з «перемикачем», залежного від значення модифікатора (рис. 3.1).

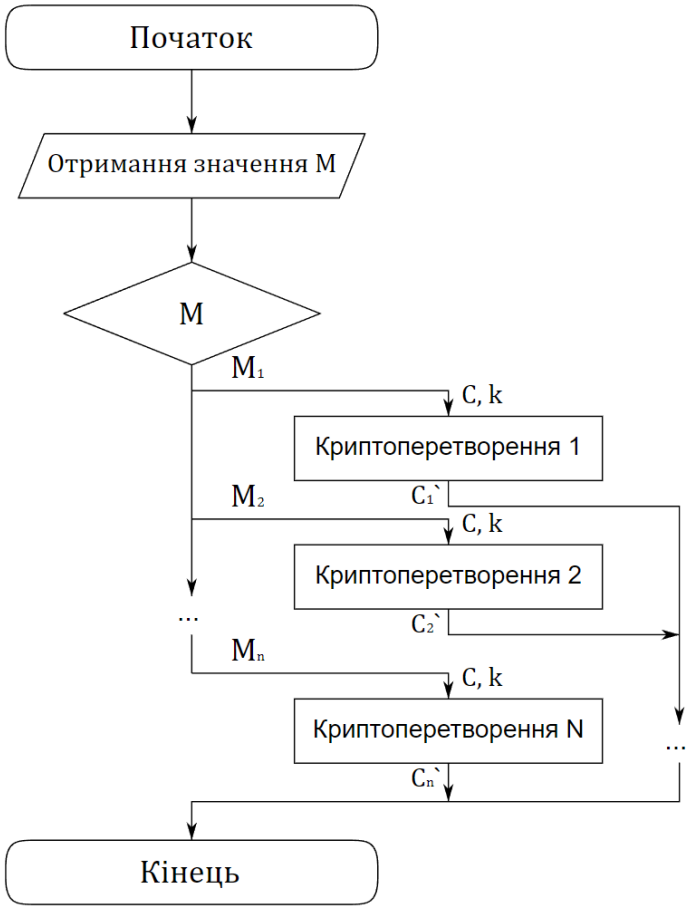


Рисунок 3.1 – Блок-схема нелінійного криптографічного алгоритму

Обов’язковою вимогою до лінійних криптографічних перетворень у складі нелінійної системи є те що вони повинні бути подібними за своєю структурою: вони повинні мати однакову кількість подібних за типом, розміром та логікою вхідних параметрів, а також ідентичні за такими ж характеристиками вихідні блоки.

У якості лінійних криптографічних примітивів можуть бути однакові за логікою елементи, що відрізняються виключно технічними особливостями. Розглянемо приклад нелінійного примітиву.

Для наглядної демонстрації у якості лінійних криптографічних перетворень візьмемо звичайне зчеплення блоків. Велика кількість алгоритмів мають в своїй структурі подібне зчеплення або конкатинацію у складі початкових чи кінцевих раундових або загальних перетворень. Припустимо у нас є два блока даних, розміром 8 біт, які слід з’єднати деяким чином. Дану логіку ми можемо реалізувати великою кількістю способів, зчіплюючи блоки за різною маскою (рис. 3.2).

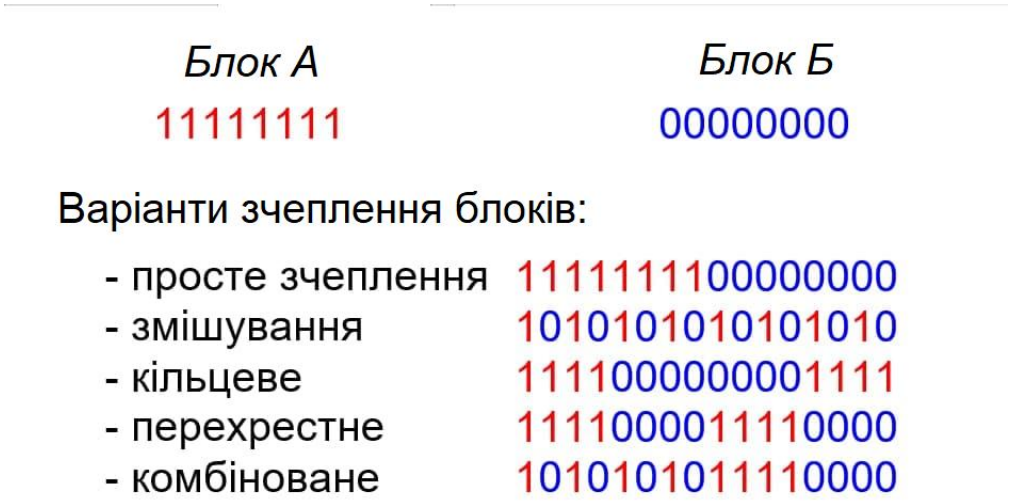


Рисунок 3.2 – Варіанти зчеплення блоків

Використання того чи іншого способу зчеплення ніяким чином принципово не впливає на криптосистему глобально, хоча це звичайно варто вивчати та перевіряти для кожної криптографічної системи індивідуально, проте

використання різних видів зчеплення блоків кардинально вплине на вихідний шифротекст, повністю його змінивши.

Користуючись наведеною вище блок-схемою та вказаним прикладом, можемо змоделювати нелінійну криптосистему, в якій в залежності від модифікатора M блоки C та k будуть зчіплюватись одним з наведених способів. На вході даний алгоритм завжди прийматиме два блоки даних, розміром по 8 біт кожен, а на виході завжди повертатиме один 16-бітний з'єднаний блок, проте яким чином відбулось це з'єднання можна сказати виключно знаючи значення модифікатора M.

Криптографічний аналітик, маючи вихідний «шифрований» блок та не знаючи значення модифікатора, постане перед досить складною задачею розділення з'єднаного блоку на складові, оскільки йому доведеться враховувати множину ймовірних варіантів зчеплення, навіть якщо кожен з них йому відомий.

При чому варто наголосити що ніякі інші складові криптографічної системи ніяким чином не змінюються, а заміна статичного лінійного криптографічного примітиву на динамічний нелінійний не змінює загальну логіку роботи алгоритму та базові механізми криптографічного захисту, а лише додає варіативність шифрування та надає алгоритму характерні особливості, описані у попередніх розділах.

При такій концепції шифрування ускладняється сама організація криптографічного аналізу, особливо при випадковій генерації модифікатора. Суттєво збільшується кількість шифрованого матеріалу, обчислювальних потужностей та, як наслідок, часу для успішної реалізації криптографічної атаки на такий алгоритм.

Подібним чином може розгалужуватись не лише один лінійний криптографічний примітив, а група таких примітивів. Наприклад у класичній мережі Фейстеля існує поняття криптографічної функції, яка також може бути реалізована багатьма способами.

Описані підходи дають можливість модифікувати вже існуючі криптосистеми. Це може переслідувати ряд завдань, зокрема:

- підвищення криптографічної стійкості алгоритму;
- створення варіативності шифрування;
- ускладнення статистичних досліджень шифротексту;
- управління вихідним шифротекстом.

Окрім цього, звичайно, на основі описаних нелінійних криптографічних примітивів може бути спроектована самостійна криптографічна система, з використанням досвіду та інструментів сучасної криптографії.

Таким чином ми розглянули підходи до формування нелінійності у криптосистемах, вивчили можливість модифікації існуючих криптографічних систем, а також можливість розробки самостійної криптосистеми на принципах нелінійності.

3.2 Розробка модифікованої нелінійної криптосистеми, на основі алгоритму шифрування DES

Користуючись інформацією із попередніх розділів, спробуємо модифікувати існуючий симетричний блоковий шифр за допомогою нелінійних криптографічних примітивів.

У якості прикладу, як об’єкт модифікації ми візьмемо добре відомий алгоритм шифрування DES, керуючись кількома причинами:

- цей алгоритм є добре вивченим та дослідженим;
- він має досить просту схему шифрування, на основі класичної мережі Фейстеля, яку можна досить просто модифікувати;
- даний алгоритм є застарілим, та вразливим до деяких сучасник атак, модифікація дозволить усунути ці вразливості.

Модифікацію алгоритму ми проведемо шляхом заміни статичної лінійної раундової криптографічної функції на динамічну нелінійну, використовуючи спосіб, що описаний раніше.

Для цього слід спочатку розробити декілька аналогів криптографічної функції, що використовується в алгоритмі шифрування, при чому сама функція не повинна змінити свою сигнатуру, а також ідентичну сигнатуру повинні мати всі створені аналоги.

Існує багато варіантів як можна змінити функцію шифрування, не змінюючи її принципову структуру. Наприклад, розглянемо лінійне криптографічне перетворення «Початкове розширення блоку», у складі даної функції. У класичній функції воно завжди виконується однаковим чином: 32-бітний блок розширяється до 48-бітного, шляхом поділу вхідного блоку на групи по 4 біта та «позичанням» двох бітів з сусідніх груп. Не змінюючи описану логіку роботи та принципову роль даного криптографічного перетворення у функції, ми можемо змінити позиції, на які стають «позичені» біти, причому зробити це можна кількома способами. Така незначна зміна не призведе до зміни логіки функціонування криптосистеми, проте змінить вихідний шифротекст (рис. 3.3).

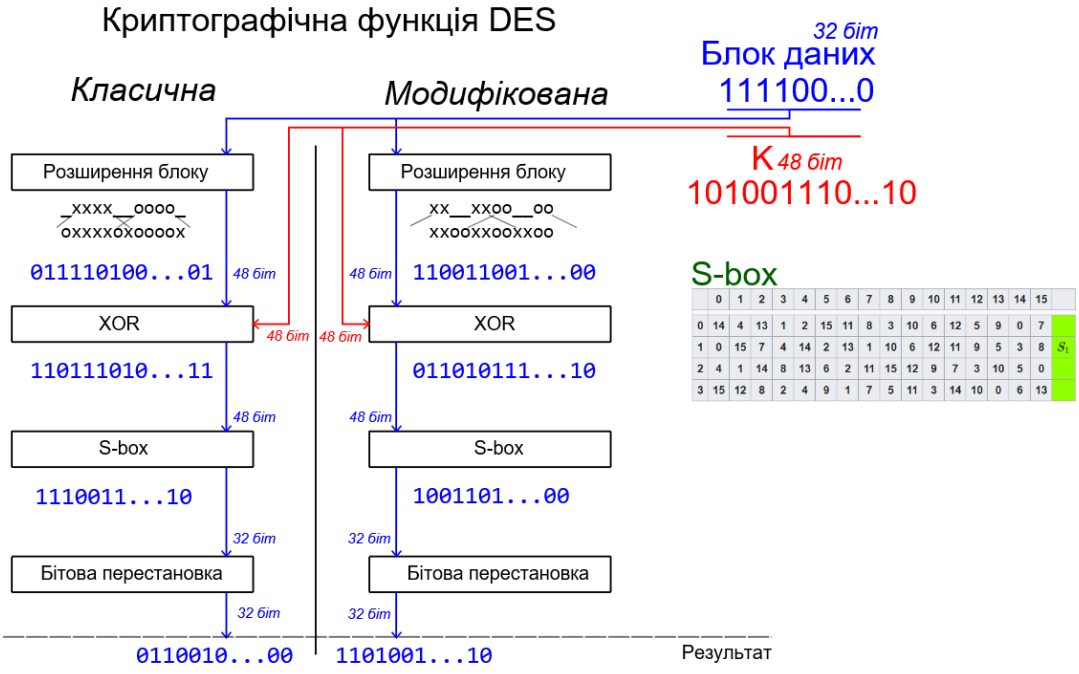


Рисунок 3.3 – Класична та один з варіантів модифікованої криптографічної функції алгоритму DES

Таким чином ми модифікували криптографічну функцію алгоритму DES, несуттєво змінивши всього лиш один лінійний криптографічний примітив в її складі.

Звісно існує безліч інших способів подібної модифікації одного або кількох складових даної функції. У будь-якому випадку, модифікуючи стандартні речі, слід уважно дивитися чи не спричинить це появу вразливостей у криптосистемі.

Модифікатор, що визначатиме яка серед альтернативних функцій використовуватиметься для даного блоку, ми, в даному випадку, братимемо випадково з діапазону $(0, n]$, де n – це кількість альтернативних функцій. Для можливості дешифрування зашифрованого тексту, використаний модифікатор слід запам’ятовувати. Ми будемо дописувати його до зашифрованого блоку (рис. 3.4).

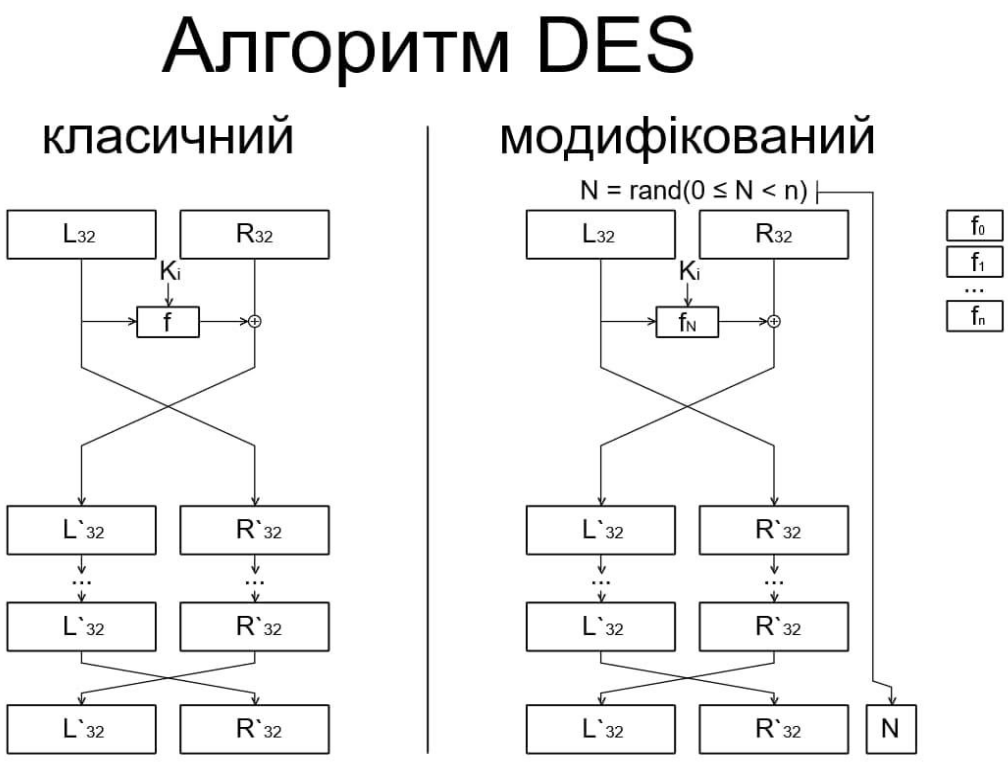


Рисунок 3.4 – Схема роботи звичайного та модифікованого алгоритму DES

Недоліком такого варіанту запису є те що криптоаналітик зможе виділити даний модифікатор та зрозуміти яка функція була використана в даному блоці.

Щоб позбавитись цього недоліку, в процесі генерації ключа кожній функції слід згенерувати один, а краще кілька, ідентифікаторів, які будуть передаватися таємно як параметри ключа.

Кількість та розмір таких ідентифікаторів залежать від кількості альтернативних функцій модифікованого алгоритму. Наприклад, якщо в нас є три функції і ми хочемо згенерувати по два ідентифікатора кожній, то їх розмір має бути мінімум 3 біта.

В такому випадку до шифрованих блоків ми дописуватимемо не публічний номер функції, а один із цих ідентифікаторів, не даючи криптографічному аналітику додаткової інформації. Недоліком такого підходу є розширення блоків шифрованого тексту.

Варто підкреслити що принцип Керкхоффса при цьому зберігається, адже всі модифіковані функції є публічними.

Наведена схема модифікації криптосистеми не є досконалою та потребує додаткових досліджень, проте вона наглядно демонструє можливість подібних модифікацій та їх вплив на систему.

Недолік від розширення шифротексту можна легко усунути, прибравши генерацію випадкових модифікаторів, передаючи їх ззовні, як це було у попередніх прикладах, або генеруючи їх, наприклад, на основі раундового ключа. В такому випадку розширення шифротексту відбуватися не буде, але й зникне ефект «випадкового шифротексту». Рішення про спосіб генерації модифікаторів слід приймати, виходячи з практичних завдань, які стоятимуть перед криптосистемою.

Отже, запропонована модифікація алгоритму DES, є введення розгалуженості у існуючу криптосистему. За рахунок незначних модифікацій вона дозволяє покращити характеристики алгоритму шифрування. У запропонованому прикладі надійність шифрування жодним чином не погіршується, оскільки він зберігає усі ключові аспекти оригінального DES, лише розширюючи його характеристики.

3.3 Розробка самостійного алгоритму нелінійного шифрування

Перейдемо до безпосередньої розробки самостійного алгоритму шифрування на основі нелінійних криптографічних перетворень.

В основі даного алгоритму лежатиме доволі проста за суттю блокова підстановка. Не дивлячись на свою простоту, даний метод шифрування є досить надійним за умови його грамотного використання.

Загальна логіка функціонування даного алгоритму буде подібною до логіки наведеної раніше блок-схеми нелінійного криптографічного алгоритму. У якості базового лінійного криптографічного перетворення ми візьмемо звичайну підстановку за S-блоком. Таких одновимірних S-блоків у нас буде велика кількість. Ми випадковим чином генеруватимемо модифікатор, у діапазоні $[0, n)$, де n – це кількість наших S-блоків.

Для ускладнення аналізу та приховування того який S-блок було використано у тому чи іншому випадку, ми скористаємось описаним у попередньому підрозділі методом: кожному S-блоку ми згенеруємо m ідентифікаторів, які використовуватимемо для «сліпого» маркування замінених блоків у шифротексті.

Усі згенеровані S-блоки, разом з їхніми ідентифікаторами, зведемо в єдину структуру, яку назвемо «ключ-таблиця» (рис. 3.5).

	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 3.5 – Ключ-таблиця

Даний алгоритм не матиме ключа у класичному криптографічному сенсі. Замість нього запропонована криптосистема матиме структуровану ключ-таблицю. Вона матиме значно більший розмір, проте це, з іншого боку, забезпечуватиме надвисоку стійкість алгоритму, оскільки, скажімо, атакою грубої сили такий ключ неможливо буде підібрати за прийнятний час навіть володіючи квантовим комп'ютером.

Дана криптосистема повинна також включати алгоритм генерації ключа, який на виході даватиме ключ-таблицю (рис. 3.6).

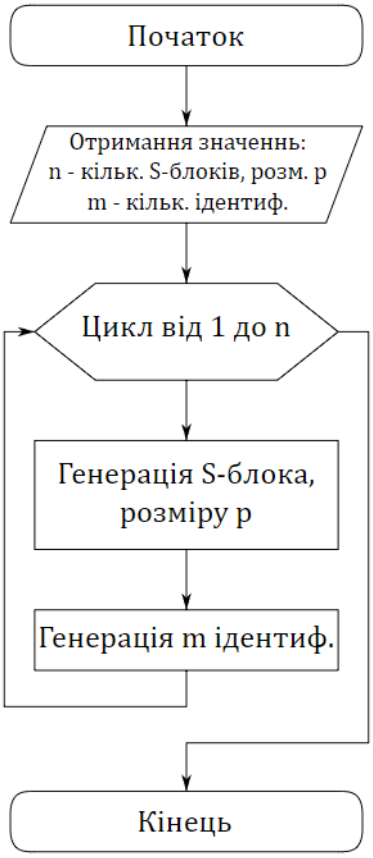
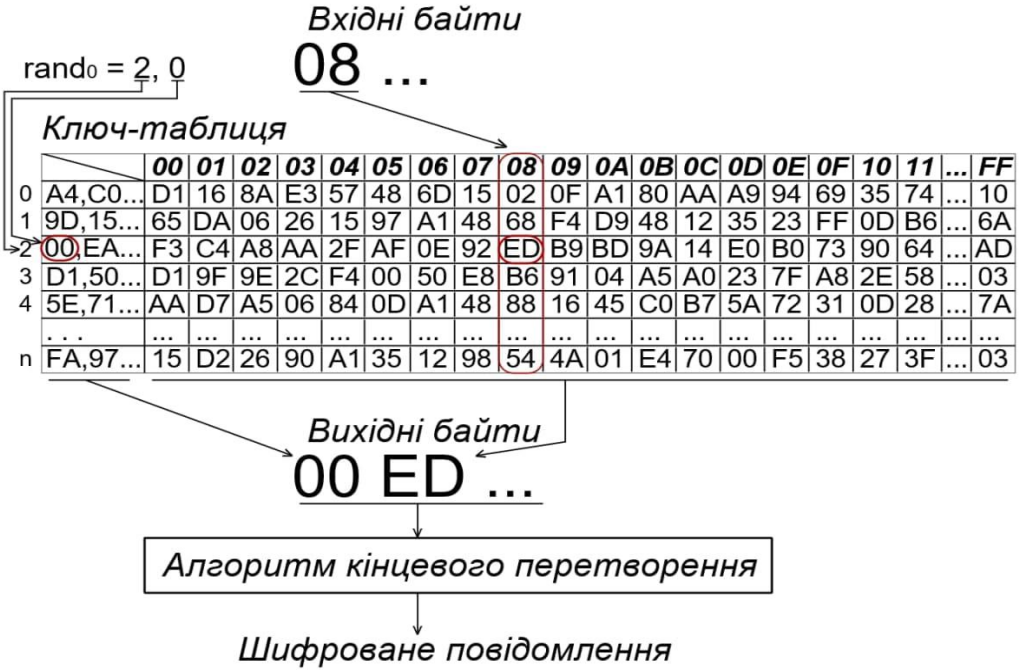


Рисунок 3.6 – Алгоритм генерації ключ-таблиці

З іншого боку розмір структурованого ключа у декілька десятків кілобайт не є жодною проблемою на сьогоднішньому рівні розвитку комп'ютерної техніки, адже, наприклад, фотографія на мобільному телефоні може займати кілька мегабайт.

Шифрування відбуватиметься шляхом нелінійної підстановки за S-блоками, відповідно до принципів нелінійного шифрування, сформованих раніше (рис. 3.7).



	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 3.7 – Схема роботи алгоритму нелінійного шифрування

Шифротекст такого алгоритму представляє пару «ідентифікатор-блок», сконкатинованих певним чином, згідно запропонованих раніше варіантів. Зчеплення цих елементів також може представляти з себе нелінійний криптографічний примітив.

Для підвищення криптостійкості після нелінійної підстановки отримані дані слід додатково обробити. Дану функцію виконує алгоритм кінцевого перетворення. Він необхідний для «затирання» ідентифікаторів та підставлених значень S-блоків.

Роль алгоритму кінцевого перетворення може виконувати два або чотири раунди класичної мережі Фейстеля. Ключ для даного алгоритму генерується на основі загальної ключ-таблиці.

Алгоритм дешифрування є дзеркальним до алгоритму шифрування.

Окремо слід нагадати що вимога криптографічної стійкості ставиться і перед алгоритмом генерації випадкових чисел, який буде використовуватись в алгоритмі шифрування.

Також, як і у випадку модифікації алгоритму DES, при випадковій генерації модифікатора, за рахунок необхідності якимось чином зберігати дане згенероване значення, шифротекст збільшується. Коефіцієнт збільшення може коригувати користувач за рахунок керування розміром елемента S-блока та розміром ідентифікатора.

Розроблена криптосистема надійно захищена від різного роду атак. Зокрема вона надійно захищена від атак грубою силою завдяки величезному по криптографічних мірках розмірах ключу, а також завдяки використанню нелінійних примітивів.

Кількість можливих варіантів S-блоків ключ-таблиці N_a можна обчислити за формулою $N_a = (2^p)! * n$, де p – розмір блоку даних у двійковому форматі, а n – кількість алфавітів.

Загальна кількість можливих ключ-таблиць N обраховується за формулою $N = N_a \times N_{id}$, де N_a – кількість варіантів S-блоків ключ-таблиці, N_{id} – кількість можливих ідентифікаторів.

Отже, навіть при 5 S-блоках, з розміром елемента 7 байт, їх кількість сягне понад $1{,}9 \times 10^{216}$, враховуючи ще кількість можливих ідентифікаторів, навіть зі нереальною швидкістю перебору у мільярд ключів за секунду, на перебір навіть однієї тисячної частини цих ключів піде принципово непрактичний час.

Даний алгоритм також стійкий перед усіма видами класичних криптоатак, таких як атака по масці, частотний та статистичний аналіз тощо.

Також не помічено жодних вразливостей перед математичним аналізом, оскільки в ньому практично відсутні будь-які алгебраїчні ітерації, а базову криптографічну складність представляє собою бітова підстановка.

Розроблена криптосистема не схожа на сучасну криптосистему у класичному розумінні, оскільки вона потребує значно більше комп’ютерної

пам’яті для зберігання та роботи, проте вона гарантує високий рівень криптографічного захисту та має свою сферу прикладного використання.

Окрім цього вона має ряд унікальних характеристик, які ще більше розширюють її практичну користь. Використовуючи дані унікальні характеристики ми спробуємо адаптувати дану криптосистему для стеганографічного використання.

3.4 Адаптація розробленого симетричного алгоритму нелінійного шифрування для стеганографічного використання

У попередніх підрозділах, зазначалось що створена криптосистема має ряд унікальних властивостей, зокрема те, що шифротексти одного й того ж повідомлення, при ідентичному ключі є різними після кожного шифрування. Це є наслідком використання нелінійних криптографічних примітивів та генерацією випадкового модифікатора в процесі шифрування.

Шифруючи ідентичні повідомлення багато разів підряд можна помітити що ймовірність повтору шифротексту вкрай мала, особливо за умови великого повідомлення та великої кількості S-блоків в ключ-таблиці. Оскільки дані S-блоки в ключ-таблиці генеруються випадково, вхідний блок даних може бути зашифрований будь-яким іншим блоком такої ж довжини.

Звідси виникає закономірне запитання, чи можна штучно створити деякий ключ, за якого деяке конкретне повідомлення, зашифроване конкретним способом, на виході утворило б шифротекст у вигляді іншого осмисленого повідомлення?

Проаналізувавши особливості даної криптосистеми, можна однозначно відповісти, що це дійсно можливо. Користуючись даною особливістю розробленого алгоритму шифрування, ми можемо адаптувати його для стеганографічного використання.

Для наглядної демонстрації принципу роботи стеганографічної версії алгоритму створимо текстову модель ключ-таблиці, в якій загальна структура

залишиться незмінною із бітовою версією, проте замість самих бітів будуть кириличні символи за українською абеткою (рис. 3.8).

Рисунок 3.8 – Схема роботи алгоритму нелінійного шифрування

Найпростіше поставлене завдання вирішується в такій послідовності дій:

- формуємо таємне повідомлення у двійковому форматі – це те що нам необхідно замаскувати;
- формуємо друге «публічне» повідомлення-контейнер у двійковому вигляді – це наш «шифротекст»;
- виконуємо звичайний алгоритм дешифрування, передавши в нього байти контейнера. В результаті цього отримуємо певний набір байт, який ми маємо отримати внаслідок підстановки;
- покроково генеруємо ключ-таблицю, на основі таємного повідомлення;
- коли ми дійшли до кінця таємного повідомлення, закінчуємо генерацію ключ-таблиці випадковими значеннями.

При практичній реалізації деяких з перелічених етапів можуть виникнути ускладнення.

По-перше, розмір повідомлення-контейнера повинен бути більшим за розмір таємного повідомлення в декілька разів.

По-друге, виникають проблеми з АКП, у випадку якщо його ключ генерується на основі ключ-таблиці, оскільки на момент дешифрування контейнера дана таблиця ще не сформована.

Вирішенням цієї проблеми може бути або відмова від алгоритму кінцевого перетворення, або використання окремого ключа для АКП. Даний ключ може бути представлений як стеганографічний пароль.

В ключ-таблиці для стеганографічного використання є сенс задавати велику довжину, а разом з цим і кількість, ідентифікаторів. Такі налаштування дозволять значно зменшити колізії значень.

Стеганографічний варіант використання розробленого алгоритму має значний потенціал прикладного використання. Ключ-таблиця, згенерована таким методом, по своїх властивостях та можливостях буде ідентичною ключ-таблиці, що згенерована звичайним алгоритмом генерації ключа, а значить віна буде придатною для звичайного криптографічного застосування.

Існує багато інших практичних видів застосування розробленій технології. Наприклад, повідомлення, зашифроване іншим алгоритмом, може використовуватись як контейнер для даного алгоритму. Таким чином один шифротекст можна буде дешифрувати двома різними алгоритмами, отримавши різні результати.

Таким чином стеганографічна адаптація алгоритму на основі нелінійних криптографічних примітивів розроблена та має багато сфер прикладного застосування.

Подібні алгоритми з отриманими особливостями та можливостями мало розповсюджені на сьогодні та мають суттєвий потенціал розвитку в майбутньому.

3.5 Висновок

Таким чином, за результатами розробки, було досягнуто успіхів, зокрема:

- модифіковано криптосистему DES, з використанням нелінійних криптографічних примітивів, в результаті чого дана система підвищила свої криптографічні характеристики та отримала нові особливості;

- розроблено самостійну криптосистему з використанням нелінійних елементів;
- експлуатуючи унікальні особливості розробленої криптосистеми на основі нелінійних криптографічних примітивів, створено її стеганографічну адаптацію.

Усі розроблені алгоритми мають в своїй основі описані раніше нелінійні криптографічні примітиви. Їх використання підвищує криптографічну стійкість та загальну захищеність алгоритмів, а також, у поєднанні з випадковою генерацією модифікатора, вони набувають нових властивостей, однією із яких є «випадковий» шифротекст.

Модифікація алгоритмів DES показує можливість та перспективність модифікації існуючих криптосистем за допомогою описаних інструментів, з метою покращення їх характеристик.

Подібні модифікації можливо розробити для будь яких інших криптосистем.

Розроблений самостійний алгоритм шифрування також показує хороші результати криптографічної стійкості, а також, що важливіше, велику гнучкість в роботі: користувач самостійно може вказати власні параметри генерації ключ-таблиці у АГК та цим самим змінювати коефіцієнт розширення шифротексту, розмір та кількість S-блоків, в залежності чого буде змінюватись і розмір самої ключ-таблиці.

Розроблений алгоритм не позбавлений недоліків, проте вони не є критичними і в повній мірі перекриваються перевагами та новими можливостями, які він дає. Зазначені недоліки мають бути додатково вивчені та мінімізовані в подальших дослідженнях на цю тему.

Стеганографічна версія розробленого алгоритму також демонструє цікаві особливості, що можуть знайти практичне використання. Зокрема, даний алгоритм фактично вирішує завдання контрольованого шифрування. Інакше кажучи, використовуючи розроблений нелінійний алгоритм шифрування дає

можливість штучно згенерувати такий ключ, зашифрувавши деяке повідомлення ним, ми отримаємо інший, попередньо заданий текст.

Розроблені алгоритми знаходяться у експериментальному стані, а отже, вони не є стандартизованими, а значить багато також залежить від рішень, прийнятих на етапі їх програмної чи апаратної реалізації.

Підсумовуючи сказане, завдання розробки виконано в повній мірі, розроблено життєздатні алгоритми на основі нелінійних криптографічних примітивів, які якісно та ефективно показали себе за теоретичною оцінкою та є перспективними для подальших досліджень. На даний момент розроблені алгоритми повністю готові до програмної реалізації.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО АЛГОРИТМУ

4.1 Загальний опис засобів та підходів програмної реалізації алгоритмів

Перед початком роботи над програмною реалізацією, опишемо інструменти, необхідні для успішного виконання завдання, а також загальні підходи, що будуть використані.

Оскільки програмні реалізації будуть створюватись під ОС Windows 10, то найбільш підходящим та зручним середовищем розробки є Microsoft Visual Studio.

Microsoft Visual Studio – це програмний комплекс що забезпечує зручну розробку, налагоджування та експлуатацію як консольних програм, так і програм із графічним інтерфейсом, у тому числі з підтримкою інтерфейсу Windows Forms, а також веб-застосунки, веб-сайти для всіх платформ, що підтримуються ОС Microsoft Windows.

Розробка буде вестись на мові програмування $C\#$, з використанням функціоналу фреймворку .NET Framework та програмного функціоналу інтерфейсу Windows Forms. Даний набір інструментів чудово підходить для успішного виконання поставлених завдань та добре підтримується у середовищі розробки Microsoft Visual Studio.

Мова програмування C# є об’єктно-орієнтованою мовою програмування. Її програмний код компілюється у виконувані файли у форматі «.exe», що забезпечує швидкодію роботи, а також ефективне використання ресурсів комп’ютера.

Розробка програмних реалізацій проводитиметься в об’єктно-орієнтованій парадигмі, з дотриманням загальноприйнятих стандартів програмування. Об’єктно-орієнтоване програмування – це універсальний підхід, який дозволяє ефективно вирішувати практично будь-які завдання. Окрім того, дотримання загальноприйнятих норм та стандартів дозволить в майбутньому іншим

програмістам працювати над даним ПЗ. Також це дає можливість працювати над одним проектом команді програмістів, не заважаючи один одному.

Програмні реалізації розроблених алгоритмів нелінійного шифрування будуть створені у вигляді динамічних бібліотек DLL, оскільки вони не прив'язані лише до одного проекту та можуть використовуватись і з іншими графічними інтерфейсами, а також в складі інших КСЗІ у ІКС, чи автоматизованих систем захисту інформації. Використання технології динамічних бібліотек є правильним, зручним та дозволяє уникнути масштабного дублювання програмного коду, адже створені таким чином інструменти криптографічного захисту можна легко використовувати у будь-яких сторонніх проектах, просто підключивши дану бібліотеку у необхідному місці та отримавши весь її публічний інструментар.

Окрім цього, в проекті буде використана система керування версіями на основі технології Git. Це технологія яка дає можливість розподіляти розробку проекту на гілки, періодично створювати контрольні точки – комміти, та в цілому бачити загальний прогрес розробки, маючи можливість повернутись до попередніх контрольних точок, а також синхронізуватись з віддаленим репозитарієм, маючи можливість працювати над одним проектом з різних точок земної кулі.

Адміністрування системою керування версіями здійснюватиметься за допомогою веб-платформи GitHub. Даний сервіс дає можливість безкоштовно створювати публічні та приватні репозитарії, керувати правами доступу та іншими налаштуваннями. Використання даного сервісу дає можливість в майбутньому опублікувати відкритий код проекту в відкритий доступ для всіх охочих.

Також розглянемо що саме ми будемо розробляти, відповідно до поставлених раніше задач.

Відповідно до сформованої мети, а також згідно поставленій у одному з попередніх розділів задачі проектування, в ході розробки ми створимо два

програмних продукти: реалізацію самостійної криптосистеми нелінійного шифрування, а також її модифікацію для стеганографічного використання.

Програмна реалізація самостійної криптосистеми нелінійного шифрування включатиме в себе реалізації алгоритми шифрування, дешифрування, генерації ключ-таблиці, об’єднані спільним графічним інтерфейсом, на основі результатів розробки з попереднього розділу, у відповідності до наведених схем. Окрім цього програмне забезпечення включатиме в себе інші алгоритми та модулі, необхідні для їх коректного функціонування, зокрема генератор випадкових чисел тощо.

Модифікація алгоритму нелінійного шифрування для стеганографічного використання включатиме в себе видозмінений алгоритм генерації ключ-таблиці, з попередньо заданими її параметрами, а також алгоритми приховування таємного повідомлення в контейнер та вилучення, на основі розроблених алгоритмів шифрування та дешифрування.

Обидві програмні реалізації будуть виконані з урахуванням можливості практичного їх використання в системах захисту сторонніми особами. З цією метою, як зазначалось раніше, програмна розробка буде вестись з дотриманням загальноприйнятих норм, писаних та неписаних правил об’єктно-орієнтованого програмування.

Окремою особливістю, на яку слід звернути увагу при розробці зазначених програмних реалізацій, є швидкодія їх роботи, адже вона є ключовою характеристикою при практичному використанні ПЗ. Різні реалізації одного й того ж алгоритму шифрування можуть мати принципово різну швидкість функціонування через різні особливості даних реалізацій. Нашим завданням є створити програми, що здатні виконувати шифрування та дешифрування за умовно прийнятний час, що може збільшуватись чи зменшуватись, в залежності від розміру вхідного повідомлення.

За результатами розробки створені програмні продукти будуть належним чином протестовані.

Перейдемо до розробки програмної реалізації самостійного алгоритму нелінійного шифрування.

4.2 Експериментальна реалізація розробленого алгоритму нелінійного шифрування

Керуючись інформацією із попередніх розділів, почнемо розробку програмної реалізації розробленого самостійного алгоритму нелінійного шифрування, використовуючи описані підходи та інструменти.

Для початку створимо новий проект у середовищі Microsoft Visual Studio. Серед усіх наявних типів проекту знайдемо тип «Бібліотека класів (.NET Framework)» та створимо новий проект даного типу з вихідним розширенням «.dll» (рис. 4.1).

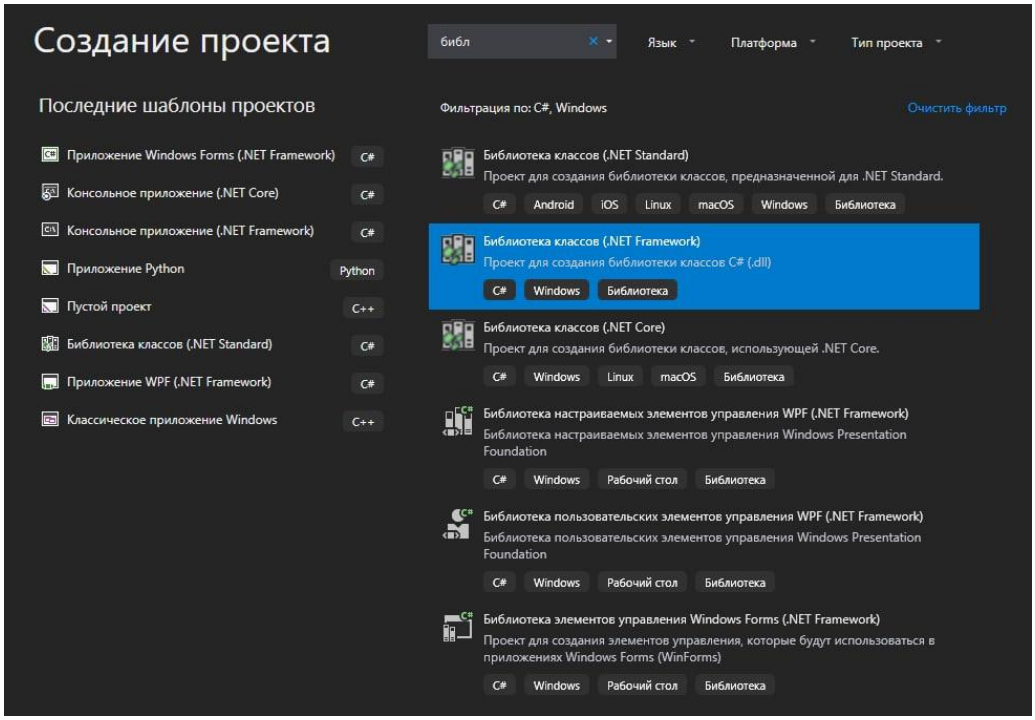


Рисунок 4.1 – Створення динамічної бібліотеки

У налаштуваннях створеного проекту оберемо останню можливу версію фреймворку, у нашому випадку це версія 4.7.2.

У створеному проекті створимо три нових класи:

- NonlinearCryptBytesBlock;
- NonlinearCryptKey;
- NonlinearCrypt;

Клас NonlinearCryptBytesBlock буде основою для всіх наступних. Він міститиме в собі два параметри: BitArray, з типом даних bool, а також Size, з типом даних int.

Окрім цього даний клас міститиме:

- конструктор з вказаними вище параметрами;
- методи-сетери з різними типами вхідних значень, які коректно задаватимуть параметри класу;
- метод «Random», який заповнюватиме даний блок випадковими значеннями;
- статичний метод «MixBlock», який буде змішувати два інших блоки за маскою.

Реалізуємо перелічені пункти у програмному коді (рис. 4.2).

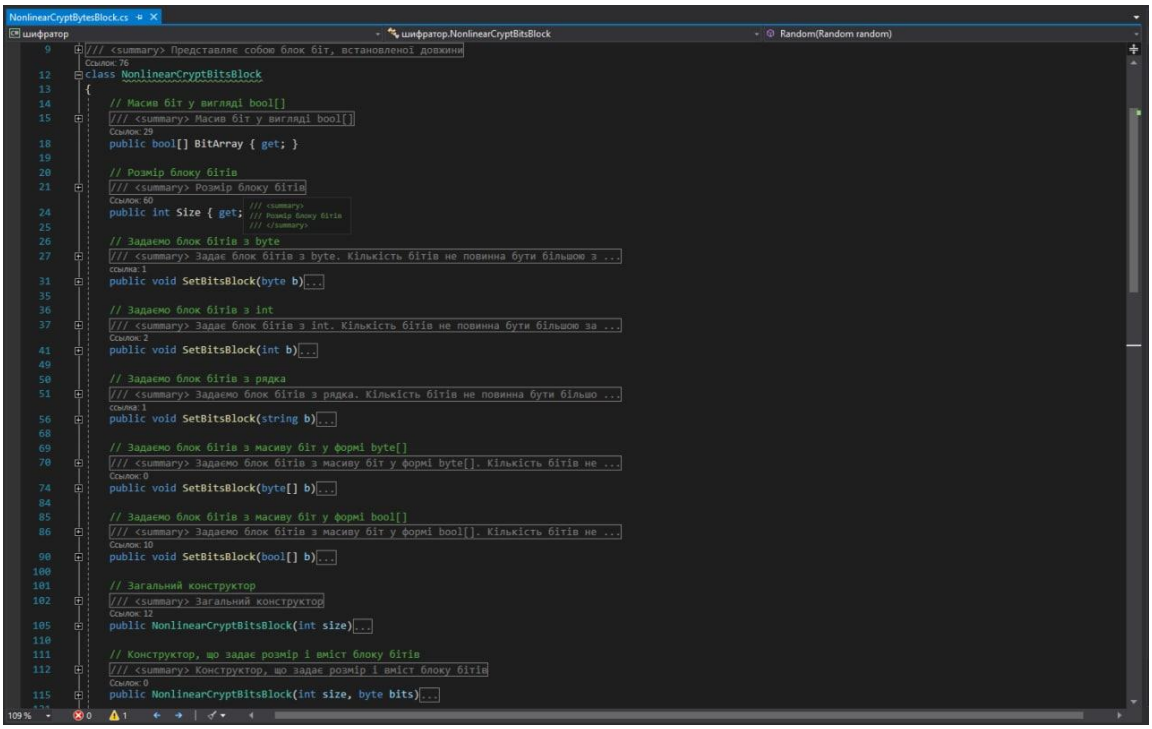


Рисунок 4.2 – Клас бітового блоку

Також, керуючись заявленими підходами розробки, притримуватимемось правил форматування, а також записуватимемо коментарі та анотації, з метою полегшення роботи з програмним кодом в майбутньому.

Клас NonlinearCryptKey – це, фактично, ключ-таблиця. Він міститиме в собі:

- параметр Alphabets, тип даних – двовимірний масив NonlinearCryptBytesBlock;
- параметр Identifiers, тип даних – двовимірний масив NonlinearCryptBytesBlock;
- параметр AlphabetsCount, тип даних – int;
- параметр DataBlockSize, тип даних – int;
- параметр IdentifiersSize, тип даних – int;
- параметр Mask, тип даних – NonlinearCryptBytesBlock;
- конструктор класу, з аргументами alphabets, identifiers та mask;
- статичний метод генерації нового ключа;
- статичний метод збереження ключа у файл;
- статичний метод завантаження ключа із файлу;
- приватні методи перевірки основних параметрів класу NonlinearCryptKey на коректність.

Як видно за переліком елементів, клас NonlinearCryptKey активно використовуватиме створений раніше клас NonlinearCryptBytesBlock у своїй роботі, що добре відповідає правилам та принципам роботи у об’єктно-орієнтованій парадигмі.

Як і в попередньому класі, при створенні притримуватимемось правил форматування, записуватимемо коментарі та анотації.

На основі вказаних даних та з урахуванням наведеної раніше блок-схеми алгоритму генерації ключ-таблиці, реалізуємо клас NonlinearCryptBytesBlock у нашому проекті динамічної бібліотеки класів (рис. 4.3).

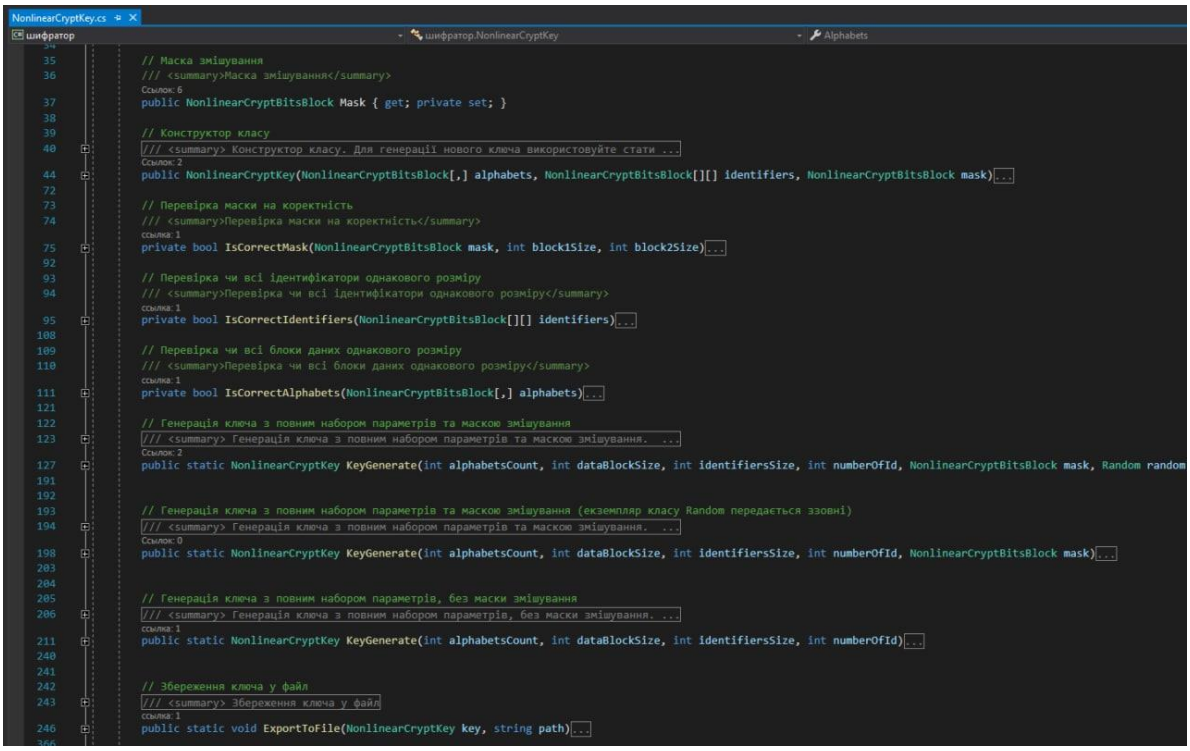


Рисунок 4.3 – Клас ключ-таблиці

Перейдемо до створення основного класу – NonlinearCrypt. Це базовий об’єкт програмної реалізації розробленого алгоритму нелінійного шифрування. Він міститиме в собі такі елементи:

- параметр OpenData, тип даних – масив байт;
- параметр CryptData, тип даних – масив байт;
- параметр Key, тип даних – NonlinearCryptKey;
- конструктор;
- метод шифрування;
- метод дешифрування;
- приватні технічні методи: поділ на блоки, кінцеве перетворення, зворотне кінцеве перетворення тощо.

Фактично, сторонній користувач даної динамічної бібліотеки класів повинен працювати лише з цим класом та його публічними методами, а також з статичними методами класу ключ-таблиці, а все інше лише забезпечує нормальне функціонування цих класів.

Аналогічно до інших класів, реалізуємо клас NonlinearCrypt (рис. 4.4).

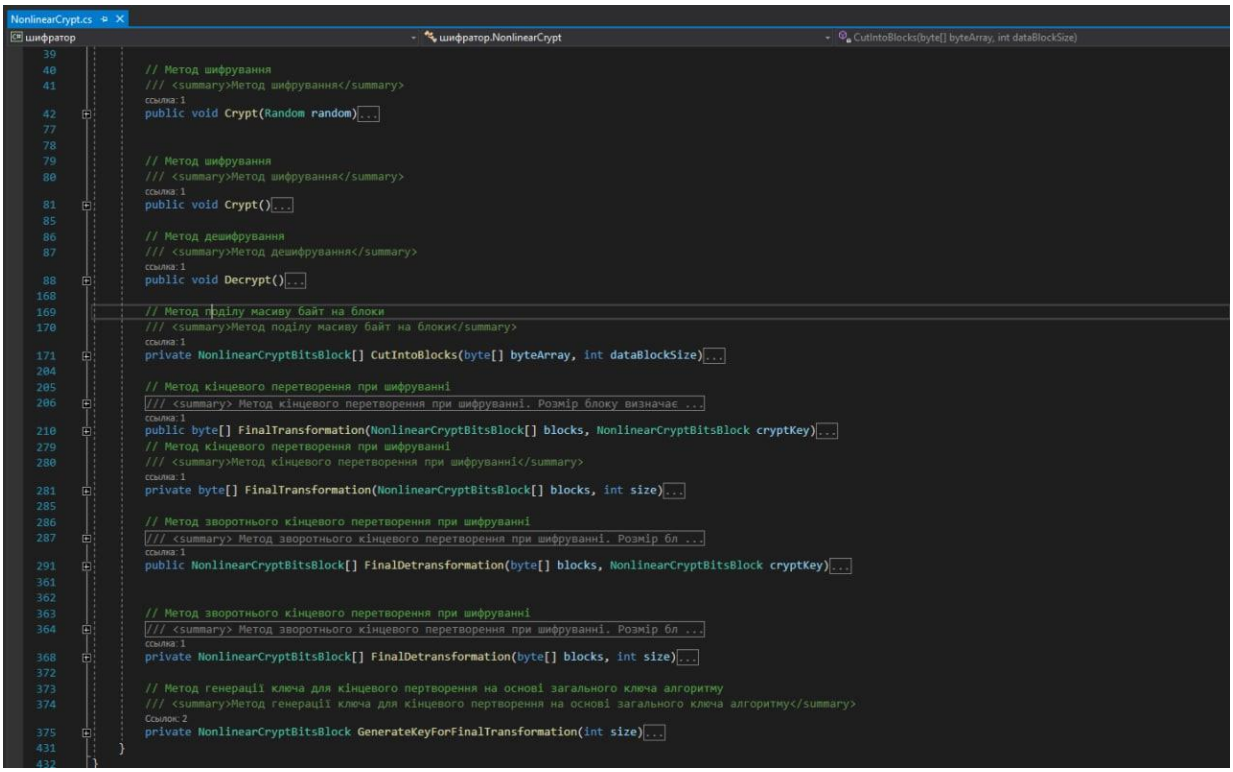


Рисунок 4.4 – Основний клас бібліотеки

Таким чином ми повністю реалізували увесь необхідний функціонал програмної реалізації алгоритму нелінійного шифрування.

Для можливості наочного використання розробленої динамічної бібліотеки класів, створимо простий графічний інтерфейс, з використанням програмного інтерфейсу Windows Forms. Для цього створимо окремий проект у Microsoft Visual Studio, вибравши відповідний тип.

Графічний інтерфейс буде складатись із наступних елементів:

- текстові поля відкритого та шифрованого тексту;
- поля вибору відкритого та шифрованого файлів;
- перемикачі файлового та текстового режимів роботи;
- кнопки виконання шифрування та дешифрування;
- поле вибору та завантаження ключа;
- таймер виконання операції.

Створимо нову форму, додавши на неї перераховані елементи. Підключимо до проекту динамічну бібліотеку класів, що була розроблена раніше. Прив’яжемо відповідні дії до тригерів натискання кнопок (рис. 4.5).

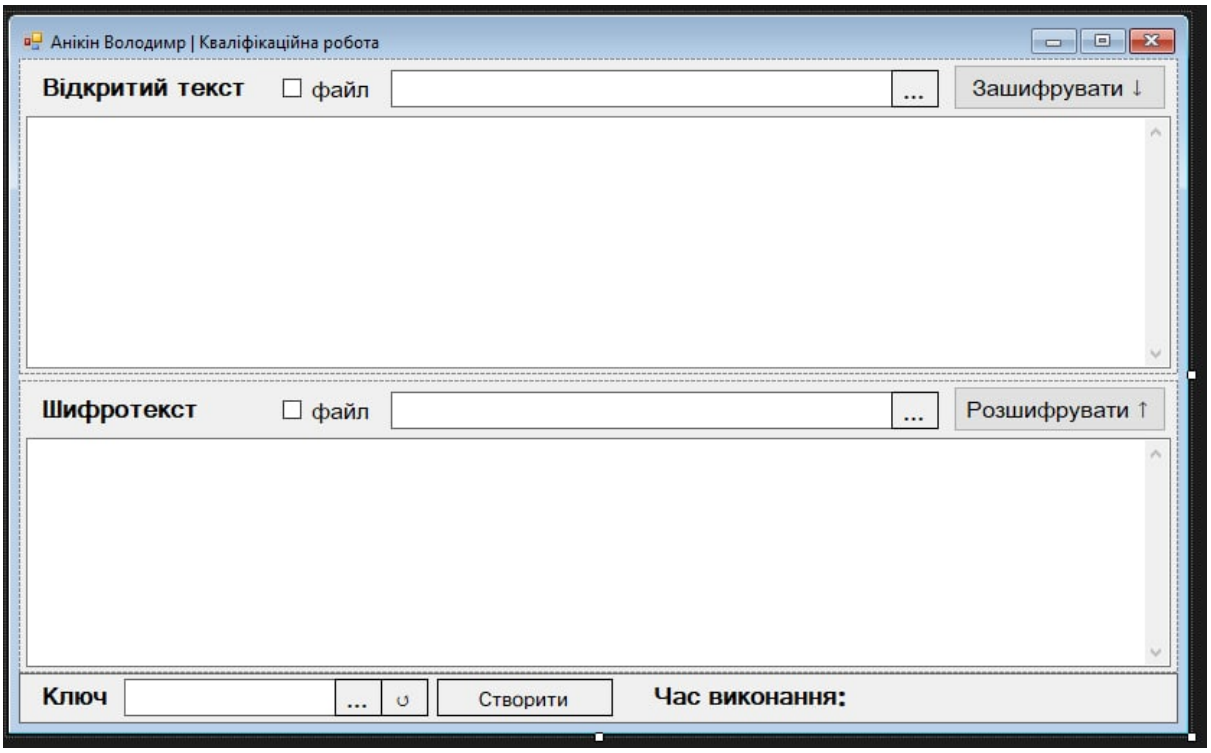


Рисунок 4.5 – Графічний інтерфейс шифратора

Отже, ми завершили розробку програмної реалізації розробленого алгоритму нелінійного шифрування, в повній мірі виконавши усі поставлені завдання. Розроблена програмна реалізація має чітку структуру, виконана у об’єктно-орієнтованій парадигмі та може легко використовуватись у будь-яких сторонніх проектах.

Перейдемо до розробки програмної реалізації стеганографічної складової розробленого алгоритму.

4.3 Програмна реалізація стеганографічної складової розробленого алгоритму

Програмна реалізація стеганографічної модифікації алгоритму нелінійного шифрування, в цілому, буде подібна до реалізації звичайного алгоритму нелінійного шифрування, описаного вище.

Для її розробки ми також створимо окрему динамічну бібліотеку класів у середовищі Microsoft Visual Sudio.

Основними відмінностями звичайної та стеганографічної реалізації алгоритму нелінійного шифрування є:

- статичний розмір блоку ключ-таблиці в стеганографічній версії;
- відсутність алгоритму кінцевого перетворення в стеганографічній версії;
- різні алгоритми генерації ключа.

Основними елементами стеганографічного класу є:

- приватні параметри active_id, active_rows та active_skip_bytes;
- метод KeyGenerate_s;
- метод дешифрування;
- інші технічні методи.

Розробимо описаний клас (рис. 4.6).

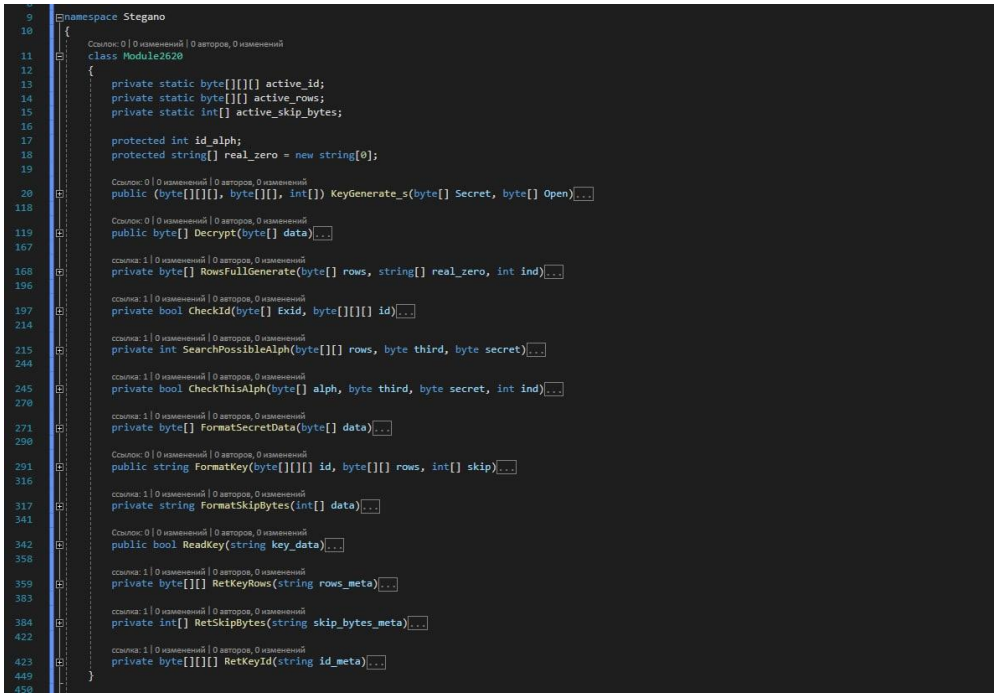


Рисунок 4.6 – Основний клас стеганографічної бібліотеки

Як і у попередньому випадку, розробимо графічний інтерфейс для створеної бібліотеки у вигляді окремого проекту, з використанням програмного інтерфейсу Windows Forms.

Графічний інтерфейс стеганографічної програми буде складатись із наступних елементів:

- текстові поля секретного повідомлення та тексту-контейнера;
- меню вибору ключів;
- кнопка генерації створеного ключа;
- кнопка виконання дешифрування контейнера.

Як і у випадку реалізації звичайного алгоритму нелінійного шифрування, створимо нову форму, додавши на неї перераховані елементи. Підключимо до проекту графічного інтерфейсу стеганографічну динамічну бібліотеку класів. Встановимо відповідні обробники подій на кнопки, користуючись методами із підключеної бібліотеки (рис. 4.7).

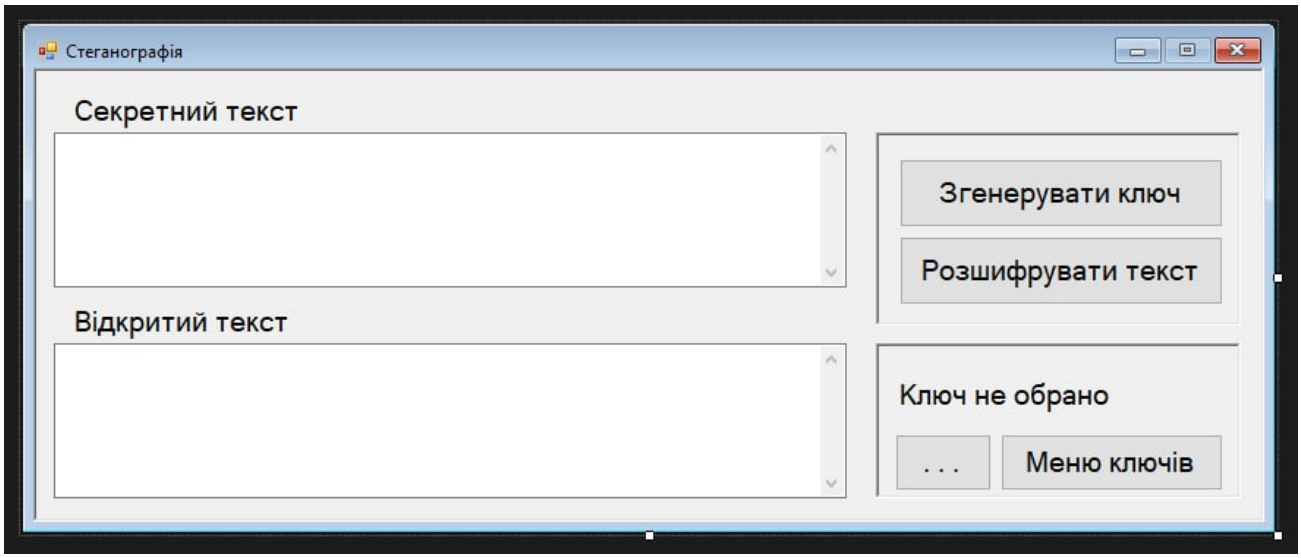


Рисунок 4.7 – Графічний інтерфейс стеганографічної утиліти

Таким чином ми завершили розробку програмної реалізації стеганографічного варіанту алгоритму нелінійного шифрування, подібно до реалізації звичайного алгоритму дана утиліта виконана на основі динамічної бібліотеки класів, у об’єктно-орієнтованій парадигмі та може використовуватись у будь-яких сторонніх проектах.

Усі можливості цієї та попередньої програмної реалізації ми зможемо детально вивчити на етапі тестування.

4.4 Тестування програмних реалізацій

Тестування створеного ПЗ є однією з найбільш важливих складових будь-якої розробки. Ми протестуємо розроблені реалізації алгоритму симетричного нелінійного шифрування, а також його стеганографічну модифікацію, за трьома критеріями:

- можливість коректного функціонування за призначенням;
- швидкодія;
- наявність помилок виконання.

Розпочнемо з реалізації звичайного алгоритму нелінійного шифрування. Для перевірки коректності його функціонування, виконаємо таку послідовність дій:

- згенеруємо новий ключ;
- зашифруємо зображення у форматі JPEG створеним ключем;
- розшифруємо зображення;
- порівняємо вхідний та вихідний файл.

Файл зображення є більш чутливим та зміна хоча б одного біта може призвести до неможливості відображення його вмісту, тому така перевірка буде достатньо об’єктивною.

Відсутність сторонніх повідомлень, зависань чи інших несправностей, разом із коректним функціонуванням за призначенням, свідчитиме про відсутність помилок виконання.

Для більшої надійності перевірки ми запишемо ключове слово «диплом» у мета-дані файлу. Наявність ключового слова в мета-даних дешифрованого файлу додатково свідчитиме про загальну коректність функціонування розробленої програмної реалізації.

Виконаємо вказані дії, прослідкуємо за ходом їх виконання та порівняємо значення вхідного та вихідного файлу (рис. 4.8).

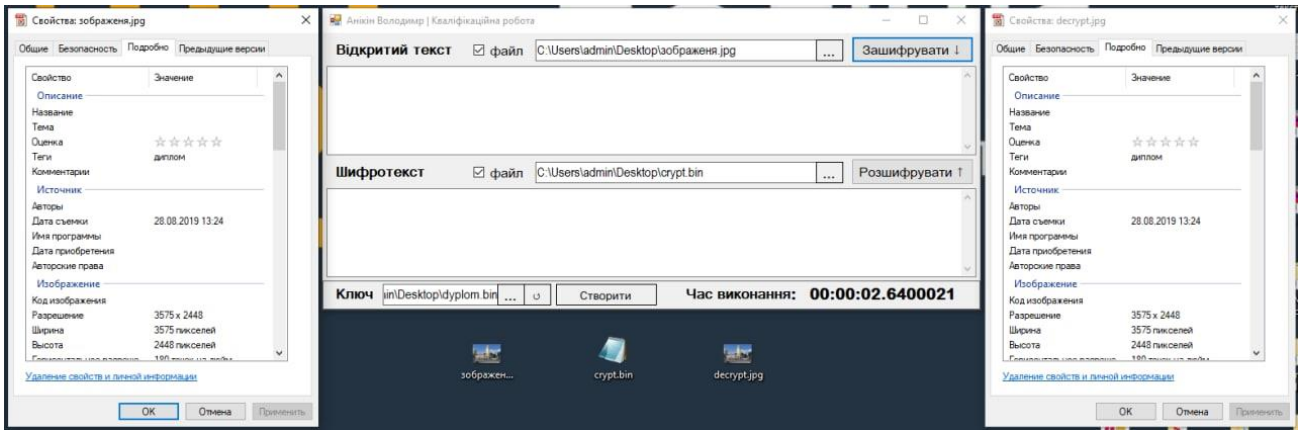


Рисунок 4.8 – Тестування програми-шифратора

Усі операції пройшли без помилок, вхідний та вихідний файл фактично ідентичні між собою. Це свідчить про відсутність у програмі серйозних недоліків.

Операція шифрування виконалась за 2.64 секунди, що є задовільною швидкістю для подібної реалізації. Таким чином, можемо констатувати що даний програмний продукт повністю та в повній мірі виконує критерії нашого тестування.

Перейдемо до перевірки стеганографічної адаптації алгоритму нелінійного шифрування.

Оскільки перед реалізацією даного алгоритму стоять інші вимоги то і критерії тестування, частково, будуть іншими. Критерії тестування даної утиліти залишаться такими ж як і у попередньому випадку, за винятком швидкодії, яку ми перевіряти не будемо, оскільки даний критерій менш значущий для даного алгоритму.

Проведемо тестування стеганографічної модифікації у аналогічній послідовності:

- задамо деякий текст-контейнер;
- задамо таємне повідомлення;
- згенеруємо новий ключ, на основі введених значень;
- зіб’ємо задані значення та дешифруємо контейнер згенерованим ключем;

- порівняємо вхідний та вихідний текст.

Виконаємо вказану послідовність дій (рис. 4.9).

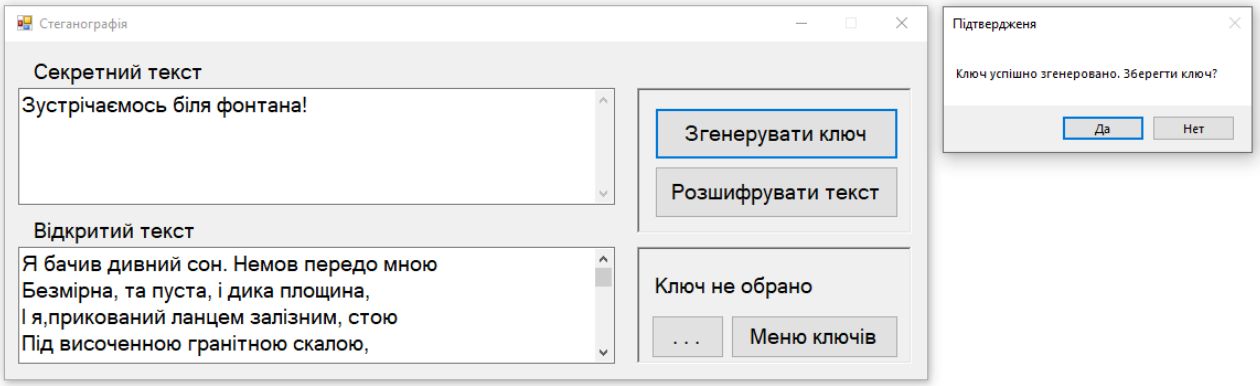


Рисунок 4.9 – Тестування стеганографічної програми

Стеганографічна утиліта також без будь-яких проблем справилось із поставленими завданнями: ключ було швидко згенеровано, а також програма з легкістю дешифрувала контейнер, за допомогою згенерованого ключа. Значення початкового та отриманого таємного повідомлення повністю співпадають на бітовому рівні.

Таким чином, обидві із розроблених програм успішно пройшли тестування, відповідно до встановлених критеріїв та, в цілому, добре показали себе в роботі.

4.5 Висновок

У цьому розділі ми проаналізували та провели опис засобів та підходів програмної реалізації алгоритмів, на основі чого виконали безпосередню розробку самостійного алгоритму симетричного нелінійного шифрування, та супутніх йому алгоритмів, а також стеганографічну модифікацію даного алгоритму. Програмні реалізації базувались на теоретичному підґрунті, дослідженому у попередніх розділах, було використано наведені раніше блок-схеми та матеріали.

Усі зазначені програмні продукти були створені на основі технології динамічних бібліотек класів та були під’єднані до окремих проектів з реалізацією графічного інтерфейсу для створення можливості їх зручної експлуатації.

Розроблене ПЗ було протестовано та добре показало себе в роботі. За результатами тестувань було не було виявлено жодних суттєвих недоліків.

За результатами розробки програмних реалізацій створених алгоритмів, сформовані раніше мета та завдання кваліфікаційної роботи виконані в повному обсязі.

ВИСНОВКИ

За результатом дослідження, у відповідності до мети кваліфікаційної роботи, у повній мірі було виконано увесь перелік завдань, зокрема:

- було досліджено та проаналізовано предметну область, в ході чого, серед іншого, було проведено аналіз існуючих симетричних, вивчено їх переваги та недоліки;
- досліджено вплив та особливості нелінійного шифрування;
- розроблено систему модифікації існуючих криптосистем, з використанням нелінійних криптографічних примітивів;
- також спроектовано та програмно реалізовано окрему нелінійну симетричну криптосистему з можливістю як криптографічного, так і стеганографічного режиму роботи.

Розроблені програмні реалізації нелінійних алгоритмів шифрування успішно пройшли тестування, в ході якого не було виявлено жодних критичних неполадок.

Метою кваліфікаційної роботи було створення криптостійкого симетричного нелінійного алгоритму шифрування з можливістю комбінованого криптографічно-стеганографічного використання.

У відповідності до цього, можемо сказати, що мета кваліфікаційної роботи досягнута в повному обсязі.

В ході роботи ми проаналізували предметну область, дослідили розвиток та сучасний стан криптографічного та стеганографічного захисту інформації в цілому, та проаналізували найбільш популярні блокові симетричні криптосистеми та методи їх криптографічного аналізу.

Також було досліджено поняття нелінійного шифрування, порівняно лінійні та нелінійні криптографічні примітиви а також оцінено вплив останніх на загальні характеристики криптосистем.

На основі цього було розроблено модифікацію криптосистеми DES, з використанням нелінійних криптографічних примітивів, спроектовано

самостійний алгоритм нелінійного шифрування, який повністю базується на схемі нелінійності, а також модифікували його для стеганографічного використання.

Розроблені в даній роботі теоретичні алгоритми та їх програмні реалізації мають цікаві унікальні властивості, є перспективними та готові до впровадження у будь-яку КСЗІ у ІКС, чи іншу автоматизовану систему захисту. Розроблені утиліти виконують закладені у них функції, а також мають прийнятну швидкодію. Програма-шифратор, наприклад, в ході тестування показала швидкість роботи близько 524 288 байт за секунду, зашифрувавши та дешифрувавши файл зображення в форматі «JPEG» безпомилково.

Висвітлена в кваліфікаційній роботі тематика має перспективу подальших досліджень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

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

2) 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: 30.05.2022)

3) BELLARE, Mihir; ROGAWAY, Phillip. Introduction to modern cryptography. Ucsd Cse, 2005, 207: 207.

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

7) ШЕВЧЕНКО, Олексій Тарасович. Застосування квантових алгоритмів Саймона та Бернштейна-Вазірані для криптоаналізу узагальненої мережі Фейстеля. 2019. Bachelor's Thesis. КПІ ім. Ігоря Сікорського.

8) 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: 30.05.2022).(нормативний документ)

9) KADHIM, Inas Jawad, et al. Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. Neurocomputing, 2019, 335: 299-326.

10) MORKEL, Tayana; ELOFF, Jan HP; OLIVIER, Martin S. An overview of image steganography. In: ISSA. 2005. p. 1-11.

11) SUBRAMANIAN, Nandhini, et al. Image steganography: A review of the recent advances. IEEE Access, 2021.

12) 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. 12-19.

13) Карпінський, М. П., Кінах, Я. І., Яциковська, У. О., Паславський, Р. І., Кужда, Т. І., & Бойко, І. В. (2020). Експлуатація багатокористувацької програмної системи для криптоаналізу асиметричних алгоритмів шифрування даних. Матеріали Міжнародної науково-технічної конференції „Фундаментальні та прикладні проблеми сучасних технологій “до 60-річчя з дня заснування Тернопільського національного технічного університету імені Івана Пулюя та 175-річчя з дня народження Івана Пулюя, 155-156.

14) Смірнова, Т. В., Константинова, Л. В., Смірнов, С. А., Якименко, Н. М., & Смірнов, О. А. Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах.

15) 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.

16) Деркач, О. Перевірка Модельних Припущень у Криптоаналізі Arx-Шифрів.

17) Harmash, D. V. (2021). Властивості багатовимірного алгоритму Rainbow та його здатність протистояти різноманітним методам криптоаналізу і атаці сторонніми каналами. Radiotekhnika, (205), 79-84.

18) 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

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

20) Халімов, Г. З., Котух, Є. В., Сергійчук, Ю. О., & Марухненко, О. С. (2018). Аналіз складності реалізації криптосистеми на групі Судзукі. Radiotekhnika, (193), 75-81.

ДОДАТОК А

Програмний код реалізацій нелінійних алгоритмів

```
// Метод шифрування
        /// <summary>Метод шифрування</summary>
        public void Crypt()
        {
            Crypt(new Random());
        }
        // Метод дешифрування
        /// <summary>Метод дешифрування</summary>
        public void Decrypt()
        {
            if (CryptData == null)
            {
                throw new Exception("Шифровані дані не задано");
            }
            if (Key == null)
            {
                throw new Exception("Ключ не задано. Передайте існуючий ключ, або згенеруйте новий");
            }
            NonlinearCryptBitsBlock[] cryptBitsBlocks = FinalDetransformation(CryptData, 64);
            NonlinearCryptBitsBlock[] openBitsBlocks = new NonlinearCryptBitsBlock[cryptBitsBlocks.Length];
            for (int m = 0; m < cryptBitsBlocks.Length; m++)
            {
                NonlinearCryptBitsBlock[] blocks = NonlinearCryptBitsBlock.DemixBlocks(cryptBitsBlocks[m], Key.Mask);
```

```
            for (int i = 0; i < Key.Identifiers.Length; i++)
            {
               for (int j = 0; j < Key.Identifiers[i].Length; j++)
               {
                  if (Key.Identifiers[i][j].Equals(blocks[0]))
                  {
                     for (int k = 0; k < Math.Pow(2, (double)Key.DataBlockSize);
k++)
                     {
                        if (Key.Alphabets[i, k].Equals(blocks[1]))
                        {
                           openBitsBlocks[m] = new
NonlinearCryptBitsBlock(Key.DataBlockSize, k);

                           j = Key.Identifiers[i].Length;
                           i = Key.Identifiers.Length-1;
                           break;
                        }
                     }
                  }
               }
            }
         }
         bool[] out_ = new bool[openBitsBlocks.Length * Key.DataBlockSize];
         for (int i = 0; i < openBitsBlocks.Length; i++)
         {
            for (int j = 0; j < Key.DataBlockSize; j++)
```

```
    {
        out_[i * Key.DataBlockSize + j] = openBitsBlocks[i].BitArray[j];
    }
}
int addBits = 0;
bool[] temp_ab = new bool[8];
Array.Copy(out_, out_.Length - 8, temp_ab, 0, 8);
for (int i = 0; i < temp_ab.Length; i++)
{
    if (temp_ab[i])
        addBits |= 1;
    addBits <<= 1;
}
addBits >>= 1;
Array.Resize(ref out_, out_.Length - addBits);
byte[] out_b = new byte[(int)Math.Ceiling((double)out_.Length / 8)];
for (int i = 0; i < out_b.Length; i++)
{
    for (int j = 0; j < 8; j++)
    {
        if ((i * 8 + j) < out_.Length)
        {
            if (out_[i * 8 + j])
                out_b[i] |= 1;
        }
        if (j != 7)
            out_b[i] <<= 1;
```

```
            }
        }
        OpenData = out_b;
    }



// Генерація ключа з повним набором параметрів та маскою змішування
    /// <summary>
    /// Генерація ключа з повним набором параметрів та маскою змішування.
    /// Значення повинні бути додатніми
    /// </summary>
    public static NonlinearCryptKey KeyGenerate(int alphabetsCount, int
dataBlockSize, int identifiersSize, int numberOfId, NonlinearCryptBitsBlock
mask, Random random)
    {
        if (alphabetsCount <= 0  dataBlockSize <= 0  identifiersSize <= 0 ||
numberOfId < 0)
        {
            throw new Exception("Не додатні значення не допустимі");
        }


        // Генерація алфавітів
        int blocksCount = (int)Math.Pow(2, (double)dataBlockSize);
        NonlinearCryptBitsBlock[,] alphabets = new
NonlinearCryptBitsBlock[alphabetsCount, blocksCount];
        for (int i = 0; i < alphabetsCount; i++)
        {
            for (int j = 0; j < blocksCount; j++)
            {
```

```
            bool isFind = false;
            NonlinearCryptBitsBlock block = null;
            while (!isFind)
            {
                isFind = true;
                // Генеруємо новий блок в алфавіті
                block = new NonlinearCryptBitsBlock(dataBlockSize);
                block.Random(random);
                // Перевіряємо чи даний блок вже не використано в даному
алфавіті
                for (int l = 0; l < j; l++)
                    if (alphabets[i, l].Equals(block))
                    {
                        isFind = false;
                        break;
                    }
            }
            alphabets[i, j] = block;
        }
    }

    // Генерація ідентифікаторів
    NonlinearCryptBitsBlock[][] identyfiers = new
NonlinearCryptBitsBlock[alphabetsCount][];
    for (int i = 0; i < alphabetsCount; i++)
    {
        identyfiers[i] = new NonlinearCryptBitsBlock[numberOfId];
        for (int j = 0; j < numberOfId; j++)
```

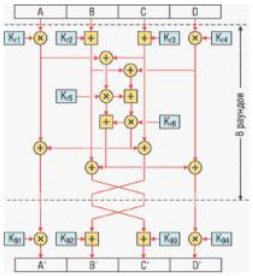
```
            {
                bool isFind = false;
                NonlinearCryptBitsBlock block = null;
                while (!isFind)
                {
                    isFind = true;
                    // Генеруємо новий ідентифікатор
                    block = new NonlinearCryptBitsBlock(identifiersSize);
                    block.Random(random);
                    // Перевіряємо чи даний ідентифікатор вже не використовується
                    for (int l = 0; l <= i; l++)
                        for (int m = 0; m < numberOfId; m++)
                            if (identyfiers[l][m] != null && identyfiers[l][m].Equals(block))
                            {
                                isFind = false;
                                l = i + 1;
                                break;
                            }
                }
                identyfiers[i][j] = block;
            }
        }
        return new NonlinearCryptKey(alphabets, identyfiers, mask);
    }
```

ДОДАТОК Б

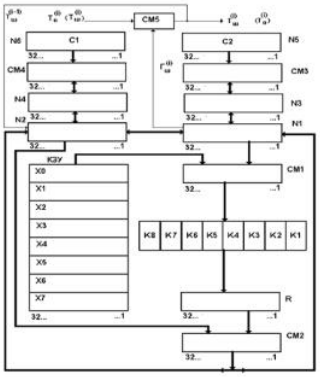
Копія графічної частини

КРКБ.180125.18.01.01 Е8

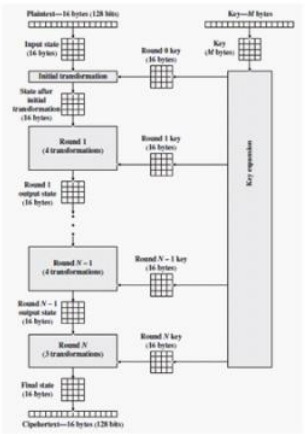
Алгоритм IDEA



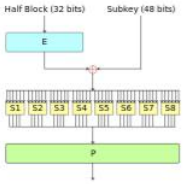
Алгоритм ДСТУ ГОСТ 28147:2009



Алгоритм AES



Алгоритм DES



КРКБ.180125.18.01.01 Е8

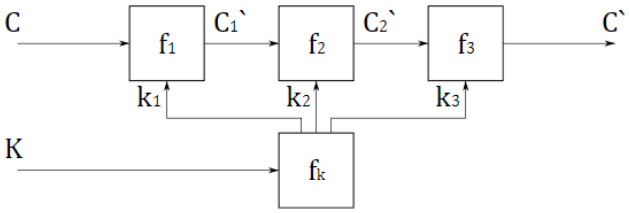
Криптографічна система надійного шифрування з можливістю стеганографічного застосування

Алгоритми блокові шифри

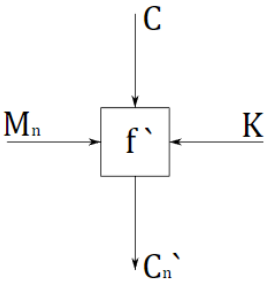
ХНУ, КБ-18-1

КРКБ.180125.18.01.01 Е8

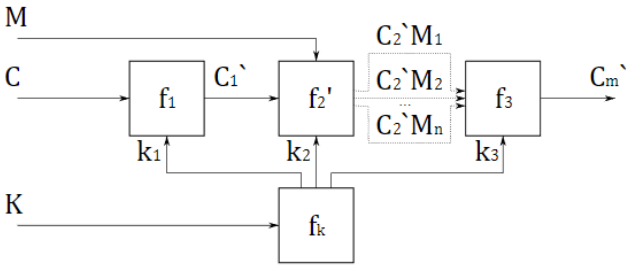
Нелінійний криптографічний примітив



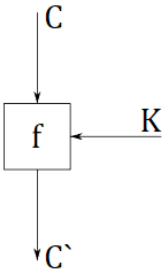
Лінійна криптографічна система



Лінійний криптографічний примітив

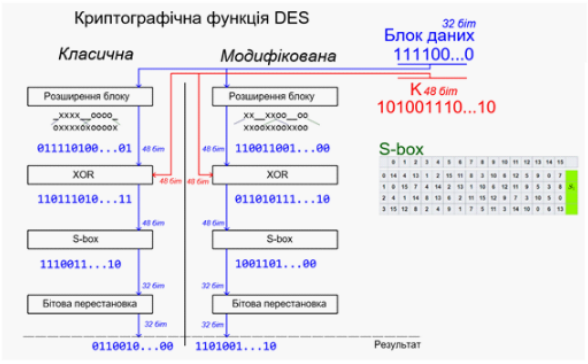


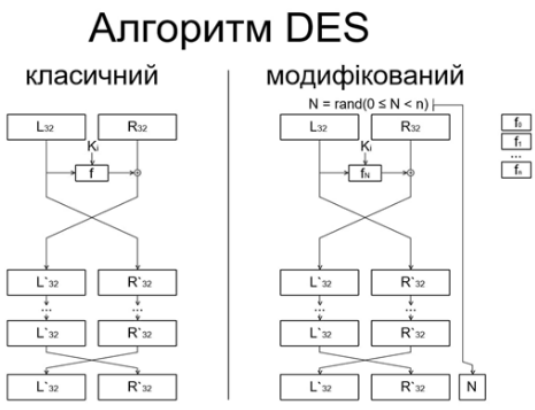
Нелінійна криптографічна система



КРКБ.180125.18.01.01 Е8			
Криптографічна система нелінійного шифрування з послідовністю отримання/графічною застосування. Лінійні та нелінійні криптопримітиви			
Розроб. Авілін В.А.	Літера	Маса	Масштаб
Перевір. Мулер І.В.	Н		
Т. контр.	Аркуш	Аркушів 1	
Н. контр. Мостовий С.В.	ХНУ, КБ-18-1		
Затв. Кільєц Ю.П.			

КРКБ.180125.18.01.01 Е8





Блок А 11111111

Блок Б 00000000

Варіанти зчеплення блоків:

- просте зчеплення 1111111100000000
- змішування 1010101010101010
- кільцеве 1111000000001111
- перехрестне 1111000011110000
- комбіноване 1010101011110000

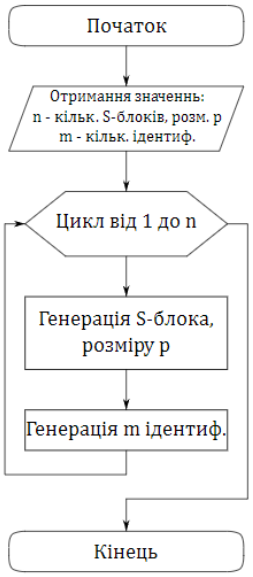
КРКБ.180125.18.01.01 Е8			
Криптографічна система нелінійного шифрування з послідовністю отримання/графічною застосування. Модифікація DES			
Розроб. Авілін В.А.	Літера	Маса	Масштаб
Перевір. Мулер І.В.	Н		
Т. контр.	Аркуш	Аркушів 1	
Н. контр. Мостовий С.В.	ХНУ, КБ-18-1		
Затв. Кільєц Ю.П.			

КРКБ.180125.18.01.01 Е8

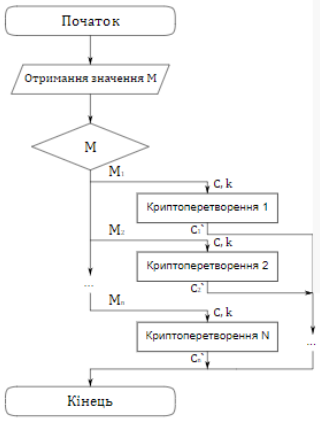
Ключ-таблиця

	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

Алгоритм генерації ключа



Логіка нелінійного шифрування

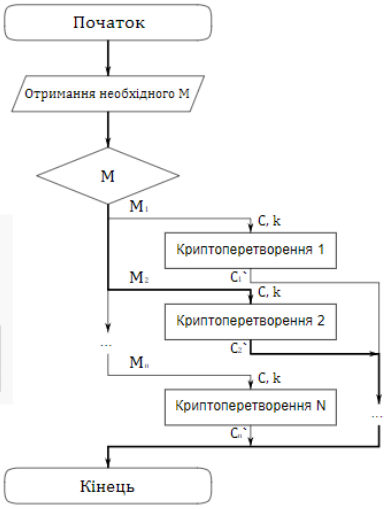


	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	FF
0	A4,C0...	D1	16	8A	E3	57	48	6D	15	02	0F	A1	80	AA	A9	94	69	35	74	...	10
1	9D,15...	65	DA	06	26	15	97	A1	48	68	F4	D9	48	12	35	23	FF	0D	B6	...	6A
2	00,EA...	F3	C4	A8	AA	2F	AF	0E	92	ED	B9	BD	9A	14	E0	B0	73	90	64	...	AD
3	D1,50...	D1	9F	9E	2C	F4	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

				КРКБ.180125.18.01.01 Е8			
Розроб.	Авдєєв В.А.			Криптографічна система нелінійного шифрування з неоднозначністю стеганографічного застосування. Алгоритм нелінійного шифрування	Літера	Маса	Масштаб
Перевір.	Мулява І.В.				Н		
Н. контр.	Мостовий С.В.				Аркуш	Аркушів 1	
Затв.	Кільош Ю.П.				ХНУ, КБ-18-1		

КРКБ.180125.18.01.01 Е8

Логіка управління шифротекстом



				КРКБ.180125.18.01.01 Е8			
Розроб.	Авдєєв В.А.			Криптографічна система нелінійного шифрування з неоднозначністю стеганографічного застосування. Стеганографічна версія алгоритму	Літера	Маса	Масштаб
Перевір.	Мулява І.В.				Н		
Н. контр.	Мостовий С.В.				Аркуш	Аркушів 1	
Затв.	Кільош Ю.П.				ХНУ, КБ-18-1		

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: Криптографічна система нелінійного шифрування з можливістю стеганографічного застосування

Автор Анікін Володимир Андрійович
Спеціальність: 125 – Кібербезпека
Освітня програма: освітньо-професійна
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 94.58%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 100.00%

Згідно з Положенням про дотримання академічної доброчесності в Хмельницькому національному університеті (http://www.khnu.km.ua/root/files/01/10/03/0005.pdf) така авторська робота, обсяг оригінального тексту у відсотках до загального обсягу матеріалу в якій складає 90-100 %, визнається роботою з високою унікальністю тексту.

Керівник роботи Ігор Муляр

Гарант ОП Віктор Чешун

Завідувач кафедри КБ Юрій Кльоц

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Анікін Володимир Андрійович

Тема *Криптографічна система нелінійного шифрування з можливістю стеганографічного застосування*

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 5; кількість сторінок записки 67.

1. Короткий зміст роботи та прийнятих рішень
в кваліфікаційній роботі досліджуються симетричні криптосистеми нелінійного шифрування, можливість їх адаптації для стеганографічного використання. За результатами роботи створюється прикладне програмне забезпечення, відповідно до теоретичної основи, з можливістю використання у комплексних системах захисту інформації.

2. Висновок про відповідність кваліфікаційної роботи завданню *Кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині*

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: *У першому розділі було досліджено та проаналізовано суміжну предметну область, у другому розділі було обґрунтувано використання нелінійного шифрування задля усунення недоліків існуючих симетричних криптосистем, у третьому розділі було розроблено вдосконалений алгоритм нелінійного шифрування з можливістю стеганографічного використання, програмно реалізувати розробленого алгоритму*

4. Позитивні сторони роботи
В роботі запропоновано новий та цікавий підхід до криптографічного захисту інформації, а також стеганографічні інтерпритації криптографічних алгоритмів

5. Негативні сторони роботи роботи

Запропоновані алгоритми шифрування не є сертифікованими та потребують подальшого вивчення та сертифікації перед початком їх практичної експлуатації

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В цілому кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження .

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «відмінно».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Нічепорук Андрій Олександрович, к.т.н., доцент кафедри комп'ютерної інженерії та інформаційних систем

«08» червня 2022. (підпис)

UNICHECK

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1011504774

Дата перевірки:
08.06.2022 12:57:31 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.06.2022 13:16:58 EEST

ID користувача:
100008300

Назва документа: **АнікінКваліфікаційнаРобота**

Кількість сторінок: **68** Кількість слів: **11120** Кількість символів: **88215** Розмір файлу: **3.28 MB** ID файлу: **1011379842**

5.42%
Схожість

Найбільша схожість: **1.88%** з Інтернет-джерелом (https://lpnu.ua/sites/default/files/2021/pages/12564/kriptosistema.pdf.

3.35% Джерела з Інтернету	75	Сторінка 70
2.81% Джерела з Бібліотеки	78	Сторінка 70

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	1

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 10%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 104745 Название: Криптографічна система нелінійного шифрування з можливістю стеганографічного застосування Добавлено в БД: 2022-06-08 Авторы: Анікін Володимир Андрійович Руководители: Муляр І.В. Консультанты: Опоненты:	Символы	Лексемы	Символы	Лексемы
	71233	1075	795 (1%)	16 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы