

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій
(повне найменування факультету)

Кафедра інформаційної безпеки та наноелектроніки
(повне найменування кафедри)

Пояснювальна записка

до дипломної роботи

бакалавра

(ступінь вищої освіти)

на тему Програмна реалізація алгоритму шифрування SM4

(назва теми)

Software implementation of the SM4 encryption algorithm

Виконав: студент 4 курсу, групи БК-710

Спеціальності 125 Кібербезпека

(код і найменування спеціальності)

Освітня програма

Системи технічного захисту інформації,
автоматизація її обробки

КАЦЮБА В.В.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЗИНА Г.Л.

(ПРИЗВИЩЕ та ініціали)

Рецензент МАЛИЙ О.Ю.

(ПРИЗВИЩЕ та ініціали)

2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет інформаційної безпеки та електронних комунікацій

Кафедра інформаційної безпеки та наноелектроніки

Ступінь вищої освіти бакалавр

Спеціальність 125 Кібербезпека

(код і найменування)

Освітня програма Системи технічного захисту інформації, автоматизація її обробки

(назва освітньої програми)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІБтаН

Андрій КОРОТУН

«_____» _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТА

КАЦЮБИ Віктора Вікторовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту Програмна реалізація алгоритму шифрування SM4

Software implementation of the SM4 encryption algorithm

керівник проєкту доцент, к. ф.-м. н., КОЗІНА Галина Леонідівна

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 03 » травня 20 24 року №190

2. Строк подання студентом проєкту 03.06.2024

3. Вихідні дані до проєкту Розроблена програма SM4 Demo (сторінка в GitHub:

github.com/vityak2k22/SM4_App)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Відомості про алгоритм шифрування SM4: використання і поширеність, загальні характеристики, опис алгоритму. 2. Програмна реалізація: мета програмної реалізації, опис програми, тестування роботи програми. 3. Приклад виконання програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації PowerPoint (14 слайдів)

6. Консультанти розділів проєкту

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	КОЗИНА Г.Л., доцент кафедри ІБтаН	15.04.2024	06.06.2024
Нормоконтроль	КОРОЛЬКОВ Р.Ю., доцент кафедри ІБтаН	06.06.2024	06.06.2024

7. Дата видачі завдання « 15 » квітня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту	Строк виконання етапів проєкту	Примітка
1	Постановка завдання проєкту	1 тиждень	Виконано
2	Дослідження алгоритму SM4	1 тиждень	Виконано
3	Реалізація алгоритму SM4 шляхом написання функціональної частини програми.	2, 3 тиждень	Виконано
4	Створення засобами CLI віконного інтерфейсу програми; поєднання функціональної частини програми та віконного інтерфейсу	4 тиждень	Виконано
5	Додання можливості зберігання введених даних для прискорення ініціалізації вхідних параметрів при кожному наступному запуску програми	4 тиждень	Виконано
6	Обробка помилок та виключень; введення додаткових елементів інтерфейсу та їх властивостей; реалізація демонстраційного режиму; реліз програми	4 – 6 тиждень	Виконано
7	Оформлення пояснювальної записки та підготовка презентації	7, 8 тиждень	Виконано
8	Нормоконтроль та рецензування	8 тиждень	
9	Захист дипломної роботи	9 тиждень	

Студент

Віктор КАЦЮБА

(підпис)

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту

Галина КОЗИНА

(підпис)

(Ім'я ПРИЗВИЩЕ)

АНОТАЦІЯ

Пояснювальна записка до дипломного проєкту: 85 с., 25 рис., 2 дод., 13 джерел.

ІНСТРУМЕНТИ ЗАГАЛЬНОМОВНОГО ІНТЕРФЕЙСУ СЛІ, КИТАЙСКИЙ СТАНДАРТ ШИФРУВАННЯ SM4, МОВА ПРОГРАМУВАННЯ C++, ПРОГРАМНА РЕАЛІЗАЦІЯ, СИМЕТРИЧНА КРИПТОГРАФІЯ.

Об'єкт роботи – програма, написана з метою дослідження роботи китайського алгоритму шифрування SM4.

Мета роботи – дослідити загальні характеристики, етапи та принцип роботи алгоритму шифрування SM4, написати програмну реалізацію цього алгоритму для демонстрації його роботи.

Методи дослідження – моделювання етапів алгоритму з використанням написаного демонстраційного програмного забезпечення; оброблення та аналіз отриманих результатів.

Досліджено китайський симетричний алгоритм шифрування SM4, який станом на сьогодні є одним із найбільш поширених криптографічних алгоритмів і використовується в бездротових мережевих протоколах, мережевих та крипто-бібліотеках.

У ході дослідження алгоритму SM4 розглянуто математичний апарат алгоритму, написано програму, яка виконує та демонструє процеси шифрування та дешифрування, проаналізовано результати тестування.

ABSTRACT

Explanatory note to the diploma project: 85 p, 25 figures, 2 appendixes, 13 references.

CHINESE ENCRYPTION STANDARD SM4, CLI COMMON LANGUAGE INTERFACE TOOLS, C++ PROGRAMMING LANGUAGE, SOFTWARE IMPLEMENTATION, SYMMETRIC CRYPTOGRAPHY.

The object of work is a program written to study the operation of the Chinese encryption algorithm SM4.

Purpose is to study the general characteristics, stages and principle of the SM4 encryption algorithm, to write a software implementation of this algorithm to demonstrate its operation.

Research methods are modeling the stages of the algorithm using the written demonstration software; processing and analysis of the results.

The Chinese symmetric encryption algorithm SM4, which is currently one of the most common cryptographic algorithms and is used in wireless network protocols, network and crypto libraries, is investigated.

In the course of the study of the SM4 algorithm, the mathematical apparatus of the algorithm is considered, a program is written that performs and demonstrates the encryption and decryption processes, and the test results are analyzed.

ЗМІСТ

	С.
Перелік термінів.....	7
Вступ.....	8
1 Дослідження алгоритму SM4	10
1.1 Загальні характеристики	10
1.2 Етапи алгоритму.....	10
1.3 Можливості вдосконалення алгоритму	13
2 Програмна реалізація алгоритму шифрування SM4	15
2.1 Використані засоби та етапи реалізації.....	15
2.2 Опис програми SM4 Demo	16
2.2.1 Опис графічного інтерфейсу	16
2.2.2 Опис роботи програми	18
2.2.3 Режим демонстрації.....	19
2.3 Тестування програми SM4 Demo	21
3 Приклад роботи програми SM4 Demo.....	26
3.1 Керівництво користувача	26
3.3 Приклад шифрування текстового повідомлення.....	26
3.3 Розшифрування отриманої шифрограми.....	29
3.4 Робота SM4 Demo з файлами різних типів.....	31
Висновки.....	36
Перелік джерел посилання.....	37
Додаток А Код програми SM4 Demo	39
Додаток Б Вивід файлу Demonstration.txt під час тестування програми SM4 Demo	53

ПЕРЕЛІК ТЕРМІНІВ

Блоковий шифр – алгоритм симетричної криптографії, який працює з блоками фіксованих розмірів. При шифруванні чи дешифруванні повідомлення ділиться на фіксовані блоки, з якими може працювати блоковий шифр.

Виключне АБО (англ. XOR) – побітова операція, яка повертає нуль при однакових вхідних бітах, та одиницю при різних вхідних бітах. Цю операцію представляють як додавання за модулем 2.

Загальнономовна інфраструктура (англ. Common Language Infrastructure, CLI) – засіб програмування, розроблений компанією Microsoft, який представляє собою сукупність типів, функцій, класів та інших інструментів.

Лінійна заміна – одна з операцій в симетричній криптографії, яка полягає в лінійному перетворенні байту/бітового вектора в інший байт/бітовий вектор, при цьому кожному вхідному байту/бітовому вектору відповідає одне значення перетвореного байта/бітового вектора. Зазвичай лінійна заміна представляє собою лінійну функцію.

Мережа Фейстеля – одна із архітектур блокових шифрів, основним принципом якої є однаковість алгоритмів шифрування та розшифрування за винятком порядку використання раундових ключів.

Нелінійна заміна – одна з операцій в симетричній криптографії, яка полягає в нелінійному перетворенні байту/бітового вектора в інший байт/бітовий вектор, при цьому кожному вхідному байту/бітовому вектору відповідає одне значення перетвореного байта/бітового вектора. Зазвичай нелінійна заміна представляє собою просту заміну з таблиць констант.

Циклічний зсув – побітова операція, під час якої біти переміщуються по розрядах вліво чи вправо, а біти, які вийшли за межі бітового вектору, переміщуються у протилежний бік.

ВСТУП

Алгоритм шифрування SM4 є одним із найбільш поширених криптографічних алгоритмів. Він був розроблений у 2003 році для китайського внутрішнього ринку та опублікований у 2006 році [1]. У 2010-х роках SM4 набуває чинності китайського промислового стандарту GM/T 0002-2012 [2] та національного стандарту GB/T 32907-2016 [3][1]. У 2021 р. він набуває чинності міжнародного стандарту та доданий в ISO/IEC 18033-3:2010 у доповненні Amd 1:2021 [4].

Станом на сьогодні цей алгоритм широко використовується в системах бездротового зв'язку, системах шифрування в інтегрованих системах, та захищених мережових та крипто-бібліотеках. Одними із прикладів використання SM4 є мережевий протокол бездротового зв'язку WAPI, який позбавлений вразливостей протоколів WPA2 та WPA3 [5], є частиною ARMv8.4-A [6], бібліотеки Libgcrypt з версії 1.9.0 [7] та OpenSSL з версії 1.1.1 [8].

Широке поширення також зумовлене великою кількістю апаратних та програмних реалізацій, зокрема SM4 реалізовано різними мовами програмування з використанням особливостей кожної з них. Частіше за все програмні реалізації алгоритму представляють собою частини бібліотек, які можна підключити до своїх проєктів. Також існують проєкти, як прикладні програми та онлайн-сервіси, які виконують процес шифрування для текстових повідомлень та файлів різних типів.

Метою дипломної роботи є дослідження алгоритму SM4, його характеристик, етапів та можливостей його вдосконалення. Одним із методів дослідження є написання програми з наглядною демонстрацією етапів алгоритму. У подальшому написану програму можна використовувати у навчальному процесі в якості учбового посібника.

У першому розділі роботи розглядається алгоритм SM4, його структура, характеристики (розмір блоку та ключа, кількість раундів, використовувані

елементарні операції) та етапи. Також в цьому розділі розглядаються можливості вдосконалення алгоритму.

Другий розділ присвячено програмній реалізації алгоритму SM4, яка була виконана у ході дослідження алгоритму – програмі SM4 Demo. В розділі описано графічний інтерфейс програми, його роботу та ключові риси. Увагу приділено режиму демонстрації як головній рисі цієї програми. Також у розділі наведено результати з тестування програми.

У третьому розділі описано керівництво користувача з користування написаною програмою, наведено приклади шифрування/розшифрування та проаналізовано отримані результати.

Далі зроблено висновки та вказано перелік джерел посилань.

В Додатку А наведено код написаної програми мовою програмування C++ з використанням засобів загальномовної інфраструктури CLI.

1 ДОСЛІДЖЕННЯ АЛГОРИТМУ SM4

1.1 Загальні характеристики

SM4 є блоковим алгоритмом, розмір блоку та розмір ключа якого мають довжину 128 біт. Шифрування та розшифрування складається з 32 раундів нелінійних підстановок. Алгоритм побудований на мережі Фейстеля, яка полягає в однаковості процесів шифрування та розшифрування за винятком порядку використання раундових ключів. Для розшифрування порядок є зворотним відносно порядку використання раундових ключів для шифрування. [9]

Елементарні математичні операції, які використовуються в цьому алгоритмі:

$\oplus$ – побітове виключне АБО (операція XOR) для додавання двох 32-бітних векторів;

$\lll i$ – циклічний зсув 32-бітного слова на i позицій вліво. [9]

1.2 Етапи алгоритму

Основою шифрування є перетворення T . Функція T генерує 32-бітне слово з вхідного 32-бітного слова та представляє собою суперпозицію лінійної та нелінійної заміни. Функція має наступний вигляд:

$$T(\cdot) = L(\tau(\cdot)), \quad (1.1)$$

де $\tau(\cdot)$ – функція нелінійної заміни;

$L(\cdot)$ – функція лінійної заміни [9].

Нелінійна заміна $\tau(\cdot)$ має наступний вигляд:

$$T(a_0, a_1, a_2, a_3) = (S(a_0), S(a_1), S(a_2), S(a_3)), \quad (1.2)$$

де a_i – 8-бітне слово;

$S(a_i)$ – підстановка з таблиці констант (S-блок, S-box). Таблиця підстановки зображена на рис. 1.1.

Лінійна заміна $L(\cdot)$ для шифрування має наступний вигляд:

$$L(B) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24), \quad (1.3)$$

де B – 32-бітне слово [9].

Алгоритм шифрування та дешифрування є наступним.

1. На вхід подається 128-бітний ключ MK , який розкладається на чотири 32-бітних слова: (MK_0, MK_1, MK_2, MK_3) . Текст повідомлення або шифртексту розбивається на блоки по 128 біт (якщо недостатньо байт для блоку, то такий блок заповнюється байтами нулів), кожний з яких розкладається на чотири 32-бітних слова: (X_0, X_1, X_2, X_3) .

2. Генерація раундових ключів (розширення ключа, *Key Expansion*). На вхід подається MK , $FK = (FK_0, FK_1, FK_2, FK_3) = (a3b1bac6, 56aa3350, 677d9197, b27022dc)$ та $CK: ck_{i,j} = (4i + j) \times 7 \bmod 256, i = 0, 1, \dots, 31, j = 0, 1, 2, 3$ (рис. 1.2).

Генерація ключів відбувається в два етапи:

$$K_j = MK_j \oplus FK_j, j = 0, 1, 2, 3, \quad (1.4)$$

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i), i = 0, 1, \dots, 31 \quad (1.5)$$

де T' представляє собою ту саму функцію T за виключенням функції лінійної підстановки:

$$L'(B) = B \oplus (B \lll 13) \oplus (B \lll 23) \quad (1.6)$$

3. Шифрування або дешифрування блоку. Шифрування блоку відбувається в 32 раунди за допомогою раундових ключів. Шифрування має наступний вигляд:

$$X_i = X_{i+4} = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i), i = 0, 1, \dots, 31 \quad (1.7)$$

Зашифрований блок має вигляд: $(Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32})$.

Дешифрування має той самий алгоритм за винятком того, що раундові ключі застосовуються для (1.7) в зворотному порядку: $(rk_{31}, rk_{30}, \dots, rk_0)$.

Схема виконання одного раунду зображена на рис. 1.3.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	d6	90	e9	fe	cc	e1	3d	b7	16	b6	14	c2	28	fb	2c	05
1	2b	67	9a	76	2a	be	04	c3	aa	44	13	26	49	86	06	99
2	9c	42	50	f4	91	ef	98	7a	33	54	0b	43	ed	cf	ac	62
3	e4	b3	1c	a9	c9	08	e8	95	80	df	94	fa	75	8f	3f	a6
4	47	07	a7	fc	f3	73	17	ba	83	59	3c	19	e6	85	4f	a8
5	68	6b	81	b2	71	64	da	8b	f8	eb	0f	4b	70	56	9d	35
6	1e	24	0e	5e	63	58	d1	a2	25	22	7c	3b	01	21	78	87
7	d4	00	46	57	9f	d3	27	52	4c	36	02	e7	a0	c4	c8	9e
8	ea	bf	8a	d2	40	c7	38	b5	a3	f7	f2	ce	f9	61	15	a1
9	e0	ae	5d	a4	9b	34	1a	55	ad	93	32	30	f5	8c	b1	e3
a	1d	f6	e2	2e	82	66	ca	60	c0	29	23	ab	0d	53	4e	6f
b	d5	db	37	45	de	fd	8e	2f	03	ff	6a	72	6d	6c	5b	51
c	8d	1b	af	92	bb	dd	bc	7f	11	d9	5c	41	1f	10	5a	d8
d	0a	c1	31	88	a5	cd	7b	bd	2d	74	d0	12	b8	e5	b4	b0
e	89	69	97	4a	0c	96	77	7e	65	b9	f1	09	c5	6e	c6	84
f	18	f0	7d	ec	3a	dc	4d	20	79	ee	5f	3e	d7	cb	39	48

Рисунок 1.1 – Таблиця нелінійної підстановки S-бокс для SM4 [9]

```

00070e15, 1c232a31, 383f464d, 545b6269,
70777e85, 8c939aa1, a8afb6bd, c4cbd2d9,
e0e7eef5, fc030a11, 181f262d, 343b4249,
50575e65, 6c737a81, 888f969d, a4abb2b9,
c0c7ced5, dce3eaf1, f8ff060d, 141b2229,
30373e45, 4c535a61, 686f767d, 848b9299,
a0a7aeb5, bcc3cad1, d8dfe6ed, f4fb0209,
10171e25, 2c333a41, 484f565d, 646b7279

```

Рисунок 1.2 – Таблиця констант CK_i [9]

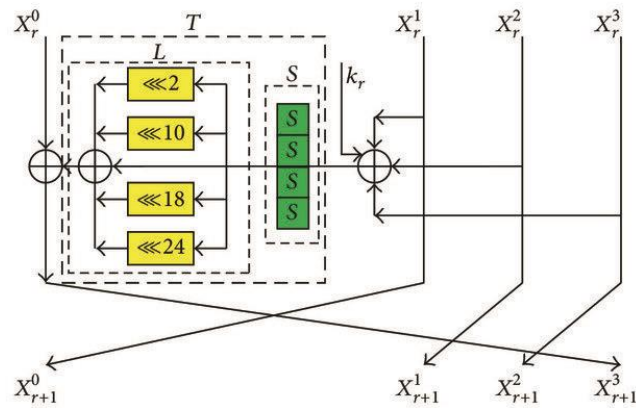


Рисунок 1.3 – Схема одного раунду шифрування SM4 [9]

1.3 Можливості вдосконалення алгоритму

Елементарними операціями, які використовуються у функціях лінійної заміни (1.3) та (1.6), є виключне АБО (XOR) та циклічний зсув вліво. Циклічний зсув можна представити як множення матриці зсуву на бітовий вектор.

Матриця зсуву уліво у бітовому представленні представляє собою одиничну матрицю, зсунуту на певну кількість біт вправо. Наприклад, в результаті зсуву бітового вектора 10011101 на 3 позиції уліво отримаємо:

$$10011101 \lll 3 = 11101100$$

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Оскільки матриці зсуву є множниками, то запис функцій (1.3) та (1.6)

зводиться до запису многочленів, у яких 32-бітні слова B є спільними множниками. В результаті отримуємо суму матриць зсуву для кожної функції лінійної заміни. Суми матриць зсуву є таблицями констант, використання яких значно збільшує швидкодію алгоритму.

Вище описаний підхід використовується в алгоритмі потокового шифрування СТРУМОК та є наведеним у державному стандарті [10]. Використовується цей підхід на етапі нелінійної підстановки (за алгоритмом операція виконується через циклічний зсув вправо і множення на байтовий вектор) та обчислення множення в розширеному полі $GF(2^{64})$. Перевагою цього методу збільшення швидкодії є значний вплив на швидкість СТРУМКА. У науковій статті [1] вказано, що з використанням цього підходу оптимізації його швидкість становить понад 10 Гбіт/с, і за цим показником він випереджає практично всі найбільш поширені алгоритми, зокрема і алгоритм SNOW2.0.

Недоліком цього методу оптимізації є необхідність використання більшого об'єму пам'яті. Для SM4 додатковий обсяг усіх таблиць констант складає 8 кБ, проте для переважної більшості систем цей об'єм не є значним чи критичним.

2 ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ШИФРУВАННЯ SM4

2.1 Використані засоби та етапи реалізації

У ході дослідження алгоритму SM4 було виконано його програмну реалізацію. Отримана програма має назву SM4 Demo та може виконувати шифрування та розшифрування файлів різних типів. Також передбачено демонстраційний режим, у якому програма виводить опис етапів виконання алгоритму. Це дає можливість використовувати програму як інструмент для шифрування файлів, так і для навчальних цілей.

Програма була написана мовою програмування C++ з використанням засобів загальномовної інфраструктури CLI та виконуваного середовища CLR, а також з використанням типів даних Windows API. CLI використовується для створення програм для операційної системи Windows, в тому числі з графічним інтерфейсом. Графічний інтерфейс для програми був реалізований за допомогою інструменту Windows Forms.

Написання програми виконувалось в наступні етапи.

1. Реалізація алгоритму SM4 шляхом написання функціональної частини.
2. Створення засобами CLR віконного інтерфейсу програми. Для зручності проектування інтерфейсу було використано інструмент Windows Forms.
3. Поєднання функціональної частини програми та віконного інтерфейсу.
4. Додання можливості зберігання введених даних для прискорення ініціалізації вхідних параметрів при кожному наступному запуску програми.
5. Обробка помилок та виключень; введення демонстраційного режиму та додаткових елементів інтерфейсу та їх властивостей.

Код програми знаходиться в GitHub-репозиторії [12] та Додатку А.

2.2 Опис програми SM4 Demo

2.2.1 Опис графічного інтерфейсу

Інтерфейс програми SM4 Demo (рис. 2.1) складається з наступних елементів:

- текстове поле `initialtext (path)`. В це поле записується шлях до файлу. В SM4 Demo початковий текст (`initialtext`) – це файл на початку алгоритму, який, залежно від обраного режиму шифрування або дешифрування, може бути відкритим текстом або шифротекстом. Шлях можна ввести вручну або скористатися вибором файлу за допомогою кнопки `Open` (рис. 2.2);

- текстове поле `finaltext (path)`. Кінцевим текстом (`finaltext`) є повідомлення після роботи алгоритму. В залежності від вибраного режиму може бути як шифротекстом, так і відкритим текстом. Як і `initialtext`, шлях також можна ввести вручну або скористатися вибором файлу за допомогою кнопки `Open`;

- перемикач режиму `Mode`. Завдяки ньому можна вибрати режим роботи алгоритму;

- текстове поле `Key`. В це поле вводиться ключ у символічному вигляді, який використовується під час роботи алгоритму. За замовчуванням він є замаскованим, проте для перевірки правильності можна скористатись кнопкою `Look`, яка показує зміст ключа. Саме текстове поле має обмеження вводу символів на 16 байт;

- прапорець «Виконати у режимі демонстрації» («`with Demonstration mode`»). Якщо цей прапорець встановлено, програма згенерує файл, який показує проміжні кроки алгоритму SM4, такі як робота T-функції, генерації раундових ключів та процес шифрування першого блоку початкового тексту;

- прапорець «Згенерувати файл `settings.ini`» («`generate settings.ini file checkbox`»). Якщо прапорець встановлено, програма генерує файл `settings.ini`, в якому зберігаються останні вхідні дані після виконання алгоритму для наступного швидкого запуску програми;

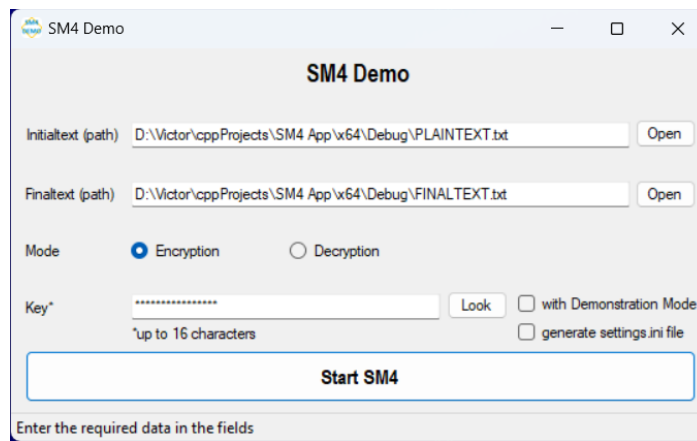


Рисунок 2.1 – Інтерфейс програми SM4 Demo [12]

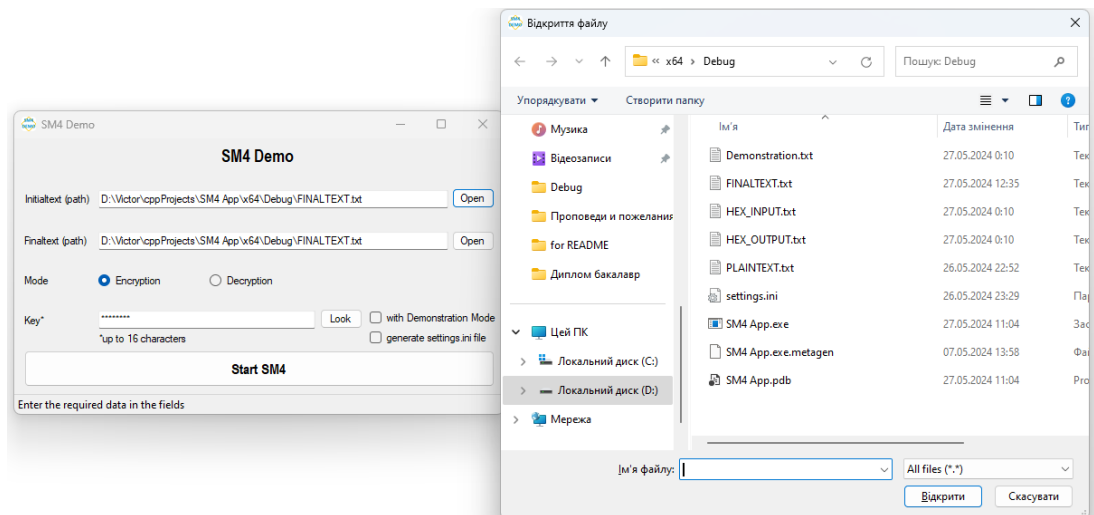


Рисунок 2.2 – Ініціалізація initialtext path за допомогою опції Open

- кнопка Start SM4. При натисканні відбувається обробка вхідних даних та робота алгоритму шифрування;

- статус-рядок. Він виводить стан роботи програми. Під час обробки введених даних, перед початком та після роботи алгоритму у ньому можуть бути наступні повідомлення:

“Enter the required data in the fields” – введіть необхідні дані в поля;

“Invalid paths to files” – неправильно введені шляхи для файлів;

"SM4 process is working. . ." – вхідні дані оброблені, та алгоритм SM4

працює;

"Done!" – алгоритм успішно завершив роботу. [12]

Однією із особливостей SM4 Demo є можливість записати однаковий шлях у текстові поля `initalex` і `finaltext`. При натисканні кнопки Start SM4 весь вміст поля `initalex` повністю вивантажується в оперативну пам'ять, оброблюється алгоритмом і зберігається в полі `finaltext`. Таким чином, якщо вказано однаковий шлях до текстових полів `initalex` і `finaltext`, результат буде збережено у файлі `initalex`. Це працює як для шифрування, так і для розшифрування. [12]

2.2.2 Опис роботи програми

Після введення усіх ключових даних і натиснення на кнопку Start SM4 робота програми починається з перевірки правильності введених полів. У разі їхньої коректності виконується наступні дії [12].

1. Перетворення ключа в масив чотирибайтових беззнакових цілих чисел. Воно представляє собою побайтове введення кожного байту рядка в масив. Якщо введений рядок має довжину, меншу за граничну допустиму у 16 байт, то після введення всіх значимих байтів у кінець масиву записуються нулі.

2. Запуск алгоритму. Алгоритм роботи програми поділяється на наступні складові:

- ініціалізація початкового тексту (`initalex`). У кінець початкового тексту записуються нуль-символи (`'\0'`) для поділу повідомлення по 128-бітним блокам;
- представлення початкового тексту у шістнадцятковому вигляді та його збереження у файлі `"HEX_INPUT.txt"`;
- генерація раундових ключів (Key Expansion) з ключа;
- процес шифрування. Блоки із рядку початкового тексту, який переведений у шістнадцятковий формат, присвоюються до блоку, який проходить процес шифрування. Отриманий результат зберігається у рядку

кінцевого тексту у шістнадцятковому вигляді;

- отриманий рядок зберігається у файлі “HEX_OUTPUT.txt” та подається у символічному вигляді у файлі кінцевого тексту (finaltext).

3. Відображення успішного виконання алгоритму у статус-рядку повідомленням “Done!”.

4. Зберігання вхідних даних у файлі ініціалізації settings.ini (якщо встановлено прапорець «Згенерувати файл settings.ini»). Формат файлу ini представляє собою список параметрів «ключ=значення». Оформлення settings.ini зображено на рис. 3.1. [12]

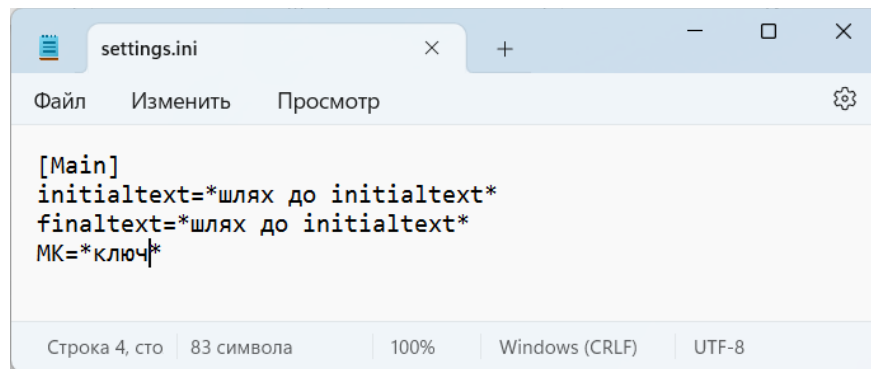


Рисунок 2.3 – Формат файлу settings.ini [12]

2.2.3 Режим демонстрації

Під час демонстраційного режиму програма генерує записи у файлі Demonstration.txt. Кожен запис складається з наступних частин.

1. Заголовок. Містить дату і час запуску алгоритму, шлях до початкового і кінцевого тексту, а також вибраний режим шифрування.

2. Секція Key Expansion (рис. 2.4). Містить процес обчислення раундових ключів.

3. Секція роботи з блоком SM4 Block Operation (рис. 2.5). Містить процес

обчислення першого блоку початкового тексту.

Зазначені секції складаються з Т-функції та основних обчислень. Т-функція містить операції S-box (нелінійна підстановка байтів) та L (лінійна підстановка).

Формат файлу Demonstration.txt показано на рис. 2.4 та рис. 2.5.

```

=====SM4 Demo (27.05.2024, 11:04:15)=====
Initialtext is D:\Victor\cppProjects\SM4 App\x64\Debug\PLAINTEXT.txt
Finaltext is D:\Victor\cppProjects\SM4 App\x64\Debug\FINALTEXT.txt
Encryption mode is Encryption
=====Key Expansion=====
MK = 69666865 776F6967 75686476 626E7375;

K[0] = MK[0] xor FK[0] = 69666865 xor A3B1BAC6 = CAD7D2A3
K[1] = MK[1] xor FK[1] = 776F6967 xor 56AA3350 = 21C55A37
K[2] = MK[2] xor FK[2] = 75686476 xor 677D9197 = 1215F5E1
K[3] = MK[3] xor FK[3] = 626E7375 xor B27022DC = D01E51A9

T function:
1) S-box:
'6A' = '7C'
'F0' = '18'
'C9' = 'D9'
'E3' = '4A'
S-box result: Tau = 4AD9187C
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 4AD9187C xor (4AD9187C <<< 13) xor (4AD9187C <<< 23) = 57F3FDAB

rk[0] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[0] ) = CAD7D2A3 xor T'(21C55A37 xor 1215F5E1 xor D01E51A9 xor 00070E15) =
= CAD7D2A3 xor T'(E3C9F06A) = 9D242F08

T function:
1) S-box:
'71' = '00'
'A1' = 'F6'
'0C' = '28'
'43' = 'FC'
S-box result: Tau = FC28F600
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = FC28F600 xor (FC28F600 <<< 13) xor (FC28F600 <<< 23) = E296FDFE

rk[1] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[1] ) = 21C55A37 xor T'(1215F5E1 xor D01E51A9 xor 9D242F08 xor 1C32A31) =
= 21C55A37 xor T'(430CA171) = C353A7C9

```

Рисунок 2.4 – Секція Key Expansion у файлі Demonstration.txt

```

=====SM4 Block Operation=====

T function:
1) S-box:
'E6' = '77'
'91' = 'AE'
'23' = 'F4'
'17' = 'C3'
S-box result: Tau = C3F4AE77
2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
= C3F4AE77 xor (C3F4AE77 <<< 2) xor (C3F4AE77 <<< 10) xor (C3F4AE77 <<< 18) xor (C3F4AE77 <<< 24) = D08333DB

X[4] = X[0] xor T(X[1] xor X[2] xor X[3] xor rk[31]) = 2B8E03EE xor T(14D3F385 xor 285CE39D xor CDBAB487 xor E6163579) =
= 2B8E03EE xor T(172391E6) = FB0D3035

T function:
1) S-box:
'8D' = '61'
'AC' = '0D'
'8F' = 'A1'
'42' = 'A7'
S-box result: Tau = A7A10D61
2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
= A7A10D61 xor (A7A10D61 <<< 2) xor (A7A10D61 <<< 10) xor (A7A10D61 <<< 18) xor (A7A10D61 <<< 24) = E93181F0

```

Рисунок 2.5 – Секція SM4 Block Operation у файлі Demonstration.txt

Важливо зазначити, що у режимі демонстрації програма не стирає старі нотатки у файлі `Demonstration.txt`. Вона додає нові нотатки в кінець файлу. Це зроблено для того, щоб попередні результати можна було порівняти з поточними.

Отже, можна побачити, що програма SM4 Demo має зручний зрозумілий інтерфейс та можливості для вивчення етапів алгоритму на реальних прикладах файлів, що робить програму корисною для навчальних цілей.

2.3 Тестування програми SM4 Demo

У роботі [9] наведено приклад шифрування байтів. Вхідні дані з прикладу наступні:

- відкритий текст: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10;
- ключ: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10.

Значення раундових ключів та шифрованого блоку повинні бути наступними:

rk[0] = f12186f9	X[0] = 27fad345
rk[1] = 41662b61	X[1] = a18b4cb2
rk[2] = 5a6ab19a	X[2] = 11c1e22a
rk[3] = 7ba92077	X[3] = cc13e2ee
rk[4] = 367360f4	X[4] = f87c5bd5
rk[5] = 776a0c61	X[5] = 33220757
rk[6] = b6bb89b3	X[6] = 77f4c297
rk[7] = 24763151	X[7] = 7a96f2eb
rk[8] = a520307c	X[8] = 27dac07f
rk[9] = b7584dbd	X[9] = 42dd0f19
rk[10] = c30753ed	X[10] = b8a5da02
rk[11] = 7ee55b57	X[11] = 907127fa

rk[12] = 6988608c	X[12] = 8b952b83
rk[13] = 30d895b7	X[13] = d42b7c59
rk[14] = 44ba14af	X[14] = 2ffc5831
rk[15] = 104495a1	X[15] = f69e6888
rk[16] = d120b428	X[16] = af2432c4
rk[17] = 73b55fa3	X[17] = ed1ec85e
rk[18] = cc874966	X[18] = 55a3ba22
rk[19] = 92244439	X[19] = 124b18aa
rk[20] = e89e641f	X[20] = 6ae7725f
rk[21] = 98ca015a	X[21] = f4cba1f9
rk[22] = c7159060	X[22] = 1dcdfa10
rk[23] = 99e1fd2e	X[23] = 2ff60603
rk[24] = b79bd80c	X[24] = eff24fdc
rk[25] = 1d2115b0	X[25] = 6fe46b75
rk[26] = 0e228aeb	X[26] = 893450ad
rk[27] = f1780c81	X[27] = 7b938f4c
rk[28] = 428d3654	X[28] = 536e4246
rk[29] = 62293496	X[29] = 86b3e94f
rk[30] = 01cf72e5	X[30] = d206965e
rk[31] = 9124a012	X[31] = 681edf34

Значення шифротексту: 68 1e df 34 d2 06 96 5e 86 b3 e9 4f 53 6e 42 46. [9]

Створимо файл `hex example.txt` та запишемо в нього байти відкритого тексту, вказаного в прикладі. Для вводу байтів у шістнадцятковому вигляді використано шістнадцятковий редактор HxD (рис. 2.6). Результат зберігатиметься у файлі `hex example out.txt`.

Оскільки в програмі не передбачено ввід ключа в шістнадцятковому вигляді, то було змінено рядки коду, який відповідає за присвоєння значення масиву ключа:

```
DWORD MK[4] = {0x01234567, 0x89abcdef, 0xfedcba98, 0x76543210};
//convert_string_to_dword_for_MK(mkTextBox, MK); -> Цей рядок відповідає за конвертацію
введенного ключа та зберігання в масиві. Його закоментовано
```

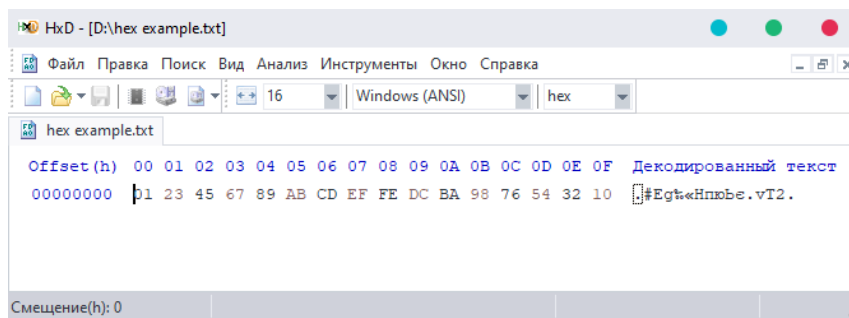


Рисунок 2.6 – Введення байтів відкритого тексту у файл hex example.txt за допомогою редактора HxD

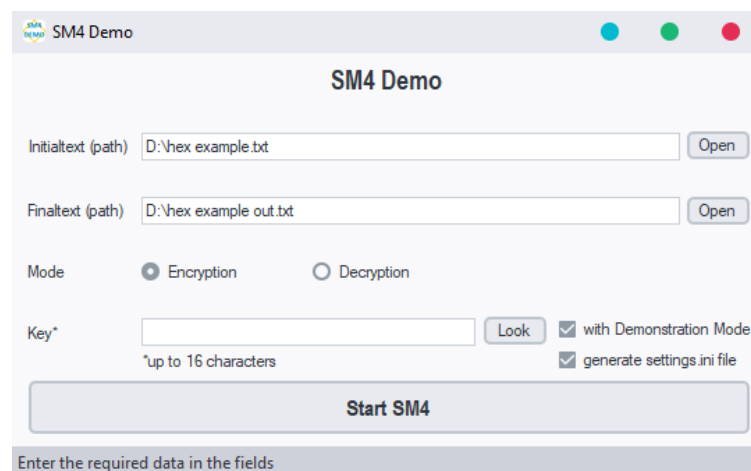


Рисунок 2.7 – SM4 Demo із введеними вхідними даними

У програмі SM4 Demo введено наступні поля (рис. 2.7):

- **initialtext (path):** “D:\hex example.txt”;
- **finaltext (path):** “D:\hex example out.txt”;
- **Mode:** Encryption;
- **Key:** не введено;
- **with Demonstration Mode:** прапорець встановлено;
- **generate settings.ini file:** прапорець встановлено.

Частина змісту секцій Key Expansion та SM4 Block Operation зображена на рис. 2.8 та 2.9. Повний зміст файлу Demonstration.txt наведено в Додатку Б.

Зміст файлу hex example out.txt у шістнадцятковому вигляді зображено на

рис. 2.10.

```

Demonstration.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
=====
SM4 Demo (05.06.2024, 21:20:27)
=====Key Expansion=====
MK = 01234567 89ABCDEF FEDCBA98 76543210;

K[0] = MK[0] xor FK[0] = 01234567 xor A381BAC6 = A292FFA1
K[1] = MK[1] xor FK[1] = 89ABCDEF xor 56AA3350 = DF01FEBF
K[2] = MK[2] xor FK[2] = FEDCBA98 xor 677D9197 = 99A12B0F
K[3] = MK[3] xor FK[3] = 76543210 xor B27022DC = C42410CC

T function:
1) S-box:
'69' = '22'
'CB' = '41'
'83' = 'D2'
'82' = '8A'
S-box result: Tau = 8AD24122
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 8AD24122 xor (8AD24122 <<< 13) xor (8AD24122 <<< 23) = 53B37958

rk[0] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[0] ) = A292FFA1 xor T'(DF01FEBF xor 99A12B0F xor C42410CC xor 00070E15) =
= A292FFA1 xor T'(8283CB69) = F12186F9

T function:
1) S-box:
'0B' = 'C2'
'97' = '55'
'87' = 'B5'
'B0' = 'D5'
S-box result: Tau = D5B555C2
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = D5B555C2 xor (D5B555C2 <<< 13) xor (D5B555C2 <<< 23) = 9E67D5DE

rk[1] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[1] ) = DF01FEBF xor T'(99A12B0F xor C42410CC xor F12186F9 xor 1C232A31) =
= DF01FEBF xor T'(B087970B) = 41662B61

T function:
1) S-box:
'19' = '44'
'FB' = '3E'
'5C' = '70'
'4C' = 'E6'
S-box result: Tau = E6703E44
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = E6703E44 xor (E6703E44 <<< 13) xor (E6703E44 <<< 23) = C3CB9A95

rk[2] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[2] ) = 99A12B0F xor T'(C42410CC xor F12186F9 xor 41662B61 xor 383F464D) =
= 99A12B0F xor T'(4C5CFB19) = 5A6AB19A

```

Рисунок 2.8 – Секція Key Expansion файлу Demonstration.txt

```

=====SM4 Block Operation=====
T function:
1) S-box:
'9E' = 'B1'
'C3' = '92'
'02' = 'E9'
'F0' = '18'
S-box result: Tau = 18E992B1
2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
= 18E992B1 xor (18E992B1 <<< 2) xor (18E992B1 <<< 10) xor (18E992B1 <<< 18) xor (18E992B1 <<< 24) = 26D99622

X[4] = X[0] xor T(X[1] xor X[2] xor X[3] xor rk[0]) = 01234567 xor T(89ABCDEF xor FEDCBA98 xor 76543210 xor F12186F9) =
= 01234567 xor T(F002C39E) = 27FAD345

T function:
1) S-box:
'AC' = '0D'
'70' = 'D4'
'14' = '2A'
'EE' = 'C6'
S-box result: Tau = C62AD40D
2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
= C62AD40D xor (C62AD40D <<< 2) xor (C62AD40D <<< 10) xor (C62AD40D <<< 18) xor (C62AD40D <<< 24) = 2820815D

X[5] = X[1] xor T(X[2] xor X[3] xor X[4] xor rk[1]) = 89ABCDEF xor T(FEDCBA98 xor 76543210 xor 27FAD345 xor 41662B61) =
= 89ABCDEF xor T(E4170AC) = A18B4CB2

T function:
1) S-box:
'7D' = 'C4'
'1C' = '49'
'4F' = 'A8'
'AA' = '23'
S-box result: Tau = 23A849C4
2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
= 23A849C4 xor (23A849C4 <<< 2) xor (23A849C4 <<< 10) xor (23A849C4 <<< 18) xor (23A849C4 <<< 24) = EF1D58B2

X[6] = X[2] xor T(X[3] xor X[4] xor X[5] xor rk[2]) = FEDCBA98 xor T(76543210 xor 27FAD345 xor A18B4CB2 xor 5A6AB19A) =
= FEDCBA98 xor T(AA4F1C7D) = 11C1E22A

```

Рисунок 2.9 – Секція SM4 Block Operation файлу Demonstration.txt

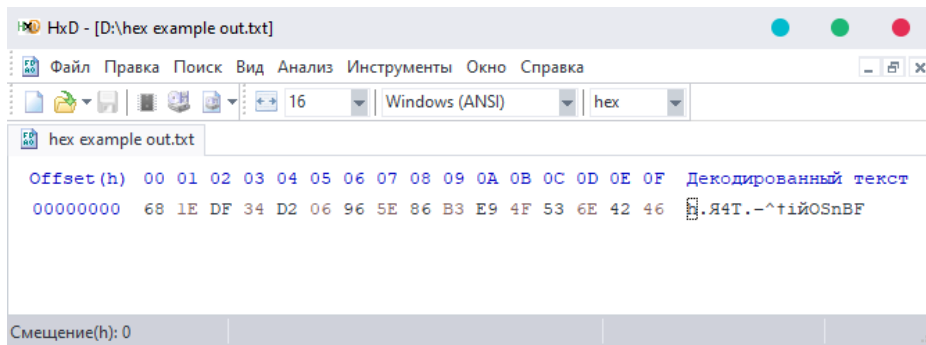


Рисунок 2.10 – Зміст файлу hex example out.txt

За результатами шифрування можна побачити, що на виході ми отримали той самий шифротекст, який вказаний у прикладі [9], а отже, програма SM4 Демо працює коректно.

З ПРИКЛАД РОБОТИ ПРОГРАМИ SM4 DEMO

3.1 Керівництво користувача

Для виконання шифрування/дешифрування програмою SM4 Demo необхідно дотримуватись наступних кроків.

1. Підготувати файл, який треба зашифрувати/розшифрувати.
 2. Відкрити SM4 Demo на комп'ютері.
 3. У поле `initialtext (path)` ввести шлях до файлу, який треба зашифрувати/розшифрувати або вибрати файл, натиснувши кнопку `Open`.
 4. У поле `finaltext (path)` ввести шлях, назву і розширення файлу, куди треба зберегти результат шифрування/розшифрування або вибрати файл, натиснувши кнопку `Open`.
- За необхідності результат можна зберегти у вхідному файлі. Для цього необхідно вказати той самий шлях, що і в полі `initialtext (path)`.
5. У поле `Key` ввести ключ у символьному вигляді. Якщо необхідно перевірити правильність введеного ключа, натиснути на кнопку `Look`.
 6. У разі потреби дослідження етапів алгоритму натиснути на прапорець «with Demonstration Mode».
 7. У разі потреби збереження введених вхідних даних для наступного швидкого запуску алгоритму натиснути на прапорець «generate settings.ini file».
 8. Натиснути на кнопку `Start SM4` та дочекатись повідомлення у статус-рядку «Done! Result saved in <шлях, вказаний в `finaltext`>».

3.3 Приклад шифрування текстового повідомлення

Для тестування шифрування використано файл `PLAINTEXT.txt`, в якому збережено текст великого обсягу (більш ніж 3,5 млн символів). Частина його

змісту можна побачити на рис. 3.1.

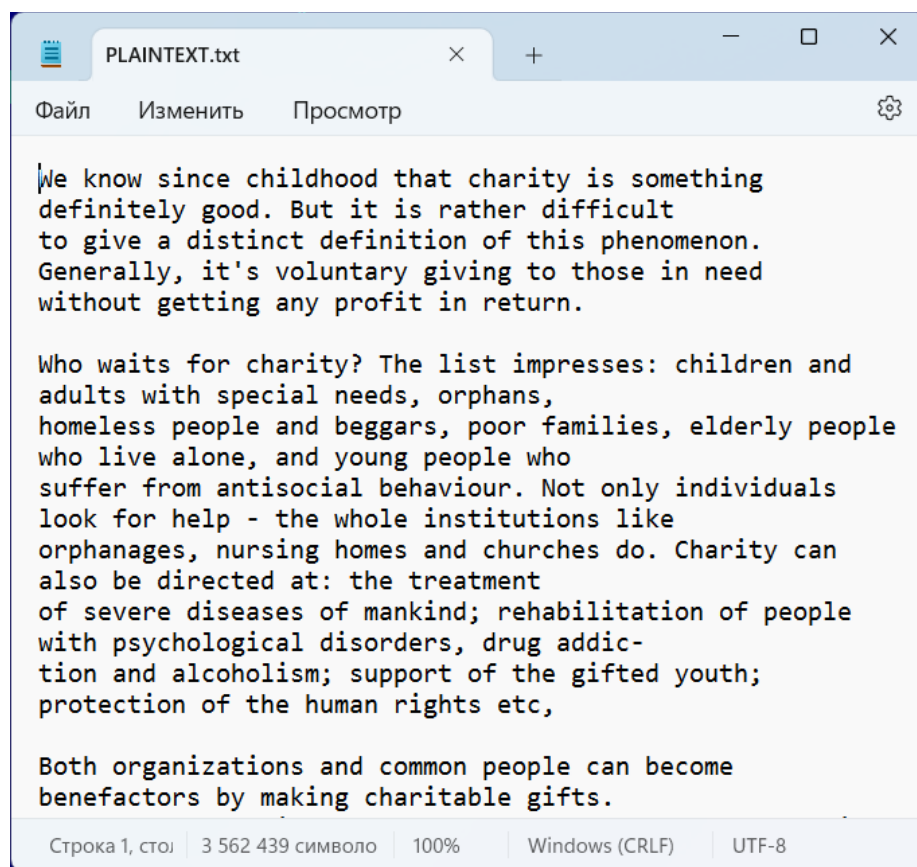


Рисунок 3.1 – Зміст файлу PLAINTEXT.txt

Результат шифрування зберігатиметься у файлі, який буде створений програмою. Результуючий файл матиме назву FINALTEXT.txt.

Отже, поля в SM4 Demo мають наступні значення (рис. 3.2):

- **initialtext (path):** “D:\Victor\cppProjects\SM4 App\x64\Debug\PLAINTEXT.txt”;
- **finaltext (path):** “D:\Victor\cppProjects\SM4 App\x64\Debug\FINALTEXT.txt”;
- **Mode:** Encryption;
- **Key:** test_one;
- **with Demonstration Mode:** прапорець встановлено;
- **generate settings.ini file:** прапорець встановлено.

SM4 Demo із введеними вхідними даними має вигляд, зображений на рис. 3.2.

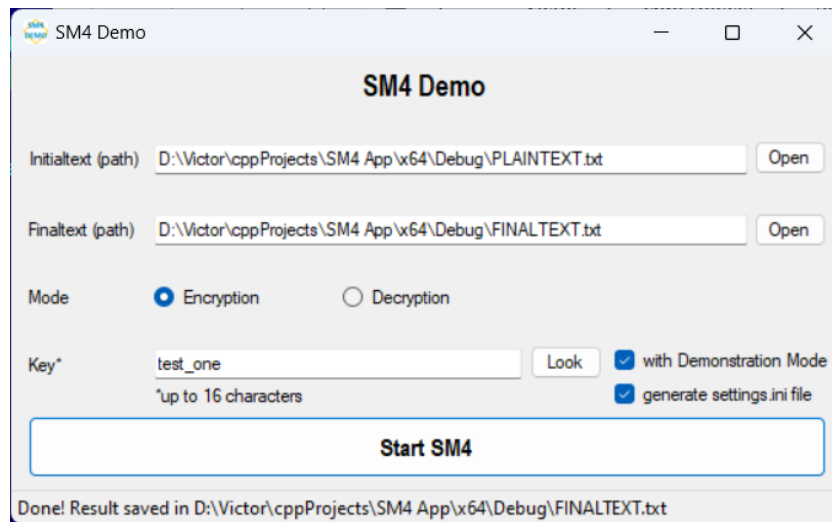


Рисунок 3.2 – SM4 Demo із вхідними параметрами

В ході роботи програми було згенеровано файли FINALTEXT.txt, settings.ini (рис. 3.3) та Demonstration.txt (рис. 3.4). FINALTEXT.txt представляє собою шифrogramу, в якому всі байти були оброблені алгоритмом. Файл settings.ini містить збережені вхідні дані, які будуть вставлені в поля після наступного запуску програми.

У файлі Demonstration.txt описано етапи генерації раундових ключів та першого блоку повідомлення.

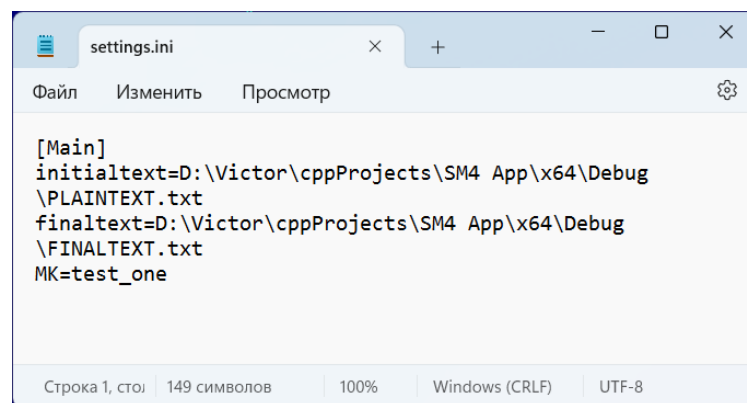
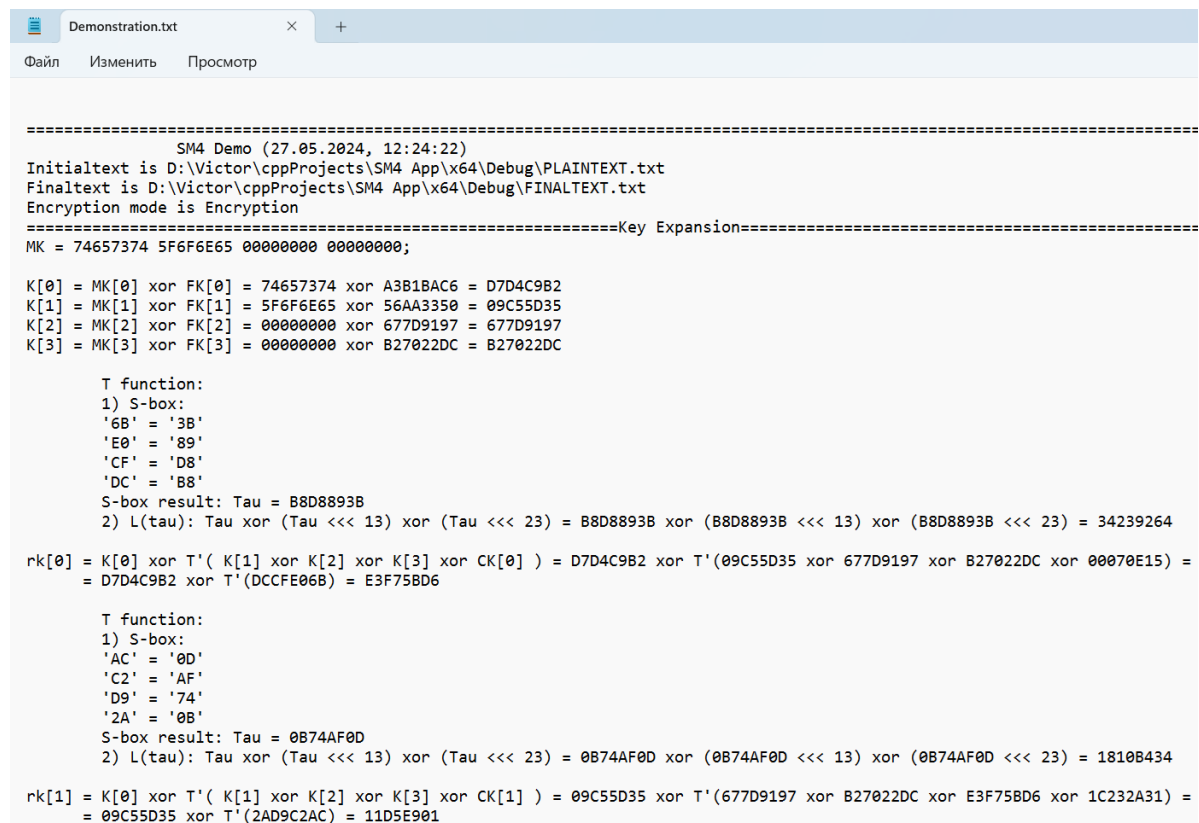


Рисунок 3.3 – Зміст файлу settings.ini



```

=====
SM4 Demo (27.05.2024, 12:24:22)
Initialtext is D:\Victor\cppProjects\SM4 App\x64\Debug\PLAINTEXT.txt
Finaltext is D:\Victor\cppProjects\SM4 App\x64\Debug\FINALTEXT.txt
Encryption mode is Encryption
=====Key Expansion=====
MK = 74657374 5F6F6E65 00000000 00000000;

K[0] = MK[0] xor FK[0] = 74657374 xor A3B1BAC6 = D7D4C9B2
K[1] = MK[1] xor FK[1] = 5F6F6E65 xor 56AA3350 = 09C55D35
K[2] = MK[2] xor FK[2] = 00000000 xor 677D9197 = 677D9197
K[3] = MK[3] xor FK[3] = 00000000 xor B27022DC = B27022DC

T function:
1) S-box:
'6B' = '3B'
'E0' = '89'
'CF' = 'D8'
'DC' = 'B8'
S-box result: Tau = B8D8893B
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = B8D8893B xor (B8D8893B <<< 13) xor (B8D8893B <<< 23) = 34239264

rk[0] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[0] ) = D7D4C9B2 xor T'(09C55D35 xor 677D9197 xor B27022DC xor 00070E15) =
= D7D4C9B2 xor T'(DCCFE06B) = E3F75BD6

T function:
1) S-box:
'AC' = '0D'
'C2' = 'AF'
'D9' = '74'
'2A' = '0B'
S-box result: Tau = 0B74AF0D
2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 0B74AF0D xor (0B74AF0D <<< 13) xor (0B74AF0D <<< 23) = 1810B434

rk[1] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[1] ) = 09C55D35 xor T'(677D9197 xor B27022DC xor E3F75BD6 xor 1C232A31) =
= 09C55D35 xor T'(2AD9C2AC) = 11D5E901

```

Рисунок 3.4 – Зміст файлу Demonstration.txt

У наступному підрозділі описано процес розшифрування шифрограми у відкритий текст.

3.3 Розшифрування отриманої шифрограми

Під час розшифрування повідомлення результат буде збережений у тому самому файлі, в якому на той момент знаходиться шифрограма.

Вхідними даними для розшифрування повідомлення є:

- **initialtext (path):** “D:\Victor\cppProjects\SM4 App\x64\Debug\FINALTEXT.txt”;
- **finaltext (path):** “D:\Victor\cppProjects\SM4 App\x64\Debug\FINALTEXT.txt”;

- **Mode:** Decryption;
- **Key:** test_one;
- **with Demonstration Mode:** прапорець встановлено;
- **generate settings.ini file:** прапорець встановлено.

SM4 Demo із введеними вхідними даними має вигляд, зображений на рис. 3.5.

На рис. 3.6 можна побачити, що при зворотному розшифруванні отримано початкове текстове повідомлення без помилок та пошкоджень.

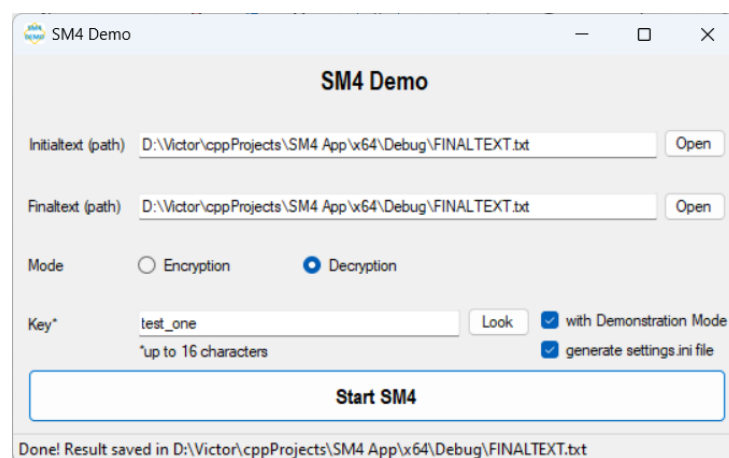


Рисунок 3.5 – SM4 Demo із введеними вхідними даними

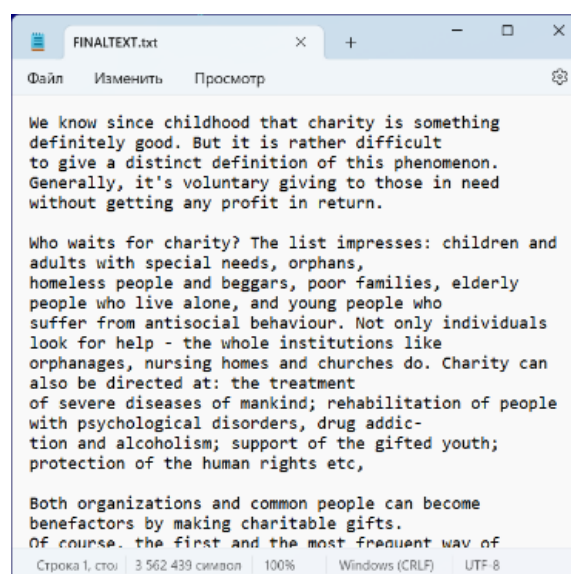


Рисунок 3.6 – Зміст файлу FINALTEXT.txt

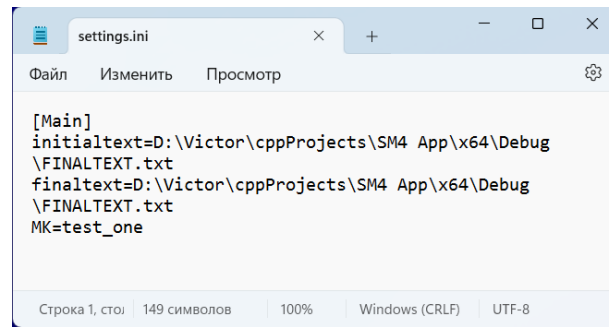


Рисунок 3.7 – Зміст файлу settings.ini

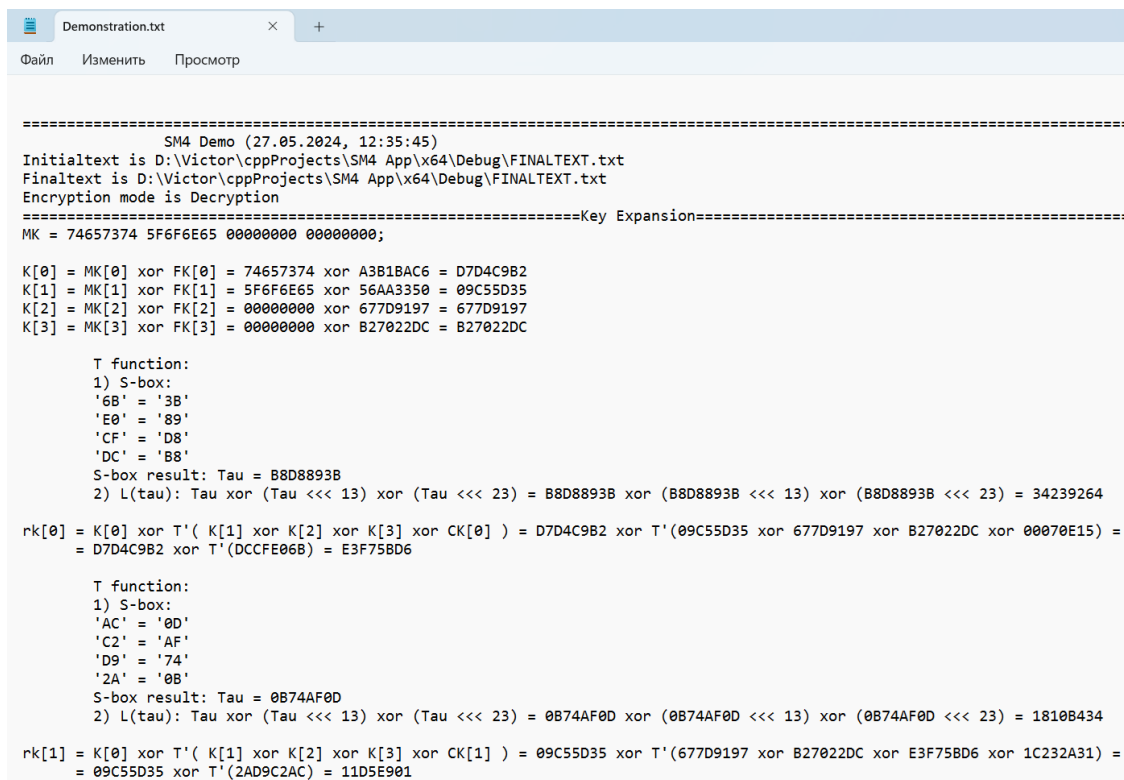


Рисунок 3.8 – Зміст файлу Demonstration.txt

3.4 Робота SM4 Демо з файлами різних типів

Оскільки в SM4 Демо реалізовано бінарне введення даних з файлів, починаючи з нульового байту, то таким чином можна зашифрувати та

розшифрувати файл незалежно від його типу.

Розглянемо процеси шифрування та дешифрування docx-файлу. В SM4 Demo введено наступні вхідні дані (рис. 3.9):

- **initialtext (path):** “D:\doc Plain.docx”;
- **finaltext (path):** “D:\doc Cipher.txt”;
- **Mode:** Encryption;
- **Key:** test_02;
- **with Demonstration Mode:** прапорець не встановлено;
- **generate settings.ini file:** прапорець не встановлено.

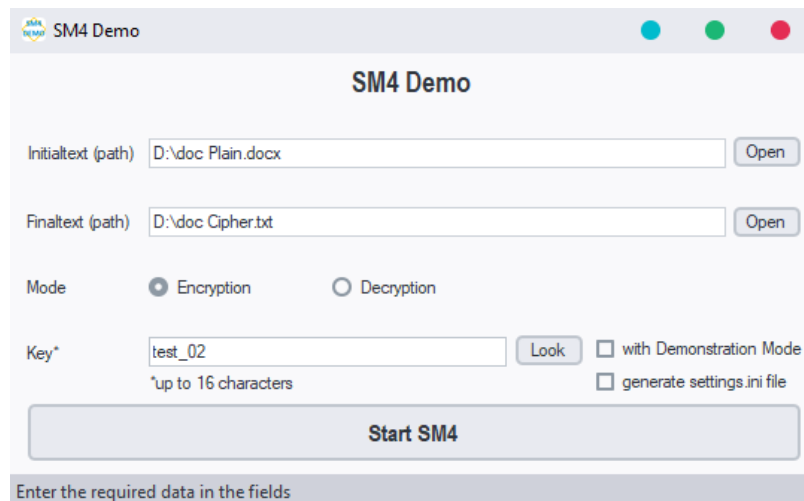


Рисунок 3.9 – SM4 Demo із введеними вхідними даними

Програма може зберігати результат шифрування/дешифрування у файлах, які можуть мати інші типи. В даному випадку результат шифрування зберігається у файлі типу txt.

Зміст файлу doc Plain.docx зображено на рис. 3.10. Файл формату .docx – це документ Microsoft Word у форматі Open XML [13], тому сам документ представляє файл з розміченим текстом. При шифруванні усі байти, починаючи з нульового, оброблюються та спотворюються алгоритмом, тому, якби результат зберігався в форматі docx, програма Microsoft Word не змогла б його прочитати.

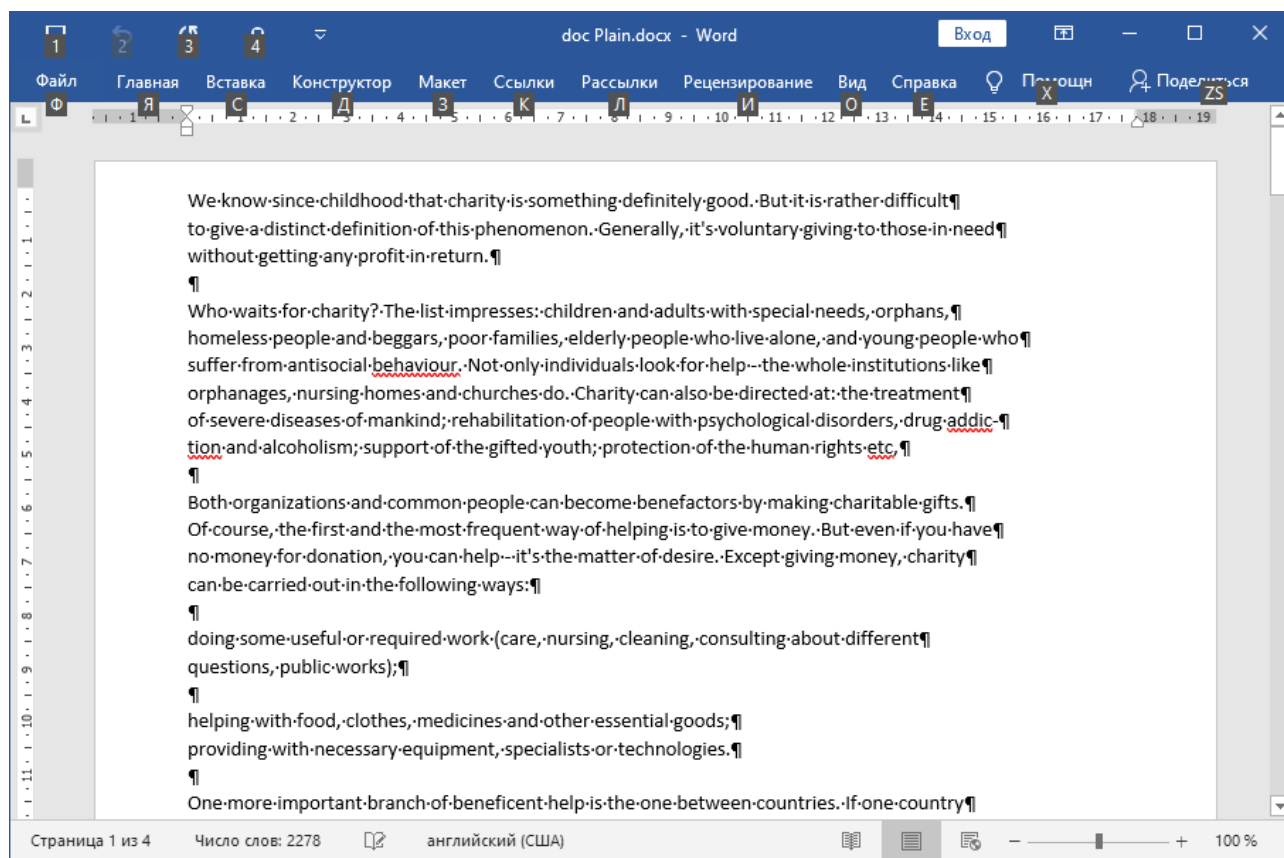


Рисунок 3.10 – Зміст файлу doc Plain.docx

Виконаємо розшифрування файлу doc Cipher.txt у файл з docx-розширенням. Введено наступні вхідні дані (рис. 3.11):

- **initialtext (path):** "D:\doc Cipher.txt";
- **finaltext (path):** "D:\doc Plain Cipher.docx";
- **Mode:** Decryption;
- **Key:** test_02;
- **with Demonstration Mode:** прапорець не встановлено;
- **generate settings.ini file:** прапорець не встановлено.

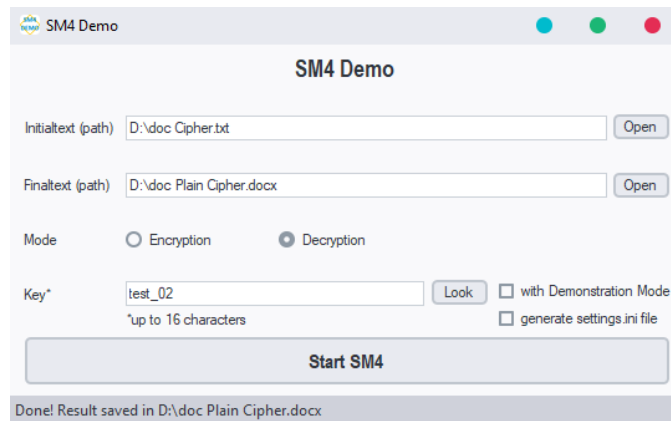


Рисунок 3.11 – SM4 Demo із введеними вхідними даними

При розшифруванні файлу отримано документ із відкритим текстом (рис. 3.12). Проте необхідно звернути увагу на те, що Microsoft Word не відразу відкриває файл. Для початку він запропонує його відновити. При виборі опції відновлення Word відкриває зміст файлу у *новому* документі, який через опцію «Зберегти як» необхідно зберегти. Але, не дивлячи на це, процес дешифрування виконано успішно.

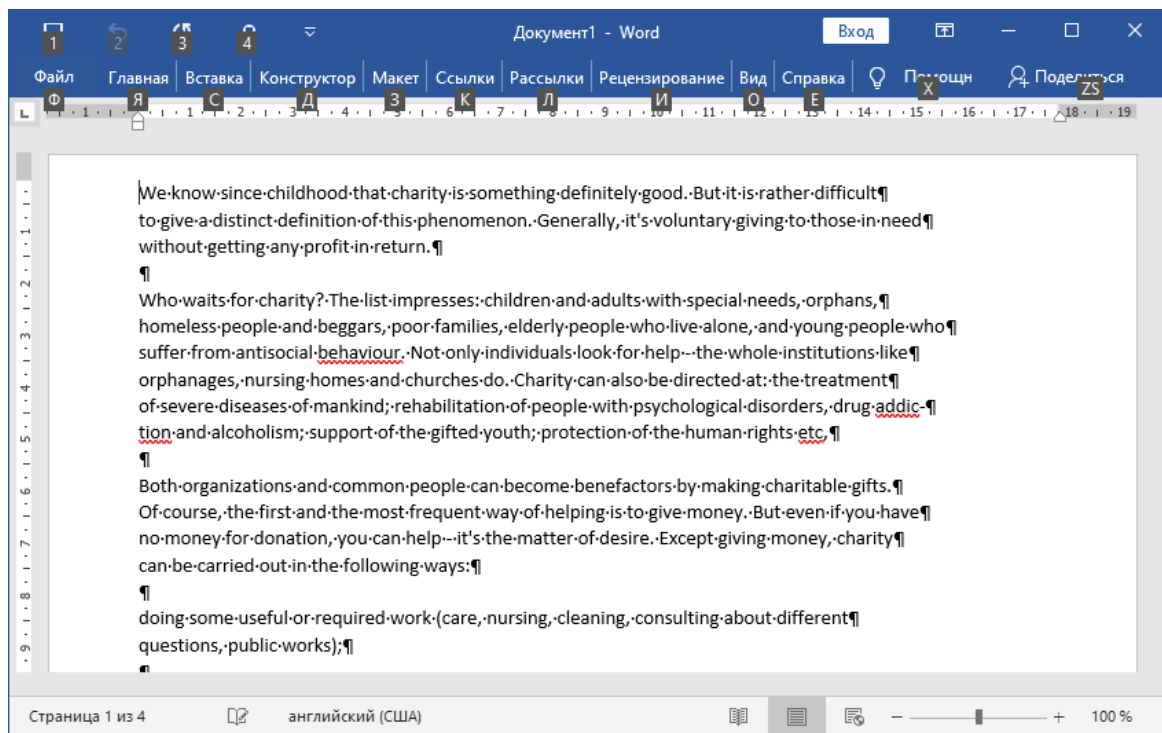


Рисунок 3.12 – Зміст файлу після розшифрування

Програма SM4 Demo може обробляти не тільки текстові файли. Вона може обробляти аудіо-, відеофайли, бінарні файли та ін. Також можна результат зберігати у файлі, якому можна присвоїти будь-яке інше розширення. Головне, щоб при розшифруванні файл мав той самий формат, який мав файл на початку шифрування.

ВИСНОВКИ

У ході дослідження алгоритму шифрування SM4 було виконано наступне.

1. Досліджено структуру та основні операції алгоритму SM4.

2. Написано програму для шифрування файлів алгоритмом SM4, яка має назву SM4 Демо. Ця програма може шифрувати та розшифровувати файли різних форматів. Головною особливістю програми є наявність демонстраційного режиму, завдяки якому її можна використовувати у навчальних цілях. Протестовано програму на перевіреному прикладі та отримано позитивний результат.

3. Виконано програму на текстових повідомленнях та файлах різних типів. За отриманими результатами можна побачити, що програма коректно обробляє вхідні дані, проводить процес шифрування та дешифрування та зберігає отриманий результат, що дозволяє користуватись програмою не тільки у навчальних цілях, а як інструмент для ручного шифрування файлів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tse R. H., Kit W. W. History. The SM4 Block Cipher Algorithm And Its Modes Of Operations. 2017. С. 4. URL: <https://datatracker.ietf.org/doc/draft-crypto-sm4/> (дата звернення: 04.06.2024).

2. GM/T 0002-2012. SM4 block cipher algorithm. Чинний від 2012-03-21. Вид. офіц. State Cryptography Administration of the People's Republic of China, 2012. 10 с. URL: <https://www.chinesestandard.net/Related.aspx/GMT0002-2012> (дата звернення: 04.06.2024).

3. GB/T 32907-2016. Information security technology - SM4 block cipher algorithm. Чинний від 2016-08-29. Вид. офіц. General Administration of Quality Supervision, Inspection and Quarantine; Standardization Administration of PRC, 2016. 12 с. URL: <https://www.chinesestandard.net/Related.aspx/GBT32907-2016> (дата звернення: 04.06.2024).

4. ISO/IEC 18033-3:2010/Amd 1:2021. Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers Amendment 1: SM4. Чинний від 2021-06-22. Вид. офіц. 2021. URL: <https://cdn.standards.iteh.ai/samples/81564/bcb7f53a40934e2f9078998884eb27ad/ISO-IEC-18033-3-2010-Amd-1-2021.pdf> (дата звернення: 04.06.2024).

5. Awati R. WLAN Authentication and Privacy Infrastructure (WAPI). URL: <https://www.techtarget.com/searchsecurity/definition/WAPI-WLAN-Authentication-and-Privacy-Infrastructure> (дата звернення: 28.05.2024).

6. Introducing 2017's extensions to the Arm Architecture. Arm Community. URL: <https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/introducing-2017s-extensions-to-the-arm-architecture> (дата звернення: 28.05.2024).

7. Libgcrypt 1.9.0 relased. lists.gnu.org. URL: <https://lists.gnu.org/archive/html/info-gnu/2021-01/msg00013.html> (дата звернення: 28.05.2024).

8. Libraries - /docs/man1.1.1/man3/index.html. openssl.org. URL:

<https://www.openssl.org/docs/man1.1.1/man3/index.html> (дата звернення: 28.05.2024).

9. Diffie W., Ledin G. SMS4 Encryption Algorithm for Wireless Networks. 2008. 6 p.

10. ДСТУ 8845:2019 Криптографічний захист інформації. Алгоритм симетричного потокового шифрування. Київ: Держстандарт України, 2019. 54 с.

11. Математична структура потокового шифру СТРУМОК/ О. Кузнецов та ін. *Радіотехніка*. 2018. № 193. С. 11.

12. GitHub - vityak2k22/SM4_App: Program is realization SM4 Encryption Algorithm. GitHub. URL: https://github.com/vityak2k22/SM4_App (дата звернення: 28.05.2024).

13. Про формати файлів - Підтримка від Microsoft. Microsoft Support. URL: <https://support.microsoft.com/uk-ua/office/про-формати-файлів-56dc3b55-7681-402e-a727-c59fa0884b30#:~:text=docx%20%20це%20документ%20Microsoft%20Word%20у%20форматі%20Open%20XML> . (дата звернення: 04.06.2024).

ДОДАТОК А

Код програми SM4 Demo

1. Файл Main.h

```
#pragma once
#include "SM4.h"

namespace SM4App{
    /// <summary>
    /// Summary for Main
    /// </summary>
    public ref class Main : public System::Windows::Forms::Form
    {
    public:
        Main(void)
        {
            InitializeComponent();
        }

    protected:
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        ~Main()
        {
            if (components)
            {
                delete components;
            }
        }
        private: System::Windows::Forms::Button^ startButton;
        private: System::Windows::Forms::Label^ Intro;
        private: System::Windows::Forms::TextBox^ initialtextTextBox;
        private: System::Windows::Forms::Label^ initextLabel;
        private: System::Windows::Forms::RadioButton^ cryptonRadioButton;
        private: System::Windows::Forms::Label^ finaltextLabel;
        private: System::Windows::Forms::TextBox^ finaltextTextBox;
        private: System::Windows::Forms::Label^ modeLabel;
        private: System::Windows::Forms::RadioButton^ decryptionRadioButton;
        private: System::Windows::Forms::Label^ mkLabel;
        private: System::Windows::Forms::TextBox^ mkTextBox;
        private: System::Windows::Forms::Button^ initialtextOpenFileButton;
        private: System::Windows::Forms::Button^ finaltextOpenFileButton;
        private: System::Windows::Forms::StatusStrip^ statusStrip;
        private: System::Windows::Forms::ToolStripStatusLabel^ statusLabel;

    protected:
        LPCWSTR setting_ini_path = L"./settings.ini";
        private: System::Windows::Forms::CheckBox^ demoCheckBox;
        private: System::Windows::Forms::CheckBox^ settingsiniCheckBox;

        private: System::Windows::Forms::Label^ keyexplainLabel;
```

protected:

private:

```

    /// <summary>
    /// ReqMainred designer variable.
    /// </summary>
    System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// ReqMainred method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        System::Windows::Forms::Button^ lookmkButton;
        System::ComponentModel::ComponentResourceManager^ resources = (gcnew
System::ComponentModel::ComponentResourceManager(Main::typeid));
        this->startButton = (gcnew System::Windows::Forms::Button());
        this->Intro = (gcnew System::Windows::Forms::Label());
        this->initialtextTextBox = (gcnew System::Windows::Forms::TextBox());
        this->initextLabel = (gcnew System::Windows::Forms::Label());
        this->cryptioRadioButton = (gcnew System::Windows::Forms::RadioButton());
        this->finaltextLabel = (gcnew System::Windows::Forms::Label());
        this->finaltextTextBox = (gcnew System::Windows::Forms::TextBox());
        this->modeLabel = (gcnew System::Windows::Forms::Label());
        this->decryptionRadioButton = (gcnew System::Windows::Forms::RadioButton());
        this->mkLabel = (gcnew System::Windows::Forms::Label());
        this->mkTextBox = (gcnew System::Windows::Forms::TextBox());
        this->initialtextOpenFileButton = (gcnew System::Windows::Forms::Button());
        this->finaltextOpenFileButton = (gcnew System::Windows::Forms::Button());
        this->statusStrip = (gcnew System::Windows::Forms::StatusStrip());
        this->statusLabel = (gcnew System::Windows::Forms::ToolStripStatusLabel());
        this->demoCheckBox = (gcnew System::Windows::Forms::CheckBox());
        this->settingsiniCheckBox = (gcnew System::Windows::Forms::CheckBox());
        this->keyexplainLabel = (gcnew System::Windows::Forms::Label());
        lookmkButton = (gcnew System::Windows::Forms::Button());
        this->statusStrip->SuspendLayout();
        this->SuspendLayout();
        //
        // lookmkButton
        //
        lookmkButton->Location = System::Drawing::Point(338, 188);
        lookmkButton->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);
        lookmkButton->Name = L"lookmkButton";
        lookmkButton->Size = System::Drawing::Size(47, 22);
        lookmkButton->TabIndex = 19;
        lookmkButton->Text = L"Look";
        lookmkButton->UseVisualStyleBackColor = true;
        lookmkButton->Click += gcnew System::EventHandler(this, &Main::lookmkButton_Click);
        //
        // startButton
        //
        this->startButton->AutoSize = true;
        this->startButton->Cursor = System::Windows::Forms::Cursors::Hand;
        this->startButton->Enabled = false;
        this->startButton->Font = (gcnew System::Drawing::Font(L"Arial Narrow", 12,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        this->startButton->Location = System::Drawing::Point(11, 234);
        this->startButton->Name = L"startButton";

```

```

this->startButton->Size = System::Drawing::Size(520, 40);
this->startButton->TabIndex = 0;
this->startButton->Text = L"Start SM4";
this->startButton->UseVisualStyleBackColor = true;
this->startButton->Click += gcnew System::EventHandler(this, &Main::startButton_Click);
//
// Intro
//
this->Intro->Anchor = System::Windows::Forms::AnchorStyles::Top;
this->Intro->Font = (gcnew System::Drawing::Font(L"Arial Narrow", 14.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->Intro->Location = System::Drawing::Point(0, 0);
this->Intro->Name = L"Intro";
this->Intro->Size = System::Drawing::Size(541, 36);
this->Intro->TabIndex = 2;
this->Intro->Text = L"SM4 Demo";
this->Intro->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// initialtextTextBox
//
this->initialtextTextBox->Location = System::Drawing::Point(93, 57);
this->initialtextTextBox->Name = L"initialtextTextBox";
this->initialtextTextBox->Size = System::Drawing::Size(387, 20);
this->initialtextTextBox->TabIndex = 3;
this->initialtextTextBox->TextChanged += gcnew System::EventHandler(this,
&Main::initialtextTextBox_TextChanged);
//
// initextLabel
//
this->initextLabel->AutoSize = true;
this->initextLabel->Location = System::Drawing::Point(9, 60);
this->initextLabel->Name = L"initextLabel";
this->initextLabel->Size = System::Drawing::Size(78, 13);
this->initextLabel->TabIndex = 4;
this->initextLabel->Text = L"Initialtext (path)";
//
// cryptonRadioButton
//
this->cryptonRadioButton->AutoSize = true;
this->cryptonRadioButton->Checked = true;
this->cryptonRadioButton->Location = System::Drawing::Point(93, 148);
this->cryptonRadioButton->Name = L"cryptonRadioButton";
this->cryptonRadioButton->Size = System::Drawing::Size(75, 17);
this->cryptonRadioButton->TabIndex = 6;
this->cryptonRadioButton->TabStop = true;
this->cryptonRadioButton->Text = L"Encryption";
this->cryptonRadioButton->UseVisualStyleBackColor = true;
//
// finaltextLabel
//
this->finaltextLabel->AutoSize = true;
this->finaltextLabel->Location = System::Drawing::Point(8, 106);
this->finaltextLabel->Name = L"finaltextLabel";
this->finaltextLabel->Size = System::Drawing::Size(76, 13);
this->finaltextLabel->TabIndex = 7;
this->finaltextLabel->Text = L"Finaltext (path)";
//
// finaltextTextBox
//
this->finaltextTextBox->Location = System::Drawing::Point(93, 103);
this->finaltextTextBox->Name = L"finaltextTextBox";

```

```

this->finaltextTextBox->Size = System::Drawing::Size(387, 20);
this->finaltextTextBox->TabIndex = 8;
this->finaltextTextBox->TextChanged += gcnew System::EventHandler(this,
&Main::finaltextTextBox_TextChanged);
//
// modeLabel
//
this->modeLabel->AutoSize = true;
this->modeLabel->Location = System::Drawing::Point(8, 150);
this->modeLabel->Name = L"modeLabel";
this->modeLabel->Size = System::Drawing::Size(34, 13);
this->modeLabel->TabIndex = 9;
this->modeLabel->Text = L"Mode";
//
// decryptionRadioButton
//
this->decryptionRadioButton->AutoSize = true;
this->decryptionRadioButton->Location = System::Drawing::Point(216, 148);
this->decryptionRadioButton->Name = L"decryptionRadioButton";
this->decryptionRadioButton->Size = System::Drawing::Size(76, 17);
this->decryptionRadioButton->TabIndex = 10;
this->decryptionRadioButton->Text = L"Decryption";
this->decryptionRadioButton->UseVisualStyleBackColor = true;
//
// mkLabel
//
this->mkLabel->AutoSize = true;
this->mkLabel->Location = System::Drawing::Point(8, 194);
this->mkLabel->Name = L"mkLabel";
this->mkLabel->Size = System::Drawing::Size(29, 13);
this->mkLabel->TabIndex = 11;
this->mkLabel->Text = L"Key*";
//
// mkTextBox
//
this->mkTextBox->Location = System::Drawing::Point(93, 190);
this->mkTextBox->Name = L"mkTextBox";
this->mkTextBox->>PasswordChar = '*';
this->mkTextBox->Size = System::Drawing::Size(240, 20);
this->mkTextBox->TabIndex = 12;
this->mkTextBox->TextChanged += gcnew System::EventHandler(this,
&Main::mkTextBox_TextChanged);
this->mkTextBox->KeyPress += gcnew System::Windows::Forms::KeyPressEventHandler(this,
&Main::mkTextBox_KeyPress);
//
// initialtextOpenFileButton
//
this->initialtextOpenFileButton->Location = System::Drawing::Point(484, 55);
this->initialtextOpenFileButton->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);
this->initialtextOpenFileButton->Name = L"initialtextOpenFileButton";
this->initialtextOpenFileButton->Size = System::Drawing::Size(47, 22);
this->initialtextOpenFileButton->TabIndex = 16;
this->initialtextOpenFileButton->Text = L"Open";
this->initialtextOpenFileButton->UseVisualStyleBackColor = true;
this->initialtextOpenFileButton->Click += gcnew System::EventHandler(this,
&Main::initialtextOpenFileButton_Click);
//
// finaltextOpenFileButton
//
this->finaltextOpenFileButton->Location = System::Drawing::Point(484, 102);
this->finaltextOpenFileButton->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);
this->finaltextOpenFileButton->Name = L"finaltextOpenFileButton";

```

```

this->finaltextOpenFileButton->Size = System::Drawing::Size(47, 22);
this->finaltextOpenFileButton->TabIndex = 17;
this->finaltextOpenFileButton->Text = L"Open";
this->finaltextOpenFileButton->UseVisualStyleBackColor = true;
this->finaltextOpenFileButton->Click += gcnew System::EventHandler(this,
&Main::finaltextOpenFileButton_Click);
//
// statusStrip
//
this->statusStrip->ImageScalingSize = System::Drawing::Size(20, 20);
this->statusStrip->Items->AddRange(gcnew cli::array<System::Windows::Forms::ToolStripItem^
>(1) { this->statusLabel });
this->statusStrip->Location = System::Drawing::Point(0, 282);
this->statusStrip->Name = L"statusStrip";
this->statusStrip->Padding = System::Windows::Forms::Padding(1, 0, 10, 0);
this->statusStrip->Size = System::Drawing::Size(541, 22);
this->statusStrip->SizingGrip = false;
this->statusStrip->TabIndex = 18;
this->statusStrip->Text = L"statusStrip1";
//
// statusLabel
//
this->statusLabel->Name = L"statusLabel";
this->statusLabel->Size = System::Drawing::Size(191, 17);
this->statusLabel->Text = L"Enter the required data in the fields";
//
// demoCheckBox
//
this->demoCheckBox->AutoSize = true;
this->demoCheckBox->Location = System::Drawing::Point(393, 190);
this->demoCheckBox->Name = L"demoCheckBox";
this->demoCheckBox->Size = System::Drawing::Size(146, 17);
this->demoCheckBox->TabIndex = 20;
this->demoCheckBox->Text = L"with Demonstration Mode";
this->demoCheckBox->UseVisualStyleBackColor = true;
//
// settingsiniCheckBox
//
this->settingsiniCheckBox->AutoSize = true;
this->settingsiniCheckBox->Location = System::Drawing::Point(393, 212);
this->settingsiniCheckBox->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);
this->settingsiniCheckBox->Name = L"settingsiniCheckBox";
this->settingsiniCheckBox->Size = System::Drawing::Size(136, 17);
this->settingsiniCheckBox->TabIndex = 21;
this->settingsiniCheckBox->Text = L"generate settings.ini file";
this->settingsiniCheckBox->UseVisualStyleBackColor = true;
//
// keyexplainLabel
//
this->keyexplainLabel->AutoSize = true;
this->keyexplainLabel->Location = System::Drawing::Point(91, 214);
this->keyexplainLabel->Margin = System::Windows::Forms::Padding(2, 0, 2, 0);
this->keyexplainLabel->Name = L"keyexplainLabel";
this->keyexplainLabel->Size = System::Drawing::Size(103, 13);
this->keyexplainLabel->TabIndex = 22;
this->keyexplainLabel->Text = L"*up to 16 characters";
//
// Main
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(541, 304);

```

```

        this->Controls->Add(this->keyexplainLabel);
        this->Controls->Add(this->settingsiniCheckBox);
        this->Controls->Add(this->demoCheckBox);
        this->Controls->Add(lookmkButton);
        this->Controls->Add(this->statusStrip);
        this->Controls->Add(this->finaltextOpenFileButton);
        this->Controls->Add(this->initialtextOpenFileButton);
        this->Controls->Add(this->mkTextBox);
        this->Controls->Add(this->mkLabel);
        this->Controls->Add(this->decryptionRadioButton);
        this->Controls->Add(this->modeLabel);
        this->Controls->Add(this->finaltextTextBox);
        this->Controls->Add(this->finaltextLabel);
        this->Controls->Add(this->cryptonRadioButton);
        this->Controls->Add(this->initextLabel);
        this->Controls->Add(this->initialtextTextBox);
        this->Controls->Add(this->Intro);
        this->Controls->Add(this->startButton);
        this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedDialog;
        this->Icon = (cli::safe_cast<System::Drawing::Icon^>)(resources->GetObject(L"$this.Icon"));
        this->Name = L"Main";
        this->Text = L"SM4 Demo";
        this->Load += gcnew System::EventHandler(this, &Main::Main_Load);
        this->statusStrip->ResumeLayout(false);
        this->statusStrip->PerformLayout();
        this->ResumeLayout(false);
        this->PerformLayout();
    }
#pragma endregion
//=====
// Дозволяє виконати процес, коли всі поля заповнені
private: void check_textboxes() {
    if (initialtextTextBox->Text != "" && finaltextTextBox->Text != "" && mkTextBox->Text != "")
        startButton->Enabled = true;
}
//=====
// Створення нових записів файлі у settings.ini
private: void save_new_settingsfile() {
    auto initialtext_path = Marshal::StringToHGlobalUni(initialtextTextBox->Text);
    auto finaltext_path = Marshal::StringToHGlobalUni(finaltextTextBox->Text);
    auto mk_path = Marshal::StringToHGlobalUni(mkTextBox->Text);

    WritePrivateProfileString(L"Main", L"initialtext", (LPCWSTR)initialtext_path.ToPointer(), setting_ini_path);
    WritePrivateProfileString(L"Main", L"finaltext", (LPCWSTR)finaltext_path.ToPointer(), setting_ini_path);
    WritePrivateProfileString(L"Main", L"MK", (LPCWSTR)mk_path.ToPointer(), setting_ini_path);

    Marshal::FreeHGlobal(initialtext_path);
    Marshal::FreeHGlobal(finaltext_path);
    Marshal::FreeHGlobal(mk_path);
}
//=====
private: System::Void Main_Load(System::Object^ sender, System::EventArgs^ e) {
    FileInfo^ save_file = gcnew FileInfo(gcnew String(setting_ini_path));
    if (save_file->Exists) {
        const WORD SIZE = 256;
        TCHAR buffer[SIZE] = {};

        GetPrivateProfileString(L"Main", L"initialtext", L"", buffer, SIZE, setting_ini_path);
        initialtextTextBox->Text = gcnew String(buffer);
        GetPrivateProfileString(L"Main", L"finaltext", L"", buffer, SIZE, setting_ini_path);
        finaltextTextBox->Text = gcnew String(buffer);
    }
}

```

```

        GetPrivateProfileString(L"Main", L"MK", L"", buffer, SIZE, setting_ini_path);
        mkTextBox->Text = gcnew String(buffer);
    }
}

private: System::Void initialtextOpenFileButton_Click(System::Object^ sender, System::EventArgs^ e) {
    open_file_dialog_for_textbox(initialtextTextBox);
}
private: System::Void finaltextOpenFileButton_Click(System::Object^ sender, System::EventArgs^ e) {
    open_file_dialog_for_textbox(finaltextTextBox);
}

//=====
private: System::Void startButton_Click(System::Object^ sender, System::EventArgs^ e) {
    statusLabel->Text = "Data initializing...";
    Application::DoEvents();

    String^ path_initialtext = initialtextTextBox->Text;
    String^ path_finaltext = finaltextTextBox->Text;

    File^ file_path;
    if (file_path->Exists(path_initialtext)) {
        E_Encrypt_Mode encrypt_mode = (E_Encrypt_Mode) cryptonRadioButton->Checked;
        E_Demo_Mode demo_mode = (E_Demo_Mode) demoCheckBox->Checked;
        DWORD MK[4] = {};
        convert_string_to_dword_for_MK(mkTextBox, MK);

        statusLabel->Text = "SM4 process is working...";
        Application::DoEvents();

        if (demo_mode) {
            time_t demo_time;
            time(&demo_time);
            struct tm *demo_date_and_time = localtime(&demo_time);

            make_demonstration_note(L"\n\n\n=====
            "\n"
            "\t\tSM4 Demo (%02d.%02d.%d, %02d:%02d:%02d)\n",
            demo_date_and_time->tm_mday, demo_date_and_time->tm_mon + 1, demo_date_and_time->tm_year + 1900, demo_date_and_time->tm_hour, demo_date_and_time->tm_min,
            demo_date_and_time->tm_sec);
            make_demonstration_note(L"Initialtext is %s\nFinaltext is %s\nEncryption mode is
%s\n",

            (LPCWSTR) Marshal::StringToHGlobalUni(path_initialtext).ToPointer(),

            (LPCWSTR) Marshal::StringToHGlobalUni(path_finaltext).ToPointer(),
            encrypt_mode == EEM_Encrypt ? L"Encryption" : L"Decryption");
        }

        SM4Process(path_initialtext, path_finaltext, encrypt_mode, MK, demo_mode);
        statusLabel->Text = "Done! Result saved in " + finaltextTextBox->Text;

        if (settingsiniCheckBox->Checked)
            save_new_settingsfile();
    }
    else
        statusLabel->Text = "Invalid paths to initialtext file";
}

//=====
///

```

```
///
///
///
```

```
private: System::Void initialtextTextBox_TextChanged(System::Object^ sender, System::EventArgs^ e) {
    check_textboxes();
}
private: System::Void finaltextTextBox_TextChanged(System::Object^ sender, System::EventArgs^ e) {
    check_textboxes();
}
private: System::Void mkTextBox_TextChanged(System::Object^ sender, System::EventArgs^ e) {
    check_textboxes();
}

private: System::Void lookmkButton_Click(System::Object^ sender, System::EventArgs^ e) {
    mkTextBox->PasswordChar = mkTextBox->PasswordChar? '\0' : PASS_CHAR;
}
private: System::Void mkTextBox_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs^ e) {
    e->Handled = mkTextBox->Text->Length >= MK_MAXLENGTH;
    if (e->KeyChar == (char)Keys::Back)
        e->Handled = false;
}
};

}
```

2. Файл Main.cpp

```
#include "Main.h"

using namespace SM4App;

[STAThread]
int WINAPI WinMain(_In_ HINSTANCE hInstance, _In_opt_ HINSTANCE hPrevInstance, _In_ LPSTR lpCmdLine, _In_ int
nShowCmd) {
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Application::Run(gcnew Main);
    return 0;
}
```

3. Config.h

```
#pragma once
#include <Windows.h>
#include <string>
#include <fstream>
#include <sstream>
#include <stdarg.h>
#include <ctime>

#define HEX_DWORD_BLOCK_SIZE 8
#define BIN_DWORD_BLOCK_SIZE HEX_DWORD_BLOCK_SIZE * 4
#define MK_MAXLENGTH 16
#define BACKSPACE 8
#define PASS_CHAR '*'
#define DEMO_FILE_PATH "Demonstration.txt"

using namespace std;
```

```

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Text::RegularExpressions;
using namespace System::IO;
using namespace System::Runtime::InteropServices;

//=====
string Input_from_File(ifstream&in); //
Внесення всього тексту у рядок (враховуючи переноси рядка)
void open_file_dialog_for_textbox(TextBox^ textbox); // Відкриття файлу через вікно
OpenFileDialog
void convert_string_to_dword_for_MK(TextBox^ MK_textbox, DWORD MK[]); // Перетворення введенного ключа в
16-байтовий масив
void fill_text(string& text); //
Заповнення останнього блоку тексту нулями
string translate_hex_to_text(string& hex_text); //
Конвертація та зберігання тексту у символічному вигляді у файл
string translate_text_to_hex(string& text); //
Конвертація та зберігання тексту у шістнадцятковому вигляді у файл
void take_hex_value_for_dword(DWORD X[], string& hex_text, size_t index); // Функція, яка бере шістнадцяткове
значення з рядка та присвоює (X0, X1, X2, X3)
void save_finaltext_dword(DWORD X[], string& finaltext_hex); // Функція, яка зберігає результат
шифрування у finaltext
void make_demonstration_note(const wchar_t* format, ...); // Додання запису у файл
Demonstration.txt
//=====
namespace SM4_Consts {
    const BYTE SBox[256] = {
        0xd6, 0x90, 0xe9, 0xfe, 0xcc, 0xe1, 0x3d, 0xb7, 0x16, 0xb6, 0x14, 0xc2, 0x28, 0xfb, 0x2c, 0x05,
        0x2b, 0x67, 0x9a, 0x76, 0x2a, 0xbe, 0x04, 0xc3, 0xaa, 0x44, 0x13, 0x26, 0x49, 0x86, 0x06, 0x99,
        0x9c, 0x42, 0x50, 0xf4, 0x91, 0xef, 0x98, 0x7a, 0x33, 0x54, 0x0b, 0x43, 0xed, 0xcf, 0xac, 0x62,
        0xe4, 0xb3, 0x1c, 0xa9, 0xc9, 0x08, 0xe8, 0x95, 0x80, 0xdf, 0x94, 0xfa, 0x75, 0x8f, 0x3f, 0xa6,
        0x47, 0x07, 0xa7, 0xfc, 0xf3, 0x73, 0x17, 0xba, 0x83, 0x59, 0x3c, 0x19, 0xe6, 0x85, 0x4f, 0xa8,
        0x68, 0x6b, 0x81, 0xb2, 0x71, 0x64, 0xda, 0x8b, 0xf8, 0xeb, 0x0f, 0x4b, 0x70, 0x56, 0x9d, 0x35,
        0x1e, 0x24, 0x0e, 0x5e, 0x63, 0x58, 0xd1, 0xa2, 0x25, 0x22, 0x7c, 0x3b, 0x01, 0x21, 0x78, 0x87,
        0xd4, 0x00, 0x46, 0x57, 0x9f, 0xd3, 0x27, 0x52, 0x4c, 0x36, 0x02, 0xe7, 0xa0, 0xc4, 0xc8, 0x9e,
        0xea, 0xbf, 0x8a, 0xd2, 0x40, 0xc7, 0x38, 0xb5, 0xa3, 0xf7, 0xf2, 0xce, 0xf9, 0x61, 0x15, 0xa1,
        0xe0, 0xae, 0x5d, 0xa4, 0x9b, 0x34, 0x1a, 0x55, 0xad, 0x93, 0x32, 0x30, 0xf5, 0x8c, 0xb1, 0xe3,
        0x1d, 0xf6, 0xe2, 0x2e, 0x82, 0x66, 0xca, 0x60, 0xc0, 0x29, 0x23, 0xab, 0x0d, 0x53, 0x4e, 0x6f,
        0xd5, 0xdb, 0x37, 0x45, 0xde, 0xfd, 0x8e, 0x2f, 0x03, 0xff, 0x6a, 0x72, 0x6d, 0x6c, 0x5b, 0x51,
        0x8d, 0x1b, 0xaf, 0x92, 0xbb, 0xdd, 0xbc, 0x7f, 0x11, 0xd9, 0x5c, 0x41, 0x1f, 0x10, 0x5a, 0xd8,
        0x0a, 0xc1, 0x31, 0x88, 0xa5, 0xcd, 0x7b, 0xbd, 0x2d, 0x74, 0xd0, 0x12, 0xb8, 0xe5, 0xb4, 0xb0,
        0x89, 0x69, 0x97, 0x4a, 0x0c, 0x96, 0x77, 0x7e, 0x65, 0xb9, 0xf1, 0x09, 0xc5, 0x6e, 0xc6, 0x84,
        0x18, 0xf0, 0x7d, 0xec, 0x3a, 0xdc, 0x4d, 0x20, 0x79, 0xee, 0x5f, 0x3e, 0xd7, 0xcb, 0x39, 0x48
    };
    const DWORD FK[4] = { 0xa3b1bac6, 0x56aa3350, 0x677d9197, 0xb27022dc };
    const DWORD CK[32] = {
        0x00070e15, 0x1c232a31, 0x383f464d, 0x545b6269,
        0x70777e85, 0x8c939aa1, 0xa8afb6bd, 0xc4cbd2d9,
        0xe0e7eef5, 0xfc030a11, 0x181f262d, 0x343b4249,
        0x50575e65, 0x6c737a81, 0x888f969d, 0xa4abb2b9,
        0xc0c7ced5, 0xdce3eaf1, 0xf8ff060d, 0x141b2229,
        0x30373e45, 0x4c535a61, 0x686f767d, 0x848b9299,
        0xa0a7aeb5, 0xbcc3cad1, 0xd8dfe6ed, 0xf4fb0209,
        0x10171e25, 0x2c333a41, 0x484f565d, 0x646b7279
    };
}

//=====

```

4. Файл Config.cpp

```
#include "Config.h"

//=====
// Внесення всього тексту у рядок (враховуючи переноси рядка)
string Input_from_File(ifstream& in) {
    string str;
    int c;
    while ((c = in.get()) != EOF)
        str += (char)c;
    return str;
}

//=====
// Відкриття файлу через вікно OpenFileDialog
void open_file_dialog_for_textbox(TextBox^ textbox) {
    OpenFileDialog^ ofd = gcnew OpenFileDialog;
    ofd->Filter = "All files (*.*) | *.*";
    if (ofd->ShowDialog() == System::Windows::Forms::DialogResult::OK)
        textbox->Text = ofd->FileName;
}

//=====
// Перетворення введенного ключа в 16-байтовий масив
void convert_string_to_dword_for_MK(TextBox^ MK_textbox, DWORD MK[]) {
    for (BYTE i = 0, k = 0; i < MK_textbox->Text->Length; i++, k = i / 4) {
        MK[k] = (MK[k] << 8) | MK_textbox->Text[i];
        if (i == MK_textbox->Text->Length - 1)
            MK[k] <<= BIN_DWORD_BLOCK_SIZE - MK_textbox->Text->Length % 4 * 8;
    }
}

//=====
// Заповнення останнього блоку тексту нулями
void fill_text(string& text) {
    BYTE init_length_mod = text.length() % (HEX_DWORD_BLOCK_SIZE * 2);
    DWORD zero_count = init_length_mod ? HEX_DWORD_BLOCK_SIZE * 2 - init_length_mod : 0;
    text.insert(text.end(), zero_count, 0);
}

//=====
// Конвертація та зберігання тексту у символічному вигляді у файл
string translate_hex_to_text(string& hex_text) {
    string ascii_string = "";
    for (size_t i = 0; i < hex_text.length(); i += 2) {
        string byte_string = hex_text.substr(i, 2);
        char byte = (char)strtol(byte_string.c_str(), NULL, 16);
        ascii_string.push_back(byte);
    }
    return ascii_string;
}

//=====
// Конвертація та зберігання тексту у шістнадцятковому вигляді у файл
string translate_text_to_hex(string& text) {
    stringstream ss;
    for (size_t i = 0; i < text.length(); i++) {
        char hex_char[3];
        sprintf(hex_char, "%02x", (BYTE)text[i]);
        ss << hex_char;
    }

    return ss.str();
}
```

```
//=====
// Функція, яка бере шістнадцяткове значення з рядка та присвоює (X0, X1, X2, X3)
void take_hex_value_for_dword(DWORD X[], string& hex_text, size_t index){
    for (BYTE n = 0; n < 4; n++) {
        DWORD value = 0;
        for (size_t j = 0; j < HEX_DWORD_BLOCK_SIZE; j++) {
            char hex_char = hex_text[index + j + (size_t)n * 8];
            if (hex_char >= '0' && hex_char <= '9')
                value = (value << 4) | (hex_char - '0');
            else if (hex_char >= 'A' && hex_char <= 'F')
                value = (value << 4) | (hex_char - 'A' + 10);
            else if (hex_char >= 'a' && hex_char <= 'f')
                value = (value << 4) | (hex_char - 'a' + 10);
        }
        X[n] = value;
    }
}
//=====
// Функція, яка зберігає результат шифрування у finaltext
void save_finaltext_dword(DWORD X[], string& finaltext_hex) {
    stringstream ss;
    char X_value[9] = {};
    for (BYTE j = 0; j < 4; j++) {
        sprintf(X_value, "%08x", X[3 - j]);
        ss << X_value;
    }

    string X_hex_string = ss.str();
    finaltext_hex += X_hex_string;
}
//=====
// Додання запису у файл Demonstration.txt
void make_demonstration_note(const wchar_t* format, ...) {
    FILE* demo_file = fopen(DEMO_FILE_PATH, "a");

    va_list args;
    va_start(args, format);
    vfwprintf(demo_file, format, args);
    va_end(args);

    fclose(demo_file);
}
//=====
```

5. Файл SM4.h

```
#pragma once
#include "Config.h"
#define HEX_DWORD_BLOCK_SIZE 8
// Розбір 32-бітного блока у шістнадцятковому вигляді
#define C_SHL(x, k) (((x) << (k)) | ((x) >> (sizeof(x)*8L - (k)))) // Операція
циклічного зсуву уліво

//=====
enum E_Encrypt_Mode {
    EEM_Decrypt,
    EEM_Encrypt
};
//=====
```

```
enum E_Demo_Mode{
    EDM_Disabled,
    EDM_Enabled
};
//=====
void SM4Process(String^ path_initialtext, String^ path_finaltext, // Функція
    виконання алгоритму SM4 із заміром часу виконання
    E_Encrypt_Mode encrypt_mode, DWORD MK[], E_Demo_Mode demo_mode); //
//=====
```

6. Файл SM4.cpp

```
#include "SM4.h"

using namespace SM4_Consts;

//=====
// Функція заміни T
DWORD T(DWORD x, E_Encrypt_Mode is_encrypt, E_Demo_Mode is_demo_mode){
    if (is_demo_mode)
        make_demonstration_note(L"\n\tT function:\n");

    // 1. SBox()
    BYTE q[4] = {};
    DWORD tau = 0;

    if (is_demo_mode)
        make_demonstration_note(L"\t1) S-box:\n");

    for (BYTE n = 0, j = 0; n <= 24 && j < 4; n += 8, j++){
        q[j] = (BYTE)((x >> n) & 0xFF);

        if (is_demo_mode)
            make_demonstration_note(L"\t\t'%02X' = '%02X'\n", q[j], SBox[q[j]]);

        q[j] = SBox[q[j]];
        tau ^= q[j] << n;
    }
    if (is_demo_mode)
        make_demonstration_note(L"\tS-box result: Tau = %08X\n", tau);

    // 2. L(tau)
    if (is_encrypt == EEM_Encrypt) {
        if (is_demo_mode)
            make_demonstration_note(L"\t2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18)
xor (Tau <<< 24) = \n"
                                   "\t = %08X xor (%08X <<< 2) xor (%08X <<< 10) xor (%08X <<< 18) xor (%08X <<< 24) =
%08X\n",
                                   tau, tau, tau, tau, tau, tau ^ C_SHL(tau, 2) ^ C_SHL(tau, 10) ^ C_SHL(tau, 18) ^
C_SHL(tau, 24));

        return tau ^ C_SHL(tau, 2) ^ C_SHL(tau, 10) ^ C_SHL(tau, 18) ^ C_SHL(tau, 24);
    }
    else {
        if (is_demo_mode)
            make_demonstration_note(L"\t2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = %08X xor (%08X
<<< 13) xor (%08X <<< 23) = %08X\n",
                                   tau, tau, tau, tau ^ C_SHL(tau, 13) ^ C_SHL(tau, 23));

        return tau ^ C_SHL(tau, 13) ^ C_SHL(tau, 23);
    }
}
```

```

    }
}

//=====
// Генерація раундових ключів
void Key_Expansion(DWORD MK[], DWORD rk[], E_Demo_Mode is_demo_mode) {
    if (is_demo_mode)

        make_demonstration_note(L"=====Key
Expansion=====\\n"
                                "MK = %08X %08X %08X %08X;\\n", MK[0], MK[1],
MK[2], MK[3]);

    DWORD K[4] = {};
    K[0] = MK[0] ^ FK[0];
    K[1] = MK[1] ^ FK[1];
    K[2] = MK[2] ^ FK[2];
    K[3] = MK[3] ^ FK[3];

    if (is_demo_mode)
        make_demonstration_note(L"\\nK[0] = MK[0] xor FK[0] = %08X xor %08X = %08X\\n"
                                "K[1] = MK[1] xor FK[1] = %08X xor %08X = %08X\\n"
                                "K[2] = MK[2] xor FK[2] = %08X xor %08X = %08X\\n"
                                "K[3] = MK[3] xor FK[3] = %08X xor %08X = %08X\\n",
                                MK[0], FK[0], K[0],
                                MK[1], FK[1], K[1],
                                MK[2], FK[2], K[2],
                                MK[3], FK[3], K[3]);

    for (BYTE i = 0; i < 32; i++) {
        rk[i] = K[0] ^ T(K[1] ^ K[2] ^ K[3] ^ CK[i], EEM_Decrypt, is_demo_mode);

        if (is_demo_mode)
            make_demonstration_note(L"\\nrk[%d] = K[0] xor T'( K[1] xor K[2] xor K[3] xor CK[%d] ) = "
                                    "%08X xor T'(%08X xor %08X xor %08X xor %08X) = \\n"
                                    " = %08X xor T'(%08X) = %08X\\n",
                                    i, i, K[0], K[1], K[2], K[3], CK[i], K[0], K[1] ^ K[2] ^ K[3] ^ CK[i], rk[i]);

        K[0] = K[1];
        K[1] = K[2];
        K[2] = K[3];
        K[3] = rk[i];
    }
}

//=====
// Раунд шифрування блоку
void SM4_Block(DWORD X[], DWORD rk[], E_Encrypt_Mode is_encrypt, E_Demo_Mode is_demo_mode) {
    if (is_demo_mode)

        make_demonstration_note(L"\\n\\n=====SM4
Block Operation=====\\n\\n");

    DWORD Xtemp = 0;
    for (BYTE i = 0; i < 32; i++) {
        if (is_encrypt == EEM_Encrypt)
            Xtemp = X[0] ^ T(X[1] ^ X[2] ^ X[3] ^ rk[i], EEM_Encrypt, is_demo_mode);
        else
            Xtemp = X[0] ^ T(X[1] ^ X[2] ^ X[3] ^ rk[31 - i], EEM_Encrypt, is_demo_mode);

        if (is_demo_mode){
            if (i == 31)
                make_demonstration_note(L"\\n~~~~~X Result:");
        }
    }
}

```

```

        make_demonstration_note(L"\nX[%d] = X[%d] xor T(X[%d] xor X[%d] xor X[%d] xor rk[%d]) =
%08X xor T(%08X xor %08X xor %08X xor %08X) = \n"
        "    = %08X xor T(%08X) = %08X\n",
        i+4, i, i+1, i+2, i+3, is_encrypt ? i : 31 - i,
        X[0], X[1], X[2], X[3], rk[is_encrypt ? i : 31 - i],
        X[0], X[1] ^ X[2] ^ X[3] ^ rk[is_encrypt ? i : 31 - i], Xtemp);
    }

    X[0] = X[1];
    X[1] = X[2];
    X[2] = X[3];
    X[3] = Xtemp;
}

}

//=====
// Функція виконання алгоритму SM4
void SM4Process(String^ path_initialtext, String^ path_finaltext, E_Encrypt_Mode encrypt_mode, DWORD MK[],
E_Demo_Mode demo_mode) {
    ifstream in((LPCWSTR)Marshal::StringToHGlobalUni(path_initialtext).ToPointer(), ios::binary);
    string initialtext = Input_from_File(in);
    if (encrypt_mode == EEM_Encrypt)
        fill_text(initialtext);
    in.close();

    string initialtext_hex = translate_text_to_hex(initialtext);

    ofstream out("HEX_INPUT.txt");
    out << initialtext_hex;
    out.close();

    // Обчислення кількості блоків, на які ділиться текст повідомлення
    DWORD block_count = (DWORD)initialtext_hex.length() / (HEX_DWORD_BLOCK_SIZE * 4);
    DWORD X[4] = {}, rk[32];
    string finaltext_hex;

    // 1. Замір часу генерації ключів
    Key_Expansion(MK, rk, demo_mode);

    // 2. Замір часу шифрування блоку
    for (size_t i = 0; i < block_count; i++) {
        take_hex_value_for_dword(X, initialtext_hex, i * 32);
        i == 0 ? SM4_Block(X, rk, encrypt_mode, demo_mode) : SM4_Block(X, rk, encrypt_mode, EDM_Disabled);
        save_finaltext_dword(X, finaltext_hex);
    }

    // Вивід кінцевого тексту
    // 1. У hex-вигляді
    ofstream hex_out("HEX_OUTPUT.txt");
    hex_out << finaltext_hex;
    hex_out.close();

    // 2. У строковому вигляді
    out.open((LPCWSTR)Marshal::StringToHGlobalUni(path_finaltext).ToPointer(), ios::binary);
    string finaltext = translate_hex_to_text(finaltext_hex);
    if (encrypt_mode == EEM_Decrypt) {
        size_t last_non_null = finaltext.find_last_not_of('\0');
        finaltext.erase(last_non_null + 1);
    }
    out << finaltext;
    out.close();
}

//=====

```

ДОДАТОК Б

Вивід файлу **Demonstration.txt** під час тестування програми **SM4 Demo**

=====

SM4 Demo (05.06.2024, 21:20:27)

=====Key Expansion=====

MK = 01234567 89ABCDEF FEDCBA98 76543210;

$K[0] = MK[0] \text{ xor } FK[0] = 01234567 \text{ xor } A3B1BAC6 = A292FFA1$

$K[1] = MK[1] \text{ xor } FK[1] = 89ABCDEF \text{ xor } 56AA3350 = DF01FEBF$

$K[2] = MK[2] \text{ xor } FK[2] = FEDCBA98 \text{ xor } 677D9197 = 99A12B0F$

$K[3] = MK[3] \text{ xor } FK[3] = 76543210 \text{ xor } B27022DC = C42410CC$

T function:

1) S-box:

'69' = '22'

'CB' = '41'

'83' = 'D2'

'82' = '8A'

S-box result: Tau = 8AD24122

2) L(tau): $\text{Tau xor (Tau} \lll 13) \text{ xor (Tau} \lll 23) = 8AD24122 \text{ xor}$
 $(8AD24122 \lll 13) \text{ xor } (8AD24122 \lll 23) = 53B37958$

$rk[0] = K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[0]) = A292FFA1 \text{ xor}$

$T'(DF01FEBF \text{ xor } 99A12B0F \text{ xor } C42410CC \text{ xor } 00070E15) =$

$= A292FFA1 \text{ xor } T'(8283CB69) = F12186F9$

T function:

1) S-box:

'0B' = 'C2'

'97' = '55'

'87' = 'B5'

'B0' = 'D5'

S-box result: $\text{Tau} = \text{D5B555C2}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = \text{D5B555C2} \text{ xor } (\text{D5B555C2} \lll 13) \text{ xor } (\text{D5B555C2} \lll 23) = 9\text{E}67\text{D5DE}$

$\text{rk}[1] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[1]) = \text{DF01FEBF} \text{ xor } \text{T}'(99\text{A12B0F} \text{ xor } \text{C42410CC} \text{ xor } \text{F12186F9} \text{ xor } 1\text{C232A31}) =$
 $= \text{DF01FEBF} \text{ xor } \text{T}'(\text{B087970B}) = 41662\text{B61}$

T function:

1) S-box:

'19' = '44'

'FB' = '3E'

'5C' = '70'

'4C' = 'E6'

S-box result: $\text{Tau} = \text{E6703E44}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = \text{E6703E44} \text{ xor } (\text{E6703E44} \lll 13) \text{ xor } (\text{E6703E44} \lll 23) = \text{C3CB9A95}$

$\text{rk}[2] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[2]) = 99\text{A12B0F} \text{ xor } \text{T}'(\text{C42410CC} \text{ xor } \text{F12186F9} \text{ xor } 41662\text{B61} \text{ xor } 383\text{F464D}) =$
 $= 99\text{A12B0F} \text{ xor } \text{T}'(4\text{C5CFB19}) = 5\text{A6AB19A}$

T function:

1) S-box:

'6B' = '3B'

'7E' = 'C8'

'76' = '27'

'BE' = '5B'

S-box result: Tau = 5B27C83B

2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 5B27C83B xor
(5B27C83B <<< 13) xor (5B27C83B <<< 23) = BF8D30BB

rk[3] = K[0] xor T'(K[1] xor K[2] xor K[3] xor CK[3]) = C42410CC xor
T'(F12186F9 xor 41662B61 xor 5A6AB19A xor 545B6269) =
= C42410CC xor T'(BE767E6B) = 7BA92077

T function:

1) S-box:

'09' = 'B6'

'C4' = 'BB'

'D2' = '31'

'10' = '2B'

S-box result: Tau = 2B31BBB6

2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 2B31BBB6 xor
(2B31BBB6 <<< 13) xor (2B31BBB6 <<< 23) = C752E60D

rk[4] = K[0] xor T'(K[1] xor K[2] xor K[3] xor CK[4]) = F12186F9 xor
T'(41662B61 xor 5A6AB19A xor 7BA92077 xor 70777E85) =
= F12186F9 xor T'(10D2C409) = 367360F4

T function:

1) S-box:

'B8' = '03'

'6B' = '3B'

'23' = 'F4'

'9B' = '30'

S-box result: $\text{Tau} = 30\text{F}43\text{B}03$

2) $\text{L}(\text{tau})$: $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 30\text{F}43\text{B}03 \text{ xor } (30\text{F}43\text{B}03 \lll 13) \text{ xor } (30\text{F}43\text{B}03 \lll 23) = 360\text{C}2700$

$\text{rk}[5] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[5]) = 41662\text{B}61 \text{ xor } \text{T}'(5\text{A}6\text{A}\text{B}19\text{A} \text{ xor } 7\text{B}\text{A}92077 \text{ xor } 367360\text{F}4 \text{ xor } 8\text{C}939\text{A}\text{A}1) =$
 $= 41662\text{B}61 \text{ xor } \text{T}'(9\text{B}236\text{B}\text{B}8) = 776\text{A}0\text{C}61$

T function:

1) S-box:

'5F' = '35'

'FA' = '5F'

'1F' = '99'

'92' = '5D'

S-box result: $\text{Tau} = 5\text{D}995\text{F}35$

2) $\text{L}(\text{tau})$: $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 5\text{D}995\text{F}35 \text{ xor } (5\text{D}995\text{F}35 \lll 13) \text{ xor } (5\text{D}995\text{F}35 \lll 23) = \text{ECD}13829$

$\text{rk}[6] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[6]) = 5\text{A}6\text{A}\text{B}19\text{A} \text{ xor } \text{T}'(7\text{B}\text{A}92077 \text{ xor } 367360\text{F}4 \text{ xor } 776\text{A}0\text{C}61 \text{ xor } \text{A}8\text{A}\text{F}\text{B}6\text{B}\text{D}) =$
 $= 5\text{A}6\text{A}\text{B}19\text{A} \text{ xor } \text{T}'(921\text{F}\text{F}\text{A}5\text{F}) = \text{B}6\text{B}\text{B}89\text{B}3$

T function:

1) S-box:

'FF' = '48'

'37' = '95'

'69' = '22'

'33' = 'A9'

S-box result: $\text{Tau} = \text{A}9229548$

2) $\text{L}(\text{tau})$: $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = \text{A}9229548 \text{ xor } (\text{A}9229548 \lll 13) \text{ xor } (\text{A}9229548 \lll 23) = \text{A}9229548$

$$\lll 13) \text{ xor } (A9229548 \lll 23) = 5FDF1126$$

$$\begin{aligned} \text{rk}[7] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[7]) = 7BA92077 \text{ xor} \\ T'(367360F4 \text{ xor } 776A0C61 \text{ xor } B6BB89B3 \text{ xor } C4CBD2D9) &= \\ &= 7BA92077 \text{ xor } T'(336937FF) = 24763151 \end{aligned}$$

T function:

1) S-box:

'76' = '27'

'5A' = '0F'

'40' = '47'

'05' = 'E1'

S-box result: Tau = E1470F27

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= E1470F27 \text{ xor } (E1470F27 \\ \lll 13) \text{ xor } (E1470F27 \lll 23) &= 93535088 \end{aligned}$$

$$\begin{aligned} \text{rk}[8] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[8]) = 367360F4 \text{ xor} \\ T'(776A0C61 \text{ xor } B6BB89B3 \text{ xor } 24763151 \text{ xor } E0E7EEF5) &= \\ &= 367360F4 \text{ xor } T'(05405A76) = A520307C \end{aligned}$$

T function:

1) S-box:

'8F' = 'A1'

'82' = '8A'

'EE' = 'C6'

'CB' = '41'

S-box result: Tau = 41C68AA1

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= 41C68AA1 \text{ xor} \\ (41C68AA1 \lll 13) \text{ xor } (41C68AA1 \lll 23) &= C03241DC \end{aligned}$$

$$\begin{aligned}
 rk[9] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[9]) = 776A0C61 \text{ xor} \\
 &T'(B6BB89B3 \text{ xor } 24763151 \text{ xor } A520307C \text{ xor } FC030A11) = \\
 &= 776A0C61 \text{ xor } T'(CBEE828F) = B7584DBD
 \end{aligned}$$

T function:

1) S-box:

'BD' = '6C'

'6A' = '7C'

'11' = '67'

'2E' = 'AC'

S-box result: Tau = AC677C6C

$$\begin{aligned}
 2) L(\tau): \tau \text{ xor } (\tau \lll 13) \text{ xor } (\tau \lll 23) &= AC677C6C \text{ xor} \\
 (AC677C6C \lll 13) \text{ xor } (AC677C6C \lll 23) &= 75BCDA5E
 \end{aligned}$$

$$\begin{aligned}
 rk[10] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[10]) = B6BB89B3 \text{ xor} \\
 &T'(24763151 \text{ xor } A520307C \text{ xor } B7584DBD \text{ xor } 181F262D) = \\
 &= B6BB89B3 \text{ xor } T'(2E116ABD) = C30753ED
 \end{aligned}$$

T function:

1) S-box:

'65' = '58'

'6C' = '01'

'44' = 'F3'

'E5' = '96'

S-box result: Tau = 96F30158

$$\begin{aligned}
 2) L(\tau): \tau \text{ xor } (\tau \lll 13) \text{ xor } (\tau \lll 23) &= 96F30158 \text{ xor } (96F30158 \\
 \lll 13) \text{ xor } (96F30158 \lll 23) &= 5A936A06
 \end{aligned}$$

$$\begin{aligned}
 rk[11] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[11]) = 24763151 \text{ xor} \\
 &T'(A520307C \text{ xor } B7584DBD \text{ xor } C30753ED \text{ xor } 343B4249) =
 \end{aligned}$$

$$= 24763151 \text{ xor } T'(E5446C65) = 7EE55B57$$

T function:

1) S-box:

'62' = '0E'

'1B' = '26'

'ED' = '6E'

'5A' = '0F'

S-box result: $\text{Tau} = 0F6E260E$

$$\begin{aligned} 2) L(\text{tau}): \text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= 0F6E260E \text{ xor} \\ (0F6E260E \lll 13) \text{ xor } (0F6E260E \lll 23) &= CCA850F0 \end{aligned}$$

$$\begin{aligned} \text{rk}[12] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[12]) = A520307C \text{ xor} \\ T'(B7584DBD \text{ xor } C30753ED \text{ xor } 7EE55B57 \text{ xor } 50575E65) &= \\ &= A520307C \text{ xor } T'(5AED1B62) = 6988608C \end{aligned}$$

T function:

1) S-box:

'B7' = '2F'

'12' = '9A'

'19' = '44'

'B8' = '03'

S-box result: $\text{Tau} = 03449A2F$

$$\begin{aligned} 2) L(\text{tau}): \text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= 03449A2F \text{ xor} \\ (03449A2F \lll 13) \text{ xor } (03449A2F \lll 23) &= 8780D80A \end{aligned}$$

$$\begin{aligned} \text{rk}[13] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[13]) = B7584DBD \text{ xor} \\ T'(C30753ED \text{ xor } 7EE55B57 \text{ xor } 6988608C \text{ xor } 6C737A81) &= \\ &= B7584DBD \text{ xor } T'(B81912B7) = 30D895B7 \end{aligned}$$

T function:

1) S-box:

'F1' = 'F0'

'38' = '80'

'3A' = '94'

'AF' = '6F'

S-box result: $\text{Tau} = 6F9480F0$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 6F9480F0 \text{ xor } (6F9480F0 \lll 13) \text{ xor } (6F9480F0 \lll 23) = 87BD4742$

$\text{rk}[14] = K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[14]) = C30753ED \text{ xor } T'(7EE55B57 \text{ xor } 6988608C \text{ xor } 30D895B7 \text{ xor } 888F969D) = C30753ED \text{ xor } T'(AF3A38F1) = 44BA14AF$

T function:

1) S-box:

'2D' = 'CF'

'53' = 'B2'

'41' = '07'

'B9' = 'FF'

S-box result: $\text{Tau} = FF07B2CF$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = FF07B2CF \text{ xor } (FF07B2CF \lll 13) \text{ xor } (FF07B2CF \lll 23) = 6EA1CEF6$

$\text{rk}[15] = K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[15]) = 7EE55B57 \text{ xor } T'(6988608C \text{ xor } 30D895B7 \text{ xor } 44BA14AF \text{ xor } A4ABB2B9) = 7EE55B57 \text{ xor } T'(B941532D) = 104495A1$

T function:

1) S-box:

'6C' = '01'

'DA' = 'D0'

'E1' = '69'

'A4' = '82'

S-box result: $\text{Tau} = 8269\text{D}001$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 8269\text{D}001 \text{ xor } (8269\text{D}001 \lll 13) \text{ xor } (8269\text{D}001 \lll 23) = \text{B}8\text{A}8\text{D}4\text{A}4$

$\text{rk}[16] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[16]) = 6988608\text{C} \text{ xor } \text{T}'(30\text{D}895\text{B}7 \text{ xor } 44\text{B}\text{A}14\text{A}\text{F} \text{ xor } 104495\text{A}1 \text{ xor } \text{C}0\text{C}7\text{C}\text{E}\text{D}5) =$
 $= 6988608\text{C} \text{ xor } \text{T}'(\text{A}4\text{E}1\text{D}\text{A}6\text{C}) = \text{D}120\text{B}428$

T function:

1) S-box:

'D7' = 'BD'

'DF' = 'B0'

'3D' = '8F'

'59' = 'EB'

S-box result: $\text{Tau} = \text{E}8\text{B}\text{F}0\text{B}\text{D}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = \text{E}8\text{B}\text{F}0\text{B}\text{D} \text{ xor } (\text{E}8\text{B}\text{F}0\text{B}\text{D} \lll 13) \text{ xor } (\text{E}8\text{B}\text{F}0\text{B}\text{D} \lll 23) = 436\text{D}\text{C}\text{A}14$

$\text{rk}[17] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[17]) = 30\text{D}895\text{B}7 \text{ xor } \text{T}'(44\text{B}\text{A}14\text{A}\text{F} \text{ xor } 104495\text{A}1 \text{ xor } \text{D}120\text{B}428 \text{ xor } \text{D}\text{C}\text{E}3\text{E}\text{A}\text{F}1) =$
 $= 30\text{D}895\text{B}7 \text{ xor } \text{T}'(593\text{D}\text{D}\text{F}\text{D}7) = 73\text{B}55\text{F}\text{A}3$

T function:

1) S-box:

'27' = '7A'

'78' = '4C'

'2E' = 'AC'

'4A' = '3C'

S-box result: Tau = 3CAC4C7A

2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 3CAC4C7A xor
(3CAC4C7A <<< 13) xor (3CAC4C7A <<< 23) = 883D5DC9

rk[18] = K[0] xor T'(K[1] xor K[2] xor K[3] xor CK[18]) = 44BA14AF xor
T'(104495A1 xor D120B428 xor 73B55FA3 xor F8FF060D) =
= 44BA14AF xor T'(4A2E7827) = CC874966

T function:

1) S-box:

'C4' = 'BB'

'80' = 'EA'

'09' = 'B6'

'7A' = '02'

S-box result: Tau = 02B6EABB

2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 02B6EABB xor
(02B6EABB <<< 13) xor (02B6EABB <<< 23) = 8260D198

rk[19] = K[0] xor T'(K[1] xor K[2] xor K[3] xor CK[19]) = 104495A1 xor
T'(D120B428 xor 73B55FA3 xor CC874966 xor 141B2229) =
= 104495A1 xor T'(7A0980C4) = 92244439

T function:

1) S-box:

'B9' = 'FF'

'6C' = '01'

'21' = '42'

'1D' = '86'

S-box result: $\text{Tau} = 864201\text{FF}$

2) $L(\text{tau})$: $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 864201\text{FF} \text{ xor } (864201\text{FF} \lll 13) \text{ xor } (864201\text{FF} \lll 23) = 39\text{BED}037$

$\text{rk}[20] = K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } \text{CK}[20]) = \text{D120B428} \text{ xor } T'(73\text{B55FA3} \text{ xor } \text{CC874966} \text{ xor } 92244439 \text{ xor } 30373\text{E45}) =$
 $= \text{D120B428} \text{ xor } T'(1\text{D216CB9}) = \text{E89E641F}$

T function:

1) S-box:

'21' = '42'

'33' = 'A9'

'6E' = '78'

'FA' = '5F'

S-box result: $\text{Tau} = 5\text{F78A942}$

2) $L(\text{tau})$: $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 5\text{F78A942} \text{ xor } (5\text{F78A942} \lll 13) \text{ xor } (5\text{F78A942} \lll 23) = \text{EB7F5EF9}$

$\text{rk}[21] = K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } \text{CK}[21]) = 73\text{B55FA3} \text{ xor } T'(\text{CC874966} \text{ xor } 92244439 \text{ xor } \text{E89E641F} \text{ xor } 4\text{C535A61}) =$
 $= 73\text{B55FA3} \text{ xor } T'(\text{FA6E3321}) = 98\text{CA015A}$

T function:

1) S-box:

'01' = '90'

'57' = '8B'

'1F' = '99'

'8A' = 'F2'

S-box result: $\text{Tau} = \text{F2998B90}$

2) $L(\text{tau})$: $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = \text{F2998B90} \text{ xor } (\text{F2998B90} \lll 13) \text{ xor } (\text{F2998B90} \lll 23)$

$$\lll 13) \text{ xor } (F2998B90 \lll 23) = 0B92D906$$

$$\begin{aligned} \text{rk}[22] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[22]) = CC874966 \text{ xor} \\ T'(92244439 \text{ xor } E89E641F \text{ xor } 98CA015A \text{ xor } 686F767D) &= \\ &= CC874966 \text{ xor } T'(8A1F5701) = C7159060 \end{aligned}$$

T function:

1) S-box:

'BC' = '6D'

'67' = 'A2'

'CA' = '5C'

'33' = 'A9'

S-box result: Tau = A95CA26D

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= A95CA26D \text{ xor} \\ (A95CA26D \lll 13) \text{ xor } (A95CA26D \lll 23) &= 0BC5B917 \end{aligned}$$

$$\begin{aligned} \text{rk}[23] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[23]) = 92244439 \text{ xor} \\ T'(E89E641F \text{ xor } 98CA015A \text{ xor } C7159060 \text{ xor } 848B9299) &= \\ &= 92244439 \text{ xor } T'(33CA67BC) = 99E1FD2E \end{aligned}$$

T function:

1) S-box:

'A1' = 'F6'

'C2' = 'AF'

'99' = '93'

'66' = 'D1'

S-box result: Tau = D193AFF6

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= D193AFF6 \text{ xor} \\ (D193AFF6 \lll 13) \text{ xor } (D193AFF6 \lll 23) &= 5F05BC13 \end{aligned}$$

$$\begin{aligned} rk[24] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[24]) = E89E641F \text{ xor} \\ &T'(98CA015A \text{ xor } C7159060 \text{ xor } 99E1FD2E \text{ xor } A0A7AEB5) = \\ &= E89E641F \text{ xor } T'(6699C2A1) = B79BD80C \end{aligned}$$

T function:

1) S-box:

'93' = 'A4'

'7F' = '9E'

'AC' = '0D'

'55' = '64'

S-box result: Tau = 640D9EA4

$$\begin{aligned} 2) L(\tau): \tau \text{ xor } (\tau \lll 13) \text{ xor } (\tau \lll 23) &= 640D9EA4 \text{ xor} \\ (640D9EA4 \lll 13) \text{ xor } (640D9EA4 \lll 23) &= 85EB14EA \end{aligned}$$

$$\begin{aligned} rk[25] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[25]) = 98CA015A \text{ xor} \\ &T'(C7159060 \text{ xor } 99E1FD2E \text{ xor } B79BD80C \text{ xor } BCC3CAD1) = \\ &= 98CA015A \text{ xor } T'(55AC7F93) = 1D2115B0 \end{aligned}$$

T function:

1) S-box:

'7F' = '9E'

'D6' = '7B'

'84' = '40'

'EB' = '09'

S-box result: Tau = 09407B9E

$$\begin{aligned} 2) L(\tau): \tau \text{ xor } (\tau \lll 13) \text{ xor } (\tau \lll 23) &= 09407B9E \text{ xor} \\ (09407B9E \lll 13) \text{ xor } (09407B9E \lll 23) &= C9371A8B \end{aligned}$$

$$\begin{aligned} rk[26] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[26]) = C7159060 \text{ xor} \\ &T'(99E1FD2E \text{ xor } B79BD80C \text{ xor } 1D2115B0 \text{ xor } D8DFE6ED) = \end{aligned}$$

$$= C7159060 \text{ xor } T'(EB84D67F) = 0E228AEB$$

T function:

1) S-box:

$$'5E' = '9D'$$

$$'45' = '73'$$

$$'63' = '5E'$$

$$'50' = '68'$$

$$\text{S-box result: } \text{Tau} = 685E739D$$

$$\begin{aligned} 2) L(\text{tau}): \text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= 685E739D \text{ xor} \\ (685E739D \lll 13) \text{ xor } (685E739D \lll 23) &= 6899F1AF \end{aligned}$$

$$\begin{aligned} \text{rk}[27] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[27]) = 99E1FD2E \text{ xor} \\ T'(B79BD80C \text{ xor } 1D2115B0 \text{ xor } 0E228AEB \text{ xor } F4FB0209) &= \\ &= 99E1FD2E \text{ xor } T'(5063455E) = F1780C81 \end{aligned}$$

T function:

1) S-box:

$$'FF' = '48'$$

$$'8D' = '61'$$

$$'6C' = '01'$$

$$'F2' = '7D'$$

$$\text{S-box result: } \text{Tau} = 7D016148$$

$$\begin{aligned} 2) L(\text{tau}): \text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) &= 7D016148 \text{ xor } (7D016148 \\ \lll 13) \text{ xor } (7D016148 \lll 23) &= F516EE58 \end{aligned}$$

$$\begin{aligned} \text{rk}[28] &= K[0] \text{ xor } T'(K[1] \text{ xor } K[2] \text{ xor } K[3] \text{ xor } CK[28]) = B79BD80C \text{ xor} \\ T'(1D2115B0 \text{ xor } 0E228AEB \text{ xor } F1780C81 \text{ xor } 10171E25) &= \\ &= B79BD80C \text{ xor } T'(F26C8DFF) = 428D3654 \end{aligned}$$

T function:

1) S-box:

'7F' = '9E'

'8A' = 'F2'

'E4' = '0C'

'91' = 'AE'

S-box result: $\text{Tau} = \text{AE0CF29E}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = \text{AE0CF29E} \text{ xor } (\text{AE0CF29E} \lll 13) \text{ xor } (\text{AE0CF29E} \lll 23) = 7F082126$

$\text{rk}[29] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[29]) = 1D2115B0 \text{ xor } \text{T}'(0E228AEB \text{ xor } F1780C81 \text{ xor } 428D3654 \text{ xor } 2C333A41) =$
 $= 1D2115B0 \text{ xor } \text{T}'(91E48A7F) = 62293496$

T function:

1) S-box:

'1E' = '06'

'58' = 'F8'

'93' = 'A4'

'99' = '93'

S-box result: $\text{Tau} = 93A4F806$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 13) \text{ xor } (\text{Tau} \lll 23) = 93A4F806 \text{ xor } (93A4F806 \lll 13) \text{ xor } (93A4F806 \lll 23) = 0FEDF80E$

$\text{rk}[30] = \text{K}[0] \text{ xor } \text{T}'(\text{K}[1] \text{ xor } \text{K}[2] \text{ xor } \text{K}[3] \text{ xor } \text{CK}[30]) = 0E228AEB \text{ xor } \text{T}'(F1780C81 \text{ xor } 428D3654 \text{ xor } 62293496 \text{ xor } 484F565D) =$
 $= 0E228AEB \text{ xor } \text{T}'(9993581E) = 01CF72E5$

T function:

1) S-box:

'5E' = '9D'

'02' = 'E9'

'00' = 'D6'

'45' = '73'

S-box result: Tau = 73D6E99D

2) L(tau): Tau xor (Tau <<< 13) xor (Tau <<< 23) = 73D6E99D xor
(73D6E99D <<< 13) xor (73D6E99D <<< 23) = 605CAC93

rk[31] = K[0] xor T'(K[1] xor K[2] xor K[3] xor CK[31]) = F1780C81 xor
T'(428D3654 xor 62293496 xor 01CF72E5 xor 646B7279) =
= F1780C81 xor T'(4500025E) = 9124A012

=====SM4 Block Operation=====

T function:

1) S-box:

'9E' = 'B1'

'C3' = '92'

'02' = 'E9'

'F0' = '18'

S-box result: Tau = 18E992B1

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau
<<< 24) =
= 18E992B1 xor (18E992B1 <<< 2) xor (18E992B1 <<< 10) xor (18E992B1
<<< 18) xor (18E992B1 <<< 24) = 26D99622

X[4] = X[0] xor T(X[1] xor X[2] xor X[3] xor rk[0]) = 01234567 xor T(89ABCDEF
xor FEDCBA98 xor 76543210 xor F12186F9) =

$$= 01234567 \text{ xor } T(F002C39E) = 27FAD345$$

T function:

1) S-box:

'AC' = '0D'

'70' = 'D4'

'14' = '2A'

'EE' = 'C6'

S-box result: Tau = C62AD40D

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \\ \lll 24) = \\ = C62AD40D \text{ xor (C62AD40D} \lll 2) \text{ xor (C62AD40D} \lll 10) \text{ xor} \\ (C62AD40D \lll 18) \text{ xor (C62AD40D} \lll 24) = 2820815D \end{aligned}$$

$$\begin{aligned} X[5] &= X[1] \text{ xor } T(X[2] \text{ xor } X[3] \text{ xor } X[4] \text{ xor rk}[1]) = 89ABCDEF \text{ xor} \\ T(FEDCBA98 \text{ xor } 76543210 \text{ xor } 27FAD345 \text{ xor } 41662B61) &= \\ &= 89ABCDEF \text{ xor } T(EE1470AC) = A18B4CB2 \end{aligned}$$

T function:

1) S-box:

'7D' = 'C4'

'1C' = '49'

'4F' = 'A8'

'AA' = '23'

S-box result: Tau = 23A849C4

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \\ \lll 24) = \\ = 23A849C4 \text{ xor (23A849C4} \lll 2) \text{ xor (23A849C4} \lll 10) \text{ xor} \\ (23A849C4 \lll 18) \text{ xor (23A849C4} \lll 24) = EF1D58B2 \end{aligned}$$

$$\begin{aligned}
 X[6] &= X[2] \text{ xor } T(X[3] \text{ xor } X[4] \text{ xor } X[5] \text{ xor } rk[2]) = \text{FEDCBA98} \text{ xor } T(76543210 \\
 &\text{ xor } 27\text{FAD345} \text{ xor } \text{A18B4CB2} \text{ xor } 5\text{A6AB19A}) = \\
 &= \text{FEDCBA98} \text{ xor } T(\text{AA4F1C7D}) = 11\text{C1E22A}
 \end{aligned}$$

T function:

1) S-box:

'AA' = '23'

'5D' = '56'

'19' = '44'

'EC' = 'C5'

S-box result: Tau = C5445623

$$\begin{aligned}
 2) \text{ L}(\tau): & \text{ Tau xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \\
 & \lll 24) = \\
 & = \text{C5445623} \text{ xor } (\text{C5445623} \lll 2) \text{ xor } (\text{C5445623} \lll 10) \text{ xor } (\text{C5445623} \\
 & \lll 18) \text{ xor } (\text{C5445623} \lll 24) = \text{BA47D0FE}
 \end{aligned}$$

$$\begin{aligned}
 X[7] &= X[3] \text{ xor } T(X[4] \text{ xor } X[5] \text{ xor } X[6] \text{ xor } rk[3]) = 76543210 \text{ xor } T(27\text{FAD345} \\
 &\text{ xor } \text{A18B4CB2} \text{ xor } 11\text{C1E22A} \text{ xor } 7\text{BA92077}) = \\
 &= 76543210 \text{ xor } T(\text{EC195DAA}) = \text{CC13E2EE}
 \end{aligned}$$

T function:

1) S-box:

'82' = '8A'

'2C' = 'ED'

'2A' = '0B'

'4A' = '3C'

S-box result: Tau = 3C0BED8A

$$\begin{aligned}
 2) \text{ L}(\tau): & \text{ Tau xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \\
 & \lll 24) = \\
 & = 3\text{C0BED8A} \text{ xor } (3\text{C0BED8A} \lll 2) \text{ xor } (3\text{C0BED8A} \lll 10) \text{ xor }
 \end{aligned}$$

$$(3C0BED8A \lll 18) \text{ xor } (3C0BED8A \lll 24) = DF868890$$

$$\begin{aligned} X[8] &= X[4] \text{ xor } T(X[5] \text{ xor } X[6] \text{ xor } X[7] \text{ xor } rk[4]) = 27FAD345 \text{ xor } T(A18B4CB2 \\ &\text{ xor } 11C1E22A \text{ xor } CC13E2EE \text{ xor } 367360F4) = \\ &= 27FAD345 \text{ xor } T(4A2A2C82) = F87C5BD5 \end{aligned}$$

T function:

1) S-box:

'70' = 'D4'

'57' = '8B'

'C4' = 'BB'

'52' = '81'

S-box result: Tau = 81BB8BD4

$$\begin{aligned} 2) L(\text{tau}): \text{ Tau xor } (\text{ Tau } \lll 2) \text{ xor } (\text{ Tau } \lll 10) \text{ xor } (\text{ Tau } \lll 18) \text{ xor } (\text{ Tau } \\ \lll 24) = \\ = 81BB8BD4 \text{ xor } (81BB8BD4 \lll 2) \text{ xor } (81BB8BD4 \lll 10) \text{ xor } \\ (81BB8BD4 \lll 18) \text{ xor } (81BB8BD4 \lll 24) = 92A94BE5 \end{aligned}$$

$$\begin{aligned} X[9] &= X[5] \text{ xor } T(X[6] \text{ xor } X[7] \text{ xor } X[8] \text{ xor } rk[5]) = A18B4CB2 \text{ xor } T(11C1E22A \\ &\text{ xor } CC13E2EE \text{ xor } F87C5BD5 \text{ xor } 776A0C61) = \\ &= A18B4CB2 \text{ xor } T(52C45770) = 33220757 \end{aligned}$$

T function:

1) S-box:

'DF' = 'B0'

'37' = '95'

'F6' = '4D'

'B1' = 'DB'

S-box result: Tau = DB4D95B0

$$2) L(\text{tau}): \text{ Tau xor } (\text{ Tau } \lll 2) \text{ xor } (\text{ Tau } \lll 10) \text{ xor } (\text{ Tau } \lll 18) \text{ xor } (\text{ Tau }$$

$$\begin{aligned}
&<<< 24) = \\
&= \text{DB4D95B0} \text{ xor } (\text{DB4D95B0} <<< 2) \text{ xor } (\text{DB4D95B0} <<< 10) \text{ xor} \\
&(\text{DB4D95B0} <<< 18) \text{ xor } (\text{DB4D95B0} <<< 24) = 663520\text{BD}
\end{aligned}$$

$$\begin{aligned}
X[10] &= X[6] \text{ xor } T(X[7] \text{ xor } X[8] \text{ xor } X[9] \text{ xor } \text{rk}[6]) = 11\text{C1E22A} \text{ xor} \\
T(\text{CC13E2EE} \text{ xor } \text{F87C5BD5} \text{ xor } 33220757 \text{ xor } \text{B6BB89B3}) &= \\
&= 11\text{C1E22A} \text{ xor } T(\text{B1F637DF}) = 77\text{F4C297}
\end{aligned}$$

T function:

1) S-box:

'44' = 'F3'

'AF' = '6F'

'DC' = 'B8'

'98' = 'AD'

S-box result: $\text{Tau} = \text{ADB86FF3}$

$$\begin{aligned}
&2) \text{L}(\text{tau}): \text{Tau} \text{ xor } (\text{Tau} <<< 2) \text{ xor } (\text{Tau} <<< 10) \text{ xor } (\text{Tau} <<< 18) \text{ xor } (\text{Tau} \\
&<<< 24) = \\
&= \text{ADB86FF3} \text{ xor } (\text{ADB86FF3} <<< 2) \text{ xor } (\text{ADB86FF3} <<< 10) \text{ xor} \\
&(\text{ADB86FF3} <<< 18) \text{ xor } (\text{ADB86FF3} <<< 24) = \text{B6851005}
\end{aligned}$$

$$\begin{aligned}
X[11] &= X[7] \text{ xor } T(X[8] \text{ xor } X[9] \text{ xor } X[10] \text{ xor } \text{rk}[7]) = \text{CC13E2EE} \text{ xor} \\
T(\text{F87C5BD5} \text{ xor } 33220757 \text{ xor } 77\text{F4C297} \text{ xor } 24763151) &= \\
&= \text{CC13E2EE} \text{ xor } T(98\text{DCAF44}) = 7\text{A96F2EB}
\end{aligned}$$

T function:

1) S-box:

'57' = '8B'

'07' = 'B7'

'60' = '1E'

'9B' = '30'

S-box result: $\text{Tau} = 301\text{EB}78\text{B}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$

$= 301\text{EB}78\text{B} \text{ xor } (301\text{EB}78\text{B} \lll 2) \text{ xor } (301\text{EB}78\text{B} \lll 10) \text{ xor } (301\text{EB}78\text{B} \lll 18) \text{ xor } (301\text{EB}78\text{B} \lll 24) = \text{DFA}69\text{BAA}$

$X[12] = X[8] \text{ xor } T(X[9] \text{ xor } X[10] \text{ xor } X[11] \text{ xor } \text{rk}[8]) = \text{F87C5BD5} \text{ xor }$

$T(33220757 \text{ xor } 77\text{F4C297} \text{ xor } 7\text{A96F2EB} \text{ xor } \text{A520307C}) =$

$= \text{F87C5BD5} \text{ xor } T(9\text{B600757}) = 27\text{DAC07F}$

T function:

1) S-box:

'BE' = '5B'

'BD' = '6C'

'E0' = '89'

'9D' = '8C'

S-box result: $\text{Tau} = 8\text{C896C5B}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$

$= 8\text{C896C5B} \text{ xor } (8\text{C896C5B} \lll 2) \text{ xor } (8\text{C896C5B} \lll 10) \text{ xor } (8\text{C896C5B} \lll 18) \text{ xor } (8\text{C896C5B} \lll 24) = 71\text{FF084E}$

$X[13] = X[9] \text{ xor } T(X[10] \text{ xor } X[11] \text{ xor } X[12] \text{ xor } \text{rk}[9]) = 33220757 \text{ xor }$

$T(77\text{F4C297} \text{ xor } 7\text{A96F2EB} \text{ xor } 27\text{DAC07F} \text{ xor } \text{B7584DBD}) =$

$= 33220757 \text{ xor } T(9\text{DE0BDBE}) = 42\text{DD0F19}$

T function:

1) S-box:

'60' = '1E'

'6E' = '78'

'96' = '1A'

'DC' = 'B8'

S-box result: Tau = B81A781E

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
 = B81A781E xor (B81A781E <<< 2) xor (B81A781E <<< 10) xor
 (B81A781E <<< 18) xor (B81A781E <<< 24) = CF511895

X[14] = X[10] xor T(X[11] xor X[12] xor X[13] xor rk[10]) = 77F4C297 xor
 T(7A96F2EB xor 27DAC07F xor 42DD0F19 xor C30753ED) =
 = 77F4C297 xor T(DC966E60) = B8A5DA02

T function:

1) S-box:

'33' = 'A9'

'4E' = '4F'

'47' = 'BA'

'A3' = '2E'

S-box result: Tau = 2EBA4FA9

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
 = 2EBA4FA9 xor (2EBA4FA9 <<< 2) xor (2EBA4FA9 <<< 10) xor
 (2EBA4FA9 <<< 18) xor (2EBA4FA9 <<< 24) = EAE7D511

X[15] = X[11] xor T(X[12] xor X[13] xor X[14] xor rk[11]) = 7A96F2EB xor
 T(27DAC07F xor 42DD0F19 xor B8A5DA02 xor 7EE55B57) =
 = 7A96F2EB xor T(A3474E33) = 907127FA

T function:

1) S-box:

'6D' = '21'

'92' = '5D'

'81' = 'BF'

'03' = 'FE'

S-box result: $\text{Tau} = \text{FEBF5D21}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$

$= \text{FEBF5D21} \text{ xor } (\text{FEBF5D21} \lll 2) \text{ xor } (\text{FEBF5D21} \lll 10) \text{ xor } (\text{FEBF5D21} \lll 18) \text{ xor } (\text{FEBF5D21} \lll 24) = \text{AC4FEBFC}$

$X[16] = X[12] \text{ xor } T(X[13] \text{ xor } X[14] \text{ xor } X[15] \text{ xor } \text{rk}[12]) = 27\text{DAC07F} \text{ xor } T(42\text{DD0F19} \text{ xor } \text{B8A5DA02} \text{ xor } 907127\text{FA} \text{ xor } 6988608\text{C}) =$
 $= 27\text{DAC07F} \text{ xor } T(0381926\text{D}) = 8\text{B952B83}$

T function:

1) S-box:

'CC' = '1F'

'43' = 'FC'

'99' = '93'

'93' = 'A4'

S-box result: $\text{Tau} = \text{A493FC1F}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$

$= \text{A493FC1F} \text{ xor } (\text{A493FC1F} \lll 2) \text{ xor } (\text{A493FC1F} \lll 10) \text{ xor } (\text{A493FC1F} \lll 18) \text{ xor } (\text{A493FC1F} \lll 24) = 96\text{F67340}$

$X[17] = X[13] \text{ xor } T(X[14] \text{ xor } X[15] \text{ xor } X[16] \text{ xor } \text{rk}[13]) = 42\text{DD0F19} \text{ xor } T(\text{B8A5DA02} \text{ xor } 907127\text{FA} \text{ xor } 8\text{B952B83} \text{ xor } 30\text{D895B7}) =$
 $= 42\text{DD0F19} \text{ xor } T(939943\text{CC}) = \text{D42B7C59}$

T function:

1) S-box:

'8F' = 'A1'

'64' = '63'

'75' = 'D3'

'8B' = 'CE'

S-box result: $\text{Tau} = \text{CED363A1}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$
 $= \text{CED363A1} \text{ xor } (\text{CED363A1} \lll 2) \text{ xor } (\text{CED363A1} \lll 10) \text{ xor } (\text{CED363A1} \lll 18) \text{ xor } (\text{CED363A1} \lll 24) = 97598233$

$X[18] = X[14] \text{ xor } T(X[15] \text{ xor } X[16] \text{ xor } X[17] \text{ xor } \text{rk}[14]) = \text{B8A5DA02} \text{ xor } T(907127\text{FA} \text{ xor } 8\text{B952B83} \text{ xor } \text{D42B7C59} \text{ xor } 44\text{BA14AF}) =$
 $= \text{B8A5DA02} \text{ xor } T(8\text{B75648F}) = 2\text{FFC5831}$

T function:

1) S-box:

'4A' = '3C'

'9A' = '32'

'06' = '3D'

'60' = '1E'

S-box result: $\text{Tau} = 1\text{E3D323C}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$
 $= 1\text{E3D323C} \text{ xor } (1\text{E3D323C} \lll 2) \text{ xor } (1\text{E3D323C} \lll 10) \text{ xor } (1\text{E3D323C} \lll 18) \text{ xor } (1\text{E3D323C} \lll 24) = 66\text{EF4F72}$

$X[19] = X[15] \text{ xor } T(X[16] \text{ xor } X[17] \text{ xor } X[18] \text{ xor } \text{rk}[15]) = 907127\text{FA} \text{ xor } T(8\text{B952B83} \text{ xor } \text{D42B7C59} \text{ xor } 2\text{FFC5831} \text{ xor } 104495\text{A1}) =$

$$= 907127FA \text{ xor } T(60069A4A) = F69E6888$$

T function:

1) S-box:

'C8' = '11'

'F8' = '79'

'69' = '22'

'DC' = 'B8'

S-box result: Tau = B8227911

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \\ \lll 24) = \\ = B8227911 \text{ xor (B8227911} \lll 2) \text{ xor (B8227911} \lll 10) \text{ xor (B8227911} \\ \lll 18) \text{ xor (B8227911} \lll 24) = 24B11947 \end{aligned}$$

$$\begin{aligned} X[20] &= X[16] \text{ xor } T(X[17] \text{ xor } X[18] \text{ xor } X[19] \text{ xor rk}[16]) = 8B952B83 \text{ xor} \\ T(D42B7C59 \text{ xor } 2FFC5831 \text{ xor } F69E6888 \text{ xor } D120B428) &= \\ &= 8B952B83 \text{ xor } T(DC69F8C8) = AF2432C4 \end{aligned}$$

T function:

1) S-box:

'DE' = 'B4'

'5D' = '56'

'F3' = 'EC'

'05' = 'E1'

S-box result: Tau = E1EC56B4

$$\begin{aligned} 2) L(\text{tau}): \text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \\ \lll 24) = \\ = E1EC56B4 \text{ xor (E1EC56B4} \lll 2) \text{ xor (E1EC56B4} \lll 10) \text{ xor} \\ (E1EC56B4 \lll 18) \text{ xor (E1EC56B4} \lll 24) = 3935B407 \end{aligned}$$

$$\begin{aligned}
X[21] &= X[17] \text{ xor } T(X[18] \text{ xor } X[19] \text{ xor } X[20] \text{ xor } rk[17]) = D42B7C59 \text{ xor} \\
&T(2FFC5831 \text{ xor } F69E6888 \text{ xor } AF2432C4 \text{ xor } 73B55FA3) = \\
&= D42B7C59 \text{ xor } T(05F35DDE) = ED1EC85E
\end{aligned}$$

T function:

1) S-box:

'74' = '9F'

'DB' = '12'

'23' = 'F4'

'78' = '4C'

S-box result: Tau = 4CF4129F

$$\begin{aligned}
2) L(\text{tau}): & \text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \\
&\lll 24) = \\
&= 4CF4129F \text{ xor } (4CF4129F \lll 2) \text{ xor } (4CF4129F \lll 10) \text{ xor } (4CF4129F \\
&\lll 18) \text{ xor } (4CF4129F \lll 24) = 7A5FE213
\end{aligned}$$

$$\begin{aligned}
X[22] &= X[18] \text{ xor } T(X[19] \text{ xor } X[20] \text{ xor } X[21] \text{ xor } rk[18]) = 2FFC5831 \text{ xor} \\
&T(F69E6888 \text{ xor } AF2432C4 \text{ xor } ED1EC85E \text{ xor } CC874966) = \\
&= 2FFC5831 \text{ xor } T(7823DB74) = 55A3BA22
\end{aligned}$$

T function:

1) S-box:

'81' = 'BF'

'04' = 'CC'

'BD' = '6C'

'85' = 'C7'

S-box result: Tau = C76CCCBF

$$\begin{aligned}
2) L(\text{tau}): & \text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \\
&\lll 24) = \\
&= C76CCCBF \text{ xor } (C76CCCBF \lll 2) \text{ xor } (C76CCCBF \lll 10) \text{ xor}
\end{aligned}$$

$$(C76CCCBF \lll 18) \text{ xor } (C76CCCBF \lll 24) = E4D57022$$

$$\begin{aligned} X[23] &= X[19] \text{ xor } T(X[20] \text{ xor } X[21] \text{ xor } X[22] \text{ xor } rk[19]) = F69E6888 \text{ xor} \\ T(AF2432C4 \text{ xor } ED1EC85E \text{ xor } 55A3BA22 \text{ xor } 92244439) &= \\ &= F69E6888 \text{ xor } T(85BD0481) = 124B18AA \end{aligned}$$

T function:

1) S-box:

'C9' = 'D9'

'0E' = '2C'

'68' = '25'

'42' = 'A7'

S-box result: Tau = A7252CD9

$$\begin{aligned} 2) L(\tau): \tau \text{ xor } (\tau \lll 2) \text{ xor } (\tau \lll 10) \text{ xor } (\tau \lll 18) \text{ xor } (\tau \lll 24) &= \\ &= A7252CD9 \text{ xor } (A7252CD9 \lll 2) \text{ xor } (A7252CD9 \lll 10) \text{ xor} \\ (A7252CD9 \lll 18) \text{ xor } (A7252CD9 \lll 24) &= C5C3409B \end{aligned}$$

$$\begin{aligned} X[24] &= X[20] \text{ xor } T(X[21] \text{ xor } X[22] \text{ xor } X[23] \text{ xor } rk[20]) = AF2432C4 \text{ xor} \\ T(ED1EC85E \text{ xor } 55A3BA22 \text{ xor } 124B18AA \text{ xor } E89E641F) &= \\ &= AF2432C4 \text{ xor } T(42680EC9) = 6AE7725F \end{aligned}$$

T function:

1) S-box:

'8D' = '61'

'D1' = 'C1'

'C5' = 'DD'

'B5' = 'FD'

S-box result: Tau = FDDDC161

$$2) L(\tau): \tau \text{ xor } (\tau \lll 2) \text{ xor } (\tau \lll 10) \text{ xor } (\tau \lll 18) \text{ xor } (\tau \lll 24) =$$

$$\begin{aligned}
& \lll 24) = \\
& = \text{FDDDC161} \text{ xor } (\text{FDDDC161} \lll 2) \text{ xor } (\text{FDDDC161} \lll 10) \text{ xor} \\
& (\text{FDDDC161} \lll 18) \text{ xor } (\text{FDDDC161} \lll 24) = 19\text{D569A7}
\end{aligned}$$

$$\begin{aligned}
X[25] &= X[21] \text{ xor } T(X[22] \text{ xor } X[23] \text{ xor } X[24] \text{ xor } \text{rk}[21]) = \text{ED1EC85E} \text{ xor} \\
& T(55\text{A3BA22} \text{ xor } 124\text{B18AA} \text{ xor } 6\text{AE7725F} \text{ xor } 98\text{CA015A}) = \\
& = \text{ED1EC85E} \text{ xor } T(\text{B5C5D18D}) = \text{F4CBA1F9}
\end{aligned}$$

T function:

1) S-box:

'6C' = '01'

'5B' = '4B'

'72' = '46'

'4B' = '19'

S-box result: Tau = 19464B01

$$\begin{aligned}
& 2) L(\text{tau}): \text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \\
& \lll 24) = \\
& = 19464\text{B01} \text{ xor } (19464\text{B01} \lll 2) \text{ xor } (19464\text{B01} \lll 10) \text{ xor } (19464\text{B01} \\
& \lll 18) \text{ xor } (19464\text{B01} \lll 24) = 486\text{E4032}
\end{aligned}$$

$$\begin{aligned}
X[26] &= X[22] \text{ xor } T(X[23] \text{ xor } X[24] \text{ xor } X[25] \text{ xor } \text{rk}[22]) = 55\text{A3BA22} \text{ xor} \\
& T(124\text{B18AA} \text{ xor } 6\text{AE7725F} \text{ xor } \text{F4CBA1F9} \text{ xor } \text{C7159060}) = \\
& = 55\text{A3BA22} \text{ xor } T(4\text{B725B6C}) = 1\text{DCDFA10}
\end{aligned}$$

T function:

1) S-box:

'98' = 'AD'

'D4' = 'A5'

'00' = 'D6'

'1A' = '13'

S-box result: $\text{Tau} = 13\text{D6A5AD}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$

$= 13\text{D6A5AD} \text{ xor } (13\text{D6A5AD} \lll 2) \text{ xor } (13\text{D6A5AD} \lll 10) \text{ xor } (13\text{D6A5AD} \lll 18) \text{ xor } (13\text{D6A5AD} \lll 24) = 3\text{DBD1EA9}$

$X[27] = X[23] \text{ xor } T(X[24] \text{ xor } X[25] \text{ xor } X[26] \text{ xor } \text{rk}[23]) = 124\text{B18AA} \text{ xor}$

$T(6\text{AE7725F} \text{ xor } \text{F4CBA1F9} \text{ xor } 1\text{DCDFA10} \text{ xor } 99\text{E1FD2E}) =$

$= 124\text{B18AA} \text{ xor } T(1\text{A00D498}) = 2\text{FF60603}$

T function:

1) S-box:

'E6' = '77'

'85' = 'C7'

'6B' = '3B'

'71' = '00'

S-box result: $\text{Tau} = 003\text{BC777}$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$

$= 003\text{BC777} \text{ xor } (003\text{BC777} \lll 2) \text{ xor } (003\text{BC777} \lll 10) \text{ xor } (003\text{BC777} \lll 18) \text{ xor } (003\text{BC777} \lll 24) = 85153\text{D83}$

$X[28] = X[24] \text{ xor } T(X[25] \text{ xor } X[26] \text{ xor } X[27] \text{ xor } \text{rk}[24]) = 6\text{AE7725F} \text{ xor}$

$T(\text{F4CBA1F9} \text{ xor } 1\text{DCDFA10} \text{ xor } 2\text{FF60603} \text{ xor } \text{B79BD80C}) =$

$= 6\text{AE7725F} \text{ xor } T(716\text{B85E6}) = \text{EFF24FDC}$

T function:

1) S-box:

'7F' = '9E'

'A6' = 'CA'

'E8' = '65'

'C0' = '8D'

S-box result: Tau = 8D65CA9E

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
 = 8D65CA9E xor (8D65CA9E <<< 2) xor (8D65CA9E <<< 10) xor
 (8D65CA9E <<< 18) xor (8D65CA9E <<< 24) = 9B2FCA8C

X[29] = X[25] xor T(X[26] xor X[27] xor X[28] xor rk[25]) = F4CBA1F9 xor
 T(1DCDFA10 xor 2FF60603 xor EFF24FDC xor 1D2115B0) =
 = F4CBA1F9 xor T(C0E8A67F) = 6FE46B75

T function:

1) S-box:

'41' = '07'

'A8' = 'C0'

'C2' = 'AF'

'A1' = 'F6'

S-box result: Tau = F6AFC007

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =
 = F6AFC007 xor (F6AFC007 <<< 2) xor (F6AFC007 <<< 10) xor
 (F6AFC007 <<< 18) xor (F6AFC007 <<< 24) = 94F9AABD

X[30] = X[26] xor T(X[27] xor X[28] xor X[29] xor rk[26]) = 1DCDFA10 xor
 T(2FF60603 xor EFF24FDC xor 6FE46B75 xor 0E228AEB) =
 = 1DCDFA10 xor T(A1C2A841) = 893450AD

T function:

1) S-box:

'85' = 'C7'

'78' = '4C'

'5A' = '0F'

'F8' = '79'

S-box result: Tau = 790F4CC7

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =

= 790F4CC7 xor (790F4CC7 <<< 2) xor (790F4CC7 <<< 10) xor (790F4CC7 <<< 18) xor (790F4CC7 <<< 24) = 5465894F

X[31] = X[27] xor T(X[28] xor X[29] xor X[30] xor rk[27]) = 2FF60603 xor

T(EFF24FDC xor 6FE46B75 xor 893450AD xor F1780C81) =

= 2FF60603 xor T(F85A7885) = 7B938F4C

T function:

1) S-box:

'C0' = '8D'

'82' = '8A'

'CE' = '5A'

'DF' = 'B0'

S-box result: Tau = B05A8A8D

2) L(tau): Tau xor (Tau <<< 2) xor (Tau <<< 10) xor (Tau <<< 18) xor (Tau <<< 24) =

= B05A8A8D xor (B05A8A8D <<< 2) xor (B05A8A8D <<< 10) xor (B05A8A8D <<< 18) xor (B05A8A8D <<< 24) = BC9C0D9A

X[32] = X[28] xor T(X[29] xor X[30] xor X[31] xor rk[28]) = EFF24FDC xor

T(6FE46B75 xor 893450AD xor 7B938F4C xor 428D3654) =

= EFF24FDC xor T(DFCE82C0) = 536E4246

T function:

1) S-box:

'31' = 'B3'

'A9' = '29'

'E0' = '89'

'C3' = '92'

S-box result: $\text{Tau} = 928929B3$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$
 $= 928929B3 \text{ xor } (928929B3 \lll 2) \text{ xor } (928929B3 \lll 10) \text{ xor } (928929B3 \lll 18) \text{ xor } (928929B3 \lll 24) = E957823A$

$X[33] = X[29] \text{ xor } T(X[30] \text{ xor } X[31] \text{ xor } X[32] \text{ xor } rk[29]) = 6FE46B75 \text{ xor } T(893450AD \text{ xor } 7B938F4C \text{ xor } 536E4246 \text{ xor } 62293496) =$
 $= 6FE46B75 \text{ xor } T(C3E0A931) = 86B3E94F$

T function:

1) S-box:

'A0' = '1D'

'56' = 'DA'

'81' = 'BF'

'AF' = '6F'

S-box result: $\text{Tau} = 6FBFDA1D$

2) L(tau): $\text{Tau} \text{ xor } (\text{Tau} \lll 2) \text{ xor } (\text{Tau} \lll 10) \text{ xor } (\text{Tau} \lll 18) \text{ xor } (\text{Tau} \lll 24) =$
 $= 6FBFDA1D \text{ xor } (6FBFDA1D \lll 2) \text{ xor } (6FBFDA1D \lll 10) \text{ xor } (6FBFDA1D \lll 18) \text{ xor } (6FBFDA1D \lll 24) = 5B32C6F3$

$X[34] = X[30] \text{ xor } T(X[31] \text{ xor } X[32] \text{ xor } X[33] \text{ xor } rk[30]) = 893450AD \text{ xor } T(7B938F4C \text{ xor } 536E4246 \text{ xor } 86B3E94F \text{ xor } 01CF72E5) =$

$$= 893450AD \text{ xor } T(AF8156A0) = D206965E$$

T function:

1) S-box:

'45' = '73'

'9D' = '8C'

'FF' = '48'

'96' = '1A'

S-box result: $\text{Tau} = 1A488C73$

2) L(tau): $\text{Tau xor (Tau} \lll 2) \text{ xor (Tau} \lll 10) \text{ xor (Tau} \lll 18) \text{ xor (Tau} \lll 24) =$

$$= 1A488C73 \text{ xor } (1A488C73 \lll 2) \text{ xor } (1A488C73 \lll 10) \text{ xor } (1A488C73 \lll 18) \text{ xor } (1A488C73 \lll 24) = 138D5078$$

~~~~X Result:

$$\begin{aligned} X[35] &= X[31] \text{ xor } T(X[32] \text{ xor } X[33] \text{ xor } X[34] \text{ xor } rk[31]) = 7B938F4C \text{ xor} \\ T(536E4246 \text{ xor } 86B3E94F \text{ xor } D206965E \text{ xor } 9124A012) &= \\ &= 7B938F4C \text{ xor } T(96FF9D45) = 681EDF34 \end{aligned}$$