Міністерство освіти і науки України

Одеський національний політехнічний університет

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Радуш Володимир Вячеславович,

студент групи РЗ-171

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

Розробка та імплементація нелінійних перетворень сучасних шифрів, що задовольняють суворому лавинному критерію

Спеціальність:

125 Кібербезпека

Спеціалізація, освітня програма:

Кібербезпека

Керівник:

Соколов Артем Вікторович,

к.т.н., доцент

Одеса – 2021

Міністерство освіти і науки України

Одеський національний політехнічний університет

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Рівень вищої освіти перший (бакалаврський)

Спеціальність 125 − Кібербезпека

Освітня програма – Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри КБПЗ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

д.т.н.,проф. А.А.Кобозєва

\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_202\_р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

*Радуш Володимиру Вячеславовичу*

1.Тема роботи: *Розробка та імплементація нелінійних перетворень сучасних шифрів, що задовольняють суворому лавинному критерію,*

керівник роботи *Соколов Артем Вікторович, к. ф.-м. н., доцент,*

затверджені наказом ректора ОНПУ від „\_\_\_\_”\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р. №\_\_\_\_\_\_\_ .

2.Зміст роботи*: аналіз проблемної області, постановка задачі, аналіз принципів побудови систем менеджменту інформаційної безпеки, розробка проекту системи менеджменту інформаційної безпеки банку, експериментальне дослідження системи менеджменту інформаційної безпеки, охорона праці.*

3. Перелік ілюстративного матеріалу: *організаційна структура банку, схема мережевої структури банку****,***  *модель PDCA, схема роботи системи, блок-схема алгоритму, слайди презентації*.

(Зворотний бік бланку завдання)

4. Консультанти розділів роботи

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|  |  | Завдання видав | Завдання рийняв |
|  |  |

5. Дата видачі завдання “\_\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання | Примітка |
| *1* | *Аналіз джерел з теми випускної кваліфікаційної роботи* | *15.11.2019* | *виконано* |
| *2* | *Обґрунтування вибору рішення.*  *Збір даних* | *15-12-2019* | *виконано* |
| *3* | *Аналіз основних аспектів побудови СМІБ* | *11-01-2020* | *виконано* |
| *4* | *Аналіз умов функціонування інформаційних систем банку* | *20-02-2020* | *виконано* |
| *5* | *Розроблення проекту побудови СМІБ банку* | *30-03-2020* | *виконано* |
| *6* | *Підготовка тексту роботи* | *11-05-2020* | *виконано* |
| *7* | *Підготовка презентації та доповіді* | *15-05-2020* | *виконано* |
| *8* | *Попередній захист* | *20-05-2020* | *виконано* |
| *9* | *Нормоконтроль, рецензування* | *20-05-2020* | *виконано* |
| *10* | *Занесення роботи в електронний архів* | *28-05-2020* | *виконано* |
| *11* | *Допуск до захисту у завідувача кафедри* | *01-06-2020* | *виконано* |

|  |  |  |
| --- | --- | --- |
| **Здобувач вищої освіти** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | *Радуш В.В* |

**Керівник роботи** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Соколов А.В.*

**ЗАВДАННЯ**

на розробку розділу “Охорона праці”

*Радуш Володимиру Вячеславовичу, група РЗ-171*

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Тема роботи *Робоче місце криптографа*

|  |  |
| --- | --- |
| Зміст розділу: |  |

1. Аналіз умов праці і вибір основних заходів виробничої безпеки.
2. Аналіз пожежної безпеки. Вибір заходів та засобів пожежної безпеки.
3. Індивідуальне завдання.

Керівник роботи Консультант з охорони праці

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р. «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.

АНОТАЦІЯ

Кваліфікаційна робота на тему “Система менеджменту інформаційної безпеки банку” на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 − Кібербезпека, спеціалізація, освітня програма: Управління кібербезпекою, містить 7 рисунків, 4 таблиці, 2 додатки, 25 літературних джерел за переліком посилань. Робота виконана на 60 сторінках загального тексту і 50 сторінках основного тексту.

Метою роботи є підвищення рівня безпеки бізнес-процесів банку шляхом розробки та впровадження системи менеджменту інформаційної безпеки.

У роботі проведено аналіз стандарту СОУ Н НБУ 65.1 1.0:2010, що дозволило на основі моделі PDCA визначити етапи побудови системи менеджменту інформаційної безпеки.

У результаті виконання кваліфікаційної роботи розроблено проект системи менеджменту інформаційної безпеки банку, що дозволило зробити аналіз існуючих ризиків ІБ та впровадити систему управління ризиками ІБ. Проведено аналіз захищеності бізнес-процесу банку до та після впровадження системи. Зроблено висновок, що за рахунок впровадження системи менеджменту інформаційної безпеки середній рівень ризику ІБ зменшиться на 50%, захищеність операцій з активами зросте на 3.5%.

Результати даної роботи можуть бути використані при побудові аналогічної системи менеджменту інформаційної безпеки.

ІНФОРМАЦІЯ, ІНФОРМАЦІЙНА БЕЗПЕКА, БАНК, СИСТЕМА МЕНЕДЖМЕНТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ, РИЗИК, БІЗНЕС-ПРОЦЕС.

ANNOTATION

Кваліфікаційна робота на тему “Система менеджменту інформаційної безпеки банку” на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 − Кібербезпека, спеціалізація, освітня програма: Управління кібербезпекою, містить 7 рисунків, 4 таблиці, 2 додатки, 25 літературних джерел за переліком посилань. Робота виконана на 60 сторінках загального тексту і 50 сторінках основного тексту.

Метою роботи є підвищення рівня безпеки бізнес-процесів банку шляхом розробки та впровадження системи менеджменту інформаційної безпеки.

У роботі проведено аналіз стандарту СОУ Н НБУ 65.1 1.0:2010, що дозволило на основі моделі PDCA визначити етапи побудови системи менеджменту інформаційної безпеки.

У результаті виконання кваліфікаційної роботи розроблено проект системи менеджменту інформаційної безпеки банку, що дозволило зробити аналіз існуючих ризиків ІБ та впровадити систему управління ризиками ІБ. Проведено аналіз захищеності бізнес-процесу банку до та після впровадження системи. Зроблено висновок, що за рахунок впровадження системи менеджменту інформаційної безпеки середній рівень ризику ІБ зменшиться на 50%, захищеність операцій з активами зросте на 3.5%.

Результати даної роботи можуть бути використані при побудові аналогічної системи менеджменту інформаційної безпеки.

ІНФОРМАЦІЯ, ІНФОРМАЦІЙНА БЕЗПЕКА, БАНК, СИСТЕМА МЕНЕДЖМЕНТУ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ, РИЗИК, БІЗНЕС-ПРОЦЕС.

ЗМІСТ

[ВСТУП 8](#_Toc69740518)

[1 СУВОРИЙ ЛАВИННИЙ КРИТЕРІЙ ТА КОНСТРУКЦІЯ НІБЕРГ. МЕТОДИ СИНТЕЗУ ЯКІСНИХ S-БЛОКІВ 10](#_Toc69740519)

[1.1 Суворий лавинний критерій: основні визначення та принцип оцінки 10](#_Toc69740520)

[1.2 Шифр AES 11](#_Toc69740521)

[1.3 Конструкція Ніберг 13](#_Toc69740522)

[1.4 Схема Кіма 15](#_Toc69740523)

[2 СИНТЕЗ S-БЛОКІВ, ЩО ЗАДОВОЛЬНЯЮТЬ ВИМОГАМ СЛК КОМПОНЕНТНИХ 4-ФУНКЦІЙ ТА КРИТЕРІЮ МАКСИМАЛЬНОГО ЛАВИННОГО ЕФЕКТУ БУЛЕВИХ ФУНКЦІЙ 17](#_Toc69740524)

[2.1 Дослідження конструкціі Ніберг на відповідність суворому лавинному критерію (СЛК) 17](#_Toc69740525)

[2.2 Метод створення якісних S-блоків 20](#_Toc69740526)

[3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ СИНТЕЗУ ЯКІСНИХ S-БЛОКІВ 25](#_Toc69740527)

[3.1 Середовище програмування 25](#_Toc69740528)

[3.2 Модуль створення якісних S-блоків 26](#_Toc69740529)

[3.3 Графічний модуль програми 28](#_Toc69740530)

[3.4 Практичне використання 32](#_Toc69740531)

[ПЕРЕЛІК ПОСИЛАНЬ 34](#_Toc69740532)

[Додаток А. Q-послідовність 37](#_Toc69740533)

[Додаток Б. Відхилення компонентних функцій S-блоків конструкції Ніберг від СЛК 38](#_Toc69740534)

[Додаток В. Створений якісний S-блок 39](#_Toc69740535)

# ВСТУП

Сьогодні важко навіть уявити комплексну систему захисту інформації, яка обходилася б без застосування в ній блокових симетричних шифрів. При цьому одним з найпоширеніших блокових симетричних криптографічних алгоритмів є криптоалгоритм AES, який використовується в багатьох системах захисту інформації. Криптоалгоритм AES характеризується високим рівнем криптостійкості і простотою програмної і апаратної реалізації. Проте, постійний розвиток методів криптоаналіза вимагає уважного вивчення його криптографічних властивостей, а також подальшого вдосконалення його структури і криптографічних примітивів, з яких він складається.

Найважливішим криптографічним примітивом, що входить до складу будь-якого блочного симетричного криптоалгоритма є S-блок, який багато в чому визначає загальний рівень дифузії і конфузії [1] криптографічного алгоритму, а також простоту його технічної реалізації. В даний час відомо багато конструктивних методів побудови високоякісних S-блоків, серед яких особливе місце займає конструкція Ніберг, на основі якої побудований S-блок криптоалгоритма AES [2].

Сьогодні якість будь-якого криптографічного S-блоку оцінюється за допомогою наступних критеріїв [3, 4]:

1. Висока відстань нелінійності.

2. Висока алгебраїчна степінь нелінійності.

3. Відповідність суворому лавинному критерію (СЛК).

4. Статистична незалежність виходу S-блоку підстановки від його входу.

Всі перераховані критерії засновані на уявленні S-блоку у вигляді набору булевих функцій і зараз відомо багато робіт, присвячених оцінці даних критеріїв для S-блоку криптоалгоритма AES [5, 6].

Однак, атака може здійснюватися з використанням інших математичних конструкцій, які використовуються для опису криптоалгоритма, наприклад, за допомогою апарату функцій багатозначної логіки [7]. Дана обставина вимагає більш детального вивчення криптографічної якості функцій багатозначної логіки, що входять до складу S-блоку криптоалгоритма AES.

*Метою* цієї роботи є дослідження S-блоків конструкції Ніберг на відповідність СЛК при їх уявленні q-функціями, а також створення рекурентного методу синтезу S-блоків з добрими лавинними характеристиками для криптоалгоритму AES.

Задачі дослідження:

1. дослідити представлення S-блоків AES у вигляді функцій багатозначної логіки на їх відповідність суворому лавинному критерію. Знайти S-блоки конструкції Ніберг, які характеризуються найменшим відхиленням від вимог суворого лавинного критерію;
2. розробити рекурентний метод синтезу S-блоків, що відповідають як суворому лавинному критерію компонентних 4-функцій, так і критерію максимального лавинного ефекту компонентних булевих функцій.

Предмет роботи − створення методу синтезу S-блоків.

Об’єкт дослідження − відповідність S-блоків суворому лавинному критерію.

Робота складається з чотирьох розділів. У першому розділі наведено характеристику суворого лавинного критерію, відповідність S-блоків конструкції Ніберг суворому лавинному критерію у сенсі двійкової, четвіркової та шістнадцяткової логіки. У другому розділі представлено розробку метода синтезу якісних S-блоків довжини 256 з добрими лавинними властивостями у сенсі двійкової логіки, а також з ідеальними лавинними властивосятми у сенсі четвіркової логіки. Третій розділ – реалізація програмного продукту, у якому описано середовище програмування, безпосередньо сам програмний продукт, що працює на основі методу, запропонованого у другому розділі, що також містить дружній до користувача інтерфейс. Четвертий розділ містить у собі оцінку та аналіз умов праці та пожежної безпеки у виробничому середовищі, вибір основних заходів пожежної безпеки та заходів щодо охорони праці на робочому місці користувача персонального комп’ютера.

# СУВОРИЙ ЛАВИННИЙ КРИТЕРІЙ ТА КОНСТРУКЦІЯ НІБЕРГ. МЕТОДИ СИНТЕЗУ ЯКІСНИХ S-БЛОКІВ

* 1. Суворий лавинний критерій: основні визначення та принцип оцінки

Лавинний ефект (англ. Avalanche effect) – одне з основних криптографічних понять, що використовується стосовно блокових шифрів, а також хеш-функцій. Суть цього поняття полягає у тому, що при зміні малої кількості бітів у вхідній послідовності або в ключі відбуваються суттєві («лавинні») зміни у вихідній послідовності, відомій також як шифротекст. Інакше кажучи, це залежність вихідних бітів від вихідних[8].

Стає зрозуміло, що суворий лавинний ефект якраз і забезпечує лавинні властивості блокових шифрів чи хеш-функцій. Варто зазначити, що S-блоки довжини *N*=256 можуть бути представлені як за допомогою булевих функцій (*q*=2), так і за допомогою функцій четвіркової (*q*=4) або функцій шістнадцяткової логіки (*q*=16).

Відповідно, якість S-блоку, що використовується чи потенційно може використовуватись, визначається кожним набором цих функцій. Можна дійти висновку, що для дослідження S-блоку з точки зору функцій багатозначної логіки необхідно використати більш широкий математичний апарат.

Для повного розуміння принципів функціонування СЛК необхідно надати визначення основних понять. Для початку розглянемо *q*-функцію *k* змінних *f(x)*, а також вектор *.*

**Визначення 1 [9].** Функцією *q*-значної логіки *k* змінних називається відображення 

**Визначення 2 [9].** Вагою  *q*-значного вектору назвемо кількість його ненульових компонент.

**Визначення 3 [9].** Похідною функції *f* у напрямку вектору 𝑢 назвемо функцію

 , (1.1)

де  означає додавання по модулю  *q*.

**Визначення 4 [9].** Функція *q*-значної логіки *f(x)* відповідає критерію поширення щодо вектора – KП(*u*), якщо її похідна по напрямку *u* є збалансованою функцією, тобто значення  приймаються з рівними ймовірностями:  для усіх . Інакше кажучи, , де кількість наборів значень змінних, на які похідна приймає значення *i*.

**Визначення 5 [9].** Функція *q*-значної логіки *f(x)* відповідає критерію поширення степеню *k* – КП(*k*), якщо вона відповідає критерію поширення КП(*u*) щодо всіх векторів *u* ваги .

**Визначення 6 [9].** Функція *q*-значної логіки *f(x)* задовольняє суворому лавинному критерію (СЛК), якщо вона відповідає критерію поширення степеню 1 – КП (1).

Очевидно, що наведені визначення є універсальними стосовно будь-якого представлення S-блоку, включаючи двійкову логіку[10]. Однак, визначення СЛК є досить жорсткою вимогою, тому досить часто використовується визначення критерію максимального лавинного ефекту.

**Визначення 7 [11].** Булева функція відповідаю критерію максимального лавинного критерію (МЛК), якщо усі її похідні по всім напрямкам одиничної ваги  мають вагу , що дорівнює як мінімум половині її довжини , тобто:

|  |  |
| --- | --- |
|  | (1.2) |

* 1. Шифр AES

Одним з найвідоміших та, відповідно, найпопулярніших блокових симетричних шифрів є шифр AES (Advanced Encryption Standart). Розробку шифру було розпочато у 1997 році, як альтернативу застарілому DES (Data encryption standart) та його модифікації 3-DES. Це було обумовлено тим, що DES став вразливим для атак методом грубої сили і відповідно став непридатним для якісного шифрування.

Національний інститут стандартів та технологій (NIST) отримав доручення уряду США на створення нового стандарту, що мав би не втрачати актуальності щонайменше 20 років, а тому у 1997 році оголошується конкурс AES. До фіналу надійшли 5 фіналістів: MARS, RC6, Rijndael, Serpent и Twofish. Переможцем було оголошено Rijndael. NIST почав стандартизацію нового шифру, а вже у 2001 році було випущено специфікацію AES[12].

На даний момент AES активно використовується у всьому світі, так як він є вбудованим в програмне та апаратне забезпечення і став визнаним стандартом шифрування. Варто зазначити, що AES має кілька версій, а саме AES-128, AES-192, AES-256, де принцип роботи фактично не змінюється, а змінюється довжина ключа. Більш того, Агенство національної безпеки США використовує AES-128 для зберігання інформації рівня «таємно», а 192 чи 256 для інформації рівня «цілком таємно» [13, 14].

Тепер коротко розглянемо принцип роботи шифру AES. Сам алгоритм містить у собі чотири види трансформації, а саме:

* + - 1. AddRoundKey() – додавання матеріалу секретного ключа за допомогою операції XOR;

2. SubBytes() – перетворенням у даному випадку виступає заміна кожного вхідного байту на відповідний до нього з таблиці констант, відомою також як S-блок.

3. ShiftRows() – одне з найпростіших перетворень. Відбувається циклічний зсув вліво на 1 елемент для першого рядка двомірного масиву, а для кожного наступного рядка зсув вліво збільшується на одиницю. Зворотна операція – також циклічний зсув, але вже у протилежному направленні.

4. MixColumns() – математичне перетворення, що перемішує дані у стовпці.

Роботу алгоритму представлено у вигляді схеми (рисунок 1.1)[15].

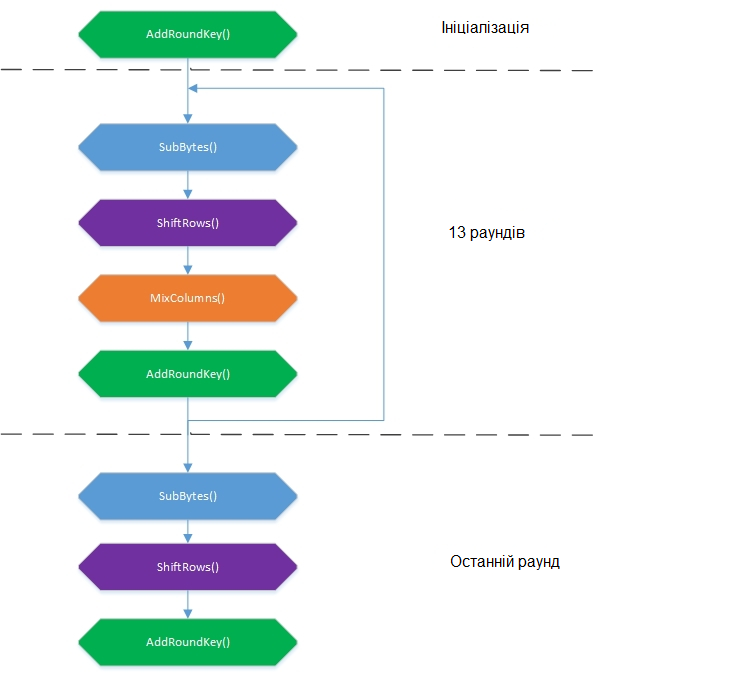


Рисунок 1.1 – Схема роботи шифру AES

1.3 Конструкція Ніберг

Як вже зазначалося вище, жоден криптоалгоритм не може вважатись надійним, якщо в його основі лежать ненадійні криптографічні примітиви, зокрема S-блоки та P-блоки. В основі AES закладено лише один S-блок, що відноситься до множини блоків конструкції Ніберг.

Конструкція Ніберг була імплементована до криптоалгоритму AES завдяки тому, що вона має високу відповідність основним критеріям криптографічної якості. Більш того, вона має можливість різної технічної реалізації, такої як табличним способом, а також за допомогою обчислень над розширеним полем Галуа. Конструкція Ніберг – це відображення, яке задається мультиплікативно зворотними елементами полів Галуа 

, (1.3)

скомбіноване разом з афінним перетворенням

, (1.4)

де  – незвідний над *GF*(2) поліном степеню *k;*

 – за визначенням; *A* – невироджена матриця афінного перетворення;

*а* – вектор зсуву;

*p* = 2 – характеристика розширеного поля Галуа;

*a, b, x, y* – елементи розширеного поля Галуа *GF*(2*k*), вони трактуються як десяткові числа, двійкові вектори, а також поліноми степеню ;

В роботі [6] встановлено, що структура S-блока конструкції Ніберг залежить від виду незвідного полінома, при цьому кількість таких поліномів [16] ступені *k* визначається за формулою

, (1.5)

яка, також визначає потужність класу S-блоків конструкції Ніберг, де:

|Wk| – кількість незвідних поліномів;

d – дільники числа k;

μ – функція Мебіуса.

Конструктори різних криптоалгоритмів обирають різні поліноми для конструювання S-блоків. Наприклад, в криптоалгоритмі AES використовується поліном . При цьому вибір криптографічних конструкцій відбувається тільки на основі аналізу криптографічної якості їх булевих функцій. Дана обставина обумовлює необхідність більш обґрунтованого вибору використовуваного для побудови S-блоку незвідного полінома з урахуванням криптографічних властивостей його функцій багатозначної логіки[17].

1.4 Схема Кіма

На даний момент існує лише один відомий в криптографії метод синтезу S-блоків, що відповідають СЛК – Схема Кіма. Вона дає можливість синтезувати S-блоки будь-якої довжини, що відповідають усім можливим представленням ступенів двійки *k*, тобто 8, 16, 32, 64 і так до нескінченності. Варто зазначити, що усю множину S-блоків, що відповідають СЛК знайти досить легко, використовуючи повний перебір.

Розглянемо схему більш детально. Нехай в нас є деякий S-блок довжини 8, який ми позначимо *Sk ,* що відповідає СЛК булевих функцій:

|  |  |
| --- | --- |
|  | (1.6) |

**Крок 1**. Розкладемо (1.6) на компонентні булеві функції (Таблиця 1.1)

Таблиця 1.1 – Розбиття S-блоку на булеві функції

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **4** | **7** | **2** | **6** | **1** | **5** | **0** | **3** |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |

**Крок 2.** Обираємо найстаршу компоненту функцію *Fm*. Згідно Таблиці 1.1 це 10001101 та подвоюємо її за наступним правилом горизонтальної конкатенації:

|  |  |
| --- | --- |
| , | (1.7) |

де – вектор довжини  , що містить 1 на позиції ;

| – символ горизонтальної конкатенації;

 – похідна функції *Fm* по направленню ;

Тоді при  = {0 0 1} отримаємо:

|  |  |
| --- | --- |
|  | (1.8) |

**Крок 3.** Далі вже подвоюємо довжину S-блоку за наступним правилом:

|  |  |
| --- | --- |
|  | (1.9) |

Тепер, знову взявши ={0 0 1}, отримуємо новий S-блок, що не є бієктивним, тобто містить однакові елементи:

|  |  |
| --- | --- |
|  | (1.10) |

**Крок 4.** Будуємо новий бієктивний S-блок , що відповідає СЛК за правилом:

|  |  |
| --- | --- |
| , | (1.11) |

звідки отримуємо:

|  |  |
| --- | --- |
|  | (1.12) |

При знаходженні ваги похідних компонентних булевих функцій  для (1.12) отримуємо, що даний S-блок відповідає СЛК (Таблиця 1.2) [18,19].

Таблиця 1.2 – Ваги похідних функцій

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0001 | 8 | 8 | 8 | 8 |
| 0010 | 8 | 8 | 8 | 8 |
| 0100 | 8 | 8 | 8 | 8 |
| 1000 | 8 | 8 | 8 | 8 |

Незважаючи на те, що схема Кіма дозволяє синтезувати S-блоки будь-яких довжини, що відповідають СЛК булевої логіки, на жаль, вони завжди відповідатимуть лише СЛК булевої логіки. Такі обмеження даного математичного апарату є досить суттєвими, так як сучасні тенденції розвитку криптографії а також великі обсяги даних диктують необхідність в нових криптографічних примітивах, що відповідали хоча б СЛК булевої та 4-логіки[20].

Варто також зауважити, що світовий стандарт шифрування AES був розроблений у 90-ті роки, ураховуючи лише можливості булевої логіки. Тоді найбільш вірогідно, що і блоки конструкції Ніберг не відповідають СЛК 4-логіки, тим самим знижуючи можливості криптоалгоритму.

# 2 СИНТЕЗ S-БЛОКІВ, ЩО ЗАДОВОЛЬНЯЮТЬ ВИМОГАМ СЛК КОМПОНЕНТНИХ 4-ФУНКЦІЙ ТА КРИТЕРІЮ МАКСИМАЛЬНОГО ЛАВИННОГО ЕФЕКТУ БУЛЕВИХ ФУНКЦІЙ

* 1. Дослідження конструкціі Ніберг на відповідність суворому лавинному критерію (СЛК)

Розглянемо конкретний приклад. Нехай заданий S-блок конструкції Ніберг на основі незвідного полінома *f(z)* = 28310 = *z*8 + *z*4 + *z*3 + *z* + 1 , який використовується в шифрі AES в вигляді Q-послідовності (Додаток А).

Як приклад розглянемо процес знаходження лавинних характеристик S-блока (Додаток А), представленого за допомогою чотирьох компонентних 4-функцій F*fouri*, *i*=1,2,…,4, з яких перша має вигляд:

|  |  |
| --- | --- |
|  | (2.1) |

Використовуючи **Визначення 3**, знайдемо, наприклад, похідну 4-функції (2.1) в напрямку вектора *u* = {0,0,0,1}, яка має вигляд

|  |  |
| --- | --- |
|  | (2.2) |

Відповідно до вимог **Визначення 4**, для того, щоб S-блок (2.3) задовольняв критерію поширення щодо вектора *u* = {0,0,0,1}, необхідно, щоб кількості символів "0", "1", "2" і "3" були рівні один одному, тобто *K*0*=K*1*=K*2*=K*3*=N/4=64* Проте, дана вимога не виконується для похідної (2.3)

|  |  |
| --- | --- |
|  | (2.3) |

Знайдемо величину відхилення похідної (2.2) від відповідності вимогам **Визначення 4**

|  |  |
| --- | --- |
|  | (2.4) |

Аналогічним чином, ми можемо знайти відхилення похідної 4-функції (2.4) від відповідності суворому лавинному критерію за кожним напрямом

|  |  |
| --- | --- |
| . | (2.5) |

При цьому ясно, що загальна якість компонентної 4-функції визначається найбільшим значенням серед відхилень (2.5), яке в нашому випадку дорівнює , при цьому загальна якість S-блоку буде визначатися максимумом серед максимальних відхилень його компонентних функцій, в нашому випадку, .

Ще одним способом охарактеризувати інтегральне відхилення похідної *DjFfour*1 (2.5) від відповідності умови критерію поширення (**Визначення 4**) є використання величини

|  |  |
| --- | --- |
|  | (2.6) |

Аналогічним чином можливо знайти суму відхилень за всіма напрямками одиничної ваги компонентної 4-функції (2.1) S-блоку (Додаток А), як того вимагає **Визначення 6** суворого лавинного критерію:

|  |  |
| --- | --- |
| . | (2.7) |

Можливо також порахувати суму відхилень від СЛК для кожної з 4-х компонентних 4-функцій S-блоку

|  |  |
| --- | --- |
|  | (2.8) |

У додатку Б представлені дані по лавинних властивостям S-блоків конструкції Ніберг, побудованих на основі повної множини незвідних поліномів степеню , представлених у вигляді булевих функцій, 4-функцій, а також 16-функцій.

Зрозуміло, що високою якістю криптографічної конструкції є мінімізація величини , а також величини . Аналіз даних додатку А показує, що компонентні булеві функції S-блоку конструкції Ніберг, який побудований на основі полінома , характеризуються найменшим відхиленням від вимог суворого лавинного критерію. У той же час, найменшим відхиленням від вимог суворого лавинного критерію характеризуються 4-функції S-блоку конструкції Ніберг, який побудований на основі полінома . При цьому також найменшим відхиленням від вимог суворого лавинного критерію характеризуються компонентні 16-функції S-блоку .

## 2.2 Метод створення якісних S-блоків

Проведені дослідження показують, що перспективною є розробка рекурентного методу побудови S-блоків з хорошими лавинними характеристиками компонентних булевих і четвіркових функцій, який представлено у вигляді покрокового алгоритму.

*Крок 1.* В якості вихідного матеріалу для розробленого методу використовується безліч S-блоків довжини , що відповідають СЛК 4-функцій, які побудовані в [21]. Потужність даної множини становить бієктивних S-блоків.

Як приклад будемо розглядати S-блок:

|  |  |
| --- | --- |
|  | (2.9) |

*Крок 2.* Задається функція , яка представляє собою старшу компонентну 4-функцію в розкладанні S-блоку на 4-функції.

У випадку S-блоку (2.9) дана компонентна 4-функція має вигляд

|  |  |
| --- | --- |
|  | (2.10) |

*Крок 3.* Формується безліч з 4-х перестановок у відповідності з наступним правилом

|  |  |
| --- | --- |
| , | (2.11) |

де  – один з векторів довжини  с 1 на одній зі своїх позицій;

вектор  пробігає четвіркові представлення чисел від 0 до ;

 – додавання по модулю 4;

 – символ поелементного множення четвіркового представлення числа  на значення .

У випадку нашого прикладу , візьмемо значення , відповідно множина перестановок буде мати наступний вигляд:

 (2.12)

*Крок 4.* Отримуємо функцію  довжини , за наступним правилом

, (2.13)

де  – множина констант, що повинна містити всі значення з множини . За замовчуванням візьмемо .

У нашому прикладі отримуємо, що функція  прийме наступний вигляд

. (2.14)

*Крок 5.* Збільшуємо довжину S-блоку до значення  використовуючи наступну конструкцію

. (2.15)

*Крок 6.* Будуємо новий бієктивний -блок довжини , що відповідає суровому лавинному критерію компонентних 4-функцій за наступним правилом

. (2.16)

У нашому випадку отримуємо наступний S-блок довжини 

 (2.17)

Отриманий S-блок (2.17) згідно до умов **Визначення 6** має наступнувагову матрицю похідних компонентних булевих функцій (Таблиця 2.1)

Таблиця 2.1 – Вага синтезованого S-блоку

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *ej* | *wt(D1,k)* | *wt(D2,k)* | *wt(D3,k)* | *wt(D4,k)* | *wt(D5,k)* | *wt(D6,k)* |
| 000001 | 32 | 40 | 32 | 24 | 32 | 40 |
| 000010 | 32 | 32 | 32 | 32 | 32 | 32 |
| 000100 | 32 | 32 | 32 | 32 | 32 | 32 |
| 001000 | 32 | 32 | 32 | 32 | 32 | 32 |
| 010000 | 32 | 40 | 32 | 24 | 32 | 24 |
| 100000 | 32 | 32 | 32 | 32 | 32 | 32 |

тобто не відповідає СЛК булевих функцій

Повторне використання розробленого методу до S-блоку (2.17) довжини  дозволяє отримати на його основі новий S-блок довжини 

 (2.18)

який теж відповідає СЛК компонентних 4-функцій, і характеризується наступною ваговою матрицею похідних компонентних булевих функцій (Таблиця 2.2)

Таблиця 2.2 – Значення синтезованого S-блоку

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *ej* | *wt(D1,k)* | *wt(D2,k)* | *wt(D3,k)* | *wt(D4,k)* | *wt(D5,k)* | *wt(D6,k)* | *wt(D7,k)* | *wt(D8,k)* |
| 00000001 | 128 | 160 | 128 | 96 | 128 | 160 | 128 | 128 |
| 00000010 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 00000100 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 00001000 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 00010000 | 128 | 160 | 128 | 96 | 128 | 128 | 128 | 96 |

Продовження таблиці 2.2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *ej* | *wt(D1,k)* | *wt(D2,k)* | *wt(D3,k)* | *wt(D4,k)* | *wt(D5,k)* | *wt(D6,k)* | *wt(D7,k)* | *wt(D8,k)* |
| 00100000 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 01000000 | 128 | 160 | 128 | 96 | 128 | 128 | 128 | 128 |
| 10000000 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |

тобто не відповідає СЛК булевих компонентних функцій.

Проведені експерименти показують, що на основі повної множини S-блоків довжини , шляхом використання розробленого методу можуть бути отримані S-блоки практично цінної довжини , що відповідають одночасно СЛК компонентних 4-функций та критерію максимального лавинного критерію булевих функцій.

Варто зазначити, що кількість S-блоків сильно залежить від значення обраних параметрів , а також від обраного вектору направлення . Наприклад, нехай задані наступні параметри: довжина S-блоку , параметри  на першій та другій ітерації використання методу, а також значення  на першій ітерації використання методу і  на другій ітерації використання методу.

Тоді отримуємо, що з множини  S-блоків довжини , отриманих на основі множини S-блоків довжини  [21]  S-блоків одночасно відповідають СЛК компонентних 4-функцій точно і критерію максимального лавинного ефекту компонентних булевих функцій.

У якості прикладу приведемо один з даних S-блоків довжини , компонентні 4-функції якого відповідають СЛК, побудованих на основі S-блоку  (додаток В) та його вагову матрицю похідних булевих функцій (Таблиця 2.3)

Таблиця 2.3 – Вага синтезованого S-блоку

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *ej* | *wt(D1,k)* | *wt(D2,k)* | *wt(D3,k)* | *wt(D4,k)* | *wt(D5,k)* | *wt(D6,k)* | *wt(D7,k)* | *wt(D8,k)* |
| 00000001 | 128 | 160 | 128 | 160 | 128 | 128 | 128 | 160 |
| 00000010 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |

Продовження таблиці 2.3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *ej* | *wt(D1,k)* | *wt(D2,k)* | *wt(D3,k)* | *wt(D4,k)* | *wt(D5,k)* | *wt(D6,k)* | *wt(D7,k)* | *wt(D8,k)* |
| 00000100 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 00001000 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 00010000 | 128 | 160 | 128 | 160 | 128 | 128 | 128 | 160 |
| 00100000 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |
| 01000000 | 128 | 160 | 128 | 160 | 128 | 128 | 128 | 128 |
| 10000000 | 128 | 128 | 128 | 128 | 128 | 128 | 128 | 128 |

яка підтверджує, що вказаний S-блок (додаток В) дійсно відповідає критерію максимального лавинного эффекту компонентних булевих функцій.

Згідно до отриманих результатів можна діти до висновку, що отримані S-блоки теоретично спроможні покращити будь-який блоковий симетричний шифр, що базується саме на використанні S-блоків. Більш того відповідність блоку максимальному та суворому лавинному критерію покращують криптографічну стійкість, а факт того, що 4-функції є поєднанням двох булевих функцій теоретично має підвищити швидкість роботи криптоалгоритму.

1. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ СИНТЕЗУ ЯКІСНИХ S-БЛОКІВ
   1. Середовище програмування

Для реалізації програмної частини проекту було використано мову програмування Python 3.8.6. Python було обрано виходячи з наступних умов:

1. зручність написання коду – Python вважається однією з найпростіших мов програмування, а парадигма того, що все є об’єктами значно полегшує написання алгоритмів.
2. велика кількість вбудованих бібліотек – Python має досить об’ємну базу різноманітних вбудованих бібліотек, що направлені на різні задачі. У тому числі має великий математичний апарат, що надає змогу виконувати різні математичні операції.
3. модульний підхід – дана мова програмування дозволяє розподілити код на різноманітні модулі, тобто винесення частин коду, що відповідають за різні сфери роботи програми у окремі файли, а при необхідності використвувати їх функціональні можливості у роботі головного модуля програми.
4. графічний інтерфейс – Python має декілька різноманітних інструментів для створення графічного інтерфейсу. Зокрема, у даній роботі було використано засоби відносно нової бібліотеки PySimpleGUI та Tkinter.
5. високі вимоги до написання коду – дане середовище потребує додержання відступів при написанні коду, що фактично примушує спеціаліста писати код естетично і надає змогу тим, хто захоче надалі ознайомитися з цим кодом більш легко орієнтуватися і, відповідно, швидше освоювати написаний функціонал.
6. кросплатформеність – дане середовище програмування підтримує різні платформи, зокрема і найпопулярніші, такі як Windows, Linux, MacOS, Android. Даний факт є дуже важливим, оскільки дозволяє використовувати усі розроблені рішення незалежно від платформи користування.
7. широке застосування провідними ІТ-компаніями світу, що дає можливість імплемнтувати створені розробки до програмного середовища більшості ІТ-компаній.

Для зручності використання програми її основні функціональні блоки було розподілено на декілька модулів:

1. base2.py – модуль програми, що відповідає за основні обчислення, що проводяться у двійкові системі у рамках проекту, такі як розбиття блоку на булеві функції, створення зворотного блок тощо.
2. base4.py – модуль програми, що відповідає за основні обчислення, що проводяться у четвірковій системі у рамках проекту, такі як операція XOR, розбиття блоку на 4-функції блоку тощо.
3. findWeight2base.py – модуль програми, у якому реалізовано знаходження СЛК S-блоку у його булевому представленні.
4. findWeight4base.py – модуль програми, у якому реалізовано знаходження СЛК S-блоку у його четвірковому представленні.
5. base4recursia.py – модуль програми, що відповідає за синтез якісних S-блоків за допомогою методу описаного у розділі 2.
6. gui.py – модуль програми, в якій реалізовано графічний інтерфейс користувача, а також основна логіка роботи програми.

Розглянемо останні два програмні модулі більш детально. Почнемо з base4recursia.py.

## 3.2 Модуль створення якісних S-блоків

Модуль у собі містить усього один клас Creation в якому і реалізовано логіку роботи алгоритму, представленого у розділі 2. Клас містить наступні методи:

1. \_\_init\_\_;
2. createDiad;
3. createG1;
4. createG0;
5. createSbox;
6. newSbox.

Покажемо програмну реалізацію методу \_\_init\_\_ (рисунок 3.1).

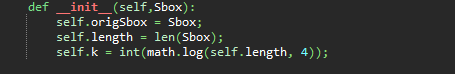


Рисунок 3.1 – метод init

Даний метод є конструктором класу відповідно до вимог об’єктно-орієнтованого програмування у мові Python. Поля класу origSbox, length, k ініціалізують відповідно початковий S-блок, його поточну довжину та параметр k відповідно до формули 2.11.

Програмна реалізація методу createDiad зображена на рисунку 3.2

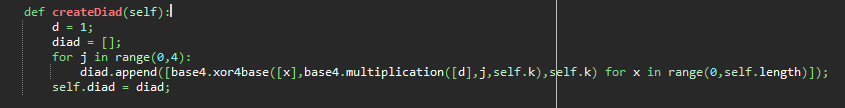


Рисунок 3.2 – метод createDiad

Даний код створює діадні зсуви відповідно до 2.11, де:

d – одиничний вектор ;

daid – список діадних зсувів;

diad.append() – додавання діадних зсувів до загального списку, що утворені згдіно зазначеної формули.

Після чого утворений список діадних зсувів присвоюється до значення поля класу.

Перейдемо до програмної реалізації методу createG1 (рисунок 3.3):

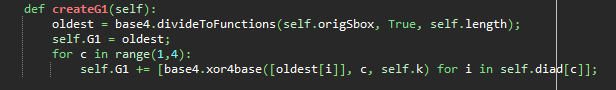


Рисунок 3.3 – метод createG1

Даний код формує елемент g1 за 2.13. відповідно до змінних наданих у зазначеній формулі маємо:

oldest – змінна, що зберігає значення найстаршої 4-функції S-блоку;

c – параметр зазначеної формули

Тепер розглянемо метод createG0 (рисунок 3.4).

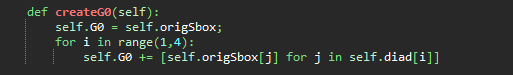


Рисунок 3.4 – метод createG0

Відповідно до своєї назви метод формує елемент g0 згідно 2.15, що представляє собою горизонтальну конкатенацію оригінального блоку з його перестановками за правилами отриманих діадних зсувів.

Метод createSbox має такий програмний код, зазначений у рисунку 3.5:

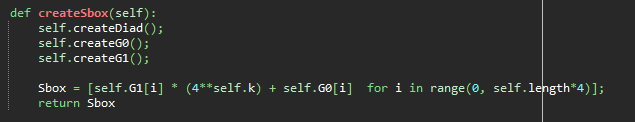


Рисунок 3.5 – метод createSbox

Даний метод викликає усі методи, що були описані раніше і синтезує новий S-блок відповідно до 2.16.

Метод newSbox робить переініціалізацію полів, що описані у методі \_\_init\_\_.

* 1. Графічний модуль програми

Для зручності використання програми було створено модуль графічного інтерфейсу, що поєднує у собі як безпосередньо саму графіку так і логіку роботи програми з використанням раніше описаних модулів.

Логічно модуль розподілено на три частини:

1. безпосередньо сам інтерфейс на базі pySimpleGUI;
2. допоміжні функції;
3. клас goodBlock;

Покажемо програмну реалізацію графічного інтерфейсу програми. Приклад наведено на рисунку 3.6.

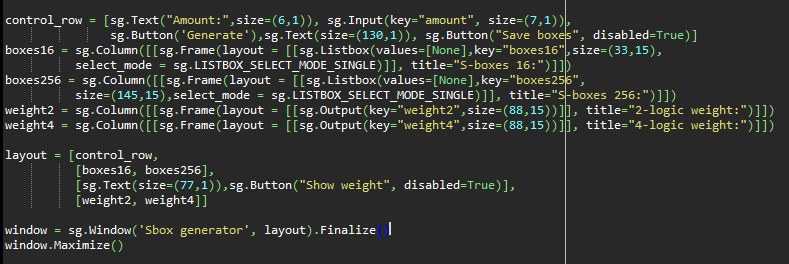


Рисунок 3.6 – Програмна реалізація графічного інтерфейсу

Змінна layout містить у собі фактично усі активні елементи графічного вікна програми, а також відповідно до місця їх ініціалізації у коді розсташовує їх на фреймі.

Змінна window відповідає за створення самого вікна програми і за допомогою змінної layout розміщує елементи (рисунок 3.7)

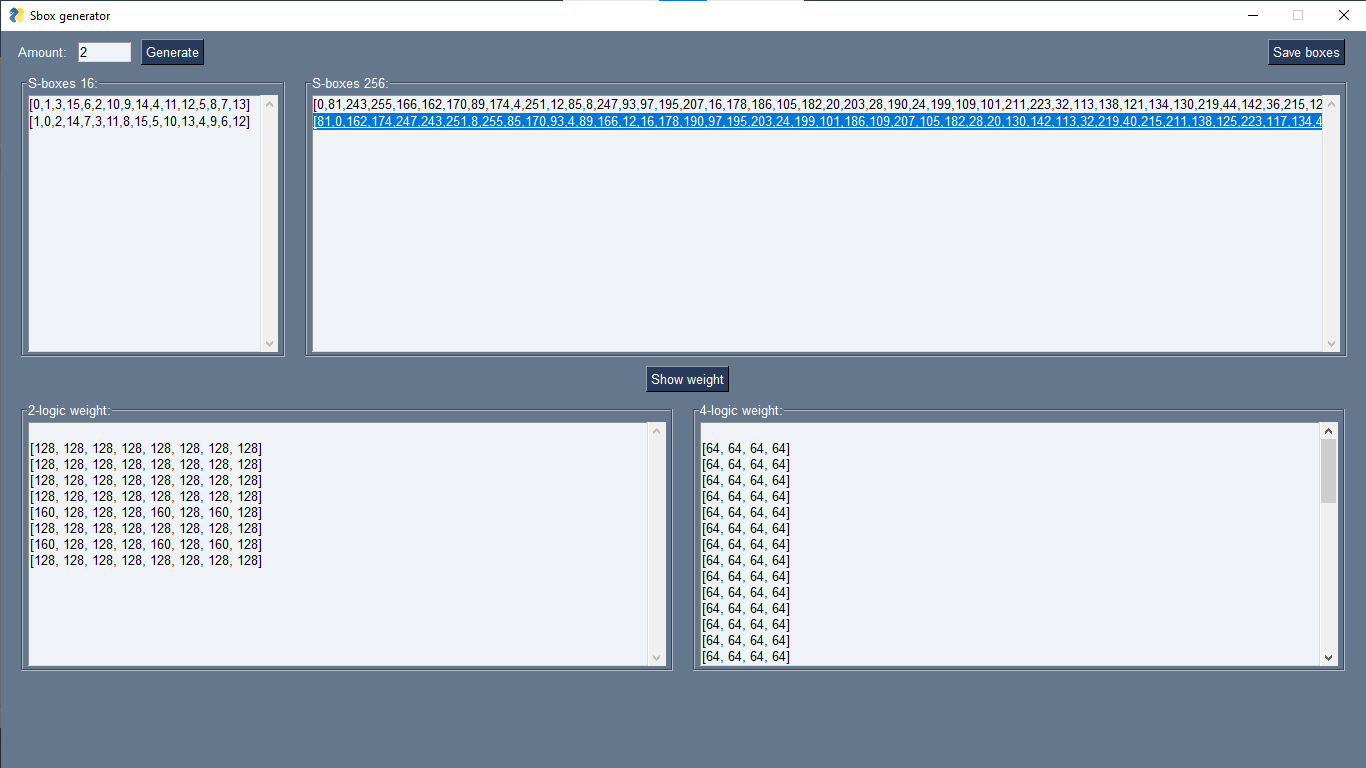


Рисунок 3.7 – Графічний інтерфейс програми

Серед допоміжних функцій варто виділити метод saveSboxes, що відповідає за збереження синтезованих блоків у окремий текстовий файл. Наведемо код даного методу на рисунку 3.8.

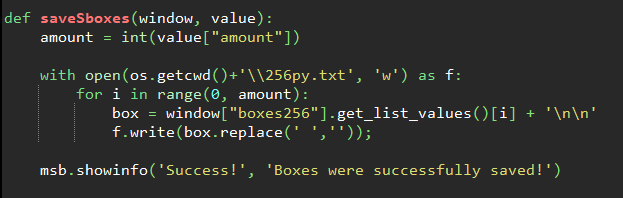


Рисунок 3.8 – функція saveSboxes

У даному коді змінна amount відповідає за необхідну кількість синтезованих S-блоків. Далі відбувається створення файлу 256py.txt у директорії, в якій знаходиться програма та запис усіх створених блоків до цього файлу. При цьому кожний записаний S-блок записується з нового рядка. Якщо запис пройшов успішно, то виводиться відповідне повідомлення.

Останньою логічною частиною програми є клас goodBlock(), який містить наступні методи:

1. метод ініціалізації;
2. checkAmount;
3. calculate;
4. check.

Метод ініціалізації відповідає за створення екземпляру класу відповідно до вимог середовища програмування Python. Метод checkAmount відповідає за валідацію даних, що були введені користувачем. Якщо було розпізнано некоректне значення, то воно автоматично заміняються на одинцю. Тобто, коли користувач цілеспрямовано або раптово вводить невірні дані, то він автоматично отримує повідомлення про некоректність введених даних та отримує один синтезований блок. Детальний код наведено у додатку Г.

Розглянемо код методу calculate (рисунок 3.9)

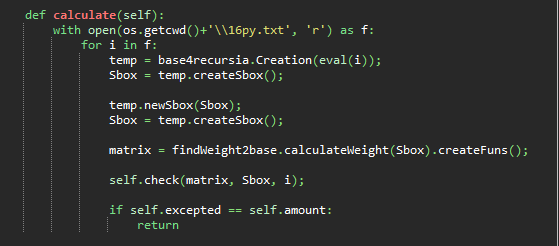


Рисунок 3.9 – метод calculate

Даний метод зчитує з файлу 16py.txt якісні блоки довжини 16, після чого на їх основі синтезує якісні блоки довжини 256. Далі отримує вагову матрицю блоку у його бінарному представленні і заносить її до змінної matrix. Після чого викликається метод check, який буде описано пізніше. Далі йде перевірка кількості отриманих якісних блоків з заданою користувачем.

Перейдемо до коду методу check (рисунок 3.10).

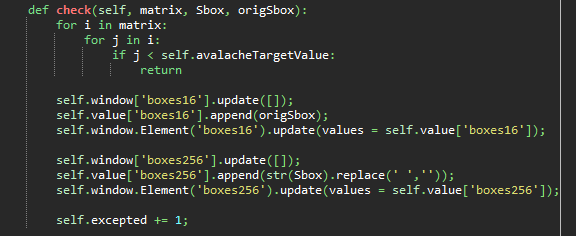


Рисунок 3.10 – метод check

У даному методі матриця, що була отримана після виклику методу calculate я основним аргументом методу. Для кожного значення ваги, що міститься у змінній matrix відповідно до **Визначення 7** перевіряється, щоб воно було не менше ніж половина блоку. У нашому випадку 128.

Якщо все гаразд, то виводиться оригінальний та синтезований блоки у відповідні текстові поля програми. Роботу програми наведено на блок-схемі (рисунок 3.11)

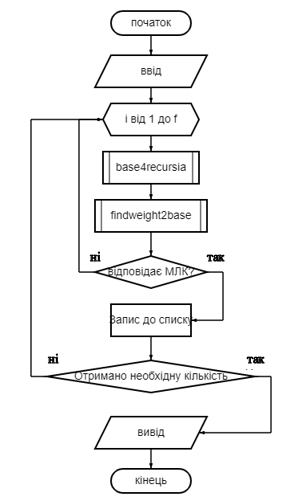


Рисунок 3.11 – блок-схема програми

* 1. Практичне використання

Для того, щоб перевірити актуальність побудованої множини якісних S-блоків, було створено один і той самий зашифрований файл за допомогою AES з класичним S-блоком з множини Ніберг та за допомогою AES на базі одного з синтезованих блоків. Дані порівняння результатів роботи приведено у таблиці 3.1, де PASS – складено, FAIL – провалено.

Таблиця 3.1 – порівняння роботи AES на базі різних S-блоків

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тест | Класичний AES | | AES з якісним блоком | |
| Значення | Статус | Значення | Статус |
| Однакові біти | 0.0391402044228257 | PASS | **0.4610894216047055** | PASS |
| Частотний блоковий | 0.5375901438621488 | PASS | **0.6699758863919024** | PASS |
| Найдовша послідовність однакових значень | 0.0258087301371597 | PASS | **0.8000073990450451** | PASS |
| Найдовша послідовність  одиниць | 0.0206243458906427 | PASS | **0.4258279984261103** | PASS |
| Ранг бінарних  матриць | **0.7200251234927659** | PASS | 0.0733987495613315 | PASS |
| Спектральний | 0.1870772994327211 | PASS | **0.672274617689417** | PASS |
| Непересічені шаблони | 1.0000000003922433 | PASS | **1.0000000009814483** | PASS |
| Пересічені  шаблони | 0 | FAIL | 0 | FAIL |
| Універсальний тест Маурера | 0.0058563020938382 | FAIL | **0.3844125149322922** | PASS |
| Лінійна складність | 0 | FAIL | 0 | FAIL |
| Підпослідовності | 0.0461895735289296 | PASS | **0.2487496469441564** | PASS |
| Ентропія | 0.0462190863989118 | PASS | **0.2490245833799422** | PASS |
| Кумулятивні суми | 0.0263193306260885 | PASS | **0.3863858469830586** | PASS |
| Довільне відхилення | 0.0630573959370637 | PASS | **0.2725316028020707** | PASS |
| Довільне варіативне відхилення | **0.1447142396564906** | PASS | 0.0747045059001115 | PASS |

Неважко помітити, що AES на базі блоку, синтезованого у цій роботі показує значно кращі результати, програючи тільки у тестах на ранг бінарних матриць та на довільне варіативне відхилення.

На практиці було доведено, що синтезовані S-блоки мають достатньо якісні характеристики. Це привело до того, що всього лише заміна блоку в криптоалгоритмі AES значно покращила якість його роботи. Виходячи з цього, отримані блоки можна рекомендувати для використання у будь-якому блоковому симетричному шифрі, наприклад AES.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Shannon, C.E. A Mathematical Theory of Cryptography. C.E. Shannon. – Bell System Technical Memo MM 45-110-02., 1945. – 132 p.
2. FIPS 197. [Electronic resource] Advanced encryption standard. – 2001. – URL: http://csrc.nist.gov/publications/
3. Жданов О.Н. Методика выбора ключевой информации для алгоритма блочного шифрования. О.Н. Жданов. – М.: ИНФРА-М, 2013 г. С. 90.
4. Соколов, А.В. Новые методы синтеза нелинейных преобразований современных шифров. А.В. Соколов. – Lap Lambert Academic Publishing, Germany, 2015. – 100 с.
5. Горбенко, І.Д. Дослідження аналітичних і статистичних властивостей булевих функцій криптоалгоритму RIJNDAEL (FIPS 197). І.Д. Горбенко, О.В. Потій, Ю.А. Ізбенко. Радіотехніка: всеукр. міжвідом. наук.-техн. зб. – Харків, 2004 . – Т. 126. – С. 132 – 138.
6. Мазурков, М.И. Криптографические свойства нелинейного преобразования шифра Rijndael на базе полных классов неприводимых полиномов. М.И. Мазурков, А.В. Соколов. Одеський політехнічний університет. Праці, 2012. – №2(39). – С. 183–189.
7. Sokolov, A.V. Prospects for the Application of Many-Valued Logic Functions in Cryptography. A.V. Sokolov, O.N. Zhdanov. – International Conference on Theory and Applications of Fuzzy Systems and Soft Computing, 2018. – P. 331–339.
8. Строгий лавинный критерий. *Матеріал з Вікіпедії – вільної енциклопедії*– 2020. – URL: <https://ru.wikipedia.org/wiki/Лавинный_эффект>.
9. Sokolov, A.V. Strict avalanche criterion of four-valued functions as the quality characteristic of cryptographic algorithms strength. A.V. Sokolov, O.N. Zhdanov. – Siberian Journal of Science and Technology, 2019. – Vol. 20. – No. 2. – P. 183–190.
10. Логачев, О.А. Булевы функции в теории кодирования и криптологии. О.А. Логачев, А.А. Сальников, В.В. Ященко. – М: Издательство МЦНМО, 2004. – 472 с.
11. М. Chandrasekharappa T.G.S., Prema K.V., Kumara Shama. S-boxes generated using Affine Transformation giving Maximum Avalanche Effect, Internation Journal of Computer Science and Engineering. Manipal Institute of Technology, India. Vol.3(#9). 2011. P.3185–3193.
12. Canright, D. A Very Compact S-Box for AES / D. Canright // LNCS 3659, Workshop on Cryptographic Hardware and Embedded Systems (CHES2005). – USA NY: Springer-Verlag. – P. 441-455. – 2005. – ISSN: 0302-9743.
13. Панасенко, С.П. Алгоритмы шифрования. Специальный справочник / С.П. Панасенко. – СПб.: БХВ - Петербург. – 2009. – 576 с.
14. Cobb M. Advanced Encryption Standard (AES) [Електронний ресурс] / Michael Cobb. – 2020. – Режим доступу до ресурсу: https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard.
15. Костюхін К. AES-128. Деталі та реалізація на python [Електронний ресурс] / Кирило Костюхін. – 2016. – Режим доступу до ресурсу: https://habr.com/ru/post/212235/.
16. Мазурков М.И. Семейства линейных рекуррентных последовательностей на основе полных множеств изоморфных полей Галуа. Мазурков М.И., Конопака Е.А. Радиоэлектроника. ― 2005. – № 11. – С. 58 – 65. (Изв. вузов).
17. Nyberg, K. Differentially uniform mappings for cryptography. Advances in cryptology. K. Nyberg. – Proc. of EUROCRYPT’93. Berlin, Heidelberg, Lecture Notes in Compuer Springer-Verlag, New York, 1994, Vol. 765, P.55–65.
18. K Kim K. A recursive construction method of S-boxes satisfying strict avalanche criterion / K. Kim, T. Matsumoto, H. Imai // Proc. of CRYPTO’90, Springer – Verlag. – 1990. – P.565–574
19. Соколов, А.В. Конструктивный метод синтеза нелинейных S-блоков подстановки, соответствующих строгому лавинному критерию / А.В. Соколов // Известия высших учебных заведений. Радиоэлектроника. – 2013. – T. 56, N 8. – С. 43–52.
20. Соколов А. В., Радуш В. В. Лавинні характеристики S-блоків конструкції Ніберг, при їх уявленні за допомогою функцій багатозначної логіки. Інформатика та математичні методи в моделюванні. 2019. С. 111-120.
21. Жданов, О.Н. О распространении конструкции Ниберг на поля Галуа нечетной характеристики / О.Н Жданов, А.В. Соколов // Известия высших учебных заведений. Радиоэлектроника, 2017. – T. 60, № 12. – С. 696–703.

# Додаток А. Q-послідовність

Q = {0 1 141 246 203 82 123 209 232 79 41 192 176 225 229 199 116 180 170 75 153 43 96 95 88 63 253 204 255 64 238 178 58 110 90 241 85 77 168 201 193 10 152 21 48 68 162 194 44 69 146 108 243 57 102 66 242 53 32 111 119 187 89 25 29 254 55 103 45 49 245 105 167 100 171 19 84 37 233 9 237 92 5 202 76 36 135 191 24 62 34 240 81 236 97 23 22 94 175 211 73 166 54 67 244 71 145 223 51 147 33 59 121 183 151 133 16 181 186 60 182 112 208 6 161 250 129 130 131 126 127 128 150 115 190 86 155 158 149 217 247 2 185 164 222 106 50 109 216 138 132 114 42 20 159 136 249 220 137 154 251 124 46 195 143 184 101 72 38 200 18 74 206 231 210 98 12 224 31 239 17 117 120 113 165 142 118 61 189 188 134 87 11 40 47 163 218 212 228 15 169 39 83 4 27 252 172 230 122 7 174 99 197 219 226 234 148 139 196 213 157 248 144 107 177 13 214 235 198 14 207 173 8 78 215 227 93 80 30 179 91 35 56 52 104 70 3 140 221 156 125 160 205 26 65 28}.

# Додаток Б. Відхилення компонентних функцій S-блоків конструкції Ніберг від СЛК

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Поліном | Двійковий випадок | | Четверичний випадок | | Шістнадцятеричний випадок | |
|  |  |  |  |  |  |
| 1 | 285 | 12 | 516 | 16 | 1040 | 11 | 2848 |
| 2 | 299 | 16 | 428 | 17 | 992 | 11 | 2836 |
| 3 | 301 | 16 | 488 | 13 | 764 | 14 | 2644 |
| 4 | 333 | 16 | 464 | 16 | 924 | 12 | 2704 |
| 5 | 351 | 16 | 556 | 18 | 952 | 11 | 2820 |
| 6 | 355 | 16 | 408 | 13 | 848 | 12 | 2724 |
| 7 | 357 | 16 | 404 | 18 | 896 | 15 | 2844 |
| 8 | 361 | 16 | 480 | 18 | 892 | 10 | 2916 |
| 9 | 369 | 12 | 388 | 14 | 768 | 13 | 2732 |
| 10 | 391 | 12 | 432 | 16 | 960 | 12 | 2992 |
| 11 | 397 | 16 | 456 | 12 | 960 | 12 | 2772 |
| 12 | 425 | 16 | 444 | 16 | 868 | 14 | 2948 |
| 13 | 451 | 12 | 408 | 12 | 684 | 15 | 2856 |
| 14 | 463 | 16 | 496 | 15 | 820 | 13 | 2924 |
| 15 | 487 | 12 | 360 | 14 | 964 | 11 | 2700 |
| 16 | 501 | 16 | 440 | 16 | 944 | 13 | 2872 |
| 17 | 283 | 12 | 376 | 12 | 748 | 12 | 2724 |
| 18 | 313 | 16 | 424 | 20 | 924 | 11 | 2836 |
| 19 | 319 | 16 | 376 | 14 | 760 | 14 | 2676 |
| 20 | 375 | 16 | 364 | 16 | 796 | 11 | 2668 |
| 21 | 379 | 16 | 512 | 20 | 988 | 10 | 2764 |
| 22 | 395 | 12 | 416 | 16 | 888 | 11 | 2920 |
| 23 | 415 | 12 | 416 | 15 | 796 | 12 | 2716 |
| 24 | 419 | 16 | 536 | 14 | 1040 | 15 | 2900 |
| 25 | 433 | 16 | 520 | 12 | 1028 | 14 | 2872 |
| 26 | 445 | 16 | 464 | 16 | 860 | 13 | 2628 |
| 27 | 471 | 12 | 472 | 18 | 912 | 13 | 2840 |
| 28 | 477 | 16 | 504 | 14 | 820 | 12 | 2556 |
| 29 | 499 | 12 | 440 | 20 | 976 | 14 | 2744 |
| 30 | 505 | 16 | 412 | 18 | 912 | 12 | 2984 |

# Додаток В. Створений якісний S-блок



Додаток Г. Лістинг програмного коду