Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Метод і засіб завадостійкого розподілу секрету»

08-20.МКР.001.00.000 ПЗ

Виконав: студент 2 курсу, групи 1БС-18м
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

_____ Бевзюк А. М.

Керівник: д. т. н., проф., зав. каф. ЗІ

_____ Лужецький В. А.

Рецензент: к.т.н., проф., доц. каф. ОТ

_____ Азарова А. О.

Вінниця - 2019 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Освітньо-кваліфікаційний рівень магістр
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри ЗІ, д. т. н., проф.

_____ В. А. Лужецький

_____ _____ 2019 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Бевзюку Антону Миколайовичу

1. Тема роботи: «Метод і засіб завадостійкого розподілу секрету»
   керівник роботи: Лужецький Володимир Андрійович, д.т.н., проф., зав.каф. ЗІ,
   затверджена наказом ректора ВНТУ№ 254 від 02.10.2019 р.
2. Строк подання студентом роботи _____ 2019 р.
3. Вихідні дані до роботи:
   - кольорове зображення;
   - формат зображення – bmp;
   - кількість частин для роподілу – 3;
   - мова програмування Node.js;
   - середовище програмування – Visual Studio Code.
4. Зміст пояснювальної записки: Вступ. Аналіз літературних джерел. Метод завадостійкого розподілу секрету. Програмний засіб для розподілу секрету. Економічна частина. Висновки. Перелік використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу.

Послідовність етапів методу розподілу секрету (плакат, А4). Формування конрольних байтів (плакат, А4). Послідовність етапів методу відновлення секрету (плакат, А4). Алгоритм формування перестановок та замін (плакат, А4). Вигляд секретного зображення (плакат, А4). Вигляд частин розподіленого секрету (плакат, А4). Застосування медіанної фільтрації (плакат, А4). Вигляд зображення при відновленні двома учасниками (плакат, А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Лужецький В. А., д.т.н., проф., зав.каф.ЗІ		
2	Лужецький В. А., д.т.н., проф., зав.каф.ЗІ		
3	Лужецький В. А., д.т.н., проф., зав.каф.ЗІ		
4	Мацкевічус С. С., ст. викл. каф. ЕПВМ		

7. Дата видачі завдання _____ 2019 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2019 – 04.09.2019	
2	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	05.09.2019 – 5.09.2019	
3	Науково-технічне обґрунтування	16.09.2019 – 22.09.2019	
4	Розробка технічного завдання	23.09.2019 – 29.09.2019	
5	Розробка рішень	30.09.2019 – 12.10.2019	
6	Практична реалізація, моделювання, експериментування, результати	14.10.2019 – 10.11.2019	
7	Розробка розділу економічного обґрунтування доцільності розробки	11.11.2019 – 17.11.2019	
8	Аналіз виконання ТЗ, висновки	18.11.2019 – 24.11.2019	
9	Оформлення пояснювальної записки	25.11.2019 – 30.11.2019	
10	Попередній захист та доопрацювання МКР	28.11.2019 – 01.12.2019	
11	Перевірка магістерської роботи на наявність плагіату	02.12.2019 – 10.12.2019	
12	Представлення МКР до захисту	11.12.2019 – 14.12.2019	
13	Захист МКР	16.12.2019 – 20.12.2019	

Студент _____ Бевзюк А. М.
(підпис)

Керівник роботи _____ Лужецький В. А.
(підпис)

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробці методу завадостійкого розподілу секретного вмісту зображення, що забезпечує криптографічний захист комп`ютерної інформації. Для усунення помилок у відновленому зображенні використовується двовимірний масив контрольних байтів та медіанна фільтрація. Для розробки програмного засобу проведено дослідження основних методів розподілу секрету, досліджено та реалізовано алгоритм формування перестановок та замін, розроблено ряд схем і алгоритмів, здійснено програмну реалізацію. Засіб перевірено на предмет коректності роботи, доведено ефективність здійснюваного захисту.

ABSTRACT

Master's qualification work is devoted to the development of the method of noise-free distribution of secret content of the image, which provides cryptographic protection of computer information. Two-dimensional array of control bytes and median filtering are used to correct errors in the reconstructed image. In order to develop the software, research was conducted on the basic methods of secret distribution, the algorithm for forming permutations and substitutions was investigated and implemented, a number of schemes and algorithms were developed, and software implementation was carried out. The tool is checked for the correctness of the work, proved the effectiveness of the protection.

ЗМІСТ

ВСТУП

Кожний вид інформації має свої специфічні особливості, що значно впливають на вибір методів її шифрування. Велике значення відіграють об’єм та необхідна швидкість передачі даних. Потреба вирішення проблеми захисту інформації обумовлює актуальність розробки різних алгоритмів для криптографічних перетворень, що використовують для захисту інформації в комп’ютерних системах та мережах.

Поняття секретних даних, як даних, які не підлягають розголошенню, або на розповсюдження яких накладено обмеження, відомо людям з давніх часів і зрозуміло кожному. Такі дані потребують захисту від неузгодженого розголошення та передачі, а також непередбаченої втрати, для цього потрібно забезпечити надійність їх зберігання.

В даний час значна частина глобального обміну даними відбувається через відкриті мережеві з'єднання. Велика кількість даних є практично всім. Необхідно гарантувати інформаційну безпеку і захист [1, 2]. Разом із застосуванням шифрування, секретні ключі повинні зберігатися в захищеному сховище. Секретні ключі безпосередньо не повинні бути доступні будь-де в системі без зарання забезпеченого фізичного захисту. Нажаль, ця умова майже нездійсненна.

З метою забезпечення безпеки ключів застосовуються порогові схеми розподілу секрету: ключ ділиться на декілька частин і зберігається в різних місцях.

Наприклад, є важлива секретна інформація, яку можна втратити. Її небезпечно довірити комусь одному. Виникає питання, як підвищити надійність і безпеку її зберігання. Перший шлях – зробити кілька копій цих даних і зберігати їх в різних місцях. Резервування забезпечує високу надійність зберігання, але якщо скомпрометована хоча б одна копія, то секретність всієї інформація буде втрачена. Другий шлях – розподілити секрет на кілька частин і зберігати їх в різних місцях, при необхідності збираючи разом.

Секретом може виступати будь-яка інформація, текстові документи, аудіо - та відео файли, зображення. Також необхідно, щоб після відновлення початкового файлу він був такий самий як і до розподілу, або був максимально наближений до нього.

Кожний вид інформації має допустиму межу для похибки. Наприклад, текст повинен бути відновлений на 100%, а якщо в зображенні чи аудіофайлі буде зіпсовано декілька пікселів чи байтів, то це не буде помітно людському зору чи слуху.

Об'єктом дослідження є процес розподілу секрету.

Предмет дослідження – метод завадостійкого розподілу секрету та засіб, що реалізує його.

Метою магістерської кваліфікаційної роботи є підвищення якості відновлення секретного вмісту.

Задачами магістерської кваліфікаційної роботи є такі:

– проаналізувати протоколи розподілу секрету;
– розробити метод розподілу секретного вмісту зображення;
– розробити алгоритм перестановки та заміни;
– розробити програмний засіб та провести тестування.

Науковим результатом магістерської кваліфікаційної роботи є удосконалений метод розподілу секрету, який відрізняється від відомих тим, що завадостійкість забезпечується використанням ітеративних кодів і медіанної фільтрації, що дозволяє покращити якість зображення, що відновлюється.

Практичне значення одержаних результатів полягає в тому, що використання запропонованого методу та конкретного рішення дозволяє отримати більш досконалий, в порівнянні з відомими, програмний засіб, який використовується для розподілу секрету. Основним практичними результатом магістерської кваліфікаційної роботи є програмний засіб, що демонструє роботу завадостійкого розподілу секрету.

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ ЗА ТЕМОЮ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1 Науково-технічне обґрунтування теми магістерської кваліфікаційної роботи

В даний час актуальним є питання захисту секретних ключів різних програмно-апаратних комплексів (ПАК) з розподіленою структурою доступу, таких як засвідчувальні центри (ЗЦ) та апаратні модулі захисту конфіденційної інформації. Відомі різні методи підвищення секретності ключів, серед яких можна виділити метод, заснований на застосуванні схем розподілу секрету (СРС). За такими схемами секретний ключ (секрет, що розподіляється) шляхом математичних перетворень «ділиться» на N частин (часток) секрету і видається N учасникам структури доступу ПАК. Далі, для відновлення вихідного секретного ключа необхідно «зібрати разом» N частин секрету.

Вперше, завдання криптографічного поділу секрету для випадку порогової структури доступу були незалежно сформульовані і вирішені Шаміром (A. Shamir) [3] і Блеклі (G.R. Blakley) [4]. За три десятиліття існування, задача розподілу секрету перетворилася в область сучасної криптографії, що активно розвивається.

Протоколи розподілу секрету використовуються для зниження ризику компрометації деяких критичних даних. Припустимо, необхідно захистити деяку банківську таємницю (можливо, секрет – код доступу до зашифрованих рахунків). З деякою ймовірністю в банку працює «агент конкуруючої фірми». Для забезпечення безпеки секрету назначається декілька співробітників, які отримують свої частини секрету. Потім секрет розподіляється на кілька частин і зберігати їх в різних місцях і у призначених співробітників, які при необхідності збираються разом.

Існуючі схеми мають дві складові: поділ і відновлення секрету. До поділу відноситься формування частин секрету і розподіл їх між членами групи, що

встановлює відповідальність за секрет між її учасниками. Зворотна схема повинна забезпечити відновлення за умови необхідної кількості учасників.

1.2 Схема Шаміра

Протоколи розподілу секрету (secret splitting) використовуються для зниження ризику компрометації деяких критичних даних.

Наприклад, необхідно захистити деяку банківську таємницю (можливо, секрет – код доступу до зашифрованих рахунків). З деякою ймовірністю в банку працює «агент конкуруючої фірми».

Для забезпечення безпеки секрету назначається декілька співробітників, які отримують свої частини секрету. Наприклад, секрет ділиться між чотирма співробітниками і використовується такий протокол.

1) адміністратор генерує три випадкові рядки S_1, S_2, S_3 такої ж довжини, що і сам секрет S;

2) далі обчислюється $S_4 = S \oplus S_1 \oplus S_2 \oplus S_3$;

3) адміністратор доручає зберігання S_1 першому співробітникові, S_2 – другому співробітникові і так далі до S_4;

4) при необхідності відновлення секрету: $S_4 = S_4 \oplus S_1 \oplus S_2 \oplus S_3$.

У цьому протоколі адміністратор – головна особа, його шахрайство вже ніяк не контролюється аж до моменту відновлення секрету. Недоліком схеми є те, що втрата одним з учасників протоколу своєї частини приведе до неможливості відновлення секрету.

Ефективнішим на практиці є протокол розподілу секрету запропонований Шаміром [5]. У цій схемі деяка інформація (секрет) ділиться на n частин, званих тінями (shadows) або частинами секрету, так, щоб будь-яких t частин було достатньо для відновлення секрету. Схема носить назву (n,t) – порогової схеми (threshold scheme).

Через дві точки можна провести необмежену кількість поліномів ступеня 2. Щоб вибрати з них єдиний – потрібна третя точка. На рис. 1.1 зображено

представлення схеми Шаміра.

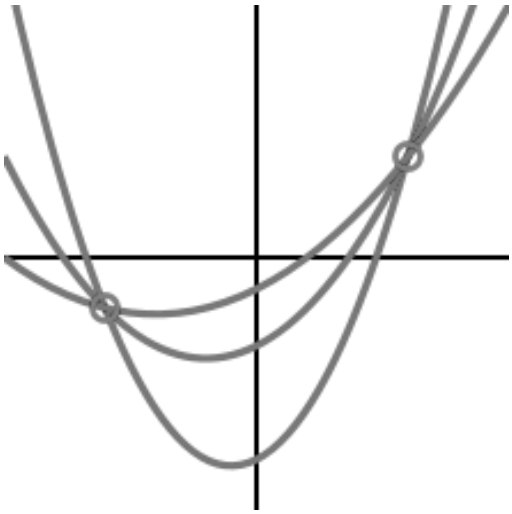


Рисунок 1.1 – Представлення схеми Шаміра

Схема Шаміра використовує так званий інтерполяційний поліном Лагранжа для системи *m* різних пар точок (x_i,y_i).

Цей поліном серед поліномів степеня не вище *m* визначається однозначно і приймає в точках x_i значення y_i .

Вибирається випадковим чином секретний набір лишків $(a_1,...,a_{t-1})$, $a_{t-1} \neq 0$ за великим модулем *p* і утворюється поліном $Q(x) = a_0 + a_1 x ... + a_{t-1}x^{t-1}$, де $a_0 = S$ (наш секрет).

Для кожного з *n* учасників протоколу вибирається відповідно попарно різні несекретні лишки $(b_1,...,b_n)$. Обчислюються значення $Q(b_i)$ і розподіляються між користувачами у вигляді частин секрету пари виду $(b_i, Q(b_i))$, $i = 1,...,n$. Тут використовується те, що степінь полінома *(t –1)* не обмежує кількість точок (n), в яких потрібно обчислити значення полінома.

Для відновлення секрету використовується формула Лагранжа, згідно з якою многочлен степеня *t –1* можна відновити за допомогою *t* попарно різних точок $(b_i, Q(b_i))$, при цьому вільний член обчислюється за формулою:

$$S = a_0 = \sum_{i=1}^{t} Q(b_i) \cdot \prod_{j \neq 1} b_j (b_j - b_i)^{-1} \bmod p$$

Даний вираз дозволяє довільній групі з *t* користувачів обчислити секрет, а

групи, що складаються з меншого числа користувачів, цієї процедури виконати не можуть.

Нижче наведена конструкція протоколу розподілу секрету, що перевіряється, з роботи [6]. Конструкція заснована на складності дискретного логарифмування.

За аналогією зі схемою Шаміра дилер формує секретний випадковий поліном $Q(x) = a_0 + a_1 x + ... + a_t x^t$ степеня t, де $a_0 = S$ (наш секрет).

Потім він публікує значення $r_i = g^{a_i} (\mathrm{mod}\, p),\ i = 0,...,t,$ де g – елемент великого порядку за модулем p. Для кожного $j = 1,...,n$ дилер пересилає значення $S_j = Q(j)$ процесору P_j по захищеному каналу.

Процесор P_j може (не знаючи $Q(x)$) проконтролювати виконання рівності $S_j = Q(j)$, оскільки для $A = a_0 + a_1 j + a_2 j^2 + ... + a_t j^t$

$$g^{S_j} = g^{Q(j)} = g^A = r_0 \cdot r_1^j \cdot r_2^{j^2} ... r_t^{j^t} (\mathrm{mod}\, p).$$

Конструкцію протоколу для фази відновлення секрету наведена в найбільш простому випадку, коли дилер чесний.

На цій фазі кожен процесор P_j пересилає по захищеному каналу кожному іншому процесору свою частину S_j.

Будь-який чесний учасник P_i, набувши деякого значення S_j від P_i, перевіряє це значення (як описано вище) і відкидає всі частини секрету, що не пройшли перевірку.

Оскільки чесних учасників не менше $t+1$, P_j отримає принаймні $t+1$ правильних частин секрету. Використовуючи процедуру відновлення секрету зі схеми Шаміра, P_i відновить значення S.

1.3 Схема Блеклі

Як відомо, система k лінійно незалежних порівнянь з k невідомими по простому модулю має рівно одне рішення. На цьому заснована порогова схема Блеклі, створена в 1979 році [7].

Дві непаралельні прямі на площині перетинаються в одній точці. Будь-які дві некомпланарні площини перетинаються по одній прямій, а три некомпланарні площини в просторі теж перетинаються в одній точці. Взагалі *n*x*n*-мірних гіперплощин завжди перетинаються в одній точці. Одна з координат цієї точки буде секретом. Якщо закодувати секрет як декілька координат точки, то вже по одній частці секрету (однієї гіперплощини) можна буде отримати якусь інформацію про секреті, тобто про взаємозалежності координат точки перетину (рис. 1.2).

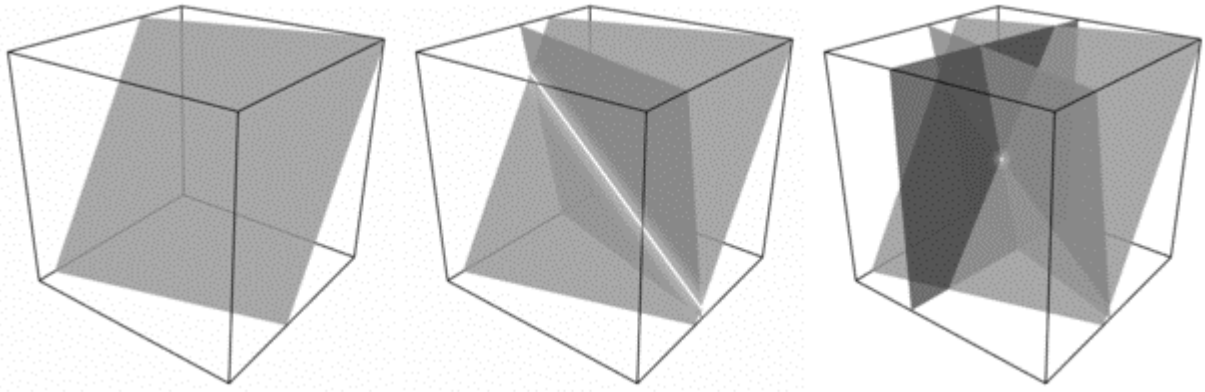


Рисунок 1.2 – Представлення схеми Блеклі

Секрет M розподіляється між n учасниками, будь-яка група, що складається не менше, ніж з k учасників є дозволеною.

Параметрами схеми є:

p – велике просте число (більше будь-якого секрету, який передбачається розподіляти в цій схемі). Тоді $M \in Z_p$.

N– число частин секрету.

K – мінімальне число частин, необхідних для відновлення секрету (розмір дозволеної групи).

З допомогою схеми Блеклі можна створити (k,n) - схему розподілу секрету для будь-яких k і n: для цього треба покласти розмірність простору дорівнює k, і кожному з n учасників дати одну гіперплощину, що проходить через секретну точку [8]. Тоді будь – k з n гіперплощин будуть однозначно перетинатися в

секретній точці.

Дилер, користувач який розподіляє секрет, випадковим чином обирає числа $x_2, x_3, ..., x_k \in Z_p$, та утворює секретну точку $Q(M, x_2, x_3, ..., x_k)$. Для *i*-го учасника (*i=1,2,…,n)* дилер вибирає випадкові рівномірно розподілені на Z_p коефіцієнти $a_{1i}, a_{2i}, a_{3i}, ..., a_{ki}$ і обраховує $b_i = a_{1i}M + a_{2i}x_2 + a_{3i}x_3 + ... + a_{ki}x_k \bmod p.$. Після чого дилер відправляє *i*-му учаснику порівняння $a_{1i}x_1 + a_{2i}x_2 + a_{3i}x_3 + ... + a_{ki}x_k \equiv b_i \bmod p$ (а саме його коефіцієнти) з невідомими $x_1, x_2, x_3, ..., x_k$. При цьому дилер повинен слідкувати за тим, щоб будь-які *k* порівняння були лінійно незалежними.

Для відновлення секрету необхідно зібрати разом *k* учасників, які можуть скласти зі своїх порівнянь систему, вирішивши яку вони знайдуть точку *Q*, перша координата якої розподіляються секретом *M*.

$$\begin{cases} a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + ... + a_{k1}x_k \equiv b_1 \bmod p \\ a_{12}x_1 + a_{22}x_2 + a_{32}x_3 + ... + a_{k2}x_k \equiv b_2 \bmod p \\ ..... \\ a_{1k}x_1 + a_{2k}x_2 + a_{3k}x_3 + ... + a_{kk}x_k \equiv b_k \bmod p \end{cases}$$

Якщо коефіцієнти порівнянь вибираються випадково, то при досить великому модулі перевіркою лінійної незалежності порівнянь можна знехтувати, так як ймовірність порушення цієї умови є вкрай низькою.

Схема Блеклі менш ефективна, ніж схема Шаміра: у схемі Шаміра кожна частина такого ж розміру, як і секрет, а в схемі Блеклі кожна частина в *k* разів більше. Існують поліпшення схеми Блеклі, що дозволяють підвищити її ефективність.

1.4 Схема, заснована на еліптичній кривій

Нехай - еліптична крива, певна в скінченому полі $F(p)$. Властивості точок еліптичної кривої визначаються як властивості адитивних абелевих груп. Відповідно для них визначаються операції додавання:

$$P(x,y) = P_1(x_1,y_1) + P_2(x_2,y_2) \quad (1.3)$$

Для точок еліптичної кривої важливою також є операція множення точки на число:

$$P(x,y) = k \cdot P_1(x_1,y_1) \quad (1.4)$$

Наведене представлення операцій на еліптичній кривій дозволяє провести аналогію з математичним апаратом нейронних мереж скінченого кільця і його адаптацією для систем розподілу секрету. Вираз (1.3) описує підсумовування на нейроні, вираз (1.4) - вагову операцію.

Очевидно, що застосовуючи (1.3) і (1.4), а також апарат нейронних мереж, можна сформувати поліном такого вигляду [9]:

$$S = P_0 + tP_1 + t^2P_2 + ... + t^{k-1}P_{k-1} \quad (1.5)$$

де P_1, P_2, ..., P_{k-1}. – випадкові точки на еліптичній кривій, P_0 – загальний секрет. Вираз (1.5) описує якийсь умовний поліном, що складається з координат точок, операції над якими виконуються за правилами складання точок на еліптичній кривій.

Нехай $t_{ji} = w_{ji}$, тоді можна сформувати матрицю вагових коефіцієнтів нейронної мережі W:

$$W = \begin{pmatrix} w_{00} & w_{01} & \cdots & w_{0(K-1)} \\ w_{10} & w_{11} & & w_{1(K-1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{(N-1)0} & w_{(N-1)1} & \cdots & w_{(N-1)(K-1)} \end{pmatrix} = \begin{pmatrix} 1 & t_0^1 & \cdots & t_0^{K-1} \\ 1 & t_1^1 & & t_1^{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_{N-1}^1 & \cdots & t_{N-1}^{K-1} \end{pmatrix} \quad (1.6)$$

Приватні секрети можна отримати, підставляючи у вираз різні *t*. Нехай *i* - номер абонента, якому відсилається секрет. Тоді приватні секрети абонентів рівні:

$$S = P_0 + iP_1 + i^2P_2 + ... + i^{k-1}P_{k-1} \quad (1.7)$$

Вираз (1.6) матиме вигляд:

$$W = \begin{pmatrix} 1 & i_0^1 & \cdots & i_0^{K-1} \\ 1 & i_1^1 & \cdots & i_1^{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & i_{N-1}^1 & \cdots & i_{N-1}^{K-1} \end{pmatrix}$$

На рисунку 1.3 представлена нейронна мережа скінченого кільця генератора приватних секретів.

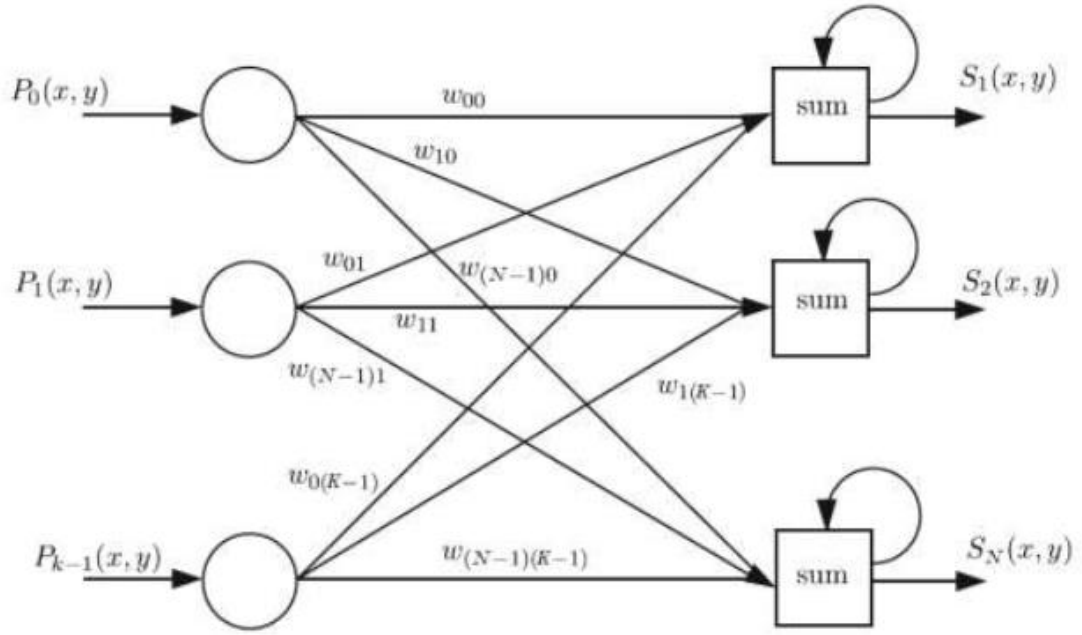


Рисунок 1.3 – Еліптичний генератор приватних секретів

Порогова (N, K) схема. Приватні секрети абонентів в загальному випадку виходять відповідно до вираження:

$$S_i\big|_0^{N-1} = \sum_{i=0}^{k-1} P_i t_j^i \quad (1.8)$$

де N – загальна кількість секретів на першому кроці, j - номер секрету.

Для структури доступу вираз (1.8) перетвориться до виду:

$$S_i\big|_0^{K-1} = \sum_{i=0}^{k-1} P_i t_j^i = \sum_{i=0}^{k-1} P_i (x_i)_j$$

При цьому, в силу наведеної операції, доданок при $i = 1$ дорівнює нулю. Шляхом аналогічних перетворень остаточно можна отримати

$$R = S_{K-1,j} = S_{K-2,j}(x_{K-1})_{j+1} - S_{K-2,j+1}(x_{K-1})_j = P_0 x_K$$

Значення *R* і *x* і є відновлений секрет.

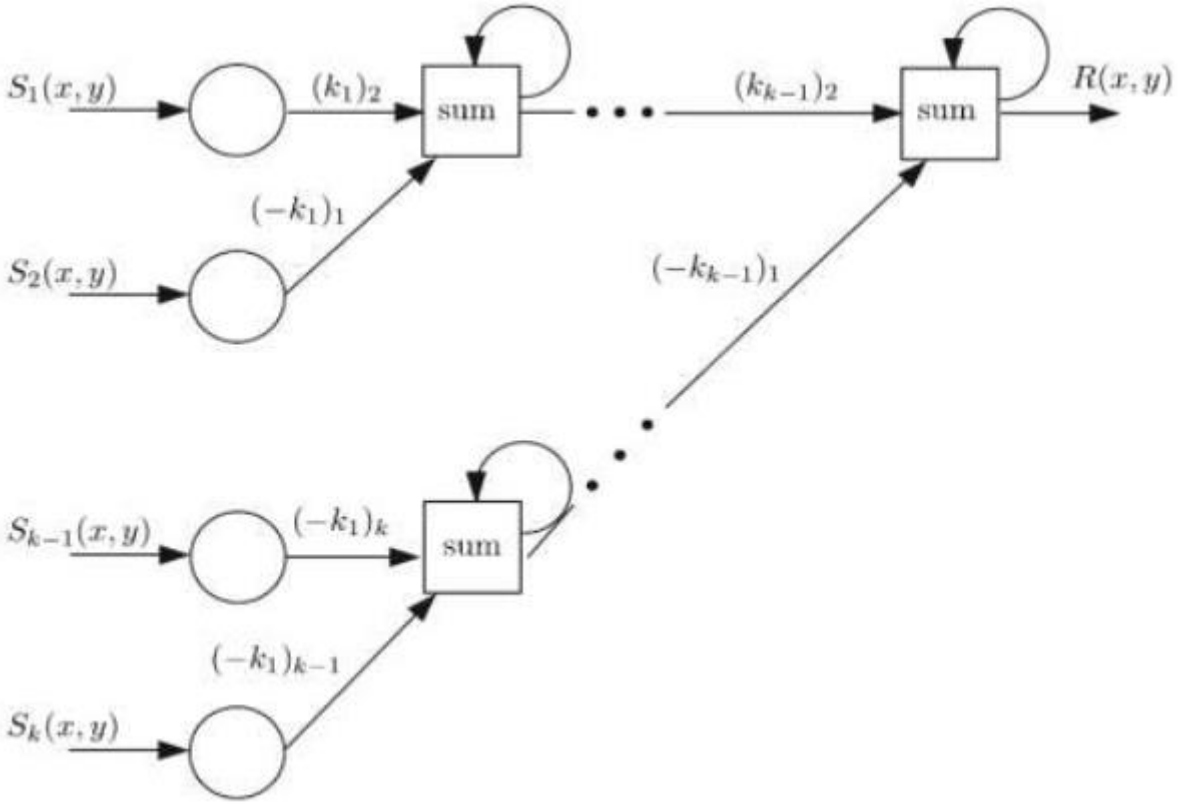


Рисунок 1.4 – Модель мережі генератора загального секрету

На рисунку 1.4 представлена модель мережі генератора загального секрету по відомим приватним.

1.5 Метод візуальної криптографії Моні Наор і Аді Шаміра для чорно-білих зображень

Автори продемонстрували візуальну схему з поділом секрету, згідно якої зображення було розподілене на N частин так, що тільки людина, яка має всі N частин, могла розшифрувати зображення, в той час як інші N - 1 частин не показали б ніякої інформації про оригінальне зображення. Кожна частина була надрукована на окремому діапозитиві, і розшифровка була виконана шляхом накладення цих частин. Тобто при накладенні всіх N частин з'являється вихідне зображення. Таким чином, для декодування не потрібно високопродуктивних обчислень, спеціальних знань і навіть комп'ютера [10].

Використовуючи цей алгоритм в комп'ютерних системах, всі частини

зображення накладаються один на одного за допомогою логічних операцій AND, OR, XOR або шляхом збільшення ступеня прозорості в графічному редакторі.

В *(k, N)* - візуальній схемі зображення розбивається на N частин так, що якщо хтось володіє k частинами, може розшифрувати його, а будь-які k-1 частин не дають ніякої інформації про оригінальний документ. При накладенні всіх k частин стає доступне вихідне зображення.

Наор і Шамір продемонстрували *(k, N)* - візуальну схему секретного обміну, де зображення було розбите на N частин, таким чином, що будь-хто, хто володів будь-якими k частинами міг розшифрувати його, в той час як будь-які k-1 частин не давали ніякої інформації про зміст вихідного зображення. Коли всі k частин будуть накладені одна на одну, вийде вихідне зображення [11].

Для того щоб розбити вихідне чорно-біле зображення на N частин, необхідно кожен піксель зображення представити у вигляді деякої кількості менших частин. Кількість білих і чорних частин завжди однакова. Якщо піксель ділиться на 4 частини, то виходить 2 білих і 2 чорних блоки. Якщо на 2, то один білий і один чорний.

(2, N) – випадок. Це випадок спільного використання секрету довільною кількістю людей N так, що мінімум 2 з них потрібні для декодування секрету. У цій схемі є секретне зображення, яке закодовано в N частинах, надрукованих на прозорій плівці. Частини довільні і не містять інформації про розшифрування секретної інформації, однак якщо будь-які 2 частини накласти одну на одну, то секретне зображення стає розшифрованим для людського ока.

Кожен піксель з секретного зображення кодується в кілька суб-пікселів у кожній частині зображення за допомогою матриці, що визначає колір пікселя.

Обман в схемі *(2, N)* візуальної криптографії. Існує метод, який дозволяє N-1 сторонам, що змовились обдурити чесну сторону. Вони виграють, знаючи закон що лежить в основі розподілу пікселів в частинах, щоб створити нові частини, які комбінують з існуючими для створення нового секретного повідомлення.

Двох частин досить для того, щоб розшифрувати секретне повідомлення за

допомогою зору людини. але розглянуті 2 частини також дають інформацію про третю частину. Наприклад, учасники, що змовились можуть подивитися свої частини, щоб визначити, в яких випадках вони обидва мають чорні пікселі, і використовувати цю інформацію, щоб визначити, що інший учасник також буде мати чорний піксель в цьому місці. Знаючи, де чорний піксель знаходиться в інших частинах, вони можуть створити нову частину, яка буде створена виходячи з раніше отриманих припущень, і дасть нове секретне повідомлення.

При *(2, 2)* – візуальній схемі секретного обміну вихідне зображення розбивається на два «тіньових» зображення, кожне з яких описує собою зображення білого шуму, але при накладенні дають вихідне зображення. Кожен піксель вихідного зображення розбивається на чотири частини, таким чином, якщо розмір вихідного зображення був M × N, то розміри «тіньових» зображень будуть 2M × 2N.

Якщо піксель на першому шарі має одне положення, піксель на другому шарі в свою чергу може мати два положення: ідентичне або інвертоване до пікселя першого шару. Якщо піксель частини 2 ідентичний пікселю частини 1, то піксель, отриманий в результаті накладання обох «тіньових» зображень, буде наполовину білий і наполовину чорний. Такий піксель називають сірим або порожнім. Якщо пікселі частини 1 і частини 2 протилежні, то піксель, отриманий в результаті накладання, буде повністю чорним. Він буде інформаційним.

1.6 Розподілення даних на основі кодів, що виправляють помилки

У цій системі використовуються коди Хеммінга [12, 13], Ріда-Мюллера [12, 13], Ріда-Соломона [12], БЧХ [12, 14]. Нижче описується принцип роботи методу розподілу секрету на прикладі коду Хеммінга (7, 4, 1). Після кодування 4-бітної вихідної інформації утворюється 7-бітна кодована інформація, яка називається кодовим словом. Це дозволить коригувати один хибний біт. Кожен 4 біт секретного файлу (який треба розподілити) кодується кодом Хеммінга (7, 4, 1), і виходять відповідні 7-бітові кодові слова. Ці кодові слова представляються у

вигляді матриці (рис. 1.5). Цей файл після кодування кодом Хеммінга (7, 4, 1) необхідно розподілити.

У цьому прикладі *k* - довжина кодового слова (в даному випадку k = 7), а *q* залежить від обсягу вихідного файлу. Кожен рядок масиву описує собою кодове слово коду Хеммінга (7, 4, 1), це означає, що будь-яка одна помилка може бути виправлена в цих 7 бітах. На основі цієї характеристики можна розподілити по стовпцях. Якщо розглядати кожен стовпець в якості окремого компонента, то очевидно, що можна відновити оригінальний файл в разі пошкодження або втрати будь-якого компонента. Це призводить до (6, 7) порогової схеми.

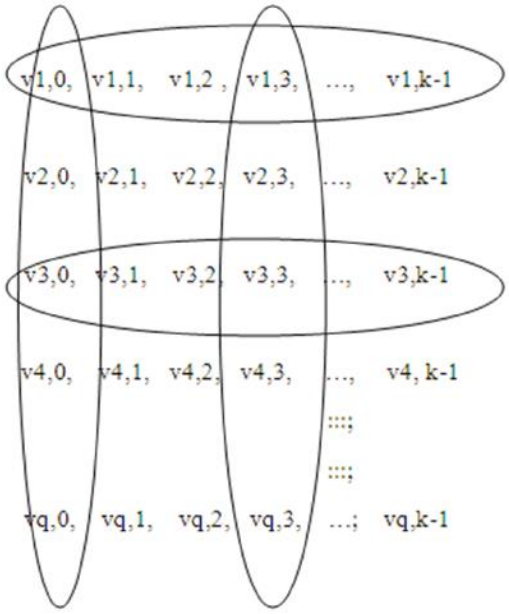


Рисунок 1.5 – Структура кодованого файлу

Щоб збільшити безпеку і мати компоненти, рівні за обсягом файлу, що розподіляється, необхідно застосувати групування за певним методом. Від кожної частини 4 стовпців повинні бути взяті дані (за певним методом), для задоволення всіх вимог. Наприклад, в разі групування згідно таблиці 1.1 отримаємо (2, 4) порогову схему.

Таблиця 1.1 – Порогова схема (2, 4)

Частина 1	3	6	2	7
Частина 2	1	6	7	5
Частина 3	5	4	2	6
Частина 4	7	5	3	4

Кожен рядок є окремою частиною (компонентом). Частина 1 містить кількість стовпців, взятих за певним методом. Таблиця 1.1 показує, що після злиття будь-яких трьох частин, вийде 6 частин, і одна частина буде відсутня. При використанні алгоритму декодування коду Хеммінга (7, 4, 1) можна відновити одну відсутню (зіпсовану) частину. Виявляється, розподіл секрету відповідно до таблиці 1.1 приводить до порогової схеми (2, 4). Секретний файл розділений на чотири частини так, що будь-які три частини достатньо для відновлення. Таблиця 1.2 описує (2, 3) порогову схему.

Таблиця 1.2 – Порогова схема (2, 3)

Частина 1	5	2	6	4
Частина 2	4	7	3	5
Частина 3	6	2	7	3

Таким чином повний розподіл даних забезпечує не тільки конфіденційність, а й цілісність і доступність даних. Для цієї мети, створена нова схема розподілу секрету на основі кодів, що виправляють помилки. Цей метод набагато швидше, оскільки використовує коди, що виправляють помилки, а ці коди використовують логічні операції.

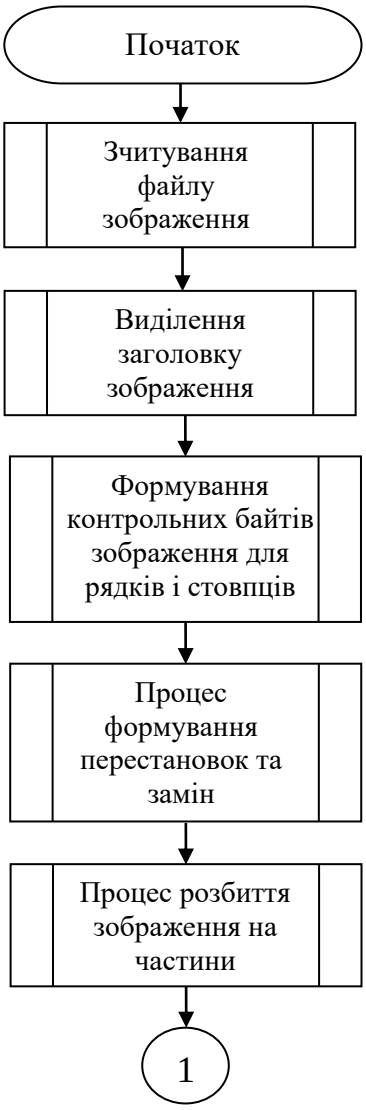
2 МЕТОД ЗАВАДОСТІЙКОГО РОЗПОДІЛУ СЕКРЕТУ

2.1 Метод розподілу секрету

Секретом виступає зображення D, яке буде розподілено між заданою кількістю учасників, кожному з яких дістанеться своя частина секрету. Початковий файл можна відтворити тільки за допомогою об'єднання усіх частин, на які був розподілений секрет.

Розподіл може відбуватись між будь-якою кількістю учасників, але для прикладу секрет буде розподілятись між трьома учасниками.

Для реалізації пропонується послідовність етапів методу розподілу секрету зображених на рисунку 2.1.



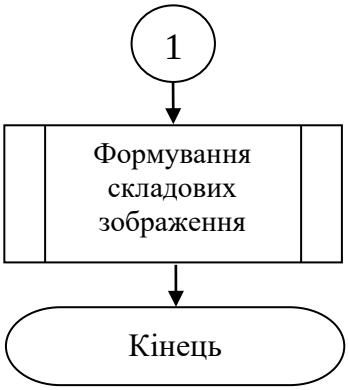


Рисунок 2.1 – Послідовність етапів методу розподілу секрету

Зображення D описується виразом 2.1:

$$D = \{D_1, D_2, D_3\} \tag{2.1}$$

де D_i – кожна частина розподіленого файлу.

Зображення використовується формату BMP та може буде довільного розміру. Обсяг зображення визначається за виразом 2.2:

$$V_D = M \cdot N = V_1 + V_2 + V_3 \tag{2.2}$$

де M – висота зображення, N – ширина, V_i – обсяг кожної частини зображення.

Під час розподілу зображення його заголовок відокремлюється. Оскільки заголовок не несе жодної секретної інформації, тому його відокремлення необхідне для того, щоб він не підпадав під процес формування перестановок та замін задля уникнення визначення алгоритму.

Формування контрольних байтів необхідне для того, якщо якусь із складових розподіленого секрету буде зіпсовано, то при відновленні початкового зображення можна було визначити пошкоджені пікселі та виправити їх. Оскільки контрольні байти додаються для рядків і стовпців, то перетин неправильного стовпця із рядком вкаже на зіпсований піксель, який можна виправити. Якщо буде лише зіпсований рядок або стовпець, то всі пікселі цього ряду або стовпця будуть виправлені за допомогою медіанної фільтрації.

Основними складовими методу розподілу є файл для розподілу (секрет), генератори псевдовипадкових послідовностей (8- та 32-розрядний), секретний ключ, блок перестановок та заміни.

Процес формування перестановок та заміни відбувається за допомогою

генератори псевдовипадкових послідовностей (8- та 32-розрядний) та секретного ключа. Він необхідний для того, щоб зробити зміну розташування пікселів в зображенні та неможливо було відновити їх правильну послідовність іншим шляхом.

Наступним кроком є розподіл перетворених даних на складові секрету. Розподіл відбувається на ту кількість рівних частин, на скількох учасників повинен поділитись секрет. До кожної частини додається виділений на початку заголовок, щоб складові секрету були окремими зображеннями. На виході отримуються файли, що і є результатом роботи методу.

Для відновлення початкового зображення, необхідно мати усі складові кожного з учасників.

2.2 Аналіз структури BMP зображення

BMP (Bitmap) – bitmap-формат або DIB-формат файлу зображень растрової графіки, в якому зображення зберігається у вигляді двовимірного масиву пікселів. Запам’ятовує одно і багатокольорові (RGB) ілюстрації у формі Pixel [15]. BMP-формат використовується в операційній системи Windows. Формат файлу BMP здатний зберігати 2D цифрові зображення довільної ширини, висоти та роздільної здатності, як монохромні так і кольорові, різної глибини кольору, і, необов’язково, зі стисненням даних, альфа-каналом та керуванням кольору. В даному форматі можна зберігати лише одношарові растри.

BMP-файл складається з трьох частин:

1) заголовок файлу (BITMAPFILEHEADER);

2) палітра (може бути відсутнім);

3) саме зображення.

На кожен піксель в різних файлах може припадати різну кількість біт (глибина кольору). Пропонуються бітності 1, 2, 4, 8, 16, 24, 32, 48 і 64. У бітності 8 і нижче, колір вказується індексом з таблиці кольорів (палітри), а при великих безпосереднім значенням. В таблиці 2.1 наведено структуру BMP заголовку.

Таблиця 2.1 – Структура BMP заголовку

Позиція в файлі (hex)	Розмір (байти)	Ім’я	Опис
0E	4	biSize	Розмір даної структури в байтах, який вказує так само на версію структури.
12	4	biWidth	Ширина растра в пікселях. Вказується цілим числом зі знаком. Нуль і від’ємні не задокументовані.
16	4	biHeight	Висота растра в пікселях. Вказується цілим числом зі знаком. Нуль і від’ємні не задокументовані.
1A	2	biPlanes	У BMP допустимо тільки значення 1. Це поле використовується в значках і курсорах Windows.
1C	2	biBitCount	Кількість біт на піксель.
1E	4	biCompression	Вказує на спосіб зберігання пікселів.
22	4	biSizeImage	Розмір піксельних даних в байтах. Може бути обнулено якщо зберігання здійснюється двовимірним масивом.
26	4	biXPelsPerMeter	Кількість пікселів на метр по горизонталі і вертикалі.
2A	4	biYPelsPerMeter	
2E	4	biClrUsed	Розмір таблиці кольорів в комірках.
32	4	biClrImportant	Кількість комірок від початку таблиці кольорів до останньої використовуваної (включаючи її саму).

Зображення ділиться на дві частини: заголовок та дані. Дані використовується в блоку перестановки та заміни. Заголовок BMP зображення має фіксований розмір – 54 біти. Змінюються лише такі поля як: ширина (biWidth), висота (biHeight) і розмір файлу (biSizeImage). Модифікований заголовок зберігається для подальшого прикріплення до даних.

2.3 Формування контрольних байтів для кожного каналу RGB

Кожен піксель в зображенні формату BMP складається з трьох байт, кожен з яких відповідає каналу R (red), G (green) і B (blue). Для формування

контрольних байтів для кожного рядка і стовпця зображення можна зробити результуюче значення для кожного кольорового каналу за допомогою суми за модулем 256 (вираз 2.3).

$$[I_R, I_G, I_B] = [(IR_0 + IR_1 + \cdots + IR_{X-1})\ mod256, (IG_0 + IG_1 + \cdots + IG_{X-1})\ mod256, (IB_0 + IB_1 + \cdots + IB_{X-1})\ mod256] \quad (2.3)$$

де $[I_R, I_G, I_B]$ – результат контрольного пікселя з перевірочними значеннями для кожного з каналів, X – довжина рядка або стовпця.

Наприклад, при значеннях [R = 120, G = 211, B = 45] і [R = 233, G = 88, B = 145], значення пікселя для кожного каналу буде: [(120 + 233) mod 256, (211 + 88) mod 256, (45 + 145) mod 256] = [97, 43, 190]. Таким чином можна сформувати двовимірний масив контрольних байтів просумувавши значення всіх каналів у всіх рядка і стовпцях за модулем 256 (рис. 2.2).

0 … *N*

[120, 211, 45]	[154, 205, 98]	[56, 137, 57]	[74, 41, 200]
[23, 105, 175]	[68, 93, 100]	[168, 230, 154]	[3, 172, 173]
[255, 134, 80]	[78, 5, 62]	[33, 21, 188]	[110, 160, 74]
[142, 194, 44]	[44, 47, 4]	[1, 132, 143]	[187, 117, 191]

M

Рисунок 2.2 – Зображення з контрольними байтами для кожного каналу

Альтернативою є варіант формування контрольних байтів за результуючим значенням кожного пікселя.

2.4 Формування контрольних байтів за значенням пікселів

На відмінну від формування контрольних байтів для кожного каналу можна зробити цю перевірку за результуючими значеннями пікселів.

Результуючим значенням для пікселя буде сума всіх байтів кольорових каналів за модулем 256 (вираз 2.4).

$$I_P = (I_R + I_G + I_B)\ mod\ 256 \quad (2.4)$$

Наприклад, при значеннях R = 120, G = 211, B = 45, значення пікселя буде: (120 + 211 + 45) mod 256 = 120. Таким чином можна сформувати двовимірний масив контрольних байтів просумувавши значення пікселів у всіх рядка і стовпцях (рис. 2.3).

0	(120 (R) + 211 (G) + 45 (B)) mod 256 = 120	(154 + 205 + 98) mod 256 = 201	(56 + 137 + 57) mod 256 = 250	(120 + 201 + 250) mod 256 = 59
	(23 + 105 + 175) mod 256 = 47	(68 + 93 + 100) mod 256 = 5	(168 + 230 + 154) mod 256 = 40	(47 + 5 + 40) mod 256 = 92
	(255 + 134 + 80) mod 256 = 213	(78 + 5 + 62) mod 256 = 145	(33 + 21 + 188) mod 256 = 242	(213 + 145 + 242) mod 256 = 88
M	(120 + 47 + 213) mod 256 = 124	(201 + 5 + 145) mod 256 = 95	(250 + 40 + 242) mod 256 = 20	239 (байти парності)

N

Рисунок 2.3 – Вигляд зображення з контрольними байтами за значенням пікселів

Цей метод є менш надлишковим, ніж контроль для кожного кольорового каналу. Але він також є менш ефективним, оскільки тільки вказує на зіпсований піксель, але контрольні байти для кожного кольорового каналу вказують і виправляють цей піксель.

2.5 Процес формування перестановок та замін

Під перестановкою розуміється зміна розташування елементів даних. Одна ітерація перестановки називається раундом.

Блоку перестановки та заміни необхідні наступні дані: довжина початкового файлу, ключ (K), початкові стани регістрів з лінійним зворотнім зв'язком та незвідні поліноми.

Оскільки секретом є зображення, тому необхідно відокремити заголовок, а

над іншими даними проводити операції (рис. 2.4).

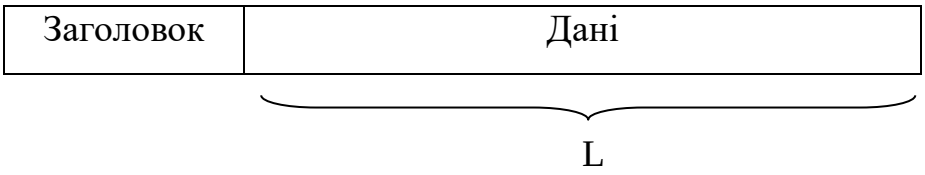


Рисунок 2.4 – Структура зображення

Усі елементи пронумеровані від 0 до L - 1. Ідея перестановки полягає в тому, щоб зчитувати елементи за номерами, які є псевдовипадковим числами від 0 до L - 1, і записувати ці елементи в природному порядку (рис. 2.5).

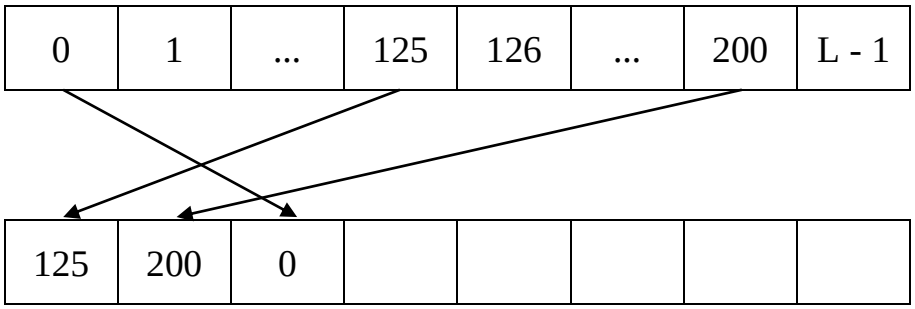


Рисунок 2.5 – Процес перестановки

Для формування псевдовипадкових чисел пропонується розбити діапазон чисел від 0 до L - 1 на чотири піддіапазони, кожен з яких характеризується мінімальним і максимальним значенням чисел. Вибір із піддіапазонів здійснюється природнім порядком, шляхом збільшення або зменшення значення. А вибір піддіапазона здійснюється на основі псевдовипадкової послідовності нулів і одиниць.

Для реалізації перестановки необхідно спочатку розбити L на 4 частини. Кожна частина має свій індекс, за яким буде визначатись, з якої частини брати елемент (рис 2.6).

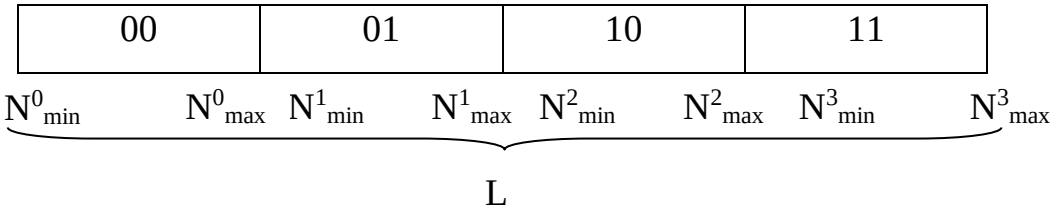


Рисунок 2.6 – Вигляд розподілених частин

Для вибору номера піддіапазона пропонується використовувати коди, що формуються регістром зсуву з лінійним зворотнім зв'язком. А для вибору чисел із піддіапазона пропонується використовувати лічильники.

Для того, щоб задати початкові значення лічильників необхідно мати ключ, що складається з біт. Кожен біт ключа відповідає певному піддіапазону. Якщо біт дорівнює нулю, тоді початковим значенням відповідного лічильника є мінімальне значення чисел піддіапазону. Якщо біт дорівнює одиниці, тоді значенням лічильника є максимальне значення чисел піддіапазону.

Ключ (K) має розмір в 4 біти та визначається першими бітами першого стану регістра. Він потрібен для того, щоб визначити правила зміни лічильників в кожній частині розподіленого файлу. Якщо значення біта ключа рівне 0, то лічильник збільшується, якщо 1 – зменшується (табл. 2.2).

Таблиця 2.2 – Початкові стани лічильників

Розряди кодів ключа				Початкові значення			
K_0	K_1	K_2	K_3	Ліч. 0	Ліч. 1	Ліч. 2	Ліч. 3
0	0	0	0	N^0_{min}	N^1_{min}	N^2_{min}	N^3_{min}
0	0	0	1	N^0_{min}	N^1_{min}	N^2_{min}	N^3_{max}
0	0	1	0	N^0_{min}	N^1_{min}	N^2_{max}	N^3_{min}
0	0	1	1	N^0_{min}	N^1_{min}	N^2_{max}	N^3_{max}
0	1	0	0	N^0_{min}	N^1_{max}	N^2_{min}	N^3_{min}
0	1	0	1	N^0_{min}	N^1_{max}	N^2_{min}	N^3_{max}
0	1	1	0	N^0_{min}	$N1_{max}$	N^2_{max}	N^3_{min}
0	1	1	1	N^0_{min}	N^1_{max}	N^2_{max}	N^3_{max}
1	0	0	0	N^0_{max}	N^1_{min}	N^2_{min}	N^3_{min}
1	0	0	1	N^0_{max}	N^1_{min}	N^2_{min}	N^3_{max}
1	0	1	0	N^0_{max}	N^1_{min}	N^2_{max}	N^3_{min}
1	0	1	1	N^0_{max}	N^1_{min}	N^2_{max}	N^3_{max}
1	1	0	0	N^0_{max}	N^1_{max}	N^2_{min}	N^3_{min}
1	1	0	1	N^0_{max}	N^1_{max}	N^2_{min}	N^3_{max}
1	1	1	0	N^0_{max}	N^1_{max}	N^2_{max}	N^3_{min}
1	1	1	1	N^0_{max}	N^1_{max}	N^2_{max}	N^3_{max}

З урахуванням цього лічильники можуть збільшувати або зменшувати свій стан. А отже, той же самий секретний ключ визначає правила функціонування лічильників. Відповідно для кожного лічильника буде встановлено початкове значення (табл. 2.3).

Таблиця 2.3 – Правила функціонування лічильників

Розряди кодів ключа				Ліч. 0	Ліч. 1	Ліч. 2	Ліч. 3
K_0	K_1	K_2	K_3				
0	0	0	0	+1	+1	+1	+1
0	0	0	1	+1	+1	+1	-1
0	0	1	0	+1	+1	-1	+1
0	0	1	1	+1	+1	-1	-1
0	1	0	0	+1	-1	+1	+1
0	1	0	1	+1	-1	+1	-1
0	1	1	0	+1	-1	-1	+1
0	1	1	1	+1	-1	-1	-1
1	0	0	0	-1	+1	+1	+1
1	0	0	1	-1	+1	+1	-1
1	0	1	0	-1	+1	-1	+1
1	0	1	1	-1	+1	-1	-1
1	1	0	0	-1	-1	+1	+1
1	1	0	1	-1	-1	+1	-1
1	1	1	0	-1	-1	-1	+1
1	1	1	1	-1	-1	-1	-1

З кожного стану регістра зсуву з лінійним зворотнім зв'язком береться по два біти для визначення, з якої частини файлу брати елемент даних.

Для прикладу, виберемо довжина $L = 256$, ключ $K = \{1, 0, 0, 1\}$, згенерований стан регістру 001011000101. На основі вхідних параметрів формуються початкові значення лічильника $N = \{63, 64, 128, 255\}$.

Таблиця 2.4 – Приклад формування послідовностей псевдовипадкових чисел

Стани	Значення	Операція	Результат
00	63	63 - 1	62
10	128	128 + 1	129
11	255	255 - 1	254
00	62	62 - 1	61
01	64	64 + 1	65
01	65	65 + 1	66
11	254	254 - 1	253
10	129	129 + 1	130

Перед тим, як записати елемент його необхідно замінити. Заміна відбувається за допомогою операції XOR ($\oplus$) зі значенням 8-бітного регістра зсуву з лінійним зворотнім зв'язком (вираз 2.5).

$$c_i = m_i \oplus g_i \tag{2.5}$$

Наприклад, якщо взято елемент з певної позиції зі значенням 45, а значення 8-бітного регістра 233, то на виході буде отримано значення 196 ($45 \oplus 233 = 196$). При зворотних обчислення для того, щоб отримати початкове значення необхідно виконати цю ж операцію: $196 \oplus 233 = 45$.

2.6 Побудова регістра зсуву з лінійним зворотним зв’язком

Двійкові псевдовипадкові періодичні послідовності, що генеруються з використанням регістрів зсуву з лінійним зворотним зв’язком, називаються РЗЛЗЗ-послідовностями або лінійними рекурентними послідовностями. Основними перевагами цих генераторів є [16]:

– LFSR добре втілюються на апаратному рівні;

– вони можуть утворювати послідовності із великими періодами;

– вони можуть утворювати послідовності з хорошими статистичними властивостями;

– завдяки своїй будові, вони легко піддаються аналізу за допомогою алгебраїчних технік.

Від певних комірок регістра робляться відводи, або точки знімання, вміст

цих комірок додається за модулем 2, а сума повертається в першу комірку регістра зсуву. На рисунку 2.7 $q_i \in \{0,1\}$ позначають наявність або відсутність точки відводу у даної комірки. Символ $\oplus$ позначає операцію сума за mod 2.

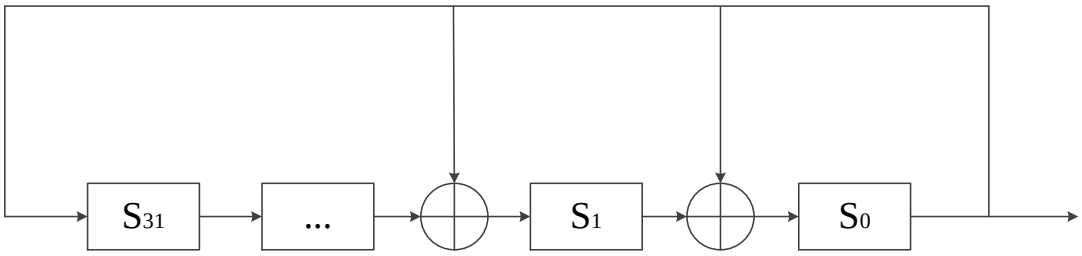


Рисунок 2.7 – 32-розрядний регістр зсуву з лінійним зворотнім зв’язком

N-бітовий LFSR може згенерувати псевдовипадкову послідовність з періодом $2^n - 1$ бітів. Такі LFSR називаються регістрами максимального періоду [16].

Для цього регістр зсуву повинен побувати в усіх $2^n - 1$ ненульових внутрішніх станах.

Функції зворотного зв’язку регістра можна поставити у відповідність поліном $m(x)$ степеня не більше n з коефіцієнтами з поля лишків за модулем два, що складається з одночленів вигляду $x^{n_i - 1}$, де $\{n_i\}$ – множина номерів точок знімання зворотного зв’язку.

Поліном $m(x)$ називається мінімальним поліномом відповідної рекурентної послідовності.

Для кожної кінцевої (або періодичної) послідовності *S* можна вказати LFSR, який, при деякому початковому заповненні, породжує цю послідовність.

Серед всіх таких регістрів існує регістр мінімальної довжини *L*. Величина *L* називається лінійною складністю послідовності *S*. Поліном називається незвідним, якщо він не може бути виражений як добуток двох поліномів меншого степеня, відмінних від констант.

Примітивний поліном степеня *n* над полем лишків за модулем два – це незвідний поліном, який ділить $x^{2^n - 1} - 1$, але не ділить $x^d - 1$ для будь-яких

$d : d \mid 2^{n-1}$.

Список практично застосовуваних примітивних поліномів наведений в [17]. Наприклад, примітивним поліномом є $x^8 + x^6 + x^5 + x^3 + 1$. Набір показників (8, 6, 5, 3, 0) означає, що взявши регістр зсуву довжиною 8 і генеруючи біт зворотного зв'язку шляхом додавання 6-го, 5-го, 3-го і 0-го бітів за модулем 2, ми отримаємо LFSR максимальної довжини ($2^8 - 1$ станами) (рис. 2.8).

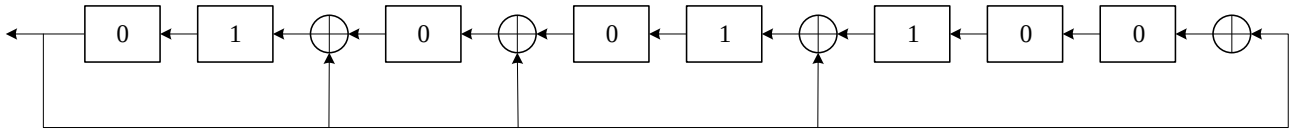


Рисунок 2.8 – 8-бітний LFSR

Послідовні біти лінійної рекуренти лінійно залежні, що робить їх непотрібними для шифрування. Досить *2n* послідовних бітів рекуренти, щоб визначити множину номерів точок знімання зворотного зв'язку.

2.7 Метод відновлення секрету

Для реалізації методу відновлення секрету пропонується послідовність етапів зображених на рисунку 2.9.

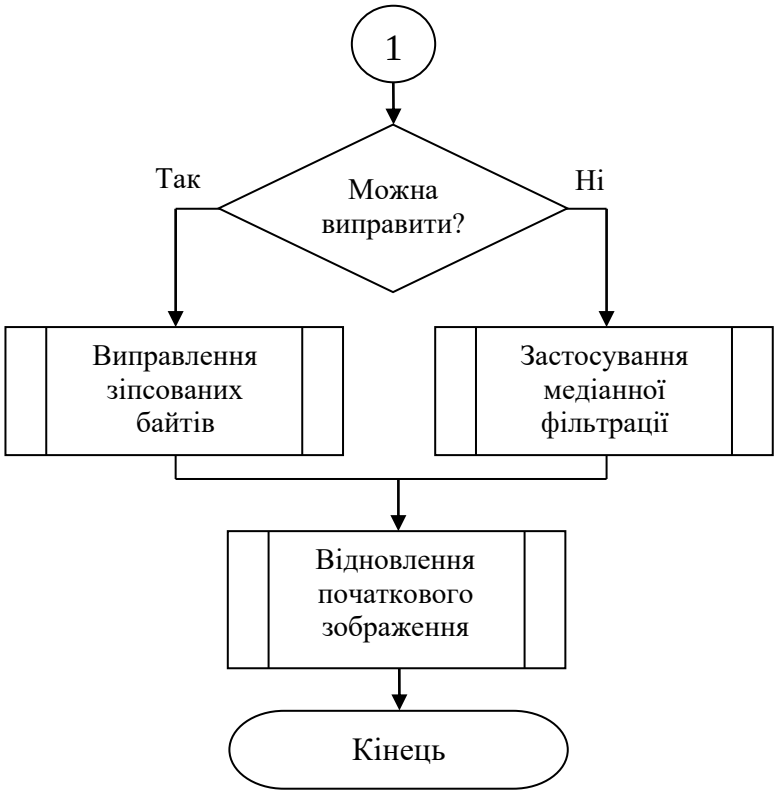


Рисунок 2.9 – Послідовність етапів методу відновлення секрету

Процес відновлення секрету передбачає такі кроки. Для початку необхідно об'єднати складові частини усіх учасників в єдиний файл. Потім застосувати процес формування перестановки та заміни для того, щоб розташувати замішані пікселі по своїм початковим позиціям. Для цього необхідно провести зворотні від розподілу дії. Зчитувати елементи потрібно в природному порядку (послідовно) та записувати їх, в попередньо сформований масив розміром від 0 до L - 1, в коміркою за адресою псевдовипадкового числа (рис. 2.10).

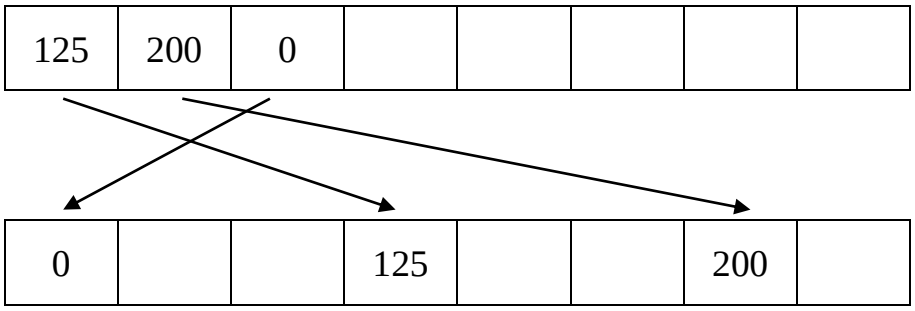


Рисунок 2.10 – Зворотній процес перестановки

Наступний крок передбачає формування контрольних байтів для

відновленого початкового файлу для всіх пікселів по рядкам та стовпцям. Нові сформовані контрольні байти необхідно порівняти з тими значеннями, які були обраховані перед розподілом та визначити позиції зіпсованих пікселів.

Оскільки використовується двовимірний контроль за модулем (додається контроль для всіх рядків та стовпців), то перетини неправильних рядків і стовпців вкажуть на позиції зіпсованих байтів.

Виправлення визначених зіпсованих байтів відбувається за виразом:

$$x_i = (K_R - x_0 - x_1 - \ \dots - x_{i-1} - x_{i+1} - x_{N-1})\ mod\ 256 \qquad (2.6)$$

де x_i – зіпсований байт, K_R – значення контрольного байту, $\{x_0, x_{1, \dots,} x_{i-1}, x_{i+1}, x_{N-1}\}$ – значення байтів рядка чи стовпця.

На рисунку 2.11 наведено приклад знаходження зіпсованого пікселя.

224	128	89	185
101	***247***	9	***99***
84	198	205	231
153	***59***	47	

Рисунок 2.11 – Зображення із зіпсованими байтами

Якщо провести обрахунок контролю за модулем, то в другому рядку і другому стовпці значення не будуть збігатися із записаними при розподілі секрету. Тому їх перетин і вказує на зіпсований байт. Тоді до цього байту необхідно застосувати обрахунки наведені вище. Наприклад, після перевірок комірка зі значенням 247 виявилась зіпсованою, тому правильне значення обраховується таким чином: (256 + 99 - 101 - 9) mod 256 = 245 mod 256 = 245 (виправлення по рядку) або (256 + 59 - 128 - 198) mod 256 = -11 mod 256 = -11 + 256 = 245 (виправлення по стовпцю).

Також можлива ситуація коли є тільки зіпсований рядок або стовпець, тоді для всіх байтів в цьому рядку або стовпці необхідно застосувати медіанну фільтрацію. Медіанна фільтрація передбачає вирахування середнього

арифметичного значення для зіпсованих каналів байтів на основі значень всіх байтів, що оточують його. Навіть якщо виправлене значення не буде відповідати початковому, для людського ока це не буде помітним. Оскільки в зображенні сусідні пікселі близькі між собою за значеннями кольорових каналів. На рисунку 2.12 наведено приклад для виправлення каналу R для другого рядку.

187 (0, 0)	180 (0, 1)	183 (0, 2)	179 (0, 3)	217 (0, 4)
186 **(1, 0)**	***200*** **(1, 1)**	**177** **(1, 2)**	***189*** **(1, 3)**	***215*** **(1, 4)**
185 (2, 0)	*175* (2, 1)	193 (2, 2)	*194* (2, 3)	*235* (2, 4)
191	*184*	190	*182*	*235*
237	*227*	*231*	*232*	

Рисунок 2.12 – Зображення із зіпсованими байтами каналу R

Отже, необхідно застосувати медіанну фільтрацію до каналу R кожного пікселя. Оскільки перший у першого байту відсутні декілька оточуючих елементів, то вони дублюються тими, що знаходяться поруч. Вигляд байту для якого застосовується медіанна фільтрація наведено на рисунку 2.13.

$x_{i-1, j-1}$	$x_{i-1, j}$	$x_{i-1, j+1}$
$x_{i, j-1}$	$x_{i, j}$	$x_{i, j+1}$
$x_{i+1, j-1}$	$x_{i+1, j}$	$x_{i+1, j+1}$

Рисунок 2.13 – Оточення пікселя $x_{i, j}$

Значення кольорового каналу пікселя обчислюється за виразом:

$$K_S = \frac{x_{i-1,j-1} + x_{i-1,j} + x_{i-1,j+1} + x_{i,j-1} + x_{i,j} + x_{i,j+1} + x_{i+1,j-1} + x_{i+1,j} + x_{i+1,j+1}}{9} \qquad (2.8)$$

де K_S – результат медіанної фільтрації, x – елемент, що оточує зіпсований байт.

Для елемента з позицією [1, 0] (перший елемент): ([0,-1] + [0,0] + [0,1] + [1,-1] + [1,0] + [1,1] + [2,-1] + [2,0] + [2,1]) / 9 = (187 + 187 + 180 + 186 + 186 + 200 + 185 + 185 + 175) / 9 = 185. Елементи [0,-1], [1,-1] та [2,-1] є дублюючими, оскільки в елемента [1,0] немає сусідів зліва. Для елемента з позицією [1, 1] (другий елемент) результат для каналу R буде: ([0,0] + [0,1] + [0,2] + [1,0] + [1,1] + [1,2] + [2,0] + [2,1] + [2,2]) / 6 = (187 + 180 + 183 + 186 + 185 + 177 + 185 + 175 + 193) / 9 = 183. Аналогічним чином для віх інших елементів потрібно провести медіанну фільтрацію.

Отже, запропонований метод завадостійкого розподілу секрету поєднує в собі дві переваги: формування контрольних байтів у вигляді двовимірної перевірки для рядків та стовпців, що виявляють та виправляють помилки та застосування медіанної фільтрації для рядків або стовпців, які не вдалось виправити.

3 ПРОГРАМНИЙ ЗАСІБ ДЛЯ РОЗПОДІЛУ СЕКРЕТУ

3.1 Вибір програмних засобів

Для реалізації програмного засобу обрано мову JavaScript та фреймворк Node.js. Вбудовані бібліотеки підтримують роботу з файлами будь-яких типів. JavaScript дозволяє легко будувати необхідний додаток, інші види компонентів, досить просто зберігати й одержувати інформацію з бази даних й інших сховищ даних.

Однією з основних привабливих особливостей Node.js є швидкість. JavaScript-код, що виконується в середовищі Node.js, може бути в два рази швидше, ніж код, написаний на компільованих мовах, на зразок C або Java, і на порядки швидше інтерпретованих мов на зразок Python або Ruby. Причиною подібного є неблокуюча архітектура платформи, а конкретні результати залежать від використовуваних тестів продуктивності, але, в цілому, Node.js - це дуже швидка платформа.

Платформа Node.js призначена для виконання високопродуктивних мережевих застосунків, написаних мовою програмування JavaScript. Платформа окрім роботи із серверними скриптами для веб-запитів, також використовується для створення клієнтських та серверних програм.

Середовищем розробки було обрано Visual Studio Code, яке надає досить легкий підхід і широкі можливості в розробці додатків на JavaScript, швидкодію компільованого коду, низьку навантаженість на систему і зручний набір плагінів, для максимальної зручності роботи з кодом.

3.2 Структура програмного засобу

Розробка програмного засобу буде виконуватись блоками, що є зручним при будь-яких модифікаціях та змінах програми.

Програмний засіб складається з таких елементів: регістрів зсуву з лінійним зворотнім зв'язком (8- та 32-розрядного), модуля роботи з файлами, модуль

формування контрольних байтів, модуль перестановки та заміни, блоку розподілу на частини та блоку відновлення.

Загальний алгоритм роботи програми наведено на рисунку 2.1.

3.2.1 Програмна реалізація регістра зсуву з лінійним зворотнім зв'язком

Регістр зсуву з лінійним зворотним зв’язком виконую роль генератора бітів. На рис. 3.1 зображено алгоритм роботи регістру. Для роботи необхідно два регістри довжиною 8 та 32 біти. В основі регістрів лежить примітивні поліноми в яких період максимальний, для 8 бітного $2^8 - 1 = 255$ і для $2^{32} - 1 = 4294967295$ відповідно, та початкові стани які задаються вручну.

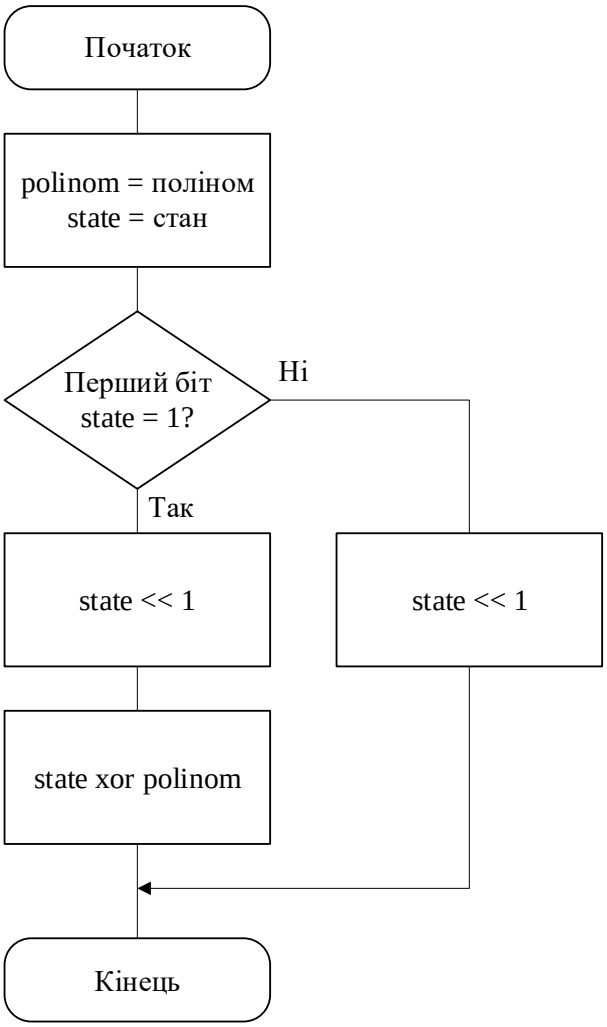


Рисунок 3.1 – Алгоритм роботи регістр зсуву

Код одного з регістрів має такий вигляд:

```
getState32(state, polynomial) {
        if ((state & 0x80000000) != 0) {
            state = (state << 1) ^ polynomial;
        } else {
            state = state << 1;
        }
        return state;
}
```

Відповідним чином формуються отримуються значення і для 8-розрядного регістру.

3.2.2 Програмна реалізація процесу формування перестановок та замін

Перестановки та заміни складаються з декількох частин. Спочатку задається початкове значення для ключа *K*, що відповідає за напрямки в лічильниках. Якщо значення дорівнює 1, тоді лічильник зменшується, якщо 0 – збільшується. Потім визначаються початкові стани лічильників та записуються в об'єкт *counters* з відповідним ідентифікатором.

Дані зберігаються у вигляді об'єкта *parts*. Оскільки дані розбиті на чотири піддіапазони, то кожен з них має свій ідентифікатор: 00, 01, 10 та 11 відповідно. Ідентифікатор *code* визначається двома молодшими бітами кожного стану 32-розрядного регістра зсуву з лінійним зворотнім зв'язком.

Код взяття одного елемента з піддіапазону має такий вигляд:

```
function getElement(code) {
    if (counters[code] !== undefined) {
        const index = counters [code];
        mixedData.push(parts[code][index]);
        index = key[code] ? (index + 1) : (index - 1);
        if (index < 0 || index >= parts[code].length) {
            counters [code] = undefined;
        } else {
            counters [code] = index;
        }
    }
}
```

На рис. 3.2 зображено алгоритм формування перестановок та замін.

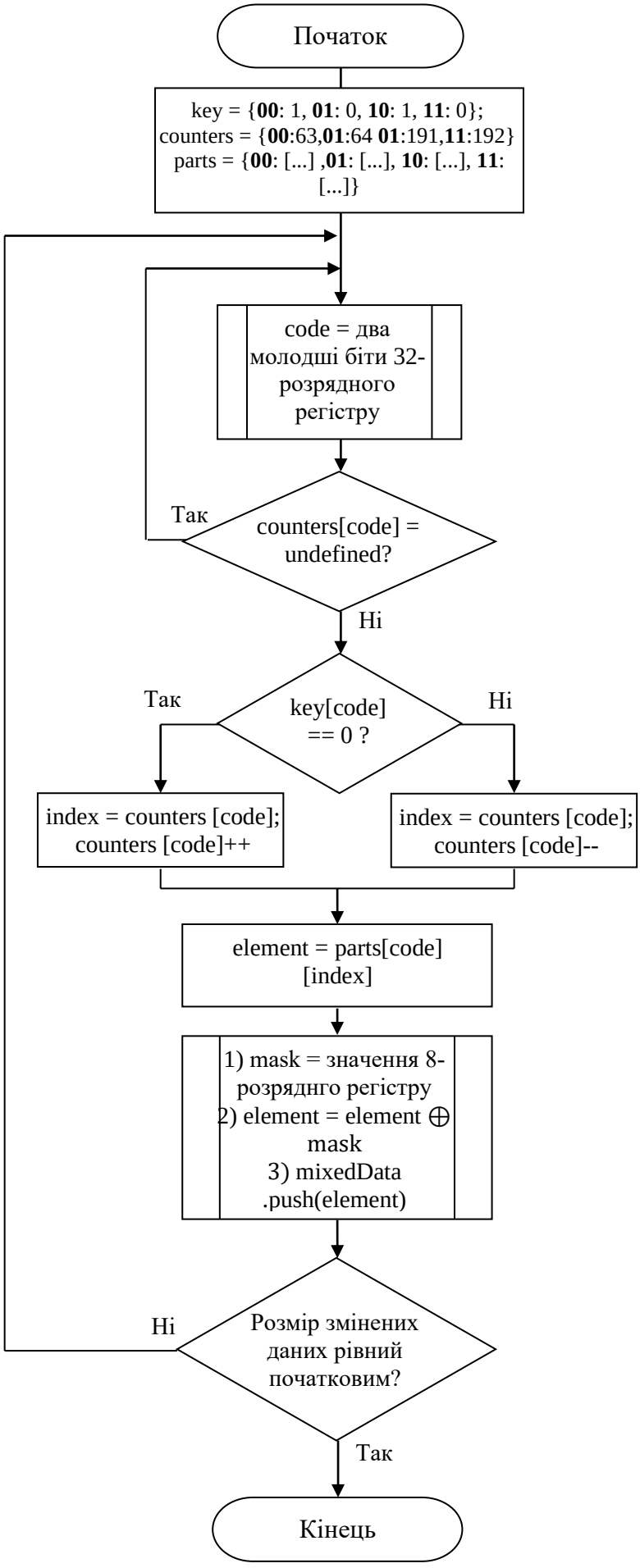


Рисунок 3.2 – Алгоритм формування перестановок та замін

3.2.3 Програмна реалізація формування контрольних байтів

Оскільки зображення має вигляд двовимірного масиву розміром M x N, то формування контрольних байтів відбувається для всіх каналів кожного пікселя по рядкам та стовпцям.

Код формування контрольного байту має такий вигляд:

```
function createControlByte(data) {
    const controlByte = new Array(data[0].length).fill(0);
    data.forEach((pixel) => {
        pixel.forEach((byte, key) => {
            controlByte[key] = (controlByte[key] + byte) % 256;
        });
    });
    return controlByte;
}
```

Після формування контрольних байтів розмір зображення збільшиться на один рядок та стовпець відповідно.

Під час відновлення зображення необхідно зробити перевірку на зіпсовані байти. Для цього необхідно для відновленого зображення ще раз перерахувати контрольні байти та порівняти їх із байтами, які були записані при розподілі зображення.

Код знаходження зіпсованих байтів має такий вигляд:

```
function checkControlByte(data) {
    const newControlByte = new Array(data[0].length).fill(0);
    const oldControlByte = data[data.length - 1];
    const badBytes = [];
    for (let i = 0; i < data.length - 1; i++) {
        const pixel = data[i];
        pixel.forEach((byte, key) => {
            newControlByte[key] = (newControlByte[key] + byte) % 256;
        });
    }
    newControlByte.forEach((byte, key) => {
        if (byte !== oldControlByte[key]) {
            badBytes.push(key);
        }
    });
}
```

При перерахуванні контрольних байтів не враховуються останній рядок та стовпець відновленого зображення, оскільки з ними відбувається порівняння.

3.2.4 Програмна реалізація медіанної фільтрації

Можлива ситуація, коли після відновлення початкового зображення та перевірці контрольних байтів, будуть окремі зіпсовані рядки або стовпці. Точне розташування зіпсованого пікселя виявити при цьому неможливо, тому необхідно для всього рядка або стовпця провести медіанну фільтрацію.

Код медіанної фільтрації має такий вигляд:

```
function getNeighborhood(positionX, positionY, data) {
    return [
        data[positionX - 1, positionY - 1],
        data[positionX - 1, positionY],
        data[positionX - 1, positionY + 1],
        data[positionX, positionY - 1],
        data[positionX, positionY],
        data[positionX, positionY + 1],
        data[positionX + 1, positionY - 1],
        data[positionX + 1, positionY],
        data[positionX + 1, positionY + 1]
    ];
}

function filtration(rowOrColumn, isRow, position, data) {
    rowOrColumn.forEach((pixel, key) => {
        const positionX = isRow ? position : key;
        const positionY = isRow ? key : position;
        const neighborhood = getNeighborhood(positionX, positionY, data);
        let sum = null;
        neighborhood.forEach((item) => {
            sum += item;
        });
        return (sum / neighborhood.length);
    });
}
```

Зрозуміло, що це наближені значення для елементів, але зображення сприймається людиною, то деякі помилкові пікселі можуть бути непоміченими.

3.3 Тестування програмного засобу

Для роботи програми необхідно зображення типу BMP та глибиною бітів 24. Таке зображення представлено на рисунку 3.3. Для перетворення будь-якого зображення в дані параметри, можна використати вбудовану програму для редагування Paint.

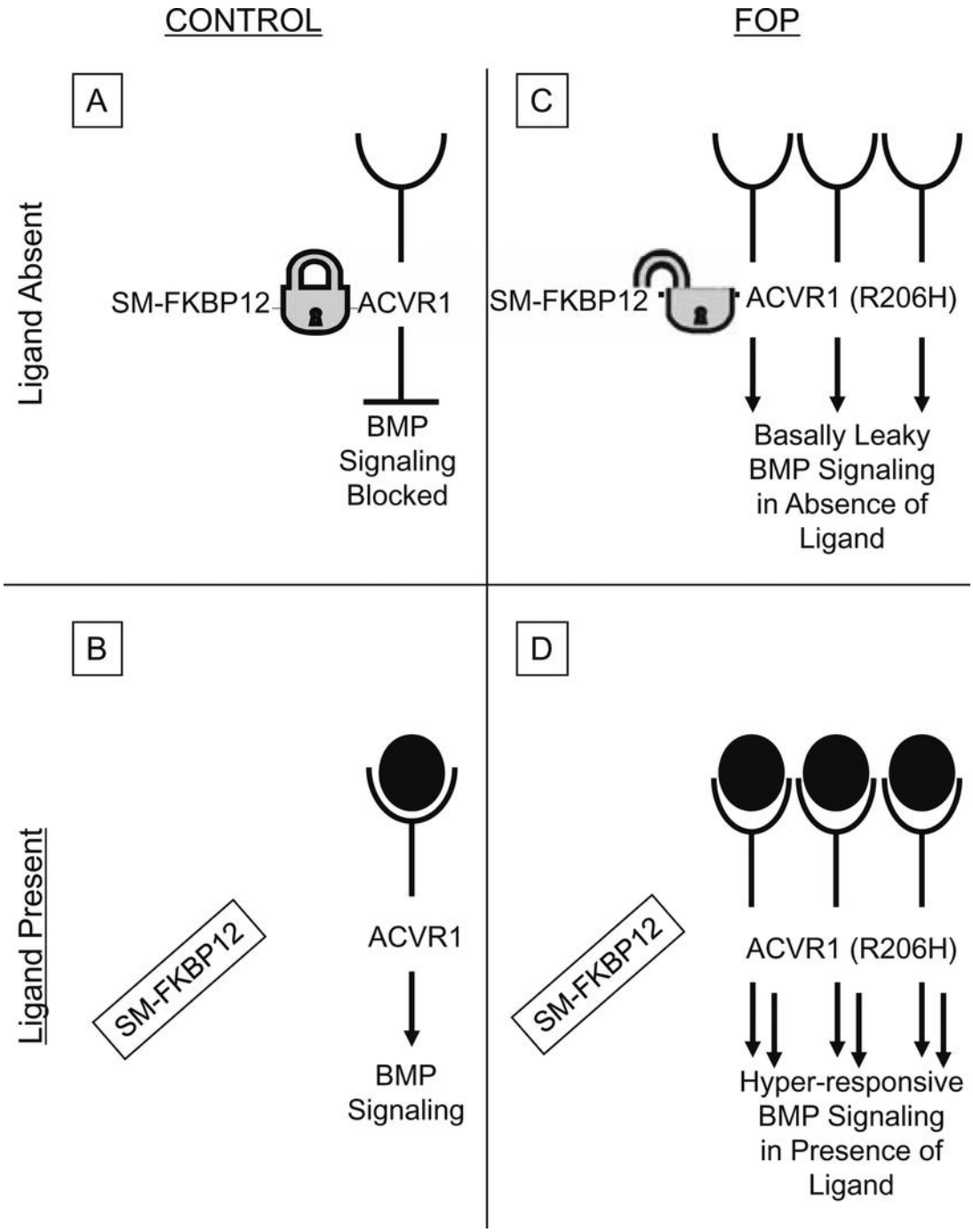


Рисунок 3.3 – Обране зображення

Після вибору зображення та запуску за допомогою терміналу командою: *node index.js separation name(ім'я файл)*, виконується розділення його на три частини. З модуля регістра зсуву надходять випадкові послідовності біт до модулю перестановки та заміни. На виході отримується 3 файли: firstImage.bmp, secondImage.bmp та thirdImage.bmp.

Результат роботи програми зображено на рисунку 3.4. Необхідно зауважити, що отримані «шуми» залежать від регістрів зсуву та перетворених байтів, які виконуються під час формування перестановок та замін.

	image1
	image2

	image3

Рисунок 3.4 – Розподілене зображення

Для відновлення початкового зображення необхідно за допомогою терміналу виконати наступну команду: командою: *node index.js union firstImage secondImage thirdImage*. Як результат буде отримано початкове зображення (див. рис. 3.3).

Відновлення зображення можливе тільки при наявності всіх трьох частин. Але якщо раптом двоє будь-яких користувачів будуть намагатись відновити початкове зображення, то в них виникнуть проблеми під час цього. Оскільки дані частини, якої не вистачає будуть замінені на будь-яке задане в застосунку значення. Тобто ц вся частина буде в одній кольоровій гамі, яка може находитись в діапазоні від 0 до 255.

Після відновлення двох правильних частин і однієї заміненої, до зображення також застосується виправлення за контрольними байтами та медіанні фільтрація, тому можлива ситуація, коли це дасть більш чіткий вигляд для цього зображення.

Вигляд зображення при відновленні двома учасниками наведено на рисунку 3.5 .

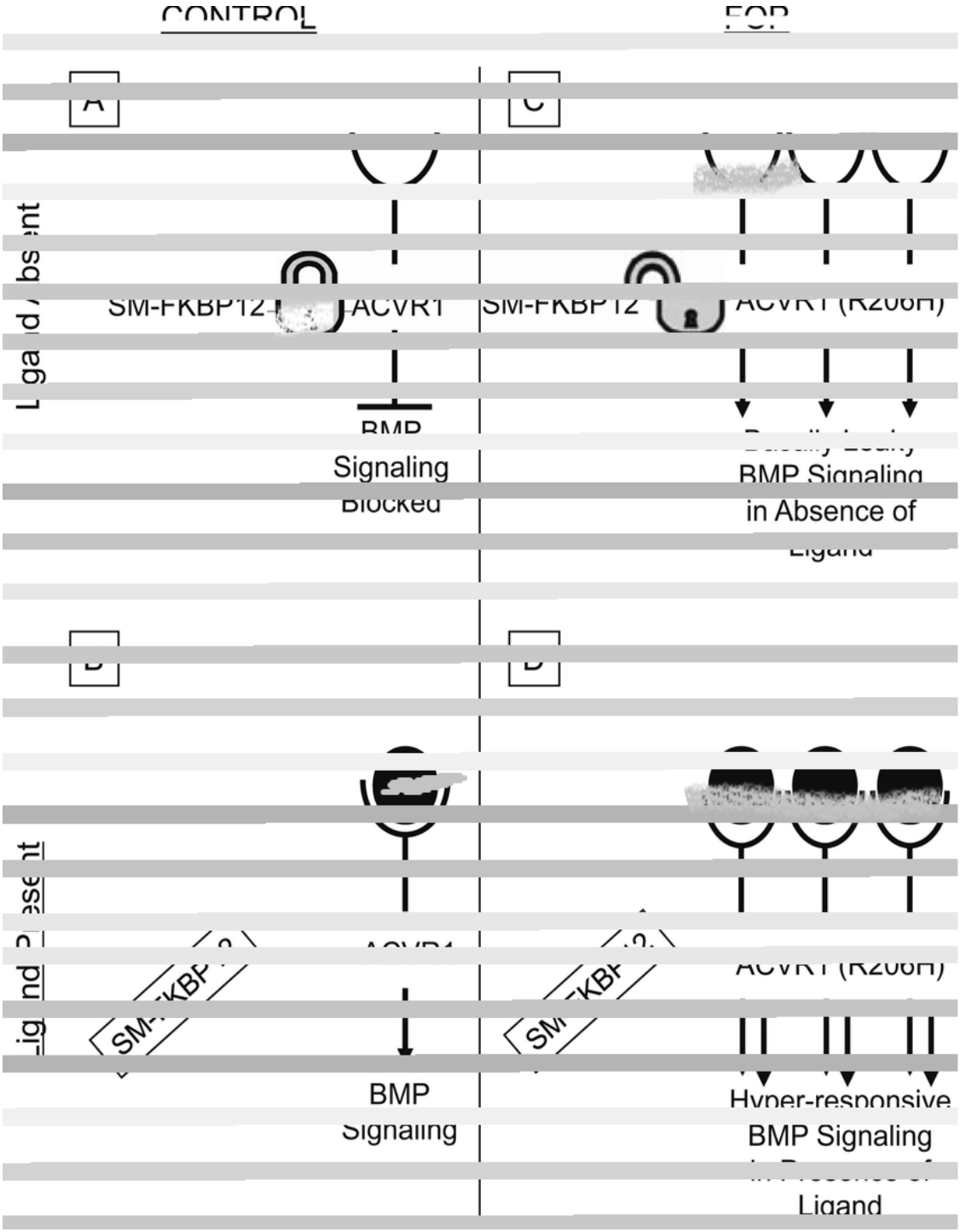


Рисунок 3.5 – Вигляд зображення при відновленні двома учасниками

Перед відновленням на будь-яку частину зображення можна накласти завади та перевірити правильність роботи двовимірної перевірки на парність. Для накладання шумів в коді програми задається, який відсоток від розміру даних необхідно зіпсувати і потім на згенеровані випадковим чином позиції пікселів накладається гама.

Код накладання шумі на зображення має наступний вигляд:

```
function random(min, max) {
    let rand = min + Math.random() * (max + 1 - min);
    return Math.floor(rand);
}

getState8(state, polynomial) {
      if ((state & 0x80) != 0) {
            state = (state << 1) ^ polynomial;
      } else {
            state = state << 1;
      }
      return state;
}

function createGama(data) {
    const percent = 0.2;
    const countFoGama = parseInt(data.length * percent);
    const usedPositions = [];
    while (countFoGama > 0) {
        const position = random(0, data.length);
        if (!usedPositions.includes(position)) {
            usedPositions.push(position);
            data[position] = data[position] ^ getState8(state, polynomial);
            countFoGama--;
        }
    }
}
```

Вигляд частини зображення після накладання завад наведено на рисунку 3.6.

Рисунок 3.6 – Вигляд зображення після накладання завад

Після відновлення початкове зображення має наступний вигляд (рис. 3.7).

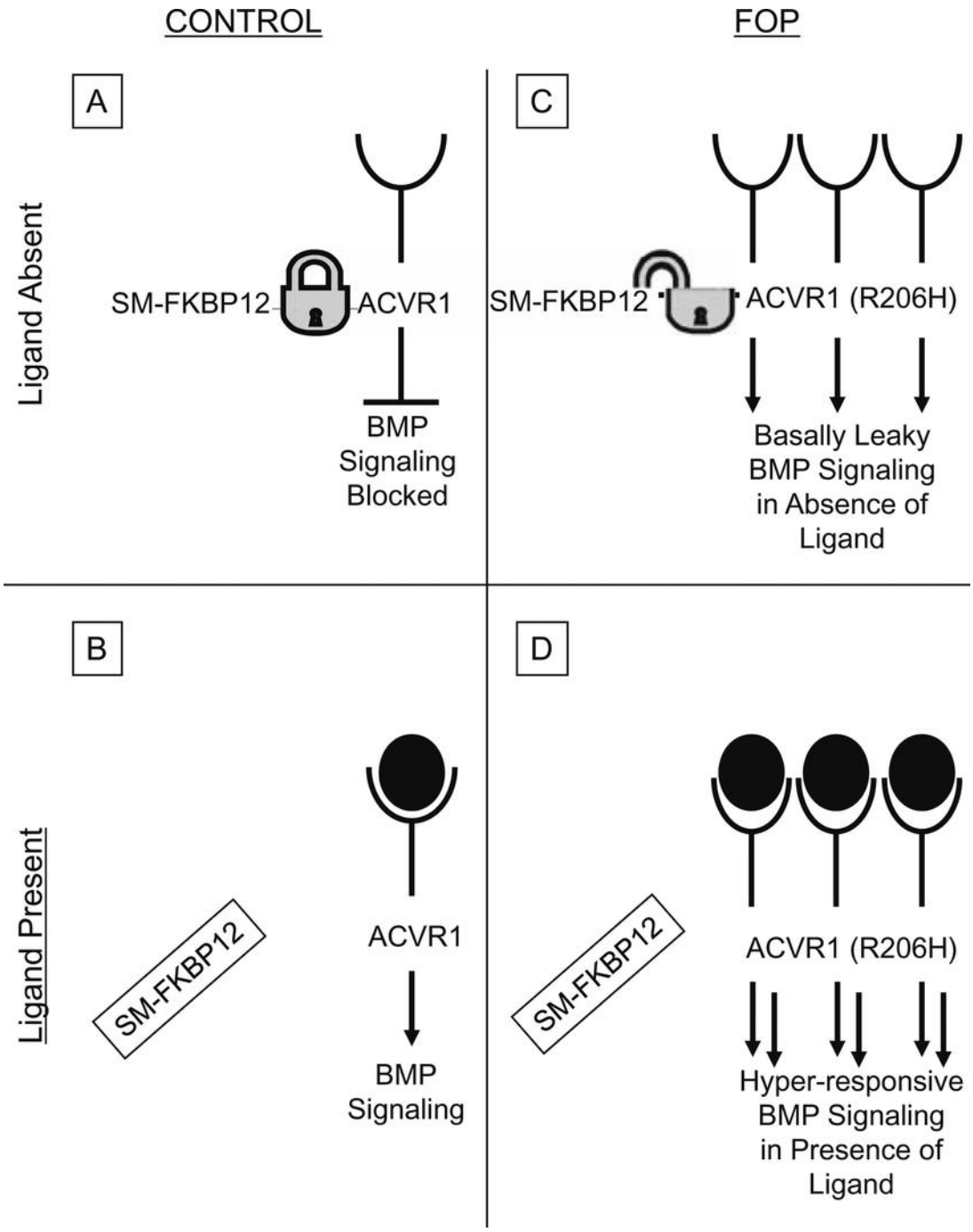


Рисунок 3.7 – Вигляд зображення після відновлення із завадами

Вигляд відновленого зображення не відрізняється від початкового. Можливо і є пошкоджені пікселі, але для людського зору вони непомітні.

Отже, проаналізувавши роботу програмного засобу не було виявлено ніяких помилок. Програма працює коректно, розподіляє та відновлює зображення правильно, збоїв та зависань під час роботи не дає.

4 ЕКОНОМІЧН ЧАСТИНА

4.1 Аналіз комерційного потенціалу розробки (технологічний аудит розробки) методу та засобу завадостійкого розподілу секрету

4.1.1 Визначення рівня комерційного потенціалу розробки методу та засобу завадостійкого розподілу секрету

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки методу та засобу завадостійкого розподілу секрету, створеної в результаті науково-технічної діяльності. В результаті оцінювання можна буде зробити висновок щодо напрямів (особливостей) організації подальшого її впровадження з врахуванням встановленого рейтингу.

Для проведення технологічного аудиту залучимо 3-х незалежних експертів. У нашому випадку такими експертами будуть керівник магістерської роботи та провідні викладачі випускової та споріднених кафедр.

Оцінювання комерційного потенціалу розробки методу та засобу завадостійкого розподілу секрету будемо здійснювати за 12-ю критеріями згідно рекомендацій.

Результати оцінювання комерційного потенціалу розробки методу та засобу завадостійкого розподілу секрету заносимо до таблиці 4.1.

Таблиця 4.1. - Результати оцінювання комерційного успіху розробки методу та засобу завадостійкого розподілу секрету

Критерії	Експерти		
	д. т. н., проф., Лужецький В.А.	к.т.н., ст. викл Лукічов В. В.	к.т.н., доцент Войтович О.П.
	Бали, виставлені експертами		
1	2	2	2
2	3	1	3
3	2	2	3
4	3	1	2
5	3	2	3

Продовження таблиці 4.1

6	2	2	2
7	3	2	3
8	2	2	2
9	3	2	1
10	3	3	3
11	2	2	3
12	3	3	2
Сума балів	31	24	29
Середньоарифметична сума балів, СБ	28		

За даними таблиці 4.1 робимо висновок щодо рівня комерційного потенціалу розробки методу та засобу завадостійкого розподілу секрету. При цьому користуємося рекомендаціями, наведеними в таблиці 4.2.

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 50	Високий

Таким чином, робимо висновок, щодо рівня комерційного потенціалу нашої розробки методу та засобу завадостійкого розподілу секрету – середній.

4.1.2 Визначення рівня якості розробки методу та засобу завадостійкого розподілу секрету

Оцінювання рівня якості розробки методу та засобу завадостійкого розподілу секрету проводиться з метою порівняльного аналізу і визначення

найбільш ефективного, з технічної точки зору, варіанта інженерного рішення.

Рівень якості – це кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Абсолютний рівень якості розробки методу та засобу завадостійкого розподілу секрету знаходимо обчисленням вибраних для її вимірювання показників, не порівнюючи їх із відповідними показниками аналогічних виробів. Для цього необхідно визначити зміст основних функцій, які повинні реалізовувати розробка, вимоги замовника до неї, а також умови, які характеризують експлуатацію, визначають основні параметри, які будуть використані для розрахунку коефіцієнта технічного рівня виробу. Система параметрів, прийнята до розрахунків, повинна достатньо повно характеризувати споживчі властивості інноваційного товару (його призначення, надійність, економічне використання ресурсів, стандартизація тощо).

Далі визначаємо величину параметрів якості в балах та встановлюємо граничні його значення (кращі, гірші, середні). Всі ці дані для кожного параметра заносимо в табл. 4.3.

Таблиця 4.3 – Основні параметри методу та засобу завадостійкого розподілу секрету

Параметри	Абсолютне значення параметра			Коефіцієнт вагомості параметра
	Краще +5…+4	Середнє +3	Гірше +1…+2	
Швидкодія	4			0,2
Ресурсозатратність		3		0,3
Якість відновлюваного файлу	4			0,4
Розмір ключа шифрування			2	0,1

Із врахуванням коефіцієнтів вагомості відповідних параметрів можна

визначити абсолютний рівень якості інноваційного рішення за формулою:

$$К_{я.а.} = \sum_{i=1}^{n} Рні \cdot аі, \quad (4.1)$$

де Рні – числове значення і-го параметра інноваційного рішення, n – кількість параметрів інноваційного рішення, що прийняті для оцінювання, аі – коефіцієнт вагомості відповідного параметра (сума коефіцієнтів вагомості всіх параметрів повинна дорівнювати 1).

Отже, абсолютний рівень якості методу та засобу завадостійкого розподілу секрету становитиме – 3,5 бали.

Одночасно визначаємо відносний рівень якості методу та засобу завадостійкого розподілу секрету, що виробляється (проектується), порівнюючи її показники з абсолютними показниками якості найліпших вітчизняних та зарубіжних аналогів (товарів-конкурентів) (табл. 4.4).

Таблиця 4.4 – Основні параметри методу та засобу завадостійкого розподілу секрету та товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Швидкодія	4	2	2	0,2
Ресурсозатратність	5	2	2,5	0,3
Якість відновлюваного файлу	60-100%	80-100%	1,3	0,4
Розмір ключа шифрування	32 біт	64 біт	0,5	0,1

Відносний рівень якості методу та засобу завадостійкого розподілу секретувизначаємо за формулою:

$$К_{я.в.} = \sum_{i=1}^{n} qі \cdot аі, \quad (4.2)$$

За розрахунками відносний рівень якості методу та засобу завадостійкого розподілу секрету становитиме – 1,72. Це означає, що наша розробка краща за якістю на 72% від товару-аналога.

4.1.3 Визначення конкурентоспроможності розробки методу та засобу завадостійкого розподілу секрету

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками, що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має відповідати критеріям оцінювання споживачів конкретного ринку в конкретний час.

Дані для розрахунку загального показника конкурентоспроможності розробки необхідно занести до таблиці 4.5.

Таблиця 4.5 – Нормативні, технічні та економічні параметри методу та засобу завадостійкого розподілу секрету і товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Швидкодія	4	2	2	0,2
Ресурсозатратність	5	2	2,5	0,3
Якість відновлюваного файлу	60-100%	80-100%	1,3	0,4
Розмір ключа шифрування	32 біт	64 біт	0,5	0,1
Ціна за продукт, тис. грн.	15000	18000	0,83	-

Загальний показник конкурентоспроможності розробки (К) з урахуванням вищезазначених груп показників визначаємо за формулою:

$$К = \frac{Іт.п.}{Іе.п.} = \frac{1,72}{0,83} = 2,07, \quad (4.3)$$

де Іт.п. – індекс технічних параметрів (відносний рівень якості інноваційного рішення); Іе.п. – індекс економічних параметрів.

$$Іе.п. = \frac{РНеі}{РБеі} = \frac{15000}{18000} = 0,83, \quad (4.4)$$

де РНеі, РБеі – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

Згідно розрахунків загальний показник конкурентоспроможності –2,07 . Це означає, що наша розробка методу та засобу завадостійкого розподілу секретубільш конкурентна на 107% від товару-аналога.

4.2 Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи

4.2.1 Розрахунок витрат, що стосуються виконавців розробки методу та засобу завадостійкого розподілу секрету

Основна заробітна плата кожного із розробників (дослідників) Зо, якщо вони працюють в наукових установах бюджетної сфери:

$$Зо = \frac{М}{Тр} \cdot t, \quad (4.5)$$

де М – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.

У 2019 році величини окладів (разом з встановленими доплатами і надбавками) рекомендується брати в межах (5000…10000) грн. за місяць; Тр – число робочих днів в місяці; приблизно Тр = (21…23) дні; t – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 2.1.

Таблиця 2.1 – Заробітна плата розробників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	30000	1429	5	7145
Інженер-програміст	20000	952	5	4760
Всього:				11905

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Зр = \sum_{i=1}^{n} ti \cdot Ci, \quad (4.6)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою:

$$Ci = \frac{Мм \cdot Ki}{Тр \cdot Тзм}, \quad (4.7)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 2.2 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Налагоджувальні	3	7	38	1,54	114
Складальні	2	4	31,5	1,27	63
Всього					177

Додаткова заробітна плата Зд всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми

основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = 0{,}1 \cdot (Зр + Зо) = 0{,}1 \cdot (11905 + 177) = 1208{,}2 \text{ грн.} \quad (4.8)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$Нзп = 0{,}22 \cdot (Зр + Зо + Зд) = 0{,}22 \cdot (11905 + 177 + 1208{,}2) = \\ = 2924 \text{ грн.} \quad (4.9)$$

Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому бути розраховані за формулою:

$$А = \frac{Ц \cdot На}{100} \cdot \frac{Т}{12},$$

де Ц – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; На – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що На = (10...25)%; Т – термін, використання обладнання, приміщень тощо, місяці.

Таблиця 2.3 - Амортизаційні відрахування

Найменування	Ціна, грн.	Норма амортизації, %	Термін використання, м.	Сума амортизації
ПК	10000	20	4	667
Всього	667			

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$Ве = В \cdot П \cdot Ф \cdot Кп, грн$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 8,45 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп< 1.

Потужність обладнання складає –0,5 кВт.

Кількість годин роботи складає – 700годин.

Коефіцієнт викор. потужності -0,9.

Ве=2662 грн.

Інші витрати Він охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати Ів можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$Ів = 1 \cdot (Зо + Зр) = 1 \cdot (11905 + 177) = 12082 \text{ грн.} \qquad (4.10)$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В.

$$В = 31625 \text{ грн.}$$

4.2.2 Розрахунок собівартості розробки методу та засобу завадостійкого розподілу секрету

Витрати на силову електроенергію Ве, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

$$Ве = В \cdot П \cdot Ф \cdot Кп, грн$$

В – вартість 1 кВт-год. електроенергії, в 2019 р. В ≈ 8,45 грн./кВт; П – установлена потужність обладнання, кВт; Ф – фактична кількість годин роботи обладнання, годин, Кп – коефіцієнт використання потужності; Кп< 1.

Потужність обладнання складає – 0,5 кВт.

Кількість годин роботи складає – 700 годин.

Коефіцієнт викор. потужності -0,9.

Ве=2662 грн.

Основна заробітна плата робітників Зр, якщо вони беруть участь у виконанні даного етапу роботи і виконують роботи за робочими професіями у випадку, коли вони працюють в наукових установах бюджетної сфери, розраховується за формулою:

$$Зр = \sum_{i=1}^{n} ti \cdot Ci, \quad (4.11)$$

де ti – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; Сі – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. Сі визначається за формулою:

$$Ci = \frac{Мм \cdot Ki}{Тр \cdot Тзм}, \quad (4.12)$$

де Мм – розмір мінімальної заробітної плати за місяць, грн.; в 2019 році мінімальна заробітна плата становить – 4173 грн., Кі – тарифний коефіцієнт робітника відповідного розряду, Тр – число робочих днів в місяці; приблизно Тр = 21…23 дні; Тзм – тривалість зміни, зазвичай Тзм = 8 годин.

Таблиця 2.2 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Налагоджувальні	3	7	38	1,54	114
Складальні	2	4	31,5	1,27	63
Всього					177

Додаткова заробітна плата Зд всіх робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = 0{,}1 \cdot (Зо) = 0{,}1 \cdot (177) = 17{,}7 \text{ грн.} \quad (4.13)$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

де Зо – основна заробітна плата розробників, грн.; Зр – основна заробітна плата робітників, грн.; Зд – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

$$Нзп = 0{,}22 \cdot (Зо + Зд) = 0{,}22 \cdot (177 + 17{,}7) = \\ = 42{,}8 \text{ грн.} \quad (4.14)$$

«Загальновиробничі витрати» належать витрати: пов'язані з управлінням виробництвом (утримання працівників апарату управління виробництвом, оплата службових відряджень персоналу цехів, витрати на інформаційне забезпечення управління тощо); на повне відновлення та капітальний ремонт основних фондів загальновиробничого призначення; витрати некапітального характеру, пов'язані з удосконаленням технологій та організацією виробництва, поліпшенням якості продукції; на утримання, обслуговування, поточний ремонт виробничих приміщень; на контроль за виробничими процесами та кістю продукції.

Крім того, загальновиробничі витрати з розрахунку на одиницю продукції можна розрахувати за нормативами відносно до основної заробітної плати основних робітників, які виготовляють продукцію:

$$ЗВВ = Нв \cdot Зо, \quad (4.15)$$

Норматив загальновиробничих витрат для програмних продуктів становить 230-270%.

$$ЗВВ = 2{,}3 \cdot 177 = 407{,}1 \text{ грн,}$$

Сума попередніх витрат утворює виробничу собівартість розробки:

$$Sв = 3307 \text{ грн.}$$

4.3 Розрахунок мінімальної ціни та чистого прибутку від реалізації розробки

Ціна – це грошовий вираз вартості товару (продукції, послуги). Вона завжди коливається навколо ціни виробництва (перетвореної форми вартості одиниці товару, що дорівнює сумі витрат виробництва й середнього прибутку) та відображає рівень суспільне необхідних витрат праці.

Виходячи з того, що розробки, як правило, приймаються та впроваджуються за завданням замовника, або коли результатом розробки є продукція, що підлягає державному регулюванню, то нижню межу ціни реалізації розробки можна розрахувати за формулою:

$$Ц = S_В \cdot \left(1 + \frac{Р}{100}\right) \cdot \left(1 + \frac{\omega}{100}\right), \quad (4.16)$$

де Sв – виробнича собівартість інноваційного рішення, грн.; Р – норматив рентабельності узгоджений із замовником або встановлений державою, (Р=30…60%); w – ставка податку на додану вартість, % (в 2019 році w=20%).

$$Ц = 3307 \cdot \left(1 + \frac{60}{100}\right) \cdot \left(1 + \frac{20}{100}\right) = 6349 \text{ грн.}$$

Чистий прибуток від реалізації розробки можна розрахувати за формулою:

$$П = \left(Ц - \frac{(Ц - МР) \cdot f}{100} - S_В - \frac{q \cdot S_В}{100}\right) \cdot \left(1 - \frac{h}{100}\right) \cdot РП, \quad (4.17)$$

де Ц – ціна розробки, грн.; МР – вартість матеріальних та інших ресурсів, що були придбані виробником для виготовлення розробки (МР=(0,1…0,2) Цр), грн.; f – зустрічна ставка податку на додану вартість, %; Sв – виробнича собівартість розробки, грн.; q – норматив, який визначає величину адміністративних витрат, витрат на збут та інші операційні витрати, % (рекомендовано q=5…10%); h – ставка податку на прибуток, %, РП – прогнозований попит продажів:

$$П = 13870 \text{ грн.}$$

4.4 Розрахунок терміну окупності коштів, вкладених в наукову розробкуметоду та засобу завадостійкого розподілу секрету

Термін окупності вкладених у реалізацію наукового проекту інвестицій Ток можна розрахувати за формулою:

$$\text{Ток} = \frac{\text{В}}{\text{П}} = \frac{31625}{13870} = 2{,}3 \text{ роки.} \quad (4.18)$$

Оскільки Ток < 3 років, то фінансування даної наукової розробки методу та засобу завадостійкого розподілу секрету є доцільним.

ВИСНОВКИ

Аналіз інформаційних джерел показав, що існують різні варіанти розподілу секрету. Кожний з видів має свої позитивні та негативні риси, різну швидкодію, надлишковість, здатність до відновлення.

В реальних умовах при зберіганні або передаванні завади можуть зіпсувати інформацію. Тому виникає необхідність в побудові завадостійкого розподілу секрету. Одним із підходів до виправлення помилок є використання ітеративних кодів, які передбачають контроль за модулем по рядках і стовпцях двовимірного подання інформації.

Однак є певні види помилок, які лише виявляються і не можуть бути виправлені. Оскільки зображення сприймається людиною, то деякі помилкові пікселі можуть бути непоміченими. Тому додатково пропонується покращувати зображення за рахунок медіанної фільтрації.

Виходячи з цього в роботі запропонований власний метод додавання контрольних байтів на основі двовимірної моделі зображення та медіанної фільтрації. Основна перевага цього методу полягає в тому, що додавання контрольних байтів для всіх кольорових каналів рядків та стовпців дає можливість визначати зіпсовані пікселі, а також виправляти їх. Медіанна фільтрація потрібна для того, щоб якщо не вдалось визначити точні зіпсовані пікселів, відкоригувати елементи в окремо зіпсованих рядках та стовпцях. Крім того, використання простих операцій при розподілі та відновленні забезпечує пришвидшення процесу.

Практичним результатом виконання є програмний засіб, реалізований мовою JavaScript та за допомогою фреймворка Node.js. Для розробки засобу підготовлено ряд схем і алгоритмів, реалізовано цілу низку окремих функцій для виконання конкретних операцій. Розроблений програмний засіб протестований з метою перевірки правильності роботи алгоритму. Застосунок може бути використаний для розподілу секрету зображень формату BMP. Але не складно переробити його і під розподіл будь-якої інформації.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.Лужецький В. А. Основи інформаційної безпеки: навч. посіб. / В. А. Лужецький, А. Д. Кожухівський, О. П. Войтович. - Вінниц. нац. техн. ун-т. - Вінниця: ВНТУ, 2013. - 220 с.

2. Шнайер Б. Прикладная криптография. / Б. Шнайер - М.: Изд-во Триумф, 2003. -816 с.

3. Shamir A. How to share a secret / A. Shamir. - Com. Of the ACM. - 1979. - Vol. 22, №11. - P.612-613.

4. Blakley G. R. Safeguarding cryptographic keys / G. R. Blakley. - Proc. Of AFIPS Nasional Computer Conference. -1979. - 48. - P.313-317.

5. Rivest R. Method for Obtaining Digital Signature and Public Key Cryptosystems. / R. Rivest, A. Shamir, L. Adleman. - Comm. of the ACM, vol. 21, Feb. 1978.

6. Feldman P. A practical scheme for non-interactive verifiable secret sharing / P. Feldman // Proc. 28th Annu. Symp. on Found. of Comput. Sci. 1987. p. 427-437.

7. Blakley G. R. Safeguarding cryptographic keys / G. R. Blakley. - Proc. Of AFIPS Nasional Computer Conference. -1979. - 48. - P.313-317.

8. Blakley G.R. Linear algebra approach to secret sharing schemes / G.R. Blakley, G.A. Kabatianskii // In Pre Proceedings of Workshop on Information Protection, 1993.

9. Спельников А. Б. Эллиптическая пороговая схема разделения секрета / А. Б. Спельников. // Вест. Сам. гос. техн. ун-та, серия Физ.-мат науки – 2009. – №1(18). – С.251 - 259.

10. Схема разделения секретной визуальной информации [Електронний ресурс]. – Режим доступа: https://habr.com/company/nordavind/blog/177355/ – Назва з екрану.

11. Лебеденко А. В. Усовершенствование протоколов передачи данных за счет применения визуальной криптографии / А. В. Лебеденко, Е. Е. Смычков, В.В. Шилин // Перспективы развития информационных технологий. – 2015. – № 24. – 171-176 с.

12. Peterson W. W. Error-Correcting Codes / Peter-son W.W., Weldon E.J. // The Massachusetts Institute of Technology, Second Edition – 1972 – P. 301-350.

13. Assmus E.F. Designs and Their codes / Assmus E.F., Key J.D. // Cambridge University Press – 1992. – P. 264-270.

14. Bose R. On A Class of Error Correcting Binary Group Codes / R. Bose, D.K. Ray-Chaudhuri // Information and Control V.3 – 1960. – P. 68-79.

15. Microsoft Windows Bitmap File Format Summary [Електронний ресурс]. – Режим доступа: https://www.fileformat.info/format/bmp/egff.htm – Назва з екрану.

16. Иванов М. А. Теория, применение и оценка качества генераторов псевдослучайных последовательностей / М. А. Иванов, И. В. Чугунков – М.: КУДИЦ – ОБРАЗ, 2003. С. 240.

17. Питерсон У. Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон. // М. : «МИР», 1976. – 594 с.

ДОДАТКИ

Додаток А

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра захисту інформації

ЗАТВЕРДЖУЮ

Завідувач кафедри ЗІ, д. т. н., проф.

_____ В. А. Лужецький

"____" _____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

"Метод і засіб завадостійкого розподілу секрету"

08-20.МКР.001.00.000 ТЗ

Розробив студент групи 1БС-18м

_____ Бевзюк А. М.

Керівник магістерської кваліфікаційної роботи
д. т. н., проф., зав.каф. ЗІ

_____ Лужецький В. А.

_____ _____ 2019 р.

Вінниця 2019

1 Найменування та область застосування

Метод і засіб завадостійкого розподілу секрету. Область застосування: криптографічний захист інформації.

2 Підстави для розробки

Розробка виконується на основі наказу № 254 ректора ВНТУ від 02.10.2019 р..

3 Мета та призначення

Метою магістерської кваліфікаційної роботи є підвищення якості відновлення секретного вмісту.

4 Джерела розробки

- Лужецький В. А. Основи інформаційної безпеки: навч. посіб. / В. А. Лужецький, А. Д. Кожухівський, О. П. Войтович. - Вінниц. нац. техн. ун-т. - Вінниця: ВНТУ, 2013. - 220 с.
- Шнайер Б. Прикладная криптография. / Б. Шнайер - М.: Изд-во Триумф, 2003. -816 с.
- Shamir A. How to share a secret / A. Shamir. - Com. Of the ACM. - 1979. - Vol. 22, №11. - P.612-613.
- Blakley G. R. Safeguarding cryptographic keys / G. R. Blakley. - Proc. Of AFIPS Nasional Computer Conference. -1979. - 48. - P.313-317.
- Лебеденко А. В. Усовершенствование протоколов передачи данных за счет применения визуальной криптографии / А. В. Лебеденко, Е. Е. Смычков, В.В. Шилин // Перспективы развития информационных технологий. – 2015. – № 24. – 171-176 с.

5 Технічні вимоги

5.1 Кольорове зображення.
5.2 Формат зображення – bmp;
5.3 Розмір зображення – довільний;
5.4 Кількість частин для розбиття – 3;
5.5 Висока якість відновленого зображення.

6 Стадії та етапи розробки

№	Зміст	Початок	Закінчення	Результат
1	Вступ. Розробка ТЗ. Огляд літературних джерел	01.09.2019	22.09.2019	Розділ пояснювальної записки, технічне завдання

2	Розробка системи, моделі, алгоритму, структури	23.09.2019	12.10.2019	Розділ пояснювальної записки
3	Програмна реалізація	14.10.2019	17.11.2019	Модулі програмного засобу
	Тестування засобу	18.11.2019	24.11.2019	Діюча програма. Пояснювальна записка
4	Аналіз виконання ТЗ, висновки. Оформлення пояснювальної записки	25.11.2019	30.11.2019	Пояснювальна записка

7 Порядок контролю та приймання

7.1 До приймання дипломної роботи представляється:

- ПЗ до магістерської кваліфікаційної роботи;
- програмний засіб розподілу секрету;
- результати тестування;
- ілюстративні матеріали для захисту.

7.2 Рубіжний контроль керівника _____

7.3 Попередній захист на кафедрі _____

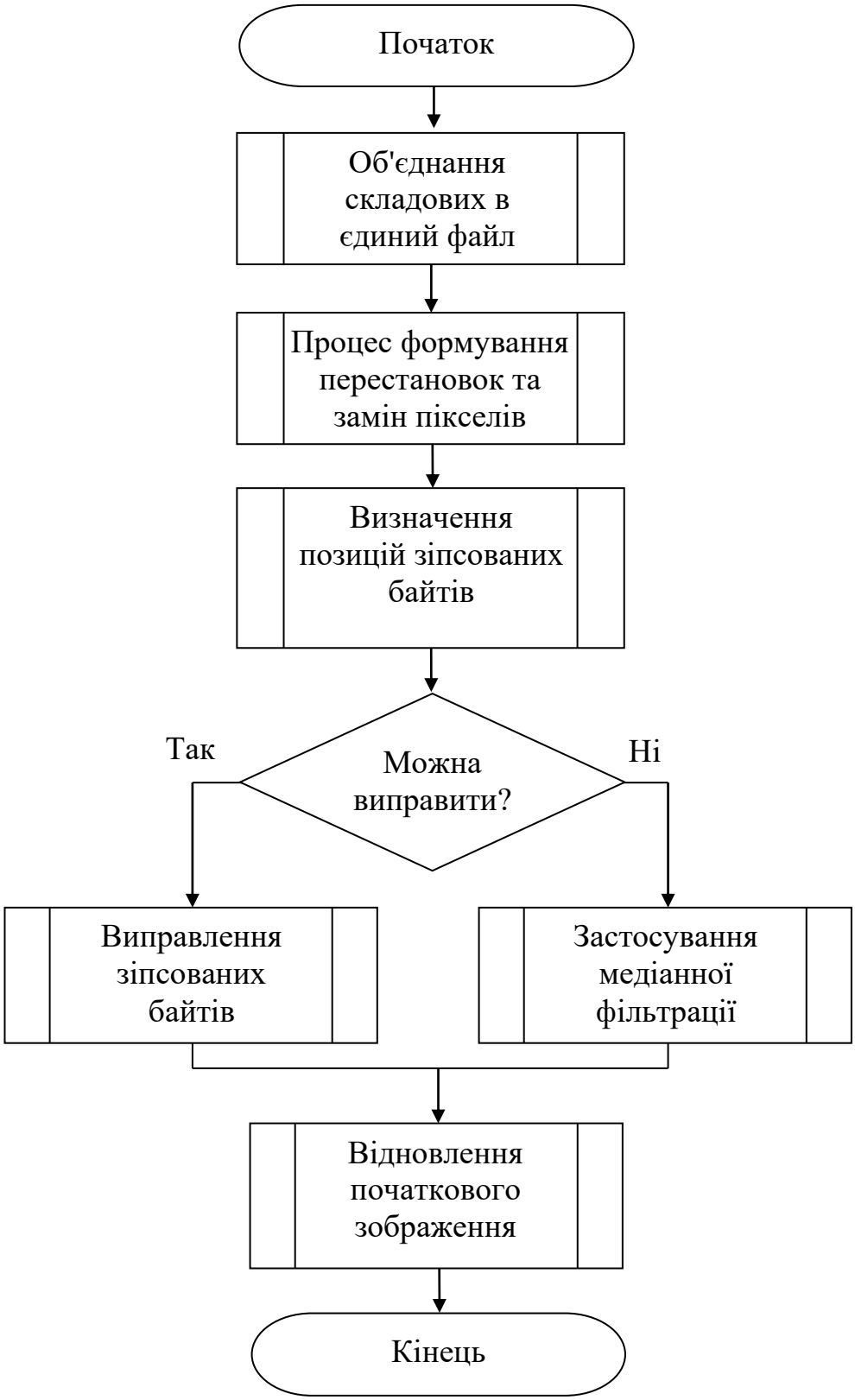
7.4 Захист на ДЕК _____

ІЛЮСТРАТИВНА ЧАСТИНА

Послідовність етапів методу розподілу секрету

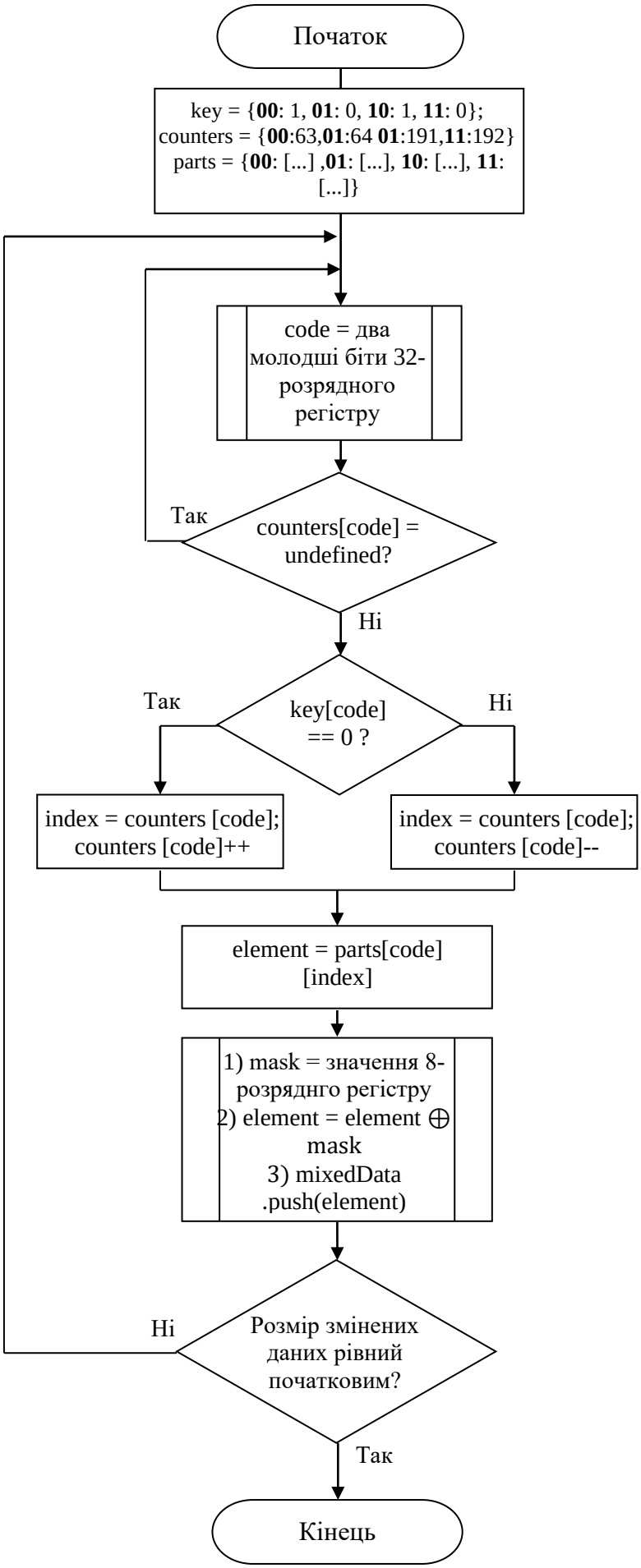
					08-20.МКР.001.00.000 ІЧ1		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Послідовність етапів методу розподілу секрету	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Послідовність етапів методу відновлення секрету



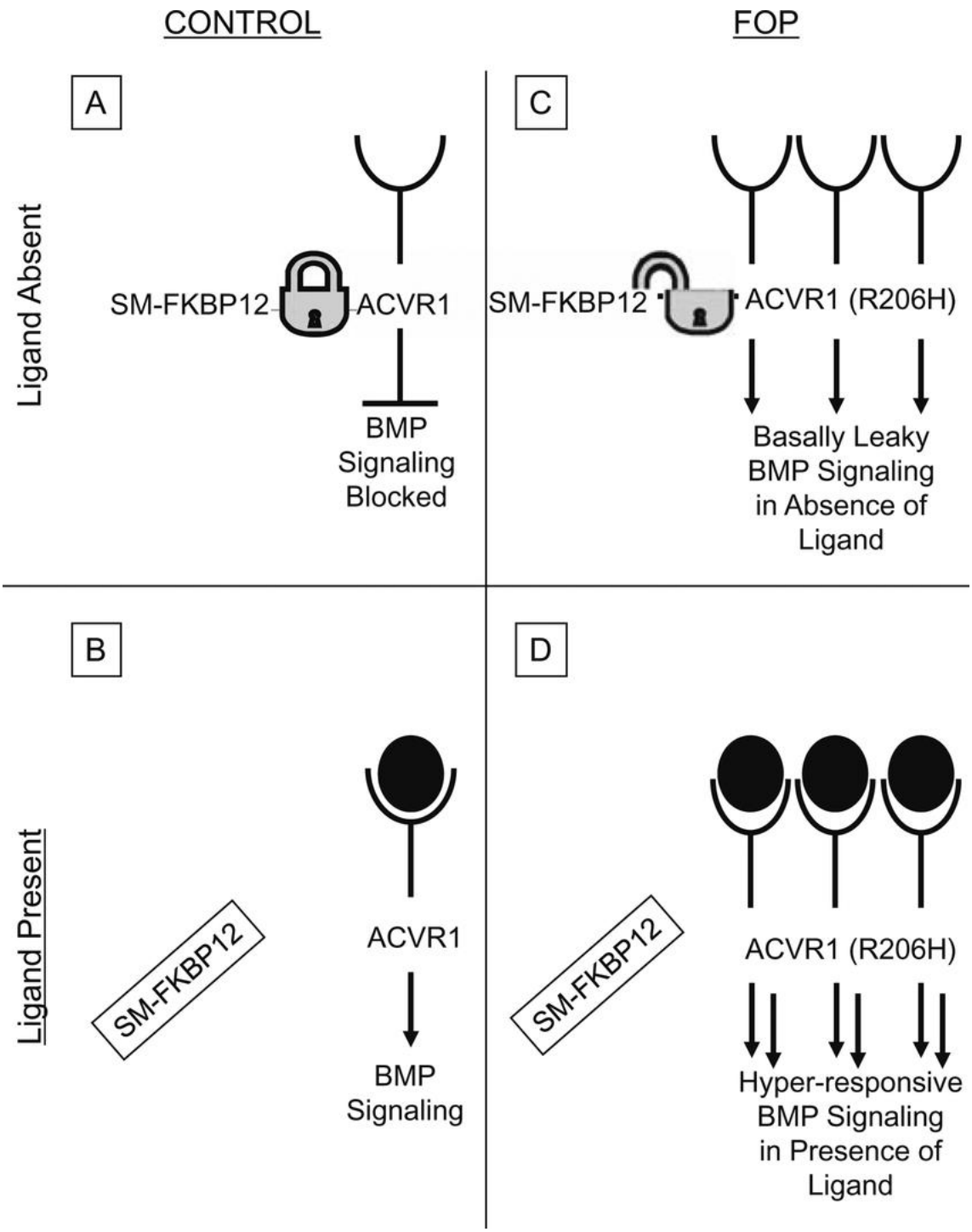
					08-20.МКР.001.00.000 ІЧ2			
Змн	Арк.	№ докум.	Підпис	Дата				
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Послідовність етапів методу відновлення секрету	Літ.	Маса	Масштаб
Перевір.		Лужецький В.А..						
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м		
Н. Контр.		Лужецький В.А.						
Затверд.		ЛужецькийВ.А.						

Алгоритм формування перестановок та замін



					08-20.МКР.001.00.000 ІЧЗ		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Алгоритм формування перестановок та замін	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Вигляд секретного зображення



					08-20.МКР.001.00.000 ІЧ4		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Вигляд секретного зображення	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Вигляд частин розподіленого секрету

	image1
	image2
	image3

					08-20.МКР.001.00.000 ІЧ5		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Вигляд частин розподіленого секрету	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Формування контрольних байтів

Для кожного каналу RGB

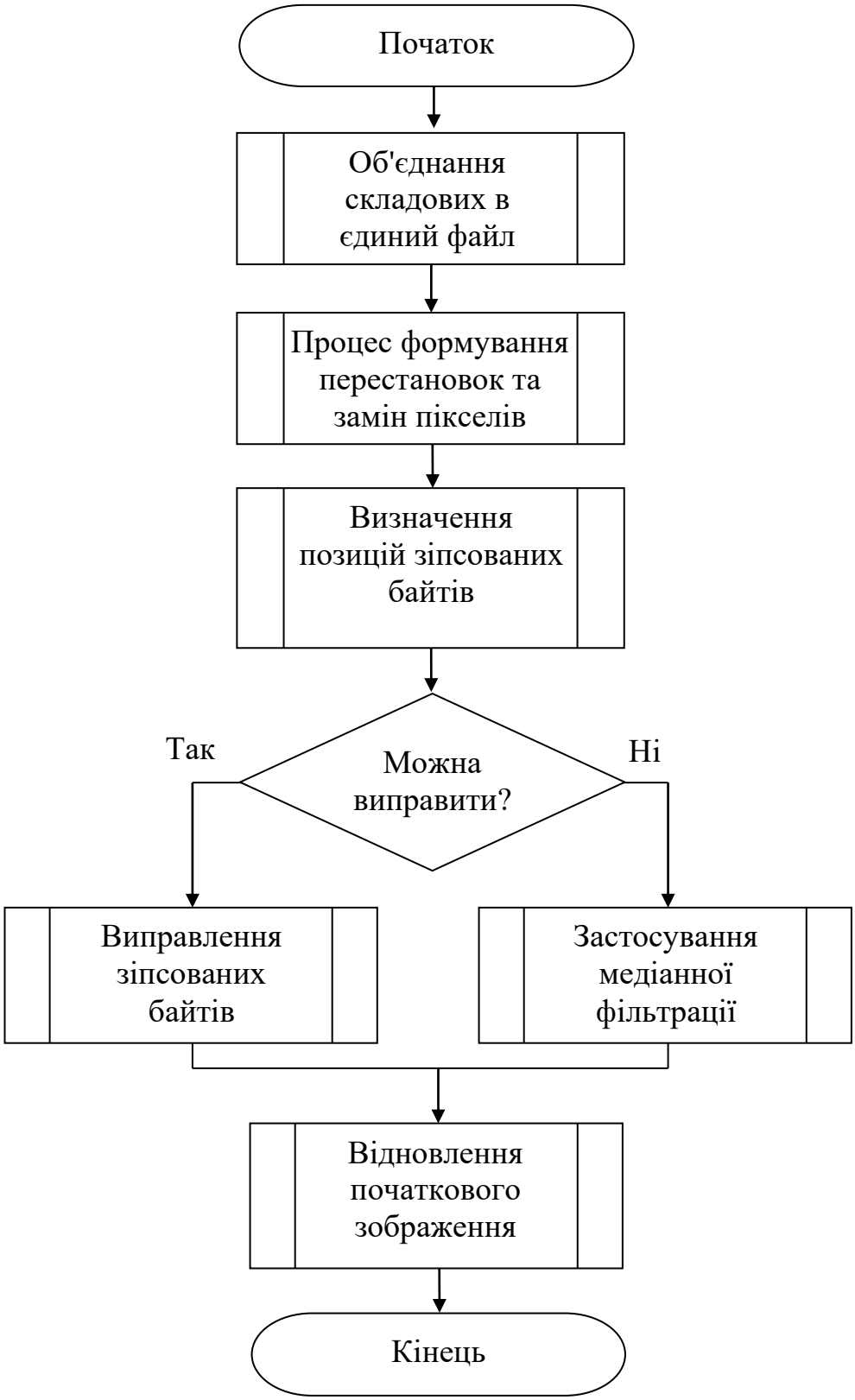
0				N
	[120, 211, 45]	[154, 205, 98]	[56, 137, 57]	[74, 41, 200]
	[23, 105, 175]	[68, 93, 100]	[168, 230, 154]	[3, 172, 173]
	[255, 134, 80]	[78, 5, 62]	[33, 21, 188]	[110, 160, 74]
M	[142, 194, 44]	[44, 47, 4]	[1, 132, 143]	[187, 117, 191]

За значенням пікселя

0				N
	(120 (R) + 211 (G) + 45 (B)) mod 256 = 120	(154 + 205 + 98) mod 256 = 201	(56 + 137 + 57) mod 256 = 250	(120 + 201 + 250) mod 256 = 59
	(23 + 105 + 175) mod 256 = 47	(68 + 93 + 100) mod 256 = 5	(168 + 230 + 154) mod 256 = 40	(47 + 5 + 40) mod 256 = 92
	(255 + 134 + 80) mod 256 = 213	(78 + 5 + 62) mod 256 = 145	(33 + 21 + 188) mod 256 = 242	(213 + 145 + 242) mod 256 = 88
M	(120 + 47 + 213) mod 256 = 124	(201 + 5 + 145) mod 256 = 95	(250 + 40 + 242) mod 256 = 20	239 (байти парності)

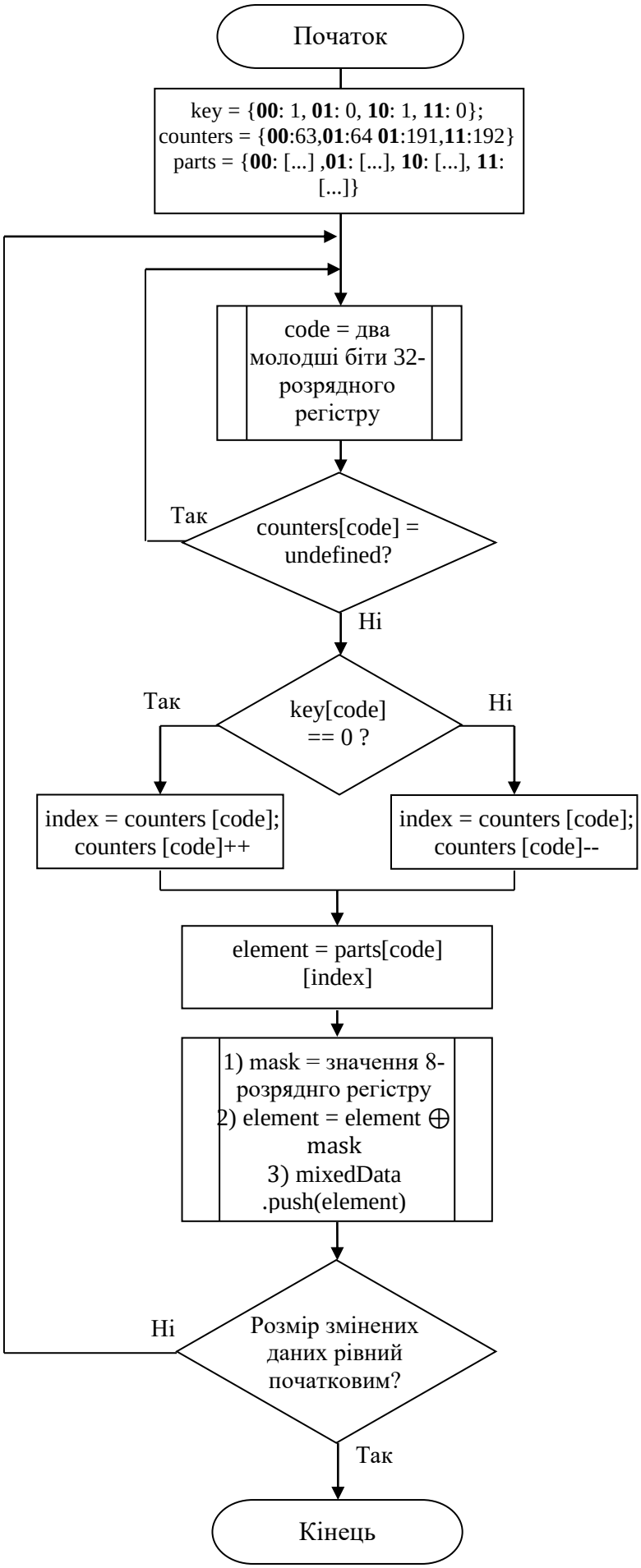
					08-20.МКР.001.00.000 ІЧ2		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Формування контрольних байтів	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		Лужецький В.А.					

Послідовність етапів методу відновлення секрету



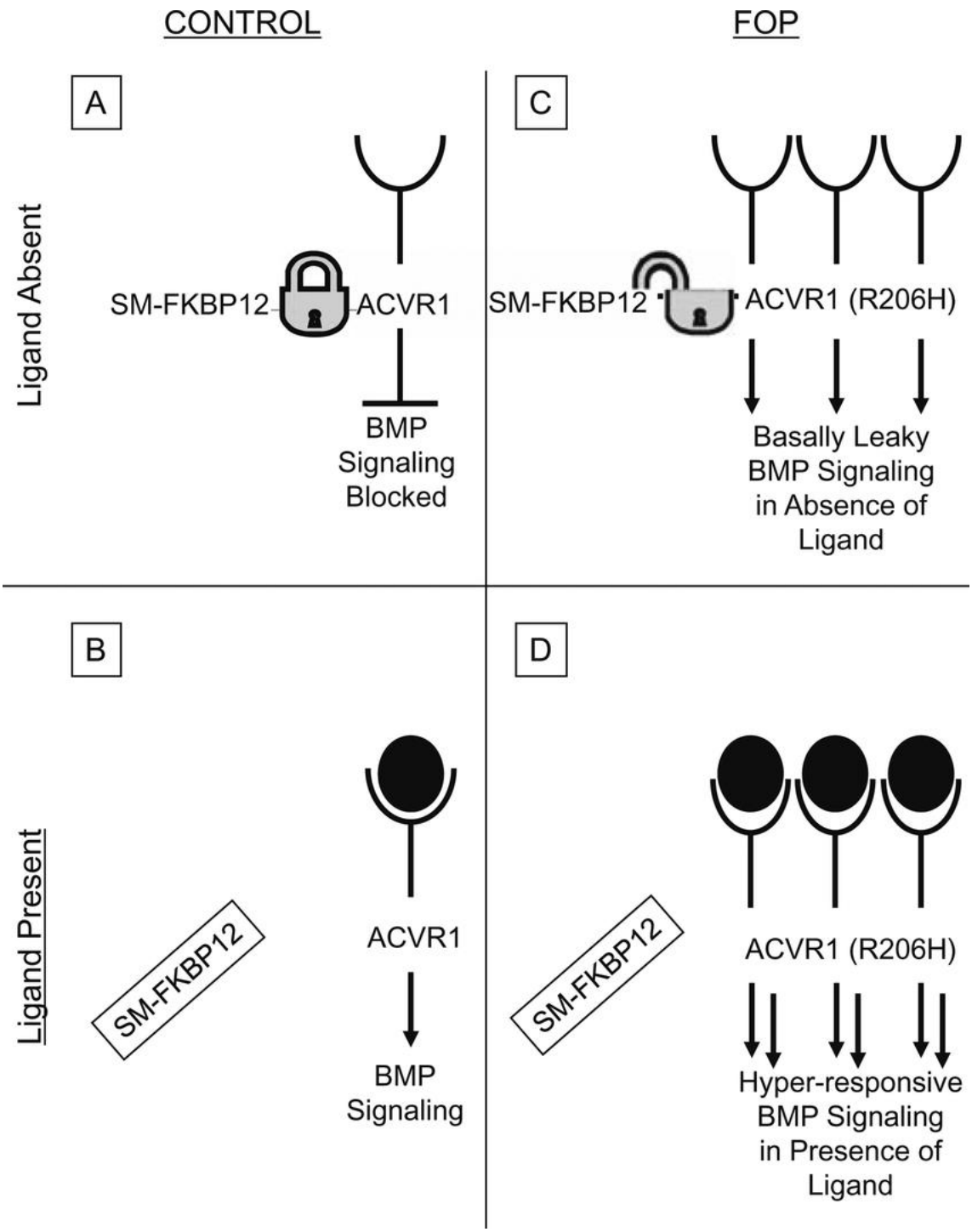
					08-20.МКР.001.00.000 ІЧЗ		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Послідовність етапів методу відновлення секрету	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Алгоритм формування перестановок та замін



					08-20.МКР.001.00.000 ІЧ4		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Алгоритм формування перестановок та замін	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Вигляд секретного зображення



					08-20.МКР.001.00.000 ІЧ5		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Вигляд секретного зображення	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Вигляд частин розподіленого секрету

	image1
	image2
	image3

					08-20.МКР.001.00.000 ІЧ6		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Вигляд частин розподіленого секрету	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					

Застосування медіанної фільтрації

Оточення зіпсованого пікселю $x_{i, j}$

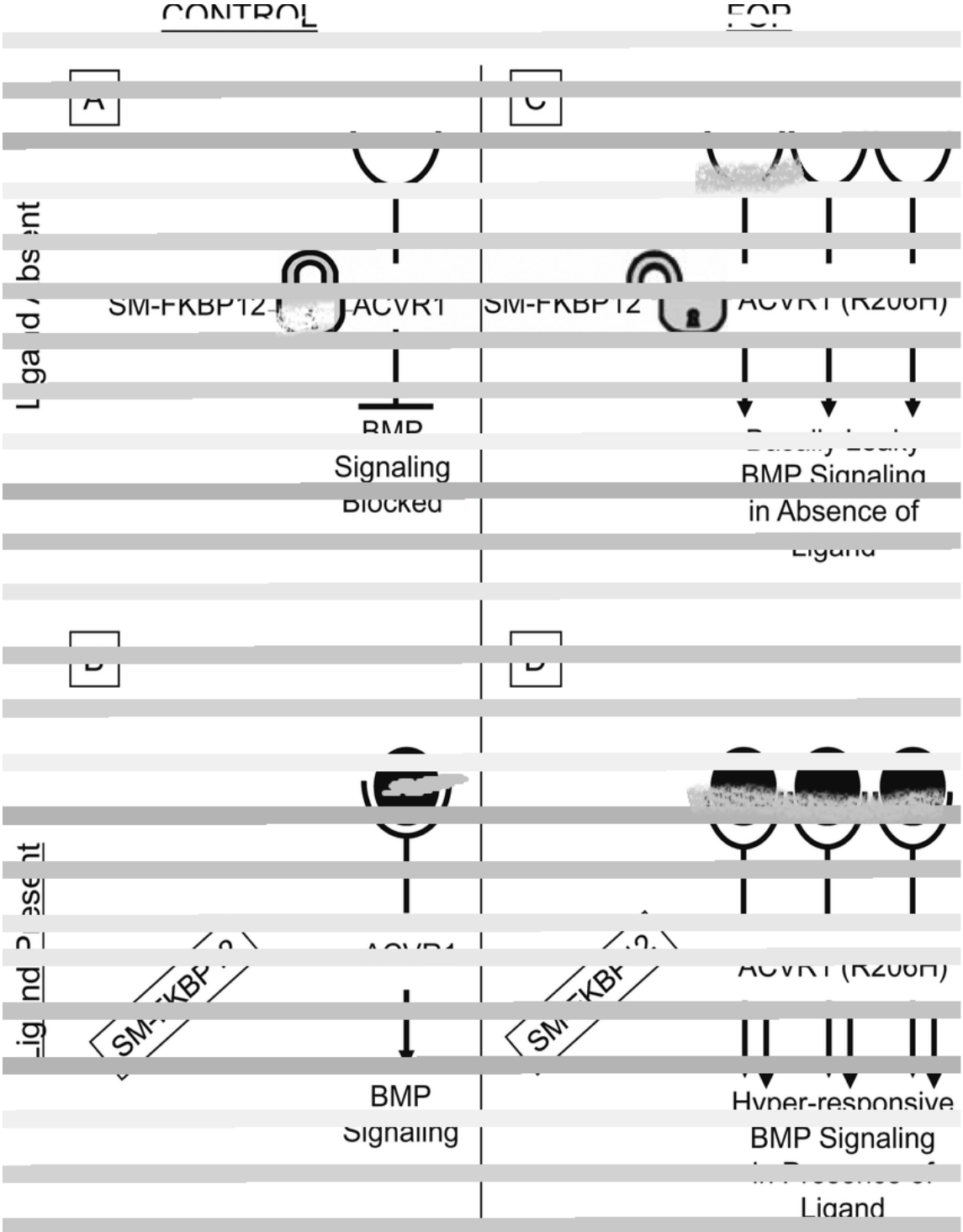
$x_{i-1, j-1}$	$x_{i-1, j}$	$x_{i-1, j+1}$
$x_{i, j-1}$	$x_{i, j}$	$x_{i, j+1}$
$x_{i+1, j-1}$	$x_{i+1, j}$	$x_{i+1, j+1}$

Зображення із зіпсованими байтами каналу R

187 (0, 0)	180 (0, 1)	183 (0, 2)	179 (0, 3)	217 (0, 4)
186 **(1, 0)**	***200*** **(1, 1)**	**177** **(1, 2)**	***189*** **(1, 3)**	***215*** **(1, 4)**
185 (2, 0)	*175* (2, 1)	193 (2, 2)	*194* (2, 3)	*235* (2, 4)
191	*184*	190	*182*	*235*
237	*227*	*231*	*232*	

					08-20.МКР.001.00.000 ІЧ7		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Застосування медіанної фільтрації	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		Лужецький В.А.					

Вигляд зображення при відновленні двома учасниками



					08-20.МКР.001.00.000 ІЧ8		
Змн	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бевзюк А.М..			Метод і засіб завадостійкого розподілу секрету. Вигляд зображення при відновленні двома учасниками	Літ.	Маса / Масштаб
Перевір.		Лужецький В.А..					
Реценз.		Азарова А.О.				ВНТУ, гр. 1 БС-18м	
Н. Контр.		Лужецький В.А.					
Затверд.		ЛужецькийВ.А.					