МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 24.00.00.000 ПЗ

Група ІПм-22-2

Гаврилюк Владислав

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Гаврилюк Владислав Володимирович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі, методи та алгоритми кодування та криптографічного захисту зображень

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Гаврилюк В.В.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Юрчишин Володимир Миколайович, д.т.н., професор

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

Допущено до захисту

В.о. завідувача кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Рецензент

доц.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
В.о. зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2023 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Гаврилюку Владиславу Володимировичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Моделі, методи та алгоритми кодування та криптографічного захисту зображень ”
керівник проекту (роботи) Юрчишин Володимир Миколайович, д.т.н., професор
затверджені наказом закладу вищої освіти від “ ” листопада 2023 р. №

2. Строк подання студентом проекту (роботи) 15 січня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій криптографічного захисту

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз методів захищеної передачі зображень
2. Технології криптографічного захисту графічних файлів та зображень
3. Дослідження процесу підвищення криптографічної стійкості захисту зображень
4. Алгоритмічна та функціональна реалізація методів захищеної передачі графічних файлів

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Просторовий розподіл функції (рис. 1.1)
2. Результат шифрування зображення методом RSA (рис. 1.2)
3. Схема шифрування (рис. 2.1)
4. Схема дешифрування (рис. 2.2)
5. Результат використання потокової модифікації (рис. 2.3)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Нормоконтроль	доц., к.т.н. Вовк Р.Б.	
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2023 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	01.10.2023	виконано
2	Аналіз концепцій та алгоритмів предметної області	25.10.2023	виконано
3	Аналіз методів захищеної передачі зображень	10.11.2023	виконано
4	Технології криптографічного захисту графічних файлів та зображень	22.11.2023	виконано
5	Дослідження процесу підвищення криптографічної стійкості захисту зображень	01.12.2023	виконано
6	Реалізація функціональності запропонованої інформаційної технології	15.12.2023	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.01.2024	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 97 с., 18 рис., 46 джерел.

Тема: Моделі, методи та алгоритми кодування та криптографічного захисту зображень

Об'єкт дослідження: процес забезпечення захищеності і достовірності передавання графічних файлів і зображень у інформаційних системах.

Мета роботи: дослідження методів підвищення криптографічної стійкості та достовірності інформації при передачі графічних файлів і зображень.

Предмет дослідження: методи, алгоритми та інформаційна технологія криптографічного захисту графічних файлів і зображень при їх передачі в інформаційних системах.

Результати дослідження:

Досліджено метод криптографічного захисту, який ґрунтується на поєднанні елементів методу RSA та кубічних алгебраїчних форм і має підвищену криптографічну стійкість при шифруванні повноколірних зображень.

Висновок:

Запропоновано модифікацію інформаційної технології захисту зображень із глибиною кольору в 1 і 4 біти внаслідок сумісного використання методу RSA, потокового та топологічного підходів, які запобігають появі контурів об'єктів на зашифрованих зображеннях.

ШИФРУВАННЯ, КРИПТОГРАФІЧНИЙ ЗАХИСТ, ГРАФІЧНИЙ ФАЙЛ, ЗОБРАЖЕННЯ, СТІЙКІСТЬ АЛГОРИТМУ, ЦИФРОВА ОБРОБКА, ДЕШИФРУВАННЯ, ІНФОРМАЦІЙНА СИСТЕМА.

ABSTRACT

Master Thesis: 97 pp., 18 fig., 46 sources.

Thesis Subject: Models, methods and algorithms for coding and cryptographic protection of images

Object of research: he process of ensuring the security and reliability of the transfer of graphic files and images in information systems.

Research goal: research of methods of increase of cryptographic stability and reliability of information at transfer of graphic files and images.

Subject of research: methods, algorithms and information technology of cryptographic protection of graphic files and images during their transmission in information systems.

The results:

The method of cryptographic protection, which is based on the combination of elements of the RSA method and cubic algebraic forms and has increased cryptographic stability in the encryption of full-color images, has been studied.

Conclusion:

It A modification of the information technology of image protection with color depth of 1 and 4 bits due to the combined use of the RSA method, streaming and topological approaches, which prevent the appearance of contours of objects on encrypted images, is proposed.

INFORMATION PRODUCT, WEBSITE, ROBUST, CONDITION, LIFE CYCLE, INFORMATION RESOURCE, CLASSIFIER, FORECASTING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ГСЦО - ґрафічна система цифрової обробки;

TDI - Transport Driver Interface – транспортний драйвер;

NDIS - Network Driver Interface Specification – специфікація інтерфейсу мережевого драйвера;

RSA - Rivest, Shamir, Adleman – криптографічний алгоритм з відкритим ключем.

ВСТУП

Актуальність роботи.

В сучасному світі інформаційний ресурс став займати одне з найважливіших місць в економічному розвитку. Володіння інформацією необхідної якості в потрібний час і в потрібному місці є запорукою успіху у будь-якому вигляді господарській діяльності. Монопольне володіння певною інформацією виявляється часто вирішальною перевагою в конкурентній боротьбі і зумовлює, тим самим, високу ціну «інформаційного чинника», який просто катастрофічно потребує свого захисту.

Безпеку даних використовують у багатьох сферах починаючи з невеликих компаній, Інтернет магазинів, закінчуючи великими корпораціями, банками, та державними структурами. Тому методи захисту інформації інтенсивно вдосконалюються. З’являються нові програми для збереження та кодування даних, попередження перед вторгненням та виключення викрадення даних, створюють нові алгоритми шифрування та дешифрування даних для того щоб твої дані не змогли відкрити не бажані особи.

У міру вдосконалення основних характеристик систем безпеки даних виникають нові можливості для вирішення завдань все зростаючого об'єму і складності. До їх числа відноситься шифрування інформації, що вимагає значних об'ємів цифрової пам'яті і відповідної кількості арифметичних операцій, через що, отримала розвиток лише останнім часом.

Підвищення ефективності передавання даних, забезпечення їх достовірності та захисту від несанкціонованого доступу є актуальною проблемою в інформаційних системах. Якщо донедавна криптографія використовувалася винятково у військових цілях, то сьогодні комп’ютерна криптографія стала поширеною поза межами військових відомств. Звичайні користувачі отримали можливість використовувати засоби захисту інформації під час здійснення звичайних операцій, наприклад, банківських транзакцій

тощо.

Проблеми захисту даних відображено в працях К. Шеннона, М. Діффі, М. Хеллмана, В. К. Задіраки. На основі аналізу їх праць можна стверджувати, що на сьогодні одним із найпоширеніших методів шифрування даних є метод RSA, основною перевагою якого є висока криптографічна стійкість. Однак застосування цього методу для шифрування зображень не є оптимальним, оскільки із зашифрованого зображення можна отримати інформативні дані, використовуючи звичайні алгоритми фільтрації зображень. Тому виник новий науковий напрям досліджень в галузі захисту зображень у комунікаційних системах, який ґрунтується на розвитку симетричних методів криптографічного аналізу та відображений у роботах Kwok-Wo Wong, Hai Yu, Zhi-Liang Zhu. Основним недоліком методів цього напряму є інформативні втрати, які стають критичними в задачах інтелектуального аналізу даних.

Альтернативним підходом є послідовне застосування методу RSA і оператора шуму, що, однак, не вирішує проблеми захисту зображень, оскільки спеціальними методами можна виявити і зняти зашумленість. Тобто, недоліків методу RSA не вдається позбутись.

Мета і задачі дослідження.

Метою магістерської роботи є дослідження методів підвищення криптографічної стійкості та достовірності інформації при передачі графічних файлів і зображень.

Для досягнення мети в роботі необхідно розв'язати **такі задачі:**

- Проаналізувати відомі методи кодування та забезпечення достовірності інформації під час передаванні зображень у інформаційних системах та дослідити їх ефективність.
- Дослідити метод шифрування повноколірних зображень з підвищеною порівняно з методом RSA криптографічною стійкістю.
- Описати метод шифрування повноколірних зображень, який має

підвищену криптографічну стійкість та оптимізує витрати пам’яті під час його застосування.

- Удосконалити інформаційну технологію захисту зображень для уникнення появи контурів об’єктів на зашифрованих зображеннях.

Об'єктом дослідження є процес забезпечення захищеності і достовірності передавання графічних файлів і зображень у інформаційних системах.

Предмет дослідження – методи, алгоритми та інформаційна технологія криптографічного захисту графічних файлів і зображень при їх передачі в інформаційних системах.

Методи дослідження ґрунтуються на: основних положеннях теорій захисту інформації; цифрової обробки зображень; теорії чисел; комп’ютерних мереж; методів шифрування; лінійної алгебри й функціонального аналізу.

Наукова новизна одержаних результатів полягає в дослідженні методу криптографічного захисту, який ґрунтується на поєднанні елементів методу RSA, кубічних алгебраїчних форм і має підвищену криптографічну стійкість при шифруванні повноколірних зображень.

Практичне значення одержаних результатів полягає в представленні модифікації інформаційної технології захисту зображень із глибиною кольору в 1 і 4 біти внаслідок сумісного використання методу RSA, потокового та топологічного підходів, які запобігають появі контурів об’єктів на зашифрованих зображеннях.

Структура і обсяг роботи. Магістерська робота складається зі вступу, 4 розділів, висновків та додатків. Загальний обсяг роботи – 97 сторінок і містить 18 рисунків. Список використаних літературних джерел складається із 46 найменувань.

РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ЗАХИЩЕНОЇ ПЕРЕДАЧІ ЗОБРАЖЕНЬ

1.1. Характеристики зображення

Характеристики зображень. У теорії цифрової обробки сиґналів зображенням вважають інформативну двовимірну матрицю, яка виражає усі характеристики відображеного об’єкта та впливи відображення, які обумовлені способами та процесами отримання зображення [1,2]. Елементами цієї матриці є значення кольору. Тут під значенням кольору розуміється узагальнене, яке є залежним від типу зображень. Наприклад, у разі напівтонових зображень елементом матриці є інтенсивність (відтінок, ґрадація) сірого кольору (яскравість). Додатковими параметрами можна вважати чіткість, контраст, оптичну густину, різкість тощо [8, 9].

У разі кольорових зображень елемент матриці визначається тріадою параметрів [10] моделі кольору, які за допомогою визначених адитивних правил можуть об’єднуватись у одне ціле число. Представлення цілим числом зазвичай використовується для зберігання чи перенесення зображень, або у ґрафічних системах цифрової обробки (ГСЦО) початкового рівня. У професійній обробці колір переважно подають трьома базовими параметрами: тоном, насиченістю та інтенсивністю або параметрами моделі представлення кольору. Колірний тон визначається піком та розподілом у спектрі видимого світла. Насиченість (інтенсивність певного тону) – ступінь відмінності яскравості хроматичного кольору від сірого. Повністю ненасичений колір буде відтінком (ґрадацією) сірого.

Додатково до зазначених параметрів дуже часто використовують такі характеристики:

- світловий параметр (ясність) – обернений ступінь відмінності від білого кольору;
- поняття глибини кольору – термін, який використовують для позначення

того, скільки бітів необхідно для зберігання інформації про колір піксела. Глибину кольору вимірюють у бітах на піксел.

Недоліком базових параметрів є неможливість їхнього безпосереднього перетворення на колірний сигнал. Проте, на відміну від параметрів моделей представлення чи представлення цілим числом, вони надають найповніше фізичне уявлення про колір. Тому основними параметрами ГСЦО є саме ці параметри, особливо у разі візуального сприйняття результатів обробки.

Отже, цифрове зображення ***P*** з шириною *l* і висотою *h* можна розглядати як матрицю пікселів:

$$< dtp_{ij} >_{1\le i\le n,\ 1\le j\le m}, \tag{1.1}$$

де dtp_{ij} – піксель з координатами *i* та *j*, *n* і *m* – розмір зображення – число пікселів по ширині *l* та висоті *h*. У загальному випадку *n* і *m* є залежними від *l* та *h*, а тому коректнішим є запис

$$n = n(l)\ im = m(h). \tag{1.2}$$

Матриці пікселів (1.1) відповідає матриця кольорів

$$\mathbf{C}=\begin{pmatrix} c_{1,1} & \dots & c_{1,m} \\ \dots & \dots & \dots \\ c_{n,1} & \dots & c_{n,m} \end{pmatrix}, \tag{1.3}$$

де c_{ij} – значення кольору у кольорових зображень або значення інтенсивності у напівтонових зображень для піксела dtp_{ij}. Тобто має місце відповідність

$$P = \mathbf{P}_{l,h} = \left[pxl_{ij} \right]_{1 \le i \le n(l), 1 \le j \le m(h)} \rightarrow \mathbf{C} = \left[c_{ij} \right]_{1 \le i \le n(l), 1 \le j \le m(h)}. \quad (1.4)$$

Під градацію яскравості піксела зазвичай виділяється 1 байт, причому яскравості 0 відповідає чорний колір, а яскравості 255 (максимальна інтенсивність) – білий. Для кольорового зображення виділяється по 1 байту під градації яскравостей всіх трьох кольорів. Загалом, можливе кодування градацій яскравості відмінним від 8 числом бітів (4, 12, або 24), проте людське око здатне розрізняти не більше 255 градацій яскравості на кожний колір, хоча спеціальна апаратура може і точніше передавати кольори.

У випадку кольорових зображень c_{ij} розглядається як вектор основних характеристик кольорової палітри. Наприклад, якщо задано зображення у 24-розрядному форматі палітри *RGB*, то

$$\mathbf{c}_{i,j} = \{c_{i,j}^R, c_{i,j}^G, c_{i,j}^B\},$$

де $c_{i,j}^R, c_{i,j}^G, c_{i,j}^B$ – значення інтенсивності червоного, зеленого та синього кольорів піксела dtp_{ij} відповідно. Тому наведений нижче алгоритм потрібно застосувати незалежно до кожної окремої характеристики.

Якщо, наприклад, по ширині розглянути вектор

$$\tilde{c}_j = \{c_{i,j} \mid i = \overline{1,n}\}, \quad (1.5)$$

то матрицю кольорів (1.3) можна записати у вигляді

$$\mathbf{C} = \left[\tilde{c}_j \right]_{j=\overline{1,m}}. \quad (1.6)$$

Ще однією важливою характеристикою зображення є присутність у зображенні контурів об’єктів. Задача виділення контура на зображенні вимагає виконання операцій над сусідніми пікселами зображення, які виявляють чутливість до змін і пригашають області постійних рівнів інтенсивності, тобто контури – це ті області зображення, де виникають різкі зміни інтенсивності.

Математично ідеальний контур – це розрив просторової функції інтенсивності в площині зображення. Тому виділення контура об’єкта на зображенні означає пошук різких змін, тобто максимумів модуля вектора градієнта. Ця особливість є однією з головних причин, що призводить до збереження контурів об’єктів на зображенні при шифруванні в системі RSA, оскільки шифрування відбувається шляхом піднесення до степеня за модулем деякого натурального числа. При цьому на контурі та на прилеглих до нього пікселях зображення піднесення до степеня значення інтенсивності дає ще більший розрив просторової функції інтенсивності.

Класи зображень. Залежно від розв’язуваних задач відомо багато різних способів класифікації зображень, наприклад: за категоріями об’єктів, за фотометричним змістом тощо.

Використаємо наступну класифікацію зображень [3]:

- монохроматичні (бінарні) зображення – визначаються тим, що усі пікселі можуть мати лише два значення: 1 – значення кольору, 0 – значення фону. Зазвичай розглядають пару чорний та білий кольори. Піксел представляється одним бітом;
- напівтонові зображення [4] – визначаються тим, що кожен піксел може мати значення (відтінки, ґрадації) кольору в діапазоні [0; 2n - 1], де n – ціле число, кратне 4 або 8 і є залежним від кількості байтів, яка виділяється для піксела. Типово виділяють один байт і розглядають зображення у 256 ґрадаціях сірого кольору;
- кольорові зображення [5] – визначаються тим, що для кожного піксела виділяється кількість байтів, яка визначається кількістю параметрів

моделі кольору. Типово це три байти; по одному для 256 ґрадацій кожного параметра кольорової моделі;

- зображення із неперервним тоном – визначається тим, що містить значення пікселів, які неістотно відрізняються між собою (наприклад, зображення ландшафтів). У візуальному сприйнятті це перетворюється на існування подібних (близьких) кольорів (або напівтонів). У цьому разі піксел подають одним байтом для напівтонів або трьома байтами для кольорових зображень;
- кусково-однорідні зображення [6] – визначаються існуванням великих ділянок (областей) із однаковим значенням кольору або напівтонів (наприклад, зображення для мультиплікації. При цьому значення пікселів у дотичних ділянках можуть істотно відрізнятись. Витрати пам'яті при зберіганні цифрових кусково-однорідних зображень є подібними до випадку зображення із неперервним тоном;
- дискретно-тонові зображення – визначаються відсутністю артефактів, які виникають під час отримання зображення. Найчастіше це синтезовані зображення. Витрати пам'яті при зберіганні цифрових кусково-однорідних зображень є подібними до випадку зображення із неперервним тоном.

1.2. Основні властивості теорії чисел

Більшість сучасних алгоритмів криптографічного захисту основуються на поняттях теорії чисел. Тому виділимо деякі елементи цієї теорії.

Властивість 1.1 [4]. Якщо $a_1 \equiv a_2 (\mathrm{mod}\ m)$, то числа a_1 і a_2 належать до одного показника за модулем m.

Властивість 1.2 [4]. Якщо a належить до показника δ за модулем m, то в послідовності степенів

$$a^0 = 1, a, a^2, a^3, ..., a^{\delta-1} \tag{1.7}$$

всі числа не порівнянні між собою за модулем m.

Властивість 1.3 [5]. Якщо a належить до показника δ за модулем m, то конгруенція

$$a^{\alpha} \equiv a^{\beta} (\text{mod } m) \tag{1.8}$$

де α і β – деякі цілі додатні числа, існує тоді і лише тоді, коли

$$\alpha \equiv \beta (\text{mod } \delta) \tag{1.9}$$

1.3. Базові схеми шифрування та дешифрування зображень

1.3.1. Симетричні алгоритми шифрування

До симетричних алгоритмів шифрування [6,7] даних належать методи та алгоритми шифрування, в яких для шифрування і дешифрування даних використовується один і той самий ключ (або, в деяких випадках, різні ключі, але такі, що є спорідненими та легко обчислюються). Симетричні алгоритми шифрування були єдиними загальновідомими та вживаними до середини 70-х років XX століття.

Сучасні дослідження в області симетричного шифрування даних переважно зосереджені на розвитку та застосуванні блочних та потокових алгоритмів. Блочні шифри, подібно до поліалфавітного шифру Альберті, отримують на вході фрагмент відкритого тексту та ключ шифрування, і на виході повертають зашифрований текст такого самого розміру. Оскільки розмір блоків шифрування зазвичай менший за розмір повідомлення, що шифрується, виникає необхідність в методі об’єднання послідовних блоків зашифрованих

даних. Існує кілька методів для об’єднання блоків, котрі відрізняються певними ключовими аспектами.

Алгоритми шифрування Data Encryption Standard (DES) та Advanced Encryption Standard (AES) [10] є стандартами блокових шифрів, затверджених урядом США (стандартизацію DES скасовано після прийняття стандарту AES). Незважаючи на скасування стандартизації DES як застарілого, він та його модифікація Triple-DES продовжують широко застосовуються в деяких галузях: шифрування в банкоматах, забезпечення приватності електронного листування, забезпечення безпечного доступу до віддалених терміналів тощо.

DES (Data Encryption Standard) [10] – симетричний алгоритм шифрування даних, що ґрунтується на шифрі Хорста Фейстеля Люцифер [12]. Цей стандарт шифрування даних був прийнятий урядом США і набув широкого міжнародного застосування в період з 1976 до кінця 1990-х років. З часу своєї появи алгоритм викликав неоднозначні відгуки. Оскільки структура DES містила певні засекречені елементи, виникали побоювання щодо можливості контролю з боку Національного агентства безпеки США (National Security Agency). Ще одним недоліком алгоритму вважалась мала довжина ключа шифрування. Однак, ці недоліки не завадили йому стати загальновживаним стандартом. Алгоритм DES став основою для багатьох сучасних блочних алгоритмів шифрування та криптоаналізу.

Враховуючи популярність та період використання алгоритму DES, існує багато досліджень його криптографічної стійкості. Ґрунтовно оцінив безпеку DES Брюс Шнайдер, який детально проаналізував велику кількість публікацій про криптоаналіз DES. Сьогодні DES вважають застарілим та нестійким [5, 6], оскільки:

1) розмір ключа шифрування, що дорівнює 56 бітів, є занадто малим, і існує загроза достанього легкої процедури взлому та знаходження ключа методом послідовного перебору;

2) алгоритм DES є нестійким до лінійного криптоаналізу, тобто лінійною атакою можна знайти ключ шифрування навіть швидше, ніж послідовним перебором.

Враховуючи поширеність та популярність алгоритму DES, було запропоновано багато модифікацій цього алгоритму з метою підвищення його криптографічної стійкості, наприклад, заміна S-блоків DES новими, стійкішими до лінійної атаки. Однак поширення жодна з модифікацій алгоритму DES, окрім 3DES, котра являє собою не модифікацію, а лише видозмінений режим шифрування, не набула.

Розвитком DES став Advanced Encryption Standard (AES), також відомий як Rijndael. AES, як і DES, є симетричним алгоритмом блочного шифрування при розмірі блока у 128 бітів та розмірі ключа у 128, 192 чи 256 бітів, і був прийнятий як американский стандарт шифрування даних урядом США у 2002 році. Такий вибір був спричинений широким застосуванням та активним криптоаналізом алгоритму. Сьогодні AES є одним із найпоширеніших симетричних алгоритмів блочного шифрування.

Необхідність розроблення нового стандарту шифрування замість застарілого та недостатньо стійкого DES виникла ще в середині 1990-х років. Апаратно орієнтована реалізація DES не забезпечувала бажаної швидкості виконання на платформах з обмеженими ресурсами, а стійкіша модифікація 3-DES була ще повільнішою.

Як зазначалось вище, розмір блоку алгоритму AES має фіксовану довжину у 128 бітів при розмірі ключа у 128, 192 або 256 бітів. Розміри блоку та ключа алгоритму Рейндол можуть набувати значень у діапазоні від 128 до 256 бітів з кроком у 32 біти.

Відмінність потокових шифрів від блочних, полягає в довільній довжині ключа, котрий застосовується для відкритого тексту побітово або політерно і має схожі риси з одноразовим блокнотом. Особливістю потокових шифрів є те, що потік шифротексту обчислюється на основі внутрішнього стану алгоритму,

котрий зазнає змін впродовж його дії. Ця зміна стану керується ключем, а в деяких алгоритмах ще і потоком відкритого тексту. Одним з найпоширеніших алгоритмів потокового шифрування є RC4.

Наступним прикладом симетричного шифрування є криптографічні хешувальні функції (cryptographic hash functions, або message digest functions). Їх особливістю є те, що вони не завжди використовують ключі шифрування, але є поширеним класом криптографічних алгоритмів шифрування. Хешувальні функції на вході отримують певні дані та обчислюють число фіксованого розміру (хеш). Якісні хешувальні функції розроблено так, що практично унеможливлюють знаходження колізій: двох відкритих текстів з однаковим значення хешу.

Коди аутентифікації повідомлень (message authentication code, MAC) подібні до криптографічних хешувальних функцій. Їх відмінність полягає в тому, що вони використовують секретний ключ для аутентифікації значення хешу при отриманні повідомлення. Ці функції пропонують захист проти атак на прості хешувальні функції.

1.3.2. Асиметричне шифрування

Асиметричні алгоритми шифрування – це алгоритми шифрування, основною особливістю яких є використання різних ключів у процесах шифрування та дешифрування. З цього випливає основна перевага асиметричного шифрування: відсутня необхідність погоджувати ключ шифрування між відправником та одержувачем по захищеному каналу.

Прикладами криптосистем з відкритим ключем є Elgamal (автор Тахір Ельгамаль), RSA (автори: Рон Рівест, АдіШамір і Леонардо Адлман), Diffie-Hellman і DSA – Digital Signature Algorithm (автор Девід Кравіц).

Проблему керування ключами вирішила криптографія з відкритим, або асиметричним ключем, концепцію якої запропонували Уітфілд Діффі і Мартін

Хеллман у 1975 році. Криптографічна система з відкритим ключем – це асиметрична схема, у якій застосовуються пари ключів: відкритий (public key), за допомогою якого шифруються дані, і відповідний йому закритий (private key), за допомогою якого дані дешифруються. Відкритий ключ, на відміну від закритого, вільно поширюється. За його допомогою будь-який агент може зашифрувати інформацію, яка розшифровується винятково закритим ключем.

Незважаючи на те, що пара ключів є математично зв'язаною, обчислити закритий ключ за допомогою відкритого практично неможливо.

Поява шифрування з відкритим ключем стала технологічною революцією, яка зробила стійку криптографію доступною в широкому використанні.

Прикладом криптосистеми з відкритим ключем є схема, запропонована Тахером Ель-Гамалем як один з варіантів алгоритму Діффі-Хеллмана. Схема Ель-Гамаля (Elgamal) ґрунтується на складності обчислення дискретних логарифмів у кінцевому полі [11]. Криптосистема містить алгоритм шифрування і алгоритм цифрового підпису. Схему Ель-Гамаля покладено в основу стандартів електронного цифрового підпису в США (DSA) і Росії (ГОСТ Р 34.10-94).

1.4. Потокові системи шифрування

Потокове шифрування – це спосіб захисту даних, за якого кожен символ шифрується окремо. Оскільки потокові криптосистеми обробляють інформацію посимвольно, застосовуючи їх, можна шифрувати та дешифрувати дані будь-якого розміру. Такий спосіб захисту даних зазвичай застосовується в каналах зв'язку. Часто перед опрацюванням даних у потокових криптосистемах застосовується операція накладання на вихідні дані попередньо згенерованої послідовності [12]. Накладання згенерованої послідовності на вихідні дані здійснюється побітово з використанням арифметичної операції за модулем 2. У

потокових криптосистемах окремі символи згенерованої послідовності прийнято позначати літерою γ. Тому потокове шифрування отримало ще одну назву – шифри гаммування.

1.4.1. Шифр з авто ключем

Шифром з автоключем називається потоковий шифр, за якого шифрується кожний символ вихідних даних залежно не лише від його позиції у відкритому тексті, а й від інших символів вихідних даних. Шифрування відбувається так: записується відкритий текст, під ним записується ключове слово $k_1, k_2, \ldots, k_r$, а далі – ще раз відкритий текст, зміщений на r символів праворуч. Отримані послідовності додаються за модулем m.

При криптоаналізі шифру з автоключем спочатку визначають довжину ключового слова r. Якщо у відкритому тексті деяка n-грама зустрічається двічі на відстані $2r$, то у зашифрованому тексті також будуть однакові n-грами на відстані r.

Тобто, якщо проаналізувати відстані між однаковими n-грамами у зашифрованому тексті, можна знайти відстань r. Для того, щоб визначити першу букву ключового слова k_1 необхідно розглянути фрагмент зашифрованого тексту $Y_1 = y_1, y_{r+1}, y_{2r+1}, \ldots$. Далі потрібно перебрати всі значення $0 \le k_1 \le m-1$ і для кожного розрахувати

$$y_1 - k_1 \pmod m = x_1,\ y_{r+1} - x_1 \pmod m = x_{r+1},\ y_{2r+1} - x_{r+1} \pmod m = x_{2r+1}, \ldots .$$

Отримані рівності виконуються в тому випадку, коли k_1 має істинне значення. При цьому частоти букв у зазначеній послідовності близькі до частот букв у відкритому тексті, а при невірному значенні k_1 частоти є згладженими. Аналогічно знаходять решту букв ключового слова.

У випадку шифрування автоключем використано один з найважливіших принципів криптоаналізу: знаходження ключа по частинах. За умови безпосереднього перебору усіх можливих значень ключового слова потрібно було б перебрати m^r можливостей (вважаючи, що ключ може бути і беззмістовним набором букв). При “розумних” значеннях m та r це практично неможливо. Знаходячи ж букви ключа по черзі, максимальна кількість спроб дорівнює mr.

1.4.2. Потокові системи шифрування, побудовані на лінійних регістрах зсуву зі зворотнім зв’язком

Цю криптографічну систему було винайдено Вернамом і названа на його честь.

У потокових криптографічних систем, побудованих на лінійних регістрах із зсувом зі зворотним зв’язком [15], кожному символу відкритого тексту відповідає один символ зашифрованого тексту. Відсутнім є поділ тексту на блоки, подібні до блоків блочного шифрування. Розмір тексту залишається постійним. Простота апаратної реалізації, а з нею і висока швидкодія системи забезпечуються використанням лише найпростіших операцій: операції додавання за модулем 2 та операції зсуву регістру.

Шифрування відбувається шляхом додавання потоку вихідного тексту та згенерованої послідовності (гами) за модулем 2. Основною особливістю операції XOR (додавання за модулем 2) є те, що при застосуванні її парну кількість разів забезпечується повернення до початкового значення. Відповідно, дешифрування відбувається шляхом додавання зашифрованого тексту з гамою за модулем 2.

Отже, безпека такої криптосистеми повністю залежить від властивостей згенерованої послідовності. Для забезпечення високої криптографічної стійкості, кожен біт гами повинен бути незалежною випадковою величиною.

У 1949 р. К. Шеннон довів, що описана система шифрування володіє абсолютною криптографічною стійкістю. Однак, оскільки шифрування та дешифрування даних відбувається за допомогою випадкової послідовності, це означає, що необхідно забезпечити надсилання цієї послідовності захищеним комунікаційним каналом. А це значно ускладнює завдання обміну зашифрованими повідомленнями, тому ця криптосистема не набула поширення.

При застосуванні шифрування на лінійних регістрах зі зсувом в реальних криптографічних системах генерується ключ певного розміру, який передається захищеним каналом. На основі цього ключа генерується послідовність (гама), яка є псевдовипадковою [16].

1.5. Проблема стійкості криптографічної системи

1.5.1. Принципи Шеннона побудови стійких шифрів

- *Розсіювання* [5] (*розповсюдження помилок*) – це властивість криптографічного перетворення, при якому зміна одного символу (біту) на вході спричиняє за собою зміну зазвичай великої кількості символів (бітів) на виході. Основною ідеєю цього принципу є поширення впливу окремого символу вхідного тексту або ключа на множину знаків шифротексту. Найвідомішою формалізацією цієї властивості в сучасних теоріях криптографічних систем є так званий “лавинний” ефект.
- *Перемішування* [6] – це такі криптографічні перетворення, які зменшують можливість атаки, яка основується на статистичному аналізі характеррис-тик відкритого і зашифрованого текстів.

Шеннон запропонував будувати криптографічно стійкі шифри за цими принципами за допомогою послідовного використання різних криптографічних

перетворень, кожне з яких саме по собі не є стійким – наприклад, за допомогою класичних перетворень підстановки, перестановки, гамування тощо.

Існують і деякі інші методи, наприклад, основані на еліптичних кривих [19]. Проте дві основні категорії криптографічних систем складають алгоритми, які ґрунтуютсья на складності знаходження дільників великих чисел і обчислень дискретних логарифмів. З математичного боку ці задачі вважаються особливо складними. А тому відсутні методи їх ефективного розв'язання.

1.5.2. Криптографічна система RSA

RSA є криптографічною системою з відкритим ключем [22]. Цей метод є першим такого типу, який використовувався як для шифрування даних, так і для цифрового підпису [9, 13]. Метод використовується у великій кількості криптографічних додатків.

Криптографічна стійкість алгоритму RSA ґрунтується на складності факторизації простих чисел [25]. Алгоритм оперує двома ключами: відкритим – для шифрування даних та закритим – для їх дешифрування.

Для генерації відкритого та закритого ключів необхідно виконати такі дії:

1) вибрати два великі прості числа z_1 і z_2;

2) обчислити їх добуток

$$z = z_1 z_2;$$

Числа z_1 і z_2 є прихованими. Вони повинні бути знищені відразу після процедури генерування пари ключів;

3) обчислити функцію Ейлера

$$\phi(z) = (z_1 - 1)(z_2 - 1);$$

4) вибрати ціле число e таке, щоб воно було взаємно простим із значенням функції Ейлера $\phi(z)$ і меншим за нього: $1 < e < \phi(z)$ [12];
5) за розширеним алгоритмом Евкліда [2, 3] знайти таке число d, щоб виконувалась конгруенція

$$ed \equiv 1 \left(\mathrm{mod}\, \phi(z)\right).$$

Пари чисел (z, e) і (z, d) є відповідно відкритою та закритою (секретною) частинами ключа. За цим означенням числа e і d називають відповідно відкритою й закритою (секретною) експонентами.

Для процедури шифрування повідомлення $c_{i,j} < z$ обчислюється значення

$$c' = c^e \bmod z.$$

Число c' використовується як шифротекст. Дешифроване значення визначається так

$$c = c'^d \bmod z.$$

У [5, 10] показано, що при розшифруванні вихідне повідомлення повністю відновлюється:

$$c'^d \equiv \left(c^e\right)^d \equiv c^{ed} \left(\mathrm{mod}\, z\right).$$

З умови конгруенції для деякого цілого k маємо

$$ed = k\phi(z) + 1.$$

Отже

$$c^{ed} \equiv c^{k\phi(z)+1} \left(\mathrm{mod}\, z\right).$$

За теоремою Ейлера [1] маємо:

$$c^{\phi(z)} \equiv 1(\mathrm{mod}\, z),$$

тому

$$c^{ed} \equiv c^{k\phi(z)+1}(\mathrm{mod}\, z) \to c'^{d} \equiv c(\mathrm{mod}\, z).$$

1.5.3. Атаки на систему RSA

Найпоширенішим способом пошуку двох великих простих чисел z_1 і z_2 для генерації ключа алгоритму RSA є використання ймовірнісних тестів [21] чисел на їх простоту. У окремих випадках використовуються таблиці попередньо згенерованих простих чисел.

Очевидно, що процедура генерації чисел z_1 і z_2 повинна бути реалізована криптографічно надійним генератором випадкових чисел. При цьому відстань між z_1 і z_2 повинна бути достатньо великою, оскільки в цьому випадку завдяки методу факторизації Ферма їх не можна буде віднайти. Більше того, у рекомендаціях із використання алгоритму RSA для уникнення атаки $z_1 - 1$ алгоритмом Поларда пропонується використовувати «сильні» прості числа.

При практичному використанні рекомендується певним способом доповнювати повідомлення. Відсутність доповнень може призвести до таких проблем [22]:

- значення $m = 0$ і $m = 1$ за будь-яких значеннь e і z у процесі шифрування породжують значення 0 і 1;
- за достатньо малого значення коефіцієнта e (наприклад, e = 3) показано, що вихідне повідомлення достатньо легко відновлюється [5];
- оскільки в процесі роботи процедури шифрування на основі алгоритму RSA не використовуються випадкові значеннння (тобто RSA є детермінованим алгоритмом), то на криптографічний захист можна

здійснити атаку з обраним відкритим (вихідним) текстом.

Вирішення цих проблем забезпечується додаванням (доповненням) у процесі шифрування повідомлення деякого випадкового значення. Технологія доповнення визначається забезпеченням виконання умов $m \neq 0$, $m \neq 1$ і $m^e > z$. Більше цього, оскільки повідомлення розширюється (доповнюється) випадковими даними, то у результаті шифрування одного і того самого вихідного (відкритого) тексту шифротекст (зашифроване повідомлення) щоразу буде іншим. А це унеможливлює атаку з обраним відкритим текстом.

Як вже зазначалось, система RSA [23] використовується також для організації захисту програмного забезпечення і в схемах цифрового підпису. Також вона використовується у відкритій сім'ї методів шифрування PGP та в інших системах шифрування (наприклад, DarkCryptTC при будь-яких значеннях e і z). Достатньо часто використовуються схеми сумісного використання RSA в поєднанні його елементів та симетричних алгоритмів шифрування.

Основним обмеженням для організації криптографічного захисту у інформаційних системах є низька швидкодія (близько 30 кбіт/с при 512-бітовому ключі на процесорі з тактовою частотою 2 Ггц). Типовим способом подолання цього обмеження є використання швидшого та простішого у реалізації симетричного методу шифрування повідомлення з випадковим ключем (сеансовий ключ). А вже сам ключ шифрують за допомогою алгоритму RSA. Недоліком цього підходу є виникнення потенційних загроз, оскільки для генерації сеансового ключа симетричного методу шифрування стає необхідним використання криптографічно стійких генераторів випадкових чисел і симетричного криптографічного алгоритму, який ефективно протистоїть атакам. Цей підхід отримав широке застосування у методах AES, IDEA, Serpent, Twofish та ін.

Теорема 1.1. Якщо середня кількість двійкових символів на один символ повідомлення асимптотично прямує до ентропії джерела повідомлення (за

відсутності завад), то система ефективного кодування дискретних повідомлень існує.

Теорема 1.2. За наявності завад в каналі завжди можна знайти таку систему кодування, при якій повідомлення будуть передані з заданою достовірністю.

Асиметричну криптосистему можливо зламати, якщо вдасться сконструювати ефективний алгоритм визначення таємного ключа, знаючи відкритий. У випадку системи RSA це зводиться до такої задачі:

<u>задано</u> : e, z; $z = z_1 z_2$, $\left(e, \phi(z)\right) = 1$;

<u>знайти:</u> d, що для всіх x виконується конгруенція $x^{ed} \equiv x \left(\mathrm{mod}\, z\right)$.

Задача пошуку d зводиться до обчислення функції Ейлера $\phi(z)$, що рівносильно факторизації числа z на два прості співмножники z_1 і z_2. Задача розкладу числа на прості множники є задачею класу *NP*, проте визначити її складність поки неможливо, оскільки невідомо, чи це є *NP*-повна задача.

Окрім описаної типовими атаками на алгоритм RSA є:

1. Метод без базового аналізу реалізації RSA при відомому відкритому ключі (e,n) та зашифрованому тексті C. У цій атаці необхідно підібрати число j, для якого виконується наступна рівність: $C^{ej} \left(\mathrm{mod}\, n\right) = C$. Тобто, здійснюється j шифрувань відкритим ключем перехопленого зашифрованого тексту: $\left(\ldots\left(\left(C\right)^e\right)^e \ldots\right)^e \left(\mathrm{mod}\, n\right) = C^{ej} \left(\mathrm{mod}\, n\right)$. Знайшовши таке j, необхідно визначити $C^{ej-1} \left(\mathrm{mod}\, n\right)$ (тобто j - 1 разів повторюється операція шифрування) – це значення і є відкритим текстом M. Це випливає з того, що $C^{ej} \left(\mathrm{mod}\, n\right) = \left(C^{ej-1} \left(\mathrm{mod}\, n\right)\right)^e = C$. Тобто, деяке число $C^{ej-1} \left(\mathrm{mod}\, n\right)$ в степені e дає зашифрований текст C, а отже, це число і є відкритим текстом M.

2. Атака на підпис RSA за вибраним зашифрованим текстом при відомому відкритому ключі. Необхідно вибирати деяке $r : r < n, (r, n) = 1$ і обчислити $x = r^e \pmod n$. Далі потрібно обчислити $t = r^{-1} \pmod n$ і $y = xC \pmod n$ та надіслати y на підпис відправнику. Відправник, нічого не підозрюючи, підписує текст y: $w = y^d \pmod n$ і відсилає (повертає) w. Після цього обчислюється $tw \pmod n = r^{-1} y^d \pmod n = x^{-d} x^d C^d = C^d = M$, де M і є вихідним повідомленням. Атака має дещо гіпотетичний характер, але тим не менш дозволяє зробити кілька важливих висновків:

- підписувати і шифрувати дані потрібно різними ключами;
- додавати випадковий вектор у підпис або використовувати механізм хеш-функції.

1.6. Використання алгебраїчних форм для задач шифрування

1.6.1. Визначеність алгебраїчних форм

Формою називається однорідний поліном від двох або більше змінних, тобто поліном, всі елементи якого мають однаковий повний степінь за сукупністю змінних [7]; наприклад, $x^2 + xy + y^2$ – форма другого степеня (квадратична форма), $x^3 - x^2 y + 3xy^2 + y^3$ – форма третього степеня (кубічна форма) . Основна проблема використання однорідних поліномів полягає у тому, що невідомо які цілі числа можна подати за допомогою форми (тобто яких цілих значень може набувати форма) при цілих значеннях змінних. Для простоти аналізу і використання обмежимося формами лише двох змінних, тобто формами виду $f(x, y) = ax^2 + bxy + cy^2$ з дискримінантом $\Delta = 4ac - b^2$.

Форми з додатним дискримінантом називаються визначеними, тому що всі значення, набуті формою $f(x, y)$ у цьому випадку, мають той самий знак,

що й a. При додатному a форма $f(x,y)$ завжди визначена і називається додатно визначеною. Форми з від'ємним дискримінантом називаються невизначеними, тому що $f(x,y)$ набуває як додатних, так і від'ємних значень.

Якщо у формі $f(x,y)$ зробити заміну змінних $x = Au + Bv$, $y = Cu + Dv$, де A, B, C, D – цілі числа, які задовольняють умову $AD - BC = \pm 1$, то одержимо нову форму $g(u,v)$. Це зумовлено тим, що будь-якій парі цілих чисел x, y відповідає пара цілих чисел u і v. Тому кожне ціле число, виражене формою f, також виражається формою g, і навпаки. У цьому випадку вважається, що форми f і g є еквівалентними. Всі форми, еквівалентні даній, утворюють клас еквівалентності [f] [8]. Кількість таких класів для форм із фіксованим дискримінантом D є скінченною [9]. Виявляється, що у випадку додатно визначених форм у кожному класі еквівалентності існує єдина форма $ax^2 + bxy + cy^2$ з такими коефіцієнтами a, b, c, що виконується або $-a < b \le a < c$, або $0 \le b \le a = c$ [10]. Така форма називається приведеною формою даного класу еквівалентності [11]. Приведена форма використовується як стандартний представник власного класу еквівалентності.

1.6.2. Приведення алгебраїчних форм

Однією з основних задач, яка вирішена у найпростішому випадку, є знаходження приведеної форми, еквівалентної даній формі. Сама процедура знаходження називається приведенням.

У випадку невизначених форм неможливо визначити нерівності, які повинні задовольняти коефіцієнти лише однієї форми з кожного класу. Однак існують нерівності, які задовольняє деяке скінченне число форм у кожному класі. Усі вони називаються приведеними формами.

Визначені і невизначені форми відрізняються також тим, що будь-яка визначена форма представляє (якщо це можливо) ціле число тільки скінченною кількістю способів.

Кількість представлень цілого числа невизначеною формою завжди або дорівнює нулю, або є нескінченною [12]. Це пояснюється тим, що, на відміну від визначених форм, невизначені мають нескінченну множину "автоморфізмів", тобто підстановок вигляду

$$x = Au + Bv,\ y = Cu + Dv, \tag{1.10}$$

у яких форма $f(x,y)$ залишається інваріантною. Властивість інваріантності полягає у тому, що $f(x,y) = f(u,v)$. Ці автоморфізми можна повністю описати в термінах розв'язання рівняння Пелля $z^2 + \Delta w^2 = 4$, де Δ – дискримінант форми f.

Окремі результати, пов'язані з представленням цілих чисел квадратичними формами [13], отримали Лагранж, Лежандр, Гаус та ін. Так, Ферма показав, що кожне просте число вигляду $8n+1$ або $8n+3$ подається формою $x^2 + 2y^2$, кожне просте число вигляду $3n+1$ – формою $x^2 + 3y^2$, а також, що не існує простого числа вигляду $3n-1$, яке подається формою $x^2 + 3y^2$.

Якщо квадатричну форму позначати тріадою (a,b,c), то для практичних досліджень справедливі такі дві теореми.

Теорема 1.3. Якщо число M можна подати формою (a,b,c) при взаємно простих значеннях змінних, то дискримінант Δ цієї форми є квадратичним залишком числа M.

Теорема 1.4. Якщо дискримінант Δ форми (a,b,c) є від'ємним, то існує хоч би одне подання заданого числа M формою (a,b,c), якщо x , y – взаємно прості.

1.7. Проблема шифрування зображень алгоритмом RSA

1.7.1. Розвиток симетричних алгоритмів для організації криптографічного захисту шифрування зображень

Традиційні симетричні методи шифрування зображень (наприклад, DES, IDEA, AES) не підходять для шифрування зображень, оскільки вони не забезпечують достатнього рівня зашумленості зашифрованого зображення. Це викликало гостру необхідність у нових алгоритмах шифрування зображень, зокрема і таких, які належать до класу симетричних.

Враховуючи високу чутливість до вихідних значень та випадковості параметрів, криптосистеми на основі хаотичних систем стали базисом для нових досліджень. Алгоритми на основі хаотичних систем виконуються в два етапи: етап зашумлення та етап заміни. На етапі зашумлення позиції пікселів оригінального зображення змінюються за допомогою хаотичних послідовностей чи деяких матриць трансформації, наприклад, трансформації Арнольда тощо. Ці алгоритми зашумлення прості в реалізації та ефективні в шифруванні.

Однак, оскільки наведені алгоритми лише змінюють позицію пікселя, але не його значення, гістограма зашифрованого зображення ідентична гістограмі оригінального зображення. Таку заміну можна усунути за допомогою методів статистичного аналізу. На етапі заміни значення пікселів змінюються хаотичними послідовностями. Більшість таких методів шифрування реалізовано хаотичною послідовністю, згенерованою однією хаотичною картою та значенням відтінку сірого пікселя оригінального зображення. У поєднанні з методом зашумлення метод заміни міг би привести до вищої криптографічної стійкості, однак результат шифрування є незадовільним. Використання однієї хаотичної карти для шифрування зображень призводить до низької криптографічної стійкості та вузького діапазону ключів.

Існують хаотичні алгоритми шифрування зображень, які ґрунтуються на дисперсійному аналізі. Запропоновані алгоритми забезпечують достаньо високий рівень зашумлення, однак діапазон ключів шифрування залишається вузьким. Також часто пропонують використовувати для шифрування багатовимірні хаотичні карти для шифрування зображень. Цей підхід забезпечив високий рівень криптографічної стійкості та не вимагає значних апаратних затрат.

1.7.2. Проблема шифрування зображень алгоритмом RSA

Метод RSA працює значно повільніше, ніж симетричні алгоритми. Тому на практиці для підвищення швидкості шифрування значення e вибирається невеликим, зазвичай 3, 17 або 65537 [10]. Ці числа у двійковому вигляді містять тільки по дві одиниці, що зменшує кількість необхідних операцій множення при піднесенні до степеня. Наприклад, для піднесення числа до степеня 17 потрібно виконати лише 5 операцій множення.

Але, як зазначалось вище, вибір малого значення e може призвести до розкриття повідомлення, якщо воно надсилається відразу декільком адресатам.

Значення секретного коефіцієнта d повинне бути достатньо великим. У [11] показано, що якщо $z_2 < z_1 < 2z_2$ і $d < z^{\frac{1}{4}}/3$, то існує ефективний спосіб обчислити d по z і e. Однак, якщо значення e вибирається невеликим, то d отримується достатньо великим, і проблема нівелюється.

Використання алгоритму шифрування RSA як найстійкішого до несанкціонованого дешифрування кодованих сигналів, стосовно зображень, які дозволяють строго виділяти контури, не дає задовільних результатів – на зашифрованих зображеннях проступають основні контури вихідного зображення. Тобто виникає ефект неповного зашумлення зображення.

Цей ефект зумовлений логікою математичного перетворення – оператора шифрування алгоритму RSA, який має вигляд:

$$c'_{i,j} = c^{e}_{i,j} \bmod z, \tag{1.11}$$

де $c_{i,j}$ – значення функції інтесивності (або яскравості у випадку рисунків у форматі ”градацій сірого кольору”) піксела з координатами (i, j); $c'_{i,j}$ — зашифроване значення функції інтенсивності піксела з координатами (i, j); mod – функція математичного ділення за модулем; z – другий операнд (ключ шифрування) функції mod.

“Зашумленість” зображення (тобто шифрування) в алгоритмі RSA дає математична операція mod. На рис. 1.1 наведено просторовий розподіл функції mod на проміжках $z \in [1;100]$ та $c_{i,j} \in [1;50]$ з кроком дискретизації 1 у напрямах z та $c_{i,j}$ (на рис. 1.1 c).

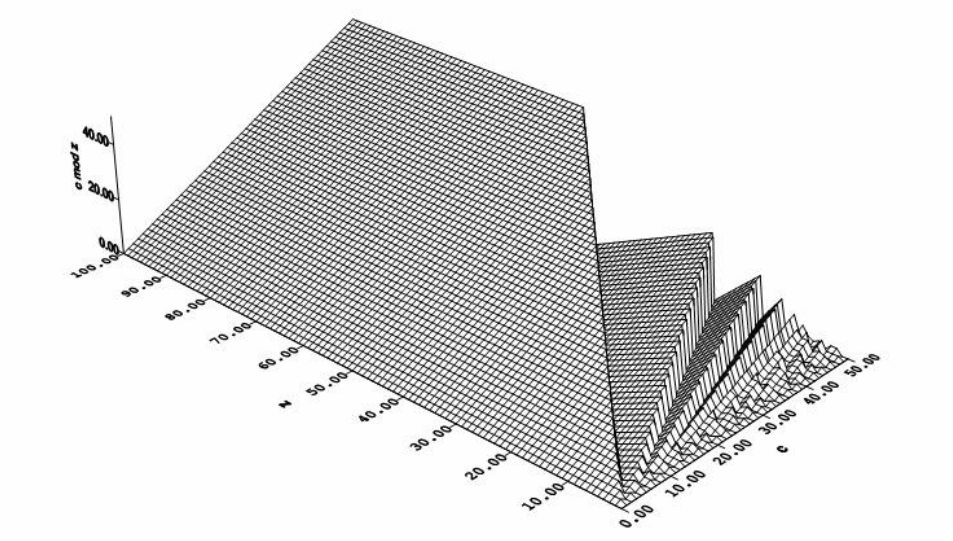
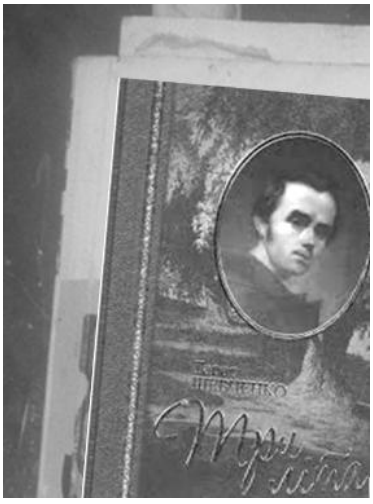


Рис. 1.1. Просторовий розподіл функції *mod*

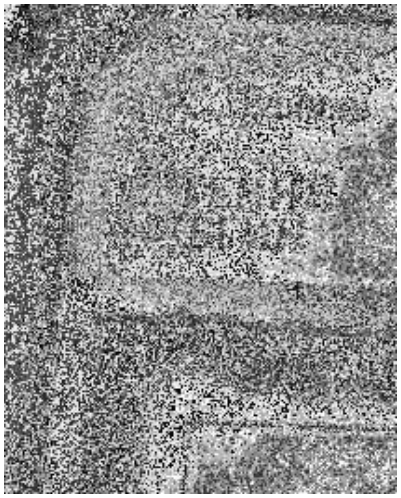
Як можна побачити з рис. 1.1, функція mod при фіксованому другому операнді z має значні однорідні області, площа яких залежить від співвідношення z до $c_{i,j}$ (див. на рис. 1.1 напрямок z, при фіксованому $c_{i,j} \le z$). А тому при близьких значеннях $c_{i,j}$ отримуємо близькі значення $c'_{i,j}$. Ця властивість дає зашумленість.

З іншого боку, якщо значення $c_{i,j}$ є різко флуктуаційними (що властиво для зображень, які дозволяють дуже чітко виділяти контури), то при фіксованому z можуть виникати флуктуації у значеннях $c'_{i,j}$ (на рис. 1.1 напрям c, при фіксованому z). Це призводить до того, що на зашифрованому зображенні можуть проступати контури вихідного зображення.

Тому отримане в результаті шифрування за алгоритмом RSA зображення не можна вважати повністю зашифрованим. Це пояснюється тим, що несанкціоноване дешифрування такого зображення, окрім традиційного злому самого алгоритму, допускає операції над цим зображенням традиційними засобами обробки зображень, зокрема фільтрації, реконструкції та ін. Незважаючи на те, що другий спосіб не дає повного відтворення вихідного зображення, за його допомогою несанкціоновано отримати певну інформацію із зашифрованого зображення видається цілком можливим.



а



Б

Рис. 1.2. Результат шифрування зображення методом RSA:
а – вихідне зображення; б – зашифроване зображення

На рис. 1.2. наведено вихідне зображення та те саме зображення, зашифроване класичним алгоритмом RSA. На зашифрованому зображенні можна побачити основні контури вхідного зображення.

Отже задача модифікації алгоритму RSA для випадку зображень є актуальною. Тому метою дослідження є модифікація алгоритму шифрування даних RSA так, щоб модифікований алгоритм:

- володів криптографічною стійкістю, не меншою за стійкість алгоритму RSA;
- забезпечував повне зашумлення зашифрованого зображення для того, щоб унеможливити отримання із зашифрованого зображення будь-якої інформації методами цифрової обробки зображень).

Потрібно зазначити, що отримані у роботі результати належать виключно до асиметричної схеми шифрування. Зменшення вимог до значень простих чисел, відстані між ними дають можливість полегшити їх використання у телекомунікаційних системах.

Використання алгебраїчних форм дозволяє вважати отримані методи криптографічного захисту належними до класу асиметричного шифрування на основі RSA із доповненням. Проте треба пам’ятати, що технологія доповнення є неявною, оскільки ґрунтується на інтегральному поєднанні в одному процесі елементів алгебраїчних форм та методу RSA.

Висновки до першого розділу

В даному розділі проведений аналіз методів криптографічного захисту, на основі теоретичних досліджень встановлено, що використання методу шифрування RSA для захисту цифрових зображень будь-якого типу не дає задовільних результатів, оскільки на зашифрованому зображенні проступають контури вихідного зображення. Наявність контурів на зашифрованому зображенні дозволяє здійснити криптографічну атаку без злому самого методу RSA – для отримання інформативності зображень може бути достатнім використання методів традиційної фільтрації зображень.

РОЗДІЛ 2. ТЕХНОЛОГІЇ КРИПТОГРАФІЧНОГО ЗАХИСТУ ГРАФІЧНИХ ФАЙЛІВ ТА ЗОБРАЖЕНЬ

2.1. Опис процесу модифікації методу RSA

Нехай задано цифрове зображення *P*. Його розмірність визначається параметрами: *h* – висота – кількість пікселів у вертикальному напрямі, та *l* – довжина – кількість пікселів у горизонтальному. Це зображення можна розглядати як матрицю пікселів $\mathrm{P}_{l,h}$, якій відповідає матриця значень функції інтенсивності $\mathrm{C}_{\mathrm{P}_{l,h}}$:

$$P = \mathrm{P}_{l,h} = \left[\, pxl_{ij} \,\right]_{1\le i\le n(l),1\le j\le m(h)} \to \mathrm{C}_{\mathrm{P}_{l,h}} = \left[\, c_{ij} \,\right]_{1\le i\le n(l),1\le j\le m(h)}, \qquad (1.12)$$

де pxl_{ij} – піксел з кординатами (*i*, *j*) дискретизованого зображення *P*; *n* та *m* – кількості пікселів у напрямках *l* та *h* відповідно.

Сформуємо потік значень функції інтенсивності “склеюванням” векторів значень кольорів усіх рядків зображення, яке підлягає шифруванню. Довжина потоку дорівнює $n \times m$. Потім розбиваємо потік на відрізки C_i завдовжки α, яке не обов’язково повинно бути кратним *n* чи *m*. Шифрування здійснюється за схемою зчеплення послідовних блоків шифру (рис.2.1).

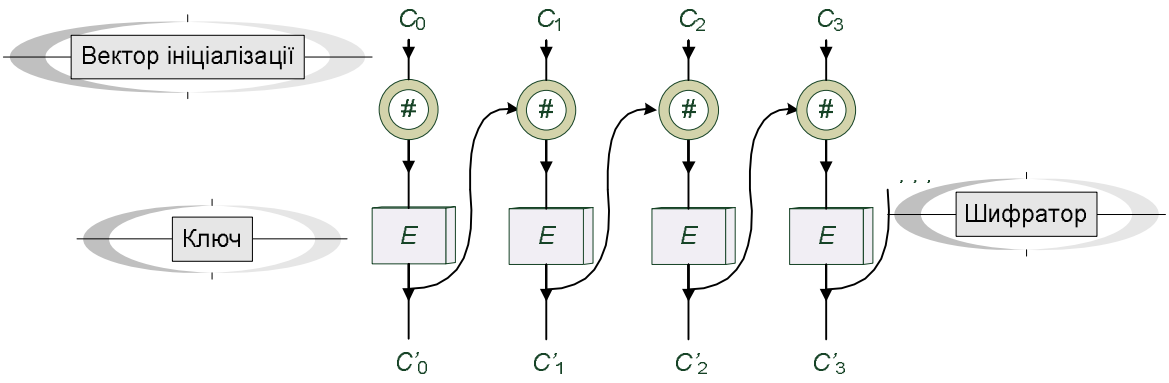


Рис. 2.1. Схема шифрування

Шифрування має такі етапи:

1. Перший блок C_0 сумується за модулем 2 із випадковим вектором ініціалізації і шифрується за алгоритмом RSA. Вектор ініціалізації використовується для ускладнення викриття шифру. Оскільки RSA має достатню криптографічну стійкість, то вектор ініціалізації можна не враховувати. Проте з використанням вектора ініціалізації можна зменшити значення ключів алгоритму RSA.
2. Кожен блок C_{i+1}, сумується за модулем 2 із першими α байтами зашифрованої послідовності C_i', отриманої на попередньому кроці.

Дешифрування здійснюватимемо за схемою, наведеною на рис. 2.2 за такими етапами:

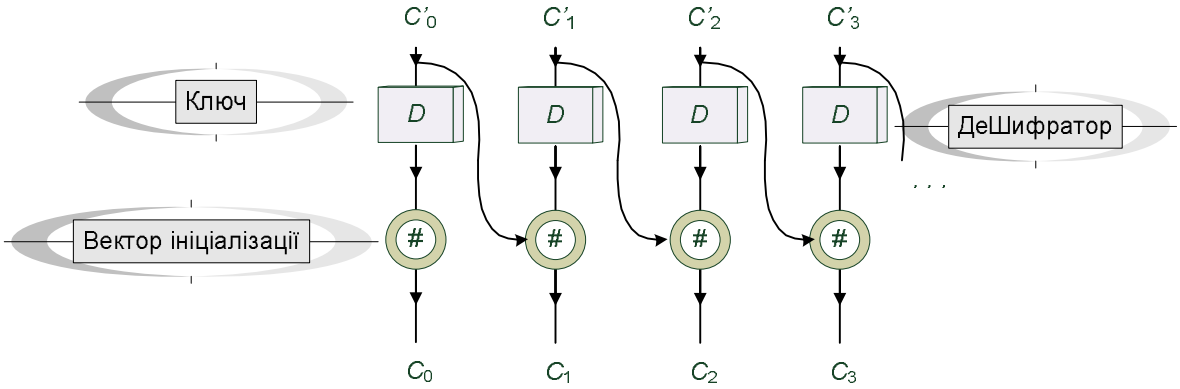


Рис. 2.2. Схема дешифрування

1. Дешифрується блок C_0', і результат сумується за модулем 2 із вектором ініціалізації.
2. Кожен наступний блок C_{i+1}' дешифровується за алгоритмом RSA, і результат сумується за модулем 2 із першими α байтами зашифрованого блоку C_i'.

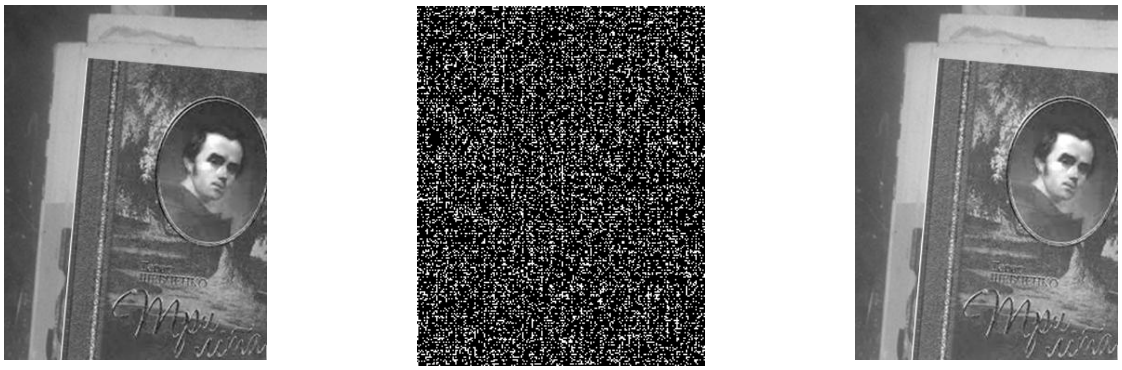
На рис. 2.3 наведено приклад застосування розробленого методу. Параметри зображення є такими: тип зображення – монохромне однобайтове зображення (у градаціях сірого кольору), розмір – 211×265 пікселів.

Стійкість до несанкціонованого дешифрування даного алгоритму повністю відповідає стійкості алгоритму RSA. При цьому, як показали результати практичного використання описаної модифікації RSA, у зашифрованому зображенні вдалось досягти повної зашумленості й уникнути існування виділених областей однорідності.

Параметри потоку є відкритою інформацією і у комунікаційному сеансі можуть передаватись разом із відкритою частиною криптографічного ключа.

За потреби підвищення криптографічної стійкості модифікації параметри потоку можуть бути також зашифровані і передані окремою транзакцією у комунікаційному сеансі.

Оскільки виявити контури вихідного зображення стає неможливим, то це нівелює використання традиційних засобів обробки зображень, наприклад, методів фільтрації чи виділення контурів, для несанкціонованого отримання інформації із зашифрованого зображення.



а б в

Рис. 2.3. Результат використання потокової модифікації: а –оригінальне зображення; б – зашифроване зображення; в – дешифроване зображення

Для засвідчення повної ідентичності оригінального та дешифрованого зображення використовувалась міра подібності, побудована на основі коефіцієнта подібності Серенсена [11]. Тобто предикат компараторної операції стосовно оригінального і дешифрованого зображення, побудований на основі

метрики Серенсена, повертає істинне значення. Отже, можна констатувати, що отримана модифікація алгоритму RSA є методом криптографічного захисту без інформаційних втрат.

Потокова модифікація алгоритму RSA орієнтована на зображення, глибина кольору якого не перевищує чотирьох розрядів. Це пояснюється тим, що отримана модифікація не допускає розпаралелювання обчислень без додаткових витрат оперативної пам'яті, оскільки байт розглядається як атомарний блок пам'яті.

Залучення додаткових витрат пам'яті дозволяє також використовувати цей метод стосовно зображень із будь-якими значеннями глибини кольору. Проте ці витрати можуть призвести до проблем з апаратною реалізацією потокової модифікації. Особливо це помітно у випадку розроблення вбудованих систем чи криптографічних машин пришвидшеного часу виконання або режиму реального часу.

2.2. Топологічна модифікація методу RSA

2.2.1. Схема модифікації алгоритму RSA

Нехай задано зображення P (1.12) розмірністю $h \times l$ пікселів. Визначальним у топологічній модифікації є формування топології [12] $\Im$ на просторі значення функції інтенсивності зображення прямокутними областями у напрямку h. Така топологія $\Im$ і скінченне покриття $\Sigma \in \Im$ прямокутними областями σ_s є можливі, оскільки плоске зображення є компактом [13].

Множина, яка складається із геометричних розмірів елементів покриття, називається параметрами покриття.

Зазначимо, що вибір напрямку і параметрів розбиття не є принциповими, проте можуть бути предметом подальших досліджень. Використання як елемента розбиття прямокутника є найзручнішим у практичній реалізації

топологічної модифікації. Тоді як використання елемента розбиття, наприклад, геометричного круга не лише ускладнить програмний код, а й призведе до проблеми покриття граничних областей. Ці області не зможуть бути ефективно (без виходу за границі зображення) закритими цією фігурою. Тому на зашифрованому зображенні можливі прояви різноманітних крайових ефектів.

Використання двох і більше фігур для формування розбиття збільшить набір параметрів, які разом із криптографічними ключами є необхідними для успішної реалізації операції дешифрування. Тому при виборі елементів розбиття треба враховувати їх геометрію, значення якої, своєю чергою, безпосередню впливає на розмірність множини параметрів розбиття.

Для формування розбиття відрізок m розіб'ємо на α частин - сегментів довільної довжини

$$I = [1, m] = [1, m_1] \cup]m_1, m_2] \cup \ldots \cup]m_{\alpha-1}, m], \tag{1.13}$$

але таких, для яких виконується умова диз'юнктивності розбиття

$$I = [1, m] = \bigcup_{s=1}^{\alpha} I_s, \quad \forall s_1, s_2 : \qquad I_{s_1} \bigcap I_{s_2} = \varnothing. \tag{1.14}$$

Тут $I_s =]m_{s-1}, m_s], \quad s = \overline{1, \alpha}, \ m_0 = 0, \ m_\alpha = m$ з формули (1.13).

У напрямку h на площині зображення P за розбиттям (1.13) сформуємо покриття з α прямокутних областей пікселів σ_s таких, що існує

$$\sigma_s = \left\{ pxl_{ij} \,|\, 1 \le i \le n, \ j \in I_s \right\}, \quad s = \overline{1, \alpha} \tag{1.15}$$

Очевидно, що для набору $\sigma = \{\sigma_s\}_{s=\overline{1,\alpha}}$ повинна виконуватись така умова:

$$\forall s_1, s_2 \in [1;\alpha]: \sigma_{s_1} \subseteq \sigma_{s_2} \wedge \sigma_\alpha = P\,. \tag{1.16}$$

Якщо подібно до випадку (1.12) у (1.15) перейти до значень функції інтенсивності, отримаємо покриття області кольорів $\mathbf{C}_{\mathbf{P}_{l,h}}$ зображення P (покриття, яке індуковане неперервним оператором (2.1)). Це покриття можна записати у вигляді

$$\Sigma_s = \left\{ c_{ij} \mid pxl_{ij} \in \sigma_s \right\},\ s = \overline{1,\alpha}\,. \tag{1.17}$$

Очевидно, що останній елемент розбиття Σ є найбільшим за розміром з усіх елементів розбиття і збігається із набором значень функції інтенсивності зображення P

$$\Sigma_\alpha = \mathbf{C}_{\mathbf{P}_{l,h}} \tag{1.18}$$

Умова диз'юнктивності, яка існувала у випадку координатного розбиття (2.3), для розбиття у колірній області (2.4) не вимагається.

Кожен елемент матриці Σ_{σ_s} кодуємо за формулою (1.11), і тоді для випадку зображення P можна записати таке правило

$$\begin{gathered} \forall i \in [1;n],\ j \in [1;m] \quad \exists s \in [1;\alpha]: \\ c_{i,j} \in \Sigma_s \quad \tilde{c}_{i,j}^{s} = \left(c_{i,j}\right)^e \bmod z. \end{gathered} \tag{1.19}$$

Тут $\tilde{c}_{i,j}^{s}$ розглядається як зашифроване значення $c_{i,j}$, яке є елементом множини Σ_s топології $\mathfrak{I}$.

На наступному кроці, тобто шифруванні області σ_{s+1}, множина Σ_s вже розглядається підмножиною множини Σ_{s+1}: $\Sigma_s \subseteq \Sigma_{s+1}$. Це означає, що частина значень функції інтенсивності, які підлягають процедурі шифрування на поточному кроці, будуть зашифрованими значеннями з попереднього кроку (ітераційного циклу).

У результаті операцію визначення шифрованих значень елементів матриці Σ_{σ_s} можна записати у вигляді

$$\begin{cases} \forall i \in \left[1; n\right], j \in \left[1; m\right] \quad \exists s \in \left[1; \alpha\right]: \\ \tilde{c}_{i,j}^{s} = \left(c_{i,j}\right)^{e} \bmod z, \quad \text{if} \quad c_{i,j} \in \Sigma_s \setminus \Sigma_{s-1}; \\ \tilde{c}_{i,j}^{s} = \left(\tilde{c}_{i,j}^{s-1}\right)^{e} \bmod z, \quad \text{if} \quad c_{i,j} \notin \Sigma_s \setminus \Sigma_{s-1}. \end{cases} \tag{1.20}$$

Після виконання операції (1.20) для кожної множини Σ_s буде побудовано $\tilde{\Sigma}_{\alpha}$ – матрицю зашифрованих значень матриці Σ_s. При цьому, подібно до (1.18), отримаємо множину $\tilde{\Sigma}_{\alpha} = \tilde{\mathbf{C}}_{\mathbf{P}_{l,h}}$ – матрицю зашифрованих значень функції інтенсивності $\mathbf{C}_{\mathbf{P}_{l,h}}$.

Особливістю матриці $\tilde{\mathbf{C}}_{\mathbf{P}_{l,h}}$ є те, що за винятком значень $c_{i,j}$, які належать $\Sigma_{\alpha} \setminus \Sigma_{\alpha-1}$, усі решта значень функції інтенсивності $c_{i,j}$ шифруватимуться за виразом (1.20) більше одного разу. Так, наприклад, найбільше, тобто α разів, шифруватимуться значення $c_{i,j}$, які належать множині Σ_1.

Використання ітераційної технології має одну особливість, яка в окремих випадках може стати суттєвим недоліком розробленої модифікації. Оскільки функція mod алгоритму RSA у випадку зображень може розглядатись як деякий згладжувальний фільтр, то завдяки повторному застосуванню цієї операції рівень зашумлення зростатиме. Очевидно, що в нескінченності ітеративне

застосування операції mod призведе до однотонності зашифрованого зображення, і дешифрувати зображення стане неможливо. Тобто операція шифрування стане незворотною.

2.2.2. Дешифрування при топологічній модифікації методу RSA

Процедура дешифрування ґрунтується на інформації про топологічні параметри та передбачає здійснення таких кроків:

1. Подібно до побудови Σ_s для матриці значень функції інтенсивності $\tilde{\mathbf{C}}_{\mathbf{P}_{l,h}}$ зашифрованого зображення $\tilde{P}$ треба сформувати набір областей значень матриці $\tilde{\tilde{\Sigma}}_s$

$$\tilde{\mathbf{C}}_{\mathbf{P}_{l,h}} = \left[\tilde{\tilde{\Sigma}}_s \right]_{1 \le s \le \alpha}. \tag{1.21}$$

Це означає, що спосіб визначення Σ_s і α, тобто параметри топологічного розбиття, повинні бути відомими до початку процедури дешифрування.

2. Кожну матрицю $\tilde{\tilde{\Sigma}}_s$, починаючи з $s = \alpha$, дешифруємо за формулою дешифрування методу RSA

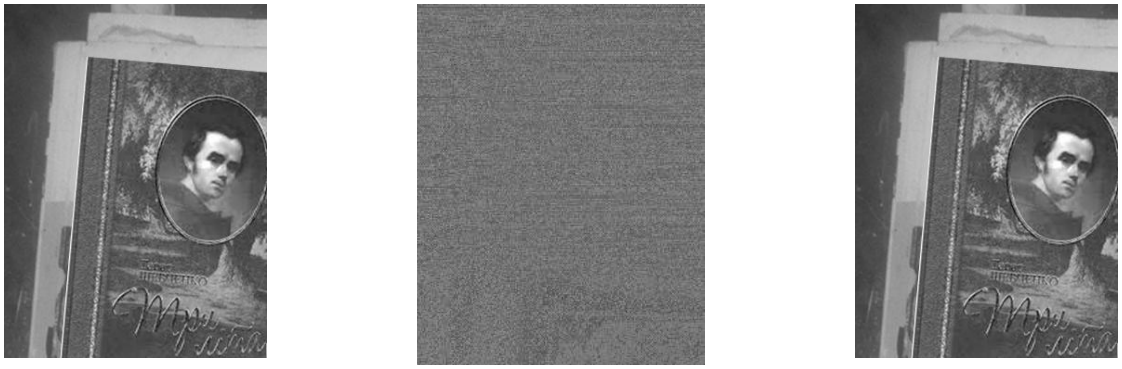
$$\begin{cases} \forall i \in \left[1; n\right], j \in I_s \quad \exists s \in \left[1; \alpha\right]: \\ c_{i,j}^s = \left(c_{i,j}^{s+1}\right)^d \bmod z, \quad \text{if} \quad \tilde{c}_{i,j} \in \tilde{\tilde{\Sigma}}_s \bigcap \tilde{\tilde{\Sigma}}_{s+1}; \\ c_{i,j}^s = \left(\tilde{c}_{i,j}^{s+1}\right)^d \bmod z, \quad \text{if} \quad \tilde{c}_{i,j} \notin \tilde{\tilde{\Sigma}}_s \bigcap \tilde{\tilde{\Sigma}}_{s+1}. \end{cases} \tag{1.22}$$

Тут d – число, яке задовольняє тотожність

$$ed \equiv 1\left(\bmod (z_1 - 1)(z_2 - 1)\right) \tag{1.23}$$

У результаті дешифрування усіх матриць $\tilde{\Sigma}_s$ після обчислення Σ_1 отримуємо остаточний вид матриці Σ_α, яка і визначає піксельну матрицю $\mathbf{C}_{\mathbf{P}_{l,h}}$. При цьому треба пам'ятати, що починаючи з $s = 2$, значення $\tilde{c}^s_{i,j}$ підлягатимуть процедурі дешифрування два і більше разів.

На рис. 2.4 наведено приклад застосування розробленого методу. Параметри вхідного зображення є аналогічні до наведеного на рис. 2.3 зображення. Мінімальний розмір прямокутної області, на основі якої формувалась топологія, – 211×5 пікселів.



а б в

Рис. 2.4. Результат використання топологічного підходу: а –оригінальне зображення; б – зашифроване зображення; в – дешифроване зображення

Як можна побачити з наведених результатів, рівень зашумлення зріс. Але також зростають обчислювальні витрати, яку зумовлені процедурами формування топології розбиття та циклічного шифрування областей розбиття. Циклічність процедури шифрування визначається співвідношенням розмірів початкового елемента топологічного розбиття (параметра розбиття) до розмірів зображення загалом.

Ідентичність оригінального і дешифрованого зображень перевірялась за мірою подібності, яка побудована на основі коефіцієнта подібності Жаккара. Вибір такого коефіцієнта для побудови компараторного предиката зумовлений

тим, що метрика Жакара будується на основі бінарних значень координат векторів порівняння.

Основною перевагою методу на основі формування топології є те, що він дає змогу шифрувати зображення із малими значеннями ключів методу RSA. Своєю чергою, малі значення ключів дозволяють в обчислювальному процесі не вийти за межі розрядної сітки, яка визначається малими значеннями глибини кольору. Тому цей метод рекомендується для випадків зображень із значеннями глибини кольору у 1, 2 або 4 розряди байта представлення значень функції інтенсивності.

На відміну від випадку потокової модифікації алгоритму RSA, топологічна модифікація надає можливість часткового використання методів розпаралелювання виконання коду процедур шифрування та дешифрування. Частковість розпаралелювання означає можливість одночасної роботи, наприклад, функцій процесу шифрування, але лише у випадку наборів зображень із однаковою топологією розбиття. Для випадку окремого зображення розпаралелювання обчислювальних процедур є неможливим, оскільки викликано атомарністю мінімальної ділянки пам’яті.

Недоліком методу є витрати на формування топологій та надлишковість математичних операцій внаслідок ітеративності процедур шифрування/дешифрування. Тому для випадку зображень із значенням глибини кольору у 1–4 байти ефективність використання цієї модифікації нівелюється зростанням ресурсних витрат, насамперед, зростанням обсягів оперативної пам’яті у процесах криптографічного захисту.

Подібно до випадку потокової модифікації, параметри розбиття зображення топологічної модифікації алгоритму RSA у телекомунікаційних сеансах можна розглядати як частину відкритого ключа. Тоді криптографічна стійкість топологічної модифікації дорівнює стійкості самого методу RSA. Таке значення є мінімальним значення криптографічної стійкості, якої можна досягнути у топологічній модифікації.

У загальному випадку топологічні параметри можна зашифрувати із власними значеннями криптографічних ключів і передати окремою транзакцією під час мережевого сеансу. Тоді криптографічна стійкість топологічної модифікації за наведеним способом організації захисту параметрів розбиття під час мережевого сеансу буде удвічі більшою від криптографічної стійкості за методом RSA. Отримане значення криптографічної стійкості є максимальним значенням, яке можна отримати у топологічній модифікації.

Загалом комбінацій використання параметрів топологічної модифікації і зображень у комунікаційних сеансах може бути багато. Наприклад, шифрування параметрів розбиття і використання стеганографічних засобів для їх занесення у незашифроване зображення. Але у цьому випадку міра подібності оригінального і дешифрованого зображення може бути ненульовою. Це означає, що у результаті сумісного використання процедур шифрування/дешифрування та стеганографічних засобів приховування інформації можуть виникати інформаційні втрати зображень. Проте в усіх комбінаціях використання параметрів розбиття криптографічна стійкість знаходитиметься в діапазоні [K; 2K], де K – криптографічна стійкість методу RSA.

2.3. Криптографічний захист напівтонових зображень

Основним недоліком потокової та топологічної модифікацій алгоритму RSA є неможливість розпаралелювання обробки і ресурсні витрати. Перевагою цих модифікацій є невихід обчислювального процесу за межі одного байта (границі, яка визначається значенням глибини кольору) і вирішення проблеми уникнення контурів на зашифрованому зображенні.

Для зменшення ресурсних витрат за умови забезпечення невиходу за межі розрядної сітки у випадку зображень із глибиною кольору у 1 та 2 байти пропонується модифікація алгоритму, отриманого за методом RSA, на основі афінних перетворень.

2.3.1. Лінійний афінний шифр

В основу модифікації алгоритму RSA з використанням афінного перетворення покладено лінійний афінний шифр – шифр афінної підстановки, який має вигляд [26, 28]:

$$y = ax + b \left(\mathrm{mod}\, m\right). \tag{1.24}$$

Тут ключем є пара (a,b), $0 < a \le m-1, 0 \le b \le m-1$, причому a повинно бути взаємно простим з m.

З рівняння (1.24) одержуємо

$$x = a'y + b' \left(\mathrm{mod} m\right), \tag{1.25}$$

де $a' = a^{-1} \bmod m$ – обернений до a елемент в кільці лишків за модулем m, $b' = -a'b$.

Рівність (1.25) має той самий вигляд, що й (1.24), отже, шифрування і дешифрування здійснюватимуться за однаковим алгоритмом, тільки з різними параметрами.

Умова взаємної простоти a з m потрібна для того, щоб існував обернений елемент a^{-1} і рівняння (1.24) за фіксованого y мало єдиний розв'язок x, тобто процедура дешифрування повинна бути однозначною. У протилежному випадку рівняння (1.25) має не єдиний розв'язок x або його взагалі не має.

Нехай y^* і y^{**} – перший і другий за частотою символи шифрованого тексту x^* і x^{**} – відповідно та, що найчастіше зустрічається і наступна за нею букви алфавіту.

Природно припустити, що при шифруванні елемент x^* перетвориться в елемент y^*, а елемент x^{**} – в елемент y^{**}. Тоді складемо систему рівнянь лінійного афінного шифру:

$$\begin{cases} y^* = ax^* + b \bmod m; \\ y^{**} = ax^{**} + b \bmod m. \end{cases} \tag{1.26}$$

Зазначимо, що в цій системі невідомими є елементи a і b, а елементи x і y є відомими.

За системою (1.26) отримаємо різницю елементів y^* і y^{**}:

$$y^* - y^{**} = a\left(x^* - x^{**}\right) \bmod m. \tag{1.27}$$

Якщо пари $x^* \leftrightarrow y^*$, $x^{**} \leftrightarrow y^{**}$ підібрано правильно, то рівняння (1.27) має розв'язок a. Знаючи значення a, за системою (1.26) знаходимо значення b. Якщо ж ці пари не знайдено, то рівняння (1.27) або не має розв'язку, або при всіх розв'язках (1.26), розшифровуючи, одержимо беззмістовний текст. Іншими словами, процедура дешифрування є неможливою.

2.3.2. Бінарне афінне перетворення

Наведемо означення бінарного афінного перетворення евклідової площини в декартових координатах.

Перетворення евклідової площини називається афінним, якщо це перетворення відображає кожну пряму в пряму.

Бінарне афінне перетворення площини в декартових координатах має вигляд:

$$\begin{aligned} x' &= a_1 x + b_1 y + d_1, \\ y' &= a_2 x + b_2 y + d_2; \end{aligned} \tag{1.28}$$

де

$$\delta = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \neq 0 . \tag{1.29}$$

Зважаючи на (1.29), для невиродженої лінійної системи (1.28) існує обернене перетворення. Тому можна отримати вирази для обчислення x і y

$$x = \frac{\Delta_x}{\delta},\ y = \frac{\Delta_y}{\delta}, \tag{1.30}$$

де визначники Δ_x і Δ_y визначаються так

$$\Delta_x = \begin{vmatrix} x' - d_1 & b_1 \\ y' - d_2 & b_2 \end{vmatrix};\quad \Delta_y = \begin{vmatrix} a_1 & x' - d_1 \\ a_2 & y' - d_2 \end{vmatrix}. \tag{1.31}$$

2.3.3. Шифрування і дешифрування за рядками матриці зображення

Процедура шифрування зображення виглядає так:

- беруться два сусідні в рядку елементи матриці зображення (1.12) і визначаються значення x і y:

$$x = c_{i,j},\ y = \frac{c_{i,j+1} + c_{i,j}}{2},\ i = \overline{1,n},\ j = \overline{1,m-1};$$

- за формулою (1.28) значення x і y шифруємо з такими параметрами:

$$a_1 = b_2 = \left(z_1 + z_2\right)^e \left(\mathrm{mod}\, z\right), b_1 = a_2 = \left(z_1 - z_2\right)^d \left(\mathrm{mod}\, z\right),$$
$$d_1 = j \cdot j, d_2 = j \cdot j \cdot j, \; j = \overline{1,m};$$

- отримані зашифровані значення записуються на відповідні місця до матриці зашифрованого зображення.

  Дешифрування проводять у зворотній послідовності:

- беруться два сусідні в рядку елементи матриці зашифрованого зображення

$$x' = c'_{i,j}, \; y' = c'_{i,j+1}, \; i = \overline{1,n}, \; j = \overline{1,m-1};$$

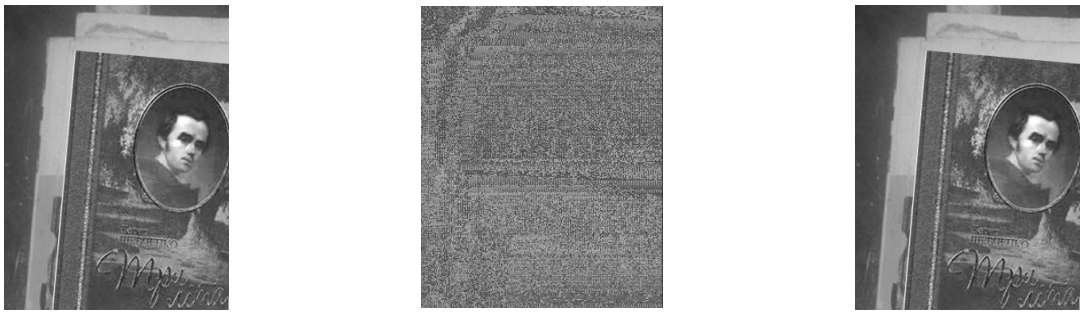
- значення x' та y' дешифруються за формулою (1.30) з урахуванням формул (1.29) та (1.31). У результаті за лінійною системою (2.18) отримаємо значення x і y;
- отримані значення x і y записують на відповідні місця в матрицю дешифрованого зображення:

$$c_{i,j} = x, c_{i,j+1} = 2y - c_{i,j}, \; i = \overline{1,n}, \; j = \overline{1,m-1}.$$

На рис. 2.5 наведено приклад застосування розробленого методу. Параметри зображення: напівтонове зображення (у градаціях сірого кольору) із глибиною кольору у 2 байти, розмір – 211×265 пікселів.

Як можна побачити з наведених результатів, рівень зашумлення є достатнім для уникнення контурів. При цьому розрядність обчислювального процесу не вийшла за межі трьох байтів.

Ідентичність оригінального і дешифрованого зображень перевіряли за мірою подібності, побудованою на основі p-метрики (p = 2) [25].



А б В

Рис. 2.5. Результат шифрування бінарним афінним перетворенням: а – оригінальне зображення; б – зашифроване; в – дешифроване зображення

Особливістю поданої модифікації методу RSA є те, що вона, подібно до топологічної модифікації, забезпечує не вихід за межі розрядної сітки, але при цьому зменшуються витрати обчислювального процесу. Проте, у випадку топологічної модифікації можна використовувати не дуже великі значення чисел ключа RSA, але більші за випадок топологічної модифікації і менші, ніж у випадку класичного методу RSA. Така вимога виникає внаслідок того, що значення глибини кольору знаходиться у межах 1–2 байтів. У випадку повноколірних зображень за тих самих значень ключа бінарне афінне перетворення вже не вирішує проблему контурів. Тому значення простих чисел для побудови ключів треба збільшувати. У результаті можливий вихід за межі розрядної сітки, що свідчить про непридатність методу для випадку організації криптографічного захисту повноколірних зображень незалежно від того яке значення глибини кольору використовується: 24 чи 32 розряди.

Висновки до другого розділу

В розділі приведені модифікації алгоритму RSA призначені для шифрування зображень із глибиною кольору 1 біт, 2 біти, 4 біти, 1 байт та 2 байти. Усі алгоритми шифрування повністю нівелюють проблему контурів. При цьому стійкість даних модифікацій дорівнює стійкості алгоритму RSA.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ПРОЦЕСУ ПІДВИЩЕННЯ КРИПТОГРАФІЧНОЇ СТІЙКОСТІ ЗАХИСТУ ЗОБРАЖЕНЬ

3.1. Використання квадратичних форм для підвищення стійкості шифрування

У випадку організації захисту повноколірних зображень із значенням глибини кольору у 3 байти довжини на один піксел зображень в оперативній пам’яті все рівно виділяється ділянка завдовжки у одне машинне слово. При цьому один байт у ділянці машинного слова не використовується. Більше того, завдяки механізму вирівнювання по границі машинного слова на зображеннях великої розмірності кількість невикористаних байтів зростає лінійно до розмірів зображення. А це породжує проблему неефективного використання оперативної пам’яті прикладного процесу.

У випадку класичного алгоритму RSA для виконання вимоги вибору великих значень простих чисел наявність невикористаного байта в машинному представленні значення функції інтенсивності піксела повно колірного зображення гарантує невихід обчислювального процесу за межі машинного слова. Проте один байт не дозволяє суттєво збільшити значення простих чисел.

Існують такі зображення, у яких нівелювання контурів за обмежень невиходу за межі машинного слова може не реалізовуватись вибором великих значень простих чисел. Тому існує потреба в розробленні універсального методу шифрування. Тобто такого, який запобігатиме прояву контурів за таких значень простих чисел, які гарантують невихід за межі машинного слова обчислювального процесу. Це означає, що метод повинен забезпечувати повне зашумлення зображення за малих значень простих чисел і володіти при цьому вищою криптографічною стійкістю порівняно із класичним методом RSA. Для вирішення цього завдання пропонується метод на основі сумісного

використання алгебраїчних квадратичних форм і модифікації методу RSA, яку наведено у п.2.3.2.

3.1.1. Схема використання квадратичних форм

Нехай задано довільні прості числа z_1, z_2. Згідно з алгоритмом RSA виконуються такі співвідношення

$$z = z_1 z_2\ ,\ \varphi(z) = (z_1 - 1)(z_2 - 1),$$
$$ed \equiv 1 (\mathrm{mod}\ \varphi(z)), e < \varphi(z), d < \varphi(z). \qquad (1.32)$$

Нехай A – ціле невід'ємне число. Виконаємо бінарне афінне перетворення

$$\begin{cases} u = \gamma e\ +\ \alpha d, \\ v\ = \alpha e\ + \gamma d; \end{cases} \qquad (1.33)$$

де γ, α – цілі числа, і будемо вимагати виконання такої конгруенції

$$(A^u)^v = A^{(\gamma e + \alpha d)(\alpha e + \gamma d)} \equiv A^1 (\mathrm{mod}\ z). \qquad (1.34)$$

Тоді за властивістю 1.3 для операндів перетворення (3.2) має виконуватись така конгруенція

$$(\gamma e + \alpha d)(\alpha e + \gamma d) \equiv 1 (\mathrm{mod}\ \varphi(z)) \qquad (1.35)$$

або

$$\gamma^2 + (e^2 + d^2)\alpha\gamma + \alpha^2 \equiv 1 (\mathrm{mod}\ \varphi(z)) \qquad (1.36)$$

Конгруенція (1.36), лівою частиною якої є квадратична форма, за теоремою (1.4) має хоча б один розв'язок у цілих, взаємно простих числах γ_1, α_1 . Підставивши знайдені значення в (1.33), отримаємо u_1, v_1, для яких виконується конгруенція (1.34).

Надалі розглянемо дві послідовності цілих чисел

$$e = u_0, u_1, u_2, \ldots, u_m, \ldots, \tag{1.37}$$

$$d = v_0, v_1, v_2, \ldots, v_m, \ldots, \tag{1.38}$$

На елементи цих послідовностей накладаємо умову: за будь-якого $i \geq 0$ для пари елементів u_i, v_i виконується умова виконання бінарного афінного перетворення (1.33)

$$\begin{cases} u_{i+1} = \gamma_{i+1} u_i + \alpha_{i+1} v_i, \\ v_{i+1} = \alpha_{i+1} u_i + \gamma_{i+1} v_i, \end{cases} \tag{1.39}$$

У системі (3.8) числа γ_{i+1}, α_{i+1} отримуються як деякий розв'язок відповідної конгруенції (1.36).

Внаслідок побудови елементів у послідовностях (3.6) та (3.7) справедливе таке твердження:

Твердження 3.1

Для кожного $k \geq 0$ $u_k v_k \equiv 1 (\mathrm{mod}\, \varphi(z))$ і для кожного натурального A виконується конгруенція

$$A^{u_k v_k} \equiv A^1 (\mathrm{mod}\ \varphi(z)) \tag{1.40}$$

На основі твердження 3.1. можна так модифікувати алгоритм шифрування RSA:

- таємний ключ треба вибирати таким, який складається з двох чисел: попереднього таємного ключа d і номера k пари двох чисел u_k, v_k , для яких виконується (1.40) і які при розшифровуванні отримуються з (1.39).

3.1.2. Аналіз криптографічної стійкості у випадку сумісного використання квадратичних форм та бінарних афінних перетворень

Отриману асиметричну криптосистему можна зламати, якщо вдасться сконструювати ефективний алгоритм визначення таємного ключа, знаючи відкритий. У випадку системи RSA це зводиться до задачі:

задано : значення e, z; $z = z_1 z_2$, $\left(e, \varphi(z)\right) = 1$;

знайти: значення d, що для всіх x виконується конгруенція $x^{ed} \equiv x \left(\mathrm{mod}\, z\right)$.

Ця задача зводиться до обчислення функції Ейлера $\varphi(z)$, що рівносильно факторизації числа n на два прості множники z_1 і z_2. Ця задача є надзвичайно обчислювально витратною за значення $z > 10^{250}$. Тому варто вибирати такі прості числа z_1 і z_2, кожне з яких має не менше сотні десяткових знаків.

З метою пришвидшення процедури шифрування пропонується, щоби значення e дорівнювало 3. Але використання такого малого значення числа e дає можливість швидко розкрити повідомлення у випадку, якщо це повідомлення буде розіслане декільком користувачам.

Застосовуючи описану бінарну модифікацію системи RSA, можна для різних користувачів надати по-різному зашифрований вхідний текст. Для цього необхідно вибрати з послідовностей (1.37) і (1.38) для кожного користувача тільки йому відомі номери k пари чисел u_k, v_k. Це значно підвищує криптографічну стійкість системи навіть за малих значень $e \geq 3$.

Якщо задати $\psi(z)$ – найменше спільне кратне чисел $z_1 - 1$ і $z_2 - 1$, то згідно із [10], за $\varphi(\psi(z)) - 1$ ітерацій систему RSA можна розкрити.

За отриманим методом криптографічну стійкість визначають так: нехай K – максимальне число елементів у послідовностях (1.37) і (1.38), $\tau(n) \geq 1$ – число розв'язків квадратичної конгруенції (1.36). Виберемо з кожної послідовності підмножину з $m \geq 1$ елементів. Підмножину з m елементів з множини, яка має K елементів, можна вибрати C_K^m способами. Це означає, що тільки за $\tau(z)\mathrm{C}_K^m(\varphi(\psi(z))-1)$ ітерацій можна розкрити зашифроване повідомлення бінарним модифікованим методом, що є значно більшим значенням, ніж в стандартному алгоритмі на основі методу RSA.

На рис.3.1 наведено приклад використання розробленого методу криптографічного захисту повноколірних зображень, побудованого на основі використання квадратичних форм і бінарних афінних перетворень. Використовувалось зображення із такими параметрами: тип зображення – повноколірне зображення із глибиною кольору у 3 байти (колірна система RGB), розмір – 211×265 пікселів.

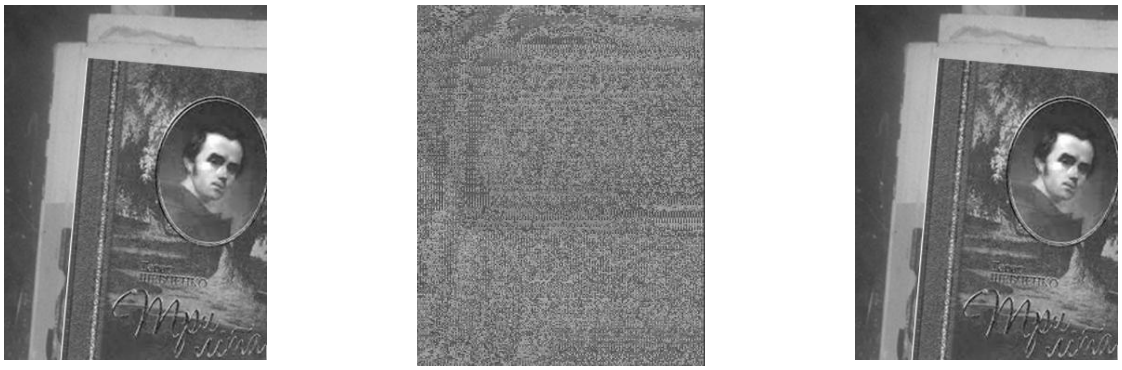
Ідентичність оригінального і дешифрованого зображень перевіряли за мірою подібності, яка побудована на основі p-метрики [33].

Розроблений метод застосовувався за кожним колірним каналом окремо. При цьому зайвий байт машинного слова, яке використовувалось для представлення значення глибини, забезпечує невихід обчислювального процесу за межі машинного слова на 32-х розрядних системах. Це означає, що четвертий байт використовувався як допоміжний в обчислювальному процесі процедур шифрування/дешифрування. Як можна побачити з результатів шифрування, отриманий рівень зашумленості зашифрованого зображення унеможливлює отримання будь-яких інформативних даних.

Варто зазначити, що відокремлений за колірними каналами (розрядність даних – один байт) процес шифрування є затратним щодо обчислювальних ресурсів: фактично його треба повторювати тричі. Проте обчислення в межах байта, забезпечують необхідність лише в одному додатковому байті. Так

гарантується невихід за межі машинного слова.

Використання одного байта у трьох обчислювальних процесах вимагає їх послідовного виконання. Це значно обмежує можливості розпаралелювання процедур шифрування та дешифрування.



а б в

Рис. 3.1. Результат шифрування та дешифрування повно колірного зображення (глибина кольору – 3 байти) з використанням квадратичних форм: а – оригінальне зображення; б – зашифроване зображення; в – дешифроване зображення

Очевидно, що алгоритм на основі описаного методу може бути застосований і до інтегральних цілих значень функції інтенсивності розрядністю 3 байти. Проте у цьому випадку максимально необхідна розрядність даних обчислюваного процесу становить 5 байтів.

Незважаючи на те, що на практиці за рахунок вибору коефіцієнтів можна обмежитись використанням 4 байтів, у загальному випадку потреби алгоритму становлять два машинні слова. Витрати в одне машинне слово пропорційно до розмірів зображення становлять сумарні витрати пам’яті обчислювального процесу, що є актуальним у апаратних реалізаціях процедур шифрування та дешифрування.

Зазначимо, що у випадку використання інтегрального підходу (маніпулювання чотирибайтовим знаковим цілим) до операцій із значень функції

інтенсивності можливості розпаралелювання процедур шифрування та дешифрування суттєво розширюються. З'являється можливість паралельної обробки різних областей зображення, що дозволяє розпочати комунікаційний сеанс до завершення криптографічних процедур. А такий спосіб передавання зображень у телекомунікаційних сеансах є основним засобом пришвидшення мережевого передавання зображень.

3.2. Шифрування елементами алгоритму RSA

Для випадку організації криптографічного захисту повноколірних зображень із глибиною кольору у 4 байти розроблено метод на основі використання тернарних афінних форм та елементів алгоритму RSA.

Особливістю описаного класу зображень є використання повного машинного слова для представлення значення функції інтенсивності у кожному пікселі. Це означає, що вже за малих значень простих чисел обчислювальний процес виходитиме за межі машинного слова, що призведе до збільшення витрат оперативної пам'яті.

Об'єктом захисту методу на основі тернарних форм є повноколірне зображення з глибиною кольору у 4 байти. Це не означає, що цей метод не можна використовувати для випадків зображень із значеннями глибини кольору 1, 2, 4 біти та 1, 2 та 3 байти. Але у випадку зображень із меншими значеннями глибини кольору треба зважати на те, що цей метод, мінімізовуючи витрати пам'яті у байтовому представленні значень функції інтенсивності, не забезпечує невихід за межі машинного слова. Тобто, найімовірніше, виникатимуть перевитрати оперативної пам'яті, які, у випадку великих за розміром зображень, можуть досягати великих значень. У зв'язку з цим розроблений метод позиціонуються саме для тих випадків, у яких уникнення перевитрати пам'яті не є принциповими.

3.2.1. Шифрування і дешифрування за рядками матриці зображення з використанням тернарних афінних форм

Наведемо означення тернарного аффінного перетворення евклідового простору в декартових координатах. Як відомо, перетворення евклідового простору називається афінним, якщо це перетворення відображає кожну площину в площину.

На відміну від бінарного, тернарне афінне перетворення евклідового простору в декартових координатах має такий вигляд:

$$\begin{aligned} x' &= a_1 x + b_1 y + d_1 z, \\ y' &= a_2 x + b_2 y + d_2 z, \\ z' &= a_3 x + b_3 y + d_3 z, \end{aligned} \tag{1.41}$$

де $\delta = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix} \neq 0$ – ненульовий визначник системи (3.10).

Оскільки система (3.10) є невиродженою (визначник матриці є ненульовим), то обернене до (1.41) перетворення також існує, і тоді розв’язки системи можна знайти за такими виразами

$$x = \frac{\Delta_x}{\delta}, y = \frac{\Delta_y}{\delta}, z = \frac{\Delta_z}{\delta}, \tag{1.42}$$

де

$$\Delta_x = \begin{vmatrix} x' & b_1 & d_1 \\ y' & b_2 & d_2 \\ z' & b_3 & d_3 \end{vmatrix}, \Delta_y = \begin{vmatrix} a_1 & x' & d_1 \\ a_2 & y' & d_2 \\ a_3 & z' & d_3 \end{vmatrix}, \Delta_z = \begin{vmatrix} a_1 & b_1 & x' \\ a_2 & b_2 & y' \\ a_3 & b_3 & z' \end{vmatrix}.$$

Нехай задано довільні прості числа z_1 і z_2. Тоді за схемою алгоритму RSA маємо

$$z = z_1 z_2,\ \varphi(n) = (z_1 - 1)(z_2 - 1);$$
$$e_1 d_1 \equiv 1(\mathrm{mod}\,\varphi(z)), e_2 d_2 \equiv 1(\mathrm{mod}\,\varphi(z)), e_3 d_3 \equiv 1(\mathrm{mod}\,\varphi(z)); \qquad (1.43)$$
$$e_1 < \varphi(z), d_1 < \varphi(z), e_2 < \varphi(z), d_2 < \varphi(z), e_3 < \varphi(z), d_3 < \varphi(z).$$

Шифрування відбувається з використанням елементів одного рядка матриці зображення за формулами (1.41), де

$$x = c_{i,j}, y = c_{i,j+1}, z = c_{i,j+2}, i = \overline{1,n},\ j = \overline{1,m}.$$

Для реалізації процедури шифрування вибирають три сусідні елементи рядка матриці так, щоб кожний елемент був вибраний тільки один раз і входив до складу лише однієї трійки.

Коефіцієнти a_1, a_2, a_3, b_1, b_2, b_3, d_1, d_2, d_3 є цілими числами, які повинні задовольняти такі конгруенції

$$a_1 = b_3 = d_2 = z_1^{e_1} (\mathrm{mod}\, z);$$
$$a_2 = b_1 = d_3 = z_2^{d_2} (\mathrm{mod}\, z);$$
$$a_3 = b_2 = d_1 = z_2^{e_3} (\mathrm{mod}\, z).$$

У загальному випадку вказані коефіцієнти можуть бути іншими, але такими, щоб в (1.42) виконувалася умова $\delta \neq 0$.

Дешифрування відбувається за формулами зворотного перетворення (1.42) з коефіцієнтами, обчисленими за алгоритмом RSA, тобто з конгруенції

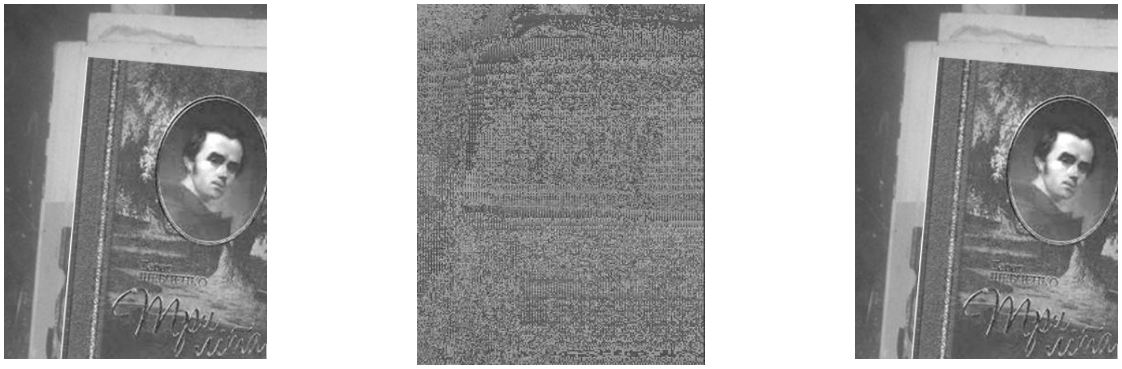
$$a_1 = b_3 = d_2 = z_1^{d_1} (\mathrm{mod}\, z);$$
$$a_2 = b_1 = d_3 = z_2^{e_2} (\mathrm{mod}\, z);$$

$$a_3 = b_2 = d_1 = z_2^{\,d_3} \left(\mathrm{mod}\, z\right).$$

3.2.2. Аналіз ефективності криптографічного захисту у випадку використання тернарних афінних форм

На рис. 3.2 наведено приклад використання розробленого методу криптографічного захисту повноколірних зображень, побудованого на основі тернарних афінних форм. Використовували зображення із такими параметрами: тип зображення – повноколірне зображення із глибиною кольору у 4 байти (колірна система ARGB), розмір – 211×265 пікселів.

Ідентичність оригінального і дешифрованого зображень перевіряли за мірою подібності, яка побудована на основі норми Гельдера.



а б в

Рис. 3.2. Результат шифрування та дешифрування повно колірного зображення (глибина кольору – 4 байти) тернарним афінним перетворенням: а –оригінальне зображення; б – зашифроване зображення; в – дешифроване зображення

Наведений метод застосовувався за інтегральним значенням чотирибайтового представлення значення функції інтесивності. При цьому перевитрати пам’яті на кожен піксел становили 1 байт.

Зазначимо, що навіть за перевитрат у 1 байт розрядність обчислювального процесу виходить за межі машинного слова на 32-бітних

системах і становить 8 байтів на піксел.

Як можна побачити з результатів шифрування, отриманий рівень зашумленості зашифрованого зображення унеможливлює отримання будь-яких інформативних даних.

Для нівелювання проблеми контурів значення простих чисел можуть бути такими, що операційні витрати оперативної пам’яті можуть у 3–4 рази перевищувати витрати у байтах, необхідні для представлення 4-байтового зображення у пам’яті.

Такі витрати не становлять проблеми у випадку використання великих обсягів оперативної пам’яті (наприклад, на типовому офісному комп’ютері). Проте у вбудованих системах пам’ять не є настільки дешевим ресурсом, як у випадку персонального комп’ютера.

Зазначимо, що серед усіх уже описаних, метод на основі використання тернарних афінних форм має найбільшу криптографічну стійкість навіть за не дуже великих значень простих чисел.

3.3. Використання кубічних форм для підвищення криптографічного захисту

Подальшим розвитком методу, побудованого на основі тернарних афінних перетворень, є його модифікація, яка полягає у запровадженні використання кубічних алгебраїчних форм. При цьому обчислювальна складність зростає не суттєво, проте значно збільшується криптографічна стійкість алгоритму побудованого на основі цієї модифікації.

Цільовим призначенням нового методу є, подібно до методу з п. 3.2, організація криптографічного захисту повноколірних зображень із глибиною кольору 4 байти. Але, на відміну від базового методу, за рахунок використання кубічних форм вдається досягати повного зашумлення зображення вже на-багато менших значеннях простих чисел. А це за несуттєвого зростання

обчислювальних ресурсів дає можливість зменшити витрати оперативної пам'яті, зокрема в апаратних реалізаціях методу.

Зазначимо, що метод криптографічного захисту повноколірних зображень на основі використання кубічних алгебраїчних форм допускає розпаралелювання процедур шифрування/дешифрування, що теоретично повинно зменшити час їх виконання у паралельних обчислювальних середовищах та у комунікаційних сеансах.

3.3.1. Схема методу криптографічного захисту повноколірних зображень на основі використання кубічних форм

Для реалізації сумісного використання елементів методу шифрування на основі тернарних афінних перетворень і кубічних алгебраїчних форм припустимо, що задано довільні прості числа z_1 і z_2.

Згідно з модифікованим алгоритмом RSA виконуються такі співвідношення

$$\begin{gathered} z = z_1 z_2,\ \varphi(n) = (z_1 - 1)(z_2 - 1), \\ e_1 d_1 \equiv 1(\mathrm{mod}\,\varphi(z)), e_2 d_2 \equiv 1(\mathrm{mod}\,\varphi(z)), e_3 d_3 \equiv 1(\mathrm{mod}\,\varphi(z)), \\ e_1 < \varphi(z), d_1 < \varphi(z), e_2 < \varphi(z), d_2 < \varphi(z), e_3 < \varphi(z), d_3 < \varphi(z). \end{gathered} \tag{1.44}$$

Розглянемо тернарне перетворення

$$\begin{aligned} u &= \alpha e + \beta d + \gamma c, \\ v &= \beta e + \gamma d + \alpha c, \\ w &= \gamma e + \alpha d + \beta c, \end{aligned} \tag{1.45}$$

для якого повинні виконуватись такі умови

$$dc \equiv 1(\mathrm{mod}\,\varphi(z)),$$

$$de \equiv 1(\mathrm{mod}\,\varphi(z)), \qquad (1.46)$$
$$ec \equiv 1(\mathrm{mod}\,\varphi(z)),$$

де $e < \varphi(z)$, $d < \varphi(z)$, $c < \varphi(z)$. Натуральні числа e, d, c в (1.46) завжди існують, наприклад, $e = \varphi(z) - 1$, $d = \varphi(z) - 1$, $c = \varphi(z) - 1$.

Знайдемо такі значення α, β, γ, щоб виконувалася конгруенція

$$((A^u)^v)^w \equiv A^1(\mathrm{mod}\,z). \qquad (1.47)$$

У силу перетворення (1.45) за властивістю 1.3 з конгруенції (1.47) отримуємо:

$$(\alpha e + \beta d + \gamma c)(\beta e + \gamma d + \alpha c)(\gamma e + \alpha d + \beta c) \equiv 1(\mathrm{mod}\,\varphi(z)) \qquad (1.48)$$

або після тотожних перетворень маємо таку конгруенцію

$$\begin{gathered} d\alpha^3 + c\beta^3 + e\gamma^3 + \\ +(e+d+c)\left[\alpha^2(\beta+\gamma) + \beta^2(\alpha+\gamma) + \gamma^2(\alpha+\beta)\right] + \\ +\left[(e^2+1)e + (d^2+1)d + (c^2+1)c\right]\alpha\beta\gamma \equiv 1(\mathrm{mod}\,\varphi(z)). \end{gathered} \qquad (1.49)$$

Конгруенція (1.49) має хоча б один розв'язок відносно значень γ, α, β [4].

Розглянемо такі три послідовності

$$u_0 = e, u_1, u_2, \ldots, u_m, \ldots, \qquad (1.50)$$
$$v_0 = d, v_1, v_2, \ldots, v_m, \ldots \qquad (1.51)$$
$$w_0 = c, w_1, w_2, \ldots, w_m, \ldots \qquad (1.52)$$

елементи яких будуються за такими рекурентними співвідношеннями

$$\begin{aligned} u_{i+1} &= \alpha_i u_i + \beta_i v_i + \gamma_i w_i, \\ v_{i+1} &= \beta_i u_i + \gamma_i v_i + \alpha_i w_i, \\ w_{i+1} &= \gamma_i u_i + \alpha_i v_i + \beta_i w_i, \end{aligned} \tag{1.53}$$

де для будь-якої трійки елементів виконується конгруенція (1.47).

Твердження 3.2. Для кожного значення $i \geq 0$ числа γ_i, α_i , β_i є розв'язком відповідної конгруенції (1.49) і для трійки чисел u_i, v_i , w_i виконується умова

$$\begin{aligned} u_i v_i &\equiv 1(\mathrm{mod}\,\varphi(z)), \\ u_i w_i &\equiv 1(\mathrm{mod}\,\varphi(z)), \\ v_v w_i &\equiv 1(\mathrm{mod}\,\varphi(z)). \end{aligned} \tag{1.54}$$

У випадку невиконання умови (1.54) при вибраному значенні i взявши до уваги (1.46), для трійки чисел u_i, v_i , w_i можна вибирати такі значення: $u_i = e$, $v_i = d$, $w_i = c$.

У силу побудови чисел в послідовностях (1.50), (1.51) і (1.52) для кожного $k \geq 0$ і для кожного натурального A виконується конгруенція

$$A^{u_k v_k w_k} \equiv A^1 (\mathrm{mod}\ \varphi(z)), \tag{1.55}$$

а в силу (1.54) такі конгруенції

$$A^{u_k v_k} \equiv A^1 (\mathrm{mod}\ \varphi(z)) \tag{1.56}$$

$$A^{u_k w_k} \equiv A^1 (\mathrm{mod}\ \varphi(z)) \tag{1.57}$$

$$A^{v_k w_k} \equiv A^1 (\mathrm{mod}\ \varphi(z)) \tag{1.58}$$

За конгруенцією (1.55) можна так модифікувати алгоритм шифрування RSA:

- таємний ключ вибирати таким, який складається з двох чисел: попереднього таємного ключа d і номера k трійки чисел u_k, v_k, w_k, для яких виконується (1.55) і які при розшифровуванні отримуються з (1.54).

Застосовуючи описану тернарну модифікацію системи RSA, різним користувачам програмної реалізації алгоритму, побудованого на основі розробленого методу, можна надати зашифрований по-різному вхідний текст, вибравши з послідовностей (1.50)–(1.52) у кожному окремому випадку власні номери k трійки чисел u_k, v_k, w_k.

3.3.2. Аналіз криптографічної стійкості у випадку використання кубічних форм

Метод шифрування повноколірних зображень на основі використання кубічних алгебраїчних форм має найбільшу криптографічну стійкість порівняно із багатьма методами та модифікаціями алгоритму RSA. Для підтвердження цієї тези проаналізовано криптографічну стійкість методу.

Нехай $\psi(z)$ – найменше спільне кратне чисел $z_1 - 1$ і $z_2 - 1$. Максимальна кількість елементів у послідовностях (1.50)–(1.52) дорівнює K, $\gamma(z) \geq 1$ – число розв'язків кубічної конгруенції (3.31). Тоді за [14] систему RSA можна розкрити за $\varphi(\psi(z)) - 1$ ітерацій.

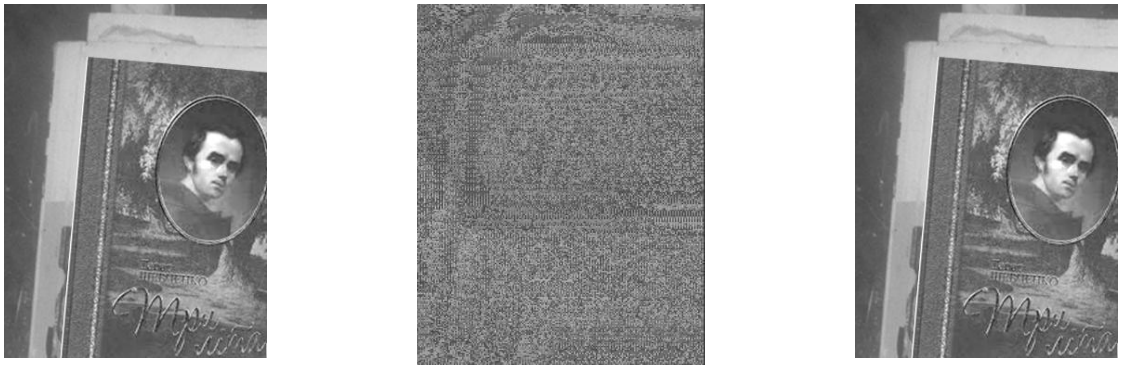
Виберемо з кожної послідовності (1.50)–(1.52) підмножину з $m \geq 1$ елементів. Як відомо з комбінаторики, підмножину з m елементів з множини, яка містить K елементів, можна вибрати C_K^m способами. Це означає, що за $\gamma(z) \cdot C_K^m \cdot (\varphi(\psi(z)) - 1)$ ітерацій можна розкрити зашифроване повідомлення тернарним методом, модифікованим використанням кубічних алгебраїчних форм. А це, своєю чергою, значно більше значення, ніж в стандартному алгоритмі RSA.

На рис.3.3 наведено приклад застосування розробленого методу. Пара-

метри зображення є такими: повноколірне зображення із глибиною кольору у чотири байти (колірна система ARGB), розмір – 211×265 пікселів.

Ідентичність оригінального і дешифрованого зображень перевіряли за мірою подібності, яка побудована на основі норми Гельдера.

Наведений метод, подібно до базового методу (описаного у п. 3.2) застосовувався за інтегральним значенням чотирибайтового представлення значення функції інтенсивності в окремому пікселі. Максимальні перевитрати пам’яті на один піксел також становили 1 байт. Рівень зашумленості зашифрованого зображення, як можна побачити з результатів, наведених на рис. 3.3, є таким, який унеможливлює отримання будь-яких інформативних даних.



а б в

Рис. 3.3. Результат шифрування та дешифрування повноколірного зображення (глибина кольору – 4 байти) тернарним афінним перетворенням з використанням кубічних форм: а –оригінальне зображення; б – зашифроване зображення; в – дешифроване зображення

Очевидно, що отриманий метод має вищу обчислювальну складність порівняно із базовим методом, побудованим на основі тернарних афінних форм. Але проблема контурів в отриманому методі вирішується за значно менших значень простих чисел. Більше того, криптографічна стійкість методу, побудованого на основі використання кубічних алгебраїчних форм, є вищою за

випадок тернарних афінних форм. Це означає, що захист повноколірного зображення при однакових значеннях ключів шифрування у першому випадку має вищий рівень.

Висновки до третього розділу

В розділі для випадку шифрування та дешифрування повноколірних зображень із значеннями глибини кольору 3 байти використано метод, який ґрунтується на сумісному використанні елементів алгоритму RSA, квадратичних форм та бінарних афінних перетворень. Криптографічна стійкість наведеного методу захисту повноколірних зображень із глибиною кольору у 3 байти значно перевищує криптографічну стійкість методу RSA.

РОЗДІЛ 4. АЛГОРИТМІЧНА ТА ФУНКЦІОНАЛЬНА РЕАЛІЗАЦІЯ МЕТОДІВ ЗАХИЩЕНОЇ ПЕРЕДАЧІ ГРАФІЧНИХ ФАЙЛІВ

4.1. Архітектура інформаційного рішення

4.1.1. Структура програмного рішення

Для реалізації процедур криптографічного захисту зображень в експериментального дослідження методів описаних у р. 2, 3, розроблено структуру програмного рішення, загальну структуру якого наведено на рис. 4.1.

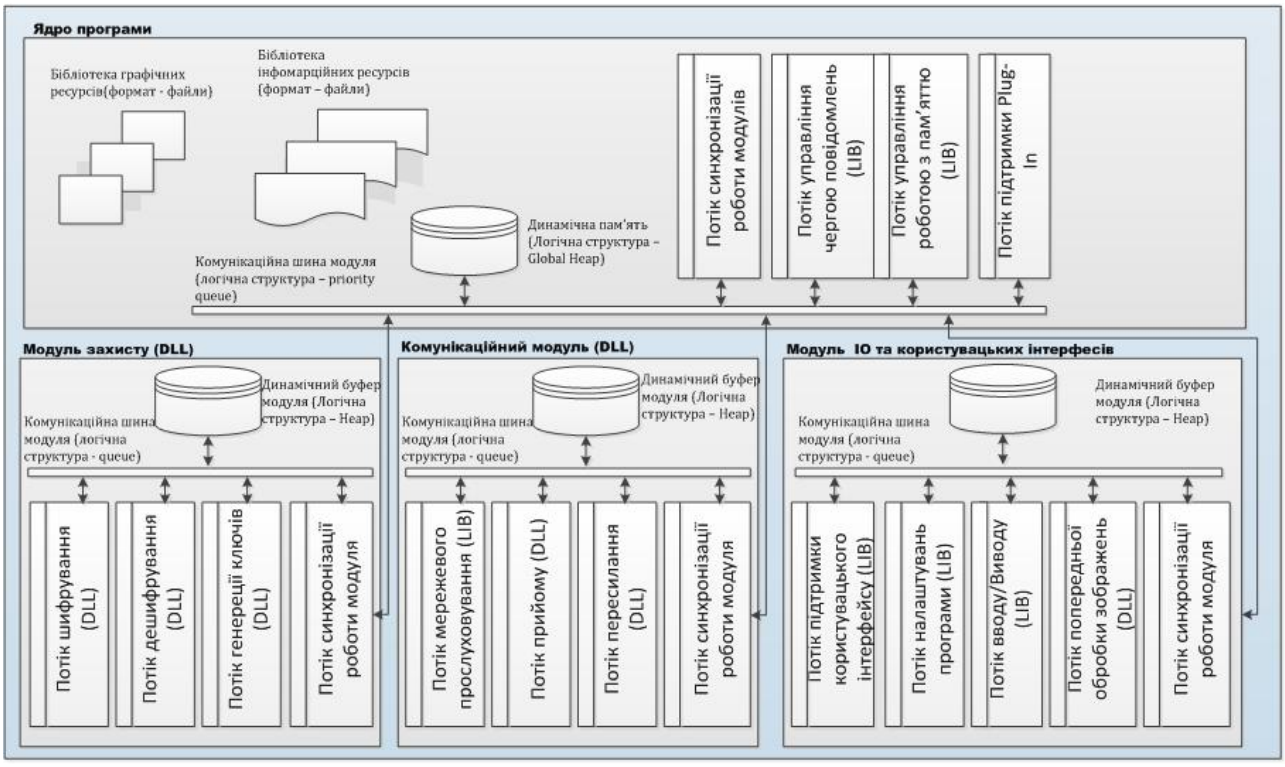


Рис. 4.1. Структурна схема програмного рішення

Система будувалась на основі модульного принципу. Кількість модулів є фіксованою. Їх структура за винятком модуля захисту є подібною.

Модуль захисту призначений для реалізації прикладних задач. Його спроектовано і розроблено за схемою динамічного наповнення функціональностями, реалізованих у форматі бінарного коду.

Така структура визначає програмне рішення як деяку оболонку для підмикання та реалізацій процедур шифрування та дешифрування цифрових зображень, які визначені у форматах jpeg, tiff, gif, pcx та bmp. Зазначимо, що у загальному випадку вхідним може бути будь-який файл. Система дасть змогу здійснити над ним процедури криптографічного захисту та мережеві транзакції.

Програмний засіб має вигляд зовнішньої програми, яка виконується в середовищі операційної системи окремим процесом. Така організація програмного засобу має свої переваги і недоліки. Перевагами є те, що:

- програмий модуль є незалежним від графічного середовища користувача, що дає змогу відлагоджувати програмний модуль незалежно від середовища;
- можна під’єднати один програмний модуль до декількох середовищ (програм) різного призначення;
- організовувати обробку в командному режимі без спеціально інтегрованого середовища;

Крім переваг, організація програми у вигляді окремого програмного модуля має і недоліки:

- ускладнений обмін між інтегрованим середовищем та програмою – обмін відбувається переважно через файли або мережеві канали;
- слабкий зв’язок інтегрованого середовища та тематичної програми ускладнює керування. Керування зводиться до запуску програми, передавання їй параметрів та аналізу результатів роботи.

Зазначимо, що система дає змогу працювати з багатьма вікнами одночасно (MDI-архітектура) та має дружній інтерфейс для управління всіма модулями.

Робота програмного забезпечення відбувається в режимі часу, наближеного до реального. Витрати на роботу процедур шифрування та дешифрування визначаються насамперед справжніми розмірами зображень та пропускною здатністю мережевих сеансів на фізичному рівні комунікаційних каналів.

Можливість візуалізації кінцевого результату дає змогу контролювати обчислення в процесі роботи.

Структурно програмне забезпечення складається з ядра програми та трьох модулів: модуля криптографічного захисту, комунікаційного модуля та модуля введення-виведення й користувацьких інтерфейсів.

На рис. 4.2 наведено схему етапів роботи програмної системи для реалізації процедур криптографічного захисту в телекомунікаційному сеансі. Зазначимо, що за наведеною схемою криптографічний захист зображень передбачає функціонування користувацьких потоків відокремлених модулів системи. В окремих випадках, наприклад, на етапі криптографічних розрорахунків, завдання виконується декількома користувацькими потоками.

Схема синхронізації роботи потоків, які реалізують етапи, що наведено на рис.4.2, переважно виконують механізми повідомлень системи (рис.4.3) та об’єкти ядра операційного середовища. Останній спосіб доволі поширений і визначається потребами забезпечення повторного використання потоку. Це означає, що користувацький потік завершує свою роботу лише за командою ядра програмного рішення.

Повторне використання потоків дає можливість мінімізувати обчислювальні витрати, пов’язані із системними процедурами створення та знищення потоків та програмними – формування структур даних і підмиканням процедур синхронізації.

У випадку розпаралелювання процедур криптографічного захисту роботу потоків синхронізують за технологією спін-блокування на основі функцій атомарного доступу операційної системи Windows.

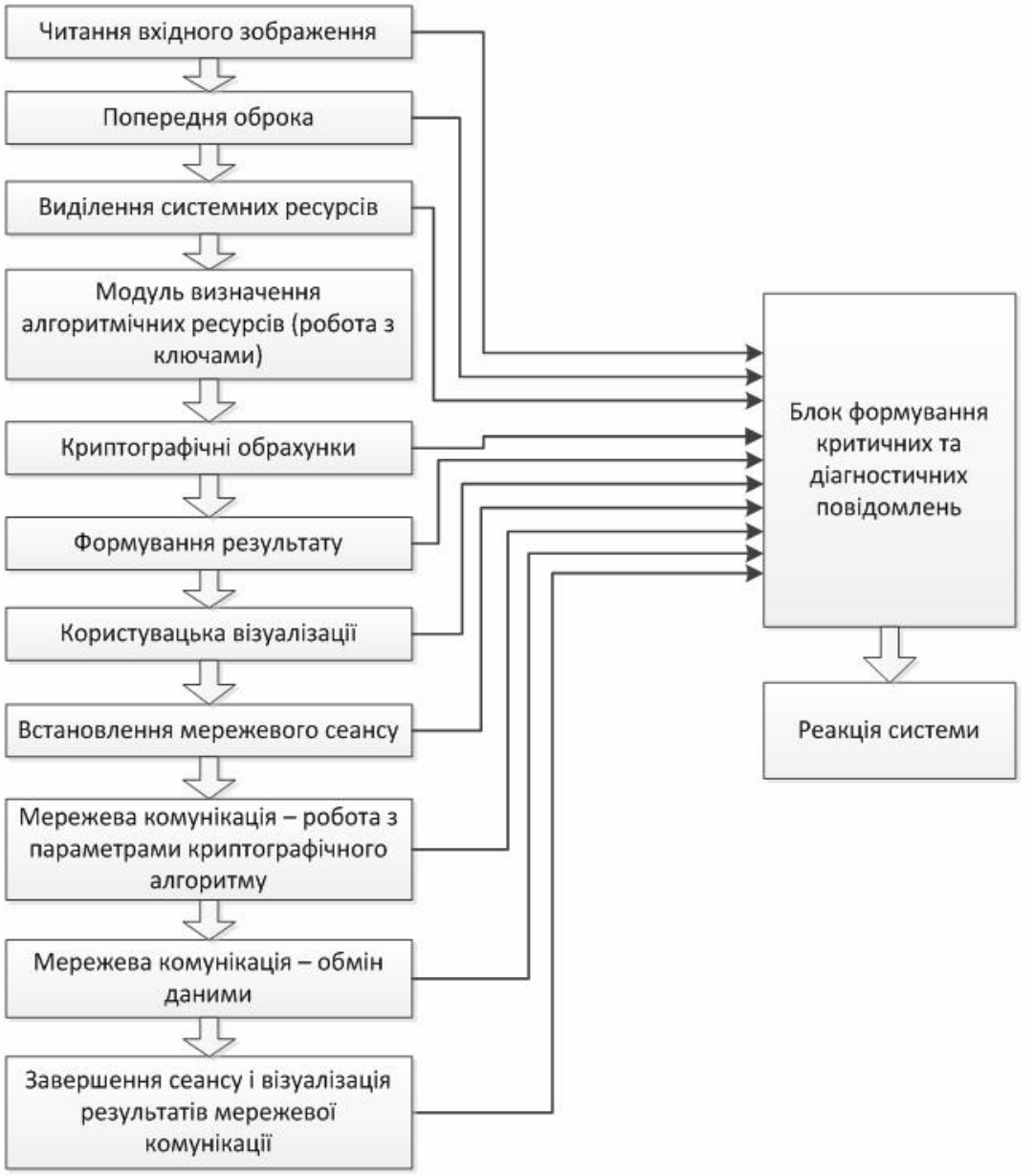


Рис. 4.2. Схема послідовності етапів реалізації криптографічного захисту

На кожному етапі здійснюється повна діагностика результатів математичних обчислень і стану потоку виконання. Усі діагностичні та критичні результати агрегує ядро програми для вироблення реакції на них. Такий підхід забезпечує реалізацію механізму, стійкого до технічних, алгоритмічних і функціональних збоїв функціонування програмного рішення на етапі виконання програми.

4.1.2. Модулі програмної системи організації криптографічного захисту

Кожен з модулів програми має власну динамічну пам’ять, комунікаційну внутрішню шину та синхронізаційний потік. Потоки синхронізації, окрім специфічних для кожного модуля завдань, виконують завдання маршрутизації повідомлень через шину між потоками модуля і шиною ядра системи. Усі комунікаційні шини програми, окрім шини ядра, реалізовуються у вигляді однонапрямлених черг повідомлень без пріоритетів. Лише шина ядра є чергою, яка маніпулює пріоритетами повідомлень.

Структуру повідомлень, які циркулюють в системі, наведено на рис. 4.3. Як можна побачити із цієї схеми, полями повідомлення є:

- receiver – ідентифікатор потоку, якому призначене повідомлення, розміщене у глобальній черзі;
- sender – ідентифікатор потоку, якому згенерував повідомлення, розміщене у глобальній черзі;
- priority – пріоритетність повідомлення. Існує два рівня пріоритетності: підвищений і звичайний. Підвищений рівень означає, що повідомлення вибиратимуть із черги першим, якщо у черзі не існує іншого повідомлення із таким самим пріоритетом;
- status – інформаційне поле, яке має два значення: виконане і невиконане. Це поле використовується потоками синхронізації для видалення деяких типів повідомлень з пам’яті.
- command – індекс у таблиці, за допомогою якого визначається команда;
- service – вказівник на буфер додаткової інформації для команди, наприклад для команди encrypt цей буфер міститиме індекс методу та його параметри;
- data – вказівник на динамічну область даних, наприклад, попіксельних значень функції яскравості цифрового зображення.

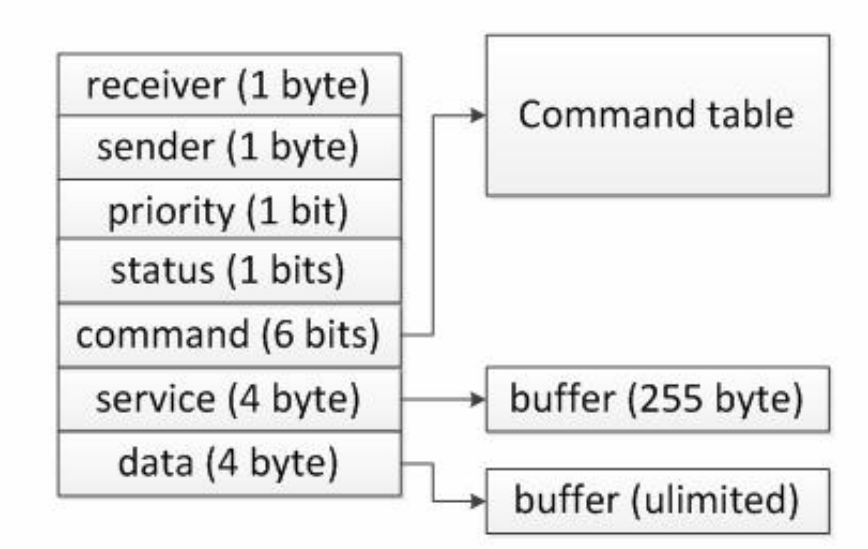


Рис. 4.3. Структура повідомлень

Коротко охарактеризуємо деякі модулі програмного рішення.

Ядро програми є також програмним модулем, забезпечує взаємодію та синхронізацію між прикладними потоками модулів системи та відповідає за функціонування програмного продукту загалом. До завдань ядра належать:

- робота з чергою задач;
- управління пріоритетами потоків;
- синхронізація роботи потоків;
- управління глобальними ресурсами програми.

Робота системи полягає у циклічній обробці черги повідомлень. Суть обробки полягає у періодичному аналізі повідомлень черги. Періодичність аналізу визначається об’єктом timer. Налаштування таймера та пріоритетності ядра визначаються налаштуваннями програми і можуть бути зміненими користувачем.

Налаштуванням за замовчуванням таймера є реалізація його роботи у режимі подібному до dynamic boost regime – у випадку відсутності активних прикладних криптографічних і мережевих потоків значення періоду таймера зменшується для того, щоб пришвидшити виконання запитів IO-підсистеми.

У випадку появи прикладних потоків значення періоду таймера збільшується – для того, щоб пришвидшити виконання комунікаційних процедур та процедур шифрування і дешифрування.

Користувацьких засобів “тонкого” налаштування boost режиму роботи таймера у програмі немає – користувач може лише задати цей режим, або його відмінити.

Зазначимо, що dynamic boost regime для прикладних потоків підтримується з використанняс операційних засобів реалізації цього режиму.

Повідомлення, які циркулюють у системі, визначають:

- прикладні завдання (наприклад, запуск процесів шифрування, передавання мережею тощо),
- системні завдання (наприклад, виконання процедур створення чи знищення потоків тощо) ;
- потоки даних (наприклад, масиви значень функції інтенсивності зашифрованого зображення).

Така схема роботи системи дає можливість відокремити системний рівень роботи програми від прикладного рівня. Фактично робота ядра системи зводиться до управління глобальними ресурсами та маршрутизації повідомлень.

Завдяки відокремленню рівнів вдалось реалізувати можливість динамічного розширення програми функціональностями криптографічного захисту, підтримки нових мережевих протоколів та забезпечити високий рівень стійкості роботи системи. Збій у роботі будь-якого модуля (складових модуля) за жодних обставин не призведе до збою роботи системи загалом.

Модуль IO та користувацьких інтерфейсів відповідає за підтримку інтерфейсу користувача, налаштування програми, введення та виведення даних та попередню обробку зображень. Модуль складається з таких потоків:

- потік підтримки користувацьких інтерфейсів;
- потік налаштування програми;

- потік введення/виведення;
- потік попередньої обробки зображень;
- потік синхронізації роботи модуля.

Потік користувацьких інтерфейсів забезпечує відображення інтерфейсу програмного забезпечення користувачеві. Потік налаштування програми забезпечує підмикання усіх динамічних бібліотек та дозволяє настроїти спосіб виведення даних для користувача. Потік введення/виведення забезпечує завантаження та вивантаження даних з жорсткого диска чи телекомунікаційної мережі на форму роботи з ними. Потік попередньої обробки зображень забезпечує конвертацію зображень між доступними типами, аналіз вхідних та оброблених зображень. Потік синхронізації роботи модуля відповідає за синхронізацію модуля та ядра програмного забезпечення.

4.2. Модуль мережевої передачі

Мережеві сеанси у програмному рішенні реалізовуються комунікаційним модулем. Цей модуль забезпечує встановлення мережевих сеансів та підтримку комунікаційних транзакцій.

Схему реалізації мережевої функціональності програми у взаємодії із підсистемами операційної системи наведено на рис. 4.4.

До складу модуля входять такі потоки:

- потік мережевого прослуховування;
- потік прийому;
- потік передачі;
- потік синхронізації роботи модуля.

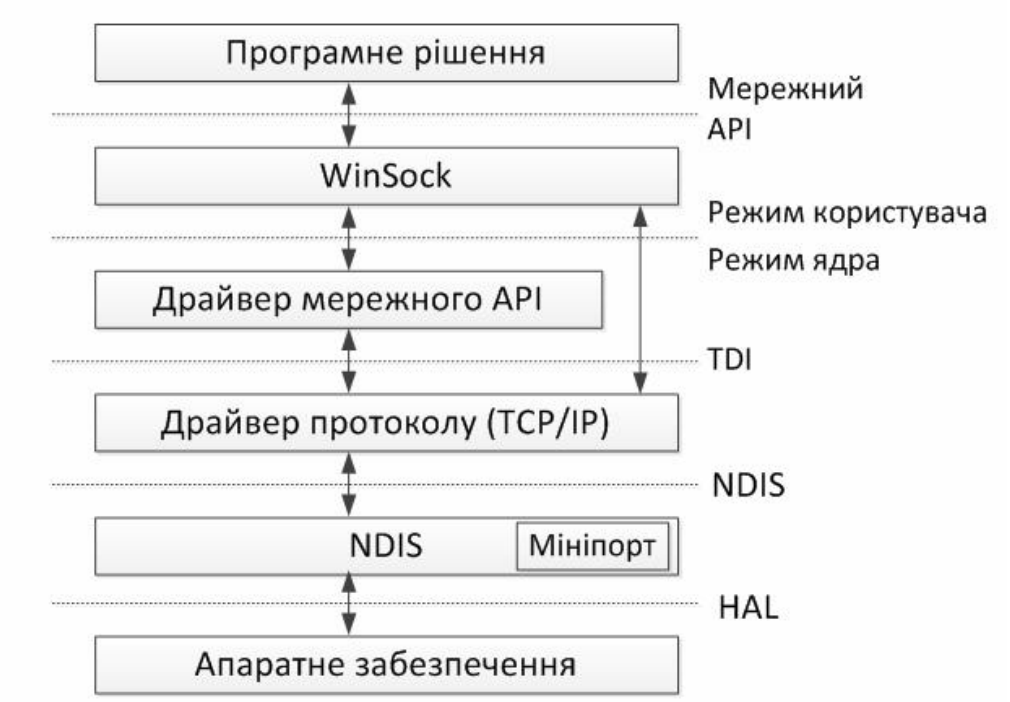


Рис. 4.4. Архітектура мережевої підтримки програмного рішення організації криптографічного захисту зображень.

На рис. 4.4.: TDI (Transport Driver Interface) – транспортний драйвер; NDIS (Network Driver Interface Specification) – специфікація інтерфейсу мережевого драйвера

Структуру комунікаційного модуля наведено на рис. 4.5. Зазначимо, що оскільки усі прикладні модулі мають схожу структуру, то принципи функціонування комунікаційного модуля є схожими на принципи функціонування модуля криптографічного захисту.

Потік мережевого прослуховування передбачає прослуховування телекомунікаційної мережі щодо виникнення події встановлення з’єднання. Потоки приймання та передавання відповідають за обмін повідомленнями – зашифрованими зображеннями – між користувачами. Їх робота ґрунтується на використанні примітивів обміну повідомленнями send і receive, які наведені на рис. 4.6. Виконання цих примітивів забезпечує виконання усіх протоколів стеку TCP/IP, які знаходяться нижче, відповідно до моделі взаємодії відкритих систем

(OSI).

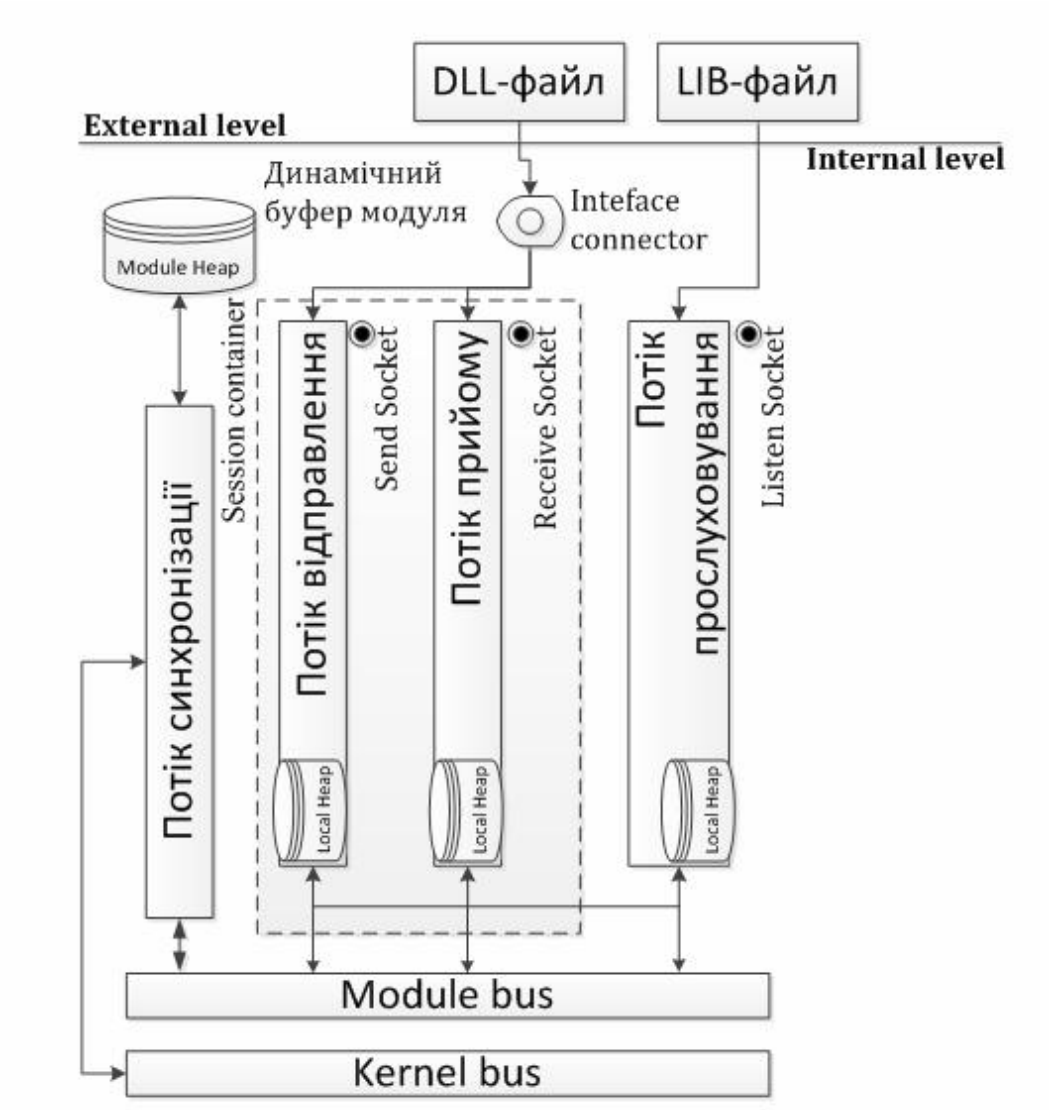


Рис.4.5. Структурна та функціональна схеми комунікаційного модуля

Функції потоків мережевої взаємодії реалізовано у вигляді динамічних бібліотек. Це означає, що вони є відокремленими від основного коду і будуть завантажені у динамічному режимі під час запуску процесу програми на виконання. Відправляють (реалізація примітиву send) та приймають (реалізація примітиву receive) повідомленя у різних бібліотеках, для того, щоб модифікація лише, наприклад, процедур відправлення не вимагала перероблення усієї програми і зокрема процедур приймання.

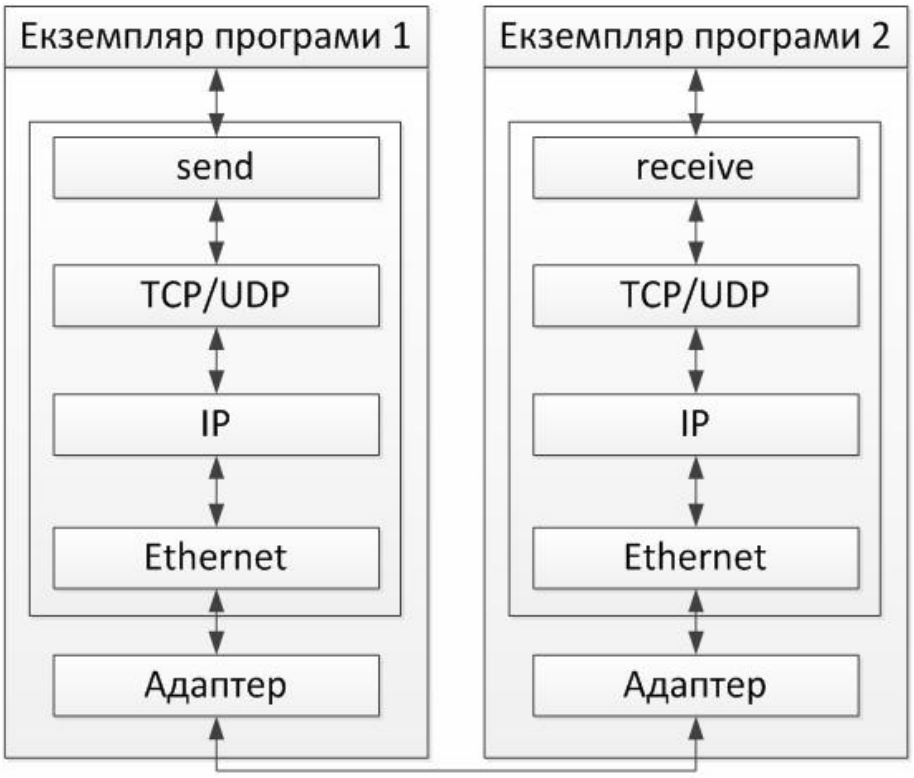


Рис. 4.6. Схема мережевої взаємодії екземплярів програми за допомогою примітивів send і receive

Зазначимо, що поверх прикладного протоколу накладено протокол, який слугує тунелем на основі протоколу TCP. Механізм тунелювання задіяно для того, щоб забезпечити комунікаційний сеанс у форматі посилання пакетів розміром по 11 байтів засобом стеку TCP/IP. Цей пакет є користувацьким й інкапсулюється пакетом протоколу TCP.

Розмір користувацького пакета можна змінити, але до визначеного користувачем розміру завжди додаватиметься один байт. Цей байт реалізовує ідею тунелю, а тому є службовим і містить інформацію користувацького протоколу, наприклад, 0 і 1 розряди у цьому байті визначають тип пакета: початковий, кінцевий чи сеансів.

Використання тунельного підходу забезпечує можливість додавання до комунікаційного сеансу інформації, яка забезпечує додаткову стійкість мережевої комунікації.

Наявність додаткового байта дає можливість реалізувати технологію динамічного використання криптографічних ключів, що сприятиме підвищенню рівня безпеки в мережевих транзакціях.

Використання концепції реалізації мережевої взаємодії через динамічні бібліотеки дає змогу розширити підтримку мережевих протоколів. Зокрема стала можливою реалізація мережевої комунікації за допомогою тунельованих протоколів.

Потік синхронізації забезпечує узгоджену із ядром програмної системи роботу програмного модуля. Такий підхід забезпечує стійкість (незалежність) роботи програми загалом від технічних збоїв функціонування комунікаційного модуля.

4.3. Модуль реалізації криптографічного захисту

Модуль захисту розв’язує задачі реалізації процедур шифрування та дешифрування, а також роботи з ключами шифрування і є основним прикладним модулем програмного рішення. Його структурну та функціональну схеми наведено на рис. 4.7.

Модуль складається з потоку синхронізації та контейнерів потоків криптографічного захисту.

Кожен контейнер криптографічного захисту, своєю чергою, складається із трьох потоків: шифрування, дешифрування та службового потоку забезпечення додаткових функціональностей, необхідних для реалізації процедур шифрування чи дешифрування.

Як і інші модулі системи, потік захисту має власну комунікаційну шину. Усі потоки будь-якого контейнера криптографічного захисту, навіть між собою, тобто в межах самого контейнера, комунікують виключно через цю шину.

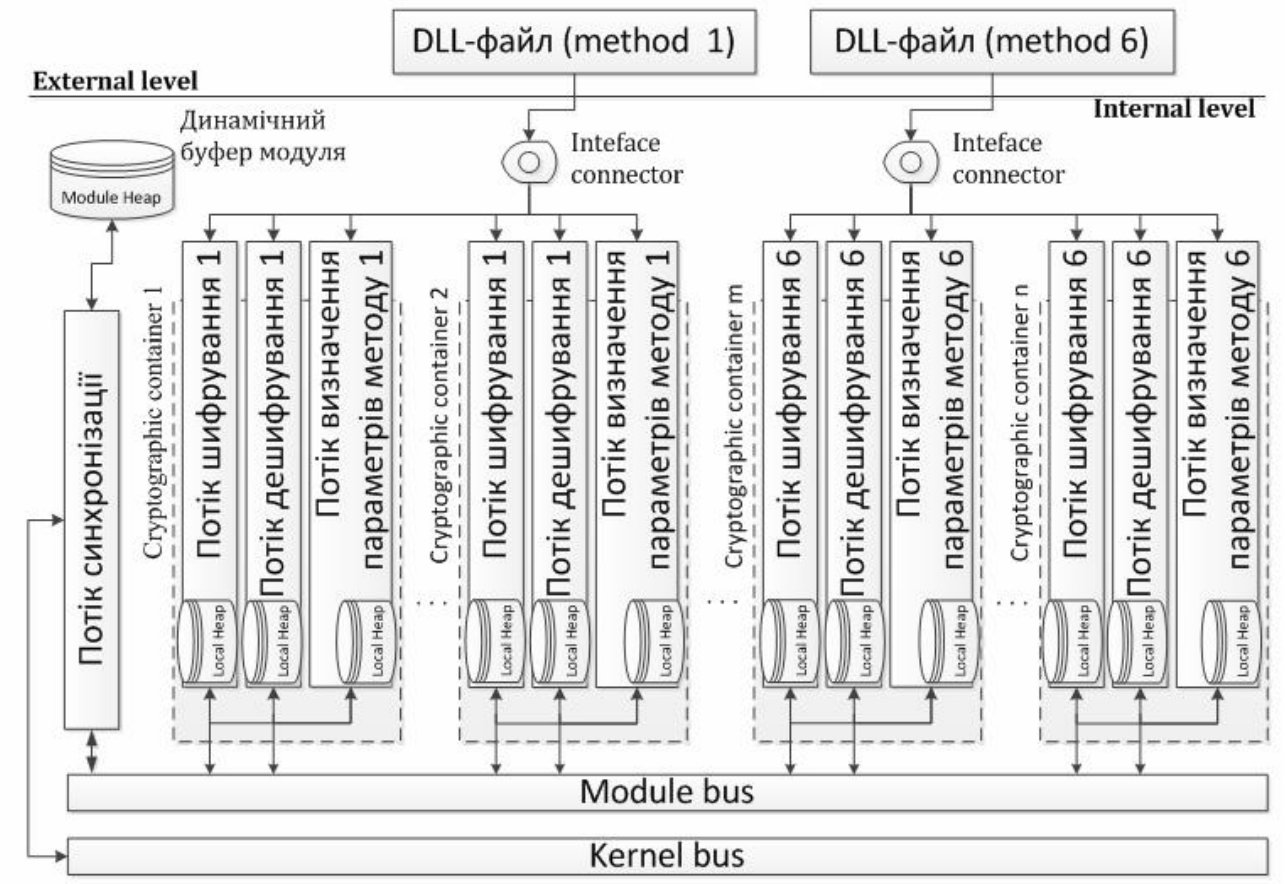


Рис. 4.7. Структурна та функціональна схеми модуля криптографічного захисту

Доступ до зовнішньої шини, тобто шини ядра, має лише потік синхронізації. Тому маршрутизація повідомлень у інші модулі системи засобом шини ядра реалізується лише цим потоком.

Усі функції, які реалізують ці потоки, завантажуються з динамічних бібліотек (DLL) за єдиним інтерфейсом. Така концепція дає можливість розширення і нарощення процедур криптографічного захисту без перероблення усієї програми. Саме для реалізації цієї концепції у розробленому інтерфейсі є деяка надлишковість, яка відображається на схемі існуванням однакових потоків визначення параметрів методу, що був завантаженим з однієї бібліотеки.

Робота системи визначається організацією криптографічного захисту

зображення, яке відтворене у користувацькому вікні. Таких вікон одночасно може бути декілька, – в усіх одночасно можуть бути запущеними, наприклад, процедури шифрування. Якщо, наприклад, у двох вікнах запущено шифрування однаковим методом, то в системі динамічно буде створено два контейнери криптографічного захисту за однаковим методом. Тому на схемі, наведеній на рис. 4.7, показано, що для одного методу може існувати декілька крипто-графічних контейнерів.

Динамічний режим створення та знищення контейнерів криптографічного захисту повністю автоматичний, тобто існує неявно для користувача програми. Створення контейнера визначається операціями шифрування і дешифрування, а знищення – процедурою закривання вікна. За незакритого вікна криптографічний контейнер існуватиме навіть у випадку реалізованих процедур шифрування чи дешифрування, оскільки можливий повторний виклик цих процедур.

Контейнер у складі потоків шифрування і дешифрування створюватиметься незалежно від початкового завдання, наприклад, шифрування. Це зроблено для того, щоб користувач в одному вікні чи над одним зображенням міг здійснювати обидві процедури.

На загальній схемі, яку наведено на рис. 4.1, для вираження основного призначення в модулі криптографічного захисту наведено потік генерації ключів. У детальній схемі цього модуля, яку наведено на рис. 4.7, цей модуль подано як модуль визначення параметрів методу. Таку заміну зроблено для того, щоб підкреслити, що цей потік повинен виконати усі сервісні процедури, які необхідні для процедур криптографічного захисту і такі, що не обмежуються лише генерацією ключів. Наприклад, у топологічній модифікації саме цей потік поряд із генерацією ключів ще визначає топологію розбиття цифрового зображення.

Як вже зазначалось, технологію підмикання криптографічних функціональностей реалізовано засобом динамічних бібліотек. Вона полягає у

тому, що перевіряється наявність файлів бібліотек у поточному каталозі. Якщо такі файли існують, то система їх завантажує до оперативної пам’яті, і лише за успішного завантаження метод стає доступним для використання у системі.

У наступній версії програми, у якій розширено підтримку комунікаційних протоколів, користувач зможе віддалено викликати процедури шифрування та дешифрування (технологія RPC).

4.4. Інтерфейс програмного рішення

4.4.1 Структура графічного інтерфейсу

Графічного інтерфейс користувача розроблено засобами мови програмування C++. Оскільки мова C++ є мовою бібліотек, то для ефективного програмування необхідно максимально використати наявні бібліотеки й інструменти.

Графічний інтерфейс побудовано за багатовіконною технологією (технологія MDI). Ця технологія передбачає наявність основного (“батьківського”) вікна, в якому зосереджено засоби керування та опрацювання зображень. Кожне зображення, яке опрацьовується відкривається в окремому “дочірньому” вікні.

Інтерфейс програмного забезпечення містить пункти (рис. 4.8), кожен з яких має певне функціональне призначення. Їх можна охарактеризувати так:

Файл – цей пункт меню призначений для завантаження та збереження зображень, а також для визначення налаштувань програми. Зображення можна завантажувати з різних графічних форматів у дочірні вікна програмної системи залежно від подальших шляхів їх використання.

Редагування – надає можливість маніпуляцій із зображеннями: додавання та видалення окремих зображень чи цілих наборів, виділення та копіювання зображень.

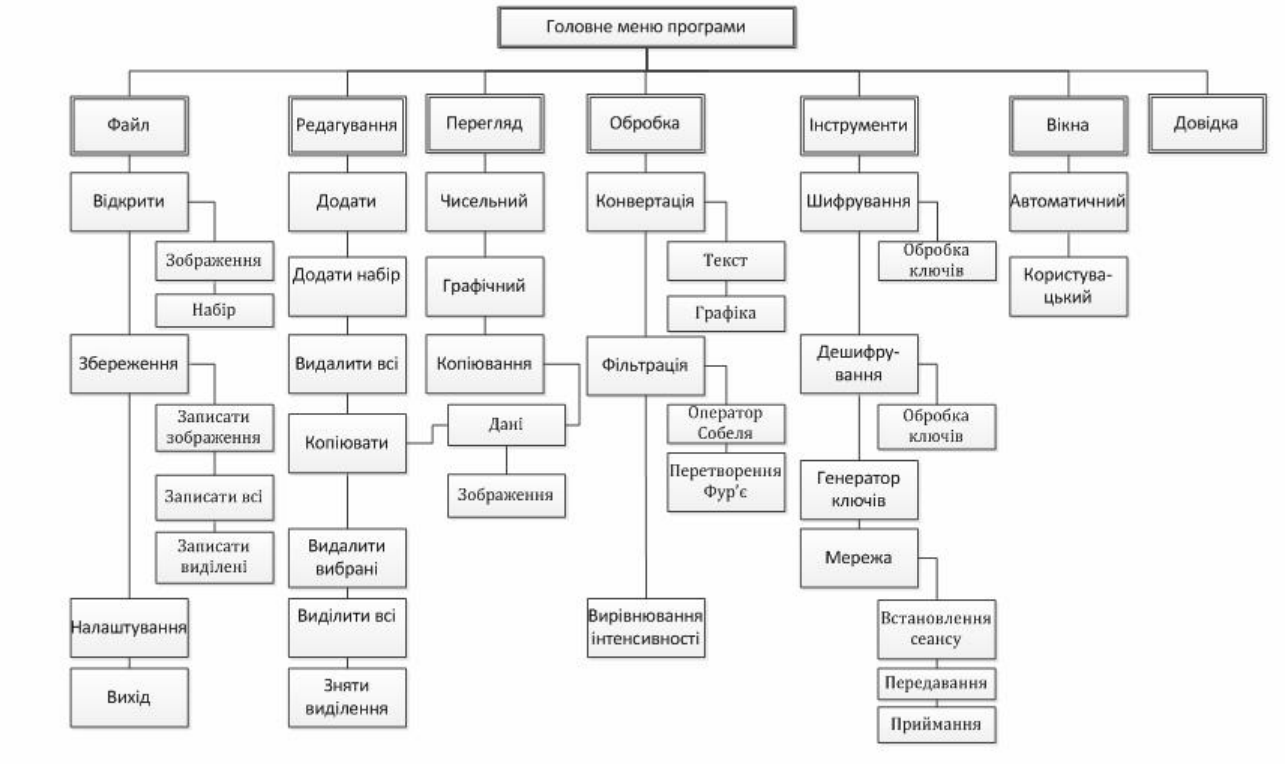


Рис. 4.8. Структура головного меню програмного забезпечення

Перегляд – пункт меню, що відповідає за спосіб та структуру відображення даних чи зображень у “дочірньому” вікні.

Обробка – це пункт, призначений для обробки зображень та роботи з ними. Дозволяє конвертувати зображення з графічного формату у текстовий і навпаки, а також забезпечує деякі засоби фільтрації зображень та роботи із значеннями функції інтенсивності.

Інструменти є основним прикладним пунктом меню, який реалізує безпосередньо процедури шифрування та дешифрування зображень, роботу з криптографічними ключами, що передбачає як обробку отриманих ключів, так і генерування нових. Також цей пункт меню реалізує комунікаційні засоби, а саме: робота з телекомунікаційними мережами, встановлення сеансів та передавання і приймання даних у цих сеансах.

Вікна – пункт меню, що відповідає за роботу з вікнами програмної системи. Оскільки структура програмного забезпечення є багатовіконною, то

цей пункт меню надає зручну та зрозумілу навігацію між робочими вікнами для забезпечення зручного користування програмним засобом.

Довідка є пунктом меню, який виводить на екран коротку довідку про програмне забезпечення, його структуру та можливості.

4.4.2. Користувацький алгоритм виконання процедур криптографічного захисту зображень

Процес опрацювання зображення у системі захищеної комунікації між робочими станціями поетапним. На першому етапі завантажується зображення. Його можна отримати як з локального ресурсу у випадку необхідності шифрування чи дешифрування зображення з подальшим його надсиланням адресату в телекомунікаційному сеансі, так і з мережевого комунікатора програмного рішення.

Зображення можна завантажити як в графічному, так і в текстовому форматі. У випадку завантаження текстових даних система за вибором користувача запитає формат конвертації цих даних у цифрове зображення. Цей формат містить таку інформацію: розрядність глибини кольору, формат значень функції інтенсивності та розміри зображення.

Оскільки процедури шифрування та дешифрування у більшості розроблених методів є багатопотоковими, то існує користувацька можливість визначення пріоритетів потоків обробки і додаткових параметрів, необхідних для роботи системи.

Нижче наведено інтерфейс системи шифрування та дешифрування зображень, а саме вікно для завантаження початкових параметрів – формату вхідних даних та самого зображення. Після вибору формату вхідних даних в модальному вікні вказується повний шлях до файла або обирається файл в системному дереві папок та файлів.

Наступним етапом шифрування є вибір методу шифрування зображення та параметрів ключа, за якими шифруватиметься зображення.

Вікно задання ключа шифрування також відповідає за перевірку ключа на відповідність вимогам методу RSA. Приклад функціонування програми на цьому етапі наведено на рис. 4.9 – 4.10.

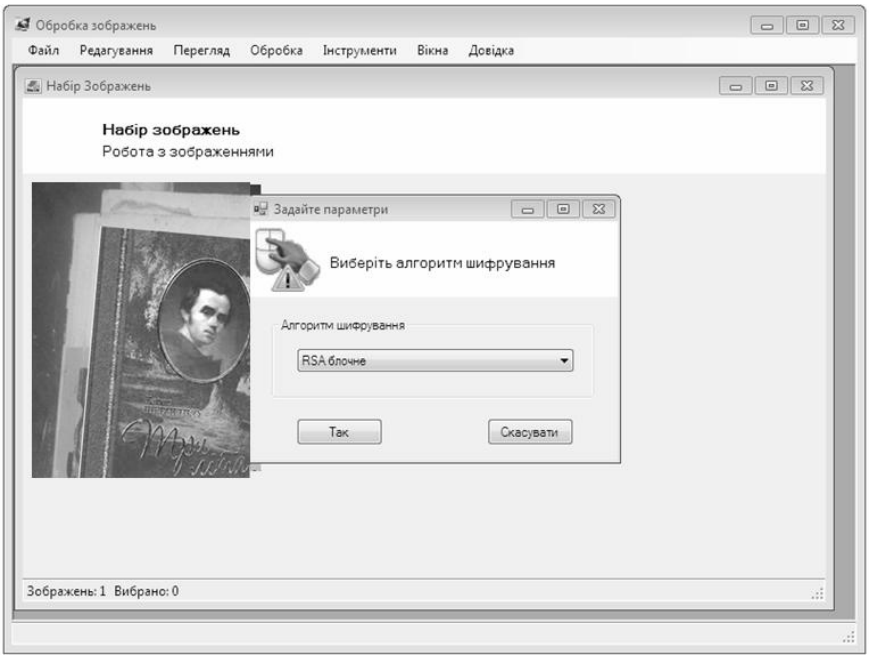


Рис. 4.9. Інтерфейс вибору методу шифрування

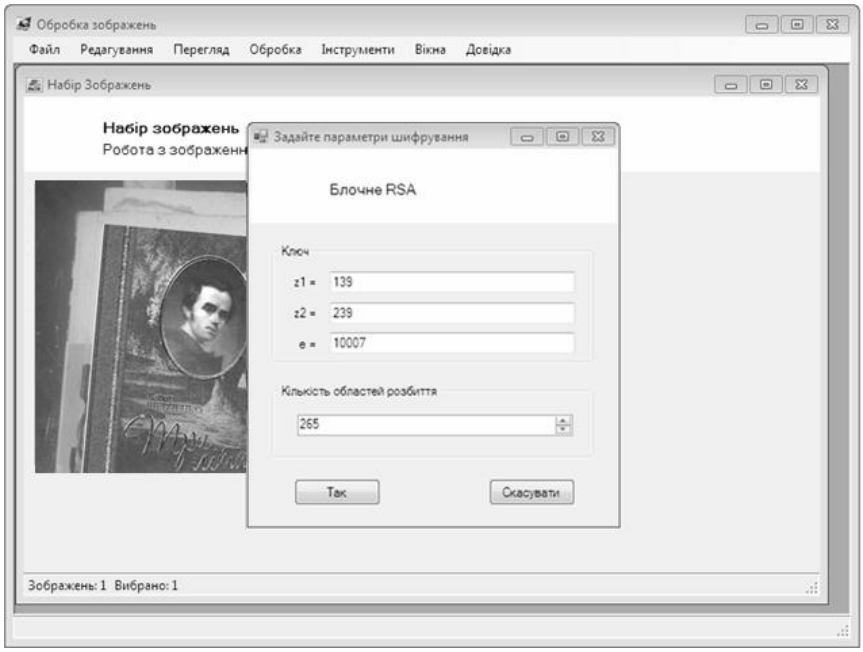


Рис. 4.10. Інтерфейс вибору ключа шифрування

Наступним етапом дешифрування є вибір методу дешифрування зображення та параметрів ключа, за якими дешифруватиметься зображення. Математичний апарат використаний для реалізації цих методів також було описано раніше. Як і під час шифрування, в процесі дешифрування перевіряється ключ дешифрування на відповідність методу RSA. Приклад функціонування програми на цьому етапі наведено на рис. 4.11.

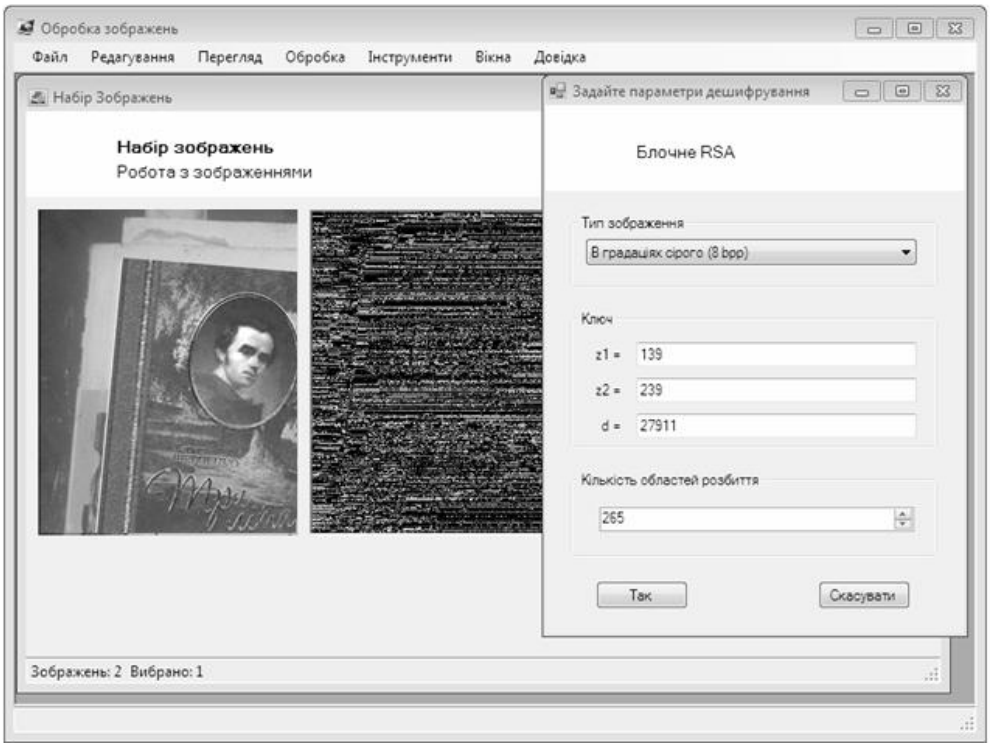


Рис. 4.11. Інтерфейс вибору ключа дешифрування

Приклад виведення у графічному форматі результатів послідовного запуску процедур шифрування та дешифрування наведено на рис. 4.12. Тут перше зображення є завантаженим з локального диска, друге – зашифроване топологічним методом перше зображення, третє – дешифроване тим самим методом друге зображення.

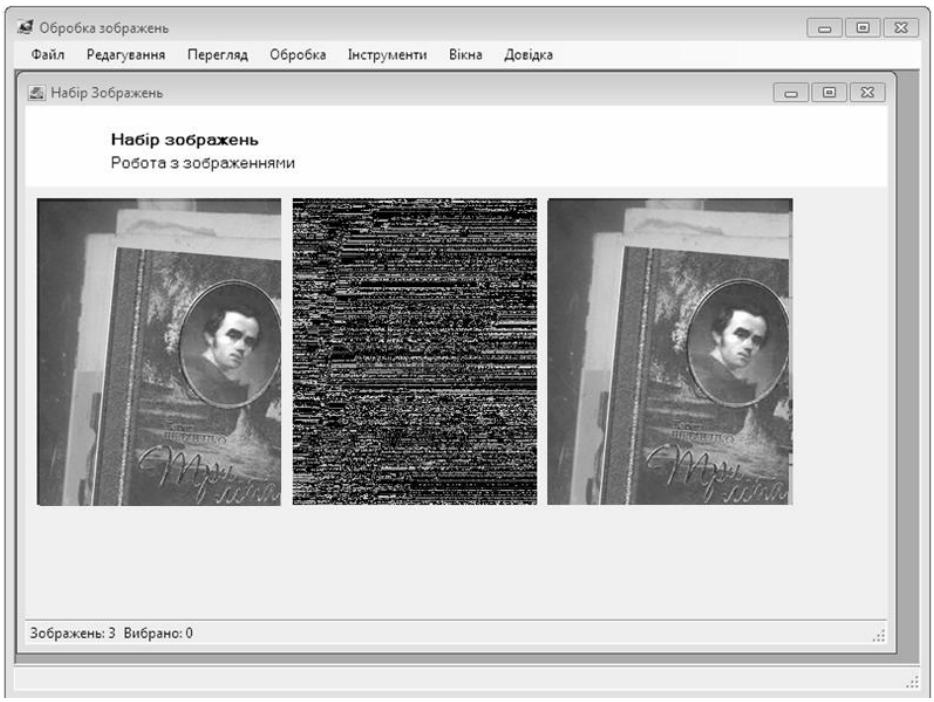


Рис. 4.12. Інтерфейс з кінцевими результатами шифрування та дешифрування отриманого зображення

Висновки до четвертого розділу

В розділі представлено структуру і алгоритмічне забезпечення системи яка дозволяє виконувати захист від несанкціонованого доступу в інформаційних системах передавання цифрових зображень та графічних файлів. Використання технологій паралельних обчислень забезпечує можливість одночасної роботи декількох прикладних функціональностей криптографічного захисту в межах одного аплікаційного процесу.

ЗАГАЛЬНІ ВИСНОВКИ

В магістерській роботі досліджено моделі, методи та алгоритми кодування та криптографічного захисту зображень та графічних файлів і описано задачу підвищення криптографічної стійкості та забезпечення достовірності інформації в інформаційних системах за рахунок застосування методів та інформаційних технологій шифрування зображень, які інтегрально поєднують метод RSA і процедури додаткової зашумленості.

На підставі опрацювання літературних джерел проаналізовано відомі методи та засоби шифрування даних і недоліки методу RSA у разі застосування його щодо зображень.

Приведено метод криптографічного захисту повноколірних зображень із глибиною кольору у 4 байти, який поєднує елементи алгоритму RSA, кубічних алгебраїчних форм. На основі сумісного використання елементів алгоритму RSA, квадратичних алгебраїчних форм і бінарних афінних перетворень описано метод шифрування та дешифрування повноколірних зображень із глибиною кольору у 3 байти має підвищену криптографічну стійкість і мінімізує витрати пам’яті.

Представлено інформаційну технологію захисту графічних файлів і зображень із глибиною кольору у 1 і 4 біти завдяки сумісному використанню методу RSA, потокового та топологічного підходів, яка зменшує контури об’єктів на зашифрованих зображеннях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Анісімов А. В. Алгоритмічна теорія великих чисел. Модулярна арифметика великих чисел / Анатолій Васильович Анісімов – К.: Видавничий дім «Академ періодика», 2001. – 153 с.
2. Березовський А. І. Деякі резерви оптимізації обчислень для множення багаторозрядних чисел / А. І. Березовський, В. К. Задірака, С. С. Мельникова, Л. Б. Шевчук // Зб. «Теорія обчислень». – К., 1999. – с. 325-329.
3. Березовський А. І. Про оптимізацію за швидкодією алгоритмів виконаня операцій над багаторозрядними числами / А. І. Березовський, М. П. Бесараб, В. К. Задірака, Л. Б. Шенчук // Вісник державного увіверситету «Львівська політехніка» “Прикладна математика”. – Львів. – 1998. – Т. 2. № 337. – с. 297 – 300.
4. Бородін О. І. Теорія чисел / О. І. Бородін. – К.: Вища школа, 1970. – 275 с.
5. Брюс Шнайдер. Прикладна криптографія / Брюс Шнайдер. – К.: Тріумф, 2003. – 815 с.
6. Вербіцький О. В. Вступ до криптології / О. В. Вербіцький. – Львів: Видавницгво науково-технічної літератури, 1998. – 247 с.
7. Використання кубічних форм для підвищення стійкості шифрування тернарними афінними перетвореннями: Матеріали міжнародної наукової конференції [“Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту (ISDMCI'2013)], (Євпаторія, 20-24 травня 2013) / ХНТУ. – Херсон, 2013. – Т1. – с. 175-177.
8. Воробель Р. А. Підвищення точності реконструкції зображень за критерієм мінімуму енергії / Р. А. Воробель, І. Б. Івасенко // Відбір і обробка інформації. – 2005. – № 23(99). – С.112 – 116.

9. Директива 1999/93/ЄС Європейського парламенту та Ради від 13 грудня 1999 року про систему електронних підписів, що застосовується в межах Співтовариства // Офіційний журнал L 013. – 19/01/2000. – с. 0012–0020.
10. Ємець В. Сучасна криптографія. Основні поняття / В. Ємець, А. Мельник, Р. Попович. – Львів: БАК, 2003. – 144 с.
11. Задірака В. К. Комп'ютерна арифметика багаторозрядних чисел: Наукове видання / В. К. Задірака, О. С. Олексюк. – К., 2003. – 264 с.
12. Задірака В. К. Комп'ютерна криптологія: Підручник / В. К. Задірака, О. С. Олесюк. – К., 2002. – 504с.
13. Закон України про електронний цифровиий підпис. – К., 22 травня 2003р., № 852-IV.
14. Adleman L. On distinguishing prime numbers from composit numbers / L. Adleman, C. Pomerance, R. S. Rumaly // Ann.V-fth/ – 1983. – V.117. – p. 173-206.
15. Ahmed A. Abd El-Latif. A hybrid chaotic system and cyclic elliptic curve for image encryption / Ahmed A. Abd El-Latif, Xiamu Niu // AEU – International Journal of Electronics and Communications. – 2013. –Vol. 67, Issue 2. – p. 136-143.
16. Ahmed A. Abd El-Latif. A new image cipher in time and frequency domains / Ahmed A. Abd El-Latif, Xiamu Niu, Mohamed Amin // Optics Communications. – 2012. – Vol. 285, Issue 21-22. – p. 4241-4251.
17. Banerjee S. Synchronization of time delayed semiconductor lasers and its applications in digital cryptography / S. Banerjee, L. Rondoni, S. Mukhopadhyay // Optics Communications. – 2011. – Vol. 284, Issue 19. – p. 4623-4634.
18. Chong Fu. A novel chaos-based bit-level permutation scheme for digital image encryption / Chong Fu, Bin-bin Lin, Yu-sheng Miao, Xiao Liu, Jun-jie

Chen // Optics Communications. – 2011. – Vol. 284, Issue 23. – p. 5415-5423.

19. Cohen H. Primality testing and Jacobi sums / H. Cohen, H. W. Lenstra // Math. Comp. –1984. – V.42(165). – p. 297-330.
20. Emad Mosa. Chaotic encryption of speech signals / Emad Mosa, Nagy W. Messiha, Osama Zahran, Fathi E. Abd El-Samie // International Journal of Speech Technology. – 2011. –Vol. 14, Issue 4. – p. 285-296.
21. Encryption and decryption of images using fractional-rational form of n degree with the elements of the RSA algorithm: Матеріали 1-ї міжнародної науково-технічної конференції [“Захист інформації і безпека інформаційних систем”], (Львів, 31 травня – 1 червня)/ НУ ЛП. – Львів, 2012. – с. 152.
22. Ercan Solak. Algebraic break of image ciphers based on discretized chaotic map lattices / Ercan Solak, Cahit Çokal // Information Sciences. – 2011. – Vol. 181, Issue 1. – p. 227-233.
23. Geng Zhao. Block Cipher Design, Issue Generalized Single-Use-Algorithm Based on Chaos / Geng Zhao, Guanrong Chen, Jingqing Fang, Gang Xu Tsinghua // Science & Technology. – 2011. – Vol. 16, Issue 2. – p. 194-206.
24. Guodong Ye. An efficient chaotic image encryption algorithm based on a generalized Arnold map / Guodong Ye, Kwok-Wo Wong // Nonlinear Dynamics. – 2012. – Vol. 69, Issue 4. – p. 2079-2087.
25. Guoji Zhang. A novel image encryption method based on total shuffling scheme / Guoji Zhang, Qing Liu // Optics Communications. – 2011. – Vol. 284, Issue 12. – p. 2775-2780.
26. Images protection by using projective and binary affine transformations with elements of RSA algorithm: Proceedings of the 11 International Conference [“The experience of designing and application of CAD systems in microelectronics” (CADSM 2011)], (Lviv – Polyana, February 23 – 25,

2011)/ Lviv Polytechnik National University. – Lviv: Publishing House Vezha&Co, 2011. – p. 321.

27. Kamlesh Gupta. Novel Approach for fast Compressed Hybrid color image Cryptosystem / Kamlesh Gupta, Sanjay Silakari // Advances in Engineering Software. – 2012. – Vol. 49. – p. 29-42.
28. Kanso A. A novel image encryption algorithm based on a 3D chaotic map / A. Kanso, M. Ghebleh // Communications in Nonlinear Science and Numerical Simulation. – 2012. – Vol. 17, Issue 7. – p. 2943-2959.
29. Lenstra H. W. Primality testing algorithms / H. W. Lenstra // Bourbaki Seminar. – 1981. –V.1980/81. – p. 243-257.
30. Maciej P. Wojtkowski. On the Real Baker Map / Maciej P. Wojtkowski // Reports on Mathematical Physics. – 2012. – Vol. 69, Issue 2. – p. 235-242.
31. Omid Mirzaei. A new image encryption method, Issue parallel sub-image encryption with hyper chaos / Omid Mirzaei, Mahdi Yaghoobi, Hassan Irani // Nonlinear Dynamics. – 2012. – Vol. 67, Issue 1. – p. 557-566.
32. Rebollo-Neira L. Self-contained encrypted image folding / L. Rebollo-Neira, J. Bowley, A.G. Constantinides, A. Plastino // Issue Statistical Mechanics and its Applications. – 2012. – Vol. 391, Issue 23. – p 5858-5870.
33. Ruisong Ye. A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism / Ruisong Ye // Optics Communications. – 2011. – Vol. 284, Issue 22. – p. 5290-5298.
34. Seyed Mohammad Seyedzadeh. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map / Seyed Mohammad Seyedzadeh, Sattar Mirzakuchaki // Signal Processing. – 2012. – Vol. 92, Issue 5. – p. 1202-1215.
35. Seyyed Mohammad Reza Farschi. A novel chaotic approach for information hiding in image / Seyyed Mohammad Reza Farschi, H. Farschi // Nonlinear Dynamics. – 2012. –Vol. 69, Issue 4. – p. 1525-1539.

36. Shiguo Lian. Traceable content protection based on chaos and neural networks / Shiguo Lian, Xi Chen // Applied Soft Computing. – 2011. – Vol. 11, Issue 7. – p. 4293-4301.
37. The use of conjunctive partitioning of images to increase strength of the RSA algorithm: Proceedings of the VIIIth International Scientific and Technical Conference ["Perspective Technologies and Methods in MEMS Design " (MEMSTECH 2012)], (Lviv-Polyana, April 18-21, 2012) / Lviv Polytechnik National University. – Lviv: Publishing House Vezha&Co., 2012 – p. 138-139.
38. The Use of Disjunctive Covering of Images to Increase Strength of the RSA Algorithm: Proceeding of the VIIth International Conference ["Perspective Technologies and Methods in MEMS Design " (MEMSTECH 2011)], (Lviv-Polyana, May 11-14, 2011) / Lviv Polytechnik National University. – Lviv: Publishing House Vezha&Co, 2011. – p. 168-169.
39. The use of quadratic forms to increase strength of the encryption of binary affine loop transformation: Proceedings of the V International Scientific and Technical Conference ["Computer Science And Information Technologies" (CSIT 2010)], (Lviv, October 14-16, 2010) / Lviv Polytechnik National University. – Lviv: Publishing House Vezha&Co., 2010. – p. 19-21.
40. Volos Ch. K. Image encryption process based on chaotic synchronization phenomena / Ch.K. Volos, I.M. Kyprianidis, I.N. Stouboulos // Signal Processing. – 2013. – Vol. 93, Issue 5. – p. 1328-1340.
41. Wei Zhang. A symmetric color image encryption algorithm using the intrinsic features of bit distributions / Wei Zhang, Kwok-wo Wong, Hai Yu, Zhi-liang Zhu // Communications in Nonlinear Science and Numerical Simulation. 2013 . – Vol. 18, Issue 3. – p. 584-600.
42. Wei Zhang. An image encryption scheme using lightweight bit-level confusion and cascade cross circular diffusion / Wei Zhang, Kwok-wo Wong,

Hai Yu, Zhi-liang Zhu // Optics Communications. – 2012. – Vol. 285, Issue 9. – p. 2343-2354.

43. Wei Zhang. An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion / Wei Zhang, Kwok-wo Wong, Hai Yu, Zhi-liang Zhu // Communications in Nonlinear Science and Numerical Simulation. – 2013. – Vol. 18, Issue 8. – p. 2066–2080.
44. Xiaoling Huang. Image encryption algorithm using chaotic Chebyshev generator / Xiaoling Huang // Nonlinear Dynamics. – 2012. –Vol. 67, Issue 4. – p. 2411-2417.
45. Xingyuan Wang. Chaotic encryption algorithm based on alternant of stream cipher and block cipher / Xingyuan Wang, Xiaojuan Wang, Jianfeng Zhao, Zhenfeng Zhang // Nonlinear Dynamics. – 2011. – Vol. 63, Issue 4. – p. 587-597.
46. Yong Wang, Kwok-Wo Wong, Xiaofeng Liao, Guanrong Chen. (2011) A new chaos-based fast image encryption algorithm. Applied Soft Computing. Vol. 11, Issue 1, 2011, Pages: 514-522.