DOI 10.15276/imms.v15.no3.419
УДК 004.056.5

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No. 3, pp. 419-429

РОЗРОБКА АДАПТИВНОГО ДО КОНТЕНТУ АЛГОРИТМУ ДЛЯ ШИФРУВАННЯ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ

А. В. Садченко, О. А. Кушніренко

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Email: koa@op.edu.ua

У сучасному інформаційному середовищі дедалі більшого значення набуває захист цифрових даних, серед яких важливу частку становлять графічні матеріали – фотографії, відсканові документи та мультимедійні файли. Вони, так само як і текстові дані, можуть містити персональну або критично важливу інформацію, що потребує надійних механізмів шифрування та збереження. Проте навіть за наявності розвинених криптографічних методів текстові й графічні файли залишаються вразливими через специфічні структурні особливості. Для текстових даних типовими є статистичні закономірності, наприклад частотний розподіл символів і слів, що може використовуватися для проведення криптоаналітичних атак у разі застосування слабких або однотипних шифрів. У графічних файлах одним із ключових ризиків є висока кореляція між сусідніми пікселями, особливо у випадку природних зображень або даних із мінімальним стисненням, що створює можливість реконструкції чи аналізу навіть частково зашифрованої інформації. Додаткову загрозу становлять метадані (EXIF, IPTC, XMP), які автоматично додаються під час створення або редагування файлів і можуть містити конфіденційні відомості про походження чи властивості даних. У роботі запропоновано комбінований алгоритм захисту текстових і графічних файлів, орієнтований на усунення зазначених вразливостей. Алгоритм реалізує багаторівневу симетричну схему, що поєднує три етапи: AES-CBC, перестановку блоків та AES-GCM. Ключі формуються на основі пароля та унікальної солі, що підвищує стійкість до атак перебором. Основною метою є формування розподілу байтів зашифрованих даних, максимально наближеного до рівномірного, що мінімізує можливість їх відновлення методами статистичного аналізу. Для підтвердження ефективності алгоритму застосовано вбудовані в середовище Matlab статистичні тести, серед яких ентропійний, χ^2, Колмогорова–Смирнова та кореляційний. Результати продемонстрували значне зростання ентропії та зниження рівня кореляції між елементами після шифрування, що свідчить про досягнення властивості невизначуваності вхідних даних. Практичне використання алгоритму передбачає можливість гнучкого налаштування – користувач може обирати компроміс між рівнем криптостійкості та показниками стиснення під час формування захищених архівів. Таким чином, запропонований підхід забезпечує комплексний захист різних типів цифрової інформації, поєднуючи високу ефективність із адаптивністю до конкретних практичних потреб.

Ключові слова: шифрування, надлишковість, графічна та текстова інформація, автентифікація даних

Вступ. У сучасному цифровому світі обсяг даних, що постійно створюються, передається та зберігається, зростає в геометричній прогресії. Це зумовлює підвищену увагу до питань забезпечення конфіденційності [1] інформації. Якщо раніше основний акцент робився на захист текстових документів, то сьогодні значну частку цифрових даних становлять графічні матеріали – фотографії, скановані копії документів, мультимедійні файли тощо. Вони, як і текстові дані, можуть містити критично важливу або персональну інформацію, тому також потребують надійного захисту від несанкціонованого доступу. Попри значний прогрес у галузі криптографії, як текстові, так і графічні дані залишаються вразливими до певних типів атак через свої внутрішні структурні особливості. У текстових файлах зберігаються характерні статистичні закономірності – зокрема, частотний розподіл символів, слів або фраз, що може

використовуватись для криптоаналізу навіть за умови зашифрованого вмісту. Це особливо актуально при використанні слабких або однотипних шифрів, які не маскують мовну структуру даних [2,3]. У графічних файлах існують інші, не менш критичні ризики. Однією з основних вразливостей є висока кореляція між пікселями [4], особливо в зображеннях із природними сценами або слабким стисненням. Така статистична залежність дозволяє використовувати методи аналізу зображень або реконструкції для витягування прихованої або частково зашифрованої інформації. Крім того, зображення часто містять відкриті метадані (наприклад, EXIF, IPTC, XMP), які автоматично додаються камерами або програмним забезпеченням під час створення чи редагування файлу. Ці метадані можуть містити інформацію про модель пристрою, дату та час створення, координати GPS, параметри зйомки, авторство тощо [5]. Окрему небезпеку становлять службові заголовки файлів, які розкривають тип кодування, формат даних і можуть слугувати додатковим джерелом технічної або персональної інформації [6]. Всі ці особливості підкреслюють необхідність не лише використовувати сучасні методи шифрування, а й звертати увагу на приховані властивості даних, що потенційно можуть бути використані для інформаційного витоку або відновлення змісту.

Метою статті є розробка алгоритму, призначеного для забезпечення надійного криптографічного захисту як текстових, так і графічних даних із використанням сучасних методів шифрування. Передбачено, що створене рішення унеможливлюватиме визначення початкового формату даних після їх шифрування, а оцінка ефективності приховування характерних ознак здійснюється статистичними методами.

Для досягнення поставленої мети необхідно виконати такі завдання:

– провести огляд та аналіз сучасних методів криптографічного захисту інформації, визначивши їх слабкі місця та можливі вектори атак;

– дослідити особливості процесів шифрування даних різних форматів, зокрема текстових і графічних файлів;

– створити математичну модель, що забезпечує функції шифрування, розшифрування та аналізу зазначених типів даних;

– здійснити експериментальне дослідження статистичних характеристик зашифрованих даних з метою оцінки можливості визначення їхнього вихідного формату, а також перевірити методи приховування таких ознак.

Характерні риси та типи існуючих криптографічних методів захисту інформації. Усі криптографічні методи умовно класифікуються на дві основні групи – класичні та сучасні, які відрізняються між собою рівнем складності реалізації, стійкістю до криптоаналітичних атак і сферами практичного використання [7].

Класичні криптографічні методи ґрунтуються на простих математичних принципах і широко застосовувалися до появи комп’ютерних обчислювальних систем. До них належать, зокрема, підстановочні та перестановочні шифри [8].

Підстановочні шифри замінюють символи відкритого тексту іншими згідно з певним фіксованим правилом. Наприклад, моноалфавітні шифри, такі як шифр Цезаря, використовують один і той самий алфавіт протягом усього повідомлення, що робить їх надзвичайно вразливими до частотного аналізу, оскільки зберігаються статистичні закономірності мови. У поліалфавітних шифрах, наприклад у шифрі Віженера [9], використовується кілька алфавітів, що ускладнює розшифрування, однак вимагає більших обчислювальних зусиль.

На відміну від підстановки, перестановочні шифри не змінюють самі символи, а лише їхній порядок у повідомленні. Вони є простими в реалізації, але не порушують частотних характеристик тексту, що обмежує їх криптографічну стійкість. Загалом класичні методи шифрування не відповідають сучасним вимогам безпеки, адже легко піддаються статистичному аналізу та не забезпечують належного рівня захисту в умовах автоматизованої обробки даних.

Сучасна криптографія включає значно складніші алгоритми, які розроблялися з урахуванням високої обчислювальної потужності сучасних комп'ютерів і необхідності протистояти складним методам злому. Вона охоплює три ключові напрямки: симетричне шифрування, асиметричне шифрування та хешування [10].

Симетричні алгоритми. Симетричні методи базуються на використанні одного секретного ключа як для шифрування, так і для дешифрування повідомлення. Їх поділяють на дві категорії: блочні та потокові шифри.

Блочні шифри, такі як AES (Advanced Encryption Standard), обробляють дані блоками фіксованого розміру (наприклад, 128 або 256 біт). Вони характеризуються високою швидкістю та стійкістю до широкого спектра атак, однак за відсутності додаткових заходів (наприклад, режимів роботи) можуть зберігати структуру повідомлення, що призводить до витоку шаблонів. Повторення однакових блоків у відкритому тексті може зумовити повторення тих самих блоків у шифротексті. Для усунення цього недоліку використовуються режими роботи блочного шифрування [11], зокрема CBC (Cipher Block Chaining) та GCM (Galois/Counter Mode), які забезпечують диференціацію блоків завдяки введенню випадкових значень (ініціалізаційних векторів).

Потокові шифри, наприклад ChaCha20, працюють з потоком даних на рівні бітів або байтів, послідовно обробляючи інформацію з використанням ключового потоку. Завдяки високій продуктивності та низькій затримці, такі алгоритми ідеально підходять для шифрування потокових даних – аудіо, відео або реального часу передавання.

Асиметричні алгоритми. Асиметричні криптосистеми передбачають використання пари ключів – публічного та приватного. Публічний ключ застосовується для шифрування, а приватний – для дешифрування.

Одним із найвідоміших прикладів є RSA (Rivest–Shamir–Adleman), який базується на складності факторизації великих чисел [12] і використовується як для шифрування даних, так і для створення цифрових підписів.

Інший важливий алгоритм – Diffie–Hellman [13]– забезпечує безпечний обмін ключами між двома сторонами без попереднього узгодження, навіть через відкриті канали зв'язку. DSA (Digital Signature Algorithm) – спеціалізований алгоритм для створення цифрових підписів, який гарантує автентичність і цілісність повідомлень [14].

Основна перевага асиметричних методів полягає у відсутності необхідності передавати секретний ключ, що є критично важливим у розподілених та відкритих мережах. Проте їхнім недоліком є значно нижча швидкість у порівнянні із симетричними алгоритмами, що обмежує сферу їхнього використання – зазвичай, асиметрія застосовується лише для початкової фази встановлення захищеного зв'язку або цифрового підпису, тоді як самі дані шифруються симетрично.

Хешування. Хеш-функції є основним інструментом для забезпечення цілісності даних. Вони перетворюють вхідне повідомлення довільної довжини у вихідний рядок фіксованої довжини – хеш, або дайджест.

На відміну від шифрування, хешування є одностороннім процесом – отриманий хеш не може бути використаний для відновлення початкових даних.

Криптографічна хеш-функція повинна відповідати кільком важливим властивостям [15]:

- детермінованість – однакові вхідні дані завжди дають однаковий хеш;
- ефективність – обчислення хешу має виконуватись швидко;
- односторонність – неможливість відновлення вихідного повідомлення з хешу;
- лавиноподібний ефект – незначна зміна вхідних даних повинна спричиняти значну зміну хешу;
- стійкість до колізій – ймовірність того, що два різні повідомлення матимуть однаковий хеш, повинна бути мінімальною.

Завдяки цим характеристикам хеш-функції широко застосовуються у зберіганні паролів, перевірці цілісності файлів, створенні цифрових підписів, а також як фундаментальний елемент блокчейн-технологій та розподілених облікових систем.

Вразливості текстових даних. Текстові дані – це інформація, представлена у вигляді символів, слів або речень, яка використовується у широкому спектрі цифрових документів: текстових файлах (наприклад, TXT, DOCX, PDF), електронному листуванні, повідомленнях, програмному коді тощо. Завдяки своїй структурованості, стандартизованості й передбачуваності, такі дані є привабливими цілями для криптоаналітичних атак. Зокрема, характерні шаблони, частотний розподіл символів і граматичні закономірності можуть бути використані для спрощення процесу дешифрування. Саме ці особливості зумовлюють необхідність у глибокому криптоаналізі – галузі, що досліджує вразливості криптографічних алгоритмів та методів захисту.

Криптоаналіз – це прикладна і теоретична дисципліна [16], що має на меті виявлення недоліків у криптографічних системах, а також розробку методів обходу засобів захисту без знання секретного ключа. На відміну від криптографії, яка створює засоби захисту, криптоаналіз спрямований на їх випробування та злам. Він спирається на аналіз математичних структур алгоритмів, статистичні особливості даних, а також на слабкості в реалізації чи управлінні ключами.

Сучасний криптоаналіз включає декілька підходів:

1) атаки на криптосистеми:
   а) Known-Plaintext Attack (KPA) – аналізуючи відомі пари відкритого тексту та шифротексту, атакуючий може вивести ключ або вразливості шифру;
   б) Chosen-Plaintext Attack (CPA) – зловмисник може обирати відкриті тексти й отримувати їхні шифровані варіанти для аналізу зв’язків;
   в) Ciphertext-Only Attack (COA) – аналізуються лише шифротексти, часто з використанням статистичних методів;
   г) Chosen-Ciphertext Attack (CCA) – атакуючий має доступ до розшифрування певних шифротекстів, що дає змогу виявити алгоритмічні слабкості;
2) статистичний криптоаналіз:
   а) частотний аналіз – вивчає частоти появи символів у шифротексті, особливо ефективний проти простих підстановочних шифрів;
   б) лінійний криптоаналіз – використовує апроксимації між відкритим текстом, шифротекстом і ключем;
   в) диференційний криптоаналіз – досліджує вплив змін у відкритому тексті на шифротекст для виявлення залежностей;
3) атаки на ключі:
   а) Brute Force – повний перебір ключів;
   б) словникові атаки – використання баз даних поширених паролів;
   в) райдужні таблиці – попередньо згенеровані відповідності між хешами та вхідними даними;
4) математичні атаки:
   а) факторизація – для зламу RSA шляхом розкладу модуля на прості множники;
   б) атаки на основі дискретного логарифма – загрожують алгоритмам Diffie-Hellman та ECC.

Захист від криптоаналізу передбачає:

– використання криптостійких алгоритмів (AES, RSA, ECC);
– впровадження випадкових компонентів (IV, соль, хешування з ключем);
– режими шифрування, що знищують статистичні ознаки (CBC, GCM);
– застосування хеш-функцій із доказаною стійкістю до колізій (SHA-2, SHA-3, bcrypt, Argon2).

Особливості захисту текстових даних полягають у необхідності розриву мовних шаблонів. Наприклад, однакові слова не повинні давати однаковий шифротекст. Для цього застосовуються режимні алгоритми з випадковими послідовностями символів, а також методи згладжування ентропії.

Вразливості графічних даних. Для графічних даних загрози мають інший характер [17]. Графіка, особливо у форматах JPEG, PNG, BMP, зберігає просторові й колірні кореляції, які можуть бути використані при атаках:

- атаки за типом пазлу (Jigsaw puzzle attack);
- атаки на сусідні пікселі (Neighbor attack);
- частотний аналіз кольорових каналів.

Для протидії цим атакам використовуються методи повного перемішування даних (наприклад, блокові шифри з комбінованими перестановками та підстановками) та хаотичні перетворення, які повністю руйнують структуру зображення. Крім того, обов'язковим є видалення або шифрування метаданих (EXIF, заголовки), які можуть зберігати чутливу інформацію навіть після шифрування пікселів.

Алгоритм (модуль) шифрування текстової інформації. Розглянемо алгоритм шифрування текстової інформації.

Для практичного застосування алгоритмів шифрування текстової інформації розроблено низку програмних рішень, що спрощують процес захисту даних та відповідають сучасним вимогам безпеки. Серед них найбільш поширеними є AES Crypt і AxCrypt [18]. Одне з рішень, забезпечує швидке шифрування файлів за допомогою алгоритму AES-256 без потреби у складному налаштуванні. Натомість AxCrypt пропонує розширені можливості: наскрізне шифрування, інтеграцію з хмарними сервісами та підтримку різних операційних систем – від Windows і macOS до Android та iOS [19].

Алгоритм AES обрано основним методом шифрування завдяки його високій стійкості до сучасних криптографічних атак, оптимальному співвідношенню швидкодії та рівня захищеності, а також широкій підтримці на різних платформах.

Advanced Encryption Standard (AES) є симетричним блочним шифром, де дані поділяються на блоки розміром 128 біт. Такий розмір позначається як $N_b = 4$, що відповідає чотирьом 32-бітним словам. Довжина секретного ключа може становити 128, 192 або 256 бітів, тобто $N_k = 4{,}6$ або 8 слів по 32 біти. Від вибору довжини ключа залежить і кількість раундів (N_r) у процесі шифрування: для AES-128 це 10 раундів, для AES-192 – 12, а для AES-256 – 14, як зазначено в табл. 1. Під час розширення ключа (Key Expansion) формується $N_r + 1$ раундових ключів (Round Keys) $K_0, K_1, \ldots, K_{Nr}$, кожен з яких має ті ж 128 біт, що збігається з розміром блоку [20].

Таблиця 1.

Комбінації ключ-блок-раунд

Алгоритм	Розмір блоку (Nb слів)	Довжина ключа (Nk слів)	Кількість раундів (Nr)
AES-128	4	4	10
AES-192	4	6	12
AES-256	4	8	14

Під час шифрування вхідний блок даних (plaintext) розміром 128 біт подається у вигляді матриці 4×4, яка називається State**.** Кожен її елемент ($a_{i,j}$) відповідає одному байту. Формування матриці здійснюється за стовпцевим принципом: спочатку послідовно заповнюється перший стовпець зверху вниз, далі – другий і так далі.

Після виконання всіх етапів перетворення ця матриця State набуває вигляду зашифрованого блоку (ciphertext).

Розглянемо приклад успішного тесту, який наведено в табл. 2.

Таблиця 2.

Параметри шифрування та розшифрування

Параметр	Значення
Ключ (16 символів)	hGkLmNpQrStUvWxY
Вихідний текст	secret
Зашифрований текст (HEX)	cf970df14a9433bbed1ff2cac9299dd295cd3aceda7d6db5cbee68 883d90ab63
Розшифрований текст	secret

При дотриманні вимог до довжини ключа та коректному введенні вихідного тексту система безпомилково шифрує та розшифровує дані. Як можна побачити, процедура шифрування сформувала випадково «cf97» та додала його до початку результату, а розшифрування з тим самим ключем відновило вихідне слово без помилок. Якщо б ключ було змінено хоча б на один символ, алгоритм не зміг би правильно дешифрувати повідомлення.

Отже, модуль AES, робота якого тестувалася у пакеті Matlab успішно пройшов тестування на коректність роботи у звичайних і виняткових ситуаціях, що підтверджує надійність обраного режиму CBC та реалізації алгоритму AES, наприклад, у вигляді окремого застосунку.

Для наочного представлення результатів було побудовано два графіки. На першому (рис 1) ліворуч відображено розподіл байтів вихідного файлу, де простежується концентрація значень у межах типового діапазону ASCII-символів (близько 32–122 у десятковому представленні), що властиво текстовим даним. Праворуч наведено розподіл байтів зашифрованого файлу, який рівномірно охоплює майже весь діапазон від 0 до 255, що свідчить про втрату видимої структури після шифрування.

У результаті обчислень зафіксовано зростання ентропії файлу до ~7,99 біт/байт, а показник розподілу імовірностей χ^2 наблизився до значення, характерного для випадкових послідовностей, що підтверджує ефективне усунення вихідних статистичних закономірностей.

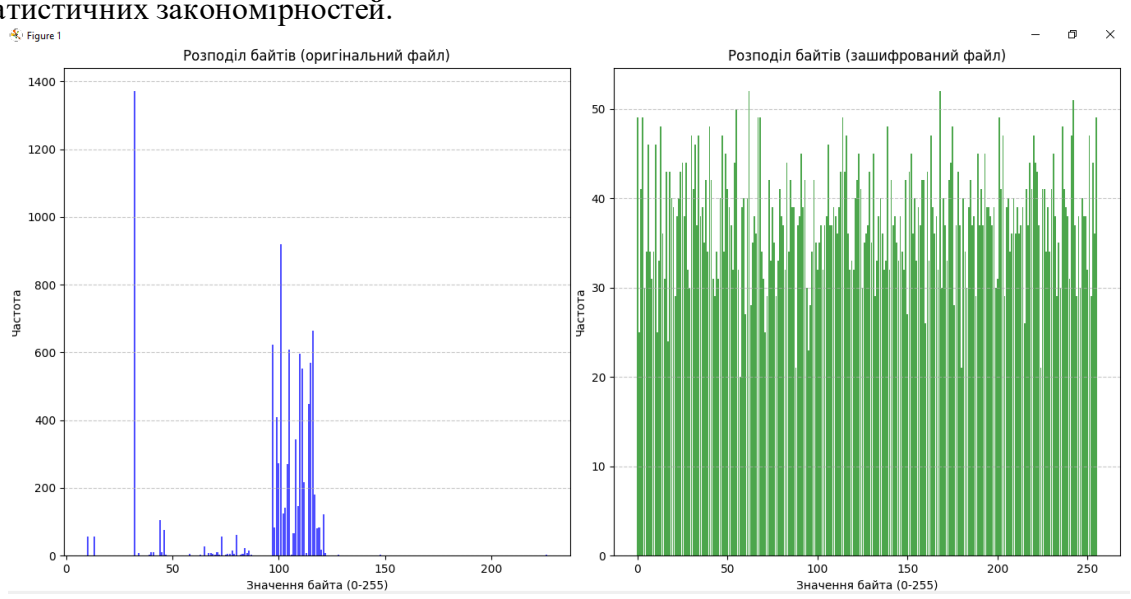

Рис. 1. Розподіл байтів у текстовому файлі до та після шифрування

Другий графік (рис. 1) відображає порівняння нормованих частот появи байтів у вихідному та зашифрованому файлах.

У початковому файлі помітні виражені піки, що відповідають найбільш уживаним символам, тоді як у зашифрованому файлі розподіл є майже рівномірним. Це свідчить про підвищену ентропію даних та ефективність застосованого методу шифрування.

Подібна візуалізація чітко демонструє суттєві зміни у структурі даних після шифрування, що підтверджується статистичними показниками та засвідчує високий рівень криптографічного захисту.

Алгоритм (модуль) шифрування графічної інформації із урахуванням статистичних характеристик. Алгоритм шифрування графічної інформації (наприклад, зображень) повинен враховувати специфіку таких даних – великий обсяг, сильну корельованість пікселів та вимоги до швидкодії. Також, так як вимогою до модулю, що розробляється є умова ускладнення процедури розрізнення типу контенту після шифрування, необхідно ввести критерій складності алгоритму аналізу зашифрованого контенту. У якості такого критерію може виступати допуск на цільове значення ентропії, що обчислюється із умови відхилення закону розподілу байтів зашифрованого файлу від рівно ймовірного.

Було запропоновано наступний алгоритм.

Крок 1. Задається початковий допуск на цільове значення ентропії графічного файлу ΔH в діапазоні 0.1…8 бітів на піксель. Задаємо значення лічильнику кількості проходів алгоритму j=0.

Крок 2. Завантажується графічний файл і здійснюється його перекодування в нестиснений формат, наприклад BMP.

Крок 3. Здійснюється обчислення обсягу зображення J в бітах. Зображення представимо у вигляді масиву із m рядків та n стовпців. Яскравість кожного пікселю зображення кодується за допомогою k розрядів. Якщо зображення кольорове то $k=k_r+k_b+k_g$, де k_r – кількість розрядів яскравості червоного каналу, k_b – кількість розрядів яскравості синього каналу, k_g – кількість розрядів яскравості червоного каналу, k_b – кількість розрядів яскравості зеленого каналу. Таким чином:

$$J = n \bullet m \bullet Log_2 k \text{ бітів.}$$

Крок 4. Обчислюємо вихідне значення ентропії H_i нестисненого зображення в бітах на піксель:

$$H_i = -\sum_{i=0}^{2^k-1} p_i Log_2(p_i),$$

де p_i – ймовірність появи пікселю із яскравістю, що дорівнює i.

Загальне значення ентропії H_{it} для всього зображення (m×n пікселів):

$$H_{it} = H_i \bullet m \bullet n$$

Крок 5. Обчислюємо максимально можливе значення ентропії H_m нестисненого зображення. Максимально можливе значення ентропії досягається при умові, що всі ймовірності появи пікселю із довільною яскравістю однакові та дорівнюють: $p_i = \frac{1}{2^k}$.

Таким чином: $H_m = -\sum_{i=0}^{2^k-1} \frac{1}{2^k} Log_2\left(\frac{1}{2^k}\right) = Log_2 2^k = k$ бітів на піксель. Тобто загальне значення ентропії H_{mt} для всього зображення (m×n пікселів):

$$H_{mt} = H_m \bullet m \bullet n$$

Крок 6. Обчислюємо різницю ентропії, що обчислені на кроках 5 та 4:

$$\Delta\hat{H} = H_m - H_i$$

Крок 7. Якщо $\Delta\hat{H} \le \Delta H$ переходимо на крок 9. Шифруємо файл із додаванням ентропії із використанням алгоритму AES-128 чи AES-256 чи AES-512 в залежності від номеру проходу j. Збільшуємо лічильник кількості проходів j = j + 1.

Крок 8. Якщо j > 3, збільшуємо початковий допуск на цільове значення ентропії графічного файлу ΔH= ΔH+0.1, прирівнюємо j=0 і переходимо на крок 2.

Крок 9. Здійснюємо стиснення зображення в необхідному форматі чи формуємо zip архів.

Для наочного відображення результатів було виконано візуалізацію даних у формі трьох графіків. На рисунку 3 представлено розподіл значень RGB-каналів вихідного оригінального зображення (Lena) (рис 2 а) розміром 512×512 пікселів, що є стандартним для задач тестування алгоритмів графічної обробки зображень.

а б

Рис. 2. Вигляд тестувального зображення «Лена» (джерело: [21]): а) оригінальне зображення; б) зашифроване зображення

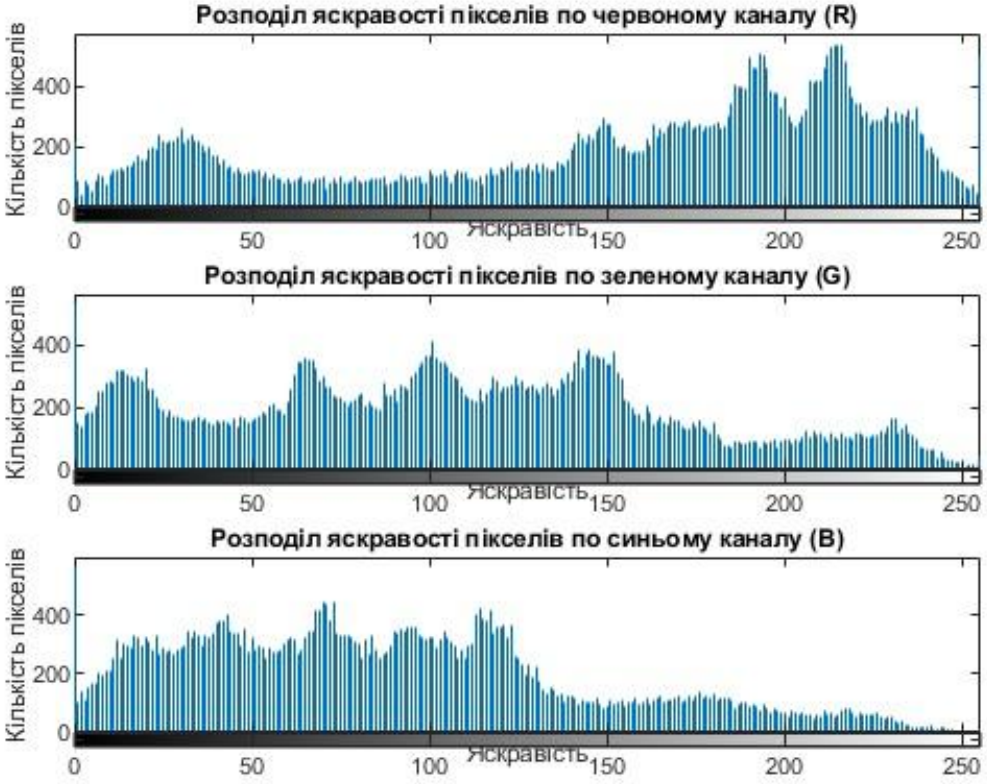

Рис. 3. Розподіл значень яскравості RGB-каналів у зображенні «Лена» до шифрування

З аналізу рисунку 3, можна побачити, що для оригінального файлу, спостерігаються характерні піки й спадини, що виникають внаслідок JPEG-стиснення. Видно, що основна частина інтенсивностей зосереджена поблизу мінімальних та

максимальних значень яскравостей (близько 0 і 255), що характерно для JPEG-файлів із підвищеною контрастністю або наявністю великих однотонних областей.

На рисунку 4 подано розподіл байтів у файлі після JPEG-стиснення до та після шифрування.

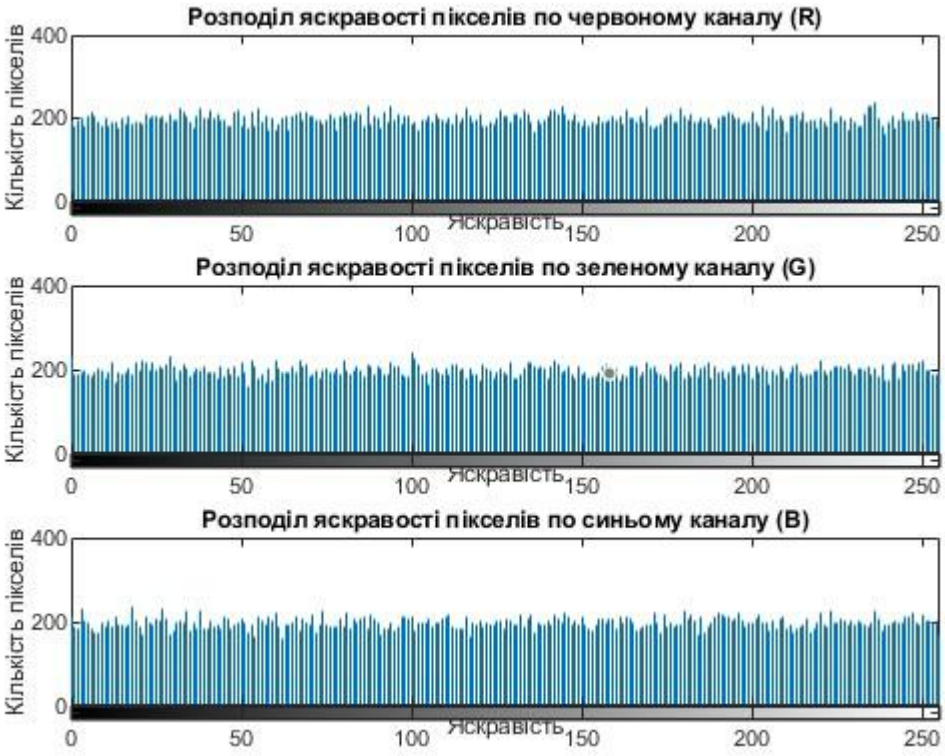

Рис. 4. Розподіл значень байтів яскравості RGB каналів у графічному файлі після шифрування

У зашифрованому файлі, рис 4, розподіл значень у межах 0 – 255 набуває майже рівномірного характеру при заданому цільовому значенні ентропії 4 біта на піксель.

Слід зазначити, що шифрування зображень за алгоритмами AES-128 чи AES-256 чи AES-512 суттєво знижують ефективність алгоритмів стиснення без втрат, таких як кодування кодом Гафмана, а після JPEG-стиснення якість зображення може бути гіршою в порівнянні із випадком відсутності шифрування.

Висновки. Встановлено, що класичні методи шифрування, зокрема шифр Цезаря, шифр Віженера, а також алгоритми DES і RC4, є недостатньо надійними, оскільки вразливі до частотного та статистичного аналізу через обмежений простір ключів і передбачуваність структури шифротексту. Натомість сучасні криптографічні алгоритми – AES (FIPS 197, офіційний стандарт NIST) у режимах CBC та GCM, RSA (PKCS #1) і ECC (ANSI X9.62) – за умови правильного вибору режимів роботи та належного управління ключами демонструють високу стійкість до частотного, диференційного й лінійного криптоаналізу.

Розроблений алгоритм реалізує багаторівневу симетричну схему, що поєднує три етапи: AES-CBC, перестановку блоків та AES-GCM, із використанням ключів, отриманих на основі пароля та солі. Такий підхід забезпечує захист як текстових, так і графічних файлів. Вбудований модуль статистичного аналізу (ентропійний, χ^2, Колмогорова–Смирнова та кореляційний тести) підтвердив ефективність перевірки властивості невизначуваності типу вхідних даних після шифрування.

Розроблений алгоритмічний модуль щодо адаптивного шифрування графічної інформації враховує такі її особливості як висока корельованість пікселів, особливо при наявності на зображенні фону із рівномірною яскравістю. З метою додаткового ускладнення ідентифікації типу контенту після шифрування було запропоновано

алгоритм, що заснований на послідовному досягненні цільового рівня ентропії, що задається користувачем на етапі підготовки даних.

Експериментальні дослідження показали, що зашифровані файли за своїми статистичними характеристиками наближені до випадкового шуму, що унеможливлює ідентифікацію їхнього початкового формату. Перевірка працездатності алгоритму шляхом моделювання в середовищі Matlab довела його придатність для автоматизованого захисту бухгалтерської та юридичної документації.

Список літератури

1. Сирбу К. Д., Садченко А. В. Розробка застосунку для криптографічного захисту текстових та графічних даних. *Безпека інформаційних технологій: матеріали XIV Міжнар. наук.-техн. конф., м. Тернопіль*. 2025. С. 175-178.
2. Константинова Л. В., Норов А. О. Криптографія та стеганографія: різниця та застосування в захисті інформації. *VI Міжнародна науково-практична конференція Інформаційна безпека та комп'ютерні технології*. 2023. С.5-7.
3. Kordov K. Text_encryption_algorithm_for_secure_communication. *International Journal of Applied Mathematics*. 2021. Vol. 34. No. 4. P. 705. URL: http://dx.doi.org/10.12732/ijam.v34i4.9
4. Sabonchi A. Kh. Sh., Akay B. A survey on the Metaheuristics for Cryptanalysis of Substitution and Transposition Ciphers. *Computer Systems Science & Engineering*. 2021. Vol. 39. No. 1. P. 47.
5. Орловський В. О. Системи і технології захисту інформації при її обміні між користувачами. *Міжнар. наук. інтернет-конф*. 2022. С.50. URL: www.konferenciaonline.org.ua.
6. Trysnyuk V., Smetanin K., Humeniuk I., Samchyshyn O. Information Encryption Method based on a Combination of Steganographic and Cryptographic Algorithm's Features. *Cybersecurity Providing in Information and Telecommunication Systems*. 2021. P. 150-159. URL: https://ceur-ws.org/Vol-3187/paper14.pdf
7. Prajapat S., Thakur R. S. Various approaches towards cryptanalysis. *International Journal of Computer Applications*. 2015. Vol. 127, No. 14. P. 15–24.
8. Schroé W. Cryptanalysis of Submission to the CAESAR Cryptographic Competition iFeed. URL: https://cosicdatabase.esat.kuleuven.be/backend /publications/files/these/262.
9. Najaftorkaman M., Kazazi N. S. A method to encrypt information with DNA-based cryptography. *International Journal of Cyber-Security and Digital Forensics*. 2015. Vol. 4. No. 3. P. 417–426.
10. Rhea Gupta, Sara Dharadhar, Prathamesh Churi; CloudJS: novel cloud-based design framework for text-file encryption. *World Journal of Engineering*. 2023. Vol.20 (3). P. 472–485. URL: https://doi.org/10.1108/WJE-04-2021-0234
11. Лужецький В. А., Кисюк Д. В. Метод хешування даних на основі їх характеристичних ознак. *II Міжнар. наук.-практ. конф. «Актуальні питання забезпечення кібербезпеки та захисту інформації»*. 2016. С. 100–102.
12. Blondeau C., Leander G., Nyberg K. Differential-linear cryptanalysis revisited. *Journal of Cryptology*. 2017. Vol. 30. P. 859–888.
13. Data Encryption Methods & Types: A Beginner's Guide. URL: https://www.splunk.com/en_us/blog/learn/data-encryption-methods-types.html
14. Mourouzis T., Song G., Courtois N., Christofi M. Advanced differential cryptanalysis of reduced-round SIMON64/128 using large-round statistical distinguishers. *Cryptology ePrint Archive*. 2015. Report 2015/481. P. 1–9.
15. Noor N.S., Hammood D.A., Al-Naji A., Chahl J. A Fast Text-to-Image Encryption-Decryption Algorithm for Secure Network Communication. *Computers*. 2022. Vol.11(3): 39. URL: https://doi.org/10.3390/computers11030039

16. Jebur, R.S.; Der, C.S.; Hammood, D.A. A Review and Taxonomy of Image Denoising Techniques. In Proceedings of the 6th International Conference on Interactive Digital Media (ICIDM), Bandung, Indonesia, 14–15 December 2020; pp. 1–6.
17. Aparna V. S., Rajan A., Jairaj I., Nandita B., Madhusoodanan P and Remya A. S., *Implementation of AES Algorithm on Text And Image using MATLAB. 3rd International Conference on Trends in Electronics and Informatics. India.* 2019. P. 1279-1283. DOI: 10.1109/ICOEI.2019.8862703
18. Huang X., Dong Y., Ye G., Yap W.S., Goi B.M. Visually meaningful image encryption algorithm based on digital signature. *Digital Communications and Networks*. 2023. Vol. 9. Is.1., P 159-165. URL: https://doi.org/10.1016/j.dcan.2022.04.028.
19. Abusukhon A., AlZu'bi S. New Direction of Cryptography: A Review on Text-to-Image Encryption Algorithms Based on RGB Color Value. 2020 *Seventh International Conference on Software Defined Systems. France*. 2020. P. 235-239, DOI: 10.1109/SDS49854.2020.9143891.
20. Abusukhon A., Anwar M.N., Mohammad Z., Alghannam B. A Hybrid. Network Security Algorithm Based on Diffie Hellman and Text-to-Image Encryption Algorithm. URL:https://researchportal.northumbria.ac.uk/ws/portalfiles/portal/18728434/Abusukhon_et_al_A_hybrid_network_security_algorithm_based_on_Diffie_Hellman_and_Text_to_Image_Encryption_algorithm_AAM.pdf
21. Banerjee M., Naskar S., Basuli K., Sarma S.S. A novel scheme for text data encryption. *Journal of Global Research in Computer Science.* 2012. Vol. 3. P 39-41. URL: https://www.rroij.com/open-access/a-novel-scheme-for-text-data-encryption-39-41.pdf

DEVELOPMENT OF A CONTENT-ADAPTIVE ALGORITHM FOR ENCRYPTION OF CONFIDENTIAL INFORMATION

A. V. Sadchenko, O. A. Kushnirenko

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Email: koa@op.edu.ua

In the modern information environment, the protection of digital data is becoming increasingly important, among which an important share is made up of graphic materials - photographs, scanned documents and multimedia files. They, as well as text data, can contain personal or critically important information that requires reliable encryption and storage mechanisms. However, even with the availability of advanced cryptographic methods, text and graphic files remain vulnerable due to specific structural features.Text data typically exhibits statistical patterns, such as the frequency distribution of characters and words, which can be exploited for cryptanalytic attacks when weak or similar ciphers are used. In graphic files, one of the key risks is the high correlation between neighboring pixels, especially in the case of natural images or data with minimal compression, which creates the possibility of reconstructing or analyzing even partially encrypted information. An additional threat is metadata (EXIF, IPTC, XMP), which is automatically added when creating or editing files and may contain confidential information about the origin or properties of the data. The paper proposes a combined algorithm for protecting text and graphic files, aimed at eliminating the above vulnerabilities. The algorithm implements a multi-level symmetric scheme that combines three stages: AES-CBC, block permutation and AES-GCM. The keys are formed based on a password and a unique salt, which increases resistance to brute force attacks. The main goal is to form a distribution of bytes of encrypted data that is as close to uniform as possible, which minimizes the possibility of their recovery by statistical analysis methods. To confirm the effectiveness of the algorithm, statistical tests built into the Matlab environment were used, including entropy, χ^2, Kolmogorov–Smirnov, and correlation. The results demonstrated a significant increase in entropy and a decrease in the level of correlation between elements after encryption, which indicates the achievement of the property of uncertainty of the input data. Practical use of the algorithm provides for flexible configuration - the user can choose a compromise between the level of cryptographic strength and compression rates when creating protected archives. Thus, the proposed approach provides comprehensive protection of various types of digital information, combining high efficiency with adaptability to specific practical needs.

Keywords: encryption, redundancy, graphic and text information, data authentication.