Таким чином, кожен тип VPN-рішень має свої переваги і недоліки з точки зору конфіденційності та безпеки.

Комерційні VPN пропонують зручність і додаткові функції, але іноді можуть бути ризики, пов'язані з юрисдикцією та політикою логування.

Корпоративні VPN забезпечують контроль і безпеку в межах організації, але вимагають адміністративних ресурсів.

Open-source VPN забезпечують максимальну прозорість і контроль, проте потребують технічних знань для налаштування і підтримки.

З огляду на численні проблеми, пов'язані з приватністю та надійністю VPN-сервісів, побудова власного VPN-рішення є найбільш ефективним шляхом для забезпечення конфіденційності та контролю над власними даними. Створення власного VPN на базі поширених технологій (наприклад, OpenVPN чи WireGuard) дозволяє повністю керувати інфраструктурою, мінімізуючи ризики витоку даних та забезпечуючи прозорість роботи системи.

Mazurek, "Investigating influencer vpn ads on youtube,"
in IEEE Symposium on Security and Privacy (SP), 2022
O. Akgul, R. Roberts, M. Namara, D. Levin, and M. L.
Mazurek, "Investigating influencer vpn ads on youtube,"
in IEEE Symposium on Security and Privacy (SP), 2022
O. Akgul, R. Roberts, M. Namara, D. Levin, and M. L.
Mazurek, "Investigating influencer vpn ads on youtube,"
in IEEE Symposium on Security and Privacy (SP), 2022

1. Lin K., Xiao Y., Chen J. As Advertised: Understanding the Impact of Influencer VPN Ads [Електронний ресурс]. ResearchGate, 2024. URL: https://www.researchgate.net/publication/381579245_As_Advertised_Understanding_the_Impact_of_Influencer_VPN_Ads (дата звернення: 15.04.2025).

2. vpnMentor. Report: Free VPNs Leak Data [Електронний ресурс]. 2021. URL: https://www.vpnmentor.com/blog/report-free-vpns-leak/ (дата звернення: 15.04.2025).

Розробка алгоритму для криптографічного захисту текстових та графічних даних

УДК 004.056.5:004.4

Катерина Сирбу

Національний університет «Одеська політехніка», 9480559@stud.op.edu.ua

У роботі представлено програмний застосунок для криптографічного захисту текстових і графічних даних, що поєднує шифрування за алгоритмом AES, цифровий підпис RSA, аналіз ентропії та статистичних характеристик файлів. Метою дослідження є забезпечення високого рівня конфіденційності оброблених файлів через багаторівневе шифрування та зменшення ймовірності ідентифікації початкового типу даних на основі залишкових статистичних характеристик після шифрування. Реалізована система дозволяє не лише шифрувати, а й виявляти потенційно вразливі структури у файлах. Ефективність застосунку підтверджено тестуванням на різних форматах даних.

Зростання обсягів переданої та збереженої цифрової інформації супроводжується ускладненням методів несанкціонованого доступу. Стандартні схеми шифрування, як правило, забезпечують конфіденційність змісту, але можуть залишати статистичні ознаки, за якими зловмисник здатен зробити припущення про тип вихідних даних. Це створює ризики витоку метаінформації, у разі використання засобів автоматизованого виявлення структурних закономірностей у зашифрованих даних. У таких випадках потрібні підходи, що не лише шифрують дані, а й унеможливлюють визначення їхнього початкового формату.

Захист конфіденційної інформації у цифровому середовищі вимагає застосування надійних криптографічних методів. Текстові файли часто мають статистично передбачувану структуру, графічні – містять метадані та просторові кореляції, що знижує ефективність базового шифрування [2]. Робота спрямована на подолання цих недоліків шляхом створення застосунку, здатного одночасно шифрувати дані та оцінювати ступінь їхньої криптостійкості.

Для досягнення цієї мети у застосунку реалізовано багаторівневу систему криптографічного захисту, яка забезпечує комплексну обробку даних на основі сучасних методів симетричного шифрування та структурного ускладнення зашифрованого потоку. Основу захисту файлів у застосунку становить модуль багаторівневого шифрування, реалізований у класі Encryptor. Він поєднує три послідовні етапи:

1) AES у режимі CBC — виконується базове шифрування з використанням ключа key1, отриманого через похідну функцію від пароля й солі [1];

2) перестановка блоків — за допомогою key2 (16 байт), що виступає насінням генератора випадкових чисел, зашифровані 16-байтні блоки перемішуються у псевдовипадковому порядку;

3) AES у режимі GCM — застосовується до результату перестановки з ключем key3, забезпечуючи як конфіденційність, так і автентичність даних (генерується тег перевірки цілісності).

Фінальний зашифрований файл містить службовий заголовок з усіма необхідними параметрами для розшифрування (сіль, вектор ініціалізації (IV), тег, розмір блоків, розширення файлу тощо) і тіло зашифрованих даних. Розшифрування відбувається у зворотному порядку з перевіркою цілісності та відновленням початкової структури файлу.

З метою оцінки ефективності шифрування та подальшого виявлення структурних закономірностей, у застосунку реалізовано модуль аналізу файлів, який досліджує як початкові, так і оброблені шифруванням дані. Цей модуль дозволяє виявити залишкові ознаки структури у зашифрованих файлах, а також порівняти властивості даних до і після криптографічної обробки. У межах аналізу враховуються такі характеристики:

- обчислення загальної та блочної ентропії (метод Шеннона);
- частотний аналіз символів, біграм і триграм у текстових файлах;
- аналіз розподілу каналів RGB у зображеннях з побудовою гістограм;

- вилучення метаданих (наприклад, EXIF з графічних файлів або службової інформації з документів);
- побудова порівняльних графіків частот до та після шифрування;
- обчислення статистичних метрик — тести Колмогорова–Смирнова та хі-квадрат, серійна кореляція та KL-дивергенція [3].

Для реалізації зазначених функцій було створено повнофункціональний програмний застосунок, написаний мовою Python із використанням сучасних бібліотек PyCryptodome для криптографічних операцій, NumPy та Pandas для обробки даних, а також з графічним інтерфейсом, реалізованим засобами Tkinter.

Для кількісного підтвердження ефективності реалізованого шифрування було проведено статистичний аналіз текстового файлу до та після криптографічної обробки. За результатами дослідження спостерігається зростання загальної ентропії з 4.46 до 7.98 біт/байт, що свідчить про значне зростання випадковості даних [4]. Значення p за критерієм χ^2 зросло з 0.0000 до 0.9538, що демонструє вирівнювання розподілу байтів до рівномірного — характерного для шифрованих даних.

KL-дивергенція зменшилась у понад 200 разів (з 3.5351 до 0.0163), що підтверджує втрату схожості із початковим розподілом. Серійна кореляція знизилася до майже нульового рівня, вказуючи на знищення послідовних зв'язків у потоці даних. Отримані зміни свідчать про руйнування впорядкованих структур і формування байтового розподілу, близького до випадкового, що суттєво ускладнює проведення частотного аналізу та реалізацію атак, заснованих на виявленні візуальних або статистичних закономірностей. У таблиці 1 наведено ключові характеристики оригінального текстового файлу та його зашифрованої версії.

Таблиця 1

Порівняльний аналіз статистичних характеристик файла до та після шифрування

Показник	Оригінальний файл (text.txt)	Зашифрований файл (text.enc)
Розмір файлу (байт)	9540	9632
Перші 16 байтів (hex)	50 6f 6c 69 74 69 63 61 6c 20 73 63 69 65 6e 63	d0 15 1c 60 75 6b 32 8c 3f 86 8f 18 5d e8 c3 af
Загальна ентропія (біт/байт)	4.4649	7.9837
Середня блочна ентропія (1024 байт)	4.3886	7.7838
p-значення Chi-square	0.0000	0.9538
KL-дивергенція	3.5351	0.0163
Серіальна кореляція	0.0455	–0.0068
Тип файлу (за метаданими)	Text file	Encrypted/Binary file (likely encrypted)

Таким чином, було розроблено застосунок, який поєднує багаторівневе шифрування з оцінкою статистичних характеристик даних. На відміну від типових засобів криптографічного захисту, система дозволяє не лише зашифрувати інформацію, а й оцінити ефективність шифрування на основі таких показників, як ентропія, KL-дивергенція, p-значення χ^2 та серійна кореляція. Отримані результати підтверджують, що застосунок забезпечує надійне руйнування структур текстових і графічних даних, знижуючи їхню передбачуваність і вразливість до статистичних атак. Це робить його корисним інструментом для галузей, де важливо не лише зашифрувати, а й верифікувати рівень криптостійкості захищених файлів — зокрема, у цифровій криміналістиці, розробці безпечного програмного забезпечення, аудиті інформаційної безпеки.

1. Gavaskar, K., et al. AES Algorithm using Dynamic Shift Rows, Sub Bytes and Mix Column Operations for Systems Security with Optimal Delay. – 2022.
2. Prajapat S., Thakur R. S. Various approaches towards cryptanalysis // International Journal of Computer Applications. – 2015. – V. 127. – №14. – P. 15-24.
3. Zhou, Xinping, Kexin Qiao, and Changhai Ou. Leakage detection with Kolmogorov-Smirnov test // Cryptology ePrint Archive. – 2019.
4. Zolfaghari B., Bibak K., Koshiba T. The odyssey of entropy: cryptography // Entropy. – 2022. – V. 24. – №2. – P. 266.

Algorithms in Cybersecurity: Encryption and Hashing

UDC 004.056.55
Marharyta Sytnyk[1], Oleksandr Oliinyk[2]
Kharkiv National University of Radio Electronics,
[1]*marharyta.sytnyk@nure.ua*, [2]*oleksandr.oliinyk1@nure.ua*

In the digital age, where information technologies permeate all areas of our lives, data protection has become one of the most important issues. Today, we perform hundreds of actions online every day: sending messages, shopping, accessing banking services, registering on websites. All these operations involve the processing of personal information, which must remain protected from unauthorized access [1].

Information security means a state of data in which their confidentiality, integrity, availability, and resilience to external threats are guaranteed. With the development of global networks, new risks have emerged that complicate security assurance: malware, phishing attacks, data leaks, etc. The Internet is an open environment where confidentiality is not guaranteed without additional control measures. The easier it is to access the network, the higher the risk of data leakage or tampering during transmission [1].

That is why cryptographic algorithms are used to protect information - mathematical methods that ensure data preservation even if an attacker gains access to it. Currently, the most common are open encryption algorithms, which are publicly available so anyone can verify their reliability, while the content remains inaccessible without the appropriate key [1].