ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА ЕЛЕКТРОННА ІНЖЕНЕРІЯ
INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONIC ENGINEERING

№ 5 (2), 2025

https://doi.org/10.23939/ictee2025.02.038

СЕЛЕКТИВНЕ ШИФРУВАННЯ ВІДЕОІНФОРМАЦІЇ НА ОСНОВІ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ З ВИКОРИСТАННЯМ НЕЙРОННОЇ МЕРЕЖІ U-NET

І. Малінін[1] [ORCID: 0009-0005-4066-6231], **Ю. Коляденко**[1] [ORCID: 0000-0002-0247-2736],
Б. Жураковський[2] [ORCID: 0000-0003-3990-5205], **О. Коваленко**[1] [ORCID: 0000-0001-9082-6014],
В. Ралко[3] [ORCID: 0000-0001-6724-5000]

[1] Харківський національний університет ім. В. Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна
[2] Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Берестейський проспект, 37, Київ, 03056, Україна
[3] Харківський національний університет Повітряних сил ім. Івана Кожедуба, вул. Полтавський Шлях, 123, Харків, 61123,Україна

Відповідальний за рукопис: Ігор Малінін (e-mail: vvbar.off@gmail.com).

(Подано 29 січня 2024)

У цифрову епоху відеоінформація посіла провідне місце серед типів даних за обсягом і значущістю. Щоденно великі масиви візуальних даних створюються за допомогою відеокамер, мобільних гаджетів, безпілотників і мережевих сервісів. Значна частина цього контенту може містити персональні або конфіденційні відомості. Хоча традиційне повне шифрування відеопотоку гарантує високий рівень захисту, воно супроводжується низкою недоліків: високим навантаженням на обчислювальні ресурси, затримками під час передавання даних і складнощами в реалізації опрацювання у реальному часі. У роботі розв'язано науково-прикладну задачу селективного криптографічного захисту відеоінформації на основі автоматичної ідентифікації об'єктів за допомогою нейронної мережі U-Net. Запропоновано метод багатораундового шифрування, який реалізує глобальну перестановку пікселів і побайтову заміну їх значень у виділеній області зображення. Мета дослідження полягає у розробленні ефективного методу криптографічного захисту цільових областей зображення після їх ідентифікації, з акцентом на дифузію, конфузію та захист від статистичного аналізу. Для генерації ключового матеріалу використано систему зв'язаних хаотичних наметових відображень, а для параметризації – механізм PBKDF2 із криптографічною сіллю. У роботі також розглянуто застосування операцій XOR та модульного додавання, які чергуються між раундами, що підвищує стійкість шифру до атак. Виконано експериментальний аналіз результатів, який підтверджує ефективність і стійкість розробленого методу до змін у вхідних даних та ключовій інформації. Результатом роботи стало розроблення нового дев'ятираундового симетричного алгоритму шифрування для виділених ROI. Цей алгоритм перетворює наданий користувачем пароль та унікальну випадкову сіль на увесь необхідний ключовий матеріал за допомогою стандартної функції посилення ключа. Кожен із дев'яти раундів шифрування передбачає дві основні операції: глобальну перестановку всіх пікселів у ROI та побайтову заміну їх значень. Для операції заміни використано ключові потоки, що генеруються системою зв'язаних хаотичних відображень, налаштування якої індивідуальні для кожного раунду. Тип операції заміни чергується між раундами. Для перевірки цілісності та автентичності зашифрованих даних використано стандартний механізм коду автентифікації повідомлень.

Ключові слова: *селективне шифрування, нейронна мережа U-Net, семантична сегментація, криптографічна стійкість, хаотичне відображення, відеозахист, PBKDF2, дифузія, конфузія, ROI.*
УДК: 621.126

Вступ

Загальна схема багатораундового шифрування для виділеної області зображення передбачає послідовне виконання фіксованої кількості раундів (у поточній реалізації – дев’яти). Кожен раунд послідовно застосовує дві основні криптографічні операції до поточного масиву пікселів, які підлягають шифруванню [1–2].

Спочатку виконується глобальна перестановка пікселів, під час якої просторове розташування усіх вибраних пікселів псевдовипадково змінюється. Для кожного раунду ця перестановка керується унікальним початковим значенням для генератора перестановок, що забезпечує різне перемішування. Основна мета цієї операції – руйнування статистичних зв’язків між сусідніми пікселями та забезпечення ефекту дифузії [3–4].

Після перестановки здійснюється заміна значень пікселів, із модифікацією значень колірних компонент (R, G, B) кожного, вже переставленого, пікселя. Ця модифікація відбувається за допомогою ключового потоку, згенерованого системою зв’язаних хаотичних наметових відображень, параметри якої також унікальні для кожного раунду. Важливою особливістю є те, що тип операції заміни чергується між раундами: у раундах із парним індексом застосовують операцію побітового XOR кожного байта пікселя із відповідним байтом ключового потоку, тоді як у раундах з непарним індексом – операцію модульного додавання. Отже, мета дослідження полягає у розробленні ефективного методу криптографічного захисту цільових областей на зображенні після їх успішної ідентифікації за допомогою нейромережевої моделі семантичної сегментації [5, 6].

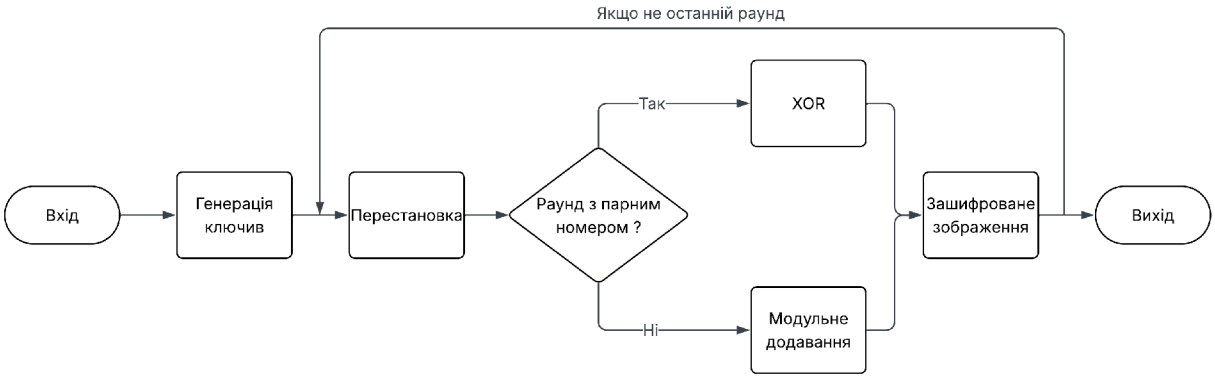

Рис. 1. Загальна схема шифрування

2. Генерація ключових матеріалів та параметри

Фундаментом розробленого методу шифрування є перетворення секретного пароля на весь набір криптографічних ключів та параметрів, необхідних для подальших операцій. Цей процес інкапсульовано у функції _derive_keys_and_params_pbkdf2, яка послідовно виконує декілька ключових кроків для генерації детермінованого, але стійкого ключового матеріалу [7, 8].

На вхід ця функція приймає пароль користувача та криптографічну сіль. Сіль являє собою випадкову послідовність байтів фіксованої довжини – 16 байтів, яка генерується під час кожного нового сеансу шифрування за допомогою генератора. Використання унікальної солі для кожної операції шифрування (навіть із однаковим паролем) гарантує генерацію унікального розширеного ключового матеріалу, що унеможливлює атаки із використанням попередньо обчислених таблиць. Це підвищує загальну безпеку системи. Згенерована сіль зберігається разом із зашифрованими даними, оскільки вона необхідна для відтворення того самого ключового матеріалу під час розшифрування [9–10].

Основним інструментом для перетворення пари пароль – сіль на бінарний ключовий матеріал слугує стандартна функція деривації ключа PBKDF2, реалізована в модулі hashlib Python [11, 12].

У програмній реалізації, всередині функції _derive_keys_and_params_pbkdf2, перед викликом hashlib.pbkdf2_hmac обчислюється загальна необхідна довжина ключового матеріалу. Вона складається із суми довжин, необхідних для параметризації всіх раундів шифрування та для ключа HMAC. Розглянемо, як здійснюється виклик функції [13, 14].

Отриманий EKM розподіляється на 32-байтовий ключ для HMAC-SHA256 та дев'ять 22-байтних блоків, кожен із яких призначений для параметризації одного раунду шифрування. Із кожного такого 22-байтного блоку послідовно вилучаються групи байтів, які за допомогою перетворень формують необхідні початкові значення $x0$, $y0$, керуючі параметри $\mu1$, $\mu2$, параметри зв'язку $\epsilon1$, $\epsilon2$, кількість ітерацій для системи зв'язаних хаотичних наметових відображень, а також цілочислове значення perm_seed для ініціалізації генератора перестановок поточного раунду. Для забезпечення стабільності хаотичного режиму згенеровані параметри $x0$, $y0$, $\mu1$, $\mu2$ додатково обмежують для перебування у валідних діапазонах [15–16].

```
total_chaos_ekm_len = chaos_ekm_len_per_round * num_rounds
total_derived_key_len = total_chaos_ekm_len + hmac_key_len

derived_material = hashlib.pbkdf2_hmac(
    'sha512',
    password.encode('utf-8'),
    salt,
    iterations,
    dklen=total_derived_key_len
)
```

Рис. 2. Код перетворення пароля і криптографічної солі на ключовий масив байтів

У результаті функція _derive_keys_and_params_pbkdf2 повертає список словників із параметрами для кожного із дев'яти раундів та ключ HMAC, забезпечуючи детерміноване розгортання початкового пароля в усі криптографічні компоненти [17, 18].

3. Підготовка зображення

Після генерації криптографічних параметрів наступним кроком є підготовка зображення до вибіркового шифрування. Спочатку вхідне зображення стандартизується, приводиться до восьмибітного формату. Потім, використовуючи маску семантичної сегментації, отриману від нейромережі, та ідентифікатор цільового класу, визначають фінальну бінарну маску, яка вказує на пікселі, що підлягають шифруванню. Ця маска формується з урахуванням вибраного режиму: або через пряме виділення пікселів цільових об'єктів, або через визначення та розширення обмежувальних прямокутників навколо кожного виявленого об'єкта [19, 20].

Для демонстрації використовуватимемо саме другий спосіб, оскільки вважаємо його ефективнішим. Тобто ми не надаватимемо потенційному порушнику силует, щоб він не зміг не тільки зрозуміти, що саме зображено в деталях, а й припустити тип об'єкта.

Нарешті, пікселі, що відповідають цій фінальній масці, вилучаються із зображення та формують окремий масив даних, який і буде безпосередньо оброблятися багатораундовим алгоритмом шифрування [21, 22].

4. Процесс шифрування, перестановка

Першою криптографічною операцією у кожному з дев'яти раундів шифрування є глобальна перестановка пікселів. Ця операція застосовується до поточного масиву даних. Мета цієї перестановки – руйнування просторових кореляцій між пікселями виділеної області та забезпечення властивості дифузії, за якої зміни в одному невеликому сегменті вхідних даних призводять до змін у багатьох частинах вихідних даних раунду. Це має істотно ускладнювати статистичний аналіз шифротексту [23, 24].

Перестановку реалізовано у функції _permute_pixels_global, яка приймає на вхід масив пікселів форми (K, 3), де K – кількість пікселів, а 3 – RGB компоненти, цілочислове початкове значення для генератора псевдовипадкових чисел та булевий прапор decrypt, що вказує напрям операції. Для кожного раунду шифрування використовують унікальне зерно, отримане з відповідного набору параметрів раунду [25, 26].

Процес перестановки відбувається так. Спочатку, на основі наданого зерна, ініціалізується генератор псевдовипадкових чисел. У нас це алгоритм PCG64 – сучасний ГПЧ, відомий хорошими статистичними властивостями та високою швидкістю. Принцип його роботи полягає у двох основних етапах: оновленні великого внутрішнього стану та застосуванні складної функції виведення до цього стану для отримання псевдовипадкового числа.

R1	G1	B1
R2	G2	B2
R3	G3	B3
…	…	…

Рис. 3. Приклад початку масиву

Внутрішній 128-бітний стан генератора PCG64 оновлюється за принципом, схожим на лінійний конгруентний генератор:

$$S_{n+1} = (S_n \times M + I) (\mathrm{mod}\, 2^{128}),$$

де S_n – поточний стан; M – константа-множник; I – константа-інкремент. Початковий стан S_0 визначається наданим зерном.

Однак ключовою особливістю PCG64, що відрізняє його від простих LCG, є те, що псевдовипадковий результат генерується не безпосередньо із цього внутрішнього стану Sn. Замість цього до стану Sn застосовується спеціальна функція виведення. Ця функція складається з ретельно підібраної послідовності побітових операцій, таких як різноманітні XOR-зсуви та бітові ротації. Основне призначення цих перетворень – істотно покращити статистичні властивості генерованих 64-бітних чисел, забезпечити їхню вищу непередбачуваність та усунути багато відомих недоліків, притаманних простішим LCG.

Важливо, що весь цей процес, починаючи з ініціалізації початкового стану S_0 за допомогою наданого зерна, повністю детермінований. Це гарантує, що для однакового зерна буде згенеровано однакову послідовність псевдовипадкових чисел, а отже, і однакову карту перемішування, що потрібно для дешифрування.

Рис. 4. Зображення із маскою від нейромережі та зображення з початковим кодуванням об'єкта

Після ініціалізації генератора псевдовипадкових чисел його використовують для створення карти перемішування. Ця карта являє собою послідовність, що містить унікальні індекси всіх пікселів, які обробляються, але розташовані у псевдовипадковій послідовності. По суті, ця послідовність визначає нове цільове місце для кожного пікселя з поточного, ще не перемішаного, масиву. Формування нового, переставленого масиву пікселів відбувається із упорядкуванням вихідних пікселів згідно із цією згенерованою картою перемішування. Тобто кожен піксель переміщується на нову позицію відповідно до цієї карти, забезпечуючи повне та, завдяки початковому значенню генератора, детерміноване перетасування всього набору оброблюваних пікселів.

5. Генерація хаотичного ключового потоку

Після того, як просторову послідовність пікселів було змінено за допомогою глобальної перестановки, наступним кроком у кожному раунді є генерація безпосереднього ключового матеріалу, який буде використано для модифікації значень колірних компонент.

Створення цього потоку покладається на набір параметрів хаотичної системи, згенерованих на етапі виведення ключів з пароля та солі. Вони є специфічними для поточного раунду: це початкові значення x_0, y_0, керуючі параметри μ_1, μ_2, параметри зв'язку $\varepsilon_1, \varepsilon_2$ та кількість початкових ітерацій.

Для генерації ключового потоку на основі цих параметрів використовується функція _generate_chaotic_keystream_py_list. Вона приймає згаданий словник round_params та необхідну довжину ключового потоку в байтах, яку розраховують відповідно до обсягу даних (кількості пікселів, помноженої на три колірні компоненти), що обробляються в цьому раунді.

```
num_pixels_in_data = processed_pixels_data.shape[0]
num_bytes_needed_for_round = num_pixels_in_data * 3
keystream_round = _generate_chaotic_keystream_py_list(round_params, num_bytes_needed_for_round)
if len(keystream_round) != num_bytes_needed_for_round:
```

Рис. 5. Код генерації ключового потоку

В основу генератора покладено систему двох зв'язаних одновимірних хаотичних наметових відображень. Використання зв'язаних систем спрямоване на отримання складнішої та менш передбачуваної динаміки. Спочатку, використовуючи початкові значення x_0, y_0 та інші параметри з round_params, система виконує skip_iterations початкових ітерацій для досягнення свого атрактора, де її поведінка стає стабільно хаотичною.

Ітераційний процес для станів системи після “прогрівання” описується рівняннями, де x_t та y_t є проміжними значеннями, отриманими із застосуванням функції наметового відображення $T(v,\mu) = \mu\min(v,1-v)$ до поточних станів x_n, y_n:

$$\begin{cases} x_t = \mu_1 * \min(x_n, 1 - x_n) \\ y_t = \mu_2 * \min(y_n, 1 - y_n) \end{cases}$$

Потім ці проміжні значення використовують для обчислення нових станів x_{n+1} та y_{n+1} з урахуванням параметрів зв’язку $\varepsilon_1, \varepsilon_2$:

$$\begin{cases} x_{n+1} = (x_t + \varepsilon_1 \times y_t) (\mathrm{mod} 1) \\ y_{n+1} = (y_t + \varepsilon_2 \times x'_{n+1}) (\mathrm{mod} 1) \end{cases}$$

де x_n, y_n – поточні значення станів на ітерації n першого та другого зв’язаних хаотичних відображень відповідно; μ_1, μ_2 – керуючі параметри для першого та другого наметових відображень відповідно; x_t та y_t – проміжні значення, отримані після застосування функції наметового відображення до поточних станів; $\varepsilon_1, \varepsilon_2$ – безрозмірні параметри зв’язку, що визначають ступінь впливу стану одного відображення на інше; x_{n+1} та y_{n+1} – значення станів двох відображень на наступній ітерації; x'_{n+1} – оновлене значення x_{n+1}, обчислене на цьому ж поточному кроці ітерації.

На кожній ітерації генерації одержані стани x_{n+1} та y_{n+1} перетворюють на цілочислові байти ключового потоку в діапазоні [0,255], виконуючи їх масштабування та беручи цілу частину. Згенеровані байти по черзі додаються до списку keystream_round, доки не буде сформовано ключовий потік необхідної довжини. Кінцева мета цього процесу – одержати ключовий потік із хорошими статистичними властивостями для подальшого використання в операціях заміни.

6. Виняткова диз’юнкція

Після того, як просторову послідовність пікселів у масиві було змінено глобальною перестановкою, а для поточного раунду було згенеровано унікальний хаотичний ключовий потік, виконується завершальна операція раунду – заміна значень пікселів. Мета цієї операції – безпосередньо модифікувати значення колірних компонент R, G, B кожного пікселя, тим самим приховуючи вихідну візуальну інформацію та забезпечуючи властивість конфузії – заплутування зв’язку між шифротекстом та ключем / відкритим текстом.

Операцію заміни застосовують побайтово до кожної з трьох колірних компонент кожного пікселя у масиві processed_pixels_data, використовуючи відповідний байт з ключового потоку. Тип операції заміни чергується залежно від індексу поточного раунду.

У раундах шифрування з парним індексом, а саме в раундах 0, 2, 4, 6 та 8, для модифікації значень пікселів використовується операція побітового виключного АБО (XOR). Ця операція фундаментальна в криптографії завдяки простоті, швидкості та властивості самооберненості.

Для кожної колірної компоненти кожного вже переставленого пікселя P_{byte} з масиву processed_pixels_data та відповідного байта K_{byte} з ключового потоку keystream_round обчислюється зашифроване значення C_{byte} за формулою:

$$C_{byte} = P_{byte} \text{ Å } K_{byte}$$

Операція виконується побітово: якщо відповідні біти P_{byte} та K_{byte} однакові, результуючий біт C_{byte} дорівнює 0; якщо біти різні – результуючий біт дорівнює 1.

Ключовою перевагою XOR є те, що для розшифрування застосовується точно така сама операція із тим самим ключовим потоком. Це забезпечує просте та якісне відновлення вихідних значень пікселів на відповідних раундах розшифрування.

Рис. 6. Використання XOR на об'єкті зображення

7. Модульне додавання

У раундах шифрування із непарним індексом для модифікації значень пікселів застосовується інший тип операції заміни – модульне додавання. Ця операція, хоча й проста, вносить в процес шифрування алгебраїчні властивості, відмінні від побітового XOR, що сприяє ускладненню загальної криптографічної стійкості із чергуванням з XOR у різних раундах.

Аналогічно до операції XOR, для кожної колірної компоненти кожного вже переставленого пікселя P_{byte} з масиву processed_pixels_data та відповідного байта K_{byte} з ключового потоку keystream_round обчислюється зашифроване значення C_{byte}. Однак, замість побітової операції, виконується арифметичне додавання за модулем 256:

$$C_{byte} = (P_{byte} + K_{byte}) (\mathrm{mod}\, 256).$$

Операція взяття за модулем 256 гарантує, що результат завжди залишатиметься в допустимому діапазоні для байта від 0 до 255. Під час програмної реалізації цієї операції важливо враховувати типи даних. Оскільки значення пікселів P_{byte} зберігаються як numpy.uint8, а елементи ключового потоку K_{byte} є стандартними цілими числами Python, щоб уникнути помилок переповнення та забезпечити коректну модульну арифметику, операнди перед додаванням явно перетворюють на стандартний цілочисловий тип int.

Для розшифрування у раундах із застосуванням модульного додавання використовується обернена операція – модульне віднімання. Щоб коректно обробити випадки, коли $P_{byte} < K_{byte}$ (що призвело б до від'ємного результату перед взяттям модуля), до різниці додають 256 перед фінальною операцією за модулем 256:

$$P_{byte} = (C_{byte} - K_{byte} + 256) (\mathrm{mod}\, 256).$$

Такий підхід гарантує відновлення вихідного значення P_{byte}. Чергування операцій XOR та модульного додавання у різних раундах шифрування – поширена практика, спрямована на ускладнення структури шифру та підвищення його стійкості до різних методів криптоаналізу, оскільки нападникові доведеться мати справу із комбінацією операцій з різними математичними властивостями.

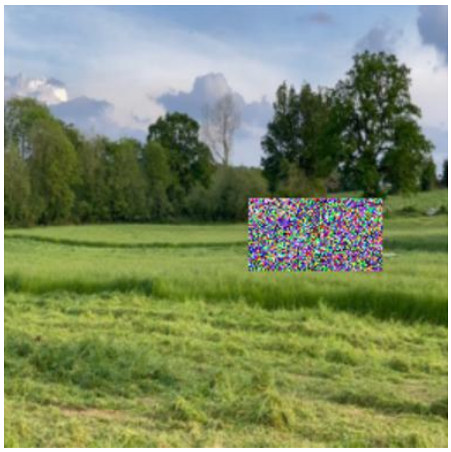

Рис. 7. Використання модульного додавання на об’єкті зображення

8. Результати і дешифрування

Після послідовного застосування усіх дев’яти раундів криптографічних перетворень виділена область переднього плану на зображенні зазнає істотних трансформацій. Вихідна візуальна інформація в цій області стає повністю нерозпізнаваною, перетворюючись на масив даних, що на вигляд нагадує випадковий шум. Приклад такого зашифрованого зображення, де чітко видно перетворену область на тлі незміненого основного зображення, що демонструє селективний характер методу, наведено на рис. 8. Проте частина зображення, не вибрана для шифрування, залишається у первинному, незміненому вигляді. Після завершення шифрування, окрім самого модифікованого зображення, система також генерує та повертає два важливі елементи даних: унікальну криптографічну сіль, використану для генерації ключів, та HMAC-тег – код автентифікації повідомлення. Ці два компоненти є частиною зашифрованих даних і необхідні для подальшого розшифрування інформації.

Рис. 8. Результат кодування

Процес розшифрування є симетричним до шифрування. Для нього використовують наданий пароль, збережену під час шифрування сіль та відповідний HMAC-тег. Спочатку система перевіряє HMAC-тег для гарантії цілісності та автентичності отриманих даних; у разі успіху застосовуються дев'ять обернених криптографічних раундів із використанням тих самих, детерміновано відновлених параметрів, для повного відновлення вихідних пікселів у зашифрованій області.

Рис. 9. Дешифроване зображення

Висновки

Результатом роботи стало створення нового дев'ятираундового симетричного алгоритму шифрування для виділених ROI. Цей алгоритм перетворює пароль, який надав користувач, та унікальну випадкову сіль на весь необхідний ключовий матеріал за допомогою стандартної функції посилення ключа. Кожен із дев'яти раундів шифрування містить дві основні операції: глобальну перестановку всіх пікселів в ROI та побайтову заміну їх значень. Для операції заміни використовують ключові потоки, які генерує система зв'язаних хаотичних відображень, налаштування якої є індивідуальними для кожного раунду. Типи операції заміни чергуються між раундами. Для перевірки цілісності та автентичності зашифрованих даних використовують стандартний механізм коду автентифікації повідомлень.

Експериментальне дослідження властивостей розробленого шифру показало його працездатність. Статистичний аналіз підтвердив, що після шифрування істотно знижується кореляція сусідніх пікселів та вирівнюються гістограми колірних каналів, що свідчить про руйнування вихідної структури зображення. Алгоритм продемонстрував високу чутливість до змін у секретному ключі: навіть незначна модифікація пароля призводить до кардинально іншого шифротексту, а розшифрування неправильним ключем блокується завдяки перевірці автентичності. Разом із тим, аналіз чутливості до змін у самому зображенні виявив обмежену здатність поточної структури раундів поширювати локальні зміни відкритого тексту на весь шифротекст.

Загалом, розроблений програмний інструмент підтвердив ефективність запропонованої концепції селективного захисту візуальної інформації.

Подальші дослідження можуть бути спрямовані на вдосконалення точності нейромережевої сегментації та модифікацію й ускладнення алгоритму шифрування для покращення його дифузійних властивостей відносно змін у відкритому тексті.

Список використаних літературних джерел

[1] Малінін І. Г. Розробка моделі селективного кодування для захисту відеоінформаційних ресурсів в інфокомунікаційних системах: дипломна робота на здобуття кваліфікаційного рівня "Бакалавр" за спеціальністю 122 "Комп'ютерні науки". https://drive.google.com/file/d/17wEt376qDb4GZz-yGBHSiCglAMhSFRoA/view

[2] Ільяшов О. А., Бурячок В. Л. До питання захисту інформаційно-телекомунікаційної сфери від стороннього кібернетичного впливу. Наука и оборона. 2010. № 4. С. 35–41.

[3] Barannik V., Stepanko O., Nikodem J., Jancarczyk D., Babenko Yu., Zawislak S. A Model for Representing Significant Segments of a Video Image Based on Locally Positional Coding on a Structural Basis. Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IEEE IDAACS-SWS 2020): proceedings of IEEE 5nd International Symposium, 2020. P. 1–5. DOI: 10.1109/IDAACS-SWS50031.2020.9297068.

[4] Теоретичні основи створення технологій протидії прихованим інформаційним атакам в сучасній гібридній війні / А. М. Алімпієв, В. В. Бараннік, Т. В. Белікова, С. О. Сідченко // Системи обробки інформації. Харків: ХНУПС, 2017. Вип. 4(150). С. 113–121.

[5] V. Barannik, N. Kharchenko, O. Kulitsa, V. Tverdokhleb, "The issue of timely delivery of video traffic with controlled loss of quality", in 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2016, pp. 902–904. DOI: 10.1109/TCSET.2016.7452220.

[6] Баранник В. В., Шульгін С. С., Онищенко Р. С., Реввa К. В., Яковенко О.В. Метод відновлення усічено-позиційних чисел в нерівномірно-діагональному спектральному просторі. Сучасна спеціальна техніка. 2023. № 3. С. 33–42.

[7] Shamir A., Rivest R. L., Adleman L. M. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. 1978. Vol. 21. Iss. 2. P. 120–126. DOI: 10.1145/359340.359342.

[8] Баранник В. В., Власов А. В., Сидченко С. А. Обоснование значимых угроз безопасности видеоинформационного ресурса систем видеоконференцсвязи профильных систем управления. Информационно-управляющие системы на ЖД транспорте. 2014. № 3. С. 24–31.

[9] Digital Transformation Age. TCSET 2022. Lecture Notes in Electrical Engineering, Vol. 965. Springer, Switzerland, Cham. https://doi.org/10.1007/978-3-031-24963-1_25.

[10] Barannik, V., Barannik, N., Sidchenko, S., Khimenko, A. The method of masking overhead compaction in video compression systems, Radioelectronic and Computer Systems, No. 2, pp. 51–63, 2021. DOI: https://doi.org/10.32620/reks.2021.2.05.

[11] Usman A. M. et al. A Comparative Analysis on Blockchain versus Centralized Authentication Architectures for IoT-Enabled Smart Devices in Smart Cities: A Comprehensive Review, Recent Advances, and Future Research Directions. MDPI Open Access Journals. URL: https://www.mdpi.com/1424-8220/22/14/5168.

[12] Wu Y. NPCR and UACI Randomness Tests for Image Encryption April 2011 Authors: Yue Wu Raytheon BBN Technologies. ResearchGate. URL: https://www.researchgate.net/publication/259190481_NPCR_and_UACI_Randomness_Tests_for_Image_Encryption.

[13] Xiaowu Li, Huiling Peng. Chaotic medical image encryption method using attention mechanism fusion ResNet model. PubMed Central. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC10373303/.

[14] Yuanlin Chen et al. A novel image encryption method based on improved two-dimensional logistic mapping and DNA computing. Frontiers. URL: https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2024.1469418/full.

[15] JPEG image scrambling without expansion in bitstream size [Text] / K. Minemura, Z. Moayed, K. Wong, X. Qi, K. Tanaka // Image Processing : proc. 19 th IEEE Int. Conf., 30 Sept.-3 Oct. 2012. Orlando, FL, USA, 2012, pp. 261–264. DOI: 10.1109/ICIP.2012.6466845.

[16] Бараннік, В. В., Шульгін, С. С., Бабенко, Ю. М., Онищенко, Р. С., Реввa, К. В., Белікова, Т. В. and Ігнатьєв, О. О. (2023). "Technology of Sliding Coding of Uneven Diagonal Sequences in Two-Dimensional Spectral Space of Transformants", Visnyk NTUU KP. Seriia – Radiotekhnika Radioaparatobuduvannia, (94), pp. 13–23. DOI: 10.20535/RADAP.2023.94.13-23.

[17] Deshmukh, M., Nain N., Ahmed M. An (n, n)-Multi Secret Image Sharing Scheme Using Boolean XOR and Modular Arithmetic [Text]. Advanced Information Networking and Applications : proc. IEEE 30 th Int. Conf. (AINA), 23–25 March 2016. Crans-Montana, Switzerland, 2016, pp. 690–697. DOI: 10.1109/aina.2016.56.

[18] Бараннік В. В., Шульгін С. С., Семенченко О. А., Пчельніков С. І., Шейгас О. К. Обгрунтування підходу для побудови технологій кодування відеоінформації. Сучасна спеціальна техніка. 2023. № 2. С. 14–23.

[19] Barannik V., Barannik D., Barannik N. Indirect Steganographic Embedding Method Based On Modifications of The Basis of the Polyadic System. Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET'2020): proceedings of 15 th IEEE International Conference, 2020, pp. 699–702. DOI: 10.1109/TCSET49122.2020.235522.

[20] Бараннік Володимир, Шульгін Сергій, Бараннік Дмитро, Оніщенко Роман, Реввa К. Метод усічено-позиційного декодування трансформант за нерівномірно-діагональним форматом. Наукоємні технології. 2023. № 3. С. 280–288.

[21] Wong K. W. Image encryption using chaotic maps. Intelligent Computing Based on Chaos. 2009, Vol. 184, pp. 333–354. DOI: 10.1007/978-3-540-95972-4_16.

[22] Бараннік В., Шульгін С., Ігнатьєв О., Онищенко Р., Бабенко Ю., Бараннік В. Концепція функціональних перетворень для формування синтаксичного опису діагоналей трансформанти. Інфокомунікаційні технології та електронна інженерія. Вип. 3, № 1. С. 24–34 (2023).
[23] Wong K., Qi X., Minemura K., Moayed Z., Tanaka K. JPEG image scrambling without expansion in bitstream size. Image Processing: proceedings of the 19 th IEEE International Conference, 2012, pp. 261–264. DOI: 10.1109/ICIP.2012.6466845.
[24] Бараннік Володимир, Шульгін Сергій, Оніщенко Роман, Ревва К., Ігнатьєв Олександр. Метод формування інформативно-позиційної ваги для усічено-позиційної кодової системи представлення трансформованих відеосегментів. Наукоємні технології. 2023. № 2. С. 156–163.
[25] Barannik V. V., Karpenko S. Method of the 3-D image processing. Modern Problems of Radio Engineering, Telecommunications and Computer Science (IEEE TCSET 2008): proceedings of IEEE International Conference, 2008, pp. 378–380.
[26] Barannik, V. et al. (2023). A Method of Scrambling for the System of Cryptocompression of Codograms Service Components. In: Klymash, M., Luntovskyy, A., Beshley, M., Melnyk, I., Schill, A. (eds) Emerging Networking in the Digital Transformation Age. TCSET 2022. Lecture Notes in Electrical Engineering, Vol. 965. Springer, Switzerland, Cham. https://doi.org/10.1007/978-3-031-24963-1_26
[27] Бараннік В.В., Шульгін С.С., Онищенко Р.С., Ігнтатьєв О.О. Методологія кодування трансформованих відеосегментів в усічено-позиційному просторі // Вчені записки Таврійського національного університету імені в.і. вернадського, Серія: Технічні науки : Том 34 (73) № 1, 2023, С 38 – 42.

SELECTIVE ENCRYPTION OF VIDEO INFORMATION BASED ON SEMANTIC SEGMENTATION USING U-NET NEURAL NETWORK

Ihor Malinin[1], Yulia Koliadenko[1], Bohdan Zhurakovskyi[2], Oleksandr Kovalenko[1], Valeriy Ralko[3]

[1] V. N. Karazin Kharkiv National University, 4, Svobody Square, Kharkiv, 61022, Ukraine
[2] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Beresteyskyi Avenue, Kyiv, 03056, Ukraine
[3] Ivan Kozhedub Kharkiv National Air Force University, 12, Poltavskyi Shlyakh str., Kharkiv, 61123,Ukraine

In the digital age, video information has taken a leading place among data types in terms of volume and significance. Large amounts of visual data are created daily using video cameras, mobile gadgets, drones and network services, and a significant part of this content may contain personal or confidential information. Although traditional full encryption of the video stream guarantees a high level of protection, it is accompanied by a number of disadvantages: high load on computing resources, delays during data transmission and difficulties in implementing real-time processing. The paper solves the scientific and applied problem of selective cryptographic protection of video information based on automatic object identification using the U-Net neural network. A multi-round encryption method is proposed, which implements global permutation of pixels and byte-by-byte replacement of their values in the selected image area. The purpose of the research is to develop an effective method for cryptographic protection of target image areas after their identification, with an emphasis on diffusion, confusion, and protection from statistical analysis. A system of coupled chaotic tent mappings is used to generate the key material, and the PBKDF2 mechanism with a cryptographic salt is used for parameterization. The paper also considers the use of XOR and modular addition operations, which alternate between rounds, which increases the cipher's resistance to attacks. An experimental analysis of the results is carried out, which confirms the effectiveness and resistance of the developed method to changes in the input data and key information. The result of the work was the development of a new 9-round symmetric encryption algorithm for selected ROIs. This algorithm converts the user-supplied password and a unique random salt into all the necessary key material using a standard key strengthening function. Each of the nine encryption rounds includes two main operations: a global permutation of all pixels in the ROI and a byte-by-byte replacement of their values. The replacement operation uses key streams generated by a system of coupled chaotic mappings, the settings of which are individual for each round. The type of replacement operation alternates between rounds. To verify the integrity and authenticity of the encrypted data, a standard message authentication code mechanism is used.

Keywords: *selective encryption, U-Net neural network, semantic segmentation, cryptographic stability, chaotic mapping, video security, PBKDF2, diffusion, confusion, ROI.*