УДК 004.932 : 004.056.53

Є.С. СУЛЕМА, С.С. ШИРОЧИН
Національний технічний університет України «Київський політехнічний інститут»

МЕТОД ЗАХИСТУ ЗОБРАЖЕНЬ НА ОСНОВІ ШИФРУВАННЯ ПАЛІТРИ

Анотація. *Необхідність передачі цифрових зображень через Інтернет або зберігання на загальнодоступних серверах спричиняє їх потрапляння у відкритий доступ. У випадку, якщо зображення мають характер особистої інформації, постає задача забезпечення їх захисту. Існуючі методи криптографічного захисту передбачають особливі процедури використання та обробки ключів. Окрім цього, ключ має певну відому структуру і його передача може бути виявленою. В статті пропонується метод шифрування палітри зображень, який ґрунтується на використанні в ролі ключа іншого зображення. Розроблений метод забезпечує можливість передачі ключа відкритими каналами зв'язку й при цьому дозволяє забезпечити надійний захист графічних даних користувача.*

Ключові слова: *шифрування, захист графічних даних, палітра.*

YEVGENIYA SULEMA, SEMEN SHYROCHYN
National Technical University of Ukraine "Kyiv Polytechnic Institute"

IMAGE PROTECTION METHOD BASED ON PALETTE ENCRYPTION

Abstract. *The necessity in either transferring digital images by the Internet or keeping them in public servers can lead to their leak into open access. In case if a digital image represents user's personality, a task of its protection becomes topical. The existing methods of cryptographic protection demand special handling of keys. In addition keys have a well-known structure, thus the key transfer can be detected. The image protection method based on palette encryption that uses another image as a key is proposed in this paper. The developed method enables the possibility of unsuspicious transfer of a key through open channels and at the same time it allows reliable protection of user's graphical data.*

The basic idea of the proposed method lies in encoding of graphical data (color hue codes) of the image to be protected by using another image as a palette (i.e. a key). To be accepted as a key the image has to include all color hues present in the image to be protected. If the image that is desirable for using as a key doesn't include all the necessary hues, it can be modified. Since the proposed method is oriented for using by ordinary users the execution time of its algorithm becomes one of crucial characteristics. To accelerate the algorithm execution parallel data processing is used. The results of parallel algorithm effectiveness analysis are presented in this paper. The next issue discussed in the paper is the compression of the encoded data. The results of compression effectiveness analysis are presented as well. The modification of the proposed method that consists in encoding color transitions between neighboring pixels in the image to be protected instead of encoding of color of separate pixels is proposed and discussed in this paper, too. The practical recommendations for the use of both the proposed method and its modification are presented.

Keywords: *encryption, graphical data protection, palette.*

Вступ

Значна кількість цифрових зображень, що передаються через Інтернет, має характер особистої інформації, що не підлягає поширенню без згоди власника. Водночас користувачі, які передають по мережі або зберігають на серверах загального користування свої фотографії, в більшості не є професіоналами у галузі захисту інформації. Тому використання складних алгоритмів криптографічного або стеганографічного захисту є для них не завжди прийнятним. Зокрема, для кінцевого користувача, що використовує середньостатистичний персональний комп'ютер, важливим є час кодування та декодування даних. Крім того, зображення є особливим класом комп'ютерних даних, які характеризуються певними особливостями, такими, як природна надлишковість та великий обсяг даних. Врахування цих особливостей може допомогти розробити більш надійні, але водночас прості у використанні методи, призначені саме для захисту цифрових зображень. Крім того, перспективною для захисту зображень є ідея поєднання криптографічного захисту [1] графічних даних з стеганографічним принципом приховування [2], який можна застосувати до ключа, що також може бути зображенням. Таким чином, розроблення нових методів, способів та алгоритмів захисту цифрових зображень є актуальною задачею, важливою для повсякденного захисту особистих графічних даних широкого кола кінцевих користувачів.

Мета дослідження

Метою дослідження, представленого у даній статті, є розроблення криптографічного методу захисту графічних даних користувача, що задовольняє такі умови:

- унеможливлення несанкціонованого доступу до графічних даних;
- забезпечення непомітності передачі ключа;
- висока швидкодія алгоритму кодування та декодування графічних даних.

Слід зазначити, що оскільки для звичайного користувача швидкодія кодування та декодування даних є важливою характеристикою, то у даному дослідженні цьому питанню приділялась особлива увага.

Опис методу

Метод, що пропонується, ґрунтується на шифруванні кодів відтінків кольорів пікселів зображення, шляхом їх заміни координатами пікселів відповідних відтінків кольорів у зображенні-ключі, що використовується як *палітра*, в якій представлені всі відтінки кольорів, наявні у зображені, що шифрується. В ролі ключа виступає звичайне зображення, яке в процесі шифрування не зазнає жодних змін. Через те, що зображення-ключ не має специфічної структури ключа і не містить вбудованих повідомлень, воно може

бути передане електронною поштою чи будь-яким іншим способом або може бути збережене на сервері загального користування разом з іншими зображеннями, а посилання на нього може бути надіслане або повідомлене адресату будь-яким зручним способом. Оскільки для шифрування і дешифрування використовується один і той самий ключ, криптосистема є симетричною, а ключ є відкритим.

Вимогою до зображення для використання в ролі ключа є наявність в його графічних даних повного діапазону всіх можливих значень байт від 0 до 255. Назвемо даний критерій ***повнотою наявних байт***. Для виявлення частки зображень, що відповідають даній умові, було проведене статистичне дослідження. Дослідження проводилось на базі декількох колекцій різноманітних зображень у кількості від 300 до 3000 зображень в колекції, в тому числі необроблених фотографій. Згідно з результатами проведеного статистичного дослідження, критерію повноти наявних байт відповідає більш, ніж 90% зображень. До решти відносяться надто світлі або надто темні фотографії, а також штучно створені зображення з фіксованим неповним діапазоном значень кодів відтінків кольорів. Критерій повноти наявних байт вносить певні обмеження щодо мінімального розміру зображення. Для зображень у відтінках сірого кольору (1 байт на піксель) мінімальним розміром є 256 пікселів (зображення 16 на 16 пікселів). Для кольорових (3 байти на піксель) мінімальним розміром є 85 пікселів (5 на 17 пікселів). Оскільки абсолютна більшість зображень має розміри, що значно перевищують мінімально необхідні, можна вважати критерій повноти наявних байт таким, що може бути задовільнений у більшості випадків. Формат зображення також не є принциповим, єдина умова – він має бути растровим. Якщо зображення є векторним, його потрібно перетворити у растровий формат та повторно перевірити на критерій повноти наявних байт.

Після перевірки на критерій повноти байт відбувається процес створення словника координат байт, що мають відповідне значення *(рис. 1)*. У випадку, якщо в словнику не залишається порожніх комірок, зображення може бути використано як ключ.

В результаті формується масив списків:

$$C_0 = \{C_{0,0}, C_{0,1}, \dots C_{0,m}\}$$
$$C_1 = \{C_{1,0}, C_{1,1}, \dots C_{1,m}\} \quad (1)$$
$$C_{255} = \{C_{255,0}, C_{255,1}, \dots C_{255,k}\},$$

де C_i – список координат пікселів зображення-ключа зі значенням коду відтінку кольору *i*,

n, m, k – кількість пікселів зображення-ключа, що мають відповідне значення коду.

Мінімально достатнім для кодування є випадок, коли кожен рядок містить по одному значенню. Проте, це буде означати однозначну відповідність при заміні кожного значення інтенсивності кольору зображення, що шифрується, певним значенням пари координат палітри, що суттєво знижує ступінь захисту, оскільки в цьому випадку стає можливим підбір значень.

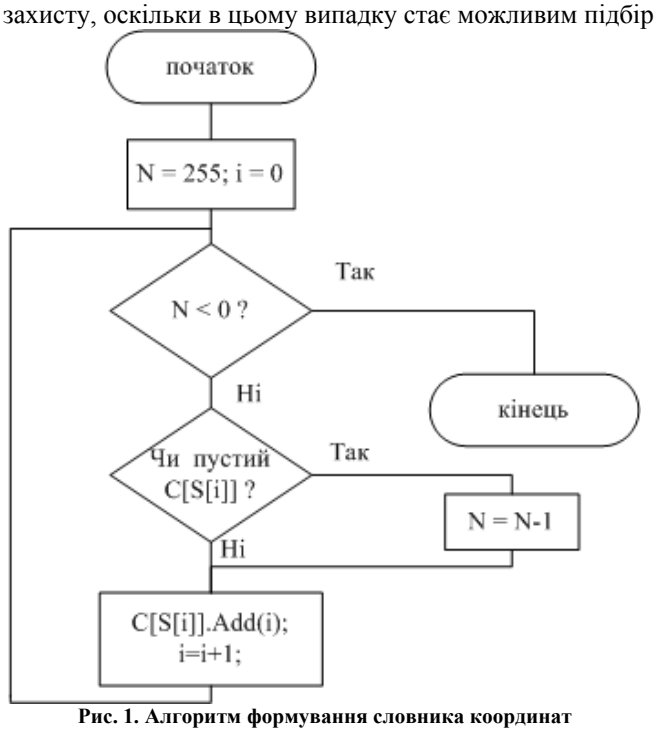

Рис. 1. Алгоритм формування словника координат

Отже, важливо, щоб для кожного з 256 значень кодів відтінків кольорів у палітрі зображення-ключа був набір з декількох варіантів значень пари координат пікселя відповідного відтінку, які при шифруванні можна обирати випадковим чином. Водночас, якщо у зображені-ключі є декілька пікселів кожного з відтінків кольорів, повний аналіз палітри вимагає тривалого часу і негативно впливає на швидкодію алгоритму реалізації методу. Тому пропонується припиняти формування словника, коли у ньому немає жодного порожнього рядка. На момент виконання даної умови всі інші рядки будуть у більшості випадків містити декілька варіантів значень пари координат. Таким чином, підбір значень у разі спроби несанкціонованого розшифрування зображення буде значно утруднений, а часових втрат на повне розкладання палітри вдасться уникнути.

Всього може бути утворено один або два таких словники: один у випадку, якщо записується абсолютний номер байта, два – якщо окремо записується кожна з координат пікселя.

Під час шифрування замість кожного байту графічних даних зображення, що підлягає захисту, записується значення координат деякого псевдовипадкового пікселя того ж відтінку кольору у зображені-ключі:

$$X_i = C_{S_i,R}, \qquad R \in [1, n] \quad (2)$$

де X_i – *i* -й байт даних зашифрованого зображення;

C_{S_i} – список координат пікселів зображення-ключа зі значенням коду відтінку кольору S_i ;

S_i – i -й байт зображення, що зашифровується;

R – випадкове число у діапазоні від 1 до n ;

n – кількість елементів у рядку C_{s_i}.

Оскільки у більшості рядків словника є більше, ніж один варіант значень координат, в елемент X_i результуючого зашифрованого зображення може бути записаний будь-який елемент списку C_{s_i}, номер якого визначається випадковим числом R.

Отриманий масив даних X зашифрованого зображення ущільнюється і зберігається в одному файлі (якщо зберігаються абсолютні номери байт) або у двох окремих файлах (якщо кожна координата зберігається окремо). Додаткові дані щодо розмірів зашифрованого зображення зберігаються у заголовку файлу.

Відновлення даних

Для відновлення даних із зашифрованого зображення необхідно мати зображення-ключ. На відміну від процесу шифрування, при розшифруванні не відбувається розкладання палітри зображення-ключа. В процесі відновлення даних відбувається зчитування графічних даних зображення-ключа K , а також масиву значень зашифрованого зображення X . Для випадку, коли кодується номер байту зображення, формула відновлення має такий вигляд:

$$S_i = K_{X_i}, \tag{3}$$

де S_i – i -й байт зображення, що розшифровується;

K_{X_i} – X_i -й елемент масиву K ;

X_i – i -й байт файлу зашифрованих даних.

Для випадку, коли кодуються координати пікселя зображення і утворюється два файли з зашифрованими даними, формула відновлення є наступною:

$$S_i = K_{H_i \times stride + W_i}, \tag{4}$$

де S_i – i -й байт зображення, що розшифровується;

K – масив байтів зображення-ключа;

H_i – i -й байт зашифрованого масиву координат висоти H;

W_i – i -й байт зашифрованого масиву координат ширини W;

$Stride$ – кількість байт в рядку зашифрованого зображення, записується в один із масивів.

В результаті розшифрування отримується масив графічних даних вихідного зображення S, що може бути збережене у будь-якому форматі.

Паралельна реалізація методу

Для прискорення роботи алгоритму було застосоване розпаралелювання між ядрами багатоядерного процесора. Розпаралелювання обчислень виконується при формуванні палітри та при шифруванні даних.

Оскільки розпаралелювання орієнтоване на персональний комп'ютер з багатоядерним процесором, кількість ядер якого обмежена, найбільш раціональним шляхом для розпаралелювання вкладених циклів буде розпаралелювання тільки зовнішнього або тільки внутрішнього циклу. Таким чином, всі процеси всередині будуть незалежними, відповідно використовуючи незалежні змінні. Це потребує більших ресурсів для забезпечення незалежності процесів (окремі змінні в кожній паралельній ітерації), проте навіть при двох ядрах це дає перевагу в часі роботи. Повне розпаралелювання всіх вкладених циклів є недоцільним, оскільки потребує значного збільшення необхідної кількості ресурсів.

Оскільки алгоритмічна реалізація запропонованого методу з використанням розпаралелювання вимагає використання більшої кількості ресурсів, ніж без розпаралелювання, то виконання паралельного алгоритму цього методу на одноядерному процесорі, є недоцільним, оскільки час роботи буде більший, ніж у алгоритму без розпаралелювання. Тому у випадку одного ядра використовується послідовна обробка даних, а починаючи з двох ядер – розпаралелена.

Для оцінки ефективності паралельної реалізації методу були проведені виміри часу шифрування п'яти різних зображень. Експерименти проводилися на двоядерному процесорі Intel Centrino. Кожен експеримент був повторений 1000 разів з різними зображеннями-ключами в паралельному і послідовному варіантах. За результат було взято середній час за 1000 вимірів (табл. 1).

Таблиця 1

Час шифрування при паралельній і послідовній реалізаціях алгоритму

Об'єм файлу зображення	Розміри зображення	Середній час шифрування (послідовна реалізація), мс	Середній час шифрування (паралельна реалізація), мс	Економія часу (%)
59 815	320 × 213	11,1	10,5	5,7
376 459	1024 × 765	110,5	103,1	7,1
109 699	800 × 1204	147,5	139,8	5,5
608 290	1324 × 2048	432,3	385,8	12,0
6 529 035	3888 × 2592	1600,9	1451,9	10,2

Як видно з табл. 1, паралельна реалізація дає економію часу від 5,5% до 12%, що є меншим, ніж очікуваний приріст за рахунок розпаралелювання обчислень на два ядра. Цей результат пояснюється тим, що паралельний алгоритм порівнювався з більш оптимальним алгоритмом послідовного обчислення.

Подальше удосконалення методу

Подальшим розвитком ідеї, що лежить в основі методу шифрування палітри, є кодування адреси не одного байта графічних даних, а послідовності байт. Метою цієї модифікації є зменшення розміру вихідного файлу зашифрованих даних. Таким чином, одна адреса буде кодувати два послідовних значення (***переходи***) кодів відтінків кольорів сусідніх пікселів зображення, що шифрується. Тому для кодування переходів вимогою до зображення є наявність повного набору всіх можливих послідовних пар байт від {0;0} до {255;255}. Назвемо даний критерій ***повнотою наявних переходів***. Цей критерій вимагає від зображення наявності всіх 65536 можливих переходів. Згідно зі статистичним дослідженням, що було проведене з використанням 5031 зображення, даному критерію відповідає 0,08% зображень, що з одного боку означає складність підбору зображень-кандидатів на роль ключа, а з іншого боку дозволяє стверджувати, що лише 0,08% зображень вимагають кодування всіх 65536 можливих переходів. Окрім того, ступінь невідповідності критерію повноти наявних переходів неоднаковий, до того ж у багатьох випадках можливою є попередня обробка графічних даних зображення-кандидата, в результаті якої модифіковане зображення буде відповідати даному критерію.

Введемо коефіцієнт відповідності *R*, що дорівнює відношенню існуючої кількості переходів до максимальної (65536). Серед проаналізованих груп зображень було обчислено (табл. 2), що середнє значення коефіцієнта *R* коливається від 0,62 до 0,84, а максимальне – перевищує 0,91. Отже, існує можливість модифікації зображення-кандидата, що полягає в додаванні в його графічні дані відсутніх там переходів. За поріг доцільності модифікації оберемо R_{min} = 0,95 (3277 відсутніх переходів).

Таблиця 2

Коефіцієнт R у групах зображень

Кількість зображень в групі	Мінімальне значення *R*	Середнє значення *R*	Максимальне значення *R*
110	0,3473663	0,8390575	1,0
154	0,08535767	0,6329114	0,9173889
176	0,2477417	0,7174703	0,9989929
211	0,1904449	0,6593649	0,928299
224	0,202774	0,7372525	0,9790344
238	0,2371979	0,6474388	0,9645538
303	0,2848969	0,7867157	0,9998169
711	0,2169647	0,7009154	0,9410858
962	0,07608032	0,6208385	0,9925537
1942	0,1728363	0,643666	1,0

Процес модифікації зображення-кандидата відбувається наступним чином. При знаходженні в матриці *C* порожнього списку $C_{i,j}$, відбувається пошук непорожніх списків для того, щоб відібрати з них один з переходів. Оскільки заміна одного числового значення *k* на *j,* впливає одразу на чотири списки (зменшує кількість переходів в *k* та з *k*, а збільшує кількість переходів з *j* та в *j*), то це також потрібно враховувати. Отже, при пошуку необхідно виконати наступні умови:

1. Пошук потрібно проводити лише серед комірок з тим самим значенням *i* або *j*, щоб відрізнялось лише одне значення координат.

2. В непорожньому списку $C_{i,k}$ має бути не менше 2 елементів, щоб після вилучення одного з них список не став порожнім.

3. Список $C_{i,c_{i,k+1}}$, на який вплине заміна *k* на *j* в $C_{i,k}$, також мусить містити не менше 2 елементів.

Для того, щоб уникнути появи візуально помітних відмінностей у модифікованому зображені-ключі, слід спочатку вести пошук в найближчих сусідніх комірках $C_{i-1,j}$, $C_{i+1,j}$., $C_{i,j-1}$., $C_{i,j+1}$. Далі відстань потрібно збільшувати, аж поки не буде знайдена комірка, що задовольняє всім вищезазначеним умовам.

Дана модифікація методу спричиняє низку відмінностей в алгоритмі побудови списків координат. Замість вектору списків формується матриця списків:

$$\begin{gathered} C_{0,0} = \left\{ C_{0,0,0}, C_{0,0,1}, \ldots C_{0,0,n} \right\}. \\ C_{0,1} = \left\{ C_{0,1,0}, C_{0,1,1}, \ldots C_{0,1,m} \right\}. \\ \ldots \\ C_{255,255} = \left\{ C_{255,255,0}, C_{255,255,1}, \ldots C_{255,255,k} \right\}, \end{gathered} \tag{5}$$

де $C_{i,j}$ – матриця координат пікселів зображення-ключа зі значенням коду відтінку кольору *i,j*.

n, m, k – кількість пікселів зображення-ключа, що мають відповідне значення коду.

При кодуванні переходів буде виконуватись формула (2), проте відрізнятися буде крок ітерації – за одну ітерацію буде кодуватись K_{X_i}, що буде дорівнювати S_i, при умові, що $K_{X_i+1} = S_{i+1}$. Так само буде відрізнятись відновлення даних, що буде повторювати формулу (3), за одну ітерацію буде відновлюватись пара значень $\{S_i, S_{i+1}\}$, що матимуть значення $\{K_{X_i}, K_{X_i+1}\}$.

Ущільнення даних

Отримані зашифровані дані являють собою масив адрес, де на кожний байт графічних даних зображення, що шифрується, припадає 4 байти адреси. Це викликає збільшення розміру вихідного файлу відносно початкового, але через те, що не всі адреси потребують всі 4 байти для кодування, цей масив даних добре ущільнюється. У модифікованому методі 2 байти початкового зображення кодуються однією адресою, що дозволяє зменшити об'єм вихідних даних в 2 рази. В даному досліджені було проведено аналіз можливостей ущільнення вихідного файлу різними алгоритмами. Отримані результати наведені у табл. 3а і табл. 3б.

Таблиця 3а

Ущільнення даних, зашифрованих звичайним методом шифрування палітри

Оригінал (jpg)	Оригінал (bmp)	Без ущільнення	rar	zip	7z	bz2
59 815	204 520	817 960	162 020	177 759	**141 701**	154 982
376 459	2 104 360	8 417 320	1 674 001	2 137 002	**1 516 381**	1 633 607
109 699	2 889 640	11 558 440	1 084 860	1 041 563	**692 926**	722 625
608 290	8 134 696	32 538 664	2 449 753	3 305 038	2 413 357	**2 114 755**
6 529 035	30 233 128	120 932 392	27 735 188	36 059 621	**22 156 015**	36 059 621

Таблиця 3б

Ущільнення даних, зашифрованих модифікованим методом шифрування палітри

Оригінал (jpg)	Оригінал (bmp)	Без ущільнення	rar	zip	7z	bz2
59 815	204 520	409 000	354 243	354 423	337 171	355 669
376 459	2 104 360	4 208 680	3 375 277	3 698 985	3 495 553	3 654 519
109 699	2 889 640	5 779 240	4 635 074	4 606 191	3 697 759	4 141 505
608 290	8 134 696	16 269 352	13 849 555	14 143 241	12 431 029	13 607 096
6 529 035	30 233 128	60 466 216	51 979 235	53 240 161	48 033 486	52 596 449

У більшості випадків найкращим для ущільнення є алгоритм 7z. Проте його ефективність відрізняється для звичайного і модифікованого методів. Модифікований метод дозволяє отримати менший об'єм вихідних даних, проте ці дані набагато гірше ущільнюються [3]. Це пов'язано з тим, що у модифікованому методі кодуються не окремі байти, а їх послідовності, отже в результаті отримуємо набагато менше повторень, які добре ущільнюються. Співвідношення розмірів вхідних і вихідних файлів наведене на рис. 2.

Для зручності за одиницю було прийнято розмір файлу оригіналу в форматі BMP. Як видно з результатів (рис. 2), метод шифрування палітри збільшує об'єм вихідних даних рівно у 4 рази, а модифікований – у 2 рази. Проте після ущільнення при звичайному методі результуючий файл завжди має коефіцієнт співвідношення менший за оригінал (від 0,24 до 0,73), а після ущільнення при модифікованому методі він завжди більший (від 1,28 до1,65).

Таким чином, відмінності у результатах роботи двох версій методу шифрування палітри визначають можливі варіанти їх застосування: якщо не виконувати ущільнення, то варто використовувати модифікований метод; в протилежному випадку слід надати перевагу звичайному методу шифрування палітри, але при цьому потрібно виконувати ущільнення отриманих зашифрованих даних.

Рекомендації щодо реалізації методу

При реалізації [4] даного методу було виявлено декілька особливостей, з яких можна вивести низку рекомендацій щодо його реалізації та використання. По-перше, необхідно визначитись з пріоритетами щодо швидкодії обробки даних та розміру вихідного файлу, що залежить в тому числі і від наявних каналів передачі даних. Якщо канал дозволяє швидко передавати великі об'єми даних, то більший пріоритет буде у швидкодії. Якщо канал є повільним, то розмір вихідного файлу має більший пріоритет, ніж швидкодія алгоритму реалізації методу. Окремим питанням є ущільнення даних. Якщо розмір файлу має вирішальне значення і передбачається подальше ущільнення, то варто використовувати звичайний метод з подальшим ущільненням. Якщо ущільнення не передбачається, а розмір має бути мінімальним, варто використовувати модифікований метод.

Значний час витрачається на побудову списку адрес, тому це варто робити одноразово одразу після завантаження ключа, а не кожного разу при шифруванні. У випадку модифікованого методу список адрес формується ще довше, окрім того може знадобитись модифікація ключа. Тому для модифікованого методу побудова списку адрес і модифікація ключа (в разі необхідності) мають відбуватись одноразово при

першому завантаженні ключа.

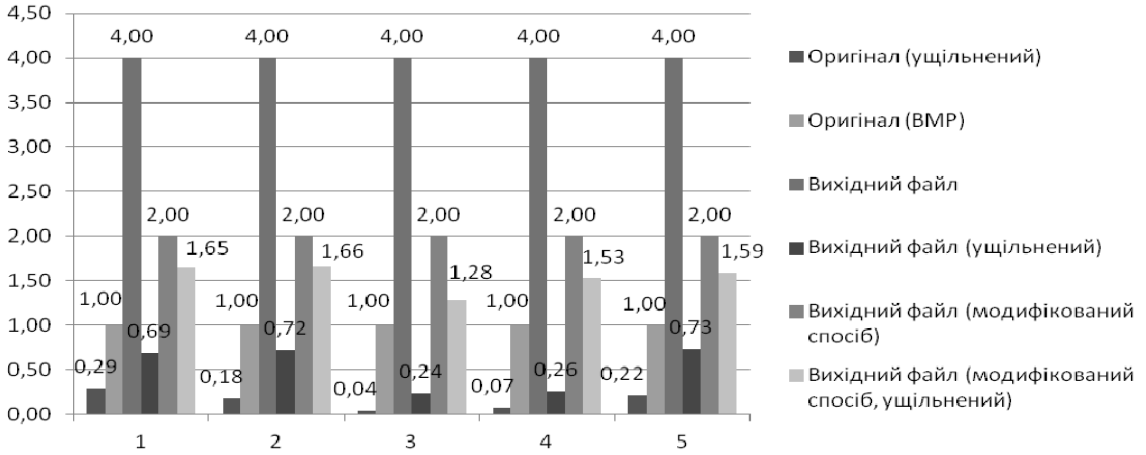

Рис. 2. Співвідношення розмірів файлів відносно оригіналу

Хоча запропонований метод не передбачає обмежень на максимальний розмір ключа, для зручності передачі рекомендується використовувати ключ невеликого розміру. Проте, це актуально лише для звичайного методу, оскільки модифікований метод вимагає великих ключів через те, що у невеликих зображеннях не знайдеться всіх необхідних переходів.

Висновки

Запропонований метод шифрування палітри дозволяє забезпечити захист графічних даних звичайного користувача. Новизною даного методу є унікальність ключа, що не має ані фіксованої довжини, ані фіксованої структури: в ролі ключа шифрування може бути використане практично будь-яке звичайне зображення. Така особливість значно ускладнює підбір ключа і підвищує надійність захисту. Для звичайного методу теоретично мінімальна довжина ключа дорівнює 256 байтам, отже складність підбору ключа становить 2^{11}. Для модифікованого методу мінімальна довжина ключа дорівнює 65536 байтам, отже складність підбору ключа становить = 2^{524288}. Це дозволяє зробити висновок про стійкість запропонованого методу захисту. В процесі шифрування ключ не зазнає жодних змін, отже він не буде мати якихось специфічних ознак. Це дозволяє пересилати ключ або посилання на нього відкритими каналами зв'язку.

Для прискорення виконання алгоритму реалізації методу були застосовані паралельні обчислення на багатоядерному процесорі персонального комп'ютера. Проведені експерименти свідчать про ефективність застосування розпаралелювання.

Незважаючи на збільшення об'єму вихідних даних, що є природнім наслідком їх захисту, у випадку звичайного методу ці дані добре ущільнюються. У випадку модифікованого методу об'єм зашифрованих даних є вдвічі меншим, а самі дані містять менше повторів. Це зменшує ступінь ущільнення даних порівняно зі звичайним методом шифрування палітри, але підвищує стійкість методу, ускладнюючи підбір ключа.

Література

1. Венбо Мао. Современная криптография. Теория и практика = Modern Cryptography: Theory and Practice [Текст]. – М.: Вильямс, 2005. – 768 с. – 2000 экз. – ISBN 5-8459-0847-7, ISBN 0-13-066943-1.

2. Cummins, J. Steganography and Digital Watermarking [Текст] / Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett. – Birmingham : School of Computer Science, the University of Birmingham, 2004.

3. Ladino, J. Data Compression Algorithms. [Електронний ресурс] / Jeffrey N. Ladino / Режим доступу: http://www.ccs.neu.edu/home/jnl22/oldsite/cshonor/jeff.html – [02.03.2014].

4. Сулема Е.С., Широчин С.С. Защита персональных графических данных пользователя при передаче по компьютерным сетям [Текст] // Материалы I Международной научно-практической конференции «Информационные технологии. Проблемы и решения». – Уфа : Уфимский государственный нефтяной технический университет. – 2014.

References

1. Wenbo Mao (2003). *Modern Cryptography: Theory and Practice*. Prentice Hall PTR. ISBN 0-13-066943-1.

2. Cummins Jonathan, Diskin Patrick, Lau Samuel and Parlett Robert (2004). *Steganography and Digital Watermarking*. School of Computer Science, the University of Birmingham.

3. Ladino Jeffrey. *Data Compression Algorithm*s. Available from: http://www.ccs.neu.edu/home/jnl22/oldsite/cshonor/jeff.html [Accessed 2nd March 2014].

4. Sulema Yevgeniya, Shyrochyn Semen (2014). *User's Private Graphical Data Protection While Transferring by Computer Networks*. The Proceedings of the 1st International Scientific-Practical Conference "Information Technologies. Problems and Solutions". Ufa, Russia. The Ufa State Petroleum Technological University. [In Russian].

Рецензія/Peer review : 13.4.2014 р. Надрукована/Printed : 18.5.2014 р.