УДК 004.056.55

А.В. Сидоренко, М.С. Шишко

ШИФРОВАНИЕ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ХАОТИЧЕСКИХ ОТОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Описывается алгоритм шифрования изображения на основе хаотической динамики, оптимизированный для параллельных вычислений. Для уменьшения объема шифруемых данных используется вейвлет-сжатие. При этом часть вейвлет-коэффициентов шифруется с помощью перестановочно-рассеивающего алгоритма, уточняющие коэффициенты – с помощью алгоритма на основе клеточных автоматов. Проводится тестирование алгоритма и показывается хорошая стойкость к статистическому и дифференциальному криптоанализу. Тестирование алгоритма с помощью статистических тестов SP 800-22 позволило установить, что двоичная последовательность, генерируемая алгоритмом, близка к случайной. Показывается, что при оценке производительности алгоритма скорость шифрования при сжатии без потерь составила 8 Мбит/с.

Введение

В настоящее время практически во всех сферах жизнедеятельности человека получают широкое распространение информационные технологии. Неотъемлемой частью мультимедийных приложений является видеоинформация. Технологии беспроводной передачи информации и Интернета не могут гарантировать должную степень защиты и конфиденциальности данных при передаче информации.

Среди разнообразных методов защиты информации и обеспечения ее целостности выделяются криптографические методы. Одним из перспективных направлений в современной криптографии является разработка алгоритмов шифрования на основе динамического хаоса [1–4]. Предложено большое количество алгоритмов шифрования изображений на основе хаоса [2–4]. Типичная структура таких алгоритмов имеет две независимые стадии перестановки и рассеяния (модификации значений) пикселов шифруемого изображения. Алгоритм данного типа представлен в работе [2], в которой используется слегка модифицированная схема перестановки-рассеяния, включающая дополнительную операцию рассеяния путем применения простых последовательных операций XOR и циклического сдвига на каждом этапе перестановки. Это позволяет ускорить процесс шифрования. Для перестановки пикселов используется двухмерное стандартное хаотическое отображение.

В работе [3] описан симметричный алгоритм для шифрования изображений в реальном времени. Для шифрования применяется трехмерное отображение кота Арнольда, полученное авторами статьи путем обобщения его двухмерного отображения. В алгоритме для перестановки пикселов изображения используется отображение кота Арнольда, а для рассеяния – другое отображение, что значительно увеличивает устойчивость алгоритма к статистическим и дифференциальным атакам. Детальный анализ демонстрирует высокую безопасность и большую скорость шифрования данного алгоритма. Более подробно узнать об алгоритмах шифрования изображений можно из работы [4].

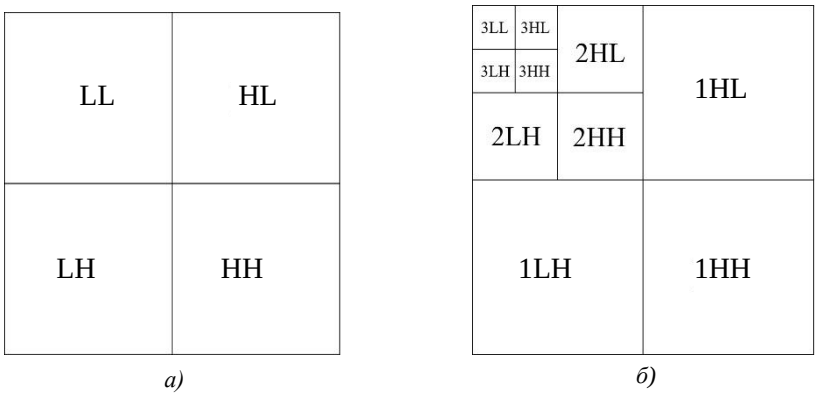
При разработке алгоритма шифрования следует учесть, что изображения характеризуются большим объемом занимаемой памяти, вследствие чего увеличивается время обработки и передачи файла с изображением. Поэтому целесообразно для уменьшения избыточности в изображении использовать предварительное сжатие. Для повышения же быстродействия можно применять распараллеливание операций шифрования и сжатия и выполнять эти операции одновременно в разных программных потоках в многопоточной среде.

В настоящей работе представлен алгоритм шифрования изображения на основе хаотической динамики, оптимизированный для параллельных вычислений. Проведена оценка стойкости данного алгоритма к статистическому и линейному криптоанализу, а также оценка производительности алгоритма.

1. Алгоритм шифрования изображений на основе хаоса

Для уменьшения количества шифруемой информации в процессе шифрования производится сжатие растрового изображения. Для этого к исходному шифруемому изображению применяется дискретное вейвлет-преобразование (ДВП). Для прямого ДВП могут использоваться как обратимые целочисленные, так и необратимые вещественные вейвлеты. Поскольку сжатие без потерь требует, чтобы данные не терялись вследствие округления, для этого типа сжатия используется обратимое вейвлет-преобразование. Напротив, сжатие с потерями допускает некоторую потерю данных в процессе сжатия, и поэтому для сжатия могут быть использованы необратимые вейвлет-преобразования с нерациональными коэффициентами фильтра. Для преобразования на границах сигнала используется симметричное периодическое расширение сигнала. Симметричное расширение добавляет зеркальное изображение сигнала за его пределами, чтобы исключить большие искажения на границах сигнала после вейвлет-преобразования. Для сжатия с потерями применяется вейвлет-преобразование CDF 9/7, а для сжатия без потерь – биортогональный вейвлет CDF 5/3.

В представленной работе данные вейвлет-преобразования применяются поочередно ко всем строкам, а затем ко всем столбцам растрового изображения. При этом изображение разбивается на четыре поддиапазона (рис. 1, *а*): LL, HL, LH, HH. LL-поддиапазон содержит уменьшенную копию исходного изображения, а LH-, HL- и HH-поддиапазоны – уточняющие коэффициенты, позволяющие восстановить изображение. Для LL-поддиапазона вейвлет-преобразование может применяться снова, при этом получится следующий уровень декомпозиции и т. д. На рис. 1, *б* показана схема трехуровневой декомпозиции изображения.



а) *б)*

Рис. 1. Вейвлет-декомпозиции изображения: *а)* одноуровневая; *б)* трехуровневая

После вейвлет-преобразования коэффициенты LL-поддиапазона отделяются от остальных вейвлет-коэффициентов. Как было сказано выше, LL-поддиапазон является уменьшенной копией исходного изображения. Он содержит максимальное количество информации об исходном изображении по сравнению с другими поддиапазонами и поэтому шифруется отдельно. Так как данный поддиапазон является, по сути, обычным изображением с небольшим разрешением, для него можно применить обычные методы шифрования изображений.

В настоящей работе LL-поддиапазон шифруется с помощью перестановочно-рассеивающего алгоритма [5]. Общая архитектура перестановочно-рассеивающих алгоритмов шифрования изображений показана на рис. 2.

В криптографических системах используются два взаимно независимых этапа: перестановка и рассеяние. На этапе перестановки все пикселы изображения меняются местами согласно некоторым преобразованиям, не меняя своих значений. Чтобы декоррелировать смежные пикселы, перестановка выполняется n раз, где $n \geq 1$. После данного этапа каждый пиксел заменяется другим пикселом из этого же изображения. Это приводит к большому беспорядку пикселов. Однако, поскольку значения пикселов не изменялись, гистограмма распределения яркости зашифрованного изображения совпадает с гистограммой исходного изображения, что

негативно сказывается на стойкости алгоритма шифрования. Для устранения данной особенности в алгоритм включен этап диффузии. На этом этапе значения пикселов меняются таким образом, чтобы гистограмма зашифрованного изображения отличалась от гистограммы шифруемого и походила на гистограмму равномерного шума. Цикл перестановки-рассеяния повторяется несколько раз для достижения удовлетворительного уровня беспорядка.

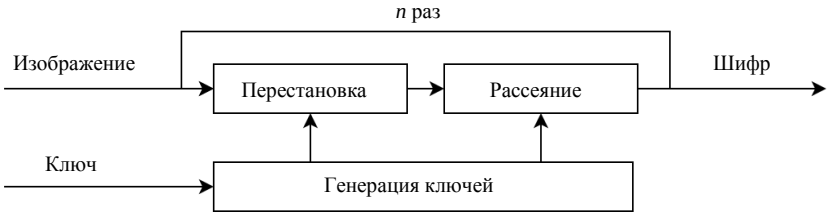


Рис. 2. Перестановочно-рассеивающий алгоритм

Для улучшения безопасности параметры, управляющие перестановкой и диффузией, могут быть различными в разных раундах. Это достигается циклическим ключевым генератором с изначальным секретным ключом. В криптографических системах этап перестановки, этап диффузии и ключевой генератор могут быть реализованы с помощью хаотических отображений.

В данной работе для перестановки и рассеяния используется отображение «тент», имеющее следующий вид:

$$f(x_n)=\begin{cases}\mu x_n \ \text{ при } x_n < \dfrac{1}{2},\\ \mu(1-x_n) \ \text{ при } \ \dfrac{1}{2} \le x_n.\end{cases} \tag{1}$$

Перестановка производится в пределах строки или столбца согласно формуле

$$I(i,k) \leftrightarrow I(i, \lfloor x_k(i) \bullet N \rfloor),\ k=\overline{1,N}, \tag{2}$$

где $I(i,k)$ – пиксел изображения, $x_k(i)$ – значение хаотического отображения для данного пиксела, N – количество столбцов в изображении.

Затем происходит изменение значений пикселов (рассеяние) по правилу

$$I'(i,k) = I(i,k) \oplus \lfloor x_{N+k}(i) \bullet 256 \rfloor,\ k=\overline{1,N}, \tag{3}$$

где $\oplus$ – операция XOR (побитовое исключающее ИЛИ).

После завершения преобразования над строками аналогичные преобразования производятся и над столбцами изображения. Данный цикл может повторяться несколько раз для лучшего зашифрования.

В качестве ключа шифрования используется начальное условие $x_0(-1) \in (0; 1)$ для хаотического отображения. Для обеспечения стойкости алгоритма к дифференциальному криптоанализу данное начальное условие модифицируется с помощью хеш-суммы, вычисляемой по алгоритму SHA-2. Далее 256-битная хеш-сумма H, вычисленная от шифруемого изображения, разбивается на 32-битные блоки и модифицирует начальное условие согласно формулам

$$H = h_1 h_2 h_3 h_4 h_5 h_6 h_7 h_8,$$

$$s = \frac{h_1 \oplus h_2 \oplus ... \oplus h_7 \oplus h_8}{2^{32}}, \qquad (4)$$

$$x_0(0) = \frac{x_0(-1) + s}{2},$$

где $x_0(0) \in (0; 1)$ – модифицированное начальное условие.

Далее на основе модифицированного начального условия $x_0(0)$ генерируются начальные условия для каждой строки и каждого столбца по следующему правилу:

$$x_0(n+1) = \sin(\pi \bullet x_0(n)), \quad n = \overline{0, M+N}, \qquad (5)$$

где $x_0(n)$ – начальные условия для строк и столбцов, N и M – число строк и столбцов соответственно.

Для сжатия изображения производится вложенное кодирование уточняющих вейвлет-коэффициентов с помощью алгоритма HBCT (Hardware Block Cluster Tree), предложенного В.В. Новицким и В.Ю. Цветковым в работе [6]. Данный метод использует построение кластерных деревьев в пределах квадратных блоков битовых плоскостей матрицы вейвлет-коэффициентов и прогрессивное вложенное кодирование.

В процессе кодирования кодер проходит по пространственно-частотным диапазонам битовой плоскости, начиная с LHn и заканчивая HH1. Каждый диапазон компактно описывается с помощью «нуль-дерева», знаков, значимых и уточняющих бит. «Нуль-дерево» строится в два этапа: на первом этапе кодовый блок (поддиапазон) проходит кластеризацию, строя квадродерево (каждый последующий уровень квадродерева получается путем разбиения предыдущего на кластеры размером 2×2 и присвоения соответствующей позиции последующего уровня «0» в случае, если в кластере все биты равны «0», и «1» в любом другом случае); на втором этапе кодер двигается по единичным битам квадродерева, начиная с вершины и игнорируя нулевые кластеры. В алгоритме HBCT используются блоки фиксированной размерности (оптимальная размерность 32×32). Также введена адаптация по количеству уровней подобных деревьев.

После сжатия уточняющие коэффициенты шифруются блочным алгоритмом шифрования на основе обратимых клеточных автоматов (Reversible cellular automata, RCA) [7]. Если определить состояние автомата на каждом шаге как C^t, можно построить обратимый клеточный автомат CA (Cellular Automata) второго порядка, используя элементарный CA:

$$C^t = F(C^{t-1}) \oplus C^{t-2}, \qquad (6)$$

где F представляет собой эволюционную функцию базового клеточного автомата.

Отсюда можно легко найти обратное преобразование:

$$C^{t-2} = F(C^{t-1}) \oplus C^t. \qquad (7)$$

RCA второго порядка, определенные с помощью данных уравнений, всегда обратимы, даже если базовый CA, определенный с использованием отображения F, является необратимым. Этот принцип используется для построения блочного алгоритма шифрования, при помощи которого шифруются уточняющие вейвлет-коэффициенты. Сначала шифруемые данные делятся на блоки размером 256 бит. Каждый такой блок данных шифруется независимо от остальных (рис. 3).

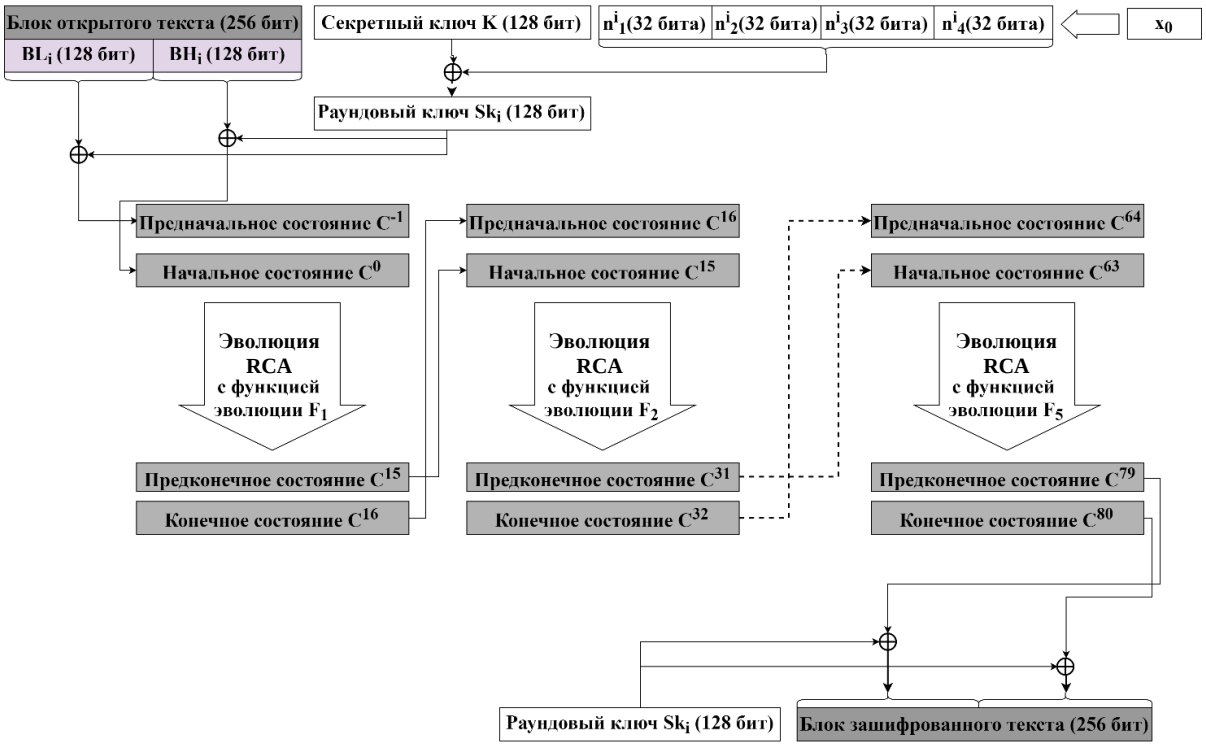


Рис. 3. Схема шифрования уточняющих вейвлет-коэффициентов

Сначала для каждого блока генерируется собственный ключ. Для этого с помощью кусочно-линейного отображения

$$f(x_n)=\begin{cases} \dfrac{x_n}{p}, \text{ если } 0<x_n<p, \\ \dfrac{x_n-p}{1/2-p}, \text{ если } p<x_n<1/2, \\ f(1-x_n), \text{ если } 1/2<x_n<1, \end{cases} \tag{8}$$

для каждого блока генерируются четыре числа:

$$\begin{cases} x_1^1 = f(x_0); \\ x_1^i = f(x_4^{i-1});\ x_2^i = f(x_1^i);\ \ x_3^i = f(x_2^i);\ \ x_4^i = f(x_3^{i-}), \end{cases} \tag{9}$$

где f – кусочно-линейное отображение, i – номер блока, x_0 – начальное условие.

Данные четыре числа приводятся к целочисленному представлению

$$n_k^i = \left\lfloor 2^{32} \bullet x_k^i \right\rfloor, \tag{10}$$

и из них строится маска для блока N_i. Затем формируется подключ Sk_i путем сложения по модулю два маски для блока N_i и общего ключа шифрования K:

$$Sk_i = K \oplus N_i. \tag{11}$$

Далее происходит собственно процесс шифрования. Шифруемый 256-битный блок данных делится на две равные части: BL_i и BH_i (рис. 3). Обе части складываются по модулю два с подключом Sk_i:

$$\begin{aligned} BL'_i &= BL_i \oplus Sk_i, \\ BH'_i &= BH_i \oplus Sk_i. \end{aligned} \tag{12}$$

В качестве начальной конфигурации клеточного автомата C^0 принимается BH'_i, а BL'_i принимается в качестве конфигурации, предшествующей начальной C^{-1}. Затем происходит 80 раундов эволюции клеточного автомата по формуле (6). Каждые 16 раундов эволюционная функция F меняется. В данной работе в качестве эволюционной функции был использован циклический битовый сдвиг. Затем последние конфигурации C^{79} и C^{80} складываются по модулю два с подключом Sk_i, в результате чего получается блок зашифрованного текста.

Одной из основных проблем, присущих алгоритмам шифрования изображений, является большое время шифрования. Это обусловлено тем, что изначально растровые изображения обладают значительной избыточностью и занимают большой объем памяти. Одним из способов решения данной проблемы является сжатие, которое позволяет уменьшить объем шифруемой информации и таким образом увеличить производительность. Для этого в алгоритм включен метод вейвлет-сжатия HBCT. Однако даже предварительное сжатие не всегда позволяет добиться нужного уровня производительности. Одним из способов значительного увеличения производительности является применение параллельных вычислений.

Параллельные вычисления подразумевают такой способ организации компьютерных вычислений, при котором программа разрабатывается как набор взаимодействующих вычислительных процессов, работающих параллельно, независимо друг от друга. Описанный выше алгоритм разрабатывался таким образом, чтобы максимизировать использование параллельных вычислений. Реализация ДВП каждого столбца производится отдельно, что позволяет осуществлять эти вычисления в несколько потоков. При шифровании LL-поддиапазона обработка строк и столбцов также может производиться параллельно в пределах одного раунда при условии предварительно сгенерированных начальных условий хаотического отображения для каждой строки и столбца. Кодирование и шифрование вейвлет-коэффициентов происходят в пределах блоков фиксированного размера, причем каждый блок обрабатывается независимо от других. Это позволяет производить данные вычисления параллельно.

В настоящей работе для параллельных вычислений используются неспециализированные вычисления на графическом процессоре или GPGPU (general-purpose computing for graphics processing units) с применением фреймворка OpenCL.

2. Оценка стойкости алгоритма

Для оценки стойкости предложенного алгоритма к различным видам криптоанализа разработана программа на языке C++. Поскольку в алгоритме для шифрования LL-поддиапазона и остальных коэффициентов применяются принципиально разные методы, было принято решение протестировать их на стойкость отдельно, а также провести тестирование производительности алгоритма в целом. Так как алгоритм шифрования LL-поддиапазона является классическим алгоритмом шифрования изображений типа перестановки-рассеяния, он был протестирован на устойчивость к статистическому и дифференциальному криптоанализу. Блочный алгоритм шифрования вейвлет-коэффициентов был протестирован с помощью набора тестов SP 800-22. В тестах использовались изображения «Lena», «Mandrill» и «Peppers», которые являются стандартными тестовыми изображениями для проверки работы алгоритмов обработки изображений.

Известно, что первостепенное значение для криптографической системы имеет статистический анализ зашифрованного текста. Действительно, идеальный шифр должен быть устойчивым к любым видам статистических атак. Для оценки стойкости алгоритма к статистическому

криптоанализу были вычислены коэффициенты корреляции между соседними пикселами по горизонтали, вертикали и диагонали, а также информационная энтропия.

Корреляция является мерой, которая показывает зависимость между двумя соседними пикселами в изображении. Коэффициент корреляции может быть вычислен следующим образом:

$$r_{XY} = \frac{\operatorname{cov}(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}; \tag{13}$$

$$\operatorname{cov}(X,Y) = \sum_{i=1}^{P}\left(x_i - \overline{X}\right)\left(y_i - \overline{Y}\right); \tag{14}$$

$$D(X) = \frac{1}{P}\sum_{i=1}^{P}\left(x_i - \overline{X}\right)^2, \tag{15}$$

где x_i – яркость i-го пиксела; y_i – яркость соседнего по горизонтали, вертикали или диагонали (в зависимости от типа корреляции) к i-му пиксела; $\overline{X}$, $\overline{Y}$ – средние значения яркости. Коэффициенты корреляции для незашифрованного изображения, как правило, имеют значения, близкие к единице. Это означает, что соседние пикселы связаны между собой некоторой зависимостью. Для зашифрованного же изображения коэффициент корреляции должен стремиться к нулю. Чем ближе коэффициент к нулю, тем меньше связаны соседние пикселы.

Важной характеристикой изображений служит информационная энтропия, которая является мерой неопределенности, связанной со случайной величиной. Она дает количественную оценку информации, содержащейся в данных, как правило, в битах или битах на символ. Энтропия вычисляется по формуле

$$H(m) = \sum_{i=0}^{2N-1} P(m_i)\log_2\frac{1}{P(m_i)}, \tag{16}$$

где $P(m_i)$ – вероятность символа m_i.

Для источника, который выдает 2^8 символа с равной вероятностью, энтропия будет равна восьми. Следовательно, чем ближе значение энтропии изображения к восьми, тем ближе данное изображение к случайному.

Дифференциальный криптоанализ – это один из наиболее популярных видов криптоанализа. Суть его заключается в следующем: криптоаналитик или взломщик создают небольшое изменение в исходном изображении, затем шифруют исходное и измененное изображения, после чего ищут различия в двух шифрах, чтобы найти закономерности между изменениями в шифрах и исходных изображениях.

Для оценки стойкости к данному виду анализа производятся следующие действия. Открытый текст изображения зашифровывается и получается изображение-шифр C1. Затем выбирается произвольный пиксел в открытом тексте, чтобы обеспечить небольшое изменение, которое добавляется (вычитается) к его десятичному значению, или переключается младший значащий бит. Измененное изображение шифруется с использованием того же ключа для получения нового изображения-шифра C2. Эти два изображения-шифра сравниваются с помощью следующих критериев [8]:

1. Процента измененных пикселов (NPCR – Near Pixel Change Rate). Процент различных пикселов в изображениях C1 и C2 рассчитывается следующим образом:

$$NPCR = \frac{\sum_{i=1,j=1}^{M,N} D(i,j)}{M \times N} \times 100\ \%; \tag{17}$$

$$D(i,j)=\begin{cases}1, \text{ если } & C1(i,j)=C2(i,j);\\ 0, \text{ если } & C1(i,j)\neq C2(i,j).\end{cases} \quad (18)$$

Чем ближе коэффициент NPCR к 100 %, тем большую стойкость имеет рассматриваемый алгоритм к дифференциальному криптоанализу.

2. Среднего изменения интенсивности (UACI – Unified Averaged Changed Intensity) – меры различия средней интенсивности между двумя шифрами. Определяется по формуле

$$UACI=\frac{1}{M\times N}\sum_{i=1,j=1}^{M,N}\frac{C1(i,j)-C2(i,j)}{L}\times 100\,\%, \quad (19)$$

где L – число возможных уровней яркости. Чем ближе данный показатель к 33 %, тем больше стойкость к дифференциальному криптоанализу.

В табл. 1 представлены результаты тестирования части алгоритма, ответственной за шифрование LL-поддиапазона, а также статистические коэффициенты для незашифрованных изображений. Используются тестовые изображения «Lena 256×256», «Mandrill 256×256», «Peppers 256×256». Из таблицы видно, что коэффициент корреляции для незашифрованного изображения (откр.) близок к единице. Это означает, что значения соседних пикселов в незашифрованном изображении коррелируют. Для зашифрованного изображения (зашифр.) коэффициент корреляции близок к нулю. Это означает, что корреляция между соседними пикселами практически отсутствует. Данный факт затрудняет криптоаналитику или злоумышленнику статистический криптоанализ зашифрованного текста. Как видно из таблицы, энтропия зашифрованных изображений больше, чем энтропия незашифрованных, и близка к восьми. Даже для изображения с наименьшей энтропией «Peppers 256×256» энтропия зашифрованного изображения остается на уровне остальных и близка к своему максимальному значению. Значит, алгоритм шифрования вносит достаточную долю неопределенности в зашифрованное изображение и делает его похожим на случайный набор пикселов. Вкупе с низкой корреляцией данный факт позволяет говорить о высокой стойкости алгоритма к статистическому криптоанализу. Также по результатам теста видно, что NPCR близок к 100 %, а UACI – к 33,3 %. Это означает хорошую стойкость алгоритма к дифференциальному криптоанализу.

Таблица 1

Результаты тестирования

Изображение		Корреляция			Энтропия	NPCR, %	UACI, %
		гориз.	вертик.	диагон.			
«Lena 256×256»	Откр.	0,9223	0,8742	0,8656	7,75	–	–
	Зашифр.	–0,0176	–0,0086	–0,0729	7,96	99,55	33,44
«Mandrill 256×256»	Откр.	0,9813	0,9887	0,9699	7,76	–	–
	Зашифр.	0,0180	0,0019	–0,0090	7,97	99,62	33,41
«Peppers 256×256»	Откр.	0,9673	0,9658	0,9670	7,49	–	–
	Зашифр.	0,0006	–0,0132	–0,0377	7,95	99,60	33,37

Вторую составляющую алгоритма шифрования, отвечающую за шифрование уточняющих вейвлет-коэффициентов, было решено протестировать с помощью набора тестов SP 800-22 [9]. Данный инструмент был обнародован в 2003 г. Американским Национальным институтом

стандартов и технологий (American National Institute of Standards and Technology, NIST). Он представляет собой стандарт для тестирования статистики случайных и псевдослучайных последовательностей. Для проведения такого тестирования уточняющие вейвлет-коэффициенты тестовых изображений были зашифрованы и преобразованы в битовую последовательность. Тесты проводились на последовательности длиной 10^6 бит, разбитой на 10 равных по длине подпоследовательностей. В результате алгоритм не прошел универсальный тест Маурера для трех из четырех изображений. Следовательно, в двоичной последовательности существуют значительно сжимаемые участки. Также алгоритм не прошел тест на последовательности и тест энтропии. Значит, алгоритм генерирует битовую последовательность, которая по данным критериям не является случайной. Между тем остальные тесты были успешно пройдены и, следовательно, двоичная последовательность достаточно близка к случайной, чтобы препятствовать несанкционированному получению информации злоумышленниками.

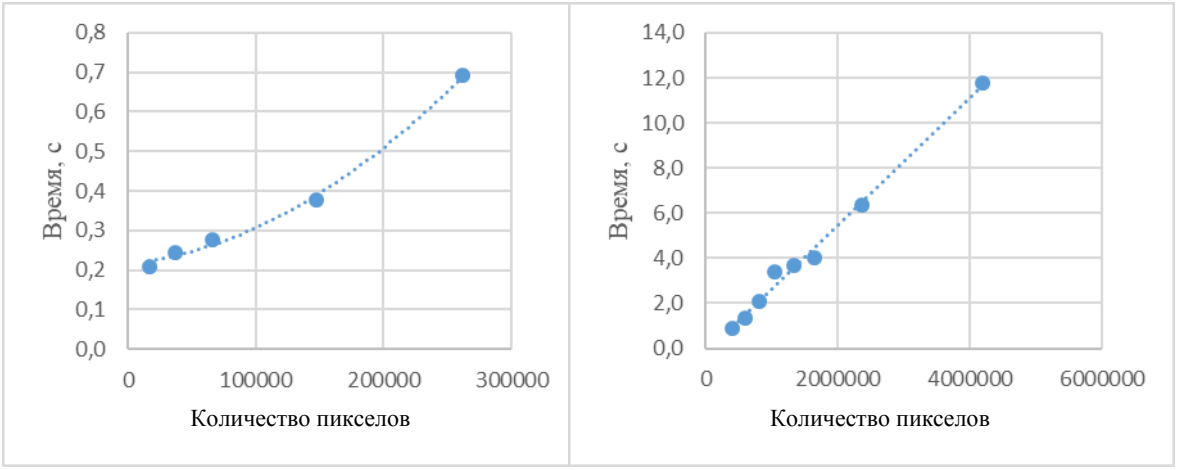
3. Оценка производительности алгоритма

В работе проведено тестирование производительности алгоритма. Для этого данным алгоритмом с использованием сжатия без потерь было зашифровано тестовое изображение «Мандрил» в различных разрешениях (табл. 2). Тестирование проводилось с помощью интегрированного графического процессора Intel(R) HD Graphics 4000 и центрального процессора Intel Core i5-3230M. На рис. 4, *а* показана зависимость времени шифрования от количества пикселов для изображений малого размера (разрешение до 512×512 пикселов). Из графика видно, что зависимость имеет экспоненциальный характер. Однако для большого размера шифруемого изображения (разрешение более 512×512 точек) зависимость имеет линейный характер (рис. 4, *б*). Это происходит из-за того, что при малом размере изображения решающую роль в формировании времени шифрования играют вспомогательные процессы, такие как перенос данных из оперативной памяти в память графического процессора и обратно, а также запись в файл. В целом же скорость шифрования при сжатии без потерь находится на уровне 8 Мбит/с. Скорость шифрования может быть увеличена при использовании сжатия с потерями, однако качество восстановленного изображения будет хуже.

Таблица 2

Оценки времени шифрования изображения «Mandrill»

Разрешение	Количество пикселов	Размер, кбайт	Время шифрования, с
128×128	16 384	48	0,209
192×192	36 864	108	0,243
256×256	65 536	192	0,278
384×344	147 456	432	0,378
512×512	262 144	768	0,692
640×640	409 600	1200	0,907
768×768	589 824	1728	1,343
896×896	802 816	2352	2,102
1024×1024	1 048 576	3072	3,398
1152×1152	1 327 104	3888	3,664
1280×1280	1 638 400	4800	4,038
1536×1536	2 359 296	6912	6,351



а) *б)*

Рис. 4. Зависимость времени шифрования от количества пикселов в изображении:
а) для изображений малого размера; *б)* для изображений большого размера

Заключение

В работе представлен алгоритм шифрования изображений с применением хаотической динамики, оптимизированный для параллельных вычислений. В качестве хаотических отображений для перестановочно-рассеивающего алгоритма используется отображение «тент», а для шифрования уточняющих вейвлет-коэффициентов – хаотическое кусочно-линейное отображение.

Проведена оценка стойкости алгоритма шифрования LL-поддиапазона вейвлет-коэффициентов к статистическому и дифференциальному криптоанализу, которая показала высокий уровень стойкости этой части алгоритма к данным видам криптоанализа.

Протестирован алгоритм шифрования уточняющих вейвлет-коэффициентов с использованием набора статистических тестов SP 800-22. Выявлена схожесть генерируемого алгоритмом выходного битового потока со случайным потоком по большинству критериев, что означает высокую стойкость данной части алгоритма к статистическому криптоанализу.

Тестирование производительности разработанного алгоритма показало его высокий уровень: средняя скорость шифрования с использованием сжатия без потери качества составляет 8 Мбит/с.

Список литературы

1. Сидоренко, А.В. Шифрование данных с использованием хаотической динамики в сенсорной сети / А.В. Сидоренко, К. С. Мулярчик // Доклады БГУИР. – 2015. – № 6(92). – С. 41–47.

2. Wong, K.-W. A Fast Image Encryption Scheme based on Chaotic Standard Map / K.-W. Wong, S. K. Bernie, W.-S. Law // Physics Letters A. – 2008. – Vol. 372. – P. 2645–2652.

3. Chen, G. A symmetric image encryption scheme based on 3D chaotic cat maps / G. Chen, Y. Mao, Ch.K. Chui // Chaos, Solitons and Fractals. – 2004. – Vol. 21. – P. 749–761.

4. Khan, M. A Literature Review on Image Encryption Techniques / M. Khan, T. Shah // 3D Res. – 2014. – Vol. 5, iss. 4.

5. Wong, K.-W. Image encryption using chaotic maps / K.-W. Wong // Intel. Computing Based on Chaos. – 2009. – Vol. 184. – P. 333–354.

6. Новицкий, В.В. Сжатие полутоновых изображений на основе кластеризации и прогрессивного вложенного кодирования вейвлет-коэффициентов / В.В. Новицкий, В.Ю. Цветков // Телекоммуникации: сети и технологии, алгебраическое кодирование и безопасность данных : материалы Междунар. науч.-техн. семинара, Минск, апрель–декабрь 2015 г. – Минск : БГУИР, 2015. – С. 45–51.

7. Faraoun, K.M. A parallel block-based encryption schema for digital images using reversible cellular automata / K.M. Faraoun // Engineering Science and Technology. – 2014. – Vol. 17. – P. 85–94.

8. Wu, Y. NPCR and UACI randomness tests for image encryption / Y. Wu, J.P. Noonan, S. Agaian // Multidisciplinary journals in science and technology. Journal of selected areas in telecommunications (JSAT). – 2011. – Apr. ed. – P. 31–38.

9. A statistical test suite for random and pseudorandom number generators for cryptographic applications / A. Rukhin [et al.]. – Special Publication 800-22. Revision 1a. – Gaithersburg : National institute of standards and technology, 2010. – 131 p.

Поступила 05.10.2017

Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: sidorenkoa@yandex.ru,
maxshishko@yandex.ru

A.V. Sidorenko, M.S. Shishko

ENCRYPTION OF IMAGES ON THE BASIS OF CHAOTIC MAPPING AND PARALLEL COMPUTING

The chaos-based image encryption algorithm using parallel computing is described. To reduce the amount of encrypted data the wavelet-based compression is used. Some of the wavelet coefficients are encrypted with the use of confusion-diffusion scheme and the qualifying coefficients are encrypted by the algorithm based on reversible cellular automations. Resistance of this algorithm to statistical and differential cryptanalysis was evaluated. The SP 800-22 statistical tests were performed. These tests demonstrated that the binary sequence generated by the algorithm is close to the random one. The algorithm testing was performed. The algorithm productivity was estimated. The encryption rate for the test image in the case of lossless compression was 8 Mbit/s.