УДК 004.05:061.068

ШИФРОВАНИЕ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ДНК-ПОСЛЕДОВАТЕЛЬНОСТЕЙ, ХАОТИЧЕСКОЙ ДИНАМИКИ И ХЕШ-ФУНКЦИЙ

А.В. СИДОРЕНКО, М.С. ШИШКО

Белорусский государственный университет
Независимости, 4, Минск, 220030, Беларусь

Поступила в редакцию 6 мая 2016

Описан алгоритм шифрования изображения с использованием ДНК-последовательностей, хаотической динамики и хеш-функций. Проведена оценка стойкости данного алгоритма к статистическому и линейному криптоанализу при различных хаотических отображениях.

Ключевые слова: шифрование изображений, хаотическое отображение, статистический криптоанализ, дифференциальный криптоанализ.

Введение

В настоящее время практически во всех сферах жизнедеятельности человека получают широкое распространение информационные технологии. Неотъемлемой частью мультимедийных приложений является видеоинформация. Одновременно с этим все большее развитие получают технологии беспроводной передачи информации и интернет. Однако указанные каналы передачи информации не могут гарантировать должную степень защиты и конфиденциальности данных.

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, 2]. При шифровании изображений следует учесть, что характерным для них является низкое значение информационной энтропии. Это приводит к необходимости создания алгоритмов шифрования с увеличением указанного показателя. Среди таких алгоритмов следует выделить алгоритмы с использованием ДНК-последовательностей.

В данной работе представлен алгоритм шифрования изображений с использованием динамического хаоса и ДНК-последовательностей. Проведенена оценка стойкости данного алгоритма к статистическому и дифференциальному криптоанализу при использовании различных отображений, включая отображения Лоренца, кусочно-линейное, логистическое, а также отображение пекаря.

Алгоритм шифрования изображений на основе хаоса

В основе рассматриваемой авторами системы лежит алгоритм шифрования, предложенный R. Guesmi в работе [3]. Алгоритм использует маскирование с помощью модели дезоксирибонуклеиновой кислоты (ДНК), хеш-функцию SHA-256 (Secure Hash Algorithm – безопасный алгоритм хеширования) и хаотические отображения. Модель ДНК-последовательности при шифровании позволяет разбить изображение на более мелкие части, чем отдельные пиксели. Это приводит к тому, что в процессе перестановки элементов ДНК-последовательности устраняется корреляция между пикселями, меняются значения пикселей, что приводит к увеличению информационной энтропии.

В алгоритме также предусмотрено применение хеш-функций. Хеш-функция представляет собой преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины, называемую хеш-суммой. В рассматриваемой криптосистеме используется хеш-функция SHA-256. Криптосистема использует 256-битное значение, вычисленное данной функцией, в качестве ключа. Таким образом, размер ключевого пространства данной системы составляет 2^{256}, что повышает стойкость системы к атакам грубой силой. Особенностью данной хеш-функции является то, что даже при отличии шифруемых изображений в один бит, значения хеш-сумм сильно различаются. А это означает, что и шифрование будет происходить по-разному. При этом увеличивается стойкость алгоритма к дифференциальному криптоанализу.

В рассматриваемом алгоритме используются хаотические отображения. Они характеризуются псевдослучайным поведением и высокой чувствительностью к начальным условиям, что позволяет увеличить стойкость алгоритма к различным видам криптоатак.

Алгоритм шифрования состоит из следующих этапов.

1. Вычисление хеш-суммы для изображения с использованием хеш-функции SHA-256. Получаем хеш-сумму длиной в 256 бит. В качестве ключа используется значение хеш-суммы. Производится вычисление начальных условий для хаотического отображения.

2. Определение начальных условий для хаотического отображения производится с помощью вычисленной на предыдущем этапе хеш-суммы. При этом хеш-сумма H делится на блоки по 8 бит каждый:

$$H = h_1 h_2 ... h_{32}. \quad (1)$$

Формирование начальных условий для хаотического отображения осуществляется согласно выражениям:

$$x_0 = \frac{k_1 \oplus k_2 \oplus ... \oplus k_{11}}{256}$$
$$y_0 = \frac{k_{12} \oplus k_{13} \oplus ... \oplus k_{22}}{256}. \quad (2)$$
$$z_0 = \frac{k_{23} \oplus k_{24} \oplus ... \oplus k_{32}}{256}$$

3. Преобразование хеш-суммы и изображения в ДНК-последовательности производится согласно одной из 8 кодировок (табл. 1). Каждые два бита преобразуются в один из нуклеотидов в соответствии с выбранной кодировкой. В результате получается ДНК-последовательность, соответствующая хеш-сумме, и три ДНК-последовательности, соответствующие каждой цветовой компоненте цветного изображения (красной, зеленой и синей).

Таблица 1. **ДНК-кодировки**

Операнд	**Номер кодировки**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
A	00	00	01	01	10	10	11	11
G	11	11	10	10	01	01	00	00
T	01	10	00	11	00	11	01	10
C	10	01	11	00	11	00	10	01

4. Проведение операции ДНК-XOR между закодированными хеш-суммой и каждой цветовой компонентой изображения осуществляется согласно выражению:

$$I_j^{'} = I_j \; XOR \, H_{j \bmod 128} \quad j = \overline{1, M*N*4}, \quad (3)$$

где I – ДНК-последовательность цветовой компоненты изображения, H – ДНК-последовательность хеш-суммы; M, N – число строк и столбцов цветного изображения, XOR – бинарная операция ДНК-XOR, производимая в соответствии с табл. 2.

Таблица 2. **Операция ДНК-XOR**

Левый операнд	**Правый операнд**			
	A	**G**	**C**	**T**
A	A	G	C	T
G	G	A	T	C
C	C	T	A	G
T	T	C	G	A

5. Генерация хаотических последовательностей и формирование перестановочной схемы. Хаотические последовательности генерируется путем итерации хаотической функции *f*:

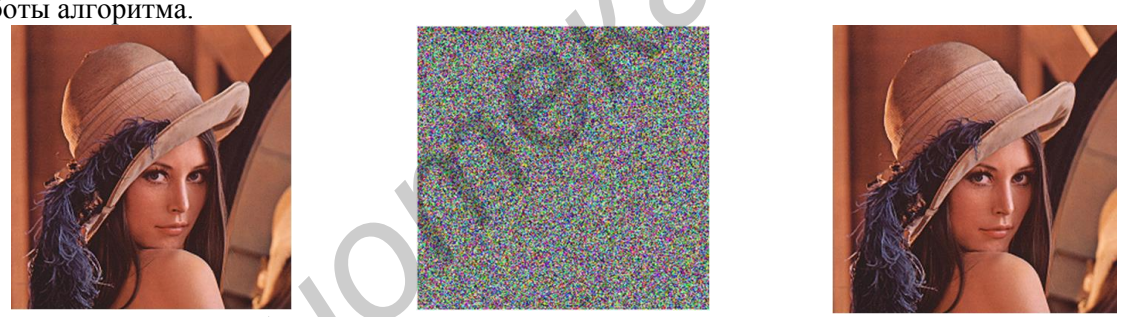
$$(x_{n+1}, y_{n+1}, z_{n+1}) = f(x_n, y_n, z_n) \quad n = \overline{0, M*N*4-1}. \tag{4}$$

Получаем три хаотические последовательности *x*, *y* и *z*, которые сортируются, формируя при этом шаблон перестановок. Шаблон перестановок создается следующим образом: если элемент x_n до сортировки находился на позиции *n*, а после сортировки оказался на позиции *m*, то в перестановочной схеме элементу *n* будет соответствовать перестановка на позицию *m*. Так с помощью хаотических последовательностей *x*, *y* и *z* формируются перестановочные схемы для закодированных красной, зеленой и синей компонент цветного изображения, соответственно.

6. Перестановка элементов ДНК-последовательностей цветовых компонент изображения производится с помощью перестановочных схем, выработанных на предыдущем этапе.

7. Декодирование ДНК-последовательностей, соответствующих цветовым компонентам изображения, производится с помощью выбранной кодировки, в результате которого получается зашифрованное изображение.

Процесс расшифрования происходит в обратном порядке. На рис. 1 показан пример работы алгоритма.



а *б* *в*

Рис. 1. Примеры изображений, *а* – исходного, *б* – зашифрованного, *в* – расшифрованного

Используемые хаотические отображения

При разработке данного алгоритма могут использоваться различные хаотические отображения. В представленной работе использовались четыре хаотических отображения. Для алгоритма необходимо применять трехмерное отображение, то есть функцию трех переменных. При использовании же одномерных отображений для каждой координаты выбираются различные начальные условия. Тестирование алгоритма проводилось с помощью следующих отображений.

1. Отображение Лоренца [4]:

$$\begin{cases} \dfrac{\partial x}{\partial t} = \sigma(y - x) \\ \dfrac{\partial y}{\partial t} = x(r - z) - y \\ \dfrac{\partial z}{\partial t} = xy - bz \end{cases} \tag{5}$$

2. Кусочно-линейное отображение [5]:

$$f(x_n)=\begin{cases}\dfrac{x_n}{p} & 0<x_n<p\\ \dfrac{x_n-p}{1/2-p} & p<x_n<1/2\\ f(1-x_n) & 1/2<x_n<1\end{cases} \quad (6)$$

3. Логистическое отображение [6]:

$$x_{n+1}=Mx_n(1-x_n) \quad (7)$$

4. Трехмерное отображение пекаря [7]:

$$(x_{n+1},y_{n+1},z_{n+1},)=\begin{cases}\left(2x_n,2y_n,\dfrac{z_n}{4}\right) & 0<x_n<\dfrac{1}{2} \quad 0<y_n<\dfrac{1}{2}\\ \left(2x_n,2y_n-1,\dfrac{z_n}{4}+\dfrac{1}{2}\right) & 0<x_n<\dfrac{1}{2} \quad \dfrac{1}{2}<y_n<1\\ \left(2x_n-1,2y_n,\dfrac{z_n}{4}+\dfrac{1}{4}\right) & \dfrac{1}{2}<x_n<1 \quad 0<y_n<\dfrac{1}{2}\\ \left(2x_n-1,2y_n-1,\dfrac{z_n}{4}+\dfrac{3}{4}\right) & \dfrac{1}{2}<x_n<1 \quad \dfrac{1}{2}<y_n<1\end{cases} \quad (8)$$

Оценка стойкости алгоритма

Для оценки стойкости алгоритма к различным видам криптоанализа авторами на языке C^{++} разработана компьютерная программа. Известно, что важное 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 корреляция между соседними пикселями по горизонтали, вертикали и диагонали, а также информационная энтропия.

Корреляция является мерой, которая показывает взаимосвязь между двумя соседними пикселями на изображении. Коэффициент корреляции может быть вычислен по следующей формуле

$$r_{XY}=\frac{\operatorname{cov}(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}, \quad (9)$$

$$\operatorname{cov}(X,Y)=\sum_{i=1}^{P}\left(x_i-\overline{X}\right)\left(y_i-\overline{Y}\right), \quad (10)$$

$$D(X)=\frac{1}{P}\sum_{i=1}^{P}\left(x_i-\overline{X}\right)^2, \quad (11)$$

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

Важной характеристикой изображений является информационная энтропия. Энтропия 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 следующей формуле:

$$H(m) = \sum_{i=0}^{2N-1} P(m_i) \log_2 \frac{1}{P(m_i)}, \quad (12)$$

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

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

Дифференциальный криптоанализ – это один из наиболее популярных видов криптоанализа. Суть его в следующем: взломщики 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 пиксель 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. Процент измененных пикселей (NPCR – Near Pixel Change Rate): рассчитывается процент различных пикселей в изображения C1 и C2 согласно следующим формулам:

$$NPCR = \frac{\sum_{i=1,j=1}^{M,N} D(i,j)}{M * N} * 100\%, \quad (13)$$

$$D(i,j) = \begin{cases} 1 & C1(i,j) = C2(i,j) \\ 0 & C2(i,j) \neq C2(i,j) \end{cases} \quad (14)$$

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

2. Среднее изменение интенсивности (UACI – Unified Averaged Changed Intensity) – это мера различия средней интенсивности между двумя шифрами, определяется формулой:

$$UACI = \frac{1}{M * N} \sum_{i=1,j=1}^{M,N} \frac{C1(i,j) - C2(i,j)}{L} * 100\% \quad (15)$$

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

Тестирование проводились для изображения «Лена» (рис. 1) при двух разрешениях (256×256 и 512×512 пикселей). Результаты тестов представлены в табл. 3. По результатам тестирования видно, что корреляция между пикселями стремится к 0, а энтропия к 8, что означает хорошую стойкость алгоритма к статистическому криптоанализу. В свою очередь, коэффициент NPCR стремится к 100 %, а UACI к 33 %, что свидетельствует о хорошей стойкости и к дифференциальному криптоанализу.

Таблица 3. **Результаты тестирования**

Изображение	Отображение	Корреляция			Энтропия	NPCR, %	UACI, %	T, с
		Гориз.	Верт.	Диаг.				
Lena 256×256	**Лоренца**	0,0267	–0,0033	–0,0061	7,9657	99,5694	33,3647	0,8876
	Кусочно-линейное	0,0014	0,0043	0,0055	7,9656	99,5259	33,1828	1,2192
	Логистическое	0,0067	0,0005	–0,0066	7,9662	99,5719	32,8946	1,1834
	Пекаря	0,0087	–0,0075	0,0038	7,9588	99,6871	34,0636	0,5927
Lena 512×512	**Лоренца**	0,0087	–0,0129	–0,0014	7,9651	99,5540	32,0870	4,1634
	Кусочно-линейное	–0,0138	–0,0017	–0,0017	7,9659	99,6272	31,2592	6,1986
	Логистическое	–0,0181	–0,0109	0,0113	7,9656	99,6875	35,5368	6,3166
	Пекаря	0,0375	0,0029	0,0002	7,9538	99,3749	34,7802	2,4982

По коэффициенту корреляции все хаотические отображения показали примерно одинаковые результаты. По энтропии, а также коэффициентам NPCR и UACI немного хуже остальных является отображение пекаря, однако время шифрования для алгоритма с использованием данного отображения почти в два раза меньше, чем с остальными отображениями. Так как быстродействие является важным параметром для криптографической системы, наиболее подходящим для криптографической системы является отображение пекаря.

Заключение

Проведен анализ стойкости алгоритма шифрования изображений с использованием ДНК-последовательностей, динамического хаоса и хеш-функций к статистическому и дифференциальному криптоанализу при различных используемых отображениях. Анализ показал хорошую стойкость алгоритма к вышеперечисленным видам криптоанализа для всех исследуемых отображений. Однако алгоритм шифрования с отображением пекаря показал лучшее быстродействие по сравнению с использованием остальных отображений. Поэтому отображение пекаря является наиболее приемлемым для данного алгоритма по сравнению с другими.

IMAGE ENCRYPTION USING DNA-SEQUENCES, CHAOTIC DYNAMICS AND HASH FUCTIONS

A.V. SIDORENKO, M.S. SHISHKO

Abstract

An image encryption algorithm based on DNA-sequences, chaotic dynamics and hash functions is described. It is evaluated the resistance of the investigated algorithm to statistical and linear cryptanalysis which are depended of the chaotic maps.

Keywords: image encryption, chaotic map, statistical cryptanalysis, differential cryptanalysis.

Список литературы

1. *Дмитриев А.С., Панас А.И.* Динамический хаос. М., 2002.
2. *Сидоренко А. В., Мулярчик К. С.* // Докл. БГУИР. 2015. № 6. С. 41–47.
3. *Guesmi R., Farah M. A. B., Kachouri A. et. al.* // Nonlinear Dynamics. 2016. Vol. 83, Iss. 3. P. 1123–1136.
4. *Lorenz E.* // Journal of the Atmospheric Sciences. № 20 (2). P. 130–141.
5. *Arroyo D., Alvarez G., Fernandez V.* // arXiv:0805.4355v1. 28 May 2008.
6. *May R.* // Nature 261(5560). 1976. P. 459–467.
7. *Mao Y.* // International Journal of Bifurcation and Chaos. 2004. Vol. 14, № 10. P. 3613–3624
8. *Yue Wu* // Journal of Selected Areas in Telecommunications (JSAT). April, 2011. P. 31–38.