МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 51

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

ассистент		М.Н. Исаева
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

БЛОКОВЫЕ ШИФРЫ

по курсу: КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 5912		В.И.Сазонова
	подпись, дата	инициалы, фамилия

Санкт-Петербург
2021

Задача

Вариант 35. Реализовать алгоритм шифрования Twofish, предусмотреть возможность работы алгоритма в режиме OFB.

Описание алгоритма Twofish

Процедура расширения ключа

Процедура расширения ключа формирует 40 32-битных подключей для использования их в 16 раундах алгоритма и для выполнения операций отбеливания.

Алгоритм Twofish использует ключи шифрования любого размера до 256 битов включительно. Исходный ключ, при необходимости, дополняется нулевыми битами до ближайшего стандартного размера(128, 192, 256). Процедура расширения ключа обрабатывает дополненный таким образом ключ.

Предварительная обработка ключа. Инициализация переменных:

- $k = N/64$, N – размер дополненного ключа шифрования в битах
- Ключ шифрования представляется в виде $8k$ байтов $m_0 \dots m_{8k-1}$ или в виде $2k$ 32-битных слов, обозначаемых как $M_0 \dots M_{2k-1}$.
- Формируются 3 массива, каждый из которых состоит из k 32-битных слов:

$$M_e = (M_0, M_2, \dots M_{2k-2});$$ - массив четных слов

$$M_0 = (M_1, M_3, \dots M_{2k-1});$$ - массив нечетных слов

$$V = (V_{k-1}, V_{k-2}, \dots V_0),$$

где:

$$V_i = \sum_{j=0}^{3} v_{i,j} * 2^{8j};$$

$$\begin{pmatrix} v_{i,0} \\ v_{i,1} \\ v_{i,2} \\ v_{i,3} \end{pmatrix} = M_2 * \begin{pmatrix} m_{8i} \\ m_{8i+1} \\ m_{8i+2} \\ m_{8i+3} \\ m_{8i+4} \\ m_{8i+5} \\ m_{8i+6} \\ m_{8i+7} \end{pmatrix}.$$

Матрица M_2 представлена в приложении 1.

Генерация подключей $K_0 \ldots K_{39}$ производится на основе вычисленных на предварительном этапе массивов M_e и M_o следующим образом(рис.1):

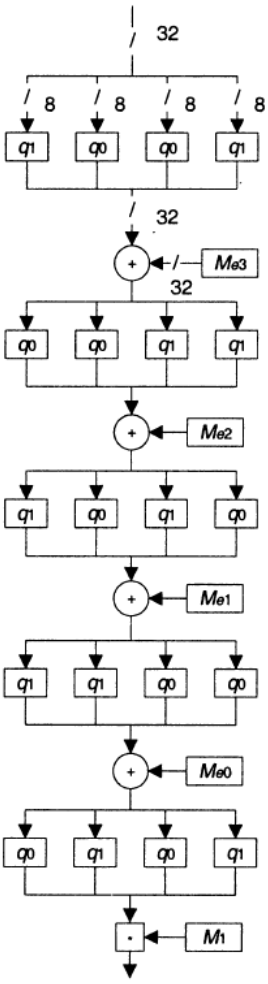
$$k_{2i} = A_i + B_i \bmod 2^{32};$$
$$k_{2i+1} = (A_i + 2B_i \bmod 2^{32}) <<< 9,$$

где $i = 0 \ldots 19$, а A_i и B_i — промежуточные величины, вычисляемые так:

$$A_i = h(2i\rho, M_e);$$
$$B_i = h((2i+1)\rho, M_o) <<< 8.$$
$$\rho = 2^{24} + 2^{16} + 2^8 + 1$$

Рисунок 1. Генерация подключей

Функция *h()* выполняется в несколько шагов, количество которых зависит от размера дополненного ключа в 64-битных фрагментах, т.е. от описанного выше значения *к*. В качестве параметров функция принимает 32-битное слово и массив 32-битных слов размерностью *к(Me)*. Алгоритм функции *h()*(рис.2):



Выполняется, если *к* меньше 4

Не выполняется, если *к* равен 2

Рисунок 2. Алгоритм функции h()

Матрица M_1 представлена в приложении 1.

Операции q_0 и q_1 вычисляют выходные значения с использованием нескольких таблиц замен 4x4 следующим образом(рис.3):

$$a_0 = \lfloor x/16 \rfloor;$$
$$b_0 = x \bmod 16;$$
$$a_1 = a_0 \oplus b_0;$$
$$b_1 = a_0 \oplus (b_0 >>>_4 1) \oplus 8a_0 \bmod 16;$$
$$a_2 = t_0(a_1);$$
$$b_2 = t_1(b_1);$$
$$a_3 = a_2 \oplus b_2;$$
$$b_3 = a_2 \oplus (b_2 >>>_4 1) \oplus 8a_2 \bmod 16;$$
$$a_4 = t_2(a_3);$$
$$b_4 = t_3(b_3);$$
$$y = 16b_4 + a_4,$$

Рисунок 3. Операции q_0 и q_1

где x и y – входное и выходное значения соответственно, t_i – табличные замены, различные для q_0 и q_1 ; таблицы замен представлены в приложении 1.

Twofish

Алгоритм Twofish разбивает шифруемые данные на четыре 32-битных субблока(*A, B, C, D*), над которыми производится 16 раундов преобразований, в каждом из которых выполняются следующие операции(рис.4):

$$B = B <<< 8;$$
$$A = g(A);$$
$$B = g(B);$$
$$A = A + B \bmod 2^{32};$$
$$B = A + B \bmod 2^{32};$$
$$A = A + K_{2r+8} \bmod 2^{32};$$
$$B = B + K_{2r+9} \bmod 2^{32};$$
$$C = C \oplus A;$$
$$D = D <<< 1;$$
$$D = D \oplus B;$$
$$C = C >>> 1.$$

Рисунок 4.Структура алгоритма

Перед первым раундом выполняется входное отбеливание с использованием подключей $K_0 \ldots K_3$, после заключительного раунда выполняется выходное отбеливание с использованием подключей $K_4 \ldots K_7$.

В конце каждого раунда, за исключением последнего, субблоки *A*(до обработки) и *C* меняются местами, субблоки *B*(до обработки) и *D* также меняются местами.

Операция *g()* представляет собой описанную ранее функцию *h(),* использующую в качестве входного значения 32-битный субблок *A* или *B,* а в качестве входного массива – описанный ранее массив *V*.

Описание режима шифрования OFB

Режим обратной связи вывода превращает блочный шифр в синхронный шифр потока: он генерирует ключевые блоки, которые являются результатом сложения с блоками открытого текста, чтобы получить зашифрованный текст. Зеркальное отражение в зашифрованном тексте производит зеркально отражённый бит в открытом тексте в том же самом местоположении. Это свойство позволяет многим кодам с исправлением ошибок функционировать как обычно, даже когда исправление ошибок применено перед кодированием.

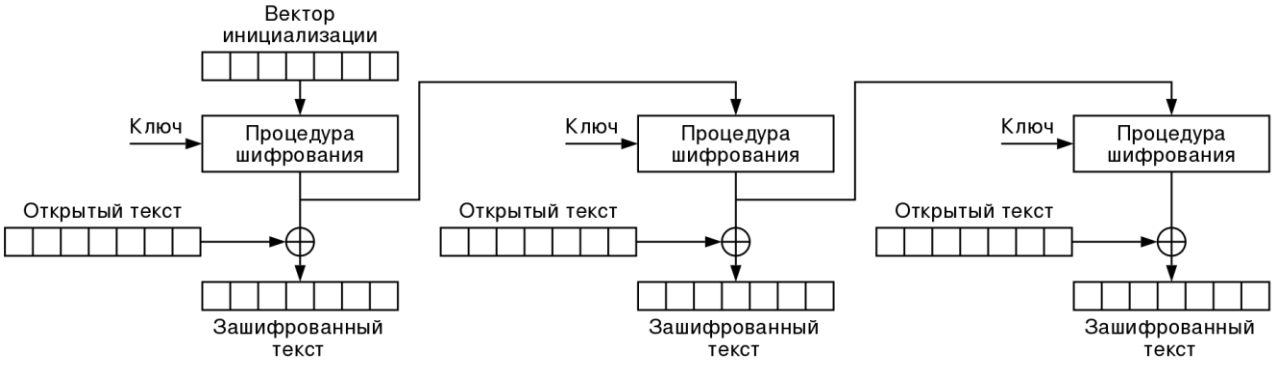


Рисунок 5. Шифрование в режиме OFB

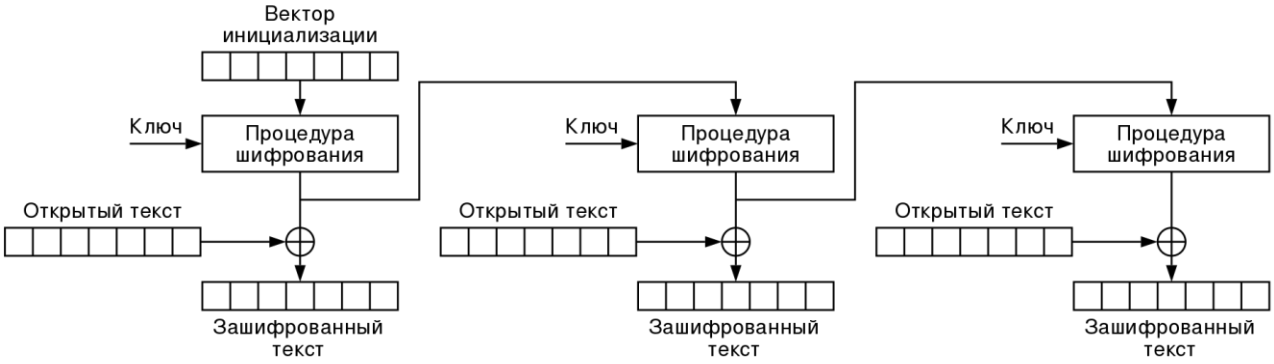


Рисунок 6. Расшифрование в режиме OFB

Описание реализации

При запуске программы вне зависимости от выбранного режима работы генерируется ключ. Если выбран режим шифрования OFB, генерируется вектор инициализации. Затем данные кодируются в соответствии с выбранным режимом и записываются в новый файл. После этого данные расшифровываются и записываются в новый файл.

Примеры

Twofish

Шифрование и расшифрование текстового файла:

Сгенерированный ключ: MP2ZDO00LFYG1U24TN7I06E5OAKFNTM0

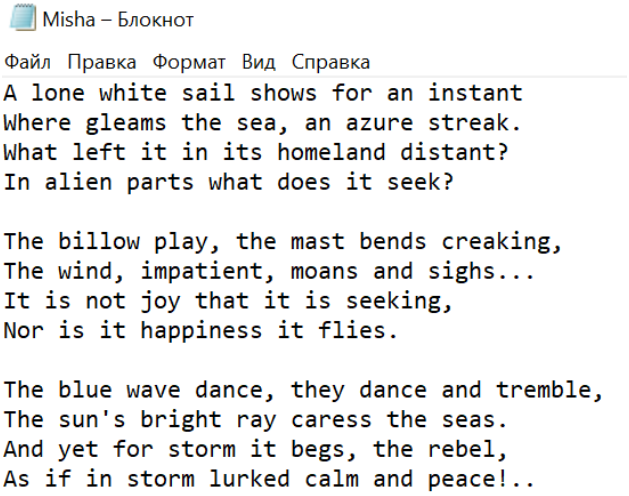


Рисунок 7. Исходный файл

Рисунок 8. Зашифрованный файл

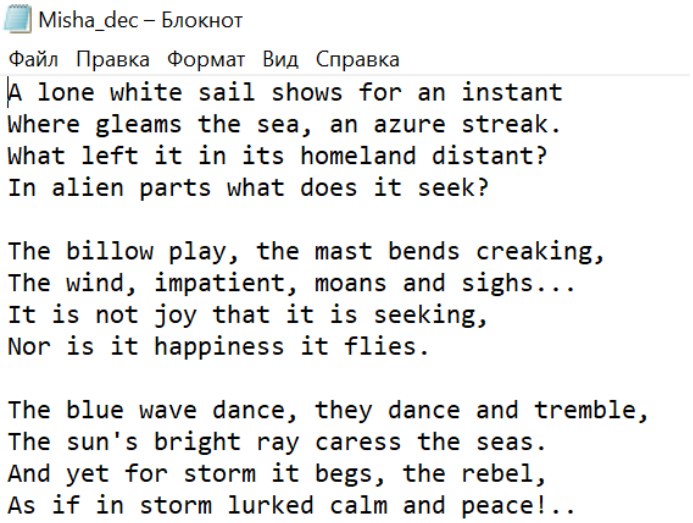


Рисунок 9. Расшифрованный файл

Шифрование и расшифрование файла BMP:

Сгенерированный ключ: 8LFJLDT8AX3WW212IIXQHLJ61CH1T1L0

Рисунок 10. Исходное изображение

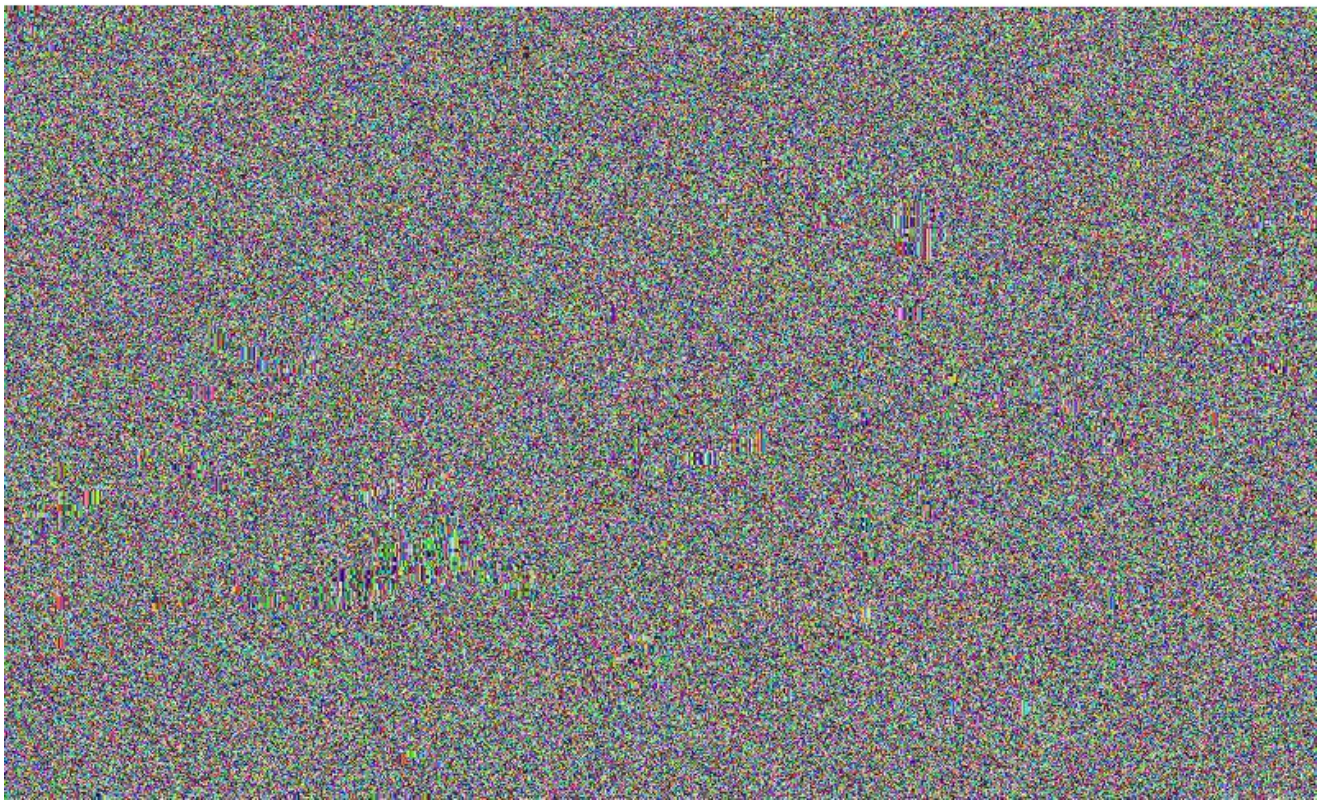


Рисунок 11. Зашифрованное изображение

Рисунок 12. Расшифрованное изображение

Если зашифрованный файл будет как-либо повреждён (рис. 13), то и дешифрованный файл будет отличаться от исходного (рис. 14):

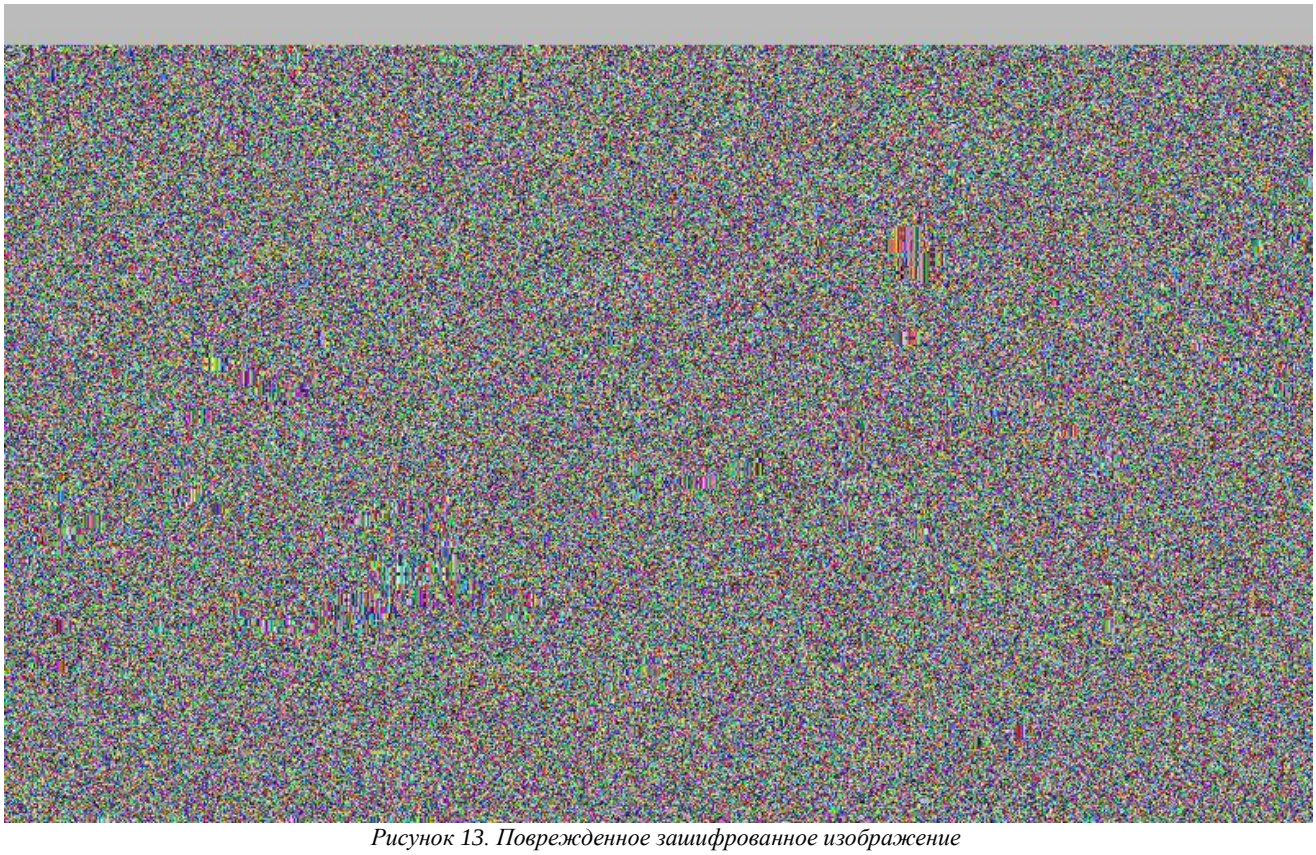


Рисунок 13. Поврежденное зашифрованное изображение

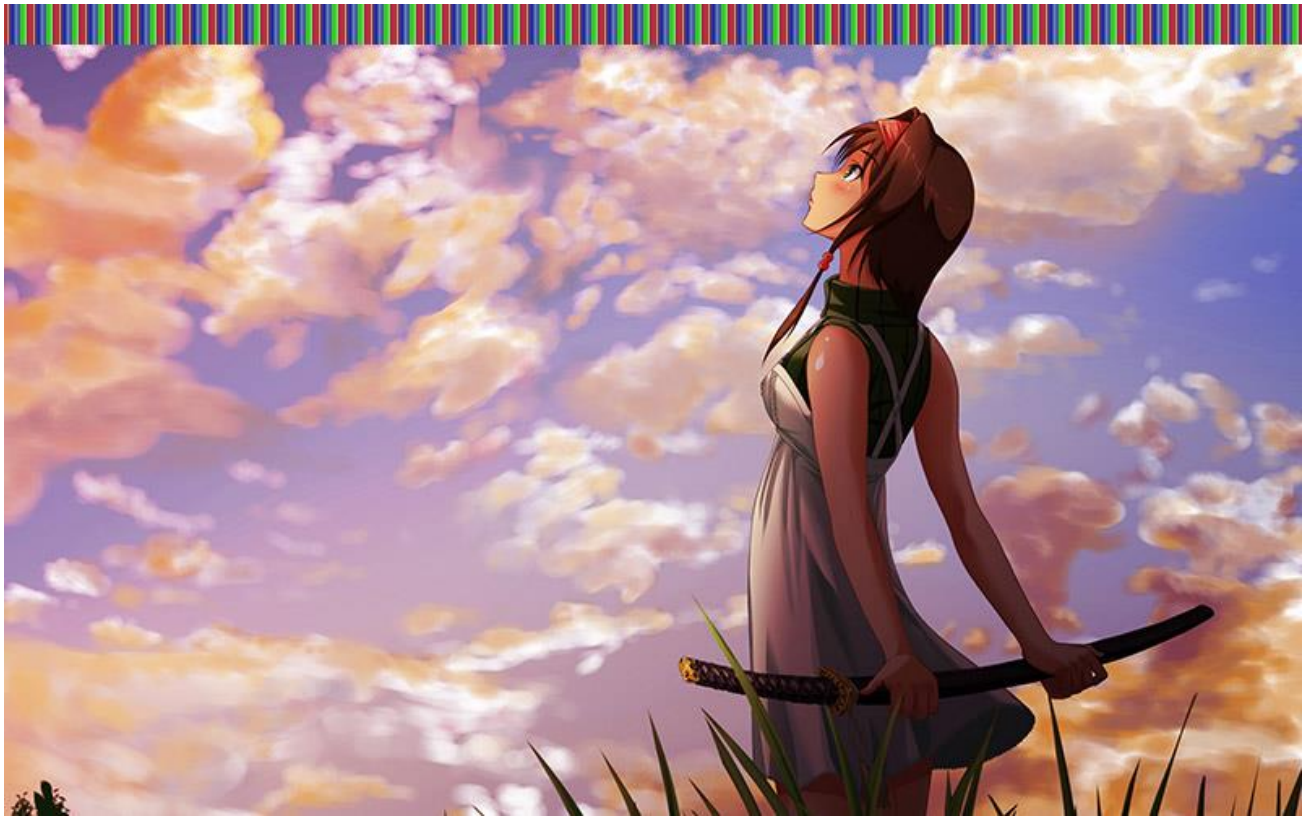


Рисунок 14. Поврежденное расшифрованное изображение

Twofish + OFB

Шифрование и расшифрование файла BMP:

Сгенерированный ключ: K2ZOQ2NUITPRCPLJ97RCJAY94GL3QH8H

Сгенерированный вектор инициализации: K2ZOQ2NUITPRCPLJ

Исходное изображение представлено на рис.10.

Рисунок 15. Зашифрованное изображение

Результат расшифрования данного изображения аналогичен результату, предоставленному на рис.12.

Коэффициент корреляции для входного и выходного потока

Вычислим коэффициент корреляции для красной компоненты исходного и закодированного изображений.

Формула для подсчета коэффициента корреляции представлена ниже:

$$\hat{r}_{A,B} = \frac{\hat{M}[(A - \hat{M}[A])(B - \hat{M}[B])]}{\hat{\sigma_A}\hat{\sigma_B}},$$

где A и B – компоненты изображения;

$M[.]$ – оценка математического ожидания в соответствии с формулой:

$$\hat{M}[I^{(A)}] = \frac{1}{WH}\sum_{i=1}^{H}\sum_{j=1}^{W} I_{i,j}^{(A)},$$

σ_A и σ_B – оценки среднеквадратичного отклонения компонент A и B, вычисляемого по формуле:

$$\hat{\sigma_A} = \sqrt{\frac{1}{WH-1}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(I_{i,j}^{(A)} - \hat{M}[I^{(A)}]\right)^2}.$$

По результатам вычислений коэффициент корреляции для красной компоненты исходного и закодированного изображений составил 0.000436764.

Оценка распределения «0» и «1» в выходном потоке

Оценим распределение «0» и «1» на примере зашифрованного изображения, представленного на рисунке 11:

- 0 – 49.9933 %
- 1 – 50.0067 %

Выводы

Реализован алгоритм шифрования Twofish, а также режим шифрования OFB. Было выяснено, что использование режима шифрования OFB позволяет улучшить качество шифрования данных. Вычислен коэффициент корреляции. Оценено распределение «0» и «1» в выходном потоке.

Достоинства алгоритма:

- Twofish эффективно реализуем аппаратно и в условиях ограниченных ресурсов;
- Зашифровывание и расшифровывание в алгоритме Twofish практически идентичны;
- Поддержка расширения ключа «на лету» (лучшая по результатам конкурса AES);
- Несколько вариантов реализации позволяют оптимизировать алгоритм для конкретных применений.

Недостатки алгоритма:

- Сложность структуры алгоритма затрудняет его анализ;
- Сложная и медленная процедура расширения ключа;
- Относительно сложно защищается от атак по времени выполнения и потребляемой мощности;
- Распараллеливание вычислений при шифровании алгоритмом Twofish реализуемо с ограничениями.

Список литературы

1. **С.П., Панасенко.** *Алгоритмы шифрования. Специальный справочник.* Санкт-Петербург : БХВ-Петербург, 2009.
2. **А.Л., Чмора.** *Современная прикладная криптография.* б.м. : Гелиос АРВ, 2002.

Во всех таблицах, представленных в данном приложении, указаны шестнадцатеричные значения.

Таблица 1. Матрица M_1

01	EF	5B	5B
5B	EF	EF	01
EF	5B	01	EF
EF	01	EF	5B

Таблица 2. Матрица M_2

01	A4	55	87	5A	58	DB	9E
A4	56	82	F3	1E	C6	68	E5
02	A1	FC	C1	47	AE	3D	19
A4	55	87	5A	58	DB	9E	03

Таблица 3. Таблица замен для q_0

t_0	8	1	7	D	6	F	3	2	0	B	5	9	E	C	A	4
t_1	E	C	B	8	1	2	3	5	F	4	A	6	7	0	9	D
t_2	B	A	5	E	6	D	9	0	C	8	F	3	2	4	7	1
t_3	D	7	F	4	1	2	6	E	9	B	3	0	8	5	C	A

Таблица 4. Таблица замен для q_1

t_0	2	8	B	D	F	7	6	E	3	1	9	4	0	A	C	5
t_1	1	E	2	B	4	C	3	7	6	D	A	5	F	9	0	8
t_2	4	C	7	5	1	6	9	A	0	E	D	8	2	B	3	F
t_3	B	9	5	1	C	3	D	E	6	4	7	F	2	0	8	A