УДК 003.26.09

Сергиенко Елена Николаевна
к.ф.-м.н., доцент
Смакаев Анатолий Витальевич
Давыденко Денис Олегович
Стрябков Александр Васильевич
Сергеев Павел Алексеевич
студенты
Белгородский Государственный Технологический
Университет имени В. Г. Шухова
Белгородская область, Россия

УЛУЧШЕНИЕ СТЕГАНОГРАФИЧЕСКИХ МЕТОДОВ ПЕРЕДАЧИ СООБЩЕНИЙ С ПОМОЩЬЮ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ

Во все времена перед людьми остро стояла задача защиты информации от несанкционированного доступа. Основными методами решения этой задачи еще в древние времена стали криптография и стеганография. Криптография скрывает содержимое сообщения с помощью его шифрования, в то время как стеганография скрывает сам факт передачи информации [1].

В связи с бурным развитием вычислительной техники и новых каналов передачи информации появились и новые стеганографические методы, в основе которых лежат особенности представления информации в компьютерных файлах и вычислительных сетях. Именно в этой области оба способа сокрытия сообщения могут быть объединены и использованы для повышения эффективности защиты информации.

Наиболее вероятным путем объединения криптографических и стеганографических методов может быть использование случайных величин в стеганографии. Одной из ключевых задач криптографии является создание надежных генераторов псевдослучайных последовательностей чисел. И их можно применить для усовершенствования механизма скрытия сообщения.

За основу возьмем популярный стеганорафический алгоритм LSB. Его суть заключается в следующе: пусть, имеется 24-х битное изображение, каждый пиксель которого кодируется 3 байтами, и в них расположены значения каналов RGB. Изменение младшего бита в каком-либо канале отдельно взятого пикселя почти не скажется на изображении в целом, однако даст возможность передать сообщение, заменив этот бит битом из сообщения [3].

Потенциальному взломщику потребуется перебрать всевозможные значения шага, с помощью которого он сможет составить исходное сообщение. Для больших изображений этот процесс займет много времени, однако приведет к нужному результату в обозримое время [2]. Для обеспечения устойчивости к подобным атакам следует сделать выбор следующей координаты для записи данных случайным.

Для повышения криптостойкости факта сокрытия изображения воспользуемся генератором случайных чисел на основе алгоритма RC4.

Криптогенератор функционирует независимо от открытого текста. Генератор имеет подстановочную таблицу (S-бокс 8 x 8): S_0, S_1, . . ., S_{255}. Входами генератора являются замененные по подстановке числа от 0 до 255, и эта подстановка является функцией от ключа изменяемой длины. Генератор имеет два счетчика i и j, инициализируемых нулевым значением.

Для генерации случайного байта гаммы выполняются следующие операции:

$i = (i+1) \bmod 256$

$j = (j+S_i) \bmod 256$

$swap\ (S_i, S_j)$

$t = (S_i+S_j) \bmod 256$

$K = S_t$

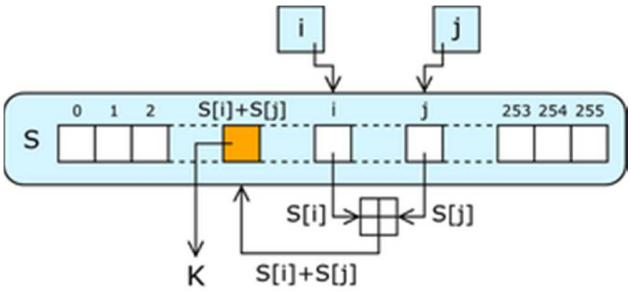

Рис. 1. Генератор ключевого потока RC4

Инициализация S-бокса столь же проста. На первом шаге он заполняется линейно:

$S_0 = 0, S_1 = 1, . . ., S_{255} = 255.$

Затем еще один 256-байтный массив полностью заполняется ключом, для чего ключ повторяется соответствующее число раз в зависимости от длины: K_0, K_1, . . ., K_{255}. Индекс j обнуляется. Затем:

for i=0 to 255

$j = (j+S_i+K_i) \bmod 256$

$swap\ (S_i , S_j)$

Схема показывает, что RC4 может принимать примерно 2^{1700} ($256! * 256^2$) возможных состояний и делает его пригодным для генерации псевдослучайной последовательности [4].

Для получения псевдослучайной последовательности требуются параметры n и k[], где n - число, необходимое для определения длины генерируемой последовательности ($l = 2^n$), а k[] – ключевая последовательность.

Пусть изображение имеет размер HxW, где H – высота, W – длина. Таким образом, количество пикселей в нем ImageSize = H*W.

Число n необходимо выбрать таким образом, чтобы выполнялось двойное неравенство: $2^n-1 < ImageSize < 2^n$.

При его соблюдении будет сгенерирована последовательность, позволяющая пронумеровать все пиксели изображения. Позиция пикселя будет определятся по следующему правилу:

x = Seq[i] mod W

y = Seq[i] div W

где Seq[i] – элемент сгенерированной последовательности.

Таким образом, выбрав n = log2(ImageSize)+1 и взяв заданную пользователем ключевую последовательность, мы сгенерируем псевдослучайную последовательность, которую будем использовать для записи сообщения в изображение. Выбор позиции следующего пикселя определяется следующим образом:

1. *t = Seq[count], где Seq[count] – очередной элемент сгенерированной последовательности.*
2. *Пока x≥ImageSize,*
   count++
   t = Seq[count]
3. *x = t mod WIDTH*
   y = t div WIDTH
   count++

Получив позицию пикселя, мы записываем в младший бит необходимое значение и повторяем процедуру, пока не запишем всю необходимую информацию.

Ключом к шифру будет являться заданная пользователем ключевая последовательность.

При дешифровании необходимо будет повторить генерацию последовательности и выполнить собрать сообщение из бит, записанных в пиксели изображения.

Таким образом можно записать сообщение в изображение-контейнер, гарантируя, что оно не может быть расшифровано в течении приемлемого для злоумышленника времени.

Тот же метод можно применить и для записи графической информации. Можно воспользоваться тем фактом, что на диске она хранится как простая последовательность битов, логически ничем не отличающаяся от любых других файлов, в том числе и текстовых [6]. Вследствие этого для применения методов стеганографии к графическим файлам необходимо следующее:

1. Побайтно считывать графические файлы
2. Каждый считанный байт записывать согласно правилам для записи отдельного символа, то есть:
   - 2.1. Получить позицию пикселя, в который необходимо записывать информационный бит на данном этапе
   - 2.2. Записать информационный бит в младший бит красного канала выбранного пикселя.

При дешифровании необходимо будет каждый выделенный из сообщения байт записать в файл-приемник.

В ходе работы были написаны приложения для стеганографической записи текстовой или графической информации в изображение, а так же для внесения искажений в изображение-контейнер.

Эксперимент 1

Запишем в графическое изображение-контейнер следующий текст:

Атака завтра в 15:00

Для распределения битов сообщения по пикселям используем псевдослучайную последовательность, полученную с помощью RC4 с ключом k = [3, 2, 4]. Сохраним ключ, содержащий в себе ключевую последовательность и длину сообщения в файле.

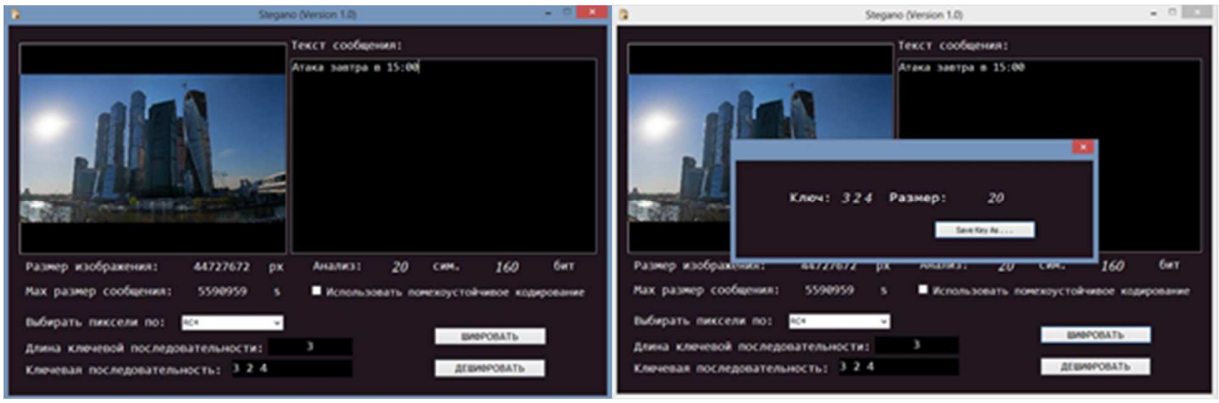

Рис. 2. Запись текстовой информации в изображение-контейнер (слева) и сохранение ключа (справа)

Далее внесем помехи в изображение-контейнер с уже записанным в него сообщением и дешифруем, используя ключ, содержащийся в ранее сохраненном файле.

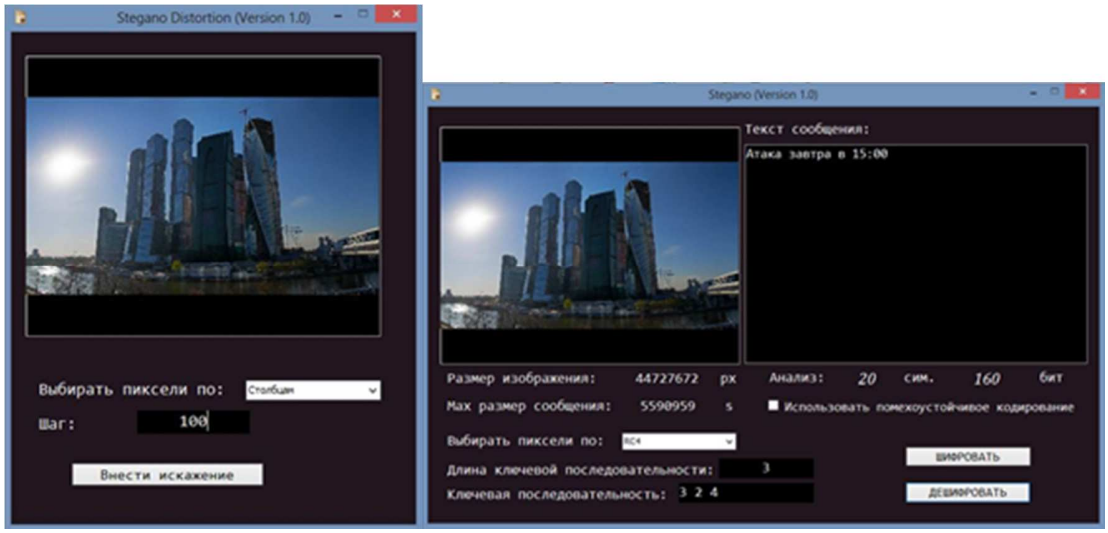

Рис. 3. Внесение искажений в изображение-контейнер (слева) и извлечение сообщения (справа)

Как видно, использование помехоустойчивого кодирования позволило противостоять внесенным искажения, таким образом на выходе получился исходный текст.

Эксперимент 2

Запишем в изображение-контейнер графическую информацию – изображение-сообщение.

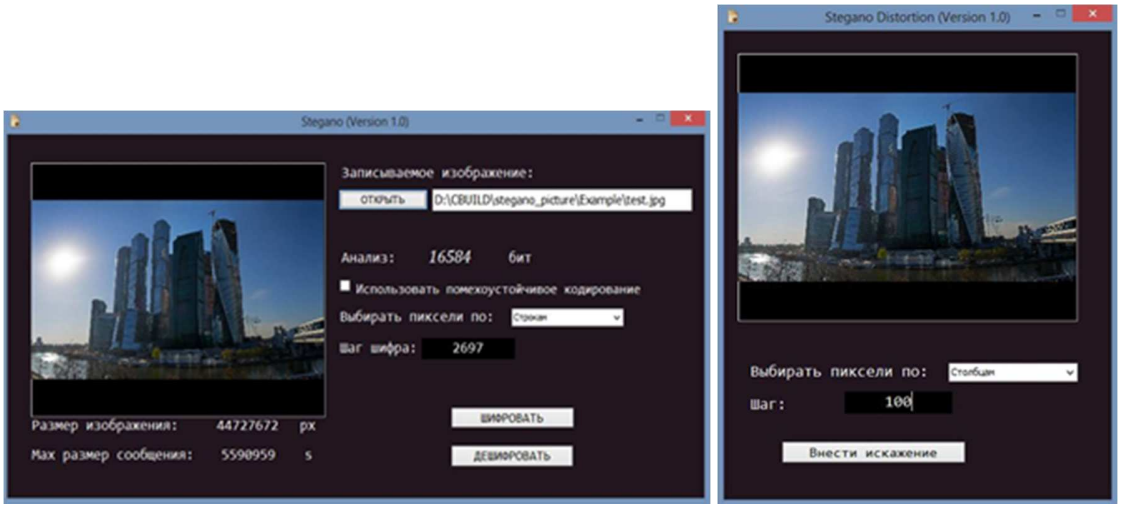

Рис. 4. Запись графической информации в изображение-контейнер (слева) и внесение в него искажений (справа)

Сохраним ключ, а затем загрузим в программу изображение-контейнер с уже встроенным сообщением и дешифруем его. Если в ходе дешифрования и декодирования последовательности байтов скрываемого изображения обнаружится, что не был поврежден заголовок BMP-файла, то у пользователя будет возможность просмотреть изображение с помощью стандартных средств операционной системы. При удачном исходе эксперимента на выходе получим следующую картинку, которая может быть незначительно искажена, как в данном случае:

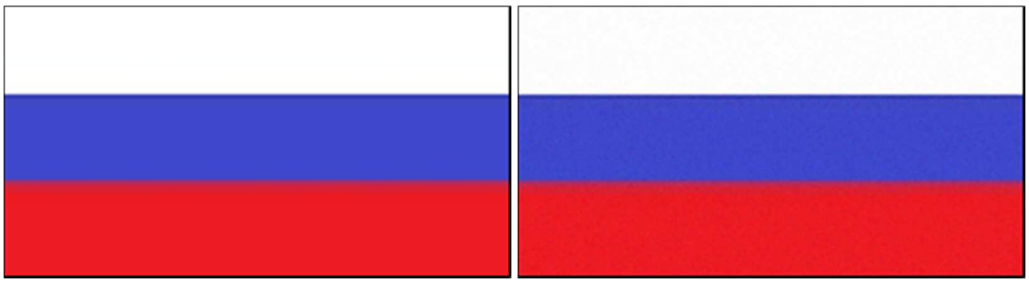

Рис. 5. Оригинальное изображение-сообщение (слева) и восстановленное изображение-сообщение (справа)

Таким образом, удалось успешно поместить и извлечь из изображения-контейнера не только текст, но и графическую информацию. Следовательно, методы стеганографии и криптографии применимы для совместного использования с целью лучшего сокрытия информации в графических изображениях с помощью применения криптографических генераторов псевдослучайных последовательностей чисел при записи сообщения в графический файл.

Список литературы:

1. *Саймон Сингх*. Книга шифров. Тайная история шифров и их расшифровки / АСТ, Астрель, 2007 г. — 446 с.

2. *Скляров Д.В.* Искусство защиты и взлома информации / СПб.: БХВ-Петербург, 2004. — 288 с.
3. *Грибунин В. Г., Оков И. Н., Туринцев И. В.* Цифровая стеганография / М.: Солон-Пресс, 2002. — 272 с.
4. *Венбо Мао.* Современная криптография: теория и практика / М.: Издательский дом «Вильямс», 2005. — 768 с.
5. *Менезис А., Пол ван Оорсхот, Ванстоун С.* Прикладная криптография / CRC-Press, 1996. — 816 с.
6. *Фергюсон Н., Шнаер Б.* Практическая криптография / М.: Вильямс, 2004. — 432 с.