УДК 621.391

С.Г. Рассомахин, Ф.А. Борщов

Харьковский национальный университет имени В.Н. Каразина, Харьков

УНИВЕРСАЛЬНЫЙ МНОГОКЛЮЧЕВОЙ АЛГОРИТМ ПОЛУЧЕНИЯ СТЕГАНОГРАММ

Разработан алгоритм, использующий встраивание данных методом наименее значащего бита, применимый для встраивания данных произвольного формата равномерно по контейнеру с использованием нескольких ключей шифрования и псевдослучайной перестановки бит исходного сообщения.

Ключевые слова: *метод наименее значащего бита, оценка помех в контейнере, метод LSB, шифрование, псевдослучайная перестановка бит.*

Введение

Информационная безопасность играет важную роль в современном техногенном обществе. Криптография и стеганография являются важными средствами обеспечения задач безопасности, хотя применяют различные подходы для достижения этой цели. Объединение этих подходов дает мощную базу для создания методов защиты информации. Компьютерная стеганография, предназначенная для маскирования самого факта наличия конфиденциальных данных, дает множество способов сокрытия информации. Объединение возможностей стегосистем и криптографических методов преобразования данных дает возможность обеспечить более высокий уровень защиты скрываемого сообщения от возможных угроз информационной безопасности.

Анализ литературы. Самым распространенным методом защиты информации на сегодня является метод криптографической защиты. Суть метода состоит в скрытии содержания сообщения за счет его шифрования, при использовании определенного алгоритма и ключа. Для тех, кто не осведомлен о работе алгоритма или не имеет доступа к ключу дешифрования, получение доступа к данным представляет задачу поиска ключа одним из известных способов криптоанализа. Если используется сертифицированный алгоритм шифрования, то для доступа к данным требуются существенно большие временные и вычислительные затраты. Применение методов стеганографических преобразований эквивалентно введению дополнительной степени защиты данных от несанкционированного доступа и улучшает характеристики стойкости [1 – 3].

Зашифрованная с помощью криптографических преобразований информация является недоступной для ознакомления без знания актуального ключа (на протяжении времени, которое определяется стойкостью криптосистемы). Стеганографические преобразования снижают вероятность обнаружения самого факта наличия криптограмм. Стеганографическая защита обеспечивает скрытие факта существования конфиденциальных сведений при хранении, обработке, передаче или манипуляциях с данными. То есть данная защита ставит под собой задачу не только невозможность выявить наличие скрытых данных в исходном сообщении (контейнере), но и не вызвать подозрений о наличии скрытых данных вообще. Задача неавторизированного извлечения данных из контейнера решается путем использования криптографических методов. В общем виде, при использовании стеганографических методов, секретное сообщение встраивается в контейнер, в качестве контейнера может выступать текст, аудио, видео, картинка, либо иной файл, который в открытом виде передается адресату.

Можно выделить две причины возрастания популярности применения методов стеганографии в настоящее время [1]: ограничение на использование криптосредств в ряде стран мира и появление проблемы защиты прав собственности на информацию, представленную в цифровом виде. Первая причина повлекла за собой большое количество разработок в духе классической стеганографии (то есть скрытия факта передачи информации), вторая – еще более многочисленные работы в области так называемых водяных знаков. Цифровой водяной знак (ЦВЗ) – специальная метка, незаметно внедряемая в изображение или другой сигнал с целью тем или иным образом контролировать его использование.

Стеганографические методы применяется в следующих областях:

- защита от копирования;
- аутентификация;
- скрытая аннотация документов;
- скрытая связь.

Основой для применения различных методов построения стегосистем является естественная избыточность большинства видов информационных сообщений, при устранении которой (за счет введения дополнительных скрываемых данных) не происходит заметных изменений качества исходных переносчиков (контейнеров). Наиболее эффективны методы стеганографии, применяемые для встраивания маскируемых дан-

© С.Г. Рассомахин, Ф.А. Борщов

ных в файлы мультимедиа: аудио, статические и динамические изображения. При этом хороший результат достигается при использовании сравнительно простого семейства алгоритмов стегопреобразований, основанных на подмене значений наименее значащих битов (LSB [1,2}) сообщений-контейнеров.

Целью статьи является разработка модифицированного универсального алгоритма встраивания данных методом наименее значащего бита. Под универсальностью подразумевается возможность встраивания секретного сообщения любого формата в любой контейнер несжатого формата.

Основная часть

Стеганографические преобразования, предложенные в данной работе, используют алгоритм встраивания данных на основе замены наименее значащего бита (LSB) в кванте данных контейнера. Под квантом данных подразумевается объем информации, требуемой для хранения одной величины данных в контейнере. Например, для картинки с глубиной цвета в 24 бита, квант данных будет равняться 8 битам, так как каждый пиксель состоит из трех субпикселей, отвечающих за красный, зеленый и синий каналы по восемь бит соответственно. Для звукового файла, квант данных будет равняться разрядности частоты квантования при оцифровке сигнала. Для описания алгоритма работы необходимо ввести основные определения и обозначения:

- **CSize** – величина контейнера, т.е. размер исходного контейнера-переносчика, выраженный в количестве байт.
- **SSize** – величина секретного сообщения, встраиваемого в контейнер, выраженная в битах.
- **QBits** – количество бит на квант данных контейнера.
- **byteOffset** – байтовый отступ;
- **dataOffset** – отступ при записи данных;
- **metaOffset** – отступ между метаданными и данными скрываемого сообщения;
- **randArray** – массив целых псевдослучайных чисел.

Параметр **byteOffset** является основным для корректного построения алгоритма рассматриваемого стеганографического преобразования. Алгоритм предусматривает равномерное распределение элементов скрываемого сообщения по объему контейнера-переносчика. Параметр **byteOffset** указывает количество байт, которое необходимо пропустить перед записью каждого последующего бита в квант контейнера. Рассмотрим реализацию предлагаемого алгоритма. При инициализации процедуры построения стеганограммы вычисляется начальное значение байтового отступа **byteOffset**. Это осуществляется следующим образом:

$$\text{byteOffset} = \frac{\text{CSize}}{\text{SSize}\cdot 8} - \left(\frac{\text{CSize}}{\text{SSize}\cdot 8} \bmod \left(\frac{\text{QBits}}{8} \right) \right). \quad (1)$$

Запись данных можно выполнять и указав байтовый отступ вручную, однако вычисление величины **byteOffset** в соответствии с выражением (1) обеспечивает минимизацию мощности дополнительных шумов в контейнере после записи в него скрываемого сообщения. Задача алгоритма заключается в наиболее рациональном размещении внутри контейнера метаданных и самого скрываемого сообщения. Под метаданными понимается пакет данных, которые необходимы для извлечения сообщения из сформированного контейнера. Метаданные состоят из двух параметров – длины сообщения в байтах и функции инициализации генератора псевдослучайных чисел. Алгоритм использует функцию инициализации генератора псевдослучайных чисел, называемую в дальнейшем зерном **Seed** для формирования псевдослучайной последовательности (ПСП), которая будет использоваться для встраивания и считывания бит дополнительного сообщения в контейнер. Это необходимо для обеспечения псевдослучайного расположения бит скрываемого сообщения в пределах контейнера.

Определение длины скрываемого сообщения **SSize** и значения зерна **Seed** генератора псевдослучайных чисел позволяет сформировать пакет метаданных. Для построения ключевой стегосистемы полученный пакет шифруется при помощи криптографического алгоритма. В контексте данной работы дли криптографического преобразования пакета метаданных и собственно скрываемых данных предполагается применение алгоритма AES-256 [4]. Ключи шифрования метаданных и сообщения могут быть идентичными, однако алгоритм разрешает использование разных ключей, что повышает стойкость системы. Предположим, что для шифрования метаданных используется ключ **metaEncryptionKey**. Если размер ключа составляет менее 256 бит, то, с помощью алгоритма SHA-256 [5], вычисляется его хэш-функция, которая используется в качестве ключа шифрования метаданных. Запись в контейнер должна производиться с учетом отступа от начала файла, чтобы не повредить его заголовок. Отступ можно выбрать произвольно, однако его величина не может быть меньше длины стандартного заголовка файлов медиа данных, представленных в несжатых форматах. Пусть значение отступа составляет величину **headerOffset**. Это значение должно быть фиксированным и известным отправителю и получателю стеганограммы. Вслед за отступом **headerOffset** размещается пакет метаданных. Один бит метаданных помещается в один байт контейнера. При этом метаданные в контейнере зашифрованы, их размер фиксирован и составляет величину **metaSize**. В рассматриваемой реализации алгоритма – это две целочисленные величины, занимающие в двоичном представлении по 64 бита. Эти величины используются при извлечении данных из контейнера для определения объема метаданных. Скрываемое сообщение перед помещением в контейнер, также как метаданные,

шифруется по алгоритмуAES-256. Метод предоставляет возможность использования уникального ключа **messageEncryptionKey** для осуществления этого шифрования. При этом, если размерность ключа составляет менее 256 бит, то вычисляется его хэш-функция по алгоритмуSHA-256 [5] для использования её в качестве ключа шифрования данных сообщения. Запись данных в контейнер выполняется с отступом в величину **dataOffset**, вычисляемую следующим образом:

$$\text{dataOffset} = \\ = \text{headerOffset} + \text{byteOffset} \cdot \text{metaSize} + \text{metaOffset}, \quad (2)$$

где **metaOffset** –отступ от записанных метаданных в контейнере.

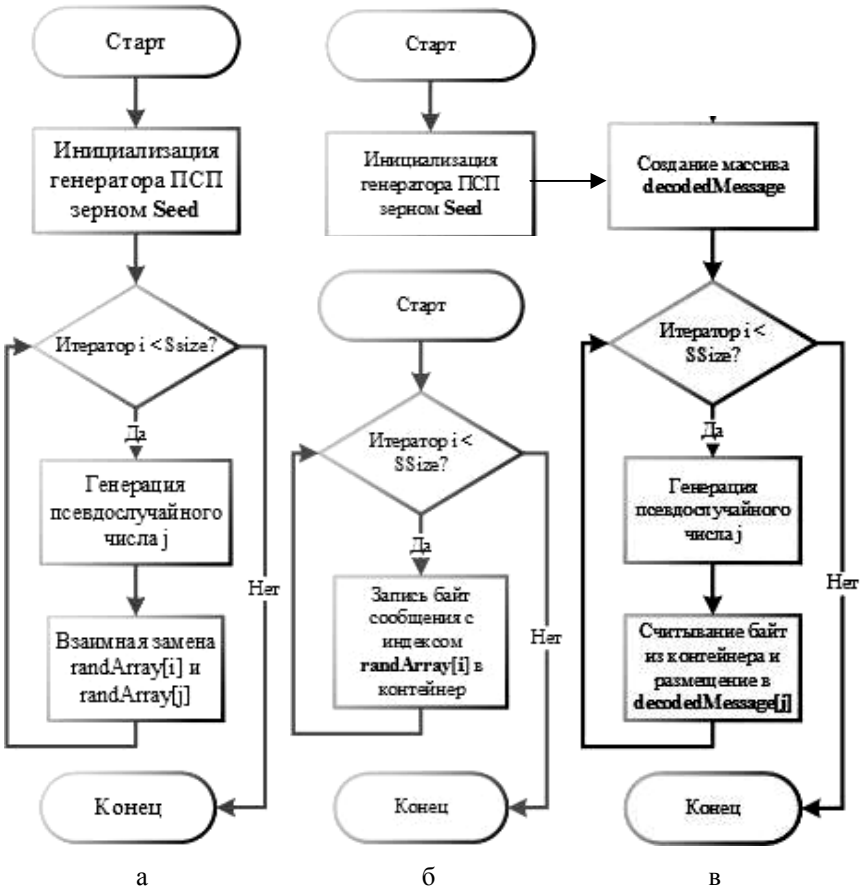
Схема алгоритма получения массива псевдослучайных чисел представлена на рис. 1, а. Алгоритм скрытия данных в контейнере заключается в следующем. Перед записью скрываемого сообщения формируется массив целых чисел **randArray** величиной в **SSize**. Данный массив требуется для размещения байт секретного сообщения равномерно по контейнеру. Числа в массиве **randArray** не должны повторяться. Предположим, что они расположены в порядке возрастания. Для получения массива генератор псевдослучайных чисел инициализируется с помощью зерна **Seed**. Генератор по случайному закону выбирает целые числа из диапазона от 0 до (**SSize – 1)**.

Далее выполняется цикл вычислений с количеством итераций **SSize**, итератором **i** с шагом 1, причем на каждом шаге генерируется псевдослучайное число **j**. В конце каждой итерации выполняется замена местами чисел с индексами i и j в формируемом массиве.

В результате получается массив **randArray** с перемешанными целыми числами.

Знание зерна инициализации **Seed** позволяет восстановить данный массив. Это является ключевым моментом, при извлечении сообщения из контейнера. Далее, с использованием известного отступа **dataOffset,** производится запись скрываемого сообщения в контейнер. Запись данных происходит на основании содержимого массива **randArray**, который читается в лексикографическом порядке. Размеры сообщения и массива **randArray** – идентичны. Каждый элемент массива является номером байта скрываемого сообщения. Байты сообщения помещаются в контейнер, причем каждый бит размещается равномерно по контейнеру с отступом в **byteOffset.**

Схема алгоритма записи байт скрываемого сообщения в контейнер изображена на рис. 1, б. В результате формирования стеганограммы получается контейнер с записанными метаданными в виде размера скрытого сообщения в байтах и зерна генератора ПСП, а также собственно скрываемым сообщением.



а б в

Рис. 1. Блок-схемы алгоритмов: а – получения массива **randArray**; б – записи байт; в – извлечения байт сообщения из контейнера

Метаданные и сообщение зашифрованы алгоритмом AES-256 с использованием двух различных ключей: **metaEncryptionKey** и **messageEncryptionKey**.

Скрываемое сообщение оказывается равномерно фрагментированным по объему контейнера при помощи генератора ПСП, инициализированного параметром **Seed**. Структура получаемого контейнера, учитывающая скрытое сообщение, представлена на рис. 2, при этом заштрихованные области соответствуют криптографически преобразованным областям контейнера.

Распаковка стеганограммы производится по следующему алгоритму. При осуществлении обратного преобразования для извлечения скрытой информации используются значения одного или двух ключей дешифрования: **metaDecryptionKey** и/или **messageDecryptionKey**.

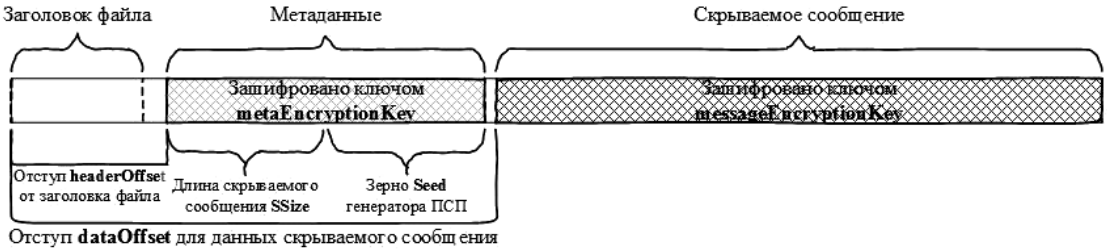


Рис. 2. Структура контейнера после записи секретного сообщения

При реализации симметричной системы подразумевается:

metaDecryptionKey = metaEncryptionKey;
messageDecryptionKey = messageEncryptionKey.

Известные параметры **headerOffset** и **metaDecryptionKey**дают возможность извлечения и дешифрования метаданных из контейнера. В результате становятся доступными два значения – **SSize** и **Seed**, что дает возможность вычислить величину **byteOffset** и определить байтовый отступ для извлечения скрытых данных.

Кроме того, по формуле (2) также вычисляется значение **dataOffset**. Известные параметры **dataOffset**, **byteOffset**, **Seed** и **dataDecryptionKey** позволяют полностью извлечь и дешифровать скрытое сообщение контейнера. Также как и в процессе формирования стеганограммы, генерируется массив целых чисел **randArray** размером **SSize**. При этом инициализация генератора псевдослучайных последовательностей производится зерном **Seed**. При помощи алгоритма сбора фрагментов создается массив **decodedMessage** размером **SSize**, для этого используются значения отступов **dataOffset** и **byteOffset**. Каждый извлеченный байт сообщения размещается в массиве **decodedMessage** на позицию, определяемую очередным индексом из **randArray**. Схема описанного алгоритма извлечения сообщения представлена на рис.1, в. Процесс извлечения скрытого сообщения из контейнера завершается дешифрованием с ключом **messageDecryptionKey**.

Выводы

Разработанный метод формирования и обработки стеганограмм является универсальным относительно выбора типа встраиваемого сообщения, так как оперирует с битовой структурой файла на объектно-ориентированном языке высокого уровня. В качестве контейнера может использоваться любой файл несжатого формата, поскольку сам метод LSB функционально ориентирован только на не преобразуемые контейнеры.

Рассмотренный алгоритм является открытым и поддерживает возможность модификации размера и типа метаданных. Модульность алгоритма позволяет использовать различные методы шифрования и генераторы псевдослучайных последовательностей. Программная реализация разработанного метода была выполнена на языке программирования Objective-C с использованием среды разработки XCode.

Список литературы

1. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – К: МК-Пресс, 2006. – 288 с.

2. Грибунин В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. – М.: Солон-Пресс, 2009. – 265 с.

3. Рябко Б.Я. Основы современной криптографии и стеганографии : монография / Б.Я. Рябко, А.Н. Фионов. – М. : Горячая линия-Телеком, 2013. – 232 с.

4. Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, Advanced Encryption Standard (AES) (FIPS PUB 197) [Электронный ресурс]: – Режим доступа к ресурсу: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

5. Secure Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, Secure Hash Standard (SHS) (FIPS PUB 180-4). – [Электронный ресурс]: – Режим доступа к ресурсу: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

Поступила в редколлегию 1.06.2016

Рецензент: д-р техн. наук, проф. В.А. Краснобаев, Харьковский национальный университет имени В.Н. Каразина, Харьков.

УНІВЕРСАЛЬНИЙ БАГАТОКЛЮЧОВИЙ АЛГОРИТМ ОТРИМАННЯ СТЕГАНОГРАМ

С.Г. Рассомахін, Ф.А. Борщов

Розроблено алгоритм, який використовує вбудовування даних методом найменш значущого біту, який є чинним для вбудовування даних довільного формату рівномірно по контейнеру з використанням декількох ключів шифрування та псевдовипадкової перестановки біт повідомлення, що приховується.

Ключові слова: *метод найменш значущого біта, оцінка перешкод в контейнері, метод LSB, шифрування, псевдовипадкова перестановка бітів.*

UNIVERSAL MULTIKEY ALGORITHM OF STEGANOGRAM OBTAINING

S.G. Rassomakhin, F.A. Borshchov

Developed an algorithm, which utilizes least significant bit data insertion method, that applies for data of random format insertion uniformly in container with multiple encryption key utilizing and pseudorandom permutation of secret messages' bits.

Keywords: *method of the least meaning bit, estimation of hindrances in a container, method of LSB, encipherement, pseudocasual transposition of bits.*