УДК 004.056.55:003.26

РЕАЛИЗАЦИЯ МЕТОДА LSB-R С ПРИМЕНЕНИЕМ ГПСЧ И ИССЛЕДОВАНИЕ НА СТЕГАНОГРАФИЧЕСКУЮ СТОЙКОСТЬ

Голоднов Е.А., Мыздриков Н.Е., Черкесова Л.В., Сафарьян О.А., Ревякина Е.А.

ФГБОУ ВО «Донской государственный технический университет», Ростов-на-Дону, e-mail: golodnov@spark-mail.ru

В данной статье рассмотрены стеганографические алгоритмы LSB-R и LSB-M сокрытия информации в медиафайлах. Данные алгоритмы позволяют программно записать достаточно много информации в контейнер, но за счет хранения скрытой информации в явном виде ее достаточно легко обнаружить в ходе проведения стегоанализа. С этой целью было предложено использовать ГПСЧ, с помощью которого можно выбирать, в каких битах контейнера можно записывать сообщение, при этом при минимальных потерях скорости работы. Поскольку аддитивный ГПСЧ дает аномалии в генерируемых последовательностях, было предложено модифицировать его и на его основе реализовать LSB-R метод внедрения информации в контейнер. Реализация показала, что при небольших потерях в скорости сокрытия информации, контейнер стал более стегостойким, что подтверждается статистическими тестами и тестами программ, определяющих наличие или отсутствие в контейнере скрытой информации. Также к достоинствам данного метода можно отнести то, что достаточно легко варьировать рейт внедрения скрытой информации в контейнер, что позволяет при постоянном размере контейнера скрывать информацию более эффективно, распределяя её по всему контейнеру.

Ключевые слова: стеганографический анализ, рейт внедрения, ГПСЧ, стеганостойкость, контейнер, расширенный алгоритм Евклида

IMPLEMENTATION OF THE LSB-R METHOD USING PRNG AND RESEARCH ON STEGANOGRAPHIC RESISTANCE

Golodnov E.A., Myzdrikov N.E., Cherkesova L.V., Safaryan O.A., Revyakina E.A.

Federal State Budgetary Institution of Higher Education Don State Technical University, Rostov-on-Don, e-mail: golodnov@spark-mail.ru

This article discusses the steganographic LSB-R and LSB-M hiding algorithms for information in media files. These algorithms allow you to write a lot of information in the container, but due to the storage of hidden information in an explicit form, it is quite easy to detect it during the steganalysis. For this purpose, it was proposed to use the PRNG with which it is possible to choose in which bits of the container a message can be recorded, while at the same time minimizing the loss of work speed. Since the additive PRNG gives anomalies in the generated sequences, it was proposed to modify it and, on its basis, implement the LSB-R method of introducing information into the container. The implementation showed that with a small loss in the speed of information hiding, the container became more resistant, as evidenced by statistical tests and tests of programs determining the presence or absence of hidden information in the container. Also, the advantages of this method include the fact that it is quite easy to vary the rate of insertion of hidden information into a container, which allows for a constant container size to hide information more effectively, distributing it throughout the container.

Keywords: steganographic analysis, rate of implementation, PRNG, stegane resistance, container, advanced Euclidean algorithm

На сегодняшний день, несмотря на совершенствование технологий шифрования, все еще остается актуальной проблема скрытой передачи информации. Помимо того, что разрабатываются новые методы встраивания информации, немалое внимание научное сообщество уделяет совершенствованию уже известных методов сокрытия информации программными средствами, использующими математические модели.

Сокрытие информации в сравнении с шифрованием имеет ряд преимуществ. Во-первых, скорость извлечения информации из контейнера в сравнении с расшифровкой сообщения, зашифрованного стойким шифром, намного выше. На небольших массивах информации это может быть и незаметно, но разница становится ощутимее с увеличением объемов информации. Также несомненным преимуществом является то, что факт передачи сообщения со скрытой информацией может попросту не привлечь к себе внимание, в то время как хаотичный набор битов заинтересует злоумышленника.

Для увеличения скрытности информации в контейнере можно использовать стеганографические методы в совокупности с криптографическими, и реализовывать это в виде отдельного программного средства: это позволит избежать определения формата скрытно передаваемых данных по первичным признакам форматов: структуре заголовочных файлов.

Исходя из вышесказанного, необходимо модифицировать алгоритм LSB-R, создать его программную реализацию и подтвердить превосходство модификации над ори-

гинальным вариантом в плане стеганографической стойкости.

Помимо достоинств, использование LSB-R метода вместе с ГПСЧ имеет некоторые недостатки. Любая программная реализация псевдослучайного генератора имеет «плохой» набор стартовых значений, которые дадут слишком короткую уникальную последовательность. Поэтому необходимо заранее заготовить такие наборы, которые гарантированно дадут хорошую последовательность, с которой можно работать [1].

Случайность появления битов, несущих информационную нагрузку, позволит увеличить стеганографическую стойкость контейнера. Для проведения исследования будет проведено три теста для картинки, используемой в качестве контейнера: архивирование контейнера, RS-тест и Sample Pair attack.

Стеганография

Под стеганографией подразумевают науку, занимающуюся передачей или хранением информации с учетом сокрытия самого факта такой передачи или хранения. Наиболее актуальной сейчас является цифровая стеганография – отрасль стеганографии, занимающаяся сокрытием информации в цифровых медиаконтейнерах (аудио/видеофайлы, изображения) [2, 3].

Наиболее популярным методом сокрытия информации является LSB-метод. Выделяют 2 метода внедрения: LSB-R и LSB-M.

LSB-R метод состоит в простой замене наименее значащих битов яркости цветовой компоненты пикселя на информационный бит. Таким образом, в одном пикселе изображения при стандартной работе алгоритма мы сможем сохранить 3 бита. В случае с LSB-M идет не простая замена наименее значащего бита, а прибавление или вычитание единицы от байта компоненты цвета. Такая модификация предназначена для того, чтобы обходить автоматические проверки контейнеров на наличие в них скрытой информации.

Достоинством LSB-алгоритмов является высокая скорость работы алгоритмов внесения и извлечения информации, а также достаточно простая программная реализация. Однако информация хранится внутри контейнера в открытом виде, и, как следствие, атаки на такого рода внедрения проводятся достаточно просто: достаточно считать последние наименее значащие биты и попытаться сформировать из них единый файл, который будет иметь явную структуру, которая даст понять, что это за файл и каким образом его можно прочитать [4].

Для увеличения стеганографической стойкости данного рода алгоритмов авторами работы было предложено внедрять информацию не в каждый наименее значащий бит. Для этого было решено программно реализовать ГПСЧ, который позволит выбирать информационные биты псевдослучайно, что усложнит поиск информации внутри контейнера аналитику.

Генератор псевдослучайных чисел

Генератор псевдослучайных чисел – детерминированный алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются равномерному распределению.

К идеальной модели ГПЧ предъявляют требования, которые гарантируют его корректную работу:

– длинный период, гарантирующий отсутствие зацикливания последовательности;

– быстрота работы алгоритма и малые затраты памяти;

– возможность заново воспроизвести ранее сгенерированную последовательность;

– одинаковое функционирование на различном оборудовании и ОС.

В качестве ГПСЧ было предложено использовать модифицированный аддитивный ГПСЧ, в котором присутствуют принципы инверсно-конгруэнтного метода. Каждый последующий член последовательности формируется по следующей рекуррентной формуле.

$$X_n = a * X_{n-4}^{-1} + X_{n-3} + b * X_{n-2}^{-1} + X_{n-1}.$$

Таким образом, для генерации последовательности требуется задать начальные значения последовательности и два коэффициента, значения которых остаются неизменными в течение генерации всей последовательности. Для достижения наилучшего результата работы ГПСЧ требуются взаимно простые коэффициенты. Генерация каждого последующего числа требует совокупности нескольких предыдущих членов. Отсюда следует вывод, что последующие результаты формулы зависят от предыдущих результатов. Следовательно, восстановить всю последовательность можно из нескольких подряд идущих чисел, что безусловно делает такой ГПСЧ уязвимым к атакам.

Однако если принять во внимание тот факт, что псевдослучайная последовательность явно не отображена, она лишь указывает на положение измененных битов, таким образом восстановление ее значений становится очень сложной задачей. Также так называемый «белый шум» наименее

значащих битов, который присутствует в любых графических изображениях, полученных с помощью фотокамеры, позволяет замаскировать изменения в цифровом контейнере.

Стеганографическая атака путём архивирования контейнера

При изменении наименее значащих битов контейнера в изображении появляются пиксели новых цветов. Вследствие этого палитра изображения становится больше, чем было изначально. Такое изображение становится сложнее сжать, и именно из-за этого архив, содержащий заполненный контейнер, имеет больший объем в сравнении с тем же пустым контейнером.

Алгоритмы архивации работают на коде Хаффмана: наиболее часто встречающимся последовательностям бит внутри файла ставится в соответствие наиболее короткое кодовое значение, и наоборот: чем реже встречается та или иная последовательность битов, тем длиннее будет ее кодовое значение [5].

В заголовке архива находится таблица соответствия информационных и кодовых последовательностей.

За счет того, что разрастается количество цветов, разрастается и видоизменяется заголовок архива. Из-за этого математическое ожидание кодовых последовательностей станет меньше и, как следствие, возрастет размер архива.

RS-тест

Суть теста заключается в том, что изображение разделяется на множество сгруппированных пикселей, для каждой группы применяется специальная процедура перестановки. На основании значения функции-дискриминанта до и после применения перестановки все группы делятся на регулярные, сингулярные и неиспользуемые подгруппы.

Алгоритм основывается на предположении, что количество регулярных и сингулярных групп пикселей в оригинальном изображении и в изображении после применения перестановки должно быть близким к эквивалентному соотношению. Если количество таких групп меняется в процессе применения перестановки, это значит, информационный контейнер исследуемого изображения является заполненным [6].

Этот метод атаки является вероятностным, это означает, что тест не может однозначно определить, является ли контейнер заполненным. Вместо этого результаты показывают процент контейнера, который предположительно является заполненным. Точность работы этого метода наиболее высока в случае обработки больших размеров контейнера. Однако главным недостатком такого теста можно считать то, что ему достаточно сложно анализировать сильно зашумленные контейнеры.

Sample pair тест

Цель теста – определить, есть ли статистическое доказательство того, что средняя разница между парными сегментами изображения по конкретному результату значительно отличается от нуля. Аналогично RS-тесту, Sample pair тест не может однозначно дать ответ, содержится ли скрытое сообщение в контейнере, он способен лишь определить, какой процент контейнера заполнен информацией [7]. Тест обладает теми же достоинствами и недостатками, однако его главное отличие заключается в меньших временных и машинных затратах, что делает этот тест предпочтительнее в случае быстрых проверок.

Реализация разработанного метода

Для реализации модифицированного метода LSB был выбран язык программирования C#. Далее будут приведены разработанные методы, которые необходимы для работы модифицированному методу LSB.

```
private IEnumerable<Pixel> CreatePixelArray()
{
   for (int i = 0; i < StartBitmap.Height; i++)
      for(int j = 0; j < StartBitmap.Width; j++)
         if (RNG.GetRandomNumber() <= contest)
            yield return new Pixel() { color = StartBitmap.GetPixel(j, i), x = j, y = i };
}
```

В этом методе создается массив пикселей, в который будет занесена информация.
Значение *contest* равно: *Поле ГПСЧ * Рейт внедрения*.

```
public int GetRandomNumber()
{
   var result = Math.Abs(Reverse(X1, Field) + A * X2 + Reverse(X3, Field) + B * X4)% Field;
   X1 = X2; X2 = X3; X3 = X4; X4 = result;
   return result;
}
```

Данный метод последовательно генерирует псевдослучайное число.

Результаты тестирования

Для анализа результативности предложенной модификации, реализованной в виде программного средства, следует провести тестирование. Все тесты были проведены с одинаковыми исходными данными для разных рейтов внедрения. Чтобы наглядно продемонстрировать результат, были составлены соответствующие таблицы результатов тестирования.

Тестирование методом SPA. По горизонтали расположены значения рейта внедрения информации, а по вертикали – два исследуемых метода, где LSB-RNG – предложенная модификация классического метода LSB. Тестирование происходило с таким объемом скрываемых данных, который позволял бы заполнить контейнер с текущим рейтом внедрения не менее чем на 50%.

Как видно из результатов тестирования методом SPA, при сокрытии информации модифицированным методом в некоторых случаях результаты прохождения теста на 30% лучше, чем классический метод LSB.

Тестирование методом RS. Аналогично тесту SPA тест был проведен для тех же исходных данных.

Тест RS дает неоднозначные результаты. При высоком рейте внедрения модифицированный и оригинальный методы показывают примерно одинаковые результаты, причем от исходных данных зависит то, какой метод себя лучше покажет. Однако при уменьшении рейта внедрения модифицированный метод стабильно показывает себя лучше, чем оригинальный.

Так же был проведен тест архивацией. Под воздействием метода LSB в контейнере с информацией, в нашем случае изображение в формате «.bmp», изменяется состав и распределение цветов. Изменяя последний бит в цвете каждого пикселя, вероятно добавление нового цвета. Таким образом, сжатие такого контейнера становится менее эффективным. Для проведения анализа достаточно сравнить результаты архивации для каждого случая. Результат записан в байтах для алгоритмов RAR и ZIP. Размер архива с пустым контейнером был равен 4331879 и 6558123 байт для RAR и ZIP архивов соответственно.

Проанализировав результаты тестирования, следует заметить, что модифицированный метод LSB показал себя немногим лучше классического метода LSB. Это объясняется тем, что в основе модифицированного метода стоит псевдослучайное распределение, которое не позволяет отличить внедряемое сообщение от белого шума. Наиболее эффективным алгоритм сокрытия становится в тех случаях, когда размер скрываемого сообщения приближается к максимальному объему скрываемой в контейнере информации.

Таблица 1

Тестирование методом SPA

	1/2	1/3	1/5	1/10	1/20	1/25
LSB	0,024555	0,02097	0,02146	0,03018	0,03183	0,02251
LSB-RNG	0,018491	0,0197	0,01968	0,022	0,02263	0,02195

Таблица 2

Тестирование методом RS

	1/2	1/3	1/5	1/10	1/20	1/25
LSB	0,02302	0,03283	0,0313	0,03749	0,04126	0,04079
LSB-RNG	0,02751	0,03148	0,0311	0,03698	0,04058	0,03879

Таблица 3

Тестирование архивацией

RAR/ZIP	1/2	1/3	1/5	1/10	1/20	1/25
LSB	4343189 / 6576627	4345465 / 6582211	4349689 / 6583420	4356634 / 6591246	4360458 / 6595923	4358860 / 6595948
LSB-RNG	4343182 / 6573657	4344548 / 6579574	4347666 / 6585035	4354055 / 6592246	4359431 / 6597641	4358618 / 6597740

Заключение

В ходе проделанной работы авторами была предложена программная реализация модифицированного метода LSB. В качестве результата исследования стоит отметить более высокую стеганографическую стойкость модифицированного алгоритма внедрения в сравнении с его исходным вариантом. Результат этот был выявлен согласно проделанному тестированию. Также стоит отметить, что, несмотря на то, что показатели стеганографической стойкости улучшились, возросла сложность алгоритма, что в свою очередь привело к увеличению нагрузки на ЭВМ.

Согласно проведенному тестированию, программная реализация модифицированного LSB-метода показала, что он является более стойким по ряду параметров при эквивалентных рейтах внедрения в сравнении с классическим LSB. При том модифицированному методу удалось сохранить все достоинства классического метода LSB.

Список литературы

1. Мыздриков Н.Е., Голоднов Е.А., Черкесова Л.В., Ревякина Е.А. Модификация аддитивного ГПСЧ и его реализация на языке C# // Молодой исследователь Дона. 2019. № 2. С. 42–48.

2. Грибунин В.Г., Костюков В.Е., Мартынов А.П., Николаев Д.Б., Фомченко В.Н. Стеганографические системы. Критерии и методическое обеспечение: учебно-методическое пособие / Под ред. докт. техн. наук В.Г. Грибунина. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2016. 324 с.

3. Коржик В.И., Небаева К.А. Основы стеганографии: учебно-методическое пособие по выполнению практических занятий. СПб.: СПбГУТ, 2015. 20 с.

4. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. М.: Солон-Пресс, 2016. 262 с.

5. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. М.: МК-Пресс, 2006. 286 с.

6. Алиев А.Т. О применении стеганографического метода LSB к графическим файлам с большими областями монотонной заливки // Вестник ДГТУ. 2004. № 4 (22). С. 454–460.

7. Аграновский А.В., Балакин А.В., Грибунин В.Г., Сапожников С.А. Стеганография, цифровые водяные знаки и стеганоанализ. М.: Вузовская книга, 2009. 220 с.