С.В. Ярмолик, *студентка Белорусского государственного университета информатики и радиоэлектроники*
Ю.Н. Листопад, *студент Белорусского государственного университета информатики и радиоэлектроники*

Стеганографические методы защиты информации

Введение

Стеганография – это область знаний, которая занимается вопросами скрытой передачи информации. Стеганография (от греческого steganos (секрет, тайна) и graphy (запись)) буквально означает «тайнопись». В отличие от криптографии, *скрытым является сам факт передачи информации*. Особенно эффективным является использование стеганографических методов совместно с криптографическими. Общей чертой стеганографических методов и алгоритмов является то, что скрываемое сообщение встраивается в некоторый безобидный, не привлекающий внимания объект, который затем открыто транспортируется адресату. При использовании криптографии наличие шифрованного сообщения само по себе привлекает внимание злоумышленника, в случае стеганографии наличие скрытой информации остается незаметным. Открытый текст, где скрыта зашифрованная стеганографическим алгоритмом информация, называется *контейнером*.

При использовании стеганографических методов защита информации происходит на трех уровнях:

1) неизвестен сам *факт* передачи скрытой информации;

2) неизвестен *алгоритм внедрения* скрытой информации в контейнер;

3) неизвестен *способ кодирования* информации.

Известные примеры стеганографии включают в себя использование покрытых воском дощечек, вареных яиц и даже головы раба (сообщение читалось после сбривания волос гонца). Скрытое сообщение размещали в определенные буквы невинных словосочетаний. Например, в безобидном тексте:

«A**c**cepted y**o**ur o**v**erture. N**e**xt F**r**iday, a**b**out e**l**even, c**o**me a**w**ay a**n**ywhere»

каждая вторая буква используется для передачи скрытой информации: Cover blown.

Развитие средств вычислительной техники в последнее десятилетие дало новый толчок развитию компьютерной стеганографии. Сообщения встраивают теперь в цифровые данные, как правило, имеющие аналоговую природу – речь, аудиозаписи, видео, графические изображения и даже текстовые файлы и исполняемые файлы программ.

Особый интерес был вызван к стеганографическим методам после того, как в ряде стран были введены ограничения на использование криптосистем. А проблемы защиты прав собственности на цифровую информацию вызвали многочисленные работы в области водяных знаков – специальных меток, незаметно внедряемых в изображение или другой цифровой носитель информации с целью контроля его использования.

Синтаксические методы встраивания скрытой информации в текстовые файлы

В текстовых файлах секретная информация чаще всего кодируется путем изменения количества пробелов, использования невидимых символов, регистра букв, путем изменения межстрочных интервалов, табуляций и т.д. Например, используя регистры букв, нужно спрятать букву «А» в тексте «компьютер». Для этого берем двоичное представление кода

символа «А» – «01010». Пусть для обозначения бита, содержащего единицу, используется символ верхнего регистра, а для нуля – нижнего. Поэтому результатом накладывания маски «01010» на текст «компьютер» будет «кОмПьютер». Окончание «ютер» не используется, поскольку для сокрытия одного символа используется 5 бит, а длина строки 9 символов, вот и получилось, что последние 4 символа – «лишние». Используя такую технологию, можно спрятать в текст длиной N сообщение из N/5 символов. Поскольку данное решение нельзя назвать наиболее удачным, часто используется технология форматирования текста количеством пробелов, отличным от единицы. Скажем, один пробел соответствует биту «0», а два – «1». Программа получает произвольный текст в качестве контейнера и внедряет в него какое-либо стегосообщение. Важную роль тут также играет и способ кодирования символов. Нужно получить такой код символов, чтобы он был оптимальной длины, и двойной пробел встречался минимум раз, т.е. наиболее часто используемые символы имели в коде одну или две единицы.

Чтобы избежать двойного пробела, можно использовать тот факт, что пробел обозначен символом с кодом 32, но в тексте его можно заменить символом, имеющим код 255 (или 0). Так же, как и в прошлом примере, передаем биты шифруемого сообщения, используя обычный текст. Но на этот раз 1 – это пробел, а 0 – это пробел с кодом 255 (или 0).

При использовании методов межстрочных интервалов и табуляций объем информации, встраиваемой в файл, в несколько раз меньше, чем при использовании метода изменения количества пробелов.

Очень подробно рассматривал проблему встраивания скрытой информации в текстовые файлы Брассил. К разработанным им методам относят:

1) *line-shift coding* – изменение расстояния между строками электронного текста;

2) *word-shift coding* – изменение расстояния между словами строки электронного текста. Суть метода состоит в том, что берется текст с разными расстояниями между словами. Выделяются максимальное и минимальное расстояния, которые обозначаются соответственно 1 и 0, а остальные

расстояния увеличивают или уменьшают до размеров выделенных. Частным случаем этого метода является метод изменения количества пробелов, уже рассмотренный выше;

3) *feature coding* – внесение специфических изменений в шрифты отдельных букв, например, вариации длины нижней части буквы р.

Эти методы (*синтаксические)* легко встраиваются в любой текст, независимо от его содержания, назначения и языка. Такие системы легко разрабатывать, и выполняются они автоматически. Но, к сожалению, синтаксические методы легко взламываются, и секретная информация может легко устраняться путем простейших атак типа:

type-in-it-print-it-out

Также большим недостатком является то, что этими методами нельзя передавать большое количество скрытой информации.

Лексические методы встраивания скрытой информации в текстовые файлы

Наиболее часто используются в стеганографии системы, основанные на лексической структуре текста.

Например, в *методе переменной длины слова* набираемые создателем слова должны соответствовать длине, которую укажет программа – обычно в одно слово кодируется два бита информации из стегосообщения. Например, слова длиной в 4 и 8 символов могут означать комбинацию бит «00», длиной в 5 и 9 – «01», 6 и 10 – «10», 7 и 11 букв – «11». Слова, короче 4 и длиннее 11 букв, можно вставлять где угодно для лексической и грамматической связки слов в предложении – программа-декодер будет просто игнорировать их.

Другой метод – *метод первой буквы* – может передавать еще больше скрытой информации в одном слове: обычно это три или четыре бита. Программа-помощник в этом методе накладывает ограничение уже не на длину слова, а на первую (можно на вторую) букву. Обычно одну и ту же комбинацию могут кодировать несколько букв, например, комбинацию «101» означают слова, начинающиеся с «А», «Г»

или «Т». Это дает большую свободу выбора оператору, придумывающему стегосообщение, и текст не будет нелепым.

Одним из наиболее обсуждаемых методов является метод, основанный на системе синонимов языка, используемого для написания электронного текста. Проведенные исследования для случая английского языка, показали, что среднее количество синонимов в одном подмножестве равняется 2,56. Минимальное количество синонимов во множестве равняется 2, а максимальное – 13.

В качестве примера приведем множество синонимов S_0:{«propensity», «predilection», «penchant», «proximity»}. В приведенном множестве синонимов каждое слово имеет единственное смысловое значение, что позволяет закодировать каждое слово своим уникальным кодом, например, «propensity»= 00, «predilection»= 01, «penchant»= 10, «proximity»= 11. Подобное кодирование позволяет выбирать одно из четырех слов в зависимости от двух бит секретного сообщения. Отметим, что при этом семантика (смысл) сообщения не изменится, независимо от того, какое из четырех слов будет выбрано.

Процедура передачи секретного сообщения с использованием лексической стеганографии, производится в следующей последовательности.

Отправитель и получатель имеют одинаковое множество синонимов, поддерживаемое одним и тем же электронным словарем.

1. Отправитель выбирает контейнер (текстовый файл).

2. Отправитель преобразует секретное сообщение в двоичную последовательность: ЦВЗ=>...01000..., используя криптографические методы.

3. Отправитель, последовательно анализируя текстовый файл, находит первое слово, для которого существуют N синонимов.

4. Отправитель вычисляет целую часть значения $\log_2 N$, которая определяет число символов секретного сообщения, которые могут быть внедрены в контейнер путем выбора соответствующего синонима. Например, если в тексте встретилось слово «penchant», принадлежащее множеству S_0:{«propensity», «predilection», «penchant», «proximity»},

в соответствии со значением двух бит закодированной информации – 01, это слово должно быть заменено синонимом «predilection».

Аналогичные действия выполняет и получатель. Получатель анализирует слова в контейнере (текстовом файле) на предмет принадлежности к множеству синонимов. Если текущее слово относится к одному из множеств синонимов, он определяет мощность этого множества N. Целая часть $\log_2 N$ определяет число бит, которые закодированы на основании текущего множества синонимов. Например, если получатель обнаружит в тексте слово «predilection» и определит, что оно относится к множеству синонимов S_0, состоящему из N=4 синонимов, тогда $\log_2 N=2$ бита, а слово «predilection» интерпретируется как два бита 01 секретного сообщения.

В случае слов с несколькими смысловыми значениями подобное кодирование оказывается невозможным. Также невозможно кодирование, если один из синонимов состоит из двух (или более) слов, разделенных пробелом.

К сожалению, количество синонимов не всегда равно 2^K. Например, S_1:{AAA-0, BBB-1, CCC-2}; S_2:{MMM-0, NNN-1, OOO-2, PPP-3 ,QQQ-4}; S_3:{WWW-0, XXX-1, YYY-2}.

Использование трех приведенных множеств синонимов согласно приведенной ранее идее семантической стеганографии позволяет закодировать один бит на основании первого и третьего множеств и два бита за счет второго множества. Всего может быть закодировано только четыре бита, в то время как мощность приведенных множеств позволяет закодировать 3*5*3=45 состояний или не менее, чем 5 бит.

Для увеличения объема внедряемой в контейнер секретной информации может быть использована система счисления со смешанным основанием. В этом случае работают не с одним множеством синонимов, а с группой множеств синонимов. Кодирование выполняется не путем изолированного использования каждого множества синонимов, а путем использования их совокупности. Например, для приведенных трех множеств можно закодировать $[\log_2 45]=5$ бит вместо 4. Для этого используем код со смешанным основание из трех цифр, в котором первая и последняя цифры могут принимать значения от 0 до 2, а средняя – от 0 до 4. Ниже приведена

таблица, показывающая соответствие между двоичным кодом и кодом со смешанным основанием.

Таблица 1.

Соответствие между двоичным кодом и кодом со смешанным основанием

Двоичный код	00000	00001	00010	00011	00100	…	01110	01111	…	11111
Код со смешанным основанием	000	001	002	010	011	…	042	100	…	201

Мимикрия

Другим, не менее распространенным методом передачи скрытой информации, является *мимикрия*. Мимикрия генерирует осмысленный текст, используя синтаксис, описанный в Context Free Grammar (CFG), и встраивает информацию, выбирая из CFG определенные фразы и слова. CFG – это один из способов описания языка, который состоит из статических слов, фраз, узлов, мест, где может быть принято решение, какое слово или фразу дальше вставлять в текст. Мимикрия создает бинарное дерево, которое основано на возможностях CFG, и составляет текст, выбирая те из листьев дерева, которые кодируют нужный бит. Например, нужно скрыть следующие биты – 1010, используя следующее дерево:

Старт → существительное глагол
существительное → Илья || Иван
глагол → поехал на рыбалку **куда** || поехал покататься на лодке **куда**
куда → в **направление** Корею. || в **направление** Миннесоту.
направление → северную || южную

Таблица 2.

Формирование текста для внедрения в него скрытой информации 1010

Шаг	Ответы, полученные в процессе	Скрытый бит	Выбор процедуры
1	**Старт**	–	**Старт → существительное глагол**
2	**существительное глагол**	1	**существительное →** Иван
3	Иван **глагол**	0	**Глагол →** поехал на рыбалку **куда**
4	Иван поехал на рыбалку **куда**	1	**куда →** в **направление** Миннесоту
5	Иван поехал на рыбалку в **направление** Миннесоту.	0	**направление→** северную Миннесоту

Окончательно получилось следующее предложение: Иван поехал на рыбалку в северную Миннесоту.

Недостатками этого метода являются: невозможность передачи больших объемов информации, низкая производительность метода и невысокая скрытность.

Существует множество других методов преобразования текста. В любом из них необходимо решать оптимизационную задачу: контейнер необходимо «плотно» заполнить стегоинформацией, но при этом он совершенно не должен выделяться из обычной общей массы файлов такого же формата и наполнения.

Методы встраивания скрытой информации в графические файлы

Намного проще дело обстоит с графическими изображениями, аудио- и видеофайлами. Используя в качестве контейнера графические файлы, можно встраивать не только текстовую информацию, но и изображения, и другие файлы. Единственным условием является то, что объем спрятанного рисунка не должен превышать размер изображения-хранилища. Для достижения этой цели каждая программа использует свою технологию, но все они сводятся к замене определенных пикселей в изображении.

Цифровое изображение представляет собой матрицу пикселей. Как известно, пиксель – это единичный элемент изображения. Он имеет фиксированную размерность двоичного представления. Например, пиксели полутонового изображения кодируются 8 битами (значение яркости изменяется от 0 до 255).

Младший значащий бит (LSB) изображения несет в себе меньше всего информации. Известно, что человек обычно не способен заметить изменение в младших битах и, в особенности, в самом младшем. Фактически значение младшего бита является случайным, не несущим информации, различимой глазом человека. Поэтому его можно использовать для встраивания информации. Таким образом, для полутонового изображения объем встраиваемых данных может составить 1/8 объема контейнера.

Реальные графические образы позволяют передавать большие объемы информации. Так, для графического образа

1024x768 можно в закрытом виде передать 294912 байт, используя метод наименее значимого бита – LSB.

Например, для случая задания цвета тремя цветами RGB каждый пиксель описывается тремя байтами, т.е. 24-битами. Для случая внедрения секретной информации в виде символа А→01000001 в трех последовательных пикселях, каждый из которых описывается тремя байтами:

(00100111, 11101001, 11001000)
(00100111, 11001000, 11101001)
(11001000, 01000111, 11101001)

модификации подвергается только часть значений младших бит приведенных значений байт. В результате получим

(0010011***0***, 11101001, 11001000)
(0010011***0***, 11001000, 1110100***0***)
(11001000, 01000111, 11101001)

К сожалению, не все графические форматы сохраняют значения младших разрядов при различного рода преобразованиях (сжатие, распаковка). Факт кодирования легко обнаружить, т.к. статистические свойства младших битов будут определяться шифруемым текстом и отличаться от случайных равновероятных значений. Чаще всего секретное сообщение или ЦВЗ первоначально шифруется с использованием криптографического метода на базе секретного ключа.

В цветных графических изображениях заменять можно не только один самый младший бит, а два или три младших. Эти изменения человеческим глазом практически не различимы. И еще один момент: в качестве контейнера желательно выбирать не искусственно созданные картинки (стего в них обманет только неискушенного пользователя), а отсканированные фотоизображения. Только в них присутствуют шумы квантования – случайное заполнение младших бит, под которые и маскируются кусочки стегосообщения. Необходимо избегать фотографий с большими областями очень ярких и черного цветов. На таких картинках большие области в исходном файле имеют цветовые составляющие 255 и 0, соответственно и стегобайты будут характерно выделяться при просмотре файла в кодах своими 254 и 1.

При использовании стеганографических систем следует помнить, что чем больше информации встраивается в файл-контейнер, тем меньше надежность системы.

Система «Stegano-1S»

На основе вышеперечисленных методов была разработана стеганографическая система «Stegano-1S», которая работает с текстовыми файлами типа .txt, .asm, .cpp, .pas и т.д., а также графическими файлами формата .bmp.

При работе с текстовыми файлами используется метод, основанный на изменении количества пробелов между словами. На входе система получает файл и информацию, которую необходимо внедрить в этот файл. Система кодирует полученное стегосообщение таким образом, чтобы встраиваемая информация вносила минимальные изменения. Потом система анализирует файл и при необходимости урезает данную информацию (оставшаяся информация может быть внедрена в другой текстовый файл или выбран другой файл-контейнер) и с помощью подстановки дополнительных или уничтожения ненужных пробелов встраивает заданную информацию таким образом, чтобы изменения текста были минимальными. После внедрения скрытой информации в текстовые файлы структура текста практически не меняется, а исходные тексты программ компилируются без особых проблем. В предлагаемой системе используются различные методы кодирования передаваемой информации.

При работе с графическими файлами используется метод наименее значимого бита. Работа происходит по тому же алгоритму, меняется только способ внедрения в файл-контейнер скрытого сообщения. Главным достоинством этого метода является возможность передачи большого количества информации. При этом исходное изображение практически невозможно отличить человеческим глазом от преобразованного.

Литература

1. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. – М.:СОЛОН-Пресс, 2002. – 261 с.

2. Cox I.J., Miller M.L., Bloom J.A. Digital Watermarking. Morgan Kaufmann Publisher. Academic Press. – 2002. – 542 p.

3. Business Software Alliance. The cost of software piracy: BSA's glabal enforcement policy. – 1996. http://www.rad.net.id/bsa/piracy/globalfact.html.

4. Collberg C.S., Thomborson C.. Watermarking, Tamper-proofing, and Obfuscation – Tools for Software protection. Computer Science Technical Report #170. University of Auckland, Aucland, New Zealand. February 10, 2000.

5. Charbon E., Torunoglu I.H. On Intellectual Property Protection. In proceedings of Custom Integrated Circuits Conference, 2000, pp. 517-523.

6. Torunoglu I., Charbon E. Watermarking-Based Copyright Protection of Sequential Function. IEEE Journal Solid-State Circuits, Vol, 35, №3 February 2000, pp. 434-440.

7. Katzenbeisser S., Petitcolas F.A. Information Hiding: Techniques for steganography and digital watermarking. Artech House. Inc. – 2000. – 221 p.

8. Wayner P. Disappearing Cryptography: Information Hiding, Steganography & Watermarking. Second Eddition. Morgan Kaufmann Publisher. Elsevier Science Publisher. – 2002. – 413 p.