УДК 003.26

Новый метод внедрения скрытых сообщений в алфавитное меню

И. В. Нечта

В данной статье представлен новый метод стеганографии, позволяющий встраивать скрытые сообщения в алфавитное меню. Основная идея метода заключается в том, что предлагается модифицировать один из известных методов автоматического построения иерархического меню. Здесь модификация подразумевает изменение дерева меню путём переноса крайних элементов из одного подменю в другое в соответствии со скрываемым сообщением. В ходе экспериментального анализа было показано, что средняя высота дерева меню увеличивается менее чем на единицу. Данный метод внедрения может быть применен для встраивания цифровых отпечатков пальцев или водяных знаков в иерархию элементов, имеющих алфавитную сортировку.

Ключевые слова: стеганография, алфавитное меню, стеганография в интерфейсах.

1. Введение

На сегодняшний день известно большое количество методов внедрения скрытых сообщений в цифровые объекты данных. Задача встраивания секретных сообщений, решаемая в рамках науки стеганографии, формулируется следующим образом. Пусть имеются два участника обмена сообщениями: Алиса и Боб. Их задача – создать тайный канал связи так, чтобы постороннее лицо, Ева, не заподозрила о существовании такого канала. В рамках решения данной задачи Алиса при помощи специальных стеганографических алгоритмов встраивает скрытое сообщение в безобидный на внешний вид объект данных, так называемый `контейнер`. Сам факт передачи контейнера по открытому каналу связи не является для Евы чем-то подозрительным. Боб получит контейнер, сможет извлечь и прочитать секретное сообщение.

Предполагается, что Алиса и Боб заранее договариваются о методе внедрения и ключах шифрования сообщения. Свойства стеганографических алгоритмов таковы, что Ева, подвергнув контейнер анализу, не сможет однозначно утверждать ни о наличии, ни об отсутствии факта внедрения скрытого сообщения.

Ещё одной задачей, решаемой в рамках передачи скрытой информации, является задача стегоанализа. Стегоанализ применяется для выявления факта передачи секретного сообщения. Большинство методов стегоанализа используют статические различия свойств пустого и заполненного контейнера. Известен принцип Керкгоффса [1], который предполагает, что стегоаналитик, в нашем случае – Ева, заранее знает о том, какой метод внедрения будет использоваться. Эффективность методов стегоанализа определяется ошибками 1-го и 2-го рода. Ошибка 1-го рода: случай, когда заполненный контейнер воспринимается как пустой. Ошибка 2-го рода: случай, когда пустой контейнер воспринимается как заполненный. Часто для демонстрации эффективности стегоанализа используют зависимость среднего значения ошибок от размера контейнера.

Стеганография активно применяется для защиты объектов авторского права. Часто возникают задачи встраивания специальных меток в файлы, которые позволяют идентифициро-

вать автора. Например, фотография, сделанная одним лицом, выдаётся за фотографию постороннего лица. Такие метки, называемые цифровыми водяными знаками (ЦВЗ), могут быть извлечены и использованы для доказательства истинного авторства. Существует ряд требований, предъявляемых к ЦВЗ, например, устойчивость к определённым искажениям файла (масштабирование, представление в другом формате данных, снижение качества изображения и т.д.).

Часто возникают задачи выявления несанкционированных модификаций файла. Например, пользователь при установке браузера проверяет цифровую подпись дистрибутива, что гарантирует целостность файла. Однако при дальнейшем использовании браузер может быть искажен вредоносной программой с целью кражи конфиденциальных сведений. В таких случаях встраиваемый ЦВЗ должен обладать свойством хрупкости, подразумевающим его разрушение при любом изменении файла. При каждом запуске программа анализирует ЦВЗ, и при его искажении пользователь уведомляется об опасности.

Ещё одним применением стеганографии является встраивание меток, идентифицирующих покупателя. Например, при создании программного продукта производитель в каждую продаваемую копию встраивает так называемый цифровой отпечаток пальца (ЦОП), идентифицирующий покупателя. В случае, когда будет обнаружена нелицензионная копия программы, производитель сможет без труда извлечь ЦОП и определить нарушителя лицензионного соглашения.

Подавляющее число публикаций, посвященных методам стеганографии, используют изображения для встраивания скрытых сообщений. Широко известен класс внедрения сообщений в последний значащий пиксель изображения (LSB), описываемый в работе [2]. Считается, что изображение имеет бо́льший объём внедрения по сравнению с текстовыми, аудио и исполняемыми файлами. С развитием социальных сетей и мессенджеров возросла актуальность создания методов стеганографии текстовых данных. Наиболее значимые методы представлены в обзорной статье [3]. Ряд методов внедрения в исполняемые файлы представлен в работах [4–5].

В настоящее время по-прежнему актуальна задача разработки новых методов стеганографии, использующих различные типы контейнеров, например, в социальных сетях [6]. В данной работе предлагается новый метод внедрения скрытых сообщений, который использует алфавитное меню в качестве контейнера. Здесь предлагается использовать модификацию алгоритма автоматического построения алфавитного меню, применяемого в работе [7]. В частности, предлагается дополнить исходный алгоритм перегруппировкой элементов меню в соответствии со скрываемым сообщением. В ходе экспериментального анализа было показано, что средняя высота дерева меню увеличивается менее чем на единицу.

2. Описание предлагаемого метода

При построении меню программы часто необходимо участие человека, так как нужно группировать элементы с учетом их смысла. Алфавитное меню, напротив, может быть построено в автоматическом режиме. Ряд работ посвящен разработке алгоритмов автоматического построения алфавитного меню [8–9], однако данные алгоритмы имеют экспоненциальную трудоемкость. В работе [7] было предложено использовать алгоритм построения меню с полиномиальной сложностью. Последний алгоритм позволяет производить автоматическое построение алфавитного меню с различным размером уровня меню в иерархии.

Теперь опишем подробнее предлагаемый в настоящей статье метод внедрения скрытых сообщений. Будем рассматривать меню в виде дерева. Исследуемое дерево меню относится к классу Б-деревьев. Каждый элемент соответствует либо одиночному (конечному) пункту меню, либо группе элементов.

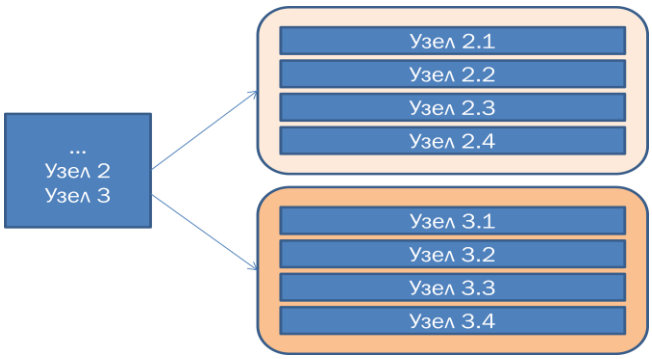

Рис. 1. Исходное меню

На рис. 1 представлена одна родительская группа узлов («Узел 2», «Узел 3») и две группы дочерних, содержащих узлы с 2.1 по 2.4 в одной группе и с 3.1 по 3.4 в другой. Пользователь сначала попадает на родительский узел и после делает переход на одну из групп дочерних узлов.

Внедрение информации происходит путём переноса граничного элемента (в данном примере – «Узел 2.4») из одной группы в другую (соседнюю) при наличии. Для внедрения нулевого бита перенос не производится. Для внедрения единичного бита происходит перенос элемента согласно рис. 2 с последующей перестройкой всех дочерних элементов меню.

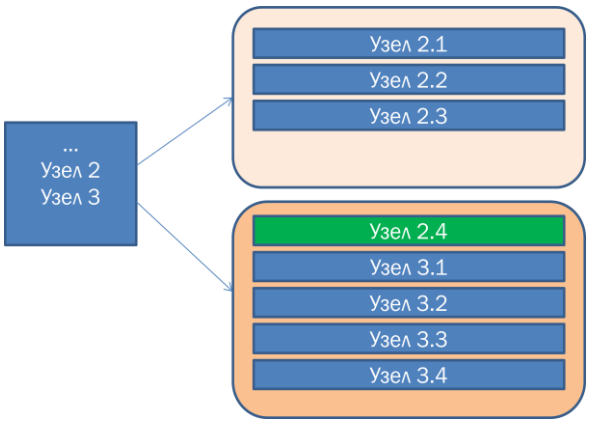

Рис. 2. Внедрение бита «1» секретного сообщения

Таким образом, для каждого родительского узла осуществляется внедрение в его дочерние группы, если имеется соседняя группа и количество элементов в текущей группе более одного. Последнее ограничение связано с тем, что в случае переноса единственного элемента из группы она останется пустой и, следовательно, разные внедрённые сообщения образуют одинаковую иерархию меню, что недопустимо.

Рассмотренный алгоритм может быть записан на псевдокоде следующим образом. Нам потребуется предварительно ввести понятие `смежных` узлов дерева. Когда производится построение дерева меню, то исходное упорядоченное множество его элементов разбивается на несколько подмножеств. Каждое такое подмножество располагается в своем узле дерева. Если количество элементов в узле больше некоторого параметра (называемого максимальной шириной меню), то подмножества также разбиваются на более мелкие, образуя дочерние узлы дерева. Пусть узел A содержит подмножество упорядоченных элементов $\{e_i, \dots e_j\}$, тогда будем называть его `смежным` с узлом B, если узел B содержит элемент меню e_{j+1} и оба узла имеют одинаковую высоту в дереве.

`Внедрение бита` осуществляется только при наличии смежного узла. При внедрении бита «1» элемент меню e_j перемещается в узел B, при внедрении «0» перемещение не производится. Предполагается, что количество смежных вершин (мест для внедрения битов) в де-

реве будет больше либо равно длине внедряемого сообщения. Неиспользованные смежные вершины заполняют нулевыми битами.

Алгоритм 1. Алгоритм внедрения секретного сообщения в меню
Вход: Дерево меню T, секретное сообщение $M = \{m_1, m_2 \dots m_N\}$, где N – длина сообщения и $m \in \{0,1\}$.
Выход: Дерево меню со скрытой информацией.

```
k=1; //номер внедряемого бита сообщения M
i=1; //текущая высота узлов дерева
пока i < высота дерева(T)
н-ц
        j=1; //номер текущего узла дерева с высотой i
        пока j < кол-во узлов дерева T с высотой i
        нц
        если узел j и j+1 смежные то
                внедрить бит m_k
                k=k+1
        кц
i=i+1;
к-ц
```

Предлагаемый в данной статье метод внедрения предназначается для встраивания цифрового отпечатка пальца. Внедрение сообщения будет происходить согласно Алгоритму 1. Проверка ЦОП происходит без его непосредственного извлечения путем сравнения хэш-функции (подробнее см. [10]), вычисленной от содержимого контейнера, с имеющимся набором (базой хэшей, соответствующих лицензированным пользователям).

Рассмотрим более подробно процесс построения исходного меню (без внедрения), предложенный в работе [7]. Пусть имеется множество элементов и задано распределение вероятностей их использования. Построим первый (родительский) уровень меню, объединяя по два соседних элемента, имеющих минимальную сумму вероятностей. Объединённые элементы превращаются в группу, которая также может быть объединена. Объединение проводится до тех пор, пока не будет достигнут требуемый размер уровня меню. На рис. 3 показаны исходные элементы a – g и приведены вероятности их использования.

На этапе построения текущего уровня меню мы можем осуществлять внедрение информации. Дочерние уровни меню строятся аналогичным образом.

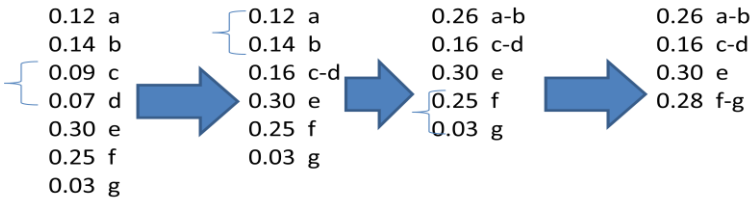

Рис. 3. Процесс построения исходного меню (без внедрения)

3. Экспериментальное исследование алгоритма

Качество построенного меню определяется средним временем поиска элементов в нём. Следовательно, чем меньше переходов по иерархии выполняет пользователь, тем меньше время поиска. В работе [7] дерево меню рассматривалось как дерево кода сжатия данных. При таком рассмотрении качество меню может быть оценено с помощью избыточности кодового дерева, т.к. среднее время поиска и средняя длина кодового слова связаны напрямую.

Очевидно, что, меняя иерархию меню в соответствии с внедряемым сообщением, мы вносим некоторую избыточность в кодовое дерево. Для определения объёма указанной избыточности и объёма внедрения необходимо произвести моделирование. Определим и зафиксируем параметры проведения эксперимента.

Во-первых, внедряемое сообщение предварительно шифруется. Известно, что одним из требований, которые предъявляются к шифрам, является статистическая неразличимость зашифрованного сообщения от истинно случайной последовательности. Следовательно, мы можем имитировать внедряемое секретное сообщение последовательностью, полученной из генератора псевдослучайных чисел.

Во-вторых, согласно исследованию, проведённому авторам работы [11], рекомендуется использовать меню из восьми элементов. С одной стороны, малый размер уровня меню вынуждает пользователя часто переходить вглубь иерархии, что в соответствии с законом Фитса [12] затратно по времени. С другой стороны, большое количество элементов согласно закону Хика [13] также увеличивает время на поиск и выбор нужного элемента. Таким образом, автор статьи [11] обосновывает и экспериментально подтверждает эффективность (по времени поиска) восьмиэлементного меню.

В-третьих, при построении меню нам необходимо использовать распределение вероятностей использования его элементов. Согласно результату исследования, представленному в работе [14], при моделировании алфавитного меню следует выбирать распределение вероятностей Ципфа [15], представленное на рис. 4. Считается, что распределение вероятностей вызова абонентов из телефонного справочника (типичный случай алфавитного меню) также соответствует данному распределению. Здесь ранг, равный 1, получает самый часто встречающийся элемент, второй по встречаемости элемент получает ранг 2 и так далее.

В работе [14] и в настоящей статье вероятность появления элементов меню ставилось в соответствие вероятностям из распределения Ципфа следующим образом. Пусть $P_{Zip} = \{p_1, \dots p_N\}$ распределение Ципфа. Тогда вероятность, соответствующая элементу меню $Q_i = p_{random(i)}$, где функция $random(i)$ генерирует число в интервале $[1; N]$, причем сгенерированное число ни разу не повторяется при $i \in [1; N]$.

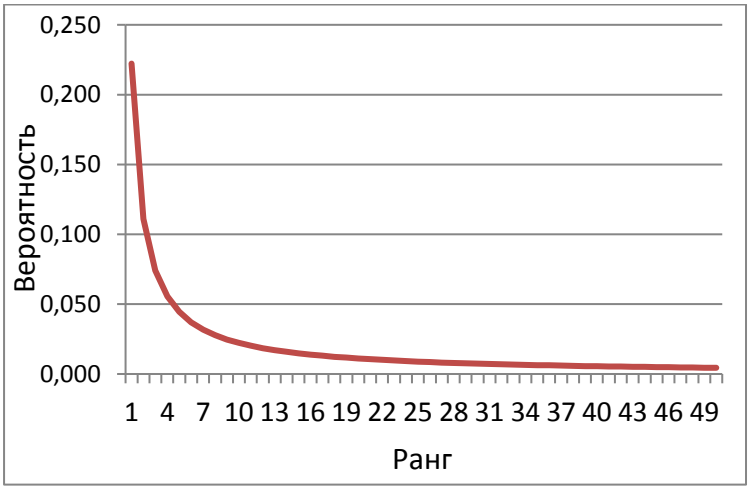

Рис. 4. Распределение Ципфа

Таким образом, зафиксировав параметры, проведём моделирование для меню с числом элементов от 100 до 1000. Результаты представлены в табл. 1. Здесь энтропия и избыточность измеряется в октабитах, так как размер уровня меню составляет восемь элементов.

Анализируя полученные результаты, можно утверждать, что объём внедрения предлагаемым методом превосходит современные методы текстовой стеганографии, например, методы замены синонимов [16], где объём внедрения составляет 0.3 бит на синоним.

Из табл. 1 видно, что избыточность, вносимая внедрением секретного сообщения, менее одного уровня меню. Таким образом, средняя скорость поиска элементов в меню уменьшается незначительно.

В результате исследования был предложен новый метод стеганографии, позволяющий встраивать секретное сообщение в алфавитное меню. Результаты моделирования показали, что предложенный метод по объёму внедрения не уступает известным методам текстовой стеганографии.

Таблица 1. Результаты моделирования

Число элементов меню	Объём внедрения, бит	Энтропия, октабит	Избыточность, октабит	
			без внедрения	с внедрением
100	26	1.77	0.280	0.38
200	43	1.99	0.410	0.52
300	68	2.12	0.190	0.35
400	104	2.21	0.380	0.51
500	125	2.28	0.270	0.40
600	160	2.34	0.280	0.49
700	160	2.38	0.270	0.39
800	181	2.42	0.240	0.42
900	220	2.46	0.261	0.39
1000	231	2.49	0.266	0.53

Литература

1. *Kerckhoffs A.* La cryptographie militaire // Journal des sciences militaires. 1893. V. 9. P. 5–83.
2. *Qin J., Xiang X., Wang M.* A Review on Detection of LSB Matching Steganography // Information Technology Journal. 2010. V. 9. P. 1725–1738.
3. *Nechta I., Fionov A.* Applying stat methods to text steganography // Applied Methods of Statistical Analysis. Simulations and Statistical Inference. NSTU. 20–22 September 2011. P. 278–284.
4. *Hamilton J., Danicic S.* A survey of static software watermarking // IEEE World Congress on Internet Security (WorldCIS). 2011. P. 100–107.
5. *Schaathun H. G.* On watermarking/fingerprinting for copyright protection // IEEE First International Conference on Innovative Computing, Information and Control. 2006. V. 3. P. 50–53.
6. *Nechta I.* Steganography in Social Networks // IEEE Siberian Symposium on Data Science and Engineering (SSDSE), Technopark of Novosibirsk Akademgorodok, Russia. 12–13 Apr. 2017. P. 33–35.
7. *Нечта И. В.* Алгоритм построения алфавитного меню полиномиальной сложности // Вестник СибГУТИ. 2017. № 1. С. 90–95.
8. *Нечта И. В., Рябко Б. Я., Савина Н. Н.* Применение алфавитного кодирования для оптимизации интерфейса // Вычислительные технологии. 2015. Т. 20, № 5. С. 97–104.

9. *Нечта И. В.* Построение меню при помощи алфавитного кода // Вестник СибГУТИ. 2015. № 4. С. 40–46.
10. *Рябко Б. Я., Фионов А. Н.* Криптографические методы защиты информации: учебное пособие для вузов. М.: Горячая линия – Телеком, 2012. 229 с.
11. *Zaphiris P. G.* Depth vs Breath in the Arrangement of Web Links // Proceedings of the Human Factors and Ergonomics Society Annual Meeting. SAGE Publications, 2000. V. 44, № 4. P. 453–456.
12. *Hick W. E.* On the rate of gain of information // Quarterly Journal of Experimental Psychology. 1952. V. 4. P. 11–26.
13. *Fitts P. M.* The information capacity of the human motor system in controlling the amplitude of movement // Journal of Experimental Psychology. 1954. V. 47 (6). P. 381–391.
14. *Witten I. H., Cleary J. G., Greenberg S.* On frequency-based menu-splitting algorithms // International Journal of Man-Machine Studies. 1984. V. 21, № 2. P. 135–148.
15. *Newman M. E. J.* Power laws, Pareto distributions and Zipf's law // Contemporary physics. 2005. V. 46, № 5. P. 323–351.
16. *Winstein K.* Tyrannosaurus lex 1999. [Электронный ресурс]. URL: `http://web.mit.edu/keithw/tlex/` (дата обращения: 22.10.2017).

Статья поступила в редакцию 15.01.2018;
переработанный вариант – 23.04.2018.

Нечта Иван Васильевич
к.т.н., доцент, начальник отдела подготовки кадров высшей квалификации СибГУТИ (630102, Новосибирск, ул. Кирова, 86), тел. (383) 2-698-272, e-mail: `aspirant@sibsutis.ru`.

New method for secret message embedding in alphabetical menu

I. Nechta

In this article, we present a new method of steganography which allows you to embed hidden messages in the alphabetical menu. The main idea of this method consists in modification earlier known methods of automatic construction of a hierarchical menu. Here the modification implies changing the menu tree by transferring boundary elements from one submenu to another one in accordance with the hidden message. In the course of the experimental analysis it was shown that the average height of the menu tree increased by less than one. This steganography method can be used for embedding digital fingerprints or watermarks in the hierarchy of items having an alphabetical order.

Keywords: steganography, alphabetical menu, steganography in interfaces.