ИСПОЛЬЗОВАНИЕ СИСТЕМНЫХ СВОЙСТВ И ПАРАМЕТРОВ ТЕКСТОВЫХ ФАЙЛОВ В СТЕГАНОГРАФИЧЕСКИХ ПРИЛОЖЕНИЯХ

П. П. Урбанович, Д. Э. Юрашевич[b]
Белорусский государственный технологический университет
Минск, БЕЛАРУСЬ
e-mail: [a]pav.urb@yandex.by, [b]dima.yurashevich32155@gmail.com

Стеганографические методы и инструментальные средства применяются для хранения и/или передачи скрытой информации. При использовании электронных текстовых документов в качестве контейнеров скрытое размещение такой информации обычно основывается на модификации цветовых или пространственно-геометрических параметров символов текста-контейнера. В статье исследуются и анализируются системные свойства и параметры электронных текстовых документов как потенциальных элементов стеганографической системы, в которые осаждается тайная информация. Основная задача исследования – оценка стеганографической стойкости текста-контейнера при модификации указанных свойств и параметров файлов в процессе осаждения тайной информации.
Ключевые слова: защита информации; текстовая стеганография; системные свойства и атрибуты фалов; стеганографическая стойкость

1 Введение

Проблема зашиты информации и защиты авторских прав в IT-сфере приобретает все большую актуальность. Одно из направлений решения проблемы предусматривает использование стеганографии, которая, в отличие от криптографии, основана даже на сокрытии самого факта использования преобразования [3,4]. Вместе с тем известно, что моделирование стеганографических и криптографических систем основывается на тех же базовых принципах [4,5,11].

Современная стеганография, как правило, «имеет дело» с электронными средствами. Это может объясняться следующими причинам. Во-первых, так как объем осаждаемой информации, как правило, довольно небольшой по сравнению с размером контейнера, в котором она будет скрыта, то в электронные контейнеры гораздо проще скрывать данные и извлекать их. Во-вторых, процедура осаждения/извлечения может быть автоматизирована с помощью специальных программных средств. В-третьих, электронный формат данных характеризуется информационной избыточностью, которой можно управлять, чтобы скрыть сообщения. Эти общие для всех электронных документов особенности, на наш взгляд, ставят знак равенства между цифровой и компьютерной стеганографией.

Основу электронного контента составляют текстовые документы, базы данных, коды компьютерных программ, а также объекты мультимедиа. Защита указанных средств от подделки или несанкционированного использования стеганографическими методами предусматривает выполнение нескольких важных условий.

К основным из них относят трудность обнаружения отличий между параметрами исходного (пустого) файла-контейнера и стеганоконтейнера – файла с осажденной информацией, а также устойчивость (неизменность) осажденной в контейнер информации при его модификации. Оба указанных условия относятся к известной характеристике методов рассматриваемого класса – стеганографической стойкости.

Для стеганоконтейнеров в виде текстовых файлов разработаны методы осаждения/извлечения тайной информации (или цифровых водяных знаков), основанные на модификации различных пространственно-геометрических (пробелы между словами, пробелы меду строками, использование невидимых символов, модификация апроша, кернинга, масштаба и др.) или цветовых параметров символов текста на основе RGB-модели [2,3,4,7,8,9,10].

На сегодняшний день задача поиска методов преобразования контейнеров, в том числе – текстовых, и встраивания в них тайной информации по-прежнему является актуальной. Часто наиболее устойчивые к атакам стеганоалгоритмы не позволяют встроить достаточный объем секретной информации в файл-контейнер. Разработка таких алгоритмов встраивания, которые, с одной стороны, повышают стеганостойкость системы, а с другой – сохраняют объем осажденных секретных данных, может быть отнесена к числу важных задач.

Одно из новых направлений в анализируемой предметной области связано с использованием так называемых альтернативных потоков файловой системы NTFS [1]. При этом стеганографическое преобразование не затрагивает собственно содержимое файла-контейнера. В настоящей статье анализируются особенности и возможности использования подхода на основе альтернативных потоков данных в приложении к текстовым документам-контейнерам. Здесь аналогом альтернативных потоков служат системные свойства и параметры файлов.

2 Системные свойства и параметры текстового файла

В качестве объекта исследования использовался файл с расширением DOCX. Причем для получения и анализа его свойств и параметров файл не должен быть пустым. Исследования проводились с использованием библиотеки *Microsoft. WindowsAPICodePack.Shell.*

Встроенные средства ОС *Windows* позволяют обычному пользователю получить информацию о следующих системных свойствах и параметрах:

- свойства описания: *название*, *тема*, *теги*, *категории*, *комментарии*;
- свойства источника: *авторы*, *кем сохранен*, *редакция*, *номер версии*, *имя программы*, *организация*, *руководитель*; *дата создания содержимого*, *дата последнего сохранения*, *последний вывод на печать*, *общее время редактирования*

- свойства содержимого: *состояние содержимого*, *тип содержимого*, *количество страниц*, *количество слов*, *количество знаков*, *количество строк*, *количество абзацев*, *шаблон*, *шкала*, *ссылки*, *язык*;

- свойства файла: *размер*, *дата создания*, *дата изменения*, *дата доступа*, *доступность*, *автономность*, *компьютер*.

Из числа приведенных свойств поддерживают модификацию следующие 13: *язык*, *тип содержимого*, *состояние содержимого*, *руководитель*, *организация*, *номер версии*, *редакция*, *авторы*, *комментарии*, *категории*, *теги*, *тема*, *название*.

Для выявления и анализа других свойств и параметров файла разработана программная реализация метода, принимающего в качестве параметра путь доступа к документу. Метод возвращает лист пар ключ–значение. Методом создаются переменные, которые представляют собой набор родительских элементов, которые, в свою очередь, содержат дочерние системные свойства и параметры. В данном методе также используется вызов другого метода, принимающего на вход объект родителя системных свойств для заполнения результирующего листа полученными ключ–значениями. В результате нами выявлено 208 свойств. Каждое значение свойства обладает своим типом данных. Этот тип данных не является уникальным. Указанные свойства по принадлежности к типу данных распределены следующим образом: *String* – 83, *UInt32* – 26, *Boolean* – 23, *String[]* – 23, *DateTime* – 13, *Object* – 12, *UInt16* – 8, *UInt64* – 8, I*nt32* – 5, *IntPtr* – 3, *Byte[]* – 2, *IntPtr[]* – 1, *IStream* – 1.

Для получения свойств, основанных на родительском свойстве, использовался метод, обладающий модификатором доступа «private» и не возвращающий никакого значения. Это обусловлено теми обстоятельствами, что его роль заключается в получении и заполнении переменной листа, и доступ к методу должен осуществляться исключительно из метода, находящегося в одном классе с данным методом.

Родительские свойства логически объединяют некоторые системные свойства. Все нижеперечисленные свойства описаны в классе *ShellProperties*. Данный класс определяет другой класс, который реализует вспомогательные методы для извлечения свойств оболочки, используя каноническое имя, ключ свойства или строго типизированное свойство. Он также обеспечивает доступ ко всем строго типизированным системным свойствам и коллекциям свойств по умолчанию.

Для модификации свойств, подразумеющей стеганографическое осаждение информации, разработано специальное приложение, функционал которого заключается в том, что оно обрабатывает действие нажатия на кнопку изменения свойств. При этом проводится проверка на валидность ввода данных.

3 Анализ результатов модификации системных свойств и параметров файла-контейнера

Выявлено, что не все свойства допускают возможностью их изменения. К ним относятся: *ItemFolderPathDisplay*, *ItemType*, *ParsingPath*, *FileName*. При попытке

модификации такого свойства файл не завершает свою работу ошибкой, а работает в штатном режиме. Однако есть такие свойства, при изменении которых документ утратит свою работоспособность (например, свойство *ItemNameDisplay*).

Кроме модификации свойств файла изменениям подвергались его атрибуты: *Archive*, *Compressed*, *Device*, *Directory*, *Encrypted*, *Hidden*, *IntegrityStream*, *Normal*, *NoScrubData*, *NotContentIndexed*, *Offline*, *ReadOnly*, *ReparsePoint*, *SparseFile*, *System*, *Temporary*. Для изменения атрибута документа был разработан метод, позволяющий производить необходимые манипуляции.

Проверка хеш-суммы. Хеш вычислялся с использованием стандартных алгоритмов MD5 и SHA256. Для реализации проверки была использована библиотека C# *System.Security.Cryptography*. При этом вычисления были основаны на использовании разработанного метода, принимающего на вход в качестве параметра массив байтов, а затем из массива байтов формирующего строку. Значение атрибута устанавливается в *Archive*.

Установлено, что изменение любого из свойств документа влечет за собой изменение хеш– суммы, изменение атрибута документа не влияет на его хеш. Кроме того, переформатирование файла (DOCX, DOC, TXT) также не изменяет хеш.

Проверка объема файла. Для подтверждения корректности оценки объема файла при использовании модификаций их свойств использовались два метода: с помощью объекта класса *FileInfo* и с объекта класса *ShellFile*. Нами анализировались следующие текстовые свойства: *Comment*, *Copyright*, *FileDescription*, *FileVersion*, *FullText*, *IdentityProperty*, *InfoTipText*, *InternalName*, *ItemClassType*, *ItemFolderNameDisplay*, *ItemNamePrefix*, *ItemUrl*, *KindText*, *Language*, *MileageInformation*, *MIMEType*, *OriginalFileName*, *OwnerSid*, *ParentalRating*, *ParentalRatingsOrganization*, *ParsingName*, *PriorityText*, *Project*, *ProviderItemID*, *RatingText*, *SensitivityText*, *SoftwareUsed*, *SourceItem*, *Status*, *Subject*, *Title*, *Trademarks* – всего – 32.

В результате многочисленных опытов и сопоставительного анализа получены следующие основные результаты:

- свойства *Title*, *Language*, *Subject* типа *String* допускают запись в значение свойства без изменения размера документа до 444 символов латинского алфавита, знаков препинания и знаков пробелов;

- свойство *Comment* допускает запись 464 символов латинского алфавита, знаков препинания и пробелов без изменения размера документа;

- первое изменение значения всех системных свойств (кроме 4-х вышеперечисленных) влечет за собой изменение размера документа-контейнера, в котором производится такое изменение; причем увеличение размера документа происходит таким образом, что этот показатель увеличивается каждые 12–20 операций по модификации свойств: например, первое изменение свойства *Copyright* приводит к увеличению объема файла сразу увеличивается на 610 байт, затем изменения не наблюдаются в течении 12 модификаций, после чего объем файла возрастает в среднем на 250 байт;

- выявлена зависимость чувствительности объема файла при модификации некоторых свойств от алфавита, на основе которого генерируется осаждаемая информация: так, например, свойство *Comment* даже при размещении в нем 5000 арабских цифр не сказывается на объеме файла;

- любая модификация свойства на основе данных типа *Bool* приводит к изменению объема файла.

4 Обсуждение результатов

Предлагается использовать системные свойства текстовых документов-контейнеров в качестве среды для осаждения тайной информации стеганографическими методами, наряду с известным использованием собственно содержимого документа. Для документов, созданных на основе процессора MS Word, выявлено существование более 200 различных системных свойств, содержание многих из которых нельзя извлечь и проанализировать встроенными стандартными средствами операционной системы и используемого приложения MS Word.

К числу наиболее подходящих для использования в стеганографических приложениях следует отнести такие свойства фалов, как *Title*, *Language*, *Subject, Comment*, модификация которых путем размещения (осаждения) дополнительно до 50 байт информации типа *String* не приводит к увеличению объема файла. Важно также, что изменение любого атрибута файла (*Archive*, *Compressed*, *Device*, *Directory*, *Encrypted*, *Hidden* и др.) также не может выявлено в процессе стеганоанализа файла-контейнера на основе, например, сопоставления объемов и хешей исходного и модифицированного (с осажденной информацией) файлов.

Со всей очевидностью можно утверждать, что предложенный подход может быть реализован не только по отношению к файлам текстовых форматов.

Доступ к системным свойствам файлов и их атрибутов, а также анализ параметров свойств и атрибутов при выполнении описанных стеганографических операций производился с использованием авторского программного продукта [6].

Библиографические ссылки

[1] Колмаков М. В., Блинова Е.А. (2018). Особенности применения стеганографических методов в альтернативных потоках файловой системы NTFS. *Информационные технологии: тезисы докладов 82-й МНТК БГТУ*. БГТУ, Минск. С. 23–24.

[2] Сущеня А. А., Блинова Е.А., Урбанович П.П. (2018). Модификация стеганографического метода изменения междустрочного расстояния электронного документа. *Технические средства защиты информации: тезисы докладов XVI Белорусско-российской научно-технической конференции, Минск, 5 июня 2018 г.* БГУИР. Минск. С. 90.

[3] Урбанович П. П. (2016). *Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ*. БГТУ, Минск.

[4] Харин Ю. С., Берник В. И., Матвеев Г. В., Агиевич С. В. (2003). *Математические и компьютерные основы криптологии*. Новое Знание, Минск / Москва.

[5] Шутько Н. П., Романенко Д. М., Урбанович П. П. (2015). Математическая модель системы текстовой стенографии на основе модификации пространственных и цветовых параметров символов текста. *Труды БГТУ*. № 6 (179), С. 152–156.

[6] Юрашевич Д.Э., Урбанович П.П. *Прикладное программное обеспечение для сокрытия информации в текстовых документах*. Государственный реестр информационных ресурсов РБ. Регистрационное свид. № 1142022658 от 28.05.2020.

[7] Brassil J.T., Low S., Maxemchuk N.F., O'Gorman L. (1995). Electronic Marking and Identification Techniques to Discourage Document Copying. *IEEE Journal on Selected Areas in Communications*. V. 13. №. 8. P. 1495–1503.

[8] Shutko N. A., Urbanovich P.P., Zukowski P. (2018). A method of syntactic text steganography based on modification of the document-container aprosh. *Przegląd Elektrotechniczny*. R. 94, № 6. P. 82–85.

[9] Taleby M., Li Q., Jun Hou, Mazraeh H., Jing Zhang. (2018). AITSteg: An Innovative Text Steganography Technique for Hidden Transmission of Text Message via Social Media. *IEEE Access*. V.6. P. 65981–65995.

[10] Urbanovich P., Chourikov K., Rimorev A., Urbanovich N. (2010). Text steganography application for protection and transfer of the information. *Przegląd Elektrotechniczny*. R. 86, № 7. P.95–97.

[11] Urbanovich P., Shutko N. (2016). Theoretical Model of a Multi-Key Steganography System. *Recent Developments in Mathematics and Informatics. Contemporary Mathematics and Computer Science*. Vol. 2, Chapter 11. KUL, Lublin. P. 181–202.