УДК 004.056

СТРУКТУРА И ОПИСАНИЕ ПРОГРАММНОГО СРЕДСТВА РЕАЛИЗУЮЩЕГО СТЕГАНОГРАФИЧЕСКУЮ СИСТЕМУ НА ОСНОВЕ КОМБИНИРОВАНИЯ МЕТОДОВ ДЛЯ ЭЛЕКТРОННОГО ТЕКСТОВОГО ДОКУМЕНТА

Сущеня А.А., Блинова Е.А.
БГТУ, г. Минск, Беларусь, asuschenya@gmail.com, evgenia.blinova@belstu.by

Стеганография — это наука о способах передачи или хранения скрытой информации, при которых скрытый канал организуется на базе и внутри открытого канала с применением особенностей контейнера информации. Контейнер — это информация, обычно организованная в виде файла, предназначенная для сокрытия, или, иными словами, внедрения сообщения. Выбор вида контейнера оказывает существенное влияние на надёжность, максимальный размер внедряемой информации, а также возможность обнаружения факта передачи скрытого сообщения [1].

При использовании стеганографических методов защита информации происходит за счет выполнения трех условий:

- сокрытие факта передачи скрытой информации;
- сокрытие алгоритма осаждения данных в контейнер;
- сокрытие способа кодирования данных.

Учитывая факт развития вычислительной техники, а также частом использовании таких контейнеров как: аудиозапись, видеофайл, текст, изображение стеганография стремительно набирает популярность среди остальных методов защиты информации. Особый интерес к ней был вызван после того, как в ряде стран были введены ограничения на использование криптосистем.

Компьютерная стеганография основана на особенностях компьютерной платформы и использования специальных свойств компьютерных форматов данных [2-3]. Одним из эффективных контейнеров является файл формата DOCX [4-5]. Формат DOCX представляет собой модернизированную версию формата DOC, причем по сравнению со своим предшественником этот формат гораздо более популярен и доступен. В отличие от файлов DOC формат DOCX не является расширенным файловым форматом. Он представляет собой файл-архив. Формат файла основан на Open XML и использует сжатие по алгоритму *ZIP* для уменьшения размера файла [6].

В виду того, что синтаксис XML избыточен — это позволяет рассматривать данный формат в качестве стеганографического контейнера.

Зачастую XML используется скорее в качестве языка разметки, а не формата данных. При описании внешнего вида документа как правило используются атрибуты, что подразумевает наличие большого числа кавычек. Эта особенность позволяет при помощи определенного алгоритма разместить в файле XML информацию, никак не влияющую на семантику документа [4].

Наряду с внедрением информации в XML составляющую DOCX документа производить осаждение можно в пробельные элементы самого текста, путем изменения смещения символа вверх или вниз.

Одновременное использование двух упомянутых методов позволяет помимо скрытой передачи сообщения, иметь возможность проверки достоверности переданного сообщения. Новизна представленного метода, а также реализующего его программного продукта заключается в том, что в зависимости от содержимого контейнера (количества пробелов и

кавычек) для внедрения сообщения выбирается тот, который позволяет внедрить наибольшее количество информации.

В качестве примера для демонстрации возможности осаждения тайной информации в DOCX контейнер, было создано программное средство «SpaceQuoteStego».

Блок-схема алгоритма осаждения скрытого сообщения в файл контейнер изображен на рисунке 1. Обозначим метод изменения межстрочного расстояния для неотображаемых символов как S, а метод замены типа кавычек с двойных на одинарные — Q.

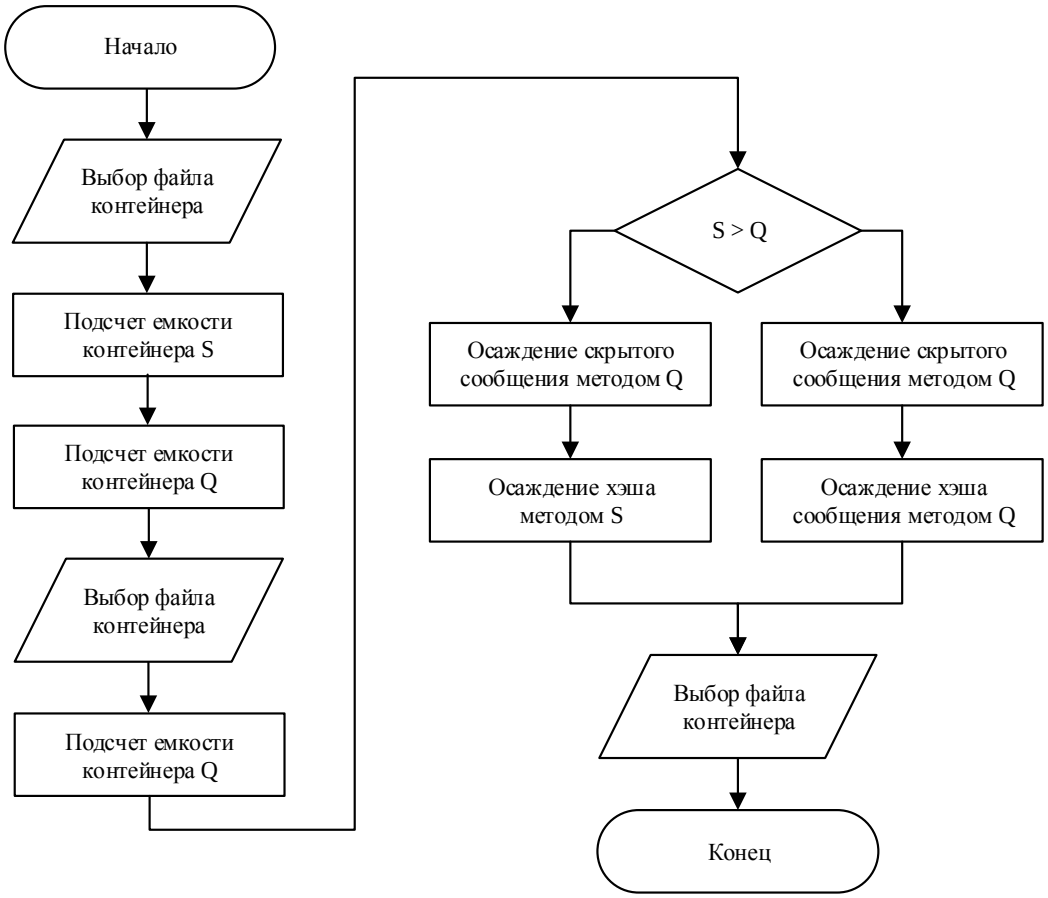

Рисунок 1 — Блок-схема алгоритма осаждения скрытого сообщения в файл контейнер

Программное средство реализовано при помощи технологии Windows Forms и с использованием языка программирования C#. Структура файлов приложения SpaceQuoteStego представлена на рисунке 2. Классы моделей которые являются абстракцией над реальными документами находятся в папке Models. Класс *DOCXFile.cs* — это абстрактная модель реального документа формата DOCX, включающая в себя сущность *XMLFile*, а также методы для подсчета емкости контейнера. В папке Stego расположены классы, реализующие внедрение\извлечение конфиденциальной информации в DOCX-контейнер, используя метод изменения межстрочного интервала. Папка Util содержит вспомогательные сущности реализующие процедуры конвертации сообщения в двоичный вид, хеширования сообщения, а также набор констант задающих основные параметры методов осаждения и извлечения.

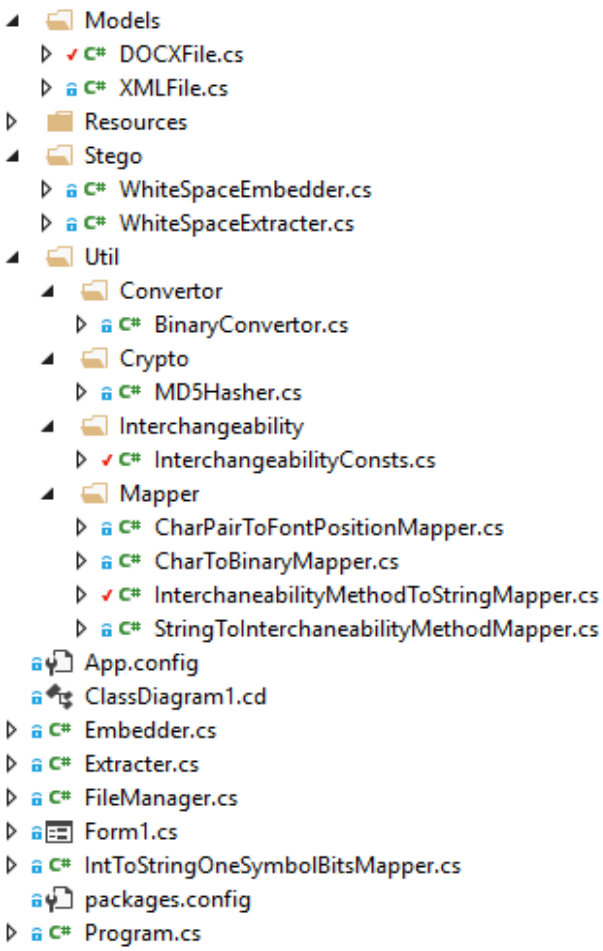

Рисунок 2 — Структура файлов приложения *SpaceQuoteStego*

Основными сущностями, реализующими процесс взаимозаменяемости методов во время осаждения, являются: перечисление *InterchangeabilityMethod*, а также классы *InterchaneabilityMethodToStringMapper* и *StringToInterchaneabilityMethodMapper*.

Перечисление *InterchangeabilityMethod* содержит два элемента: «QUATES» и «SPACES» (Листинг 1).

```
public enum InterchangeabilityMethod
{
    QUATES,
    SPACES
}
```

Листинг 1 — Программная реализация перечисления *InterchangeabilityMethod*

Класс *InterchaneabilityMethodToStringMapper* содержит структуру данных *Dictinary* позволяющую получить соответствующий кодовый бит используемого метода для внедрения его в секретное сообщение, а также его хэш (Листинг 2).

```
internal class InterchaneabilityMethodToStringMapper
{
    public static Dictionary<InterchangeabilityConsts.InterchangeabilityMethod, string> mapper
        = new Dictionary<InterchangeabilityConsts.InterchangeabilityMethod, string>
        {
            { InterchangeabilityConsts.InterchangeabilityMethod.QUATES, "0" },
            { InterchangeabilityConsts.InterchangeabilityMethod.SPACES, "1" }
        };
}
```

Листинг 2 — Программная реализация перечисления *InterchaneabilityMethodToStringMapper*

Класс *StringToInterchaneabilityMethodMapper* содержит структуру данных *Dictinary* позволяющую получить используемый метод осаждения на основании извлеченного контрольного бита.

Данный подход используется для обратного преобразования сообщения и его хэша из бинарной последовательности. Соответствующий контрольный бит внедряется в начало сообщения, а также его хэша представленных в двоичном виде. В процессе извлечения контрольный бит удаляется, после чего происходит обратное преобразование бинарных последовательностей.

В процессе внедрения информации для определения оптимального метода встраивания сообщения используется метод *GetContainerEmbeddingMethod* (Листинг 3). Данный метод на основе данных о количестве кавычек в файле document.xml, а также о количестве пробелов в тексте документа определяет в какой элемент можно внедрить больше информации.

```
public InterchangeabilityConsts.InterchangeabilityMethod GetContainerEmbeddingMethod()
{
    int numberOfQuates = (int)document.GetContainerCapacity(BITS_PER_SYMBOL);
    if (numberOfQuates > WhiteSpacesNubmer)
    {
        return InterchangeabilityConsts.InterchangeabilityMethod.QUATES;
    }

    return InterchangeabilityConsts.InterchangeabilityMethod.SPACES;
}
```

Листинг 3 — Программная реализация метода *GetContainerEmbeddingMethod*

Метод *GetContainerEmbeddingMethod* в зависимости от структуры контейнера возвращает один из элементов перечисления *InterchangeabilityMethod* который указывает на используемый метод осаждения сообщения. Выбор метода основывается на количестве символов возможных для осаждения.

После того, как определен метод осаждения можно произвести подсчет количества символов возможного для осаждения в контейнере. Получить это количество можно используя метод *GetDOCXContainerCapacity* (Листинг 4). Принимая сообщение, метод *GetDOCXContainerCapacity* на основании известного метода осаждения вычисляет разницу между доступным количеством символов для осаждения в контейнере и длиной переданного сообщения.

```
public int GetDOCXContainerCapacity(string message)
{
    if (this.GetContainerEmbeddingMethod() ==
        InterchangeabilityConsts.InterchangeabilityMethod.QUATES)
    {
        return (int)document.GetContainerCapacity(BITS_PER_SYMBOL) - message.Length;
    }
    else if (this.GetContainerEmbeddingMethod() ==
        InterchangeabilityConsts.InterchangeabilityMethod.SPACES)
    {
        return WhiteSpacesNubmer - message.Length;
    }

    return 0;
}
```

Листинг 4 — Программная реализация метода *GetDOCXContainerCapacity*

Метод *GetDOCXContainerCapacity* на основе выбранного метода, а также размера передаваемого сообщения возвращает целое число означающее число символов возможных для осаждения в стеганографический контейнер.

ЛИТЕРАТУРА

1. Урбанович П.П. Защита информации методами криптографии, стеганографии и обфускации. – Минск : БГТУ, 2016, – 220 с.
2. Сущеня, А. А. Стеганографическое преобразование текстов-контейнеров на основе языков разметки / А. А. Сущеня // 68-я научно-техническая конференция учащихся, студентов и магистрантов, 17-22 апреля, Минск : сборник научных работ : в 4 ч. Ч. 4 / Белорусский государственный технологический университет. - Минск : БГТУ, 2017. - С. 145-149.
3. Сущеня, А. А. Программное средство стеганографического преобразования текстов-контейнеров на основе языка разметки XML / А. А. Сущеня // 69-я научно-техническая конференция учащихся, студентов и магистрантов, 2-13 апреля, Минск : сборник научных работ : в 4 ч. Ч. 4 / Белорусский государственный технологический университет. - Минск : БГТУ, 2018. - С. 81-84.
4. Сущеня, А.А. Способ стеганографического осаждения информации в документ с расширением .DOCX / А. А. Сущеня // XXI Республиканская научная конференция студентов и аспирантов, 19–21 марта, Гомель: сборник научных работ / Гомельский государственный университет имени Ф. Скорины. – С. 303-304.
5. Сущеня, А.А. Идея и архитектура веб-приложения, использующего в качестве стеганографического контейнера документы формата DOCX / А. А. Сущеня // Международная научно-практическая конференция, 14–18 мая, Минск: сборник научных работ / Белорусский государственный университет. – С. 170.
6. Сущеня, А.А., Блинова Е.А., Урбанович П.П. Модификация стеганографического метода изменения междустрочного расстояния электронного документа // Технические средства защиты информации: Тезисы докладов XVI Белорусско-российской научно-технической конференции, 5 июня 2018 г., Минск. Минск: БГУИР, 2018. – С 90-91.