

РАЗРАБОТКА ПРОГРАММЫ ДЛЯ ЗАЩИТЫ ТЕКСТА МЕТОДОМ ДЕКОРАТИВНОЙ ОБФУСКАЦИИ

Сабирова Резеда Рафисовна

студент, Набережночелнинский институт (филиал) ФГАОУ ВО К(П)ФУ, РФ, г. Набережные Челны

На сегодняшний день потребность в защите интеллектуальной собственности, в частности программных продуктов, очень высока. Исходя из этого, возникает задача разработки наиболее эффективного метода защиты для того или иного программного продукта.

Одним из таких методов защиты является обфускация. Обфускация или запутывание кода – приведение исходного текста или исполняемого кода программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции [1].

Помимо термина "обфускация" существует словосочетание "декоративная обфускация". Разница между ними заключается в том, что декоративная обфускация рассматривается применительно к исходному тексту программы и предполагает выведение текста в виде некоторых фигур. В данной работе будет рассмотрен именно этот метод обфускации. В качестве объекта исследования используется любой текст или программный код.

Реализация приложения

Обфускатор реализован в виде приложения, написанного на языке C++. В качестве среды программирования используется Visual Studio.

Работу программы можно разделить на несколько этапов. Первый этап – загрузка текста из файла в поле ввода.

Для загрузки текста из файла в программе реализован метод `button1_Click`. Чтение текста организовано из файла `.txt` формата.

При нажатии на кнопку «Загрузить из файла» открывается диалоговое окно для открытия документа и, после выбора нужного текстового файла, этот файл записывается в поле ввода `richtextbox1`.

Функции `LoadFile` загружает в элемент управления `richtextbox1` определенный тип файла. Первый аргумент задает имя и расположение файла, который будет загружен в элемент управления, второй – задает тип входного потока, который используются для загрузки и сохранения данных в `RichTextBox`, а именно поток открытого текста с пробелами в местах объектов внедрения.

Обфускация текста выполняется на следующем этапе работы. Она реализована в функции `button2_Click`.

Алгоритм обфускации следующий:

1. Получается общее количество символов в тексте, при этом выполняется преобразование строки: знаки перехода на новую строку `"\n"` заменяются на знак `"#"`.

2. Производится расчет статических переменных:

l_1

- количество символов в строке,

l_2

- количество строк.

l_1 l_2

3. В строку добавляются дополнительные символы "#". Это делается для того, чтобы l_1 и l_2 были четными.

4. Формирование ключей. Ключ - это информация, необходимая для беспрепятственного шифрования и дешифрования текстов. Формирование ключей выполняется с помощью объекта класса Random. Этот класс представляет собой генератор псевдослучайных чисел, то есть устройство, которое выдает последовательность чисел, отвечающую определенным статистическим критериям случайности.

Созданный ключ должен удовлетворять следующим условиям:

l_1 l_2

· не должен превышать l_1 и l_2 ;

l_1 l_2

· должен без остатка делить l_1 и l_2 (например, для $l_1 = 6$ $l_2 = 6$

и $l_1 = 3$ $l_2 = 6$), ключ может быть

равен 2, 3 или 6).

```
//определяется случайный ключ
Random^rnd = gcnew Random();
int l=l1; if (l1 > l2) { l = l2; }
while (l1%k1 != 0 || k1 == 1)
{
    k1 = rnd->Next(2, l + 1);
    if (l2%k1 != 0) { k1 = 1; }
}k2 = k1;
//Вывод ключа
textBox1->Text = k1.ToString(); // +k2.ToString();
```

5. С помощью полученного ключа исходный текст делится на блоки. По факту, ключ представляет собой размер исходного квадрата. Далее выполняется перестановка символов в квадратах. Перестановка осуществляется следующим образом: содержимое квадратного блока переворачивается вокруг главной диагонали (рисунок 1).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}$$

Рисунок 1. Перестановка элементов блока

6. Вывод текста. После перестановки символов текст выводится в шахматном порядке в поле вывода richTextBox2.

7. Последний этап работы программы – сохранение обфусцированного текста в файл. Сохранение осуществляется в методе button3_Click, в результате текст из поля вывода richTextBox2 сохраняется в файл .txt формата.

Структурно окно приложения состоит из следующих компонентов: полей для вывода исходного и обфусцированного текста, поля для вывода ключа, кнопок для загрузки и сохранения текстового файла и кнопки обфускации. Графическое окно приложения представлено на рисунке 2.

Тестирование обфускатора

Для того, чтобы убедиться в правильности работы созданного приложения, необходимо протестировать его. Для проверки корректности работы проведены несколько тестов. Результаты показаны на рисунках 2-4.

По результатам теста видно, что в ходе работы приложения не было выявлено ошибок. Поэтому можно считать, что получен корректно работающий обфускатор.

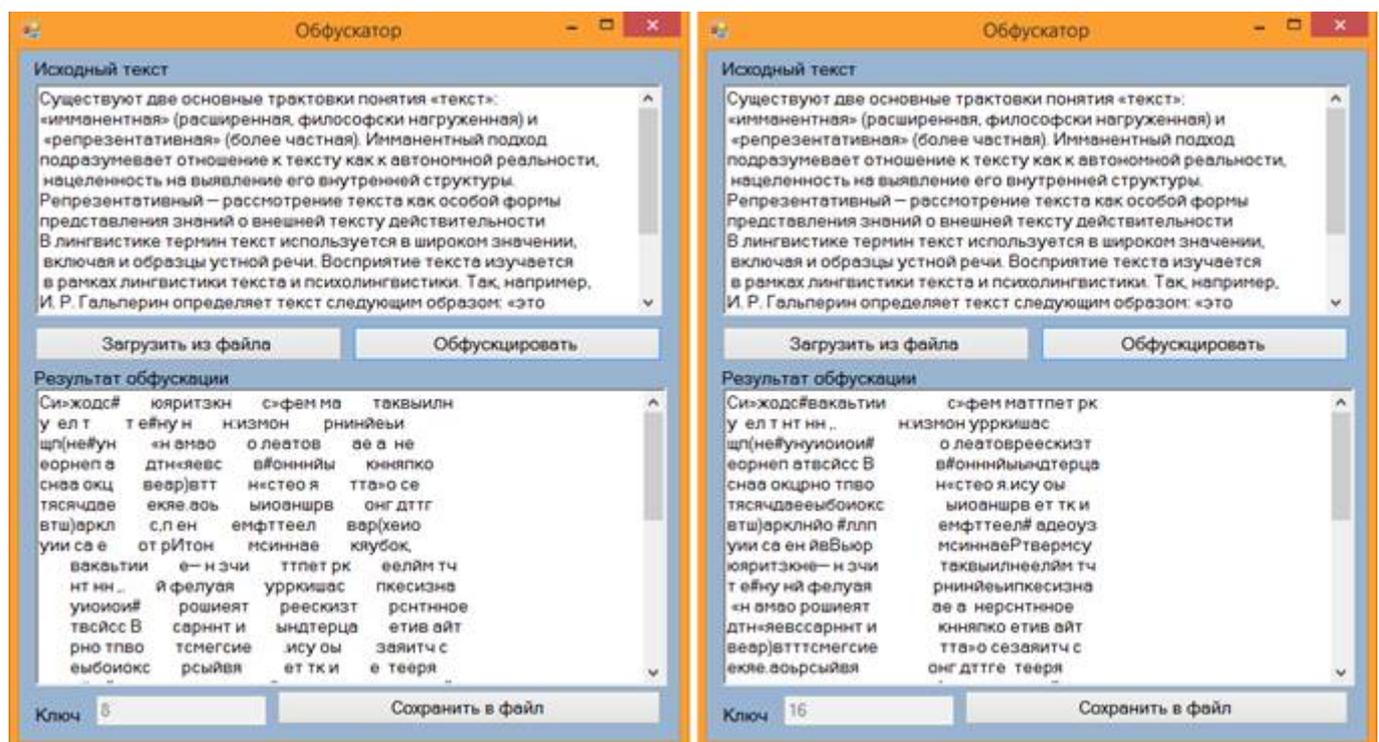


Рисунок 2. Текст, обфусцированный при ключах, равных 8 и 16

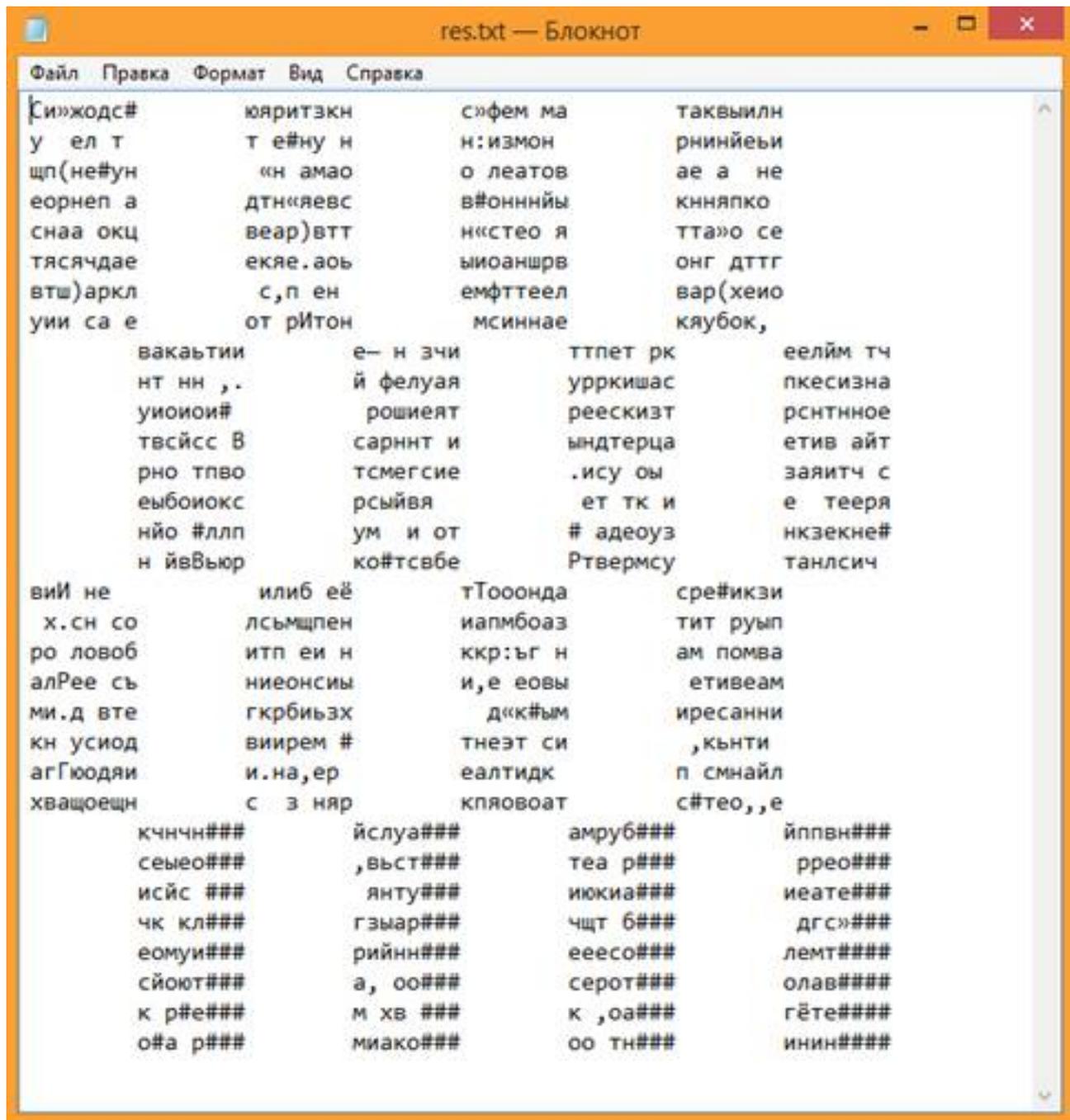


Рисунок 3. Обфусцированный текст, сохраненный в файл .txt формата

В процессе выполнения данной работы был разработан алгоритм декоративного обфусцирования, в результате которого исходный текст выводится в шахматном порядке и разработаны приложения, выполняющие обфускацию и деобфускацию текста. Приложение деобфускации выполняет преобразование обфусцированного текста в исходный вид. Алгоритм деобфускации обратный алгоритму обфускации.

Практическая часть работы может быть использована в учебных и исследовательских целях, посвящённых теме защиты программ от исследования и анализа и разработки запутывающих преобразований программ.

Список литературы:

1 Лифшиц Ю. М. Запутывание (обфускация) программ. СПб.: СПб. отд. Мат. инст. им. В.А. Стеклова РАН, 2004, - 50 с.

2 Чернов А. В. Исследование и разработка методологии маскировки программ: дис. ... канд. физ.-мат. наук / А. В. Чернов ; Моск. гос. ун-т им. М. В. Ломоносова. — М., 2010. — 139 с.