РАЗРАБОТКА СИСТЕМЫ СКРЫТОГО ХРАНЕНИЯ КОНФИДЕНЦИАЛЬНОЙ ИНФОРМАЦИИ В ОБЛАЧНЫХ ХРАНИЛИЩАХ

Мартышкин А.И., Плахина Л.Н., Лобов Р.А.

ФГБОУ ВО «Пензенский государственный технологический университет», Пенза, e-mail: alexey314@yandex.ru

В настоящей статье рассматривается процесс создания и вариант программной реализации прототипа системы, предназначенной для скрытого хранения конфиденциальной информации в облачных хранилищах, представляющих собой виртуальный носитель информации, который хранит и обрабатывает данные на многочисленных серверах, разбросанных во всемирной паутине. Целью работы является создание прототипа настольного приложения для операционной системы Microsoft Windows, обеспечивающего взаимодействие пользователя с файловой системой локального компьютера и облачного сервиса хранения данных Google Drive, а также шифрование/дешифрование конфиденциальной пользовательской информации и механизм загрузки и выгрузки зашифрованных файлов с помощью официального инструмента компании Google. Авторами проводится комплексный анализ предметной области и выявление ключевых достоинств и недостатков современных аналогов системы. Разрабатываются и описываются алгоритмы шифрования/дешифрования информации в графический файл. Создается прототип системы, предназначенный для скрытого хранения конфиденциальной информации в облачных хранилищах. Разработанное программное обеспечение функционирует корректно и без ошибок, все элементы интерфейса интуитивно понятны и просты в понимании. В результате проделанной авторами работы было проведено контрольное тестирование, которое показало, что в сравнении с современными существующими аналогами максимально реализованы их ключевые достоинства и учтены все недостатки.

Ключевые слова: облачное хранилище данных, стеганография, шифрование / дешифрование файлов, алгоритмы шифрования/дешифрования, графический интерфейс, языки программирования, прототип системы, интерфейсное и функциональное тестирование

DEVELOPMENT OF A SYSTEM FOR HIDDEN STORAGE OF CONFIDENTIAL INFORMATION IN CLOUD STORAGE

Martyshkin A.I., Plakhina L.N., Lobov R.A.

Penza State Technological University, Penza, e-mail: alexey314@yandex.ru

This article discusses the process of creating and implementing a software prototype of a system designed for hidden storage of confidential information in cloud storage, which is a virtual storage medium that stores and processes data on numerous servers scattered on the world wide web. The purpose of this work is to create a prototype of a desktop application for the Microsoft Windows operating system that provides user interaction with the file system of the local computer and the Google Drive cloud storage service, as well as encryption/decryption of confidential user information and a mechanism for downloading and uploading encrypted files using the official Google tool. The authors conduct a comprehensive analysis of the subject area and identify the key advantages and disadvantages of modern analogues of the system. Algorithms for encrypting/decrypting information into a graphic file are developed and described. A prototype of the system is being created for the hidden storage of confidential information in cloud storage. The developed software functions correctly and without errors, all interface elements are intuitive and easy to understand. As a result of the work done by the authors, a control test was conducted, which showed that in comparison with modern existing analogues, their key advantages were maximally realized and all the disadvantages were taken into account.

Keywords: cloud data storage, steganography, file encryption / decryption, encryption/decryption algorithms, graphical interface, programming languages, system prototype, interface and functional testing

Проблема передачи информации в настоящее время является актуальной и требует разработки специальных средств для скрытого хранения конфиденциальной информации в облачных хранилищах и обеспечения качественного информационного обмена между локальными устройствами хранения и облачным хранилищем [1].

Облачное хранилище данных – виртуальный носитель информации, который хранит и обрабатывает данные на многочисленных серверах, разбросанных во всемирной паутине. В наше время подобного рода хранилища получили большое распространение, и, на сегодняшний день, их количество превысило несколько сотен [2].

Основным достоинством облачных технологий является безопасность [3, 4]. Одним из наиболее эффективных способов обеспечения безопасности облачных технологий является стеганография [5, 6]. Стеганография – это междисциплинарная наука и искусство передавать сокрытые данные, внутри других, не сокрытых данных.

Цель исследования

Создание прототипа настольного приложения для операционной системы Microsoft Windows, обеспечивающего взаимодействие пользователя с файловой системой локального компьютера и облачного сервиса хранения данных Google Drive, шифро-

вание и дешифрование конфиденциальной пользовательской информации и механизм загрузки и выгрузки зашифрованных файлов в облачное хранилище Google Drive с помощью официального инструмента компании Google.

Материалы и методы исследования

Основными задачами исследования являются:

– комплексный анализ предметной области и выявление ключевых достоинств и недостатков современных аналогов;

– разработка алгоритмов шифрования и дешифрования с реализацией взаимодействия с облачными сервисами;

– программная реализация прототипа системы, предназначенной для скрытого хранения конфиденциальной информации в облачных хранилищах и обеспечения информационного обмена между локальными устройствами хранения и облачными хранилищами с использованием Google Drive API для взаимодействия с облачными сервисами;

– тестирование и верификация системы.

Итак, перейдем к решению первой задачи: проанализируем достоинства и недостатки существующих аналогов системы.

Приведенная ниже таблица составлена на основании анализа информации, представленной на официальных веб-сайтах.

Из результата сравнения становится ясно, что в большинстве аналогов отсутствует локализация продукта для российского рынка, также большинство продуктов платны (некоторые даже ведут довольно жесткую лицензионную и ценовую политику). Все эти продукты не имеют кроссплатформенного автономного приложения с графическим интерфейсом пользователя.

Устранение недостатков существующих систем позволило выделить конкретные задачи при разработке нашего прототипа системы:

– применить при разработке наиболее современные языки программирования для достижения кроссплатформенности;

– улучшить техническую поддержку пользователей и поддержку мобильных приложений;

– усовершенствовать безопасное предоставление сторонним лицам доступа к своим зашифрованным файлам.

Практическая реализация системы должна отвечать следующим требованиям.

Система должна:

– повысить безопасность хранения данных в облачных хранилищах путем применения стеганографии;

– быть кроссплатформенной. Для достижения кроссплатформенности продукт следует разрабатывать на таких языках, как «Javascript», «Python» и т.д.;

– иметь интуитивно понятный интерфейс:

– иметь двухфакторную аутентификацию входа для учетной записи;

– обеспечиваться постоянной поддержкой пользователей;

– быть локализирована для российского рынка.

Результаты сравнения вариантов шифрования данных в облаках для Windows

Параметр	Проприетарные программы	Порт encfs для Windows – encfs4win	Duplicati	Webdav клиент CarotDav
Платность	Да	Нет	Да	Да
Кроссплатформенность	Нет	Нет	Нет	Да
Русская локализация	Нет	Нет	Нет	Нет
Поддержка мобильных приложений	Да	Нет	Да	Да
Шифрование имени файлов	Да	Да	Да	Да
Безопасное предоставление сторонним лицам доступа к своим зашифрованным файлам	Да	Нет	Нет	Да
Производительность (по десятибалльной шкале)	5	4	10	7

Система должна представлять удобный пользовательский интерфейс для выполнения операций копирования файлов и папок с локальных устройств хранения данных в облачные хранилища (iCloud, Google Drive, OneDrive) с реализацией прозрачного стенографического преобразования (сокрытия) пользовательской информации и её обратного преобразования (выделение) при копировании из облачных хранилищ на локальные устройства хранения. В качестве контейнеров для хранения пользовательской информации могут использоваться графические файлы.

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако имеет высокую производительность, простоту, защищенность. Программная реализация более практична, допускает известную гибкость в использовании.

Результаты исследования и их обсуждения

В процессе программирования начального функционала программы ознакомились с интерфейсом среды разработки программного обеспечения и с набором применяемых инструментов.

Нами были описаны все ключевые элементы интерфейса и логика работы при нажатии или альтернативном взаимодействии пользователя с ними. После чего, необходимо было описать всю последовательность выполнения методов непосредственно работы с Google Drive API и логику шифрования или дешифрования файлов.

После запуска программы управление переходит в исполняемый файл «App.xaml». Данный файл определяет главный класс программы и коллекцию пользовательских стилей для объектов XAML.

После чего переход программы сводится непосредственно к исполняемому файлу «windowMain.xaml.cs». Далее происходит прорисовка главного окна «windowMain» разрабатываемой программы и инициализация всех элементов интерфейса.

Программа разрабатывается на основе графического интерфейса «WPF». «WPF» или «Windows Presentation Foundation» является новейшим пакетом графического интерфейса для платформы «.NET», разработанной компанией Microsoft [7]. Данный графический интерфейс позволяет создавать приложения с широким спектром графических элементов для работы с пользователем, такие как ярлыки, текстовые поля, контейнеры для отображения мультимедийных объектов и др.

Для придания программе современного вида, запоминаемости и интереса, некоторые объекты были снабжены изображениями, отображающими их внешний вид.

Всего для разрабатываемой программы было использовано:

– 8 графических файлов, формата «.png»;

– 1 файл-шрифт, формата «.ttf».

В качестве среды программирования нами была выбрана Microsoft Visual Studio. Данный программный продукт создан компанией Microsoft и обладает всеми современным средствами для специалистов области программирования [8].

Данная среда содержит обширный инструментарий для работы с разными языками программирования, включая C#, C++ и СИ. Имеет возможность подключать сторонние библиотеки и работать с ними. А также за все время существования программный продукт накопил качественный комплексный справочник по API для работы с инструментами – библиотеку MSDN.

Компания Google выпустила API Drive для множества языков программирования. Одно из самых значимых преимуществ этого API является то, разработчик может указать приложению сохранять данные в пользовательском Drive. Таким образом, при работе с любой программой, пользователь сможет сохранять данные на диске, которые после этого будут доступны на любых других устройствах. К примеру, можно работать с документами в поезде при помощи планшета, а потом приехать в отель и воспользоваться ноутбуком, не потеряв при этом сохраненных на планшете данных.

Прежде чем приступить к реализации идеи проекта, перед нами стоял выбор – какой язык программирования использовать. Наш выбор пал на C#.

Язык программирования C# – это высокоуровневый язык программирования, разработанный и, до сих пор, активно поддерживающийся компанией Microsoft [9, 10, 11]. Язык программирования C# на сегодняшний день является активно используемым в многочисленных сферах программной индустрии. На нем можно реализовать любые программные продукты для разных платформ, таких как Android, IOS и Линукс [12, 13].

Одним из решающих факторов нашего выбора является наличие подходящего инструментария для реализации проекта, которым является Google Drive API.

В случае с программным продуктом, реализующем алгоритмы шифрации конфиденциальной информации и работающей с облачными сервисами очень важно провести тестирование алгоритма шифрации,

а именно входных и выходных данных, убедиться, что вся необходимая информация шифруется и дешифруется корректно.

При первом запуске программы появляется стартовое окно разрабатываемого приложения.

По реализуемой логике программы, сначала можно выполнить соединение с хранилищем Google Drive, после этого скачать или загружать файлы в само хранилище, либо выполнить непосредственно шифрование или дешифрование информации без авторизации на Google, но при этом работая только с локальными файлами. После начального теста можно убедиться в том, что все работает корректно.

Тестирование механизма авторизации показало, что процесс происходит так, как и было задумано, где после нажатия на соответствующие кнопки открывается окно браузера с сайтом авторизации в Google: в рабочем каталоге программы появляются соответствующие файлы и, сама программа стабильно работает с Google Drive API.

Далее необходимо протестировать взаимодействие с Google Drive API. Для этого нами было выполнено скачивание и загрузка самих файлов изображения. Все тесты показали успешный результат.

Далее нами было выполнено тестовое сохранение и открытие локальных файлов изображения, хранящиеся на пользовательском персональном компьютере. Данные механизмы реализованы стандартными механизмами Windows и .NET. После проведения теста нами не было выявлено ошибок.

Самым важным шагом в процессе тестирования является проверка шифрации и дешифрации файлов. В качестве исходных файлов были выбраны графические файлы самых популярных форматов «img_1.png» и «img_2.jpg». В результате тестирования шифрования программа создает временный файл в директории программы, после чего он удаляется.

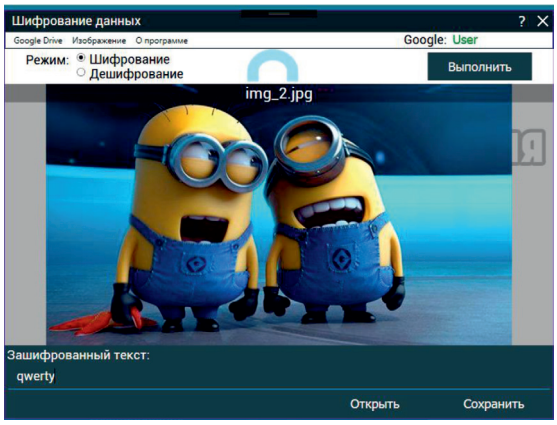

Рис. 1. Ввод текста для шифрования

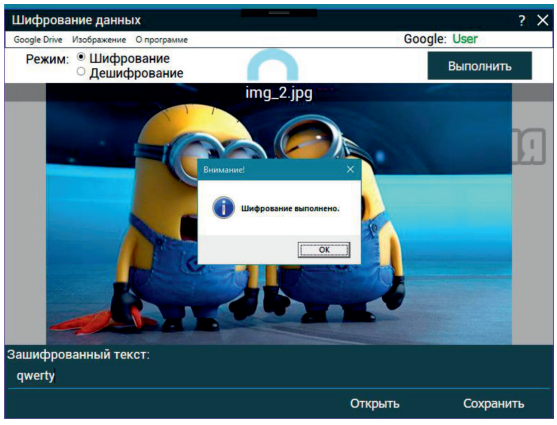

Рис. 2. Шифрование выполнено

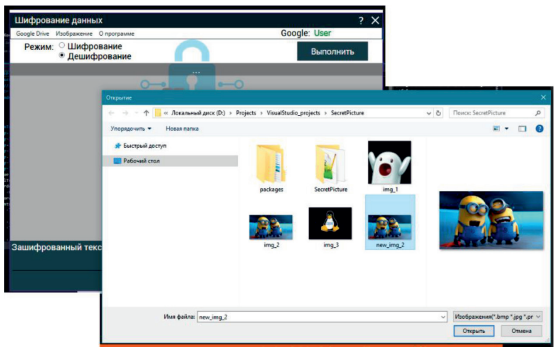

Рис. 3. Выбор файла для дешифрования

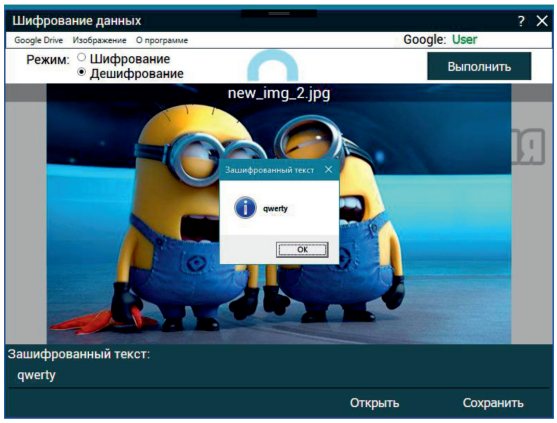

Рис. 4. Дешифрование выполнено

После тестирования было проверено сообщение, которое было зашифровано, оно совпало с исходным текстом. На рис. 1-4 показаны процессы шифрации и дешифрации рисунка файла «img_2.jpg».

Заключение

Нами было проведено контрольное тестирование разработанной программы, которое включало в себя интерфейсное и функциональное тестирование. Оно показало, что в сравнении с современными существующими аналогами в программе максимально реализованы их ключевые достоинства и учтены все недостатки. Программное обеспечение работает полностью корректно и без ошибок, все элементы интерфейса интуитивно понятны и читабельны.

Список литературы

1. Горев А.И., Симаков А.А. Обеспечение информационной безопасности. М.: Мир, 2005. 844 с.

2. Риз Д. Облачные вычисления. БХВ-Петербург, 2014. 191 с.

3. Котяшичев И.А., Бырылова Е.А. Защита информации в «Облачных технологиях» как предмет национальной безопасности // Молодой ученый. 2015. №64. С. 30–34.

4. Шнайер Брюс. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си: монография. М.: Триумф, 2012. 816 с.

5. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. М.: Солон-Пресс, 2002. 272 с.

6. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. М.: МК–Пресс, 2006. 288 с.

7. Мэтью Мак-Дональд. WPF: Windows Presentation Foundation в .NET 3.5 с примерами на C# 2008 для профессионалов. М.: «Вильямс», 2008. 928 с.

8. Петцольд Ч. Программирование для Microsoft Windows 8. 6-е издание. Питер, 2014. 1008 с.

9. Ликнесс Д. Приложения для Windows 8 на C# и XAML. Питер, 2013. 386 с.

10. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. Язык программирования C#. Классика Computers Science. 4-е издание. Питер, 2012. 784 с.

11. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е издание, 2018. 896 с.

12. Стиллмен Э., Грин Д. Head First. Изучаем C#. 3-е издание. Питер, 2013. 816 с.

13. Албахари Д., Албахари Б. C# 6.0. Справочник. Полное описание языка. М.: «Вильямс», 2018. 1040 с.