УДК 004.021

РЕАЛИЗАЦИЯ КРИПТОСИСТЕМЫ, ОСНОВАННОЙ НА АУТЕНТИФИКАЦИИ ЛИЧНОСТИ ПО ЛИЦУ И ФЛЭШ КАРТЕ, С ИСПОЛЬЗОВАНИЕМ СЛУЖБЫ И МЕЖПРОЦЕССНОГО ВЗАИМОДЕЙСТВИЯ

Н. А. ГУРЕЦКИЙ, Д. В. ПЯТКИН, канд. физ.-мат. наук, доц. О. В. ГОЛУБЕВА
(Полоцкий государственный университет)

В статье представлен практический способ создания надёжной криптосистемы с нестандартным способом хранения пароля и аутентификации. Целью работы явилось написание программы для защиты пользовательских файлов с использованием флэш карты в качестве пароля и аутентификации пользователя по лицу. Задача решалась путём разбиения основной программы на библиотеки и вспомогательные подпрограммы. Программа написана на языке программирования C# с использованием Win32 API и библиотеки распознавания OpenCV.

Ключевые слова: безопасность хранения информации, криптосистема, алгоритм шифрования, пароль, ключ, метод хранения пароля, аутентификация

DEVELOPMENT OF THE CRYPTOSYSTEM, BASED ON AUTHENTICATION OF THE PERSONALITY ON THE PERSON AND FLASHCARD, WITH USE OF SERVICE OF CROSS-PROCESS INTERACTION

N. GURETSKI, D. PYATKIN, O. GOLUBEVA

The practical way of creation of reliable cryptosystem with a non-standard way of storage of the password and authentication is presented in article. The purpose of work was writing of the program for protection of the user files with use a flash card as the password and authentication of the user on the person. The problem was solved by splitting the main program into libraries and auxiliary subprogrammes. The program is written in the C# programming language with use of Win32 API and library of recognition OpenCV.

Введение. В настоящий момент самым распространённым способом хранения информации является запись её на цифровые носители. Часть сохраняемой информации может быть конфиденциальной, приватной, секретной и нуждаться в защите. У такой информации есть законные пользователи. Но всегда существует вероятность появления пользователей незаконных, стремящихся захватить секретную информацию с целью обращения её себе во благо. Хакеры ежедневно крадут номера кредитных карт, банковские счета и

даже личность человека. Поэтому законные пользователи стремятся защитить цифровые носители информации от несанкционированного доступа.

На сегодняшний день известны несколько подходов к проблеме защиты информации, хранящейся в персональном компьютере. Один из них – шифрование данных. В статье пойдёт речь о защите пользовательских файлов на персональном компьютере с применением шифрования и нестандартного метода хранения пароля, исключающего человеческий фактор. При этом подходе злоумышленник, даже завладев данными, воспользоваться ими без знания ключа и алгоритма шифрования не сможет. Разработанный программный продукт предназначен для быстрого и надёжного шифрования файлов, дешифрования, открытия без возможности внесения изменений, сокрытие этих файлов стандартными средствами Windows; использование надёжного ключа шифрования, который не знает даже сам пользователь!

Принцип разработанного алгоритма защиты информации и его функционал. В качестве алгоритма шифрования использован алгоритм Triple DES (3DES). Этот алгоритм, созданный Уитфилдом Диффи, Мартином Хеллманом и Уолтом Тачманном в 1978 году на основе алгоритма DES, представляет собой симметричный блочный шифр. Цель создания Triple DES – устранение главного недостатка алгоритма DES: недостаточной длины ключа (56 бит), который можно взломать полным перебором. Скорость работы 3DES в 3 раза ниже, чем у DES, но криптостойкость намного выше. Время, требуемое для криптоанализа 3DES, может быть в миллиард раз больше, чем время, нужное для вскрытия DES [1].

Для генерации пароля, а потом и для дальнейшей аутентификации, пользователю необходимо использовать USB флэш карту – первый уровень защиты. Также в первоначальной настройке программы пользователю необходимо сделать фотографию своего лица для его дальнейшего распознавания – второй уровень защиты. Из логина и данных флэш карты (каждая флэш карта обладает уникальным сочетанием данных, вшитых в плату) получаем хеш длиной 1024 бит – он же и есть пароль для шифрования. В алгоритме 3DES в качестве пароля используется лишь 192 бит, поэтому хеш специальным алгоритмом урезаем до 192 бит. При аутентификации сверяться будут не пароли, а одно и то же слово, зашифрованное ими, причем правильный вариант будет храниться в защищенном разделе реестра. Пользователю необходимо лишь один раз после полного включения компьютера вставить нужную флэш карту и поставить в соответствие лицо и видеокамеру, после чего пароль будет храниться в стэке приложения.

Для удобства использования этой криптосистемы процесс аутентификации пользователя и хранение пароля вынесен в отдельную службу Windows. Службы операционной системы Windows (англ. Windows Service, службы) – приложения, автоматически (если настроено) запускаемые системой при запуске Windows и выполняющиеся вне зависимости от статуса пользователя. Имеет общие черты с концепцией демонов в Unix [2].

Для общения службы с приложением дешифрования и открытия файлов необходимо использовать межпроцессорное взаимодействие. Межпроцессорное взаимодействие (англ. Inter-Process Communication, IPC) – набор способов обмена данными между множеством потоков в одном или более процессах. Процессы могут быть запущены на одном или более компьютерах, связанных между собой сетью. IPC-способы делятся на методы обмена сообщениями, синхронизации, разделяемой памяти и удаленных вызовов (RPC). Методы IPC зависят от пропускной способности и задержки взаимодействия между потоками и типа передаваемых данных.

IPC также может упоминаться как межпотоковое взаимодействие (англ. inter-thread communication), межпоточное взаимодействие и межпрограммное взаимодействие (англ. inter-application communication). IPC наряду с концепцией адресного пространства является основой для разграничения адресного пространства [3].

Аутентификация по лицу будет производиться с помощью технологии OpenCV [4]. OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++/C#, также разрабатывается для Python, Java, Ruby, Matlab, Lua. Может свободно использоваться в академических и коммерческих целях – распространяется на условиях лицензии BSD [5]. Логика работы приложения представлена на рисунке 1.

Рис. 1. Логика работы приложения

Для экономии ресурсов и удобства работы вся программа разбита на несколько отдельных подпрограмм:

– служба Windows CryptoServiceGurezkiy.exe для мониторинга подключенных флэшек, аутентификации пользователя, мониторинга появления новых файлов на виртуальном диске и шифрования этих файлов, а также для выдачи пароля другим программам по IPC;

– программа cryper.exe для расшифровывания файлов и их копирования;

– CryptoSystem.exe для настройки приложения.

Так как эти программы использую схожий функционал, то целесообразно вынести все классы и методы в общую библиотеку классов, причём в несколько разных.

Библиотеки:

–библиотека Crypto3DES.dll для шифрования и расшифровки файлов;

–библиотека InterComunication.dll для общения службы CryptoServiceGurezkiy.exe и cryper.exe;

–библиотека LogicApp.dll для хранения общей логики программы и классов для хеширования данных, распознавания флэшек в системе, работы с реестром, сохранения изображения лица пользователя, распознавания лица пользователя, монтирования папки в качестве виртуального диска, отслеживания изменений в файловой системе виртуального диска. Общая схема зависимостей показана на рисунке 2.

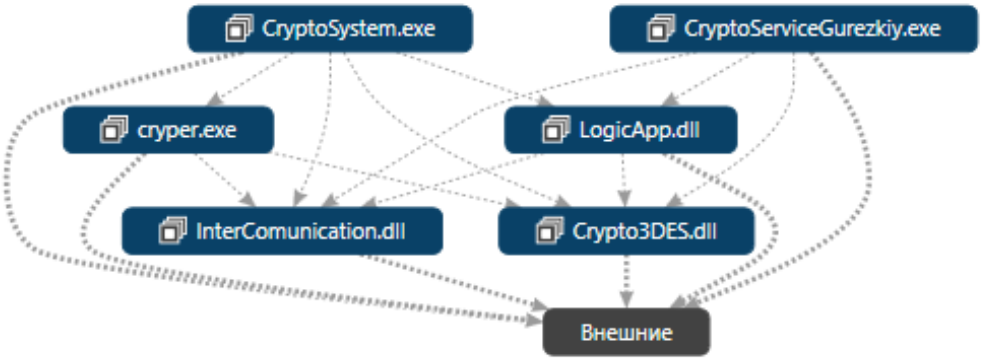

Рис. 2. Схема зависимостей проекта

Принципы работы конечного класса, который объединяет весь функционал программы, представлены следующим листингом.

Листинг кода 1 – «Реализация класса Recognize»

```
1. public class Recognize
2.     {
3.         public static bool run = false;
4.         LibLogic logic = new LibLogic();
5.         List<usbDevices> listusb = new List<usbDevices>();
6.         public void Run()
7.         {
8.             run = true;
9.             while (run)
10.            {
11.                listusb = logic.GetListUSB();
12.                if (listusb.Count !=  0)
13.                {
14.                    for(int i =  0; i < listusb.Count; i++)
15.                    {
16.                        if(logic.ValidKeyNoRec(logic.GetHash(listusb[i], ""))){
17.                            logic.StartFaceRecognize(listusb[i]);
18.                            if (LibLogic.KEY != "")
19.                            {
20.                                Serv serv = new Serv();
21.                                serv.Start(LibLogic.KEY);
22.                                Watcher w = new Watcher();
23.                                Thread a = new Thread(new ThreadStart(w.Start));
24.                                a.Start();
25.                                run = false;
26.                                break;
27.                            }
28.                        }
29.                    }
30.                }
31.                Thread.Sleep(3000);
32.            }
33.        }
34.        public void Stop()
35.        {
36.            run = false;
37.        }
38.    }
```

Инициализируем новый экземпляр класса LibLogic, в котором реализованы все те методы защиты, которые были описаны ранее. Затем, после вызова метода Run, приложение в бесконечном цикле выполняет опрос USB флэш карт. Если флэш карты есть в системе, то каждую из них проверяем на валидность ключа. Если ключ флэш карты совпадает с идеальным, то запускаем процесс аутентификации. Если ключ подходит для расшифровывания идеального значения в реестре, то создается виртуальный диск и в потоке запускается сервер, в котором и будет храниться ключ, затем запускается объект Watcher, который следит за состоянием виртуального диска. Если мы скопируем туда файл, то он сразу же зашифруется. С помощью контекстного меню можно и расшифровать этот файл на любой другой диск компьютера. Расширение файла ассоциировано с вспомогательной программой, суть которой заключается в следующем: программа запрашивает пароль у сервера, а затем копирует этот файл во временную директорию, выставляет права только для чтения и открывает его. После закрытия файла он автоматически удаляется.

Чтобы достигнуть асинхронности в приложении, необходимо использовать потоки. Поэтому во многих классах используется следующая конструкция:
Листинг кода 2– «Общая конструкция запуска метода в потоке»

```
1. <Класс> w = new <Класс>;
2. Thread a = new Thread(new ThreadStart(w.Start));
3. a.Start();
```

Таким образом, достигнув асинхронности методов, мы избежим зависание основного потока приложения и увеличим скорость его работы. После создания и отладки библиотек можно приступить к написанию программы.

Интерфейс программы необходим только в программе настройки CryptoSystem.exe. С помощью Windows Form создан интерфейс, показанный на рисунке 2.

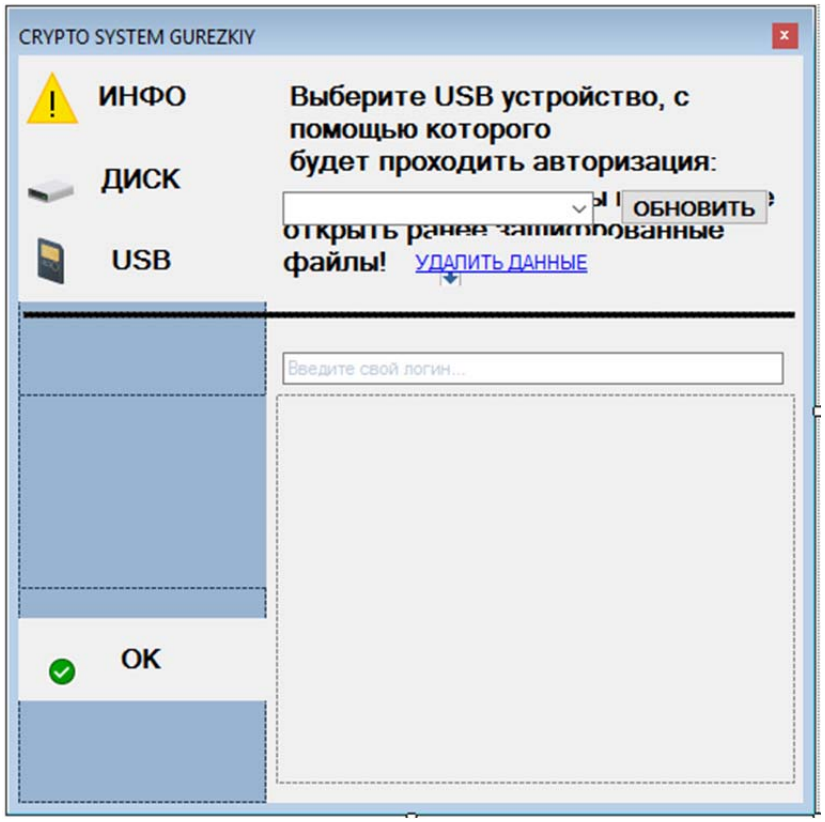

Рис. 3. Создание интерфейса CryproSystem.exe

На первый взгляд интерфейс кажется непонятным, однако не все эти элементы будут отображаться сразу.

После некоторых настроек интерфейс будет выглядеть, как показано на рисунках 3-6.

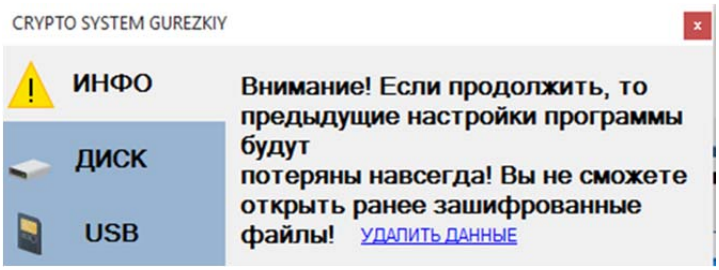

Рис. 4. Инфо

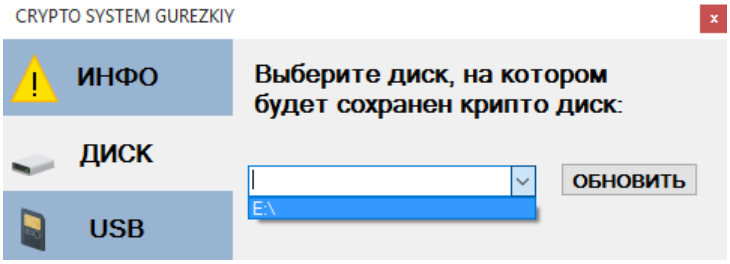

Рис. 5. Выбор диска для хранения папки с зашифрованными файлами

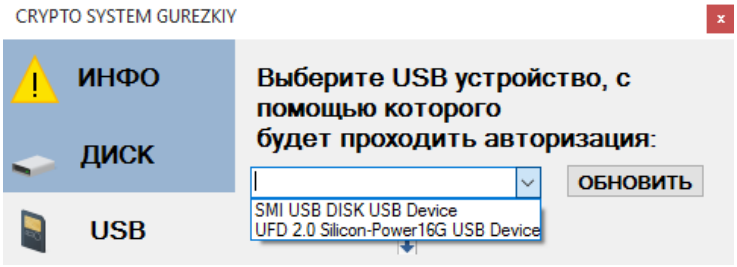

Рис. 6. Выбор флэш-карты

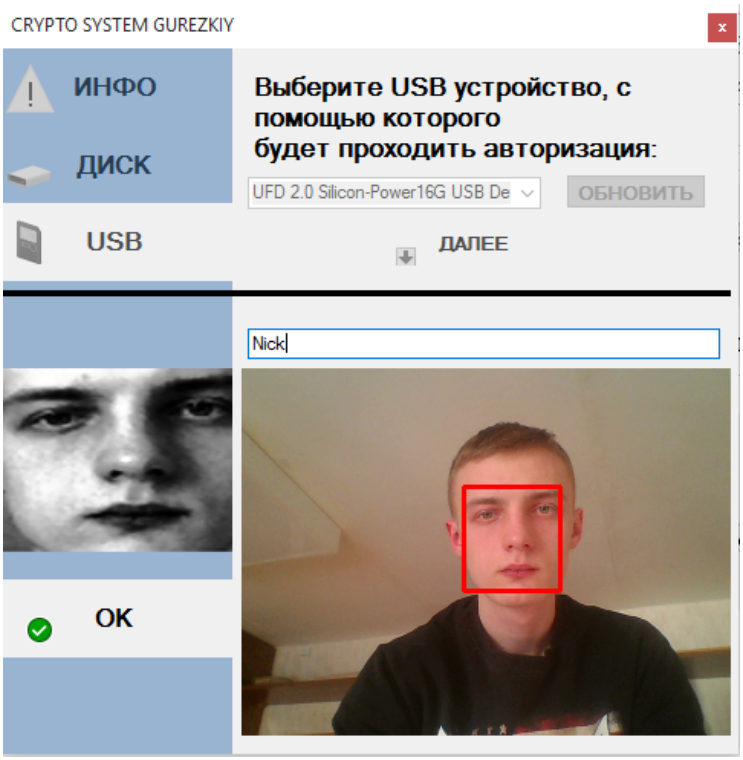

Рис. 6. Снимок лица пользователя

После настройки программы нужно нажать «ОК». Изменения программы вступят в силу после перезагрузки компьютера. Если пользователь уже однажды настраивал эту программу, то перед настройкой необходимо удалить данные, нажав на строку «Удалить данные на вкладке ИНФО».

Заключение. Авторами разработан программный продукт для надёжного хранения конфиденциальной, приватной, секретной информации на

персональном компьютере, позволяющий пользователю комфортно проходить процедуру аутентификации. Для чего достаточно подключить нужную флэш-карту к компьютеру и показать лицо видеокамере. На данный момент ведутся работы по увеличению функционала программы.

ЛИТЕРАТУРА

1. Triple DES. [Электронный ресурс] – Режим доступа https://ru.wikipedia.org/wiki/Triple_DES . Дата обращения 08.12.2015.
2. Службы Windows. [Электронный ресурс] – Режим доступа https://ru.wikipedia.org/wiki/Службы_Windows. Дата обращения 19.12.2015.
3. Межпроцессное взаимодействие. [Электронный ресурс] – Режим доступа http://dic.academic.ru/dic.nsf/ruwiki/658474 . Дата обращения 10.01.2016.
4. OpenCV. [Электронный ресурс] – Режим доступа http://opencv.org . Дата обращения 12.01.2016.
5. OpenCV. [Электронный ресурс] – Режим доступа https://ru.wikipedia.org/wiki/OpenCV. Дата обращения 13.01.2016.