Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**Пермский национальный исследовательский**

**политехнический университет**

Факультет прикладной математики и механики

Кафедра «Вычислительной математики, механики и биомеханики»

Направление 09.03.02 «Информационные системы и технологии»

Профиль программы бакалавриата «Цифровые технологии и интеллектуальные системы управления»

**КУРСОВАЯ РАБОТА**

по дисциплине **«ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ»**

на тему

**«Аутентификация по клавиатурному почерку»**

Выполнил студент группы ЦТУ-20-2б

Топорков Александр Игоревич

(Фамилия, Имя, Отчество полностью)

Проверил:

к. т. н., доц. каф. ВММБ ПНИПУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Труфанов Александр Николаевич)

(подпись) (ФИО)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023г.

**Пермь 2023**

**Содержание**

1. Введение 3
2. Основная часть 7
   1. Цель работы 7
   2. Задачи 8
   3. Техническая постановка и выбор средств реализации 8
   4. Реализация 9
3. Заключение 13
4. Список литературы 14
5. Приложение 15

Введение

Технологический прогресс задал скорость и вектор на сближение с глобальными изменениями во всех сферах жизнедеятельности общества, процесс цифровизации и внедрения IT-технологий происходит повсеместно. Это говорит о том, что теперь практически каждый человек должен иметь базовое понимание об IT-технологиях, методах и инструментах взаимодействия и обеспечения информационной безопасности (ИБ). Поскольку речь идёт не только о технических специалистах, важно детально прорабатывать структуру будущей информационной системы (далее ИС), чтобы по итогу с ней могли успешно и безопасно взаимодействовать пользователи различных сфер и возрастов.

В современном мире каждая информационная система имеет уязвимые места и может быть подвержена атакам. По статистике, наиболее распространённой угрозой для информационных систем являются несанкционированные попытки доступа (НСД) к данным. Обеспечение безопасности информационных систем означает внедрение комплексных алгоритмов защиты доступа к хранимой информации с целью устранения (снижения) риска её потери в результате НСД [1].

Эффективность обеспечения дальнейшего поддержания ИБ и определяется методами и степенью её проработки. Ключевыми процессами в системах защиты доступа являются идентификация и аутентификация [2]. На официальной странице всем известной антивирусной программы «Антивирус Касперского» предлагается следующая формулировка данных процессов:

Идентификация – это процедура, в результате выполнения которой для субъекта идентификации выявляется его идентификатор, однозначно определяющий этого субъекта в информационной системе.

Аутентификация – процедура проверки подлинности, например, проверка подлинности пользователя путём срабатывания введённого им пароля с паролем, сохранённым в базе данных (БД).

Авторизация – предоставление определённому лицу или группе лиц прав на выполнение определённых действий в рамках информационной системы.

Биометрические системы аутентификации (БМСА) стали распространяться повсеместно, благодаря надёжности и удобству использования и сопровождения. Запросы разработчиков и пользователей растут вместе с модернизацией программного обеспечения и требования к сохранению информационной безопасности в рамках системы обретают больший вес. Необходимо внедрение оптимального способа идентификации личности в сети – использование биометрических параметров, которые практически невозможно подделать или повторить. В дополнение выше сказанному, стоит отметить удобство использования биометрии для пользователя – параметры, по которым происходит идентификация, являются уникальной физической особенностью человека, которые не требуют запоминания, исключают ошибки и возможность передачи входных параметров другому [3].

Раньше подобные способы определения личности можно было встретить в фильмах о шпионах и организациях, имеющих в распоряжении данные особой важности и секретности и соответствующие методы их защиты, но сегодня, в силу безудержного информационного прогресса и увеличения расчётных мощностей, стоимость внедрения подобных систем постепенно снижается [4].

В качестве недостатков БМС выделяют так называемые ошибки первого и второго родов:

1. возможность обойти систему, обеспечивающую ИБ, с помощью вспомогательных фиктивных устройств и муляжей, например, маски идентифицированного в системе.
2. запрет получения доступа легитимному пользователю, биометрические параметры которого, в силу обстоятельств, на время (или навсегда) утратили схожесть с эталонными показателями, хранимыми в БД системы. Описанный вид ошибки может произойти если человек простыл, а идентификация происходит по голосу.

В научной литературе данные препятствия при аутентификации по биометрии называются FAR (False Acceptance Rate) и FRR (False Rejection Rate) и представляют собой «ложную тревогу» или «пропуск цели (промах)». FAR допускает вероятность совпадения биометрии разных пользователей (либо подделки), в то время как FRR «привязан» к эталонным данным настолько, что незначительные изменения в биометрии, которые способен достоверно распознать человек, для системы становятся решающими для отказа в доступе [5].

Аутентификацию по биометрии можно разделить на два класса: по статистическим и динамическим показателям человека. Первые определяют пользователя по его физическим характеристикам (отпечаток пальца, геометрия руки и лица), а вторые учитывают поведенческие особенности человека, анализируя его взаимодействие и выявляя привычные и подсознательные движения (паттерны), которые свойственны конкретному человеку. В качестве примера можно рассмотреть определение по голосу, ручному и клавиатурному почеркам.

Суть данной курсовой работы заключалась именно в реализации последнего из представленных выше динамических методов аутентификации – клавиатурного почерка.

Клавиатурный почерк пользователя (КПП) – это поведенческая характеристика человека, которая формируется на протяжении всего времени взаимодействия с устройством ввода: навыки улучшаются, а динамические показатели варьируются даже в зависимости от времени суток (по статистике, утром и вечером динамические показатели отличаются, что повышает вероятность ошибок аутентификации) и вида выполняемых задач. Описанный выше метод аутентификации рассчитан на использование клавиатуры и специализированного ПО, которое выполняет автоматизированный сбор и систематизацию данных (парсинг) пользователя для последующего анализа. Данный подход обеспечивает надёжность, требуемую для регулярной аутентификации, не прибегая к использованию дополнительных устройств.

Актуальность данного метода идентификации пользователя обоснован простотой внедрения и реализации, а также наименьшими затратами в финансовом плане, относительно иных методов анализа биометрии. Реализация узнавания динамических особенностей работы с клавиатурой является особенно комфортной для пользователя, поскольку от него не требуется ничего делать, кроме ввода текста (фразы/пароля) привычным и свойственным для него образом.

К сожалению, биометрия по КП имеет и не недостатки. К таковым относится вероятная нестабильность работы алгоритма распознавания, в связи с изменениями навыков набора текста из-за их улучшения или ухудшения, из-за травмы или смены устройства, например. Описанная проблема решается регулярной калибровкой системы и обновления базы данных с эталонными показателями пользователей.

В передовой системе аутентификации по клавиатурному почерку следует предусмотреть постоянные функциональные изменения, т.е. корректировки базы данных показателей пользователя после каждой успешной аутентификации. В наше время активно распространяется интерес к применению нейронный сетей, которые безусловно при должном обучении лучшим образом смогут решить данные проблемы, но это вопрос другого уровня. В рамках данной работы будет рассмотрен метод идентификации пользователя с помощью измерения интервалов между нажатием клавиш, длительностью удержания и проверки на соответствие введённых данных с ранее зарегистрированным образцом.

Основная часть. Постановка задач на проектирование.

Поступающая информация считывается по каждому нажатию и удержанию клавиши, потому что данные параметры напрямую связаны с опытом взаимодействия пользователя с клавиатурой и его персональным стилем. В более сложных реализациях может быть предусмотрен учёт использования «горячих клавиш» и областей ввода цифр - зависимости от предпочтений, каждый человек может использовать клавиши цифр, расположенных либо в ряд, либо в виде «калькулятора/клавиатуры PIN-кода».

Прямым элементом выборки рассмотрим время удержания клавиши.   
В режиме обучения (регистрации, в нашем случае) создаётся эталонное представление пользователя. Происходит сбор статистических данных о каждом нажатии клавиши, в результате чего получается трёхмерный массив, состоящий из N столбцов, где N – количество нажатых клавиш. Также фиксируется временной показатель удержания клавиши и интервала между нажатиями для формирования эталонной модели. После сбора данных подсчитывается математическое ожидание выборки (для каждой клавиши) и образец сохраняется в учётной записи.

Перед аутентификацией пользователю необходимо зарегистрироваться и идентифицировать себя, введя логин и пароль. При последующей авторизации запрашивается ввод, после чего происходит проверка наличия пары значений в БД и сравнение с эталонными показателями динамики ввода.

Технически, по алгоритму должна происходить инициализация массива (вектора) длинной L, где L – кол-во введённых символов, и в массив динамически должны заноситься параметры в виде времени удержания клавиш. Стоит предусмотреть возможность удаления пользователей для повторной регистрации эталонного образца.

Определяя легитимность пользователя, система проверяет итоговое численное значение (расстояние Хэмминга поступившего вектора Eп от эталонного Eэ) с пороговым, которое хранится в системе. Процесс принятия решения можно описать следующим образом:

2.1. Цель работы: написать программу с реализацией аутентификации пользователя по клавиатурному почерку и минимальный графический интерфейс, необходимый для проверки корректности работы алгоритма.

Для достижения поставленной цели необходимо сформировать и реализовать следующие задачи:

2.2. Задачи:

1. Проанализировать литературу и решения данной тематики;
2. Спроектировать собственную программную систему;
3. Выбрать средства и методы реализации;
4. Реализовать программную систему;
5. Протестировать программную систему на разных пользователях.

2.3. Техническая постановка и выбор средств реализации.

В качестве средств реализации были выбраны среда разработки Visual Studio и язык программирования (ЯП) C#, благодаря его распространению и удобству при создании оконного приложения в режиме конструктора и работе с файлами. Также имеется опыт разработки на данном ЯП.

Функциональность приложения, реализуемая в рамках данной курсовой работы, должна включать следующие возможности:

1. Регистрация эталонных показателей пользователя;
2. Формирование и вывод статистической информации для анализа работы алгоритма;
3. Аутентификация пользователя с помощью сравнения введённых параметров и эталонных показателей с учётом анализа динамического взаимодействия с клавиатурой.

2.4. Реализация.

Код приложения:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Diagnostics;

namespace Password

{

public partial class Form1 : Form

{

public static List<long> vectorList = new List<long>();

public static List<long> vectorListSample = new List<long>();

long Length\_vector = 0; // Длина пользовательского вектора

long Length\_vector\_sample = 0; //Длина эталонного вектора

long Manhattan\_distance = 0;

double Difference\_with\_the\_sample = 0;

public Form1()

{

InitializeComponent();

KeyPreview = true;

var watch = new Stopwatch();

KeyDown += (s, e) => watch.Start();

KeyUp += (s, e) =>

{

if (radioButton1.Checked) // фикс. ввод пользователя

{

watch.Stop();

if (password\_textBox.Focused)

{

var vector = watch.ElapsedMilliseconds; // измер. продолжит.

vectorListSample.Add(vector);

Vector\_of\_durations.Text += " " + vector;

}

watch.Reset();

}

else //проверка данных пользователя

{

watch.Stop();

if (password\_textBox.Focused)

{

var vector = watch.ElapsedMilliseconds; // фикс. продолжит.

vectorList.Add(vector);

Vector\_of\_durations.Text += " " + vector;

}

watch.Reset();

}

};

}

private void button1\_Click(object sender, EventArgs e) // кнопка подтвержд.

{

Vector\_of\_durations.Text = "";

Length\_vector\_sample = 0;

if (radioButton1.Checked) //ввод пользователем данных

{

var vect = textBox1.Text; //Login

var pass = password\_textBox.Text;

password\_textBox.Text = "";

for (int i = 0; i < vectorListSample.Count; i++)

{

Length\_vector\_sample += vectorListSample[i]; //длина вектора польз.

vect =vect+ '|' + vectorListSample[i].ToString();

}

LengthElapsedSample.Text = Length\_vector\_sample.ToString();

vect += "|password=" + pass;

//"UsersHandwriting" конфигурационный файл для сохранения логина/пароля

((System.Collections.Specialized.StringCollection)Password.Properties.Settings.Default["UsersHandwriting"]).Add(vect);

Password.Properties.Settings.Default.Save();

textBox1.Text = ""; //очистка конфиг. файла

vectorListSample.Clear();

}

else {

Length\_vector = 0;

for (int i = 0; i < vectorList.Count; i++)

{

Length\_vector += vectorList[i]; //длина вектора

}

LengthElapsed.Text = Length\_vector.ToString();

vectorListSample.Clear();

int c = ((System.Collections.Specialized.StringCollection)Password.Properties.Settings.Default["UsersHandwriting"]).Count;

String[,] myUsersDiff = new String[c,3];

int index = 0; //индекс сходимости

foreach (var user in ((System.Collections.Specialized.StringCollection)Password.Properties.Settings.Default["UsersHandwriting"]))

{

string[] items = user.Split('|');

myUsersDiff[index,0] = items[0]; //имя пользователя

int itemsIndex = 0;

vectorListSample.Clear();

foreach (string item in items)

{

if (itemsIndex != 0) {

if (!item.Contains("password=")) {

vectorListSample.Add(Convert.ToInt64(item));

}else {

var pass = item.Replace("password=", "");

if (pass == password\_textBox.Text) { myUsersDiff[index, 2] = "1"; } //проверка пароля

else

{

myUsersDiff[index, 2] = "0";

};

}

}

itemsIndex++;

}

Length\_vector\_sample = 0;

for (int i = 0; i < vectorListSample.Count; i++)

{

Length\_vector\_sample += vectorListSample[i]; //длина вектора пользователя.

}

if ((Length\_vector > 0) && (Length\_vector\_sample > 0)) //сравнение длин векторов.

{

if(vectorListSample.Count== vectorList.Count)

{

for (int i = 0; i < vectorListSample.Count; i++)

{

Manhattan\_distance += Math.Abs(vectorListSample[i] - vectorList[i]); // разница продолжительности векторов

}

}else { myUsersDiff[index, 2] = "0"; }

ManhattanDistance.Text = Manhattan\_distance.ToString();

Difference\_with\_the\_sample = ((float)Manhattan\_distance / (float)(Length\_vector + Length\_vector\_sample)) \* 100;

myUsersDiff[index,1] = Difference\_with\_the\_sample.ToString();

}

index++;

}

double minDiff = 999;

String ResultUser = "";

for(int ind=0;ind< myUsersDiff.GetLength(0); ind++)

{

if ((minDiff > Convert.ToDouble(myUsersDiff[ind,1]))&&(textBox1.Text==myUsersDiff[ind, 0])&& myUsersDiff[ind,2]=="1")

{ minDiff = Convert.ToDouble(myUsersDiff[ind, 1]);

ResultUser = myUsersDiff[ind, 0];

}

}

Diff.Text = minDiff.ToString("F2");

if (ResultUser != "" && minDiff <= 15) { label6.Text = ResultUser; }

else { label6.Text = "Не определён"; }

LengthElapsedSample.Text = Length\_vector\_sample.ToString();

password\_textBox.Clear();

vectorListSample.Clear();

vectorList.Clear();

Length\_vector = 0;

Length\_vector\_sample = 0;

Manhattan\_distance = 0;

}

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

Vector\_of\_durations.Text = "";

if (radioButton1.Checked)

{

vectorList.Clear();

vectorListSample.Clear();

}

}

private void Form1\_Load(object sender, EventArgs e)

{

radioButton1.Checked = true;

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

if (password\_textBox.Text != "" && textBox1.Text != "")

{

var vect = textBox1.Text;

if (radioButton2.Checked)

{

var pass = password\_textBox.Text;

password\_textBox.Text = "";

for (int i = 0; i < vectorListSample.Count; i++)

{

Length\_vector\_sample += vectorListSample[i];

vect+= '|' + vectorListSample[i];

}

vect += "password=" + pass;

LengthElapsedSample.Text = Length\_vector\_sample.ToString();

}

((System.Collections.Specialized.StringCollection)Password.Properties.Settings.Default["UsersHandwriting"]).Add(vect);

Password.Properties.Settings.Default.Save();

textBox1.Text = "";

}

}

private void button2\_Click(object sender, EventArgs e)

{

((System.Collections.Specialized.StringCollection)Password.Properties.Settings.Default["UsersHandwriting"]).Clear();

}

private void PasswordLabel\_Click(object sender, EventArgs e)

{

}

}

}

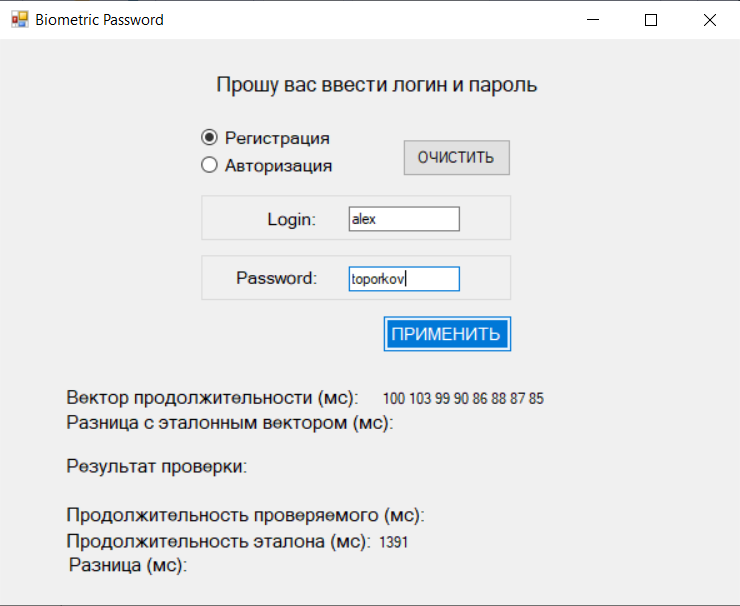
Заключение

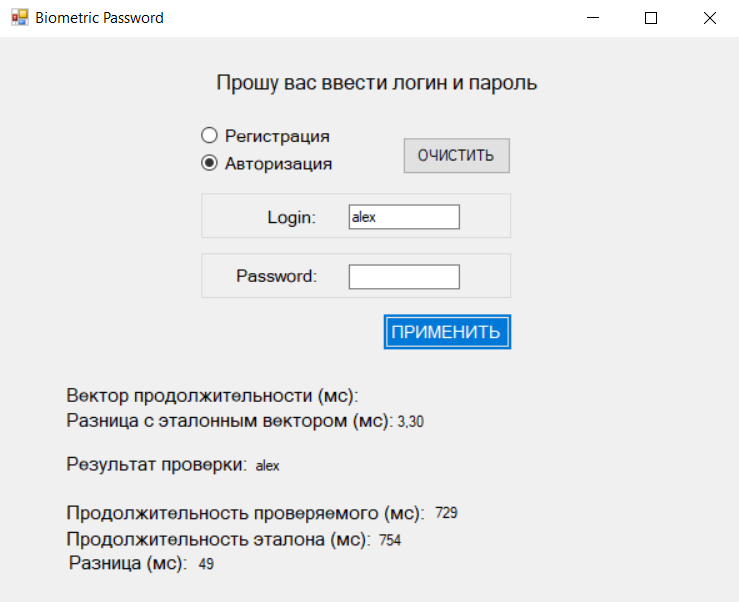
Цель и задачи, поставленные в рамках данной курсовой работы, были достигнуты. При тестировании функциональности каждая кнопка работает в соответствии с описанным алгоритмом, вывод статистической информации соответствует принятому системой решению. Пороговое значение соответствия векторов было определено в рамках 15% их различия. Данный показатель был установлен опытным путём как наиболее оптимальный при текущем виде реализации алгоритма. При пороговом значении ниже возникает вероятность ошибки FRR (отказ в доступе зарегистрированному пользователю). Код приложения представлен выше в разделе «Реализация». Скриншоты интерфейса с примером работы размещены в разделе «Приложение».

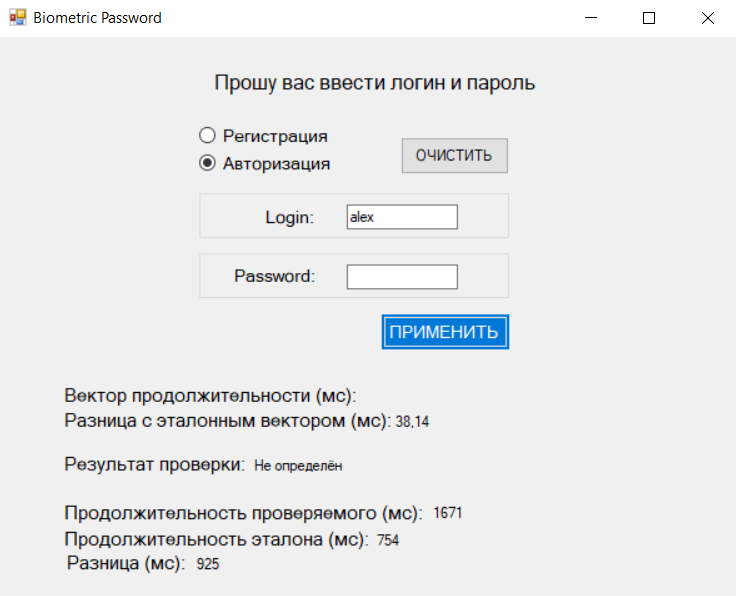
**Список литературы**

1. Аникин И.В., Глова В.И. Методы и средства защиты компьютерной информации: учеб. пособие. Казань: Изд-во Казанского государственного технического университета, 2008. 260 с.
2. Ходашинский И.А., Савчук М.В., Горбунов И.В., Мещеряков Р.В. Технология усиленной аутентификации пользователей информационных процессов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2011г. – №2-3 (24). – С. 236 – 248.
3. Сабанов А.Г. Об уровнях строгости аутентификации // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2012г. - № 2-1 (26). – С. 134 – 139.
4. Кусков Н.А. Исследование способов несанкционированного доступа к информации // Научный вестник Московского государственного технического университета гражданской авиации. – 2013г. - № 6 (192). – С. 127 – 129.
5. Островский А.А., Жариков Д.Н., Лукьянов В.С., Попов Д.С. Динамические методы биометрической аутентификации // Известия Волгоградского государственного технического университета. – 2010г. - № 8 . – Том 6. – С. 72-76.

**Приложение**

****Рис. 1. Интерфейс приложения

****Рис. 2. Вывод при корректной авторизации

****Рис. 3. Вывод при некорректном пароле или при   
нестандартом динамическом взаимодействии с клавиатурой