Biometric Authentication Tool for User Identification Based on Keystroke Dynamics

GRADUATE PROJECT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Ankita Nellimarla
June 2006

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. David R Thomas
Committee Member

Dr. Long-Zhuang Li
Committee Member
Biometric Authentication Tool for User Identification Based on Keystroke Dynamics

GRADUATE PROJECT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Ankita Nellimarla
June 2006

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. David R Thomas
Committee Member

Dr. Long-Zhuang Li
Committee Member
ABSTRACT

Biometric access methods for computer systems are gaining popularity because of governmental and corporate businesses' increased focus to secure sensitive data on computer systems and network. Biometrics is the science of measuring a unique physical characteristic about an individual as an identification mechanism. Keystroke Biometrics is a relatively new method of biometric identification and provides a comparatively inexpensive and unobtrusive method of hardening the normal login and password process. It does not require additional hardware as it uses the existing keyboard to measure keystroke dynamics. The term keystroke dynamics is used to describe an individual's typing pattern including, latencies, key depress durations and keystroke pressure. This pattern is fairly unique to each individual.

This Graduate Project aims at investigating the validity of using typing dynamics to strengthen security in a computer system. A Keystroke Dynamics Analysis tool is developed that uses statistical analysis of a user's typing patterns to perform identity verification. Results of the program are used to confirm that Keystroke Biometric Authentication is a valid method for identity verification.
TABLE OF CONTENTS

Abstract......................................................................................................................... ii
Table of Contents.......................................................................................................... iii
List of Figures.................................................................................................................. v
List of Tables.................................................................................................................. vi

1. Introduction and Background................................................................................... 1
   1.1 Introduction.............................................................................................................. 1
   1.2 Identification Methods................................................................................................. 1
   1.3 Biometric Authentication............................................................................................. 2
   1.4 How Keystroke Biometrics Works................................................................................. 3
   1.5 Importance of this Project........................................................................................... 3
   1.6 Similar Works ............................................................................................................. 4
   1.7 Summary of Report...................................................................................................... 5

2. Biometric Authentication Tool for User Identification............................................. 6
   2.1 External Aspects of Project.......................................................................................... 6
   2.2 User Interfaces............................................................................................................ 6
      2.2.1 Start Up Interface.................................................................................................. 7
      2.2.2 New User Interface............................................................................................... 7
      2.2.3 New User Registration ......................................................................................... 9
      2.2.4 Existing User Login Screen.................................................................................. 9
      2.2.5 Successful Login Message Prompt....................................................................... 11

3. System Design............................................................................................................. 12
   3.1 Main Components and System Requirements......................................................... 12
LIST OF FIGURES

Figure 2.1 Entry/Start up Screen for Access into System ................................................. 7
Figure 2.2 New User Interface, Username Selection .......................................................... 8
Figure 2.3 New User Interface, Password Selection .......................................................... 8
Figure 2.4 New User Registration Screen ........................................................................ 9
Figure 2.5 Existing User Learning Login Screen ............................................................... 10
Figure 2.6 Existing User Login Screen ............................................................................ 10
Figure 2.7 Successful User Login ..................................................................................... 11
Figure 3.1 System Architecture ....................................................................................... 13
Figure 3.2 Structure Chart for the Design Phase ............................................................... 15
Figure 3.3 Context Level Diagram of the System ............................................................... 16
Figure 3.4 Level One DFD ................................................................................................. 17
LIST OF TABLES

Table 4.1 User Responses.................................................................22

Table 4.2 False Acceptance Rate for Users Typing the Sentence...................27
1. INTRODUCTION AND BACKGROUND

1.1 Introduction

The increasing need for securing access to computer systems and networks from intruders is growing rapidly as the type of data and capabilities of these systems is becoming significantly sensitive. To provide access to these systems while preventing illegitimate access is the key requirement of modern day computing. Since biometric systems do not identify a person by what he or she knows (a code) or possesses (a card), but by a unique characteristic that is difficult for a different individual to reproduce, the possibility of forgery is greatly reduced [Kung 2004]

1.2 Identification methods

The access system should be able to correctly authenticate the identity of an individual and then allow them access to the defined resources. Since almost all artificial features used in conventional identification techniques can likely be forged or stolen (example, ID cards or passwords) more effective identification and authentication methods are now in greater demand. The main methods of user identification are something you know example, password; something you have example, swipe card and something you are example, biometrics [Gollman 1999]. The use of login names and passwords is the most commonly used mechanism for authentication purposes. Despite its popularity there are many inherent flaws to this approach. Passwords that are simple and memorable are easy to guess or searched by an attacker and a long random password is difficult to remember. Passwords are at a disadvantage of being open to compromise without the knowledge of their disclosure [Leggett 1990].
Other alternative methods of verification include tokens, swipe cards and user biometrics. A token is a type of handheld authenticator that performs authentication. This can be a secure storage device containing passwords or pins such as a bankcard, or a smart card. A biometric is a feature measured from the human body that is distinguishing enough to be used for user authentication. A biometric purports to inextricably link the authenticator to its owner, something passwords and tokens cannot do, since they can be lent or stolen [Gorman 2003].

1.3 Biometric Authentication

Biometrics is the science of measuring a unique physical characteristic about an individual as an identification mechanism. A number of widely used biometric technologies and techniques exist. Some of the common methods include fingerprints, voice characteristics, eyes, facial features, and keystroke dynamics.

Fingerprint biometrics is widely used and accepted technology. Here the biometrics is defined by the unique patterns of one’s fingers. Since the form of identification involves fingerprints, it has issues of both fear and acceptance. Some individuals involved with fingerprint authentication for network access has expressed the feeling that using their fingerprint for authentication makes them feel like a criminal [Reid 2003].

Face biometrics is used in day-to-day life. We recognize each other by faces as it the first defining character for identity. We carry photo IDs to represent our proofs of identity. The problem here is we have all at one time or another thought we recognized someone's face, only to have it not be the person we thought. In special cases of similar faces for example, twins, it is difficult to identify accurately. Not surprisingly, the use of face biometrics for identification is often questioned for deployment in a network security
environment, as it does not provide the same level of accuracy as other biometrics. Similar problems are associated with voice biometrics where sound recognition can be spoofed or misinterpreted. Newer techniques like iris recognition and keystroke dynamics are now becoming more popular as they measure stronger and reliable biometric traits of human beings. Researches have shown that these techniques have virtually no FAR (False Acceptance Rates) and extremely low 0.2% FRR (False Rejection Rates) in three attempts.

1.4 How Keystroke Biometrics Works

Keystroke recognition is completely a software-based solution. No additional equipments are needed and all that is needed is the existing computer and keyboard that the individual is currently using. It involves two processes:

1. **Enrollment Process**: This process will enroll the user and generate a template for him. To start with, the individual must type in a specific word or a phrase or a set of alphanumeric characters. This is usually a username and password.

2. **Verification Process**: This process would verify an enrolled individual to confirm the identification of the person. It would match the current template with an existing one for authentication purposes.

It is very important that the same text be used in both these processes otherwise the behavioral characteristics of typing might be significantly different and produce a mismatch between the two templates and access would be denied. If the user makes corrections while typing the system would prompt the user to start all over [Das 1999].

1.5 Importance of this Project

This Graduate Project aims at using the inexpensive and unobtrusive method of keystroke biometrics to build a user authentication system. Keystroke biometrics is a
relatively new method of biometric identification. It uses the existing keyboard to measure keystroke dynamics of an individual. The motivation to use this technology for this Graduate Project work comes from the fact that this is would be a complete software based solution for building an efficient user identification tool. The tool designed will have the following characteristics:

1. No additional or specialized hardware is needed to implement this method, as it would use the existing computer keyboard.

2. No additional costs are involved for training individuals using the system.

3. Easy to use with minimal set up time and can be readily installed on a system or network.

4. Data will be stored in the form of templates.

1.6 Similar Works

Umphress and Williams undertook one of the earlier works in this area in 1985. It used the delays between keystrokes, also known as di-graphs for the captured keystroke biometric. They had two sets of inputs required in their process with 1400 and 300 characters of prose. [D'Souza 2002] Though the study proved that keystroke biometrics is a valid method for identification, it was restricted by the huge amount of input text. Another work done by Joyce and Gupta was designing a user identification system using keystroke patterns. Their work compares keystroke latencies of a fixed string which is a password already stored in the system. This Graduate Project differs from their work in way as it does not depend on pre-selected strings for user authentication. One of the recent works in this area was conducted by Robinson et al. in which the authors used keystroke hold times to characterize the typing styles [Obaidat 1997].
1.7 Summary of Report

This report is divided into the following sections as below:

- The second section titled with the title of the project deals with the external aspects of the application in terms of user interfaces and project results.
- The third section titled “System Design” deals with the detailed description of the internal aspects of the project and the major components and their method of design and implementation.
- The fourth section “Evaluation and Results” deals with interface testing and also identifies portions where the report differs from the proposal.
- The fifth section “Future Work” deals with scope of future work related to this project.
- The final section “Conclusion” summarizes the outcomes of the project.
2. BIOMETRIC AUTHENTICATION TOOL FOR USER IDENTIFICATION BASED ON KEYSTROKE DYNAMICS

2.1 EXTERNAL ASPECTS OF PROJECT

The application is a single user system and only one user can have the access to the application at a time. The application can be installed on any number of systems with the option of running multiple versions of the application simultaneously on these machines. The application has the ability to capture user keyboard entry and identify a key entered; a keycode; key press time; key release time; duration; latency; number of single clicks; number of double clicks; number of right clicks; and the number of times the backspace key was used.

The application has two stages of new user registration or an existing user verification and authentication. During the initial registration process, the system would ask the user to enter three types of strings, which would be a username, a password and a phrase, which can be alphanumeric. The input phrase field is a constant phrase, added to the system to harden the security policy where the system can compare the typing dynamics of multiple users for the same phrase. All this information is stored in a configuration file. To enter the system, the user is now required to use the same data to fill in the login form. Verification is always done through a login screen which verifies username and password. The system allows flexibility of creating new users with all pertinent information. The only additional step for a new user is to create a biometric profile before he or she can login into the system using the same username and password as used during the registration process.
2.2 USER INTERFACES

2.2.1 Startup interface

The startup screen provides an option of selecting an existing user or creating a new user. This interface is followed by a login screen or a registration screen based on what the option selected.

Figure 2.1 Entry/Start up Screen for Access into System

2.2.2 New User interface

This screen allows for the creation of a new user whose information is stored in a configuration file as a biometric profile which the system will access upon login to match the keystroke patterns to validate the user at the time of login. The user is asked to pick a username and password and there is a standard phrase. The username cannot be more
than twelve characters and less than five. User can use alphabets, special characters and numbers to create a username. Similarly, he or she can select a password for secured login. All fields are mandatory and password verification is a must. Figure 2.2 displays this screen.

![Figure 2.2 New User Interface, Username Selection.](image)

![Figure 2.3 New User Interface, Password Selection.](image)
2.2.3 New User Registration

This screen provides the user with three options to fill in the same username, password the user had selected during the previous step. The standard phrase is common to all users to test the difference in typing characteristics for same phrase for multiple users.

![Biometric Authentication Tool - Login Form](image)

**Figure 2.4 New User Registration Screen**

2.2.4 Existing User Login Screen

This screen asks the user to enter the same username, password or pass phrase the user had entered during the initial enrollment phase. The user is prompted to enter the login information more than one times. In this case it is 10 times as more accurate characteristics of the user can be captured over a period of time.
Figure 2.5 Existing User Learning Login Screen

2.6 Existing User Login Screen
2.2.5 Successful Login Message Prompt

If the login characteristics match the saved biometric profile characteristics, the user is prompted with a successful login message. In case of profile mismatch, the user is denied access to the system and he has to try again or contact the system administrator.

![Successful User Login](image)

*Figure 2.7 Successful User Login*
3. SYSTEM DESIGN

3.1 MAIN COMPONENTS AND SYSTEM REQUIREMENTS

The main component of this project is Visual Basic 6.0. The entire application has been built using the VB programming language. Visual Basic is the most productive tool for rapidly creating a wide range of Windows and Web applications. The language is designed to be human readable and accessible to everyone from novice programmers to advanced system architects [Peacock 2000]. Visual Basic allows for easy form creation and the built in Dynamic Linked Libraries for efficient deploying windows applications. The form elements can be easily linked to the respective events to be triggered upon user selection. The object oriented framework allows for reusability of code and components [Peacock 2000].

3.1.1 SYSTEM ARCHITECTURE

The system stores each individual’s characteristics in a unique template. Starting from the enrollment template for a new user to the login process of an existing user, the system prompts the user for all the required fields to generate this unique template. Best results are obtained if enrollment occurs over a period of time rather than at one sitting as over a period of time, individual characteristics are identified more accurately [Monrose 1999]. The system will now take these raw data files and develop a feature extractor which basically generates the keystroke latencies which are measured in a number of ways as follows [Monrose 1999]:

- The length of the time each key is held down.
- The length of the time between keystrokes.
- The typing speed.
- The tendency to switch between numeric keypad and keyboard numbers
• The keystroke sequences involved in capitalization.

    All this information is stored as a feature extractor which produces feature files which can be parsed by the classifier where the user’s keystroke sample is compared with the stored unique template to produce an output of success or failure and access is granted if the submitted sample matches the template according to previously established probabilities. Figure 3.1 depicts this architecture.

![System Architecture Diagram]

**Figure 3.1 System Architecture**

3.2 DESIGN PHASE

3.2.1 Structure Chart

    The first method employed in the design phase is a structure chart. It basically decomposes a problem and is not in any sequential order. The program
will have three phases, learning phase, login phase and verification phase. In the learning phase, the tool will read three input values from the users.

- Username
- Password
- A standard phrase

The learning phase should only capture the user inputs 15 times and generate a signature profile. After a signature profile is created by the tool in the learning phase, the user should now be able to login into the system using the username, password and standard phrase information submitted during the learning phase. This next phase is the login phase. The login phase will prompt the user to login and this will be similar to the learning phase, where a biometric profile is generated. The verification phase will compare the biometric profile with the signature profile to find a match for a valid login. A log file will maintain a log of all user login attempts. Figure 3.2 depicts the chart.
3.2.2 Data Flow Diagrams (DFD)

DFDs show the flow of data from external entities into the system and how data moves from one process to another [Pressman 2005]. The design and functionality of the system is depicted in the DFDs below.

3.2.2.1 Level 0 Data Flow Diagram

Level 0 DFD or context level diagram depicts the overall data flow in the system as a whole. Figure 3.3 shows the flow of data in the system at a higher level.
3.2.2.2 Level 1 Data Flow Diagram

Level One Data Flow Diagram (DFD) explains the flow of data in the system in detail, compared to the context level DFD. Figure 3.4 shows the flow of data in the system at a higher level.
3.3 Implementation

This section narrates the project implementation details. Please see Appendix A for code snippets. The entire project is divided into seven main modules. Each module is sub-divided into several units. The following is the list of modules:

- User Selection Module
- Registration Module
- Login Module
3.3.1 User Selection Module

This module uses two global variables to save the state of the user which can be an existing user or a new user. These variables are captured during the button click event of the interface. There are two buttons on the form for existing user and new user with each having a click event linked in the code. Upon the button click, the values of the global variables get set which are passed on to the main parent application which decides which module should be displayed next based on this value.

3.3.2 Registration module

This module is only displayed if the user is a new user and requires learning login process for the registration. The learning login module is an iterative for loop with an index variable ‘index’ which increments itself upon the completion of form data per learning. The code for this module is in the form frmLearningLogin.frm. There are private variables declared for this form which keeps track of number of characters used which cannot be less than 8 or greater than 12 characters. If these characteristics are met then the user typing characteristics are measured by calling the function which measures the key press time or key hold time and inter-key time as latencies to calculate the weighted latencies and standard deviation to generate a feature profile.

3.3.3 Login module

This module appearance is the same but functionality is different. The module is for prompting an existing user to prove his authenticity for successful login. The user cannot use backspace or delete buttons. There are methods linked to onkeypress, onkeydown, onkeyup etc. which are defined to capture the keystroke latencies as a factor of milliseconds. The feature profile extraction method is the same but at the end of each
login form completion, the profile generated is matched with the feature profile. If a match is found, the user is prompted with a successful login message.

3.3.4 Output Message module:

This method is a simple output message box which displays the output or result as a success or a failure for the user. If it is a failure, it automatically exits the application.

3.4 ENHANCED LOGIN PROBLEM AND PROPOSED SOLUTION

The username-password paradigm is common throughout virtually all Web services and system login programs. Currently, a user’s account is entirely compromised if an adversary somehow discovers the user’s username and password. Problems such as continuous user authentication have available hours or days of training data per user. By contrast, a keystroke-enhanced login system must be capable of learning a user’s typing pattern for a short username and password pair within 10-15 samples of roughly 2040 characters each, over a course of 3-5 minutes. The small training set means that more patterns must be mined from the samples to generate adequate information to produce a template of the user’s typing pattern [Lau 2004].

Statistical analysis is the most viable solution to keystroke enhanced login problem [Manuel as cited in Lau 2004]. The statistical methods statistical methods used can vary from comparing the diagraphs of the entered text to the diagraph means in the template, to classification methods, such as Bayes’ Classification algorithms that use probability density functions [Bleha 1990]. The proposed system model will compute a vector of means and standard derivations for the latencies noted for each pair of keystrokes. This vector will represent the user profile. The following section will explain the implementation of this statistical analysis model.
3.4.1 Development of an Authentication Method

The algorithm will develop a signature profile for each of the three input strings entered by the user. The algorithm will use two variables, a test signature “T” which will be acquired at the login time and a mean reference signature “M” where \( M = \{ M_{\text{username}}, M_{\text{password}}, M_{\text{phrase}} \} \). Verification is performed by comparing the test signature T with M and determining the magnitude of the difference between the two profiles [Joyce 1990]. Given \( M = \{ m_1, m_2, \ldots, m_n \} \) and \( T = \{ t_1, t_2, \ldots, t_n \} \) where \( n \) is the total number of latencies in the signature, the algorithm will compute the magnitude of the difference and positive identification is declared when the difference is within the threshold variability of the reference signature. When more than 80% of all the possible latencies passed this test then input for that string would be considered valid.

The usual way of calculating standard deviations is shown below in equation (3.3.1) This method requires that all of the login digraph times need to be stored, the population standard deviation can than be calculated from all of these digraphs [Boyle 1989].

\[
S = \sqrt{\frac{\sum(x-x')^2}{n}} \quad (3.3.1)
\]

Where,

\( x \) is the value for each latency,

\( x' \) is the mean,

\( n \) is the number of logins

\( S \) is the standard deviation.

This process means that every previous login needs to be stored just so that the standard deviation can be calculated. Manipulating the above standard deviation formula, a
method can be used that does not require all the previous latency values for the standard deviation to be calculated.

Equation (3.3.2) written below will only use the value for the sum of the values for each latency and the squared sum for each latency. This enables the standard deviation to be counted more quickly and reduces the amount of storage space required for the user’s profile.

\[ S = \sqrt{\left(\frac{\sum X}{n}\right) - \left(\frac{\sum X}{n}\right)^2} \] (3.3.2)

Where,

S is the standard deviation,

n is the number of logins,

\( X = x - x' \)

x is the mean.

The main problem with using the standard deviation directly is that a digraph may be approved even though it has a high variability and consequently allows many more digraphs to be approved because the allowed error is high. [Joyce 1990]. So that even if more than 80% of latencies fit within one standard deviation of the reference profile, the profile generated may not be significant.

A way to remedy this problem is to use weighted latencies by assigning a weight to each of the deviations with regards to how the standard deviation compares with the mean as a percentage. In such a case, if the standard deviation is high in comparison to the mean, then the approval for that latency will have a low weighting [D’Souza 2002]. Equation (3.3.3) takes care of this problem by calculating weighted values for the
latencies as shown below. If the sum of the computed weights using the following equation is at least 70%, the generated profile is approved.

\[
W = \left[ \frac{1}{\sum (x'/S)} \right] \times (x'/S) \times 100 \%
\]  

(3.3.3)

Where,

- \( S \) is the standard deviation,

- \( x' \) is the mean.

For a biometric profile to be approved, more than 80% of the latencies must fit within one standard deviation of the reference profile and for each valid latency that fits within the standard deviation of the sum of the weights must add up to at least 70%. [Monrose 1997]. Verification of this method will be detailed in the project evaluation and testing to show how keystroke latencies are approved using the above equations.
4. EVALUATION AND RESULTS

4.1 USABILITY TESTING

Usability testing is an integral part of any software engineering process. Usability testing encompasses a range of methods for identifying how users actually interact with a prototype or a completely developed product. A number of factors like do users complete a task successfully, how fast do they do each task, or what paths do they take play an important role in considering the success of the usability testing. Its importance lies in the fact that it deals with user-application communication which defines the success of any application. A user interface is well designed when the program behaves exactly how the user perceives it to. The following factors play an important role in defining the ideal user interface [Dix 2003]:

1. Text
The purpose of the page or dialog should be clear with concise instructions on what to do next. The user should not be forced to infer meaning.

2. Controls
The right controls should be set up so that interacting with a page is easier and the user can keep moving quickly.

3. Design for Mistakes
Users should be able to recover from their mistakes. Application should be designed in a way which makes it easy for users to recover from their mistakes.
4. Minimize user’s memory load

The users should not have to remember their past actions every time they use the system. The interface should be friendly enough for the users so that each time they log in, they would know how to work around the functionality of the application.

4.2 QUESTIONNAIRE SURVEY

Questionnaire survey is the method adopted in this project to conduct extensive usability testing of the application. The test users include professional people from the IT industry, working in reputed companies, who are in a very good position to judge the look and feel of such an application, as developed in the project. A total of 10 people were surveyed for the said purpose. An important aspect is that none of the evaluators were a part of the development process. Each of the questions had five options to it as follows: strongly agree (5 points), agree (3 points), neutral (2 points), disagree (-2 points) and strongly disagree (-5 points). This would gauge correctly as to how the test users felt about the application. The last three questions are open ended questions giving the user the option of expressing their positive and negative thoughts on the application. The following were the questions asked in the survey:

1. It is easy for me to enter my username and password
2. It is easy for me to enter the phrase
3. It is easy to understand what the application is aiming to do by asking for repeated logins
4. It is easy to gather the saved information of an existing user
5. It is easy to read the error messages and understand the problem
6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
7. What did you like least about the application? Detail at least 2 features.
8. Would you want to use the same application for the purpose it has been designed for? Why or Why not?

4.3 RESPONSES

There were ten evaluators to the questionnaire. Most of the respondents were professionals from the IT industry working in reputed companies having 2-3 years of experience. Most of the responses were for “strongly agree” or “agree” options and only four were “neutral”. None of the responses were negative. Their answers suggested that the application was indeed usable and functionally competent.

Following the responses, the weighted average of each question was calculated in the following manner: Respective points for a response have been multiplied by the number of responses for that question and then divided by the number of evaluators. The weighted average can give a true insight into the correctness of the survey. For example, for the first question, which is “It is easy to enter username and password”, there were 8 responses of “strongly agree” (5 points) and 2 responses to “agree” (3 points). Thus the weighted average for this question becomes \[ \frac{8(5) + 2(3)}{10} = 4.6 \]. The following Table 4.1 shows the weighted average for all the questions in the survey:

Table 4.1 User Responses

<table>
<thead>
<tr>
<th>Question</th>
<th>Weighted Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is easy for me to enter my username and password</td>
<td>[ \frac{8(5) + 2(3)}{10} = 4.6 ]</td>
</tr>
<tr>
<td>It is easy for me to enter the phrase</td>
<td>[ \frac{5(5) + 5(3)}{10} = 4.0 ]</td>
</tr>
<tr>
<td>It is easy to understand what the application is aiming to do by asking for</td>
<td>[ \frac{6(5) + 4(3)}{10} = 4.2 ]</td>
</tr>
<tr>
<td>repeated logins</td>
<td>It is easy to gather the saved information of an existing user</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>It is easy to read the error messages and understand the problem</td>
<td>$[6(5) + 3(3) + 1(2)] / 10 = 4.1$</td>
</tr>
</tbody>
</table>

As can be seen from the weighted average, the values for all responses are very close to each other. This means that the users were positive in their responses. All the responses varied from strongly agree to agree and in four cases to neutral. This shows that the users were satisfied with the interface of the application. Among the minor changes that the users wanted were mainly related to the color and font size of the interface. Also, one user mentioned that it would have been very helpful if the application had a help menu which would explain the functioning of the application.

### 4.4 False Acceptance Rate Test

To test this software tool, typing samples from the above mentioned 10 users have been collected. The ten users in this case have been asked to type a particular sentence a number of times. The sentence is “My work is important to me”. Due to constraints of data extraction, it is necessary that the given sentence be typed exactly the way it is [Lau 2004].

Each user’s typing samples is filtered for common key presses, and then the statistical model is applied to find the False Acceptance Rate (FAR) and False Rejection Rate (FRR) for key press duration. It has been observed that a user’s key press durations
are consistent for a number of samples. Further, each user has a different value of duration. False Acceptance Rate results are shown in Table 4.2.

Table 4.2: False Acceptance Rate for Users Typing the Sentence

<table>
<thead>
<tr>
<th></th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
<th>2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0%</td>
<td>0%</td>
<td>8.26%</td>
<td>25.0</td>
</tr>
<tr>
<td>0.80</td>
<td>0%</td>
<td>0%</td>
<td>3.45%</td>
<td>16.79%</td>
</tr>
<tr>
<td>0.85</td>
<td>0%</td>
<td>0%</td>
<td>1.01%</td>
<td>7.39%</td>
</tr>
<tr>
<td>0.90</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>2.01%</td>
</tr>
</tbody>
</table>

The results for False Acceptance rates i.e. FAR and False Rejection Rates i.e. FRR are actually to similar to past results conducted for this tool. As the number of standard deviations increases the FAR increases and FRR decreases. Increasing the range of standard deviation increases the chances that an unknown user’s typing samples will over-lap with the distribution of a legitimate user’s key press duration distribution and causes the FAR to increase [Lau 2004].
5. FUTURE WORK

This project uses statistical analysis approach. Other methods that could be used are neural networks and fuzzy logic. Also, a larger user trial could have been used for testing purposes. Other areas that could be considered for future work is the effect that different types of keyboards, times of day and other variables have on the ability for a user to type a consistently recognizable keystroke pattern. Another area to explore might be to see if the data capture program could be written in JavaScript or ASP.
6. CONCLUSION

The project presents a novel approach to harden the passwords by incorporating a biometric authentication method into the system. The system designed is not complicated or fancy. It is a simple implementation to achieve high efficiency. The biometric authentication tool is able to do its objectives by logging data and verifying login by using statistical analysis methods. By keeping record of a history of the profiles generated for one particular user, the tool is able to continuously adapt to the changes in the users’ typing patterns.

The False Rejection rates might be high, but the False Acceptance rate is approximately equal to 0% which does not allow intruders to compromise the system. This technique thus increases the time indefinitely that would take for an attacker to intrude into the system. Overall, the project builds a successful system to establish that typing dynamics can be used to build a secure system for user authentication.
BIBLIOGRAPHY AND REFERENCES


APPENDIX A CODE SNIPPETS

1. Main Program: New User Login:

Private Sub New_User_Click()
    Dim username As String, userphrase As String
    Dim error_username As Integer, error_userphrase As Integer

    username = InputBox("Please enter a USERNAME max. 12 characters, min. 5 characters, this can be any string of characters that you are familiar with and will be used for future logins: ", "Please Enter USERNAME")

    If Len(username) > 12 Or Len(username) < 5 Then
        MessageBox "Maximum of 12 and Minimum of 5 Characters for Username allowed !", vbOKOnly + 48, "Error"
        username = ""
        New_User_Click
        Else

        userphrase = InputBox("Please enter a PHRASE or PASSWORD of your choice, minimum 8 characters, maximum 25: ", "Please Enter USER PHRASE")

        If Len(userphrase) < 8 Or Len(userphrase) > 25 Then
            MessageBox "Length of phrase must be at least 8 characters and less than 25!", vbOKOnly + 48, "Error"
            username = ""
            userphrase = ""
            New_User_Click
            Else
                'save username/phrase to file
                Call NewProfile(username, userphrase)

            End If
        End If
    End Sub

2. Main Program: New User Profile Creation:

Private Sub NewProfile(name As String, phrase As String)
    Close
    CurrentPath = App.Path
    CommonDialog1.CancelError = True
    On Error GoTo Cancel
    CommonDialog1.FileName = name
    CommonDialog1.Filter = "Keystroke Biometrics Configuration (*.KBC)*.KBC"
    CommonDialog1.InitDir = CurrentPath

    'code to save profile

End Sub
CommonDialog1.FilterIndex = 1
CommonDialog1.ShowSave
ProfileFile = CommonDialog1.FileName

'create CONFIG file for PROFILE

current_username = name
current_userphrase = phrase
login_num = 0
successful_login_num = 0
Open ProfileFile For Output As #1
    Write #1, current_username, current_userphrase, login_num,
    successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
Close #1

'create signature file for username
file = current_username & ".UNS"
Open file For Output As #1
    Write #1, 0
    For X = 1 To ((2 * Len(current_username)) - 1)
        Write #1, vbNull, 0, 0, 0
    Next X
Close #1

'create signature file for userphrase
file = current_username & ".UPS"
Open file For Output As #1
    Write #1, 0
    For X = 1 To ((2 * Len(current_userphrase)) - 1)
        Write #1, vbNull, 0, 0, 0
    Next X
Close #1

'create signature file for phrase
file = current_username & ".SPS"
Open file For Output As #1
    Write #1, 0
    For X = 1 To 25
        Write #1, vbNull, 0, 0, 0
    Next X
Close #1

'create LOG file for username
file = current_username & ".UNL"
Open file For Output As #1
Close #1

'create LOG file for userphrase
file = current_username & ".UPL"
Open file For Output As #1

Close #1

'create LOG file for phrase
file = current_username & ".SPL"
Open file For Output As #1

Close #1

frmLearningDialog.Show

Cancel:
   Exit Sub
End Sub

3. Main Program Select User for Login:

Private Sub OpenProfile()

   Dim ProfileFile As String
   Dim CurrentPath As String
   CommonDialog1.CancelError = True
      On Error GoTo Cancel
   CurrentPath = App.Path
   CommonDialog1.Filter = "Keystroke Biometrics Configuration (*.KBC);*.KBC"
   CommonDialog1.InitDir = CurrentPath
   CommonDialog1.FilterIndex = 1
   CommonDialog1.ShowOpen
   ProfileFile = CommonDialog1.FileName

      Open ProfileFile For Input As #1
         Input #1, current_username, current_userphrase, login_num,
         successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
      Close #1

   current_file = ProfileFile
   response = MsgBox("Please enter your USERNAME,
PHRASE/PASSWORD & PRE-SELECTED PHRASE, use the TAB key to move from
one box to the next, Thankyou", 0 + 64, "Login Process")
   frmLogin.Show

34
Cancel:
    Exit Sub
End Sub

4. Load and Unload Forms in a Specific Position:

Private Sub Form_Load()
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 10000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 10000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Dim i As Integer

    ' close all sub forms
    For i = Forms.Count - 1 To 1 Step -1
        Unload Forms(i)
    Next
    If MeWindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

5. New User Registration Process:

When the OK is clicked, it performs the validations and generates a profile:

If txtUserName.Text = current_username And txtUserPhrase.Text = current_userphrase And txtPhrase.Text = "the male tiger" Then

    username_length = Len(current_username)
    userphrase_length = Len(current_userphrase)
    phrase_length = 13

    ' calculate digraphs and key depress delays from key up and key down arrays

    For X = 1 To username_length
For J = 1 To username_length
    If username_keyup_array(J).KeyChar = username_keydown_array(X).KeyChar Then
        'work out key depress duration and store in key array
        username_key_array(username_key_counter).KeyChar = username_keyup_array(J).KeyChar
        username_key_array(username_key_counter).KeyTime = (username_keyup_array(J).KeyTime - username_keydown_array(X).KeyTime)
        username_key_counter = username_key_counter + 1
    For Y = J To username_length
        'remove key up entry from array
        username_keyup_array(Y).KeyChar = username_keyup_array(Y + 1).KeyChar
        username_keyup_array(Y).KeyTime = username_keyup_array(Y + 1).KeyTime
    Next Y
    Exit For
    Next J

    'work out inter-key delay and store in array

    username_key_array(username_key_counter).KeyChar = username_keydown_array(X).KeyChar & "." & username_keydown_array(X + 1).KeyChar
    username_key_array(username_key_counter).KeyTime = (username_keydown_array(X + 1).KeyTime - username_keydown_array(X).KeyTime)

    username_key_counter = username_key_counter + 1
    Next X

For X = 1 To userphrase_length

For J = 1 To userphrase_length
    If userphrase_keyup_array(J).KeyChar = userphrase_keydown_array(X).KeyChar Then
        'work out key depress duration and store in key array
        userphrase_key_array(userphrase_key_counter).KeyChar = userphrase_keyup_array(J).KeyChar
    Next J

    'work out inter-key delay and store in array

    userphrase_key_array(userphrase_key_counter).KeyChar = userphrase_keydown_array(X).KeyChar & "." & userphrase_keydown_array(X + 1).KeyChar
    userphrase_key_array(userphrase_key_counter).KeyTime = (userphrase_keydown_array(X + 1).KeyTime - userphrase_keydown_array(X).KeyTime)

    userphrase_key_counter = userphrase_key_counter + 1
    Next X
userphrase_key_array(userphrase_key_counter).KeyTime = 
(userphrase_keyup_array(J).KeyTime - userphrase_keydown_array(X).KeyTime)
userphrase_key_counter = userphrase_key_counter + 1
For Y = J To userphrase_length
  'remove key up entry from array
  userphrase_keyup_array(Y).KeyChar = userphrase_keyup_array(Y + 1).KeyChar
  userphrase_keyup_array(Y).KeyTime = userphrase_keyup_array(Y + 1).KeyTime
Next Y
Exit For
End If
Next J

'work out inter-key delay and store in array
userphrase_key_array(userphrase_key_counter).KeyChar = 
userphrase_keydown_array(X).KeyChar & "." & userphrase_keydown_array(X + 1).KeyChar
userphrase_key_array(userphrase_key_counter).KeyTime = 
(userphrase_keydown_array(X + 1).KeyTime - userphrase_keydown_array(X).KeyTime)
userphrase_key_counter = userphrase_key_counter + 1
Next X

For X = 1 To phrase_length
For J = 1 To phrase_length
If phrase_keyup_array(J).KeyChar = phrase_keydown_array(X).KeyChar Then
  'work out key depress duration and store in key array
  phrase_key_array(phrase_key_counter).KeyChar = phrase_keyup_array(J).KeyChar
  phrase_key_array(phrase_key_counter).KeyTime = 
  (phrase_keyup_array(J).KeyTime - phrase_keydown_array(X).KeyTime)
  phrase_key_counter = phrase_key_counter + 1
  For Y = J To phrase_length
    'remove key up entry from array
    phrase_keyup_array(Y).KeyChar = phrase_keyup_array(Y + 1).KeyChar
  Next Y
Next J
End If
Next X

37
phrase_keyup_array(Y).KeyTime = phrase_keyup_array(Y + 1).KeyTime
Next Y
Exit For

End If
Next J

'work out inter-key delay and store in array

phrase_key_array(phrase_key_counter).KeyChar =
phrase_keydown_array(X).KeyChar & ":" & phrase_keydown_array(X + 1).KeyChar
phrase_key_array(phrase_key_counter).KeyTime = (phrase_keydown_array(X + 1).KeyTime - phrase_keydown_array(X).KeyTime)

phrase_key_counter = phrase_key_counter + 1

Next X

Update_Signatures
Clear_Form
Unload frmLearningLogin

Else
MsgBox "Input has been typed incorrectly try again !", vbOKOnly + 48, "Error"
Clear_Form
End If
End Sub

Private Sub txtUserName_KeyDown(keycode As Integer, Shift As Integer)
If keycode = 8 Or keycode = 46 Then
MsgBox "Cannot Edit text or use Delete or Backspace, invalidates keystroke analysis, Start Again !", vbOKOnly + 48, "Error"
Clear_Form
Else
QueryPerformanceFrequency Freq
QueryPerformanceCounter Ctrl1

temp = Round((Ctrl1 / Freq) * 10000, rounding_factor)

If username_keydown_counter > 12 Then
MsgBox "Maximum of 12 Characters for Username allowed, Start Again !", vbOKOnly + 48, "Error"
Clear_Form
Else
    username_keydown_array(username_keydown_counter).KeyChar = Chr(keycode)
    username_keydown_array(username_keydown_counter).KeyTime = temp
    username_keydown_counter = username_keydown_counter + 1
End If
End If
End Sub

6. Verify the login to the stored profile:

Private Sub Verify_Login()
    'compares login statistics and compare to user profile
    Dim temp As String

    Dim username_sum As Single
    Dim username_total_weight As Single
    Dim username_match As Integer

    Dim userphrase_total_weight As Single
    Dim userphrase_match As Integer
    Dim userphrase_sum As Single

    Dim phrase_total_weight As Single
    Dim phrase_match As Integer
    Dim phrase_sum As Single
    'keystroke verification array for username
    Dim username_verification_array(31) As Verification

    'keystroke verification array for userphrase
    Dim userphrase_verification_array(53) As Verification

    'keystroke verification array for phrase
    Dim phrase_verification_array(31) As Verification

    '\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n
    file = current_username & ".KBC"
    Open file For Input As #1
        Input #1, current_username, current_userphrase, login_num,
        successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
    Close #1

39
Open file For Output As #1
    Write #1, current_username, current_userphrase, login_num,
    successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
Close #1

'\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n'
temp = temp + Str(userphrase_key_array(k).KeyTime) + ","
Next k
Write #1, temp
Close #1

'-----------------------------------------------------
'//read signature file for PHRASE\\
'\~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
file = current_username & ".SPS"
Open file For Input As #1
For k = 1 To 25
    Input #1, phrase_signature_array(k).KeyChar,
    phrase_signature_array(k).KeyMean, phrase_signature_array(k).KeySum,
    phrase_signature_array(k).KeySumSqr
Next k
Close #1

'append LOG file for phrase
temp = ""
file = current_username & ".SPL"
Open file For Append As #1
For k = 1 To 25
    temp = temp + Str(phrase_key_array(k).KeyTime) + ","
Next k
Write #1, temp
Close #1
'///////
'/////Generate LOGIN STATISTICS for USER NAME and Write into Verification
Array\\\\\\\\
\\\\\\\\

For k = 1 To (2 * (Len(current_username)) - 1)
    username_verification_array(k).StdDev =
    Sqr((username_signature_array(k).KeySumSqr / username_login_num) -
    ((username_signature_array(k).KeySum / username_login_num) ^ 2))
    username_verification_array(k).IntermediateWeight =
    (username_signature_array(k).KeyMean / username_verification_array(k).StdDev)
Next k
' find the sum of the intermediate weights

For k = 1 To (2 * (Len(current_username)) - 1)
    username_sum = username_sum +
    username_verification_array(k).IntermediateWeight
Next k

    For k = 1 To (2 * (Len(current_username)) - 1)
        username_verification_array(k).FinalWeight =
        ((username_verification_array(k).IntermediateWeight) * (1 / username_sum))
    Next k

    For k = 1 To (2 * (Len(current_username)) - 1)
        If ((username_key_array(k).KeyTime > (username_signature_array(k).KeyMean -
        username_verification_array(k).StdDev)) And (username_key_array(k).KeyTime <
        (username_signature_array(k).KeyMean + username_verification_array(k).StdDev)))
        Then
            username_match = username_match + 1
            username_total_weight = username_total_weight +
            username_verification_array(k).FinalWeight
        End If
    Next k

    If (username_total_weight > 0.5 Or ((username_match / (2 * (Len(current_username))
    - 1)) > 0.75)) Then
        verified = verified + 1
        username_verified = True
    End If


'///////////
'///////////Generate LOGIN STATISTICS for userphrase and Write into Verification
Array'///////////
'\\\\\\\\
    For k = 1 To (2 * (Len(current_userphrase)) - 1)
        userphrase_verification_array(k).StdDev =
        Sqr((userphrase_signature_array(k).KeySumSqr / userphrase_login_num) -
        ((userphrase_signature_array(k).KeySum / userphrase_login_num) ^ 2))
        userphrase_verification_array(k).IntermediateWeight =
        (userphrase_signature_array(k).KeyMean / userphrase_verification_array(k).StdDev)
    Next k
    ' find the sum of the intermediate weights

    For k = 1 To (2 * (Len(current_userphrase)) - 1)
        userphrase_sum = userphrase_sum +
        userphrase_verification_array(k).IntermediateWeight
    Next k

    For k = 1 To (2 * (Len(current_userphrase)) - 1)
userphrase_verification_array(k).FinalWeight =
((userphrase_verification_array(k).IntermediateWeight) * (1 / userphrase_sum))
Next k

For k = 1 To 2 * (Len(current_userphrase)) - 1
    If ((userphrase_key_array(k).KeyTime > (userphrase_signature_array(k).KeyMean -
    userphrase_verification_array(k).StdDev)) And (userphrase_key_array(k).KeyTime <
    (userphrase_signature_array(k).KeyMean + userphrase_verification_array(k).StdDev)))
        Then
            userphrase_match = userphrase_match + 1
            userphrase_total_weight = userphrase_total_weight +
        userphrase_verification_array(k).FinalWeight
    End If
Next k

If (userphrase_total_weight > 0.5 Or ((userphrase_match / (2 *
    (Len(current_userphrase)) - 1)) > 0.75)) Then
    verified = verified + 1
    userphrase_verified = True
End If

'///////
'///////Generate LOGIN STATISTICS for phrase and Write into Verification
Array\\\\\\\\
\\\\\\\\
For k = 1 To 25
    phrase_verification_array(k).StdDev = Sqr((phrase_signature_array(k).KeySumSqr /
    phrase_login_num) - ((phrase_signature_array(k).KeySum / phrase_login_num) ^ 2))
    phrase_verification_array(k).IntermediateWeight =
    (phrase_signature_array(k).KeyMean / phrase_verification_array(k).StdDev)
Next k
' find the sum of the intermediate weights

For k = 1 To 25
    phrase_sum = phrase_sum + phrase_verification_array(k).IntermediateWeight
Next k

For k = 1 To 25
    phrase_verification_array(k).FinalWeight =
    ((phrase_verification_array(k).IntermediateWeight) * (1 / phrase_sum))
Next k

For k = 1 To 25
If (phrase_key_array(k).KeyTime > (phrase_signature_array(k).KeyMean - phrase_verification_array(k).StdDev)) And (phrase_key_array(k).KeyTime < (phrase_signature_array(k).KeyMean + phrase_verification_array(k).StdDev)) Then
    phrase_match = phrase_match + 1
    phrase_total_weight = phrase_total_weight + phrase_verification_array(k).FinalWeight
End If
Next k

If (phrase_total_weight > 0.5 Or ((phrase_match / 25) > 0.75)) Then
    verified = verified + 1
    phrase_verified = True
End If

Update_Signatures

End Sub

7. Update the signature profiles with each trial

Private Sub Update_Signatures()

Dim temp As String
' if sample information is within tolerance in comparison to the signature, updates
signature accordingly

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Open file For Input As #1
For k = 1 To (2 * (Len(current_username)) - 1)
    Input #1, username_signature_array(k).KeyChar,
    username_signature_array(k).KeyMean, username_signature_array(k).KeySum,
    username_signature_array(k).KeySumSqr
    username_signature_array(k).KeySum =
    username_signature_array(k).KeySum + username_key_array(k).KeyTime
    username_signature_array(k).KeySumSqr =
    username_signature_array(k).KeySumSqr + (username_key_array(k).KeyTime ^ 2)
    username_signature_array(k).KeyMean =
    (username_signature_array(k).KeySum / username_login_num)
Next k
Close #1
Open file For Output As #1
For k = 1 To (2 * (Len(current_username)) - 1)
    Write #1, username_key_array(k).KeyChar,
    username_signature_array(k).KeyMean, username_signature_array(k).KeySum,
    username_signature_array(k).KeySumSqr
Next k
Close #1
End If

/////////////////////////////////////////////////////////////////////
'///update signature file for USERPHRASE\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n
If userphrase_verified = True Then

    file = current_username & ".KBC"
    Open file For Input As #1
        Input #1, current_username, current_userphrase, login_num,
        successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
    Close #1

    userphrase_login_num = userphrase_login_num + 1

    Open file For Output As #1
        Write #1, current_username, current_userphrase, login_num,
        successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
    Close #1

    file = current_username & ".UPS"
    Open file For Input As #1
For k = 1 To (2 * (Len(current_userphrase)) - 1)
    Input #1, userphrase_signature_array(k).KeyChar,
    userphrase_signature_array(k).KeyMean, userphrase_signature_array(k).KeySum,
    userphrase_signature_array(k).KeySumSqr
    userphrase_signature_array(k).KeySum =
    userphrase_signature_array(k).KeySum + userphrase_key_array(k).KeyTime
    userphrase_signature_array(k).KeySumSqr =
    userphrase_signature_array(k).KeySumSqr + (userphrase_key_array(k).KeyTime ^ 2)
    userphrase_signature_array(k).KeyMean =
    (userphrase_signature_array(k).KeySum / userphrase_login_num)
    Next k
Close #1

Open file For Output As #1
    For k = 1 To (2 * (Len(current_userphrase)) - 1)
        Write #1, userphrase_key_array(k).KeyChar,
        userphrase_signature_array(k).KeyMean, userphrase_signature_array(k).KeySum,
        userphrase_signature_array(k).KeySumSqr
        Next k
Close #1

End If

' //update signature file for PHRASE\\
' \\
If phrase_verified = True Then
    file = current_username & ".KBC"
    Open file For Input As #1
        Input #1, current_username, current_userphrase, login_num,
        successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
    Close #1

    phrase_login_num = phrase_login_num + 1

    Open file For Output As #1
        Write #1, current_username, current_userphrase, login_num,
        successful_login_num, username_login_num, userphrase_login_num, phrase_login_num
    Close #1

    file = current_username & ".SPS"
    Open file For Input As #1
        For k = 1 To 25
Input #1, phrase_signature_array(k).KeyChar,
phrase_signature_array(k).KeyMean, phrase_signature_array(k).KeySum,
phrase_signature_array(k).KeySumSqr
    phrase_signature_array(k).KeySum = phrase_signature_array(k).KeySum +
phrase_key_array(k).KeyTime
    phrase_signature_array(k).KeySumSqr =
phrase_signature_array(k).KeySumSqr + (phrase_key_array(k).KeyTime ^ 2)
    phrase_signature_array(k).KeyMean = (phrase_signature_array(k).KeySum /
phrase_login_num)
Next k
Close #1

Open file For Output As #1
For k = 1 To 25
    Write #1, phrase_key_array(k).KeyChar,
phrase_signature_array(k).KeyMean, phrase_signature_array(k).KeySum,
phrase_signature_array(k).KeySumSqr
Next k
Close #1

End If

End Sub

8. Clear Form

Private Sub Clear_Form()
    'clears keystroke logs for inputs by clearing arrays and counters
    verified = 0

    username_verified = False
    userphrase_verified = False
    phrase_verified = False

    username_keydown_counter = 1
    username_keyup_counter = 1
    username_key_counter = 1

    userphrase_keydown_counter = 1
    userphrase_keyup_counter = 1

    userphrase_key_counter = 1
    phrase_keydown_counter = 1
    phrase_keyup_counter = 1
    phrase_key_counter = 1
For i = 1 To 15
    username_keydown_array(i).KeyChar = ""
    username_keydown_array(i).KeyTime = 0
    username_keyup_array(i).KeyChar = ""
    username_keyup_array(i).KeyTime = 0
    username_key_array(i).KeyChar = ""
    username_key_array(i).KeyTime = 0
    phrase_keydown_array(i).KeyChar = ""
    phrase_keydown_array(i).KeyTime = 0
    phrase_keyup_array(i).KeyChar = ""
    phrase_keyup_array(i).KeyTime = 0
Next i

For i = 1 To 26
    username_key_array(i).KeyChar = ""
    username_key_array(i).KeyTime = 0
    userphrase_keydown_array(i).KeyChar = ""
    userphrase_keydown_array(i).KeyTime = 0
    userphrase_keyup_array(i).KeyChar = ""
    userphrase_keyup_array(i).KeyTime = 0
    phrase_key_array(i).KeyChar = ""
    phrase_key_array(i).KeyTime = 0
Next i

For i = 1 To 53
    userphrase_key_array(i).KeyChar = ""
    userphrase_key_array(i).KeyTime = 0
Next i
'remove after debug
Label2.Caption = "Login Number " + Str(login_num)
txtUserName.Text = ""
txtUserPhrase.Text = ""
txtPhrase.Text = ""
txtUserName.SetFocus

End Sub
APPENDIX B TEST RESULTS

Test results have been included in this section.
Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   - Strongly agree  O  Agree  O Neutral  O Disagree  O Strongly Disagree

2. It is easy for me to enter the phrase
   - Strongly agree  O  Agree  O Neutral  O Disagree  O Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   - Strongly agree  O  Agree  O Neutral  O Disagree  O Strongly Disagree

4. It is easy to gather the saved information of an existing user
   - Strongly agree  O  Agree  O Neutral  O Disagree  O Strongly Disagree

5. It is easy to read the error messages and understand the problem
   - Strongly agree  O  Agree  O Neutral  O Disagree  O Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   - Yes, No problem. It is secure.

7. What did you like least about the application? Detail at least 2 features.
   - Does not run on Mac Comp.
   - Can have better interface with help.
8. Would you want to use the same application for the purpose it has been designed for? Why or Why not?

Yes. It works.

Thank You for your time.
Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   ○ Strongly agree    ○ Agree    ○ Neutral    ○ Disagree    ○ Strongly Disagree

2. It is easy for me to enter the phrase
   ○ Strongly agree    ○ Agree    ○ Neutral    ○ Disagree    ○ Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   ○ Strongly agree    ○ Agree    ○ Neutral    ○ Disagree    ○ Strongly Disagree

4. It is easy to gather the saved information of an existing user
   ○ Strongly agree    ○ Agree    ○ Neutral    ○ Disagree    ○ Strongly Disagree

5. It is easy to read the error messages and understand the problem
   ○ Strongly agree    ○ Agree    ○ Neutral    ○ Disagree    ○ Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   Considering the purpose of the application, I find that it is acceptable (and comforting) that I have to enter the same information 10 times.

7. What did you like least about the application? Detail at least 2 features.
   Design
   Font size not big enough for me
   Too many prompts
Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

2. It is easy for me to enter the phrase
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

4. It is easy to gather the saved information of an existing user
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

5. It is easy to read the error messages and understand the problem
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   - [ ] Yes, it makes it more secure. Since it is used for high security reasons only it's ok.

7. What did you like least about the application? Detail at least 2 features.
   - Not enough info as to why I am entering the data it is asking for.
8. Would you want to use the same application for the purpose it has been designed for? Why or Why not?

Yes, it is more secure as it measures my personal traits which are unique to me.

Thank You for your time.
Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

2. It is easy for me to enter the phrase
   - [ ] Strongly agree  - [ ] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   - [ ] Strongly agree  - [x] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

4. It is easy to gather the saved information of an existing user
   - [ ] Strongly agree  - [x] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

5. It is easy to read the error messages and understand the problem
   - [ ] Strongly agree  - [x] Agree  - [ ] Neutral  - [ ] Disagree  - [ ] Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   - [ ] Definitely feel secure!

7. What did you like least about the application? Detail at least 2 features.
   - Better user interface would be nice!
   - More attractive colors!
Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   ○ Strongly agree  ○ Agree  ○ Neutral  ○ Disagree  ○ Strongly Disagree

2. It is easy for me to enter the phrase
   ○ Strongly agree  ○ Agree  ○ Neutral  ○ Disagree  ○ Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   ○ Strongly agree  ○ Agree  ○ Neutral  ○ Disagree  ○ Strongly Disagree

4. It is easy to gather the saved information of an existing user
   ○ Strongly agree  ○ Agree  ○ Neutral  ○ Disagree  ○ Strongly Disagree

5. It is easy to read the error messages and understand the problem
   ○ Strongly agree  ○ Agree  ○ Neutral  ○ Disagree  ○ Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   Yes, it's important

7. What did you like least about the application? Detail at least 2 features.
   The lack of help now
Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   Ø Strongly agree  O Agree  O Neutral  O Disagree  O Strongly Disagree

2. It is easy for me to enter the phrase
   Ø Strongly agree  O Agree  O Neutral  O Disagree  O Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   O Strongly agree  Ø Agree  Ø Neutral  O Disagree  O Strongly Disagree

4. It is easy to gather the saved information of an existing user
   O Strongly agree  Ø Agree  Ø Neutral  O Disagree  O Strongly Disagree

5. It is easy to read the error messages and understand the problem
   Ø Strongly agree  O Agree  Ø Neutral  O Disagree  O Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   YES

7. What did you like least about the application? Detail at least 2 features.
   The GUI could use some improvement.
Name: Eric Young
Position Title: Developer
Date: 5/17/2006

Questionnaire Survey for Biometrics Authentication Application

1. It is easy for me to enter my username and password
   - Strongly agree  O  Agree  O  Neutral  O  Disagree  O  Strongly Disagree

2. It is easy for me to enter the phrase
   - Strongly agree  O  Agree  O  Neutral  O  Disagree  O  Strongly Disagree

3. It is easy to understand what the application is aiming to do by asking for repeated logins
   - Strongly agree  O  Agree  O  Neutral  O  Disagree  O  Strongly Disagree

4. It is easy to gather the saved information of an existing user
   - Strongly agree  O  Agree  O  Neutral  O  Disagree  O  Strongly Disagree

5. It is easy to read the error messages and understand the problem
   - Strongly agree  O  Agree  O  Neutral  O  Disagree  O  Strongly Disagree

6. Are you comfortable with the fact that you have to enter the same information 10 times? Please keep in mind the purpose of the application
   - Sure.

7. What did you like least about the application? Detail at least 2 features.
   1. Font size too small
   2. During learning process, if profile wasn't been created, I shouldn't have to click "next" button to re-enter (should just pop up for me since learning requirements haven't yet been fulfilled)
8. Would you want to use the same application for the purpose it has been designed for? Why or Why not?

Maybe  Not

Thank You for your time.
8. Would you want to use the same application for the purpose it has been
designed for? Why or Why not?

N/A

Thank You for your time.
8. Would you want to use the same application for the purpose it has been designed for? Why or Why not?

Thank You for your time.