Keystroke Dynamics
How typing characteristics differ from one application to another

Hafez Barghouthi
hafez.barghouthi@hig.no
mobile: 96677276

Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2009

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

This Master thesis is about continuous authentication using keystroke dynamics. We will look to the typing characteristics in different applications and how those characteristics vary from one application to another. It is proved that there are differences in those typing characteristics when the user try to use different applications, However we need to investigate those differences to see if we can still authenticate a person regardless which application is used. In other words, by looking at a selected numbers of applications, we are trying to answer a question "***'Does a template based on a certain application allow a person to use another application and still be authenticated by the system*** "***'?***. Furthermore we will look if it is possible to authenticate a person based on one general template or we need a set of application dependent templates?.

Preface

I would like to express the deepest appreciation to my supervisor, Dr. Patrick Bours for accepting me as his student and for the support he gave me during the last six months. Without his guidance and persistent help, this thesis would not have been possible. Thank you a lot my dear supervisor.

I am very grateful to all my colleagues and friends at Gjovik University College. I would like to thank Mohammad Derawi for his valuable help in C# and for his interest and support. I would like to thank all the people who have participated in the experiment.

Last but not least, I will not forget my beloved parents and my wife. Without whom I would never have been able to achieve so much. Especially to the fact that they gave me all the supports over the years to achieve what I want.

Hafez Barghouthi, 2009/05/20

Contents

1 Introduction

1.1 Topic

One of the most important things before giving a person access to any resource is to identify or authenticate him first. A password is one way to do this. A user gives his user name (claiming an identity) and then gives his password (proving ownership of the claimed identity). However, there are some problems in using passwords. One of them is that long or short passwords can be forgotten if they have a random combination of various characters (difficult to memorize). Another problem with passwords is that they can be guessed easily when they are just derived from dictionary words or even that they can be stolen easily if they are written down by their owner. Tokens are a second approach to being authenticated (through something you have), however they can be forgotten, lost or even stolen by attackers. Biometrics is a third approach for identifying and authenticating people based on what they are. For example it is highly likely (but not proven) that everyone has different fingerprints that can be used to differentiate one person from another.

Biometrics can be divided into two categories, physiological and behavioral. The first category contains the features that are physically related to a person, for example iris, fingerprints and retina. The second category contains features that people have learned to do in a stable manner. Examples in this category are walking (gait), writing a signature and typing on a keyboard (keystroke dynamics).

In this project we will look at keystroke dynamics as a method for authentication. By keystroke dynamics we mean the way that a person types. This can be characterized by timing when keys are pressed down or released. Other characteristics pressure, angle of pressing the key and more, in which case we need special hardware, e.g. a special keyboard or a camera.There are two types of keystroke dynamics. The first one is the static keystroke dynamics in which the data that is typed is fixed and also the time this information is typed in is fixed (during login time or after a predetermined period of time). The second one is continuous keystroke dynamics in which case the typing characteristics are analyzed during a complete session. The literature concerning keystroke dynamics is focusing more on the static type, while less literature can be found on continuous type. Many experiments in this field have a small error rate meaning that we can authenticate people in a good manner using keystroke dynamics[1].

The point is that we can authenticate people through their typing behavior. However, we know in advance that the typing characteristics are different when a person uses different applications, a different keyboard or types in a different language. The mentioned topics raise a lot of open questions related to keystroke dynamics. In this project we will try to find an answer to one of these questions: ***How typing characteristics differ from one application to another and do these differences interfere with the authentication process?***

[1] But still larger then for example fingerprints or iris

1.2 Keywords

Biometrics, Authentication, Identification, Keystroke Dynamics, Different applications.

1.3 Problem description

Many experiments which are done to investigate keystroke dynamics as an authentication method have a low error rate (between 1.17% and 5%), meaning that we can rely on such kinds of techniques to authenticate people [1] [2] [3] . Some of the previous studies have proven that keystroke dynamic authentication is resistance against some type of attacks like shoulder sniffing but still weak against some attacks in which the attacker has feedback about the typing characteristics of the legitimate person [4].

Not a lot of research has been done on how different applications affect the typing characteristics of the user. One of the publications of Furnell is discussing the effect of different application in the keystroke dynamics field [5]. It is important to know if we can still depend on continuous keystroke dynamics to authenticate people when they run different applications. There might be a large difference in typing characteristics when chatting on MSN compared to writing a program in Java. You need to think, to analyze and then to type when you are writing a Java program while in MSN chatting the situation is different. Furthermore when you are writing a Java program you will use more special characters than when you are writing for example a formal letter using Microsoft Word. Our target in this project is to investigate this problem, and try to assure the stability of keystroke dynamics techniques.

1.4 Justification, motivation and benefits

Keystroke dynamics will strengthen the security of the system.Even after logging into a computer system, the user needs to know how to type. The typing rhythm should match with the legitimate users typing rhythm. Even when the user switches to another application, the system must have the ability to authenticate the user without any problem. There is a high demand to agree on a certain template to be used in order to authenticate a user regardless which application is used.

1.5 Research questions

In this project we want to investigate the following two research questions:

- How are typing characteristics different from one type of application to another and where are the similarities?
- How we can benefit from the results of those differences and similarities to generate a reliable template to authenticate a user regardless which application is used?
- Is it possible to authenticate a person based on one general template or we need a set of application dependent templates?
- Is it possible to say that the typing characteristics in one type of application are more stable than in another?

Other minor questions will arise like:

- How we can design an experiment to measure the differences in typing characteristics?

- How many participants should participate in an experiment and how long should the experiment last?
- Which applications should be chosen?

1.6 Planned contributions

This project will come up with results in how some different applications affect the ability to recognize people using keystroke dynamics. In other words we are going to design an experiment to measure differences in typing characteristics when a user try to use different kinds of applications. If the changing of the application do not significantly impact the recognizing process or if it would be possible to do some special processing according to what the application is, there should be no problems using a keystroke dynamics to authenticate user regardless which application he used.This of course will strengthen the reliability of keystroke dynamics as an authentication method.

2 Keystroke Dynamics

This chapter will give a brief and general introduction into authentication, biometrics and keystroke dynamics. In order to understand the content of this thesis, it is important to be familiar with the basic terminology that will be introduced in the following sections.

2.1 Authentication

A process which verifies if someone is, in fact, who he/she claims to be is used in many places today. For examples passwords are needed to get access to computers, PIN codes are needed to get money from ATM machines and traveling documents are needed in cross border controls. This process is called authentication. Authentication is proving ownership of the claimed identity. There are different ways in which a user can be authenticated. However all of these ways can be categorized into one of three classes [6]:

- Something you know e.g. password.
- Something you have e.g. token.
- Something you are e.g. biometric property.

In the following subsections a brief description for each class is provided.

2.1.1 Know

It basically means providing knowledge of a secret, e.g. in the form of passwords to get access to a computer or a specific resource. Another example is a PIN code to get money out of an ATM machine. Some advantages of this factor include less cost, easy to implement and fast authentication mechanism. However for many reasons keeping secrets is not the perfect way to authenticate users. Those secrets are easy to forget or can be stolen when you write them down especially when users are forced to remember many different passwords. When users choose easy passwords like birthdays, family names, pet or any combination of these, it seems that the problem of forgetting a password or the need to write it down is solved. However this may lead to the fact that attackers can easily guess the password and misuse it using social engineering techniques or brute force attacks. All these drawbacks increase the limitation of using something you know as an ideal authentication mechanism.

2.1.2 Have

This means providing a unique piece of hardware that can be matched to the user identity. Such hardware can be a key, a token, a smart card, a SIM card, etc. This authentication class has an advantage over the previous one in that there is no need to memorize something difficult which can be easily forgotten as mentioned in Section 2.1.1. However there are some drawbacks of this factor such as that it is more expensive, not only the piece of hardware the user has, but also the equipment used at the verification side. It is also important to take required actions whenever

the hardware is either lost or stolen [6].

2.1.3 Are

Basically this class is utilizing biometric properties. Nowadays one can find various types of biometrics applications in real life. Most of biometric features are unique per person and they can be found in almost all people in some way or another. For example fingerprints are unique even with identical twins. One advantage of this class over the previous two is the difficulty for attackers to steal the biometric item from the legitimate user. However it is not impossible. The difficulty of this depends on what kind of biometric property is used and how it is used [6].

Biometrics can be divided in two categories: physiological and behavioral. Physiological means features that are based on stable physiological characteristic, e.g. fingerprint, iris, or retina. While behavioral means features based on learned and alterable behavioral characteristic, e.g. keystroke dynamics, signature, or gait [7].

2.1.4 Combination of authentication factors

In case of getting money from an ATM machine, two things must be provided: the bank card and the PIN code. In such case two factors of authentication are combined. To be specific: something you know + something you have. The target of this combination is to increase the security of systems. Multi modal systems [1] also can be used to combine features from the same class like a combination of fingerprints and gait where both methods belong to the same authentication class "something you are".

2.2 Biometrics

Biometric identification has a long history. The use of biometrics was known in 14th century by the Chinese. Chinese merchants were stamping children's palm and foot prints with ink in order to distinguish young children from another [8]. Even in Shakespeare's play "*The tempest*" a hint toward gait as a biometric feature was given:"*Great Juno comes: I know her by her gait*". Around 1870 Alphonse Bertillon described a system of body measurement for identifying people which was used until the 1920 in the United States to identify prisoners [6]. Work on fingerprint recognition started in the 1880's by Henry Faulds, William Herschel and Sir Francis Galton. For a long time fingerprints were almost the only used biometric feature to authenticate people. Hand geometry, voice, signature and retina recognition have been used since the 1980 and commercial face and iris recognition has been around since the 90's. Keystroke dynamics and gait applications have only been an area of research during the last decade.

2.2.1 Biometric Features

According to [7],certain properties must be present in biometric features (also called biometric characteristics) in order to use those features in biometric systems and to be practical:

- Universality: each person should have the characteristics.
- Distinctiveness: different persons should be sufficiently different in terms of the characteristics.

[1] Another name for combined factors systems

- Permanence: stability of the characteristics over a period of time.
- Collectability: the characteristics can be measured quantitively.

The above properties are needed to be able to use the system and to make sure that the performance of distinguishing between different persons is achieved. Three properties are needed to make the system more practical and secure:

- Performance: refers to accuracy, speed and robustness in technology used.
- Acceptability: the degree of acceptance to use such a particular biometric identifier in the daily life.
- Circumvention: resistance of the system against fraudulent methods.

In general a practical system should have all of these properties to ensure high accuracy and speed in accepting legitimate users and rejecting impostors. It is also necessary to prevent that multiple people use the same identity (positive recognition) and also prevent the same person from using multiple identities (negative recognition)[6].

2.2.2 Biometric systems

According to [9], a biometric system is"*the automated identification or verification to human identity through repeatable measurement of physiological and/or behavioral characteristics*". It consists of two phases or subsystems. The first one is the enrollment phase where the biometric features are extracted from the user and converted into a template that will be stored in the system database in order to be used in the second phase. The second phase is the verification phase where the identity of the user is checked against the data obtained during the enrollment phase. Figure 1 shows the two phases of biometric system.

During the enrollment phase the biometric features of the user are transformed into a template. This transformation is necessary for two reasons. First it is due to legal aspects as it is not expected to reveal significant information about the original data of the user [10]. The second reason is that most of the biometric systems do not store raw biometric data because it can be unpractical since the template is used for comparison. In the authentication phase the user again presents his/her biometric characteristics which are extracted and then matched against the template(s) that correspond to the claimed identity of the user. A distance metric should be used to know how far/close the extracted features are from the template. Finally a decision rule should be used to determine whether the user is rejected or accepted to the system. This decision depends on a predefined threshold value which is important to calculate some of the error types related to the biometric system.

2.2.3 Calculations and error types

Biometric systems certainly offer alluring advantages over other factors of authentication. While keys and passwords can be replicated or stolen, there is such a limited possibility in biometrics systems. Furthermore, the credit cards, ATM cards and other such provisions can't be misused, when accompanied with biometric test. Sharing of biometrics characteristics is not possible and thus, they can be used to prevent the same user from using two different identities (negative recognition). However there is also a specific drawback attached to the biometric authentication

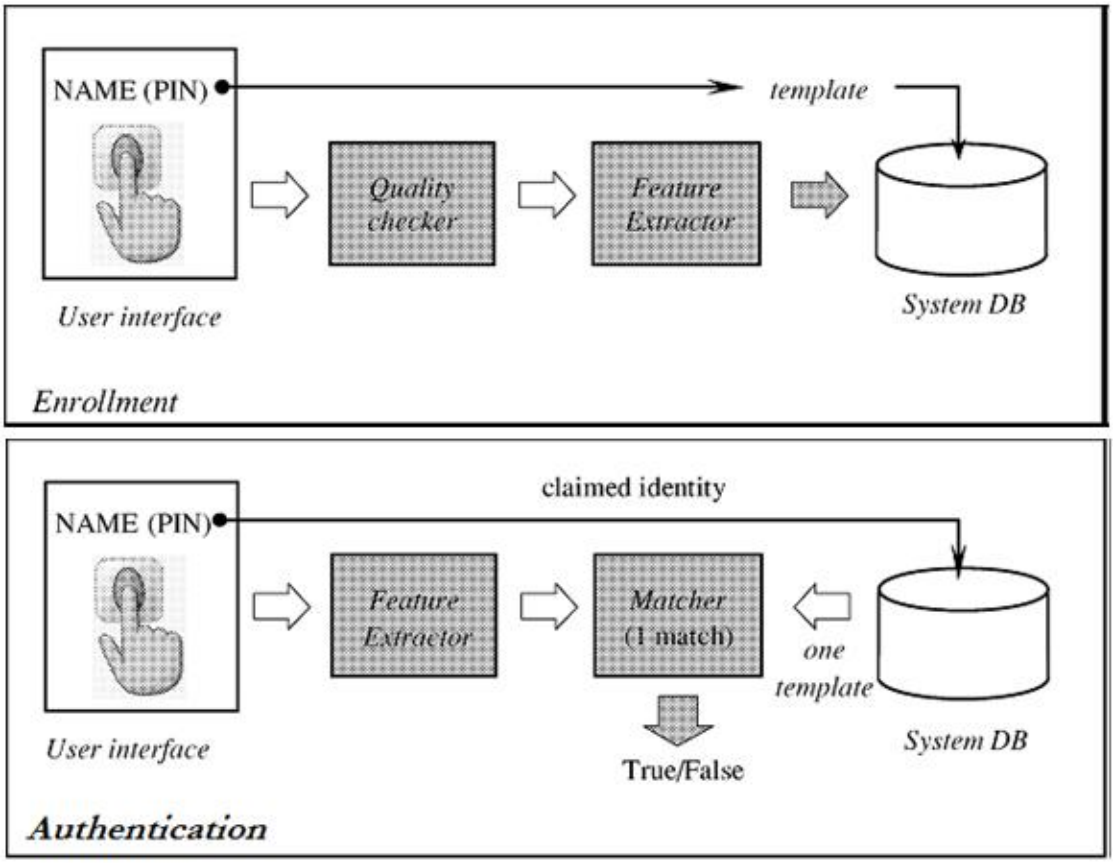

Figure 1: The two phases for biometric system [7].

but not to the other classes. With something you know, either you know the secret or you do not know it. With something you have either the hardware fits or it does not fit, so it is 100% correct or 100% wrong. However with something you are, the biometric features can never match 100%. For example looking at a finger print, it will be some sample features that matches the template and other features that do not match the template. The more matches we find the more convinced we are that a legitimate user tries to authenticate himself to the system. The differences between the extracted sample for authentication and the template sample is giving by the distance value. This value is calculated by using a distance metric for example the sum of absolute distances between corresponding values in two sets (extracted set and template set).

The idea behind a distance metric is to give in principal a small intra-class value, meaning that sets from the same user have a low distance value, and a larger inter-class distance, meaning that sets from different users should give a high distance value. A decision rule, which depends on a predefined threshold should be used to decide whether a person is accepted or rejected. During this matching some errors might occur. Two types of important errors are:

- False Match Rate (FMR): This happens when a biometric system measures two different persons to be the same person. Obviously an imposter wrongly will be accepted by the system.
- False Non Match Rate (FNMR): This happens when a biometric system measures two different samples from the same person to be from a different person. A legitimate user is wrongly rejected by the system.

Not only the mentioned two errors are produced by a biometric system but also a biometric system can produce some other errors such as Failure to Enroll Rate (FER) which measures the fraction of people that cannot enroll in the system. The quality of the extracted biometric features will be checked during the enrollment phase depending on the chosen level of quality to ensure good accuracy of the system. It might be possible that some captured samples cannot be enrolled into the system therefore increasing the FER error. Another error to be mentioned is the Failure to Capture Rate (FCR). The FCR measures the percentage of unsuccessful tries when capturing the biometric features. It occurs when the device is not able to locate the biometric features when presented. This can appear for example when there is dirt on the finger when taking a finger print image. Another one is the existence of bad light condition when trying to locate a face in face recognition process. In this project we are only concerned with FMR and FNMR.

The trade off between FMR and FNMR can be illustrated using the Receiver Operating Characteristics (ROC) or Decision Error Trade off (DET) (see Figure 2). Both curves show the system performance at different threshold values and the trade off between FMR and FNMR. The difference between ROC and DET curves is that the DET curve plots false negatives (FNMR) on the Y-axis instead of true positives. While the ROC curve plots true positives (1-FNMR) instead of false negatives (FNMR). Deciding which threshold should be used is important. This depends heavily on the application. For high security application a low FMR is required in order to reject as many impostors as possible trying to access the system. However in forensics application a higher FMR is acceptable to be sure to catch the criminals. Most of the civilian application are

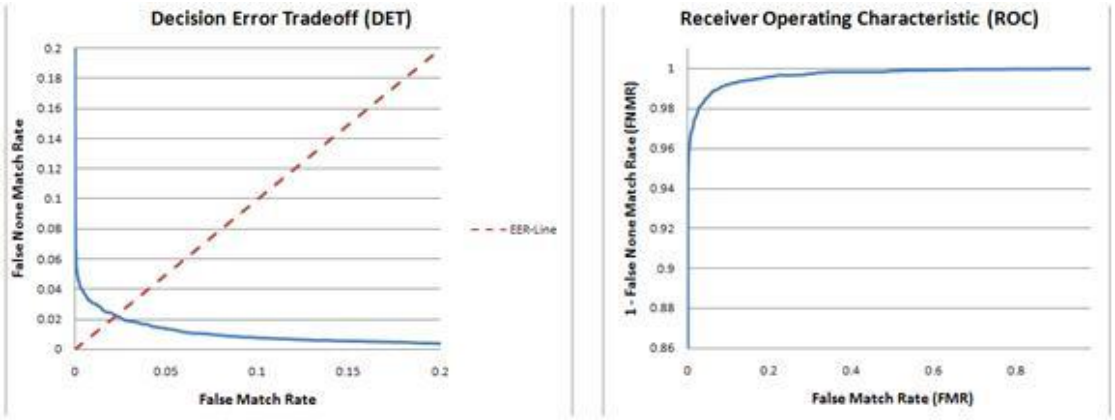

Figure 2: Examples of DET and ROC curves.

somewhere in between the previous two mentioned type of applications.

Another important point is the Equal Error Rate (EER). This rate is used to compare different systems against each other and can give a brief idea about the performance of the biometric system. However as mentioned before accuracy of the system depends on the two errors FMR and FNMR which can be calculated as given in equations 2.1 and 2.2

.

$$\text{FMR} = \frac{\text{Number of accepted impostor attempts}}{\text{Total number of impostor attempts}} \tag{2.1}$$

$$\text{FNMR} = \frac{\text{Number of rejected legitimate users}}{\text{total number of legitimate attempts}} \tag{2.2}$$

2.3 Keystroke Dynamics

Keystroke dynamics is a behavioral biometric based on the assumption that different people type in a unique manner. Neurophysiological factors make written signatures unique per person. These factors are also expected to make typing characteristics unique per person. The idea behind keystroke dynamics appeared in the 20th century when telegraph operators could recognize each other based on their distinctive patterns when keying messages over telegraph lines. Keystroke dynamics is known with other names such as keyboard dynamics, keystroke analysis, typing biometrics and typing rhythms.

2.3.1 Keystroke dynamics features

There are several different features which can be detected when the user presses keys on a keyboard. Possible features include:

- Duration (the time in which the key is held down).

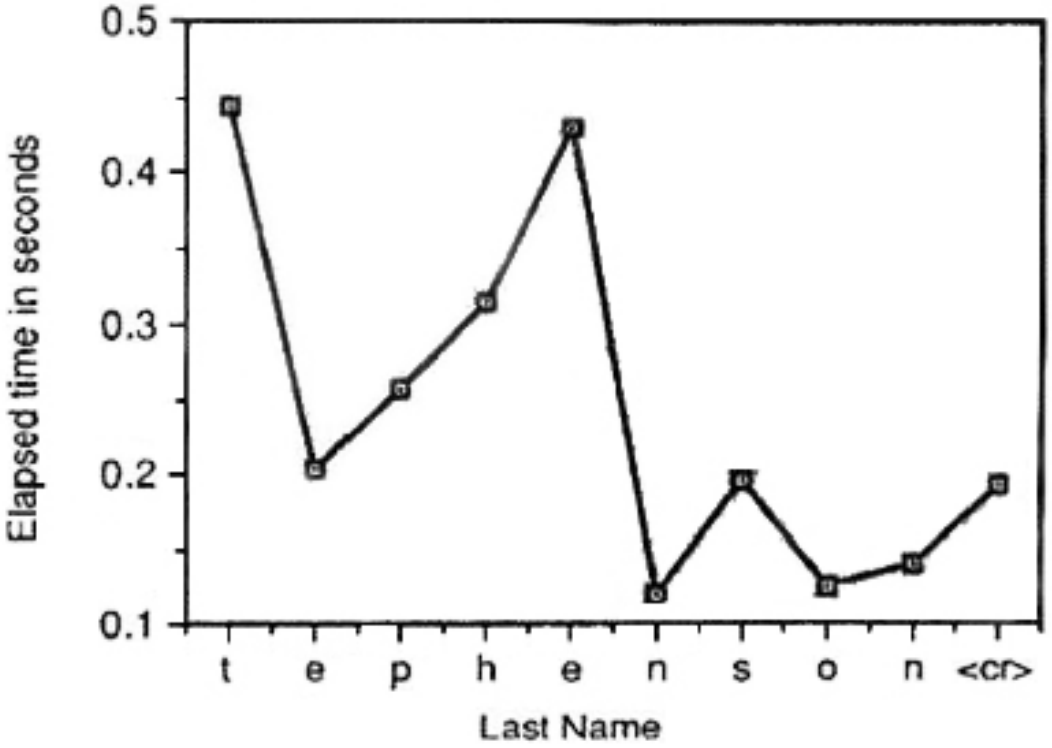

Figure 3: Latencies when typing "Stephenson"[6].

- Latency (the time between two consecutive keys [2]).
- Pressure used when hitting keys while typing (requires a special keyboard).
- Finger placement (the place where the finger is placed on the key or even the angel of the finger when pressing the key) in this case a camera is required.
- Finger choice (which finger is used for which key of the keyboard) also a camera is required.

According to [11], there are other possible measurements then the ones mentioned above such as frequency of error (how often the user has to use the backspace), the order in which the user presses keys when writing capital letters (is shift or the letter key released first) and the habit of using additional keys in the keyboard such as writing number with num pad. Systems do not necessarily employ all of these features. Most of the applications measure the first two features: duration and latency. In this master thesis we will use this approach too because this can be easily measured with only a small program and a standard keyboard (see Sections 5.1 and 5.2). Figure 3 shows an example of measuring latencies between keystrokes to find the way a person is typing his second name.

[2]Many different ways can be used to calculate latency. More details are discussed in section 2.3.3

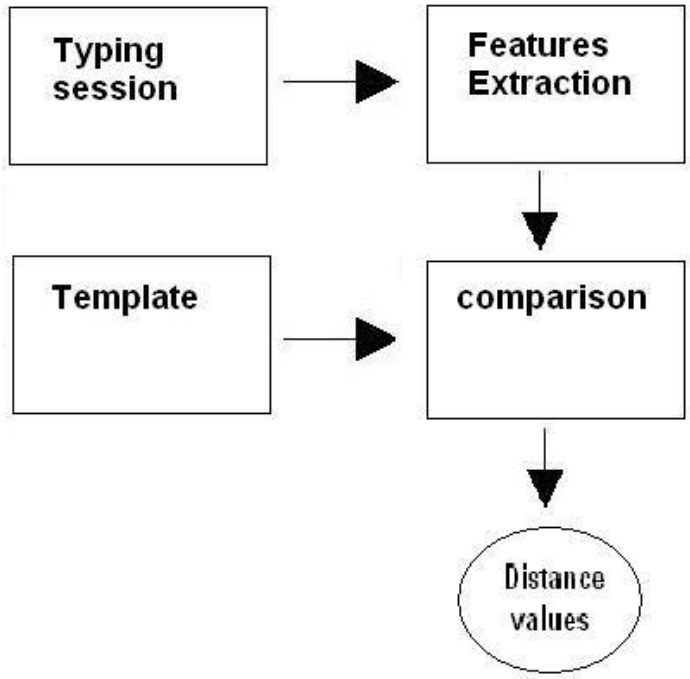

Figure 4: Keystroke Dynamics system.

2.3.2 Keystroke authentication (verification) system

As we mentioned in Section 2.2.2 a biometric system is an automated process to identify and authenticate a person based on a certain biometric features. Our target is to explain this in terms of keystroke dynamics. Figure 4 is special case from Figure 1, where a user provides some typing data from which the typing features are extracted. Then the typing features are compared with the user's stored template(s) using a distance metric. Then a decision rule is needed to determine whether the user is accepted or rejected by the system. There are two main authentication systems in keystroke dynamics: static and continuous. Static systems authenticate users based on fixed text for example at login session when the user provides the user name and password. However continuous systems authenticate users continuously during a full session, providing the possibility to reject the user at any moment during the full session. More details about the two systems are discussed in Sections 3.1and 3.2. In this master thesis our focus will be on continuous authentication.

2.3.3 Calculations

In this section we will explain the basic calculations that are used in keystroke dynamics. Specifically we will discuss three parts that are essential to do the data analysis for keystroke dynamics.

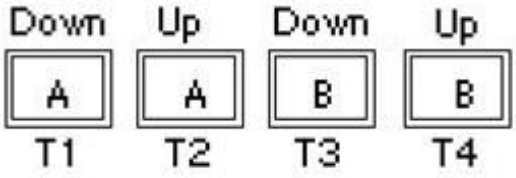

Figure 5: The key-down and key-up times for A and B are two keys, T1; T2; T3 and T4 are times when a key-up or key-down event happened.

First of them is collected timing information,second is template creation and the last is the comparison process between new input data and template.

According to [12] several timings can be calculated from capturing the time when key is pressed down and the time when key is released up. For example time between two key-downs or time between two key-ups or even time between key down and key up for the same key or time between key up for the first key and key down for the second key. The last two timings are the most popular among keystroke literatures. They are also known as duration and latency respectively.

Duration is the time a key is held down, it can be calculated by subtracting a key-down time from the key-up time for the same key. Latency is the time between two consecutive keystrokes which can be calculated by subtracting the key-up time for the first key from the key-down time for the second key. Figure 5 shows four timings T1, T2, T3 and T4 when key-down and key-up for A and B keys are used. Then the duration of A is T2-T1 and the duration of B is T4-T3 and finally the latency between A and B is T3-T2.

The latency between two keys can be negative. For example if the user presses the letter B before releasing the letter A. There are other definitions of latency, some use the key-down to key-down time. Our choice is to use the last approach of latency to assure that all latencies have positive values.

The second part is needed to generate the templates or reference profiles. A user needs to create a template before he/she can use the system. This template contains a subset of all features which the user provided. Duration and latency are the basic to build such templates. Mean, standard deviation and the number of occurrence of the key is used in this template (see Section 6.1.1).

Finally a distance metric is needed to make the comparison between the template and the new provided input data of the user. There are many different distance metrics that can be used. A property of a good distance metric is that it has a large inter-person distance and a small intra-person distance. One of a basic distance metric is the Euclidean distance. More details about this are discussed in section 6.1.3.

2.3.4 Advantages and disadvantages of Keystroke dynamics

The author of [13] mentions the advantages and disadvantages of keystroke dynamics systems. A list of advantages may include:

- Less expensive than other biometrics because it needs no additional hardware.

- Natural authentication mechanism for computers and network security.
- Possible to perform continuous authentication.
- High user acceptance and non intrusive.

As all biometrics also keystroke dynamics has some disadvantages. which may include:

- Sensitive to changes in keyboards and changes in typing languages.
- Affected by the user's physical condition (fatigue, illness and possible hand and fingers injuries).
- Not too many commercial applications.
- Awareness of certain problems such as typing errors.
- High False Non Match Rate (FNMR).

2.3.5 Applications

It seems to be hard to find literature on working systems that implement keystroke dynamics to enhance security. Biopassword is one commercial system. They did not release any information about their system. However there is a demo for this application available on their website[14]. Another available application is Typesense from Deepnet security[15]. They are both systems which use keystroke dynamics to restrict access to a system.

3 State of art

There are two types of keystroke dynamics. The first one is static keystroke dynamics in which the keystrokes are analyzed only at specific times e.g. during login. The second one is continuous keystroke dynamics in which the typing characteristics are analyzed during a complete session. Static approaches provide more robust user verification than simple passwords. However static methods do not provide continuous security, specifically they cannot detect substitution of the user after the initial verification. Continuous verification monitors the user's typing behavior throughout the session. Therefore it can be used to detect uncharacteristic typing rhythm caused by say drowsiness [16].

A lot of reports can be found on keystroke dynamics dealing with a static authentication. Less can be found on Keystroke dynamics based on continuous authentication. In this chapter we are going to talk about the literature concerning those two parts.

3.1 Static Verification

A lot of reports have tested the possibility to identify or authenticate the user during the login session i.e. when the user has to provide his or her user name and password. The authors of [2] check not only that a password is correct but they also checked the way it was typed. They achieved an FMR of 1.9% and an FNMR of 1.45%. In the same paper they also tried to mimic legitimate users by showing invalid users (impostors) how a legitimate user was typing and then try to mimic this user. They achieved an FMR of 3.66% which is worse than the previous result but still acceptable. They state that the keystroke dynamics is strong against mimicking using a shoulder sniffing attack. Furthermore they show that familiar text passwords give better results than random passwords.

The authors of [17] compared the results from fifteen scientific papers. Many authors reported FMR and FNMR less than 2.5%.However many of the good performers require users to write long text before they are authenticated. This is unacceptable in case of static authentication since it would be a very costly solution for a company, if their workers had to spend several minutes a day only to be authenticated. Other suggested solutions in this paper required the authentication system to be updated every time a new user is added, or when user typing behavior change over time. Those systems that have to be updated are almost unusable for large organizations. Only two of the fifteen reports had more than 50 participants. The majority had less than 25 participants.

One problem could arise which is mentioned in [18] in which the authors state that this method of authentication will reveal information about the passwords especially in online applications. The authors of [19] state that if latency times between keystrokes are revealed then it is much easier for an attacker to find a password. Instead of testing 50% (birthday paradox) of the password space in a brute force attack, it is only 1%-2% of the password space that has to be tested before the correct password is found. Encryption of such information (latency times) is the

effective solution for this problem. The authors of [20] have shown that keystroke dynamics can make passwords more secure against brute force attacks. Even weak passwords would require much work by the attacker. However this depends on the implementation of the system.

Two different approaches are used during the recognition phase. One of them is the statistical approach[1],[3],[12],[18],[21],[22],[23]. The other approach is the neural network classification [22],[24],[25],[26],[27],[28],[29]. The authors of [21] were following the statistical approach. They tested a scoring method based on standard deviation. They compared digraph latencies in the password, every digraph inside a threshold of n standard deviations from the mean were given the value of 1.5. Latencies outside this threshold were given the value 1. Then these values were added together and a user was accepted if the sum was higher than a predefined decision threshold. They managed to get an Equal Error Rate (EER) of 5.85%. They found that letters scattered across the keyboard provide more accuracy than letters that are close together. They used this to give latencies between some letters a higher value which reduced the EER value to 5%.

In [25] the authors classify neural networks as capable of exploring many competing hypothesis in parallel. Therefore neural networks were considered to have the greatest potential in the area of pattern recognition over the statistical approach. This idea had many followers and the authors of [29] tested Artificial Neural Network (ANN) and K-Nearest Neighbor algorithm. They had ten participants with ten different passwords, and one hundred impostors. The impostors tried to write legitimate users password a limited number of times. K-Nearest Neighbor algorithm gave an FMR of 1.03% and an FNMR of 15.4% while ANN gave an FMR of 29% and an FMNR of 1%. A very high FMR with ANN could be reduced with further training of the systems. This training phase is considered as a drawback for ANN. The authors in [28] managed to get much better results using a parallel decision tree (DT) instead of ANN. They achieved an FMR of 0.88% and an FNMR of 9.62% with a long text, and with eight characters they got an FMR of 9.19% and an FNMR of 13.97%.

Within each of these approaches, many of different distance metrics can be used. The authors of [30]used 3 different metrics Euclidean, Non Weighted Probability and Weighted Probability achieving (FMR, FNMR) of (0%,20.9%), (0%,14.4%) and (0%,9.3%) respectively. In [23] the authors tested two distance metrics: The normalized minimum distance classifier and the normalized Bayesian classifier achieving a (FMR, FNMR) of (2.8%,8.1%) and (0.5%,3.1%) respectively. In [12]the authors used three distance metrics and a sum rule to combine the sets to get the best results. The 3 distance metrics were statistical (median and standard deviation), disorder between two vectors and time classification. They achieved a final result of FMR of 0.5% and an FNMR of 6% . The author of [24]used neural network methods as a distance metric however they got problems with negative latency so they used key down to key down latency achieving at first an FMR of 6.65% and an FNMR of 2.22% but then they reduced the root mean square error (RMSE) from 0.07 to 0.03 achieving an FMR of 0% and FNMR of 1.11% .

Another original work which can be mentioned in this part is [4]in which the authors tried to make the attackers to imitate a legitimate user by giving them a feedback to learn other's typing characteristics. They have an overall achievement FMR of 2.56% and an FNMR of 10.26% .

3.2 Continuous Verification

As we mentioned before less literature can be found on the field of continuous verification. Some papers tested the possibility to verify continuously the user during a complete typing session. The authors of [1] provided that they were able to deal with typing rhythms of free text that can be chosen and entered by users without any specific constraints. They used digraphs, trigraphs and four-graphs to compare samples and template. A digraph is any set of two letters, a trigraphs is any set of three letters, while the four-graphs is any set of four letters. They used a Java script to capture data that was typed in an HTML form. They have tested their method on 205 participants achieving FMR of 0.005% and an FNMR of 5%. Based on their good results we can say that continuous verification can be used to authenticate users at least in some web-based applications.

Another article [31] in which the author described software to continuously capture the typing characteristics on an IBM PC to achieve continuous authentication. This article revealed that there are common typed diphthongs[1] which can provide quite an accurate indication of the individual identity.

The authors of [32] suggested that the keystroke dynamics concept should be implemented transparently to do not disrupt the user activity. They managed to perform their experiment with 26 participants, achieving an FMR of 15% and they concluded that it is possible to achieve much better result by creating more representative templates for each user.

There is also an article [33] which deals with two practical systems that have been implemented based upon both static and dynamic verification techniques. The static verifier uses a neural network approach, while the dynamic verifier uses the statistical analysis approach. The first system is tested with 15 participants achieving an FMR of 8% and an FNMR of 7%. The other one is tested over 30 participants achieving an FMR of 15% and an FNMR of 0%.

There is also a sequence of papers, all about continuous authentication. The first paper [5],which is considered as a base for our work, presented a series of results from a preliminary statistical analysis of multi-application keystroke data. The authors used the statistical approach and the neural network approach for the analysis. The results of their research were not as encouraging as they hoped and more investigations were suggested. The second paper [34] was an improvement for the previous one in which they introduced the concept of application-specific analysis. Specifically they created a specific template for each different application. They achieve an acceptance rate of around 60% for instant messengers and word applications compared against a general template disregarding different applications. Still they concluded that further investigations were needed to know the effect of different application on keystroke dynamics authentication.

The third paper [35] presented a series of results from a three month trial in which templates were created using digraphs and trigraphs based on latencies. Those templates are collected with 35 participants based upon a total of 5 millions keystroke samples. The results demonstrated that the technique offered significant promise as a mean of legitimate user verification, achieving an FMR of 4.9% and an FMNR of 0%. The authors improved the previous results by removing the

[1] A unitary vowel that changes quality during its pronunciation, or "glides", with a smooth movement of the tongue from one articulation to another, as in the English words eye, boy and cow

worst 5 participant from the trial results, achieving an FMR of 1.7% for digraphs and an FMR of 4.4% for trigraphs.

One related work can be found in [36] where the authors try to measure the programming performance of a person testing if there is any correlation between the student's key stroking speed and performance while they are programming using Java[37] and Ada [38]. They concluded that there is a negative correlation between speed and performance (completeness of program).

Another article [39] is about a method to authenticate a mobile phone user using keystroke dynamics. Authentication is performed when the user is entering telephone numbers or when he is typing a text message. They used neural network classifiers to achieve continuous authentication. The authors found the same problems reported in other articles too. The performance of this keystroke analysis depends heavily on the user. In this case, there are two groups which are not suitable to use this technique. One of them is the users who do not use their mobiles regularly. While the second group consists of users who do not have a consistent mobile utilization.

4 Choice of methods

Based on the relevant literature that we mentioned in the second and the third chapter of this master thesis we are going to include a description of the methods to apply in order to answer the research questions raised in Section 1.5. This chapter can be divided into two main sections. The first is the experiment by which we can collect data to acquire more knowledge on the research area and have the ability to answer the mentioned questions. The second is the data analysis by which multiple methods can be used to analyze our data collection in order to draw a conclusion about those research questions. In this part it is recommended to have multiple methods to get as good result as possible taking in consideration the accuracy issue.

Our work can be represented in Figure 6. The first stage can be done by a simple experiment in which we construct the bases of the second stage that can be done by extending the time for the same experiment. In the first stage we need to create an application-specific template for each of our participants. An extraction of specific interesting features concerning each application is also needed. In the second stage we have to develop a new version of our program by adding an incorporated filter to collect just those interesting features that we already construct in our first experiment. Finally we need to perform data analysis on this collected data by applying some different approaches to analyze the collected data.

4.1 Experiment

As mentioned above, a substantial part of this project is to design one experiment to perform the purpose of the data collection. We are going to develop a program to allow keystroke data to be collected under a certain environment. We need to implement a mechanism to collect this required data across all applications running within a user's active session continuously. The program needs to store all captured keystroke features including latency, duration, key code and also the application in which they were generated. This program is going to be used by our participants for several days on their own computers. In other words we are going to use a longitudinal study to capture keystroke features, which means that we are capturing feature from group of participants over a period of several days. That gives more realistic data then if we are capturing features in one session. From the collected data, we construct an application-specific template for each participant. Furthermore a careful look on this data is needed to collect special and interesting features for each application to provide a filter of the second stage.

For our second stage we extend the period of time of using our program. This is done by adding an incorporate filter to collect just the features extracted from the first stage and also that were used for the specific-application template creation. A comparison is needed between the template and the testing data for the authentication process. It is necessary to use a distance metric which is a function that outputs a certain number that tells the difference between two samples. Then we are able to do some calculations regarding FMR and FNMR based on a certain and suitable threshold.

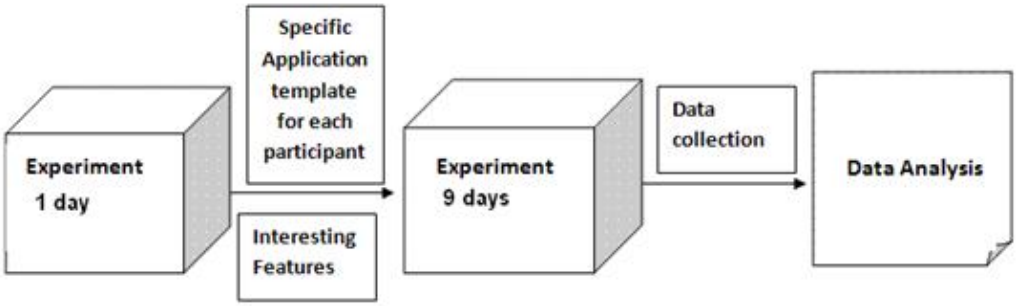

Figure 6: Diagram of the stages in our project.

It is important to mention here that the process of creating an application-specific template for each user can be done using different ways. A variation to choose which features should be used can be easily noticed by looking in Section 6.1.1. Those templates can be based on the latency or duration or even a combination between both of them. They can be also based on a single character or digraphs or even trigraphs or even more common and unique combinations. A much better result can be achieved by choosing the features that show more stability among other features. Still we think that investigating more literature is needed to determine which features our template will based on. Another thing we need after choosing those features is to normalize our data in a certain sample that make it easier to deal with it in the analysis part. The choice between certain descriptive functions such as mean, median and standard deviation is going to be discussed later in this thesis.

4.2 Data analysis

The final stage of our work is to analyze the collected data from our experiment. As we mentioned in section 2.1, two different approaches can be used in performing the analysis. The first is statistical approach and the second is the neural network classification. Sometimes both approaches can be used for example the author of [5] states that the statistical analysis is not enough to get accurate results. He suggests using the data mining approach based on neural network algorithms to achieve much better accuracy. In our thesis we will follow the first approach and let the second for further work.

A major part of our calculation is to score the differences between the templates and our testing data. A choice between different distance metrics is needed. Again as choosing the interesting features of different applications this will be a process of trial and error where we have to test various distance metrics and preprocessing approaches.

5 Experiment

This master thesis specifically considers the continuous user authentication among the usage of different applications. Our main purpose is to answer the question of how typing characteristics differ from one application to another. For this we carried out one experiment to allow keystroke data to be collected from a number of users (participants)[1]. Instead of doing two different experiments, one for the enrollment phase and another for the authentication phase (see section 2.2.2), we did only one experiment. The data collected in this experiment was split into two parts, one for enrollment data and one for authentication data.

The first aim of this chapter is to give a general overview about our experiment including the purpose of doing such an experiment. The second aim is to provide some technical details about how we implemented this experiment. The third one is to provide some information about environment and participants. Finally a pre-analysis section is presented to give an overview of how the preprocessing of the output data of our experiment was performed. The purpose of preprocessing is to ensure that this data is ready to be analyzed to get out the final results which are needed to answer the research questions.

5.1 Experiment overview

The experiment was designed to allow keystroke data to be collected under the Microsoft Windows environment. We followed the same approach of the authors of [5]. In order to collect the required data, it was necessary to implement a mechanism (program) for acquiring keystroke notification across all applications running within the users' active session. Basically the core of the program is a hook function residing in a system DLL to receive keystroke notifications for all currently running applications. Another objective is to determine which application generates those keystroke notifications. The output of this program is a data file that containing lines with this information:

Key Condition ***Key Name*** ***Time*** ***Application Name***

The Key Condition field can have one of two possible values: KeyDown when the key is pressed or KeyUp when the key is released. The Key Name field contains the character name.The Time field contains the value of time counter and finally the Application Name field contains the application in which the keystroke is generated. The entries of the output text file look like:

The basic goal of this experiment is that we want the participants to type on a keyboard as naturally as possible. To achieve this goal our program is going to be executed by each participant on his own personal computer. The program should run in the background and collect the keystroke data over a long time. During this period the participants should use the computer

[1] wherever user is mentioned it also refer to participant in Chapter 5 and Chapter 6

KeyDown	E	1253.68329689947	Microsoft Word
KeyUp	E	1253.78317984548	Microsoft Word
KeyDown	L	1253.81243257646	Microsoft Word
KeyUp	L	1253.92189876578	Microsoft Word

normally. At the end of the period the participant should send the collected data back by email for further processing. A similar experiment setup was used by other authors too [5][40][41], where the program is given to all participants and the participant need to send the output file back for analysis.

5.2 Technical details

The program is build using the C# programming language. We could have created a Java application instead of a C# program with the same functionality. However a Java application has various timing accuracies between different operating systems. Java applications running on Microsoft systems can have 10 ms or worse timing accuracy [4]. According to [42] the timing accuracy in such kind of experiments should be better than 1ms.

As we mentioned in Section 5.1, the idea behind the program is to collect the keystroke notification that the user type . The user should run the program on his personal computer for 10 days. The mission of the user is to run the program during the full day. Because our program collects all the user keystrokes, some legal considerations are taken into account to protect the user privacy and data:

- First of all, our program has a pause/resume button that can be used to temporarily stop the program from collecting data. This button can be used for example when the user types personal information, like passwords or credit card numbers or any important critical data that should be kept private. This button is also needed when another person wants to use the users' personal computer. In such a case the user needs to pause our program to stop capturing the keystrokes of this person since we are only concerned with our users' data.
- Secondly, the output data will be processed and analyzed as much as possible automatically. This means that nobody will go through the data to check what text the user has been typing. That kind of information is not relevant for this thesis. Our purpose is to perform statistical analysis on the keystroke data using single and double key information.
- Finally this data will be anonymous, meaning that it will be stored under a random identity number that is assigned to the user. The information about the link between the user and his/her identity number will not be made public and will only be used during the period of the project. It will be destroyed later to ensure complete privacy.

Figure 7 shows the graphical user interface(GUI) of the program. It is developed to make the life of the user easier. All the user needs to do is to start the program which will open in a window as shown in the figure. As we mentioned, the user can pause the program by clicking on the pause button. For resuming the program the user need to click the same button again. To completely stop the collection of keystrokes the user need to click on the finish button. This button is obligatory to be pressed by the user before turning of the program or shutting down

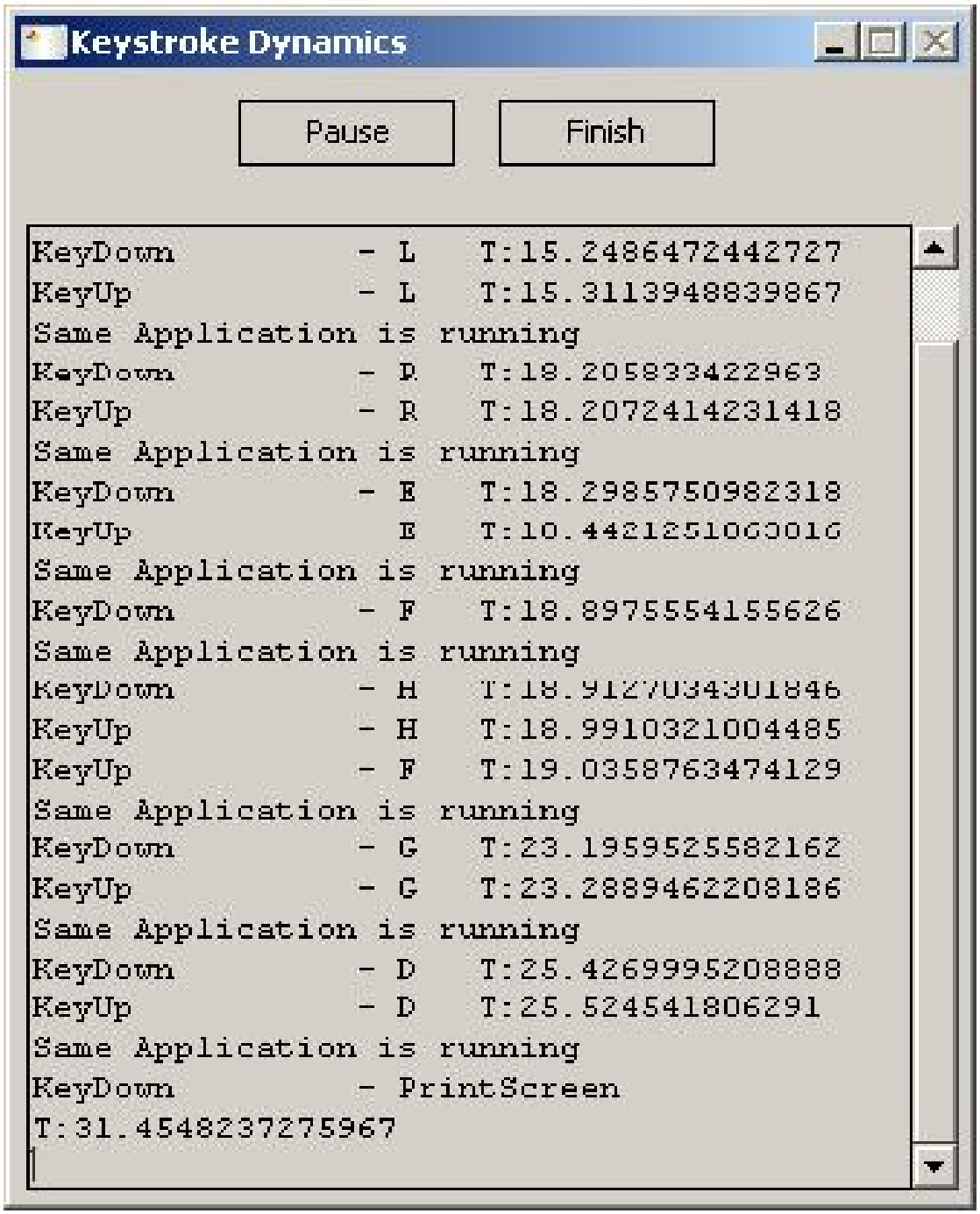

Figure 7: GUI for main program

the system otherwise data might get lost and not saved in the output log file. Furthermore the user can minimize the program and uses the computer normally without bothering about the program. The program will keep collecting data and it will appear in the program tray from where the user can open it again in order to pause, resume, or stop the program.

5.3 participants

Most of the participants of this experiment are students and employees from HIG. The others are in various job positions. There were 35 persons who volunteered to participate in the experiment. Of these, only 25 completed the experiment and returned the data. The participants were given information about the experiment goal and how the program worked. They were also provided with the information about the precautions to protect their data and their privacy. They were asked to run the program for 10 days. Because most of them sometimes forgot to run the program , it took approximately 1 month to collect the data. Some participants collected data in less than 10 days,while others used the program longer than 10 days. The shortest and longest data collection periods were 6 and 15 days respectively. Participant who provided data from less than 6 days were rejected and excluded out from the analysis. Another important thing that the data were also investigated to exclude unuseful part of this data, for example when the participant is using a single key for a long time like VolumeUp and VolumeDown. Finally the size of the data file was considered to accept or reject the participant data. Output files less than 1MB (around 9000 keystrokes) is excluded from the analysis.

The 25 participants consisted of 20 men and 5 women participated. Most of them were from different countries and, given the fact that they are using different languages, it is expected that the keystroke features is highly sensitive for the language variances. More research in this field are needed. The participants were aged between 20 and 55. The majority of participant are between 22 and 30 years old, with a mean of 29.8 and a standard deviation of 9.5.

5.4 Preprocessing

The first target for our analysis is to create a template or a reference profile[2] for each specific user and for each specific application used by the user. In order to achieve that, we need to do some preprocessing on the output file to construct these templates easily. Our purpose in this section is to emphasize the procedure of this preprocessing and focus on the final output of this procedure. Template creation will be emphasized more in Section 6.1.1.

Figure 8 shows the whole procedure of our manipulation for the output file. As we mentioned the program used in the experiment will produce a big file with all pressed keys within the user active session in the whole period (10 days). The next step is to categorize this data according to two fields: first is the day number and second is the day number and application number together. For this purpose we make a program (File Division) to split the data into a multiple of small text files with the same structure of our original output file. A systematic method of naming these files is used to facilitate the usage of them in further analysis steps. A filename will be UserID_DayNumber_ApplicationNumber where each part is represented by a number. For example a

[2]The term reference profile is more attached to keystroke dynamics, however the term template is used more in general biometrics. In this thesis we use the template term.

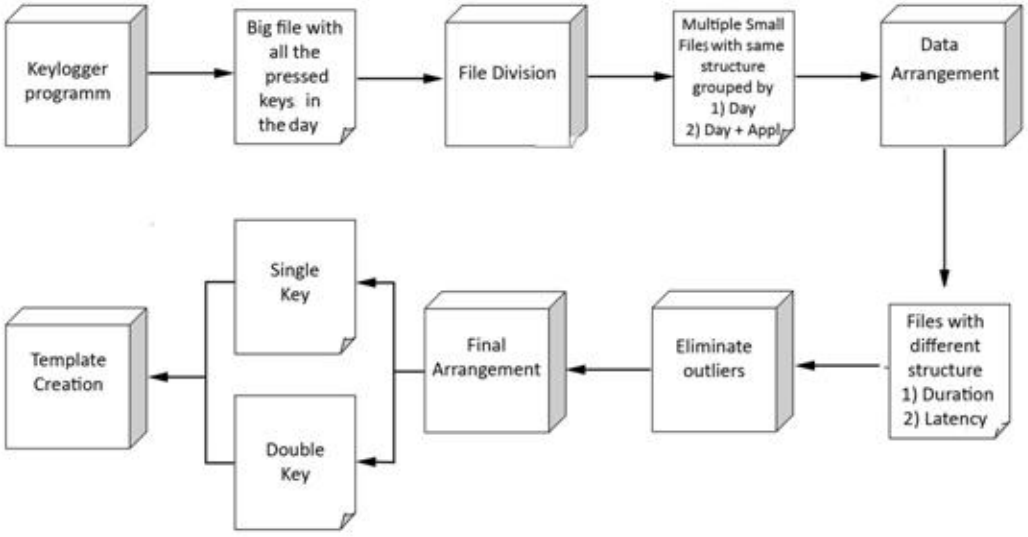

Figure 8: Preprocessing procedure

data file with 1_2_3 as a file name contains the data from user 1 on day 2 related to application number 3. For the first part of categorization (using just the day number) the third part that represents the application number obviously will not be be used. The next step is that we have to manipulate and transform the data to easier format with duration and latency timing. For that we developed a program (called Data Arrangement). After the transformation of the original format, the new structure of the output file will be:

First Key Name ***Second Key Name*** ***Duration*** ***Latency*** ***Application***

Figure 9 shows an example of the data transformation, where the duration is calculated by subtracting the KeyDown timing value from the KeyUp timing value for the same key. Latency is calculated by subtracting the KeyDown timing value from one key from the KeyDown timing value from the next key. Our choice of duration and latency in this transformation is to let both having a positive values. This is normal for the duration values due to the fact that the time when the key is released is greater than when it is pressed. However the latency values can be negative in the case where the second key is pressed before the first key is released (Tab is pressed before the LMenu is released). For that the latency is considered as the difference between the two successive key pressed values (first key and second key).

The next step is to eliminate short/long durations and latencies that may adversely affect

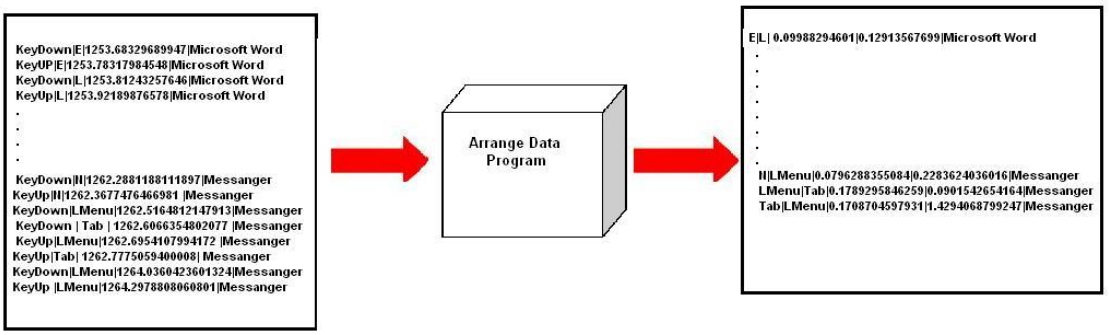

Figure 9: Example of data transformation

the distribution of single key (one character) times or double key times (digraph). The data will improve by removing those values, which are also called outliers. Those outliers can come from inaccuracy or hesitation for example two keys may be accidently struck together thus producing infeasible small latency or one key is pressed for a long time thus producing a large duration. There are several approaches for removing these outliers. One approach is proposed by the authors in [3] and [23] is called shuffling. Shuffling technique combines two entries by choosing the lowest value of them like the lowest duration for a specific key. Another approach that is used by the authors in [5]and [32] is basically using the values that fall inside a nominal range and excluding the values that fall outside this range from the original data. For example the author of [5] chooses a range between 40 ms and 750 ms where the values outside this range are discarded. A third approach is used by the authors in [30] and [40]. This approach is based on statistical values of the mean and standard deviation of the desired values. Basically by removing values further away from the mean than X standard deviations. Mathematically our nominal range is between $\mu - X \cdot \sigma$ and $\mu + X \cdot \sigma$ where X is a small positive value.A recursive calculation for the mean and standard deviation on the new set can be applied. By doing this several times until all remaining values fall within the accepted range.

In this step of the preprocessing, the second approach is applied first on all of the data and a further step is done by applying the third approach for the data that is used to create the templates. The reason behind this is to create more representative templates for the users by applying more constraints (the second and third approach). However, only the second approach is applied for the rest of data to prevent the loss of more data which can be valuable in the analysis.

The final step before choosing our template is to make a final aggregative calculation for the duration and latency. The output of this program will be two files. Data related to the single key (one character) includes the mean of durations, standard deviation of durations and finally the number of occurrence of the single key. The other output is the data considering the double keys which includes 9 different fields of data as following:

- The first key value

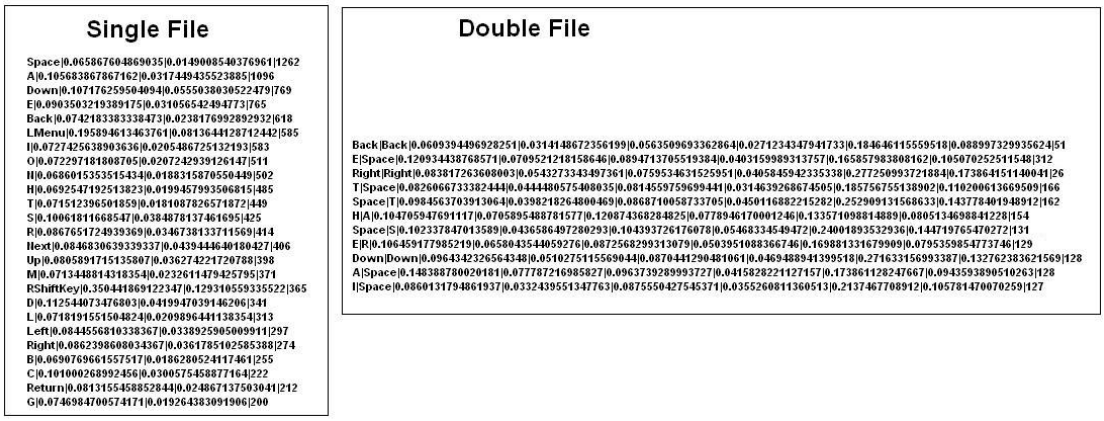

Figure 10: Example of partial data for both single and double file

- The second key value
- Mean of durations of the first Key
- Standard deviation of durations of the first Key
- Mean of latencies of the digraph
- Standard deviation of durations of the second Key
- Mean of latencies of the digraph
- Standard deviation of latencies of the digraph
- Number of occurrences of digraph

Figure 10 shows an example of both single and double file. The data in those files represents a partial set from the complete files. This data was chosen randomly to give an idea of which content the single and double file had.

6 Data analysis

Two main parts which have the same setup structure will be discussed in this chapter. Our approach is to analyze the data regardless of different applications and then making the analysis considering the different applications used by the participants. This setup will assist to achieve a comparison between the results coming from both parts. It also paves the way to answer the mentioned research questions and to draw the final conclusion of this thesis.

6.1 Analysis disregarding different applications

In this part of the analysis we consider the data division categorized by the day number. We pick a random day to generate the template for each participant and use the rest of the data for verification. A complete overview of the analysis will be given in the following sub sections.

6.1.1 Template creation

As we mentioned in Section 5.4, the output of the final arrangement program is divided into two files the single key file and the double key file. Both of them include the whole feature set of each user. Our objective is to concentrate on some interesting features and filter the data to reduce the whole feature set in order to make the next steps of the analysis easier. For this purpose we create a program to generate a template for each user using these specific features. The chosen features are those which are more common in English and also more common between users (meaning they occur more than other features). Those features include the following for the single key: Backspace, Space, E, A, T, I, N, O, S,L, Comma and Period. For the double keys they are: AT, TH, HE, ME, AN, IC, IS, OF, TE, BE, CO, OR and BY. There are two main options to create the templates for each of the users:

- General template.
- Personalized template.

In the general template we are going to include the full list of the above mentioned features with the whole data of average timings for duration and latency and the corresponding standard deviation for each feature. The number of features in this template is going to be the same for each participant. Both the single key file and the double key file are ordered according the occurrences of each feature and appended together. A part of general template can be shown in Table 1 .To see the full template table, see Table 12 in Appendix 1.

The need of a personalized template for each user is to look for those features for which the user has a small variance. These features are the most consistent ones because the user writes them in more or less the same way every time. A reason for this might be that they are used a lot by the user. In this way, each user will have a unique template with only the most consistent features for that user.

Feature	Dur mean 1st Key	Dur std 1st Key	Dur mean 2nd Key	Dur std 2nd Key	Latency mean	latency std	occur.
Space	0.08637580	0.01328496	X	X	x	X	1884
A	0.09000276	0.01615889	X	X	x	X	1451
I	0.07929272	0.01262098	X	X	x	X	1436
E	0.08230904	0.01348837	X	X	x	X	938
S	0.08313311	0.01454619	X	X	x	X	866
T	0.07853734	0.01329261	X	X	x	X	829
N	0.08163786	0.01490012	X	X	x	X	723
O	0.08061462	0.01668728	X	X	x	X	700
Back	0.07828552	0.03453691	X	X	x	X	633
L	0.08149716	0.01228543	X	X	x	X	317
OemPeriod	0.07346880	0.0172407	X	X	x	X	111
Oemcomma	0.08411327	0.011013509	X	X	x	X	98
AN	0.09203296	0.02295964	0.07565121	0.01639196	0.19993104	0.12558609	316
IS	0.07638905	0.01417211	0.07818376	0.01728256	0.18319768	0.12591701	281
ME	0.08338640	0.01349267	0.06650816	0.02535259	0.16544372	0.10750007	217
TE	0.06770484	0.01582099	0.07438959	0.01732062	0.18364483	0.08447577	214
OR	0.07472894	0.01528932	0.07166613	0.02260963	0.21820969	0.14488173	185
AT	0.08829169	0.02116809	0.06393712	0.01608110	0.22723099	0.11699529	180
NG	0.08352435	0.01185858	0.05745269	0.01626420	0.21217961	0.1472827	137
TH	0.07866979	0.01968767	0.07895768	0.01896588	0.27252661	0.16046803	130
BE	0.07752167	0.01643902	0.07890289	0.01921127	0.16378787	0.05837643	120
HE	0.07638126	0.01573692	0.08540163	0.02636182	0.16648760	0.08927590	95
OF	0.06862645	0.01082124	0.06874293	0.02191353	0.32288578	0.17615738	64
CO	0.08559011	0.02349451	0.09827384	0.03577344	0.19233384	0.11316296	53
BY	0.08159148	0.01364979	0.08049215	0.01429820	0.24434897	0.11784635	56
IC	0.07714354	0.01150201	0.07474248	0.01152103	0.48978481	0.14164011	52

Table 1: A part of general template for each user.

To create the personal template for each user, we consider two values for each feature in the general template. The First is the ratio of the standard deviation value and the mean value. The second is the number of occurrences of this feature. For the first one smaller values are better and for the second one larger values are better. Depending on predefined threshold for both values the corresponding feature will be included or excluded from the personal template. For the single file we use a threshold equal to 0.03 for the duration timing. So each feature will be included in the personal template if its corresponding ratio of standard deviation and mean for the duration is less than 0.03. An example for filtering single key features for the personal template is shown in Table 2. For the double file, we use a threshold equal to 0.05 for the duration timing and 0.13 for the latency timing. For both files we use a threshold equal to 50 for the second value (the number of occurrences of the feature), so features that occur less than 50 times are excluded from the personal template. An example of filtering double key features for the personal template is shown in Tables 3 and 4. All the threshold values were chosen by testing many different values and avoiding those values in which most of users will have all the features removed from their templates.

In Table 2 some users show a high consistency in the whole single feature set such as 2, 6, 8, 9, 11, 22 and 24. Other users such as 16 and 25 have only a few features included in their personal template. This happened because their way of typing is not as consistent as other users. This does not mean they are bad or that their results will be worse because they are not so consistent when they are typing, maybe they are easy to differentiate from other users. For example in Tables 3 and 4 some features like AT, BE and BY show a high consistency among different user however some features like TE and ME have a big hole in the consistency and will be excluded from most of the personal templates among different users.

As Shown in Tables 3 and 4 for the double keys the system is almost the same but for the double keys, there are more options. In addition to the duration of the first key, there is the duration of the second key but also the latency between the two keys. For each feature of each user, there are now three different values to check whether the corresponding feature will be included in the personal template of the user or not. Specifically if two of those three values satisfy the threshold value, the feature will be included in the personal template. Table 4 is the continuation of the Table 3 with the remainder of the double key features.

The personalized template is probably the best option since it contains the consistent features for each user. However the performance of both (general and personal) is going to be checked and depending on the results the one which performs best will be selected.

6.1.2 Analysis overview

After the template creation for each user, the analysis is going to be performed using a distance metric (see Section 6.1.3). The full data set,well actually without the data used to create templates, of each user is going to be compared with all previously created templates using this distant metric. After that a decision rule (See Section 6.1.4) is going to be applied to the result of the distance metric and depending on a predefined threshold the decision to accept or reject the user will be taken.

Figure 11 shows the whole procedure after merging the single and double files to create the

user\Features	Back	Space	E	A	T	I	N	O	S	L	Comma	Period
1	X	X	X	X	X	X	X	X	nc	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X
3	nc	nc	X	X	X	X	X	X	X	X	X	nc
4	nc	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	nc	X	X	nc	X
6	X	X	X	X	X	X	X	X	X	X	X	X
7	nc	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X
10	nc	X	X	X	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X	X	X
12	nc	X	X	X	X	X	X	X	X	X	X	X
13	X	X	X	X	X	X	X	X	X	X	X	nc
14	nc	X	nc	X	X	X	X	nc	X	X	X	X
15	X	nc	X	X	X	X	X	X	X	X	X	X
16	nc	nc	nc	X	nc	nc	nc	nc	X	nc	nc	nc
17	nc	X	X	X	X	X	nc	X	X	X	X	nc
18	nc	X	nc	X	nc	X	nc	X	X	X	X	nc
19	nc	X	X	X	X	X	X	X	X	X	X	X
20	X	X	X	X	X	nc	X	X	X	X	X	X
21	nc	X	X	X	X	X	nc	X	X	X	X	nc
22	X	X	X	X	X	X	X	X	X	X	X	X
23	nc	X	X	X	X	X	X	X	X	X	X	nc
24	X	X	X	X	X	X	X	X	X	X	X	X
25	nc	nc	nc	nc	X	X	nc	nc	nc	nc	nc	nc

Table 2: Personal template for single features (X for consistent feature and nc for non consistent feature).

	AT			TH			HE			ME			AN			IC			NG		
User\features	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L
1	X	X	X	X	nc	nc	nc	nc	X	nc	nc	nc	X	nc	nc	nc	X	nc	X	X	nc
2	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
3	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
4	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
5	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
6	X	X	X	X	X	X	nc	X	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
7	X	X	X	X	X	X	nc	X	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
8	X	X	X	X	X	X	nc	X	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
9	X	X	X	X	X	X	nc	X	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	nc
11	X	X	X	X	X	X	X	X	nc	X	X	nc	X	X	X	X	X	X	X	X	nc
12	X	X	X	X	X	X	X	X	X	X	X	nc	X	X	nc	X	X	X	X	X	nc
13	X	X	X	X	X	nc	X	X	X	X	X	nc	X	X	nc	X	X	X	X	X	nc
14	X	X	X	X	X	nc	X	X	X	X	X	nc	X	X	nc	X	X	nc	X	X	nc
15	X	X	X	X	X	nc	X	X	X	X	X	nc	X	X	nc	X	X	nc	X	X	nc
16	X	X	X	X	nc	nc	nc	nc	X	nc	nc	X	nc	nc	nc	X	X	nc	X	X	nc
17	X	X	X	X	nc	nc	nc	nc	X	nc	nc	nc	X	nc	nc	nc	X	nc	X	X	nc
18	X	X	X	X	nc	nc	nc	nc	X	nc	nc	nc	X	nc	nc	nc	X	nc	X	X	nc
19	X	X	X	X	nc	nc	nc	nc	X	nc	nc	nc	X	nc	nc	nc	X	nc	X	X	nc
20	X	X	X	X	nc	nc	nc	nc	X	nc	nc	nc	X	nc	nc	nc	X	nc	X	X	nc
21	X	X	X	X	nc	nc	nc	nc	X	nc	nc	nc	X	nc	nc	nc	X	X	X	X	nc
22	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
23	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
24	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc
25	X	X	X	X	X	X	nc	nc	X	nc	nc	nc	X	nc	X	nc	X	X	X	X	nc

Table 3: Part1 of personal template for double features (X for consistent feature and nc for non consistent feature).

	IS			OF			TE			BE			CO			OR			BY		
User\features	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L	D1	D2	L
1	nc	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
2	X	X	nc	X	X	X	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
3	X	X	nc	X	X	X	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
4	X	X	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
5	X	X	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
6	X	X	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
7	X	X	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
8	X	X	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
9	X	X	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
10	X	X	X	X	X	X	X	X	X	X	X	X	X	nc	X	X	X	X	X	X	X
11	X	X	nc	X	X	X	X	X	X	X	X	X	X	X	X	X	X	nc	X	X	X
12	X	X	nc	X	X	X	X	X	X	X	X	X	X	X	X	X	X	nc	X	X	X
13	X	X	nc	X	X	X	X	X	X	X	X	X	X	X	nc	X	X	nc	X	X	X
14	X	X	nc	X	X	nc	X	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
15	X	X	nc	X	X	nc	X	nc	X	X	X	X	X	X	nc	X	X	nc	X	X	X
16	nc	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	nc	X	nc	nc	X	X	X
17	nc	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	nc	X	X	nc	X	X	X
18	nc	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	nc	X	X	nc	X	X	X
19	X	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	nc	X	X	nc	X	X	X
20	nc	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
21	nc	nc	nc	X	X	nc	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
22	X	X	nc	X	X	X	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
23	X	X	nc	X	X	X	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
24	X	X	nc	X	X	X	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X
25	X	X	nc	X	X	X	nc	nc	X	X	X	X	X	X	X	X	X	nc	X	X	X

Table 4: Part2 of personal template for double features (X for consistent feature and nc for non consistent feature).

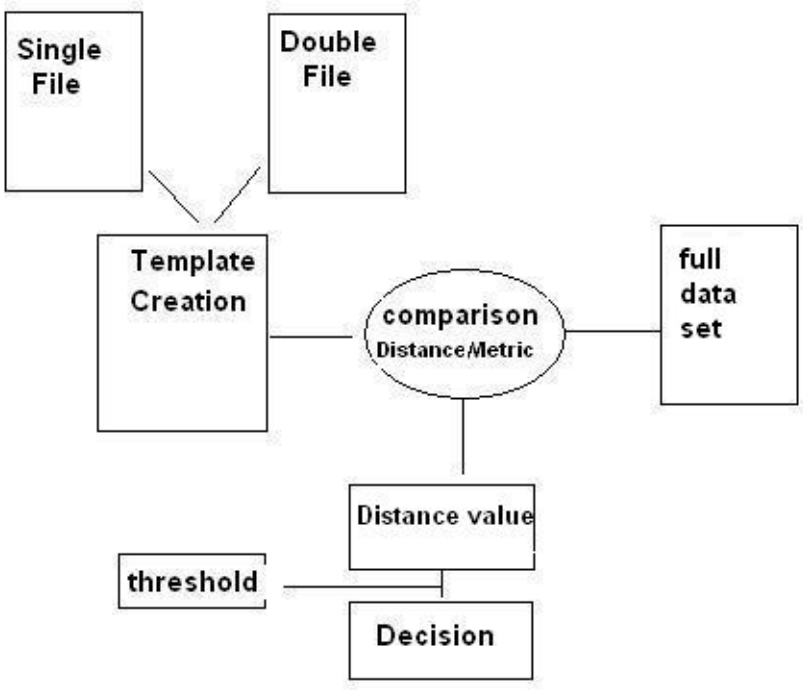

Figure 11: Analysis Procedure

template for each user and then making the comparison against the full data set of each user. The result of this procedure is the distance value on which we will build our decision to accept the user or reject him/her.

6.1.3 Distance metric

To compare the full data set of the typing sessions from the users with a template, a distance metric is going to be used. The distance metric is a function that produces a distance value. This value represents how far away the input from the template is. Many distance metrics can be used in this case. However the matter of choosing the best distance metric is left for as further research. Based on literature we decide to use a simple distance metric [5].

If we are considering a single key K, which has the key duration t_K then the distance will be:

$$D = |\frac{t_K - \mu_K}{\sigma_K}| \tag{6.1}$$

Where μ_K is the mean of the key from the template and σ_K is the standard deviation for that key from the template.In other words the distance is equal to the difference between the input timing and the corresponding mean expressed in units of the standard deviation.

The template also consists of double key data where we have three different timings values to check: the duration of the first key, the duration of the second key and the latency between both keys. we define the distance as the average of the three resultant distances as follows:

$$D_{K1} = |\frac{t_{K1} - \mu_{K1}}{\sigma_{K1}}| \tag{6.2}$$

$$D_{K2} = |\frac{t_{K2} - \mu_{K2}}{\sigma_{K2}}| \tag{6.3}$$

$$D_{L} = |\frac{t_{L} - \mu_{L}}{\sigma_{L}}| \tag{6.4}$$

$$D = |\frac{D_{K1} + D_{K2} + D_{L}}{3}| \tag{6.5}$$

Where K1 is the first key , K2 is the second key, L is the Latency between K1 and K2 and T represents the timing with respect to the key duration and latency. Furthermore μ and σ represent the mean and standard deviation from the template. The total distance is the average of the 3 distance values D_{K1}, D_{K2} and D_{L}.

6.1.4 Decision rule

A decision rule based on a predefined threshold is needed after applying the distance metric described in Section 6.1.3. The resulting distance value from the comparison of each timing value to the corresponding mean and standard deviation from the template has to be compared to a predefined threshold and a decision based on this needs to be taken.

Many decision rules could be applied. In this section we are going to discuss two of them and choose one for the further analysis.

The first decision rule tells us if the typed feature is good (0) or bad (1). When the distance value is larger than a timing threshold, it means that this timing value is outside our limits and this value is judged as a mistake (Rule = 1). Otherwise if it is less than or equal to the threshold then the feature is decided to be correct and it is judged to be a normal typing (Rule = 0). This rule is giving by the following equation.

$$Rule = \begin{cases} 0 & D \leq T, \\ 1 & D > T. \end{cases} \tag{6.6}$$

Where D is the distance value calculated by the distance metric and T is the predefined threshold.

The second decision rule does not only tell us if the typed feature is good (0) or bad (1) but also how far the input from the template is. This decision is more realistic since we have an idea of how bad was the feature typed. This decision is between good (0) and a certain value which express the distance from threshold (mistake). This rule is giving by the following equation

$$Rule = \begin{cases} 0 & D \leq T, \\ D - T & D > T. \end{cases} \tag{6.7}$$

Where D is the distance value value calculated by the distance metric and T is the predefined threshold. In addition it would also be possible to change this rule to have also the idea of how good was the feature typed. Again choosing the best decision rule will be left to further work. The purpose of that is to achieve better results.

D\T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	37	106	46.7	66.7	86.1	92.3	55.3	88.2	165	49.8	95.2	87.5	53.3	33.3	49.1	141	52.4	59.3	72.8	135	43.4	79.4	44.5	60.2	153
2	179	45.3	172	225	48.8	116	125	320	377	113	256	272	141	146	122	364	161	228	77.7	49	192	300	200	154	156
3	68.1	116	30.2	78.2	98.2	127	65.5	118	116	58.6	83.7	98.3	64.1	52.6	73.2	270	43.1	68.6	108	136	48.2	73.1	58.7	69.6	104
4	46.6	115	32.5	31.4	84.9	83.5	41.3	137	104	57.6	46.8	46.1	31.4	28.7	35.8	254	27.3	67.1	85.1	138	42.4	60.9	44	32.3	136
5	250	74.8	205	281	41.1	132	138	425	418	164	294	347	161	212	137	290	191	288	124	69.5	260	346	268	174	76
6	132	46.7	112	140	41.4	39.8	64.1	253	264	58.7	167	173	77.3	96.8	62.4	228	97.5	166	41.4	66	131	198	141	91.5	126
7	60.1	93	34.4	42.1	71.4	77.3	29.8	150	116	36.4	55.5	55.1	26.3	38.1	25.4	174	30	80.9	65.4	117	50.9	70.8	56	28.3	96
8	31.8	98.3	49	85.8	87.7	96.3	66.3	52.3	193	40.9	116	113	68.1	42.3	46.4	286	60.6	52.1	74.4	130	45.2	81.2	43.9	74.6	118
9	84.6	131	43	71	93.6	128	62.1	175	39.3	80	45.9	98.3	45.8	67	61.8	172	33.5	82	119	139	67.9	77.4	74.3	50.8	104
10	66.7	63.6	75.5	119	59.6	85.8	75.4	118	247	41.3	151	148	91.6	52.2	57.9	160	84.6	99.2	59.8	84	80.2	149	80.4	90.4	84.6
11	73.2	121	37.3	64	96	112	54.2	149	77.8	65.4	39.4	84.4	37.6	53	72.3	74	28.3	63.7	94.5	136	53.9	68.9	63.3	54.3	266
12	29	118	21.9	23	89.4	91.3	38.5	99.6	92.3	44.5	38.6	34.3	24.6	18.6	36.1	70	19.3	48.1	85.7	142	25.7	42.5	26.6	30.3	234
13	89.9	92.8	66.4	81.4	60.8	78.5	43.8	203	168	94.4	83	107	38.3	71	44.7	186	48.7	111	81.4	113	84.7	117	89.3	45.3	148
14	58.7	104	59.7	84.9	87.5	100	69.9	130	172	62.7	98.1	107	62.1	38.6	75.7	132	58.5	72.3	80.5	127	59.2	106	63.6	78.7	136
15	60.7	75.1	53.3	74.9	63.6	73.4	46	131	181	33	93.9	94.7	47.4	55.1	34.5	167	50	87.3	55.2	102	65	97.3	68.2	51.8	76
16	162	96.1	158	222	82.6	137	134	282	382	114	241	271	138	138	123	54	154	189	109	109	174	258	183	159	70
17	76.6	98	30.6	64.4	77.9	91.5	38.2	152	111	46.8	57.4	79.6	33.9	53.8	50.2	196	27.4	75.8	82	118	55.2	70.3	65	41.7	148
18	51.6	141	43.5	80.1	117	128	80.8	88.5	129	70.1	78.3	106	64.2	40.5	75.3	166	47.8	35.3	109	168	41.3	68.4	46.3	83.5	106
19	94.1	43	121	129	36.1	58.3	67.2	204	268	61.1	159	158	70.9	81.7	75.3	172	97.6	152	25.1	59.8	119	198	120	91.2	152
20	209	50.1	187	252	43.3	120	137	367	395	140	261	315	152	194	115	86	173	258	97.3	45.4	233	310	240	153	104
21	32.6	160	35.6	70.3	115	117	71.7	69.7	155	76.4	81.5	94	55.9	35.2	63.6	184	39.7	24.9	111	183	30.3	57.1	33.9	65.7	80
22	36.4	143	26.6	54.5	107	125	61.2	82.3	107	65.9	56.9	77.1	47.7	31.7	55	126	31.8	32.4	111	165	26.4	37.3	32.5	54.5	42
23	46.8	102	45.1	60.9	84.9	89.3	51.2	109	142	51.2	83	80.3	48.6	37	43.1	184	44.7	65.1	71.6	130	45.7	80.2	47.9	54.7	64
24	107	82.1	70	96.2	52.9	68.8	40.9	218	156	60.1	92.5	130	45.7	81.2	41.3	165	58.4	126	88	94.4	108	131	112	38.1	68
25	99.1	118	75.6	110	94.5	116	79.7	187	170	98.3	128	144	88.6	78.6	80.2	221	78.1	117	102	134	90.4	142	100	88.6	30

Table 5: Distance values using threshold equal to 0.5 (T = 0.5).

6.1.5 Analysis applying different thresholds

As we mentioned before the second decision rule is more realistic since it is providing more information than the first one. Therefore we decided to use this decision rule with different threshold values (T = 0.5, 1.0 and 1.5) trying to achieve a good result for the authentication phase and to see which threshold is giving the best result of the system in terms of FMR and FNMR. First we use the threshold equal to 0.5 (T= 0.5). Table 5 shows the result of each user when compared against his/her template and against other users templates. The diagonal represents the results when the data of the user is compared to his/her own template while others entries shows the results when the data of the user is compared with template of another user. The best result will be achieved when this diagonal contains the lowest number of each row. With this threshold we achieved FMR of 7.04% when we use the values of diagonal as thresholds i.e a different threshold for each user[1].

The results for T=1.0 are shown in Table 6. Since the threshold is higher, it was expected that more features are accepted to be correct and as a consequence the distance values should decrease. The distance values are lower than before but this does not mean that we will get a better result. When calculating the FMR in the same way, we get 9.76%.

Finally we use T=1.5 to increase the number of accepted features to be correct therefore decreasing the distance values which are shown in Table 6. Still we got a higher FMR of 12%.That means the best result of FMR we got it when we used a threshold of 0.5 (T =0.5) so for further analysis we use this value beside the second decision rule.

6.1.6 Penalty function

To make the analysis more realistic and to implement the continuous authentication of keystroke dynamics in commercial products, we need a kind of continuous checking among the full typing

[1]Note here that we assume that FNMR = 0% and we calculate FMR based on different threshold for each user.

D\T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	17.9	65	23.4	40.3	45.8	53.9	26.7	57	129	23	62.9	57.8	26.3	15.4	21.7	70.4	27.9	32.6	38.7	90.8	22.8	52	23.4	31	142
2	139	22.5	136	188	25	83.1	93.1	277	337	81.2	216	234	106	110	88.5	182	125	183	49.3	23.6	152	256	159	120	145
3	46.4	74.1	13.2	47.4	56.2	84.7	34.2	88.3	82.8	28.2	54.3	65	35.1	31	41.5	135	24.3	46.3	67.7	91.9	30.2	47	38.1	38	93
4	24	73.6	14.8	15.9	40.7	45.2	15.6	95.3	72	32.6	24.9	25.8	14.3	12.6	14	127	11.1	37.8	49.9	92.3	19.2	37	20.5	12.4	125
5	210	45	166	239	18.7	97.7	105	383	376	130	250	305	122	171	104	145	149	242	91.3	40.6	217	301	226	138	65
6	94.3	21.3	79.5	102	17.4	19.2	36.8	213	227	35.3	128	134	47.3	62.4	36.7	114	62.9	123	19.9	34.5	90.6	159	100	58.3	115
7	36.2	55.4	15.8	20.1	30.6	42.4	9.3	111	83.7	14.2	29.5	29.5	8.8	18.9	7.6	87	11.4	49.9	32.7	74.1	25.5	43	30.1	9.1	85
8	17	56	24.3	50.7	48.3	57.8	30.7	29.9	162	17.5	83.8	71.8	32.8	26.5	18	143	35.3	27.9	38.5	84.2	27.3	58	27.4	36.9	107
9	53.3	92.6	18.5	38.2	52.1	87.4	30.2	135	19.3	48.2	23.3	61.9	18.5	34.9	32	86	12.8	51.4	77	97	37.1	44.1	43.7	21.3	93
10	41	32.2	48.8	81.9	29.1	52	41.1	87.8	215	19.2	117	106	55.7	30.7	29.2	80	56.8	66.6	33.1	47.9	51.9	118	53	52.2	73.6
11	44.2	82.1	18.8	35.1	53.1	72.6	24.6	109	48.7	34.4	18.5	53.3	14.7	27.5	41	37	11.4	37.8	56.1	91.9	29.2	40.2	35.8	26.3	255
12	12.4	74.9	7.5	8.3	43.5	49.8	12.1	62.2	61.2	19.1	17.9	14.8	7.7	6.2	12.1	35	6.1	23.5	48.2	95.4	9.2	22.2	10.1	9.8	223
13	62.7	58.2	43.7	52.8	26.4	45.1	20.1	164	132	67	53.8	75.4	17.5	45.5	21.1	93	26.3	78	48.1	73.9	55.5	84	59.7	22	137
14	34.1	64.9	37.5	55.9	47.8	62.3	37.5	94.2	136	32.2	67.3	75.6	34.4	17.8	44	66	35.6	46.6	46.8	85.2	36.5	78.4	39.1	46.3	125
15	34.6	40.4	29	47.9	27.3	42.8	21.6	94.8	150	13.8	65.7	64.9	24.2	31.8	13.1	83.5	26.5	51.1	26.5	59.9	36.1	66.5	39.2	25.6	65
16	126	63.4	124	183	49.8	103	99.6	242	342	81.6	202	231	104	104	90.2	30	119	150	75.9	72.6	137	219	144	123	59
17	50.2	61.5	13.4	35.9	37.9	56.7	15.7	115	79.5	21.3	31.1	48.7	13.1	29.5	27.5	98	9.7	47.2	45.9	75.9	30.3	40.2	38.1	18.8	137
18	24.3	96.1	18.6	46.5	71.6	86	42.7	56.1	92.3	34.2	46	69.7	31.1	16.4	39.6	83	21.7	16.6	70	121	19.5	41.6	21.8	45.9	95
19	60.8	17.9	88.7	94.7	14.1	35.7	39.1	168	227	38.7	120	123	42.5	52.9	44.1	86	62.8	110	10.1	29.7	82.9	160	83.7	56.7	141
20	165	26.8	148	208	20.9	87.3	105	321	354	104	218	271	114	151	82.8	43	132	212	66.5	24.2	188	264	195	118	93
21	12	112	13.6	38.1	68.7	74.5	38.5	39.8	115	39.2	47.6	59.1	27.1	12.8	30.1	92	17.2	9.6	68.9	134	12	29.8	13.3	33.8	69
22	16.7	96.2	7.9	28	60.4	79.4	29.4	49.4	70.9	32.1	28.4	46.6	21	12.2	23.8	63	11.5	14.2	67.7	117	10.8	15.9	14.1	25.8	31
23	24	61.3	21.1	34.3	43.6	52.3	23.3	75.7	108	25.2	52.3	50.2	23.9	16.9	18.6	92.1	21.4	37.5	37.8	85.4	23.1	52.6	24.3	26.4	53
24	77.9	51.5	41.9	57.6	22.8	39.9	15.7	181	117	29.2	57.8	84.7	19.5	52.5	17.3	82.6	27.4	93	57	60.1	76.4	94.2	81.9	11.2	57
25	67.4	79.2	48.6	73.3	57	77.9	48.5	152	135	65.4	92.9	106	56.7	49.2	49.3	111	49.2	87.5	68.1	94.5	60.2	106	69.3	54.4	19

Table 6: Distance values using threshold equal to 1.0 (T = 1.0).

D\T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	8.9	33.7	12.7	24.8	17.5	26.1	11.3	36.3	101	9.3	40.7	38	12.3	8	9.3	35.2	14	14	16.8	54.1	12.7	35.1	12.7	14.3	68.5
2	106	12.2	106	155	13.5	61.4	69.5	238	303	59.7	182	200	78.9	81.6	62.9	91	94.7	94.7	31.7	11.3	116	217	124	92.3	70
3	35.4	40.2	6.6	29.4	23.4	50.5	16.4	68.5	59.7	13	35.4	42.6	19.4	20.8	23.8	67.5	14.5	14.5	36.1	55.1	20.3	32.2	27.2	19.4	44
4	12.7	45.7	7.7	9.3	11.7	19.1	6.4	62.4	50.5	19.2	14.1	15.6	8.2	6.4	5.8	63.5	5.3	5.3	26.2	56.6	9.1	22.4	10.2	5.7	60
5	175	25.5	134	201	8.4	71.7	80.6	345	338	104	210	266	89.8	137	79.2	72.5	115	115	66.9	22	179	258	188	108	30
6	65	8.7	55.1	70.7	6.4	9.5	21.3	179	195	22.2	96.7	101	27.8	37.2	21.8	57	38.3	38.3	9.4	15.1	59.1	126	67.4	36.1	55
7	21.8	30.5	7.4	10	7.5	21.2	2.7	80.2	60.3	4.9	14.9	15	3.1	9.4	2.4	43.5	3.9	3.9	13.5	41.4	11.9	25.4	15.2	2.6	40
8	8.6	22.5	12.4	33.2	17.6	26.6	11.4	18.5	134	4.5	58.2	47.2	14	16.2	6.6	71.5	15.1	15.1	14.1	45.4	15.1	42.3	14.8	15.8	51
9	35.2	62.2	8.7	18.5	20.6	53	10.9	101	9.3	26.9	12.4	35.3	7.2	18.9	12.9	43	4.3	4.3	42.5	63	19.7	24.3	25.2	6.1	44
10	23.3	11.5	29	59	9.2	31.6	19.6	66	188	11.4	91.4	77	33.5	18.3	11.8	40	35.4	35.4	13.6	21.4	30.1	91.6	30.5	31.3	34.3
11	27.5	53.1	10.6	18.1	20.9	41.2	9.2	79.8	28.6	16.6	8.2	32.1	5.4	14.9	20.7	18.5	4.6	4.6	29	56.9	15.9	23.2	20.7	10.4	125
12	5.9	43.6	3.4	3.6	11.5	20.3	3.5	36	40.4	7.5	8.3	6.5	3.4	2.7	4.5	17.5	2.2	2.2	22	57.5	3.5	12	4.1	3.2	109
13	45.2	33.7	29.2	34.5	6.6	23.3	9.5	134	103	51.7	34.6	53.3	8.1	30.3	9.8	46.5	14.4	14.4	25.3	44.5	36	60.2	39.4	11.1	66
14	19.9	35.7	24.9	38	18.7	34.4	19.4	67.3	109	16.1	47.1	53.6	19.6	9	25.4	33	22.2	22.2	23.7	50.3	23.7	58.8	24.3	26.3	60
15	19.2	18.5	15.3	30.4	7	23.4	9.8	67.4	127	6	47	44	13.1	18.9	4.8	41.8	14.3	14.3	10.1	30	19.4	44.7	21.1	12.5	30
16	97.6	40.7	98.1	151	28	77.3	74.9	207	308	59	170	196	79	79.7	67.1	15	91.9	91.9	52.6	46.2	107	187	114	95.5	27
17	33.8	34.3	6.3	18.9	12.9	32.1	5.6	87.3	57.2	9.3	15.8	28.6	5	15.8	15.2	49	3.1	3.1	22.1	43	16	22	21.9	7.8	66
18	10.4	58.7	8.1	25.1	32.6	51.2	18.3	34.2	66.7	12.9	26	43.2	11.6	6.3	18.1	41.5	9	9	40	77.1	9	25.3	9.7	20.8	45
19	37.8	6.5	62.5	67.4	4.8	24.6	22.5	139	194	25.5	88.1	93.5	23.1	33.5	25.6	43	37.3	37.3	4	12.8	54.7	126	55.2	33.5	68
20	126	14.7	116	170	10.6	64.7	80.1	278	318	76	181	231	84.4	115	58.1	21.5	99.3	99.3	43.8	12.7	147	222	154	90.2	44
21	4.4	69.7	4.7	18.2	30.5	40.9	17.8	21.2	84.3	16	24.9	34.3	10.1	4.6	11.9	46	6.7	6.7	35.9	87.8	4.7	14.9	5.2	13.7	32
22	9.2	54.9	1.6	12.7	23	41.5	10.7	28.9	44.2	13.1	12.2	26.2	7.6	5.4	8.7	31.5	3.2	3.2	35.6	71.6	5	6.2	6.8	8.9	13
23	12.2	31.7	10.4	19.3	15.5	26.9	9.6	51	82.8	13.1	32.6	30.9	11.5	8	8.3	46.1	10	10	16.9	48.6	11.5	34.4	11.8	11.3	24
24	55.1	28.2	24.8	39.1	6	18.1	5.3	151	86.3	11.6	31.2	55.5	7.5	40	6.4	41.3	10	10	32.8	33.7	51.1	65.5	56.1	4.5	26
25	48.5	50.2	32.5	51.4	29.3	51.2	28.6	124	108	45.5	67.9	77.7	35.9	33.7	28.3	55.3	31.3	31.3	42.2	62	42.4	79.3	47.8	34	7

Table 7: Distance values using threshold equal to 1.5 (T = 1.5).

session of each user. The shortcoming of previous analysis is that the authentication process is checked at the end of a full typing session and not during the whole session itself. Therefore it is needed to introduce the idea of the Penalty function.

Simply said, a penalty function is a function which updates a penalty value (P) for every captured feature during the typing session. Based on the distance metric and decision rule, the penalty value will increase each time the feature is outside the expected range and decrease each time the feature is considered to be correct and its corresponding time is within the expected range. In other words the user should be punished when he/she makes mistakes (increasing the penalty value) and he/she should be rewarded when he/she types the feature correctly. Furthermore the user also should be punished by relatively small value when the feature does not exist in his/her template. A general setup for the penalty function will be:

$$Penalty(feature) = \begin{cases} 0 & \text{Initial condition,} \\ \text{P+Punishment} & \text{Mistake,} \\ \text{P-Reward} & \text{Correct,} \\ \text{P}+\alpha & \text{Not Exist.} \end{cases} \tag{6.8}$$

P is the penalty value. Punishment and Reward values are determined according to a certain decision rule and α is a small value, in particular we used α =0.01.

For the analysis using the penalty function, we decided to use the threshold which achieved the best result in terms of FMR in Section 6.1.5. The penalty function idea was added to the second decision rule and using that, we set the values of Reward and Punishment as follows:

- $Punishment = |D - T|$ where D is the distance using the distant metric in Section 6.1.3 and T is the threshold value achieving the best results in term of FMR which Equals 0.5.
- $Reward = 1.3$ It was decided to us a value equal to 1.3 which achieved a best results after checking different values from 0 to 2.

Each time the typing of the feature is decided to be a mistake, the penalty function will increase the value P by the difference between the distance and threshold. On the other hand, when the typing of the feature is classified to be correct, the penalty function will decrease the penalty value P by the Reward value. One more constraint on the value P that it should be never be negative i.e $P >= 0$.

When we compare the data coming from a certain user against other user's template, the penalty function is expected to keep increasing most of the time and the user should be rejected after a relatively low numbers of features. Otherwise when we compare the data of a certain user with its own template, the penalty function should fluctuate more or less around a certain threshold value which is related specifically to each user. The value of this threshold depends on how consistent the user types the features.

After applying the penalty function to the authentication phase, three different classifications among user were detected.

- Very consistent users
- Consistent users

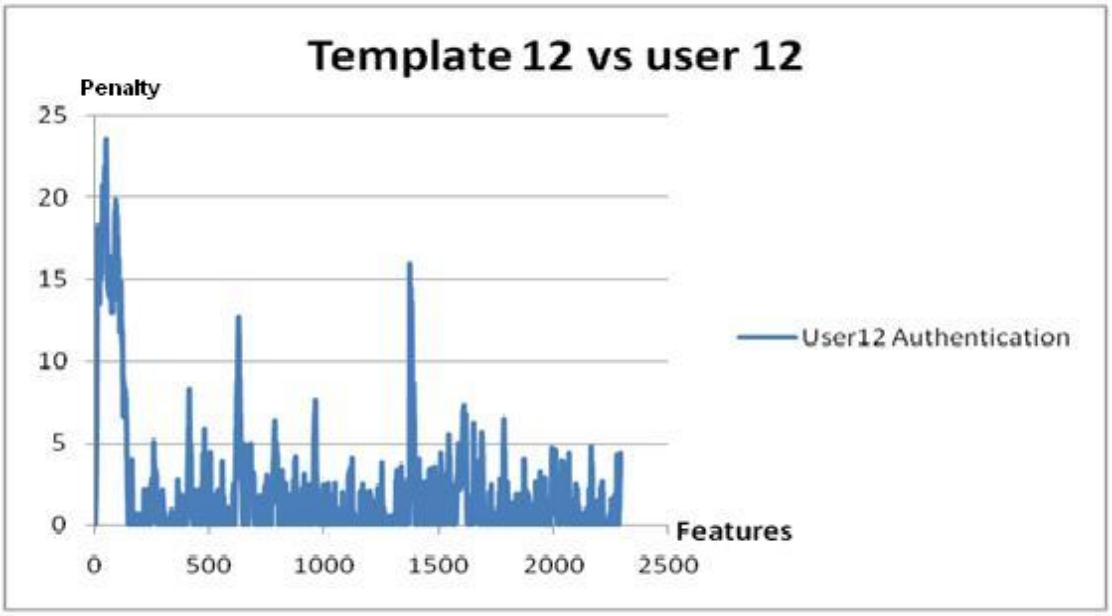

Figure 12: Penalty function for the authentication process for user 12.

- Non consistent users

For each user in the example below, we take the threshold of this user to be the maximum value of the penalty function when we compare this user's data against his/her own template. This value should be set as a boundary where every legitimate user should be always below and all other users should go over after a small number of features.

Figure 12 shows the penalty function of user 12, who has shown a high degree of consistency in his way of typing. That means a low threshold value which in this case is 23.6. Most of the time the value of the penalty function of this user was below 5, of course with some exceptions, but still having a very consistent way of typing.

In figure 13 shows the penalty function when the template of user 12 is compared with the data of user 1 and user 2. The chart on the left shows that user 1 is rejected after typing 108 features, while user 2 is rejected already after typing 46 features. It is clear that this user is far away from the other users' way of typing since it exceed the threshold after a relatively low number of features .

User 9 is an example of consistent user (the second group). Figure 14 shows that the maximum penalty value is now 35. This user typed a lot of features far from the template but also typed a lot of features matching with the template. When other users' templates were compared with user 9, it is clearly still noticeable that this user is far away from their way of typing. For example when we compare the data of user 1 against the template of user 9, then user 1 will be rejected after typing just 42 features while user 2 will be rejected after typing only 11 features which is really a good result. Figure 15 shows the penalty functions of the two comparisons.

The last example is for the non consistent group. Figure 16 shows the penalty function of

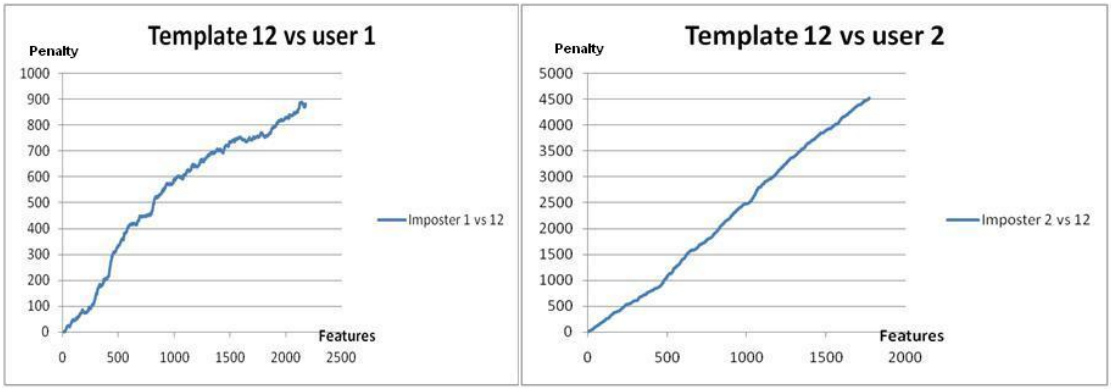

Figure 13: Penalty function for user 1 and user 2 VS user 12 template.

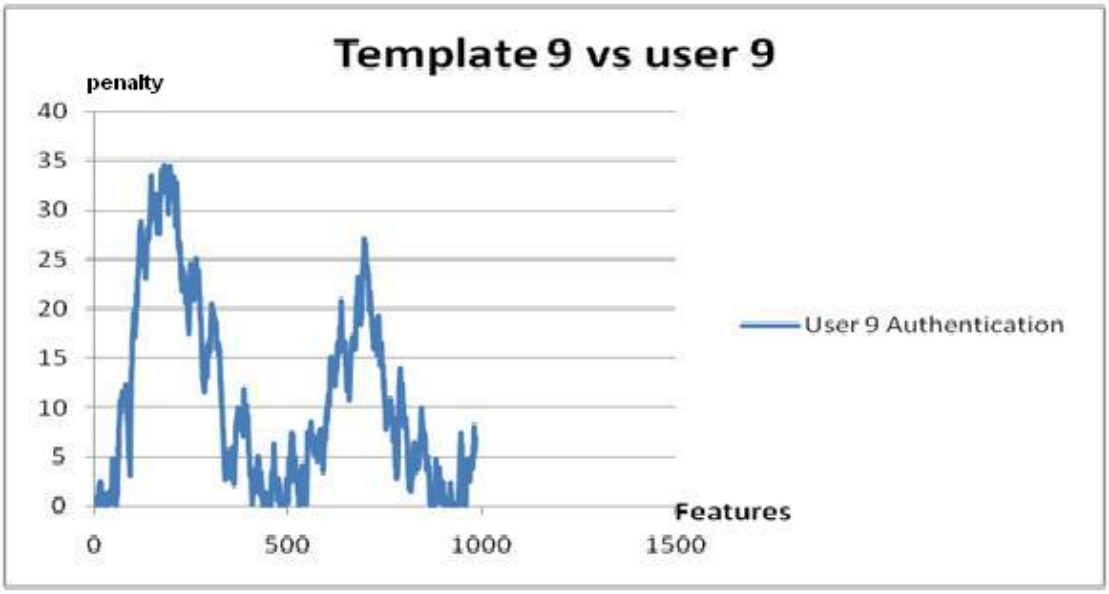

Figure 14: Penalty function for the authentication process for user 9.

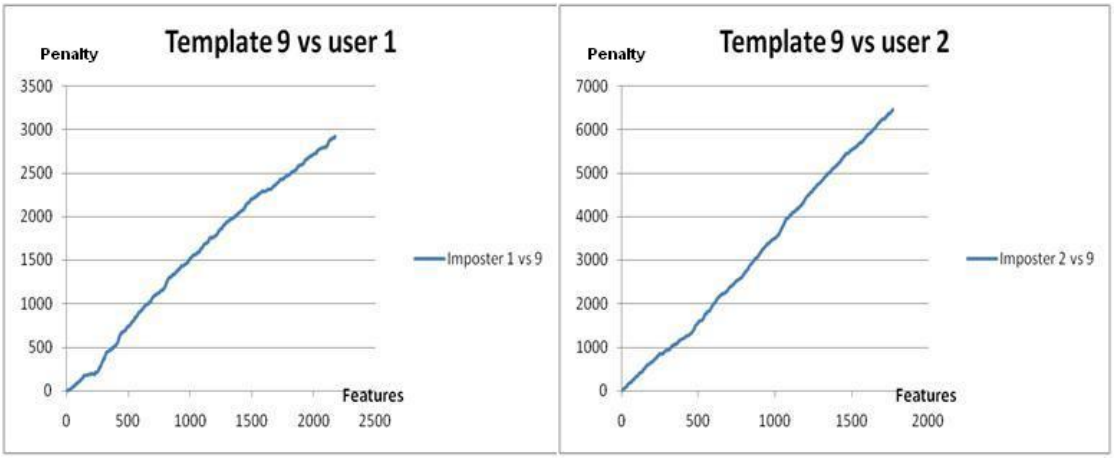

Figure 15: Penalty function for user 1 and user 2 VS user 9 template.

user 8 against his template. It presents a high threshold relative to other users with value of 146. Despite that this user is considered as a non consistent user relative to others, when other user's data were compared with this user template, it is still clearly different. For example when we compared the data of user 1 against the template of user 8, user 1 will be rejected after typing only 65 features while user 2 will be rejected after only 14 features doing the same comparison. Figure 17 shows the two different comparisons.

Table 8 shows the number of features that the imposter can type before him/her getting rejected by the system. Assuming again that all the legitimate users will be accepted (entries in diagonal are zeros). The mean of those entries shows a good potential since the imposter can be rejected in average after 181 features. This number is indicating a good result if the imposter is using only the keyboard to perform his attack. However if the imposter is using the mouse most of the time, the system will be vulnerable.

The mouse authentication is beyond the scope of this master thesis. However further research on combining mouse authentication and keystroke dynamics are needed to solve this problem.

6.2 Analysis considering different applications

In this part of the analysis we consider the data grouped by the application in which the data is generated. We will create specific application templates for each user. Each user will have a number of different templates related to the applications used by him/her. A complete overview about the analysis will be described in the following sub sections.

6.2.1 Applications classification

When we perform the experiment, there was no constraint on the users to use specific applications. As a result we had a number of different applications which seemed to be good at first glance. However the need of choosing common applications among different users is difficult. Another difficulty we faced was that some applications did not have enough data to be included in our analysis, therefore we were forced to exclude them from the analysis. By looking

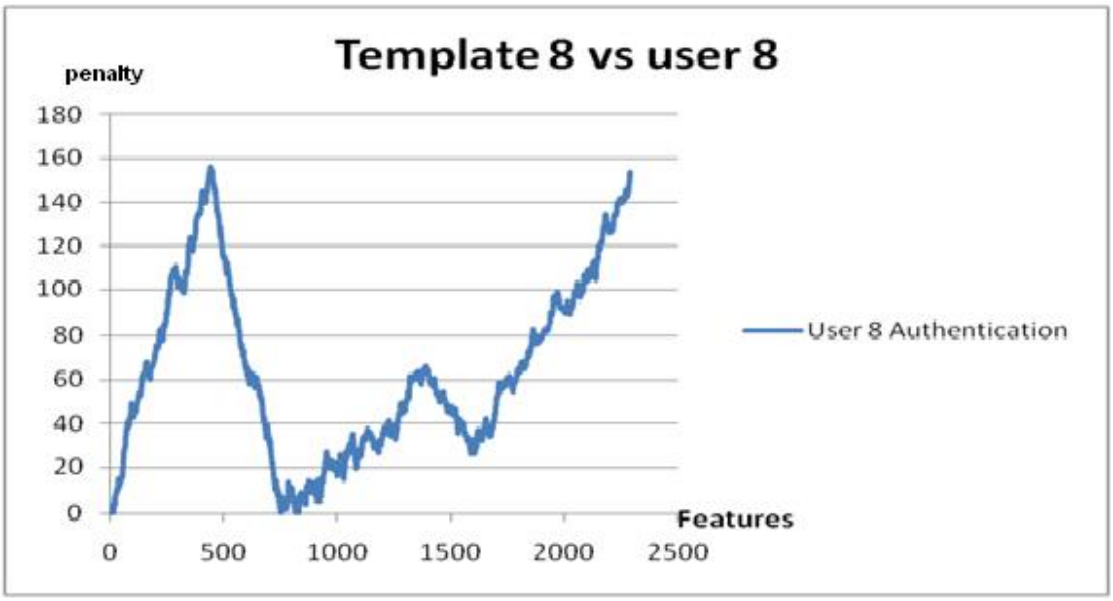

Figure 16: Penalty function for the authentication process for user 8.

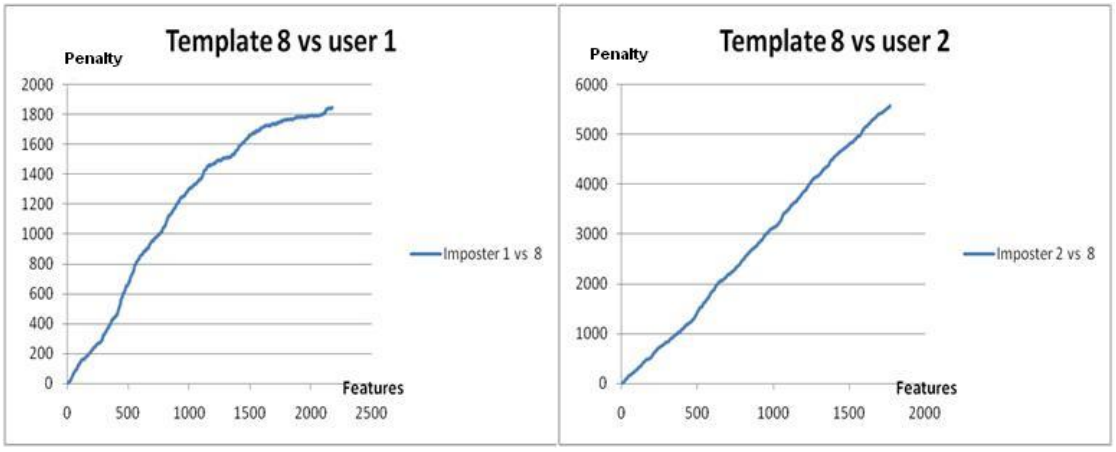

Figure 17: Penalty function for user 1 and user 2 VS user 8 template.

D\T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	128	291	204	158	147	246	65	42	273	143	108	255	408	277	97	260	229	187	101	313	171	306	226	89
2	76	0	79	60	279	117	109	14	11	121	53	46	97	93	111	37	84	60	175	278	71	45	68	88	87
3	200	117	0	174	138	107	208	115	117	232	162	138	212	259	186	50	316	198	126	100	282	186	232	195	131
4	292	118	418	0	160	163	329	99	131	236	291	295	433	474	380	54	498	203	160	99	321	223	309	421	100
5	54	182	66	48	0	103	99	32	33	83	46	39	85	64	99	47	71	47	110	196	52	39	51	78	179
6	103	291	121	97	329	0	212	54	51	232	82	78	176	140	218	60	139	82	329	206	104	69	97	149	108
7	226	146	395	323	190	176	0	90	117	374	245	247	517	357	535	78	453	168	208	116	267	192	243	481	142
8	228	138	178	159	155	141	205	0	70	233	118	121	150	322	293	48	224	261	183	105	150	167	210	100	98
9	161	104	316	192	145	106	219	78	0	170	296	138	297	203	220	79	406	166	114	98	200	176	183	268	131
10	204	214	180	114	228	159	180	116	55	0	90	92	148	261	235	85	161	137	227	162	170	91	169	150	161
11	186	113	365	213	142	122	251	92	175	208	0	161	362	257	188	184	481	214	144	100	252	197	215	250	51
12	469	115	621	591	152	149	353	137	147	306	352	0	553	731	377	194	705	283	159	96	529	320	511	449	58
13	151	147	205	167	224	173	311	67	81	144	164	127	0	192	304	73	279	122	167	120	161	116	152	300	92
14	232	130	228	160	155	136	195	105	79	217	139	127	219	0	180	103	232	188	169	107	230	128	214	173	100
15	224	181	255	182	214	185	296	104	75	412	145	144	287	247	0	81	272	156	246	133	209	140	199	263	179
16	84	142	86	61	165	100	102	48	36	120	56	50	98	98	110	0	89	72	125	125	78	53	75	85	194
17	178	139	444	211	175	149	356	89	123	291	237	171	401	253	271	69	0	179	166	115	246	193	209	326	92
18	264	96	313	170	116	106	168	154	106	194	174	128	212	336	181	82	285	0	125	81	329	199	294	163	128
19	145	316	112	106	377	233	202	67	51	223	86	86	192	166	181	79	139	89	0	227	114	69	114	149	89
20	65	271	73	54	314	113	99	37	34	97	52	43	90	70	118	158	79	53	140	0	58	44	57	89	131
21	417	85	382	193	118	116	190	195	88	178	167	145	243	386	214	74	343	546	122	74	0	238	401	207	170
22	374	95	511	250	127	109	222	165	127	206	239	176	285	429	247	108	428	420	123	82	515	0	418	250	324
23	291	133	302	223	160	152	266	124	96	266	164	169	280	368	316	74	304	209	190	105	298	170	0	249	213
24	127	166	194	141	257	198	333	62	87	226	147	105	298	167	329	82	233	108	155	144	125	104	122	0	200
25	137	116	180	124	144	117	171	73	80	138	106	94	153	173	170	61	174	117	134	101	150	96	136	153	0

Table 8: Number of features which impostors can type before they got rejected assuming all legitimate users are accepted (Zeros at diagonal)

after common applications among different users, we classified applications into three different groups:

- Group X: includes the text editors' applications such as Microsoft Word, Notepad, WordPad, etc.
- Group Y: includes the internet browsers applications such as Mozilla Firefox, IE, Google Chrome, Opera, etc.
- Group Z: includes the instant messages applications such as Yahoo, Msn, Skype, etc.

Most of the users have used those applications. However still some of them do not have enough data to be included in the analysis Table 9 shows whether a user has enough data for the specific group mentioned above or not. The idea behind this grouping is to make the comparison process between users easier.

6.2.2 Template creation

The concept of the personal template mentioned in Section 6.1.1 is applied in this part. The only difference is that we used only the data for each specific application group to create this template. This procedure produced 1,2, or 3 templates for each user. From Table 9 we see that 9 users will have 3 templates, 14 users will have 2 templates and one user (user 19) will have only one template.

6.2.3 Analysis Overview

To have all the possibilities of the comparison process, we decide to make a comparison between the different templates coming from each user (X,Y and Z) with the data coming from each different group of applications. Table 10 shows the 9 possibilities of comparison between the

User/Application Group	Group X Text Editors	Group Y Browsers	Group Z Instant Messages
1	X	X	X
2	X	X	NA
3	X	X	X
4	NA	X	X
5	X	X	X
6	NA	X	X
7	NA	X	X
8	NA	X	X
9	X	X	X
10	X	X	X
11	X	X	X
12	NA	X	X
13	NA	X	X
14	X	X	X
15	NA	X	X
16	NA	X	X
17	NA	X	X
18	NA	X	X
19	NA	X	NA
20	X	X	NA
21	X	X	X
22	NA	X	X
23	NA	X	X
24	NA	X	X
25	X	X	X

Table 9: Available group of application for each user (X for available and NA for not available).

Data\Template	Template X	Template Y	Template Z
Data X	XX	XY	XZ
Data Y	YX	YY	YZ
Data Z	ZX	ZY	ZZ

Table 10: All comparison possibilities between the data and templates.

data and the application specific templates. The first letter express the group in which data is generated and the second letter express the group which the template is generated by.

6.2.4 Penalty function with the second decision rule and Threshold = 0.5

Since we got the best result in the previous part using the second decision rule and threshold equal to 0.5, we decided to apply this to make the required comparison from Section 6.2.3. The results will be viewed in terms of the penalty function with the same setup as used in the previous analysis.

When we compare the users' template coming from the same application as where the data comes from i.e XX, YY and ZZ, we got interesting results. Knowing the application in advance and make a comparison with the template coming from this application gives lower values of the penalty function than when making the same comparison regardless of the application. Figure 18 shows the penalty function of user 1 when we compare the user data which is generated in the same application as the template is created from. The top chart is taken from the first part of analysis regardless the application and the three down charts represent the comparison for application X,Y and Z for user 1. The threshold in the top chart is equal to 250 while the values of threshold of the bottom charts are 120, 45 and 67 (from left to right respectively). This means that knowing the application gives better authentication results.

Returning to the main question of this thesis "are different applications affecting the process of authentication", we need to see the results for the other six possibilities (XY, XZ, YX, YZ, ZX and ZY). In other words we need to compare the data coming from a certain application with the template based on another application which the data come from. Figure 19 shows user 11 penalty function when we compare the data coming from group X to the template generated by group Y. No significant difference is detected, which means that the user is still authenticated successfully despite of the application he used and with also a low threshold which indicates a good result.

Figure 20 shows another example based on the same comparison that is used in the previous example. However this results is for user 10 when we compare the data coming from group X with the template generated by group Y. Despite that we have a higher threshold this time, still there is no significant difference is detected which means that the user is still authenticated successfully despite of the application he used. Based on these results the answer of the previous question is that it doesn't matter which kind of application the user used. Still this user can be authenticated successfully.

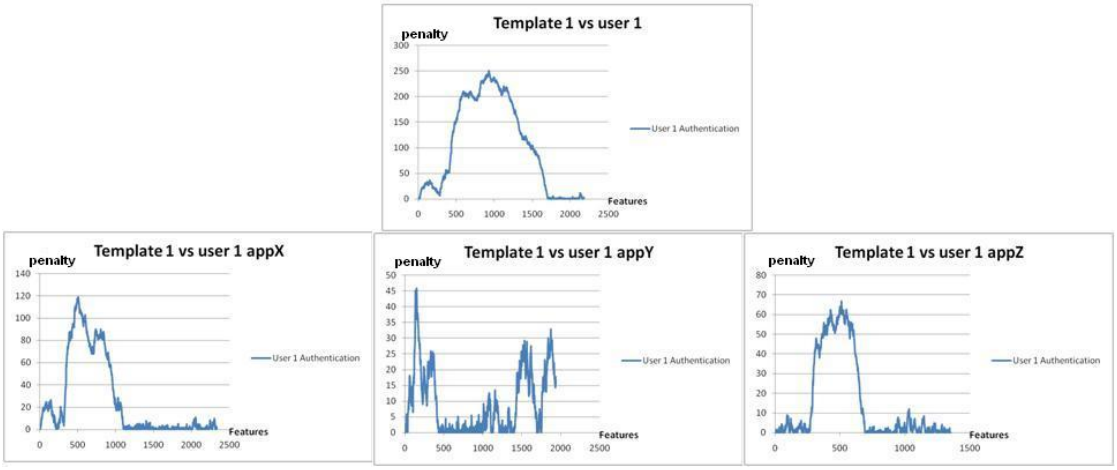

Figure 18: Penalty function for user 1 in different applications.

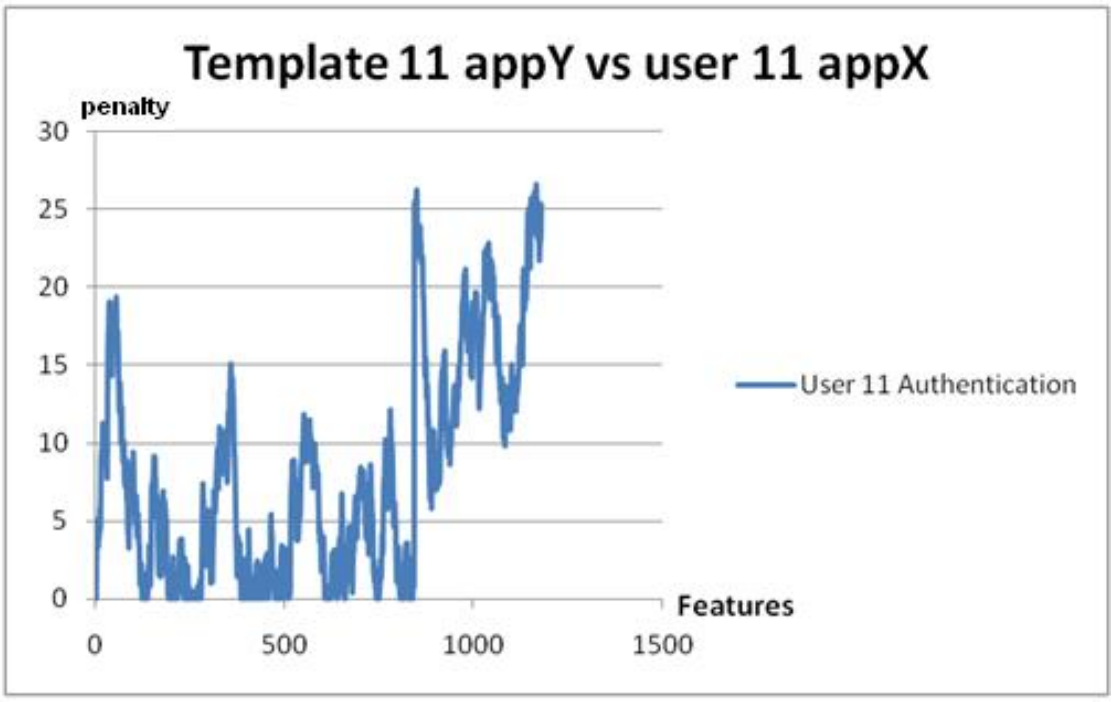

Figure 19: User 11 penalty function when we compare the data coming from group X to the template generated by group Y.

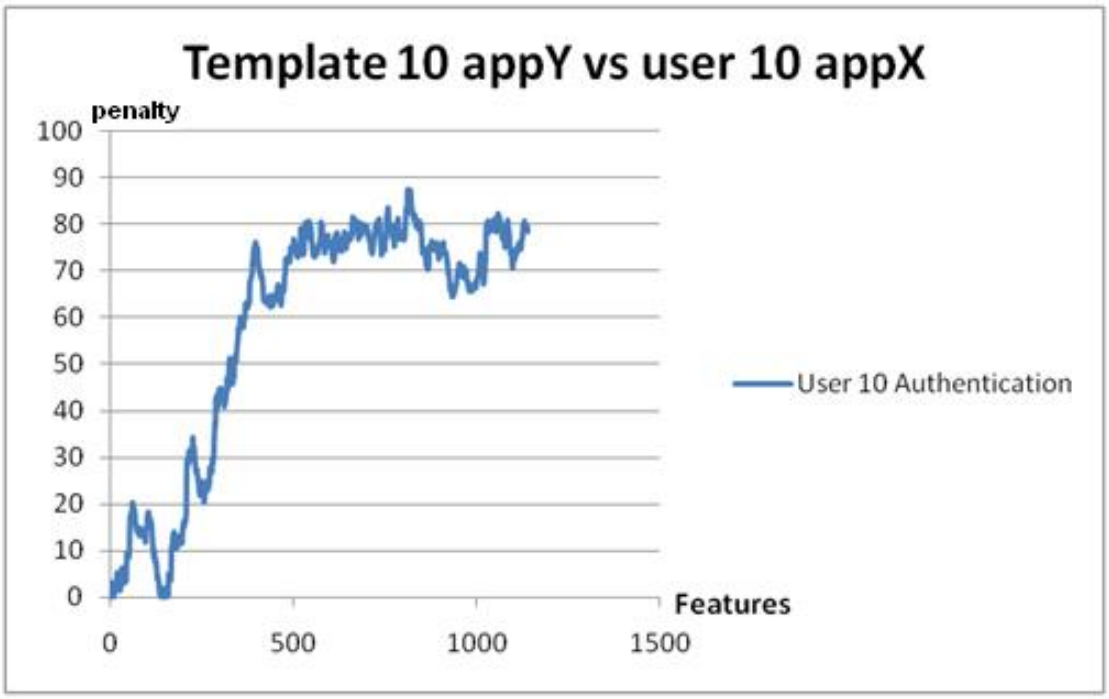

Figure 20: User 10 penalty function when we compare the data coming from group X to the template generated by group Y.

user	disregarding	same	different
1	197	106	193
2	94	54	x
3	174	94	171
4	258	136	286
5	79	46	81
6	146	80	138
7	261	138	285
8	169	95	219
9	186	100	206
10	157	86	160
11	205	109	226
12	348	181	344
13	168	91	181
14	164	90	177
15	200	107	212
16	93	54	95
17	211	113	230
18	183	98	202
19	150	82	x
20	97	56	x
21	221	117	217
22	260	137	278
23	213	113	220
24	171	92	166
25	129	71	127

Table 11: Means of number of features that the imposter can be rejected after typing regardless of applications, using same application, using different applications respectively .

Other results that assure the previous finding are shown in Table 11. A comparison between the average number of features that the imposter can type before getting rejected by the system. The first column represents the average number of each user when he/she tries to authenticate him/herself as another user regardless of which application is used. The second column contains those averages where the template and the test data are generated by the same application and the third column contains those averages where the template and the test data are generated by different applications. column 1 is generated from Figure 8. while columns 2 and 3 are generated from Table 13 and Table 14 respectively.

It is obvious that when the same application is used for both the template and the test data, the impostors will get rejected after a smaller number of features relatively to the other cases (disregarding application and different application). Furthermore columns 1 and 3 are very close together which indicates that different applications do not affect the authentication process.

Since we have a limited number of applications we cannot generalize this result. A solution for this problem is needed for further research and using number of different applications including

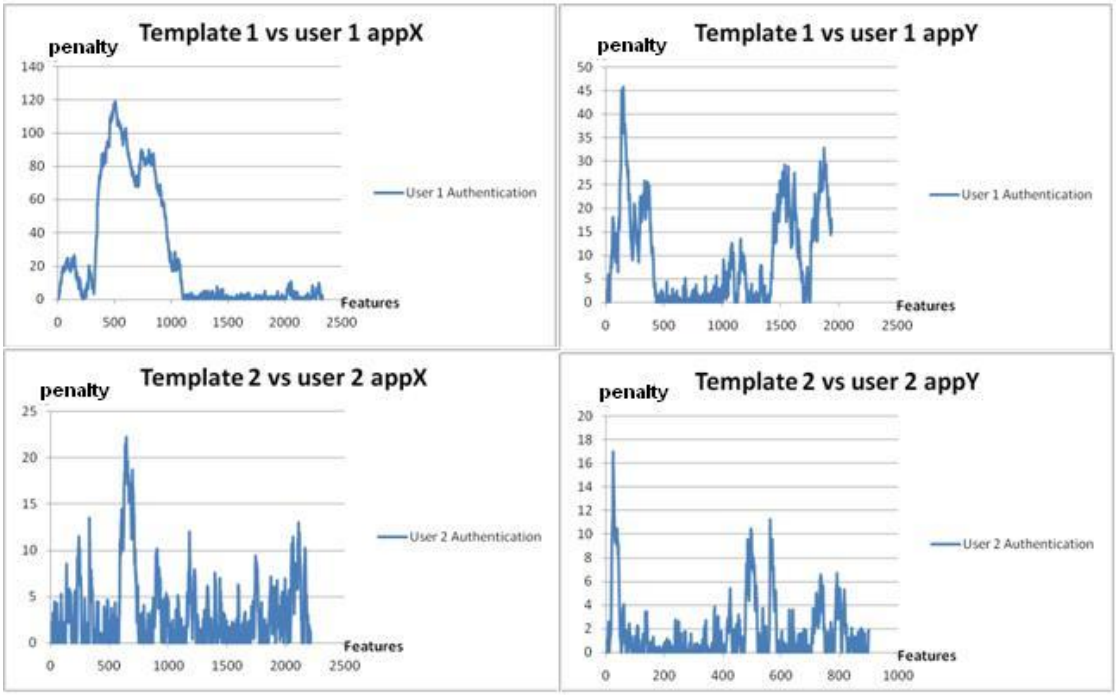

Figure 21: Stability comparison between X and Y.

e.g programming editors.

Another result which could be investigated that application Y shows a higher stability than application X. An explanation of such a result could be that users in general type a short sentence in browsers which also they are used to type such URls, filling short forms with user name and password and others. However it is not the case in the text editors group where the user has long typing sessions. Figure 21 shows an example of this result when make a comparison between X and Y for users 1 and 2. The top charts show that the threshold for user 1 using application X is 120 while for application Y is 46. For the bottom charts the threshold for user 2 using application X is 23 while using application Y,it is 17.

This is also the case when we compared application X with Z (Z shows a higher stability than X). However when we compared application Y with Z, some users show a higher stability in application Y than in Z, while others show more stability in Z than in Y. The results depends on the user's consistency more than the application stability. This can be explained that some application has a template overlapping which prevents to decide which application is more stable than the other. Figure 22 shows an example of this result when make a comparison between applications Y and Z for users 1 and 8. The top charts show that the threshold for user 1 using application Y is 46 while for the application Z is 68. For the bottom charts the threshold for user 8 using application Y is 27 while using application Z is 18.5.

There is a need of further research to include more applications to check the stability of different applications since we have limited number of application groups in this thesis. More

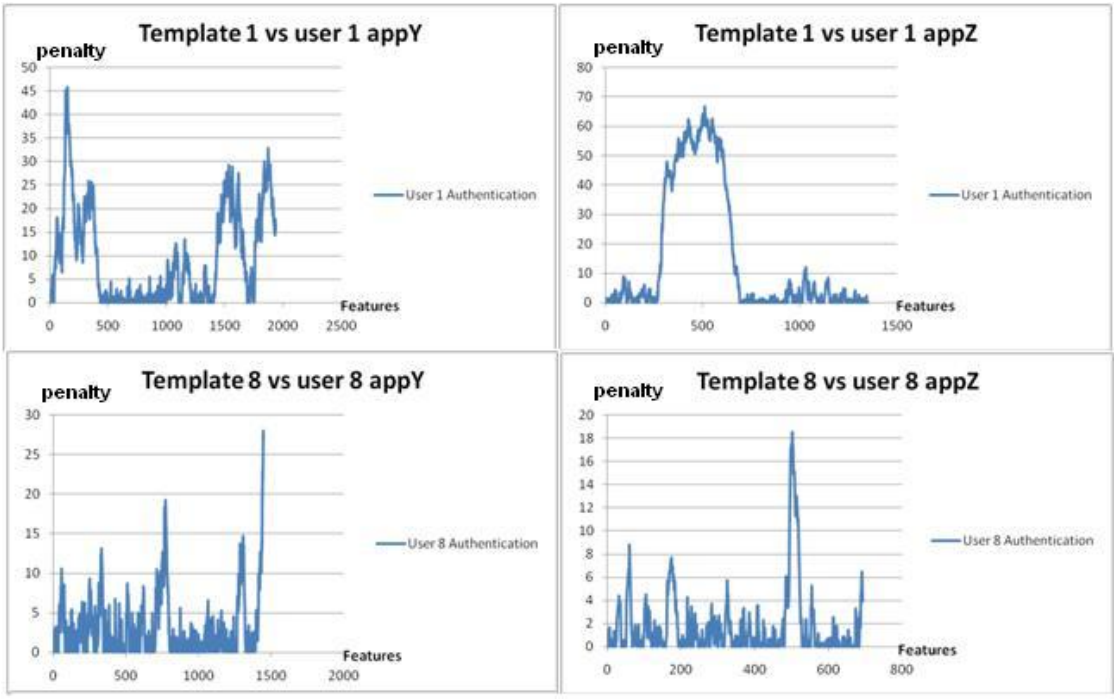

Figure 22: Stability comparison between Y and Z.

discussion about that can be found in Chapter 8.

6.3 Summary of results

6.3.1 Without penalty function and regardless of different applications

Looking at different thresholds we got the following results in terms of FMR using the second realistic decision rule:

Threshold	FMR
0.5	7.04%
1.0	9.76%
1.5	12%

We should always keep in mind that according to these results we assumed that FRR is equal to 0 in all cases. In other words, each user of the system will be accepted when comparing his/her data with his/her own template.

6.3.2 With penalty function and regardless different applications

The use of the penalty function gave us a better idea of how consistent and how differentiable the users are. Three kinds of users were detected: very consistent users, consistent users and non consistent users.

6.3.3 With penalty function and considering different applications

We can summarize the results of the final part of our analysis by the following:

- It seems to be that knowing the application is much better to get better authentication results. Since we have a lower threshold when we compare the data coming from a certain application with the user template based on the same application.
- Despite of the first results, still it doesn't matter which kind of application the user used. And still this user can be authenticated successfully. Generalizing this result cannot be achieved since we have a limited number of applications. In other words we need more applications to be sure that the authentication process is application dependant or not.
- It seems to be that some applications show a higher stability than other applications. However overlapping between some applications prevent to decide which application is more stable than the other. In this case this process will be user dependant and will differ from one user to another.

7 Further work

As we mentioned in Section 6.3.3, the limited number of different applications prevents us from generalizing the first finding about the keystroke dynamics independency on applications variation. This problem could be solved by forcing the participants to use various applications in a nonstandard environment for the experiment. The drawback of this is that a nonstandard environment will affect the realistic part since in our experiment we let the user type freely without any limitation on the applications that are used or any constraint in the experiment environment. Still the research question is open and further investigation is highly needed.

Other approaches then the statistical approach need to be applied, for example the neural network approach where less manual setup for a threshold is not needed. This may give much better results and automate the process of system investigation.

According to the analysis mention in Section 6.1.6, it is obvious that the system is vulnerable if the attacker is used the mouse to perform his attack. So authentication based on mouse movement should be considered to solve this problem.

A .NET platform under windows XP and Vista was used for the core application. However it will be better to use a multi-platform to increase the compatibility with users. This will also increase the potential number of participants.

Different applications is not only the factor that should be investigated with respect to keystroke dynamics. Other factors such as keyboard variations and language variations should be investigated to know the effect of those variables on the authentication based on keystroke dynamics. Again this will increase the potential of keystroke dynamics, such that it might become commercially attractive.

Last but not least, the choice of distance metric and decision rule should also be tested and improved. This will increase the performance of this method to warranty a good authentication with better FMR and FNMR.

8 Conclusion

During the work of the thesis we have looked on the effects of different applications on the authentication process using keystroke dynamics. When we have studied the system regardless of the applications used, the results showed good potential. Using a simple distance metric to express the difference between the template and the test data for each user gives good result in terms of FMR. For different threshold values (0.5 ,1.0,1.5) we got FMR values (7.04%, 9.76%,12%) respectively.

A more realistic environment is provided using a penalty function to authenticate users continuously during the whole session period. Some participants have shown to be not so consistent and this was reflected in our results. Other participants are shown to be very consistent and differentiable. During the second part of analysis we have studied the system considering different groups of applications. The results show that knowing the application gives much better results, since we can use lower thresholds when we compare the data coming from a certain application with the user template based on the same application.

Trying to answer the main question for the master thesis "How typing characteristics differ from one application to another" we found it doesn't matter which kind of application the user used, and still a user can be authenticated successfully. Generalizing this result cannot be achieved, since we have a limited number of applications. In other words we need more applications to be able to conclude whether the authentication process is application dependant or not.

We also found that some applications show a higher stability than other applications. However overlapping between some applications prevents to decide if an application is more stable than the other. In this case this process will be a user dependant and will differ from one user to another.

Bibliography

[1] Gunetti, D. & Picardi, C. 2005. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3), 312–347.

[2] Livia C.F.Araujo, Luiz H.R.Sucupira Jr., M. G. L. L. & Yabu-Uti, J. B. 2005. User authentication through typing biometrics features. *IEEE Transactions on Signal Processing*, 53(2), 851–855.

[3] Bleha, S. & Obaidat, M. 1991. Dimensionality reduction and feature extraction applications in identifying computer users. *IEEE Transactions on systems, Man and cybernetics*, 21(2), 452–456.

[4] Rundhaug, F. E. N. Can attackers learn someone's typing characteristics. Master's thesis, Gjovik University College, 2006.

[5] Dowland, P.S., S. H. & Furnell, S. 2001. A preliminary investigation of user authentication using continuous keystroke analysis. *Proceedings of the IFIP 8th Annual Working Conference on Information Security Management and Small Systems Security*.

[6] Bours, P. Authentication course, IMT 4721, Gjovik University College. Reader for IMT 4721 course, September 2006.

[7] Jain, A., Ross, A., & Prabhakar, S. January 2004. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1), 4–20.

[8] A Brief History of biometrics. http://ctl.ncsc.dni.us/biomet30.05.2009.

[9] CliveReedman, Biometrics - introductory key concepts and review of aspects relevant to accessibility (From presentation slide 9). 'http://www.tiresias.org/phoneability/accessible-biometrics-proceedings/reedman.htm', last visited 30.05.2009.

[10] Ross, A., Shah, J., & Jain, A. K. March 2005. Towards reconstructing fingerprints from minutiae points. *Proceedings of SPIE Conference on Biometric Technology for Human Identification II*, 5779, 68–80.

[11] Jarmo Ilonen,Keystroke dynamics,Lappeenranta University of Technology, Skinnarilankatu 34, 53850 Lappeenranta, Finland . 'http://www.it.lut.fi/kurssit/03-04/010970000/seminars/ilonen.pdf.',last visited 30.05.2009.

[12] Hocquet, S., Ramel, J.-Y., & Cardot, H. 2005. Fusion of methods for keystroke dynamic authentication. *autoid*, 0, 224–229.

[13] Anil K. Jain. Biometric authentication based on keystroke dynamics,http://www.cse.msu.edu/ cse891/sect601/keystrokercg.pdf,last visited 30.05.2009.

[14] Biopassword. 'http://www.admitonesecurity.com', last visited 30.05.2009.

[15] Deepnet Security. 'http://www.deepnetsecurity.com', last visited 30.05.2009.

[16] Monrose, F. & Rubin, A. D. 2000. Keystroke dynamics as a biometric for authentication. *Future Gener. Comput. Syst.*, 16(4), 351–359.

[17] Peacock, A., Ke, X., & Wilkerson, M. September-October 2004. Typing patterns: a key to user identification. *IEEE Security and Privacy Magazine*, 2(5), 40–47.

[18] John A. Robinson, Vicky M. Liang, J. A. M. C. & MacKenzie, C. L. 1998. Computer user verification using login string keystroke dynamics. *IEEE Transactions on systems, Man, and cybernetics*, 28(2), 236–241.

[19] Xiaodong, D., David, S., & Tian, W. X. 2001. Timing analysis of keystrokes and timing attacks on ssh. *10th USENIX Security Symposium*, 1(1), 337–352.

[20] Monrose, F., Reiter, M. K., & Wetzel, S. 1999. Password hardening based on keystroke dynamics. In *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, 73–82, New York, NY, USA. ACM.

[21] de Magalhaes, S., Revett, K., & Santos, H. Aug. 2005. Password secured sites - stepping forward with keystroke dynamics. 6 pp.–.

[22] M. S. Obaidat and B. Sadoun. Keystroke dynamics based authentication ,'http://www.cse.msu.edu/ cse891/sect601/textbook/10.pdf', last visited 30.05.2009.

[23] Saleh Bleha, C. S. & Hussien, B. 1990. Computer-access security systems using keystroke dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, 12(12), 1217–1222.

[24] Lin, D.-T. Jun 1997. Computer-access authentication with neural network based keystroke identity verification. *Neural Networks,1997., International Conference on*, 1, 174–178 vol.1.

[25] Lippman, R. P. 1987. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4–22.

[26] Lammers, A. & Langenfeld, S. 1991. Identity authentication based on keystroke latencies using neural networks. *Consortium for Computing in Small Colleges*.

[27] Bleha, S. A. . & Obaidat, M. S. 1993. Computer users verification using the perceptron algorithm. *IEEE Transactions on systems, Man and cybernetics*, 23(3), 900–902.

[28] Sheng, Y., Phoha, V., & Rovnyak, S. August 2005. A parallel decision tree-based method for user authentication based on keystroke patterns. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 35(4), 826–833.

[29] Wong, F. W. M. H., Supian, A., Ismail, A., Kin, L. W., & Soon, O. C. 2001. Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm. volume 2, 911–915 vol.2.

[30] Monrose, F. & Rubin, A. 1997. Authentication via keystroke dynamics. In *CCS '97: Proceedings of the 4th ACM conference on Computer and communications security*, 48–56, New York, NY, USA. ACM Press.

[31] Shepherd, S. J. 1995. Continuous authentication by analysis keyboard typing characteristics. *European Convention on Security and Detection*, 6(408), 16–18.

[32] Steven M. Furnell, Peter W. Sanders, C. T. S. 1995. The use of keystroke analysis for continuous user identity verification and supervision. *Proceedings of MEDIACOMM 95 - International Conference on Multimedia Communications*, 189–193.

[33] S.M. Furnell, J.P. Morrissey, P. S. & Stockel, C. Applications of keystroke analysis for improved login security and continuous user authentication. *Network Research Group*.

[34] P.S. Dowland, S.M. Furnell, M. P. Keystroke analysis as a method of advanced user authentication and response. *Network Research Group*.

[35] Paul S. Dowland, S. M. F. A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. *Network Research Group*.

[36] Richard C. Thomas, A. K. & Kennedy, G. E. 2005. An investigation into keystroke latency metrics as an indicator of programming performance. *Conferences in Research and Practice in Information Technology*, 42, 127–134.

[37] Gosling, J., Bill Joy, G. S., & Bracha, G. 2005. The java language specification, third edition.

[38] Savitch, W. & Peterson, C. Ada: An introduction to the art and science of programming.

[39] Clarke, N. L. & Furnell, S. M. 2006. Authenticating mobile phone users using keystroke analysis. *Int. J. Inf. Secur.*, 6(1), 1–14.

[40] Joyce, R. & Gupta, G. 1990. Identity authentication based on keystroke latencies. *Commun. ACM*, 33(2), 168–176.

[41] Hocquet, S., Ramel, J.-Y., & Cardot, H. Oct. 2005. Fusion of methods for keystroke dynamic authentication. 224–229.

[42] Lucy Jin, Xian Ke, R. M. M. W. December 2003. Keystroke dynamics: A software-based biometric solution. *6.857Computer and Network Security Massachusetts Institute of Technology*.

A Appendix 1

Feature	Dur mean 1st K	Dur std 1st Key	Dur mean 2nd Key	Dur std 2nd Key	Latency mean	Latency std	Occur.	Dur Ratio 1	Dur Ratio2	LatencyRatio3
Space	0.086375808	0.013284965	x	x	x	x	1884	0.153804232	x	x
A	0.090002769	0.016158898	x	x	x	x	1451	0.179537788	x	x
I	0.079292723	0.01262098	x	x	x	x	1436	0.159169467	x	x
E	0.082309047	0.013488379	x	x	x	x	938	0.163874804	x	x
S	0.083133119	0.014546194	x	x	x	x	866	0.174974717	x	x
T	0.078537347	0.013292618	x	x	x	x	829	0.169252198	x	x
N	0.081637865	0.014900129	x	x	x	x	723	0.182514925	x	x
O	0.080614628	0.016687283	x	x	x	x	700	0.207000681	x	x
Back	0.078285527	0.034536916	x	x	x	x	633	0.441166045	x	x
L	0.081497167	0.012285431	x	x	x	x	317	0.150746724	x	x
OemPeriod	0.073468802	0.017240791	x	x	x	x	111	0.23466819	x	x
Oemcomma	0.08411327	0.011013509	x	x	x	x	98	0.130936641	x	x
AN	0.092032961	0.022959644	0.075651206	0.016391962	0.199931043	0.125586091	316	0.249471967	0.216678134	0.62814703
IS	0.076389047	0.014172106	0.078183763	0.017282557	0.183197678	0.125917009	281	0.185525369	0.221050466	0.68732863
ME	0.083386401	0.013492668	0.066508156	0.025352594	0.165443723	0.107500069	217	0.16180897	0.381195267	0.649768193
TE	0.067704836	0.015820994	0.074389589	0.017320622	0.183644827	0.084475767	214	0.233675987	0.232836633	0.459995351
OR	0.074728939	0.015289321	0.07166613	0.022609627	0.218209692	0.144881732	185	0.204597057	0.315485526	0.663956448
AT	0.088291688	0.021168091	0.063937117	0.016081104	0.227230994	0.11699529	180	0.239751799	0.251514377	0.51487382
NG	0.083524353	0.011858584	0.057452691	0.016264202	0.212179605	0.147282704	137	0.141977556	0.283088611	0.694141663
TH	0.078669791	0.01968767	0.078957683	0.018965882	0.272526612	0.160468035	130	0.250257047	0.240203117	0.588816019
BE	0.07752167	0.016439019	0.078902892	0.019211271	0.163787872	0.058376429	120	0.212057079	0.243479941	0.356414845
HE	0.076381258	0.015736924	0.085401634	0.02636182	0.166487599	0.089275901	95	0.206031223	0.308680505	0.536231536
OF	0.068626454	0.010821238	0.068742929	0.021913526	0.322885782	0.176157379	64	0.157683195	0.318774991	0.545571806
CO	0.085590112	0.02349451	0.098273838	0.035773441	0.192333838	0.113162964	53	0.274500281	0.364017953	0.588367421
BY	0.081591476	0.013649787	0.080492152	0.0142982	0.244348968	0.117846345	56	0.167294278	0.177634711	0.48228706
IC	0.077143536	0.011502013	0.074742475	0.011521031	0.489784806	0.141640113	52	0.149098855	0.154143023	0.289188458

Table 12: Example of a full template table.

B Appendix 2

D\T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	71	153	109	86	81	130	40	28	144	79	61	135	211	146	56	137	122	101	58	164	93	160	120	52
2	45	0	47	37	147	66	62	14	13	68	34	30	56	54	63	26	49	37	95	146	43	30	41	51	51
3	107	66	0	94	76	61	111	65	66	123	88	76	113	137	100	32	165	106	70	57	148	100	123	105	73
4	153	66	216	0	87	89	172	57	73	125	153	155	224	244	197	34	256	109	87	57	168	119	162	218	57
5	34	98	40	31	0	59	57	23	24	49	30	27	50	39	57	31	43	31	62	105	33	27	33	46	97
6	59	153	68	56	172	0	113	34	33	123	48	46	95	77	116	37	77	48	172	110	59	42	56	82	61
7	120	80	205	169	102	95	0	52	66	194	130	131	266	186	275	46	234	91	111	65	141	103	129	248	78
8	121	76	146	87	85	78	110	0	42	90	66	68	66	168	100	31	119	94	99	60	158	91	162	98	65
9	88	59	165	103	80	60	117	46	0	92	155	76	156	109	117	47	210	90	64	56	107	95	99	141	73
10	109	114	97	64	121	87	97	65	35	0	52	53	81	138	125	50	88	76	121	88	92	53	92	82	88
11	100	64	190	114	78	68	133	53	95	111	0	88	188	136	101	99	248	114	79	57	133	106	115	132	33
12	242	65	318	303	83	82	184	76	81	160	183	0	284	373	196	104	360	149	87	55	272	167	263	232	36
13	83	81	110	91	119	94	163	41	48	79	89	71	0	103	159	44	147	68	91	67	88	65	83	157	53
14	123	72	121	87	85	75	105	60	47	116	77	71	117	0	97	59	123	101	92	61	122	71	114	94	57
15	119	98	135	98	114	100	155	59	45	213	80	79	151	131	0	48	143	85	130	74	112	77	107	139	97
16	49	78	50	38	90	57	58	31	25	67	35	32	56	56	62	0	52	43	70	70	46	34	45	50	104
17	96	77	229	113	95	82	185	52	69	153	126	93	208	134	143	42	0	97	90	65	130	104	112	170	53
18	139	55	164	92	65	60	91	84	60	104	94	71	113	175	98	48	150	0	70	48	172	107	154	89	71
19	80	165	63	60	196	124	108	41	33	119	50	50	103	90	98	47	77	52	0	121	64	42	64	82	52
20	40	143	44	34	164	64	57	26	24	56	33	29	52	42	66	86	47	34	77	0	36	29	36	52	73
21	216	50	198	104	66	65	102	105	51	96	91	80	129	200	114	44	179	280	68	44	0	126	208	111	92
22	194	55	263	132	71	62	118	90	71	110	127	95	150	222	131	61	221	217	69	48	265	0	216	132	169
23	153	74	158	119	87	83	140	69	55	140	89	92	147	191	165	44	159	112	102	60	156	92	0	132	114
24	71	90	104	78	136	106	174	38	51	120	81	60	156	91	172	48	124	61	85	79	70	59	68	0	107
25	76	65	97	69	79	66	93	44	47	76	60	54	84	94	92	38	94	66	74	58	82	55	75	84	0

Table 13: Table of results for participants vs templates with a threshold of 0.5 and using the decision rule 2 with our defined penalty function. Each number is the number of features that the imposter need to type to be rejected by the system assuming that the system will accept all legitimate users. The comparison is using the template and test data which coming from the same application specifically YY possibility.

D\T	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	x	289	201	155	145	243	63	50	271	141	105	253	305	250	221	183	153	x	x	227	195	300	199	113
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
3	227	x	0	171	135	105	205	113	115	207	159	100	177	203	152	73	234	221	x	x	234	209	210	200	155
4	319	x	415	0	157	161	327	97	129	233	289	293	331	471	377	77	521	227	x	x	345	247	234	300	123
5	81	x	63	45	0	200	97	29	31	81	43	37	83	61	97	71	95	71	x	x	75	63	75	101	203
6	131	x	119	95	327	0	209	51	49	229	79	75	173	137	215	83	163	105	x	x	127	93	121	173	131
7	253	x	393	321	187	173	0	87	115	371	243	245	515	355	533	101	477	191	x	x	291	215	267	505	165
8	455	x	275	157	153	139	203	0	67	331	115	119	197	319	291	71	247	285	x	x	325	191	333	205	139
9	189	x	313	189	143	103	217	75	0	167	293	135	295	201	217	103	429	189	x	x	223	199	207	291	155
10	231	x	177	111	225	157	177	113	53	0	87	89	145	259	233	109	185	161	x	x	193	115	193	173	185
11	213	x	363	211	139	119	249	89	173	205	0	159	359	255	185	207	505	237	x	x	275	221	239	273	75
12	497	x	519	589	149	147	251	135	145	303	349	0	451	600	375	217	529	307	x	x	450	343	499	300	81
13	179	x	203	165	221	171	309	65	79	141	161	125	0	200	301	97	303	145	x	x	185	139	175	323	115
14	259	x	225	157	153	133	193	103	77	215	137	125	217	0	177	127	255	211	x	x	253	151	237	197	123
15	251	x	253	179	211	183	293	101	73	409	143	141	285	245	0	105	295	179	x	x	233	163	223	287	203
16	111	x	83	59	163	97	99	45	33	117	53	47	95	95	107	0	113	95	x	x	101	77	99	109	217
17	205	x	441	209	173	147	353	87	121	289	235	169	399	251	269	93	0	203	x	x	269	217	233	349	115
18	291	x	311	167	113	103	165	151	103	191	171	125	209	333	179	105	309	0	x	x	353	223	317	187	151
19	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
20	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
21	445	x	317	191	115	113	187	193	85	175	165	143	241	300	111	97	367	369	x	x	0	201	325	231	193
22	401	x	509	247	125	107	219	163	125	203	237	173	283	338	245	131	351	400	x	x	539	0	441	273	347
23	319	x	299	221	157	149	263	121	93	263	161	167	277	365	300	97	227	133	x	x	321	193	0	273	237
24	155	x	191	139	255	195	331	59	85	223	145	103	195	165	227	105	157	131	x	x	149	127	145	0	223
25	165	x	100	121	141	115	169	71	77	135	103	91	151	150	167	85	197	141	x	x	173	119	100	99	0

Table 14: Table of results for participants vs templates with a threshold of 0.5 and using the decision rule 2 with our defined penalty function. Each number is the number of features that the imposter need to type to be rejected by the system assuming that the system will accept all legitimate users. The comparison is using the template and test data which coming from different applications specifically YZ possibility.