Pomona College
Department of Computer Science

# A Study of Keystroke Dynamics as a Practical Form of Authentication

Charles Zhou

May 1, 2008

Submitted as part of the senior exercise for the degree of
Bachelor of Arts in Computer Science

Professor Everett Bull, adviser

## Abstract

This project tested four keystroke analysis strategies and their effectiveness for a practical implementation of keystroke dynamics. A web based keystroke analysis system was implemented and a total of 70 participants were involved with the experiment. We designed a new method of typing data storage that is less storage intensive than previous studies. Error rates for each analysis was graphed out across 1000 thresholds so we could see how sensitive each strategy was to threshold adjustments. This method of graphing error rates is something that we have not seen in any previous studies and is something we believe future research should consider because it is significantly more informative than simply just reporting the best possible False Alarm Rate (FAR) and Impostor Pass Rate (IPR).

The results of the experiment indicated that the ratio method is the strategy with the lowest error rates. It could achieve a False Alarm Rate (FAR) of 4.29%, an Impostor Pass Rate (IPR) of 3.37%, and a classification error rate of 18.57%. We also concluded that the ratio method is the best keystroke analysis strategy to use in a practical implementation because it is the least sensitive to threshold adjustments.

Our method of typing data storage was sufficient though it may have accounted for some of our high error rates. The slightly higher error rates than previous studies is likely due to the fact that typing behavior information is lost when we convert raw timing information into our data structure.

# Contents

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . i
List of Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . v
List of Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . vii

**1 Introduction 1**
1.1 Authentication: The Current Situation . . . . . . . . . . . . . 1
1.2 Keystroke Dynamics . . . . . . . . . . . . . . . . . . . . . . . 2
1.3 Goals . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

**2 The History of Keystroke Dynamics 5**
2.1 1980-1989: The First Studies . . . . . . . . . . . . . . . . . . 5
2.2 1990-1999: Practical Keystroke Authentication and Neural Networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.3 2000-Present: New Ideas and Commercial Products . . . . . . 9

**3 Data Collection 13**
3.1 Collecting Data . . . . . . . . . . . . . . . . . . . . . . . . . . 13
3.2 Preprocessing Data . . . . . . . . . . . . . . . . . . . . . . . . 14

**4 Analysis Methods 17**
4.1 The Standard Deviation Method . . . . . . . . . . . . . . . . . 17
4.2 The Euclidean Distance Method . . . . . . . . . . . . . . . . . 18
4.3 The Ratio Method . . . . . . . . . . . . . . . . . . . . . . . . . 18
4.4 The Vector Disorder Method . . . . . . . . . . . . . . . . . . . 19
4.5 Measure of Success: FAR vs. IPR . . . . . . . . . . . . . . . . 20
4.6 Thresholds . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

**5 Results 23**
5.1 Authentication . . . . . . . . . . . . . . . . . . . . . . . . . . 23
5.2 Classification . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

**6 Discussion of Results** **31**

6.1 Goals . . . . . . 31

6.2 Vector Disorder . . . . . . 32

6.3 Classification . . . . . . 32

6.4 Keystroke Durations . . . . . . 32

6.5 Free Text: Global Patterns . . . . . . 33

**7 Future Work** **35**

7.1 Improved Data Collection . . . . . . 35

7.2 User Acceptance . . . . . . 36

7.3 Dynamic Authentication . . . . . . 36

7.4 Conclusion . . . . . . 36

**A Glossary of Key Terms** **39**

**B Data Collection: Supplemental** **41**

**C Scale of Graphs** **43**

**D Database Statistics** **45**

**E Additional Graphs** **47**

**Bibliography** **51**

# List of Figures

3.1 Schema of the Typing Data Structure . . . . . . . . . . . . . 15

4.1 A example of vector disorder between a reference and claimant for the typing sample "POMONA" . . . . . . . . . . . . . . . 19

4.2 A graph of an analysis method that is sensitive to threshold adjustments . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

4.3 A graph of an analysis method that is *not* sensitive to threshold adjustments . . . . . . . . . . . . . . . . . . . . . . . . 22

5.1 A graph of the standard deviation method vs. the Euclidean distance method. . . . . . . . . . . . . . . . . . . . . . . . . 24

5.2 A graph of the vector disorder method with 0 entries and with 0 entries removed. . . . . . . . . . . . . . . . . . . . . . . . 25

5.3 A graph of standard deviation method vs. the vector disorder method. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

5.4 A graph of the standard deviation method vs. the ratio method. 27

5.5 A graph of the ratio method using only key latencies versus key latencies and key durations . . . . . . . . . . . . . . . . . 28

5.6 A graph of the ratio method on free text authentication. . . . 29

B.1 A screenshot of the web application we implemented . . . . . 41

E.1 A graph of the standard deviation method scaled from 0% to 100% . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

E.2 A graph of the Euclidean distance method scaled from 0 to 338000 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

E.3 A graph of the ratio method scaled from 0% to 100% . . . . . 48

E.4 A graph of the vector disorder method scaled from a normalized disorder of 1.0 to 0.0 . . . . . . . . . . . . . . . . . . . . 49

# List of Tables

5.1 Table of Classification Results . . . . . . . . . . . . . . . . . . . . 29

C.1 Scales for graphs . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

D.1 Database Statistics . . . . . . . . . . . . . . . . . . . . . . . . . . 46

# Chapter 1

# Introduction

The ubiquitous computer systems in our society today have become the guardians of an exponentially increasing collection of information. A significant portion of this data is often sensitive and should only be accessed by authorized individuals. Keystroke dynamics has often been suggested as a method to harden current authentication schemes because it does not need expensive hardware and has been shown to be a feasible biometric system. The goals of this project was to review previous work done on keystroke dynamics, examine the existence of "global" typing patterns, test a method of data representation, and to identify the ideas that would be best suited for a practical implementation of a keystroke authentication system.

## 1.1 Authentication: The Current Situation

Current security solutions have been traditionally categorized into the three factors of something the user *has* (usb token, mobile phone), something the user *knows* (password, social security number), and something the user *is* (biometrics). The most widely implemented of these three has been password authentication systems. Password systems benefit from low implementation costs and effectiveness in terms of accuracy (a password is either correct or wrong). However, passwords are plagued by several inadequacies and numerous effective methods are available for attacking passwords. [Bis03].

None of the three factors are flawless and this has led to the adoption of two-factor authentication systems by groups where security is a top priority. In fact, The U.S federal government has already recognized the weakness of single-factor authentication and recommends that banks adopt some form of two-factor authentication[Cou]. Of the three factors, biometrics offers the most promise yet has been slow in it's adoption. This is due to high costs associated with biometrics and the lack of standardization [Pol97]. Further work needs to be done to lessen the gap between biometrics in theory and biometrics in practice. The results from this project will hopefully aid in this goal.

## 1.2 Keystroke Dynamics

Within in biometrics, there are the two categories of *physiological* (fingerprints) vs. *behavioral* (handwriting) [Pol97]. Keystroke dynamics falls within the category of behavioral biometrics. The idea behind keystroke dynamics is that people have different typing styles and by analyzing the timings of keystrokes, a person can be identified. A benefit of this metric is that measuring keystrokes can be done through a keyboard, thus negating the cost of typical physiological biometric systems which require expensive hardware to measure physical attributes.

## 1.3 Goals

The first goal was to compare the effectiveness of numerous metrics suggested in previous research and to determine which one is the most successful. Each of these metrics was modified to use our data representation. A review and comparison needed to be done because numerous studies claimed high success rates but the studies varied significantly on how they collected the data and how they "preprocessed" the data before analyzing it. Additionally, several of the studies were conducted under conditions that were controlled and therefore their results may not be replicated when applied to a different environment. A more detailed review of past studies can be found in Chapter 2. The experiment in this paper involved collecting typing samples from 70 users.

The second goal was to examine the existence of "global" typing patterns. We tested this by performing tests on free text. The third goal was to test a method of data representation. Previous studies stored each typing sample individually. While this may be OK in an experimental setting, in the real world this would mean that the typing data for a user would constantly be increasing as time goes by. We used our own original method of representing typing samples that reduced keystroke timings into a constant sized reference profile. The last goal was to focus on ideas for a practical authentication mechanism. Our experiment was designed to be more realistic and practical in it's data collection and practicality was also our key criteria in analyzing the different authentication methods. A detailed explanation of our experimental and analysis methods can be found in Chapter 3 and Chapter 4. We hope that the results from this study will contribute to the development of practical **and** effective keystroke dynamics authentication systems.

# Chapter 2

# The History of Keystroke Dynamics

This chapter uses several keywords that are considered common knowledge for most researchers involved in computer security and keystroke dynamics. A glossary of these words is included at the end of this paper in Appendix A if the reader needs clarification on any of the terms used in this chapter.

## 2.1 1980-1989: The First Studies

The pioneering research done in keystroke dynamics dates back to the Rand report in 1980 [GLPS80]. Inspired by the idea that that individuals have unique rhythms when they sent telegraphs, the U.S government funded research to study if this same behavior was exhibited by people using a computer keyboard. It is from this preliminary study that the concept of a digraph was described. A digraph is a pair of two keystrokes and the time elapsed between the typing of the first and second keystroke. The study recorded digraph measurements for the participants and measured: mean, variance, kurtosis, and variance. A simple t-test was used to classify a user. The statistical analysis indicated that keystroke analysis was definitely a feasible biometric. While the authors of the Rand report were able to achieve 100% success rate in classification, many researchers argue that this is insignificant due to fact that only 7 test subjects were involved in the study and that a significant amount of fine tuning of their metric had to be done (over-fitting).

After the Rand report, more experimental studies were conducted that confirmed the relevance of digraphs in identifying user typing signatures. In

1985, Umphress and Williams conducted a more thorough experiment and gave more credence to the idea that keyboard dynamics was viable [UW85]. This was followed by a study in 1988 done by Williams and Leggett and a 1989 study done by Umphress, Williams, and Leggett [LW88] [LWU89]. These studies used statistics to compare a claimant typing sample against a reference profile in order to classify users. For example, one of the methods was to calculate the standard deviation of all the digraphs. When comparing a claimant sample against a reference sample, each claimant digraph was checked to see if it was within .5 standard deviations of the reference digraph. If this was true, then the digraph was considered “valid”. If the claimant sample had more than 60% valid digraphs, the user would be authenticated. The important conclusions from these studies were that: digraphs were confirmed to be a good measure of keystrokes, mean digraph time (essentially typing speed) was determined to be not useful in classification, removing digraphs more than 500 milliseconds seemed to be a good method of removing outliers in typing samples, and using all the digraphs yielded better results than using specific digraphs for classification.

It is around this time that the first patents were granted for keystroke dynamics. Garcia’s 1986 patent described a scheme where users typed their names in order to authenticate [Gar86]. The rationale is that this will be easy to remember and users will hopefully exhibit more consistent digraphs when typing something familiar. An interesting idea proposed in the patent was the use of a vector of mean keystroke latencies (digraphs) as a reference. The Mahalanobis distance function was then used to compare a claimant vector against the reference vector. If the distance calculated is greater 100, the claimant vector is rejected and if the distance was less than 50, the claimant vector was accepted. A distance in between 50 and 100 would prompt the claimant to type the sample again. Garcia’s patent also described a system where users are asked to type 1000 of the most common words 10 times to generate a reference profile and users are then given randomly generated phrases when they want to authenticate. This idea is a clear extension of Garcia’s claim that keystroke dynamics should involve words that are familiar to users. However, it would be impractical to implement.

Three years later, a patent was granted to Young and Hammon for their description of a keystroke authentication method [YH89]. This patent mentions the use of keystroke latencies and keystroke pressures as important measurements of keystroke behavior. The authentication method incorporated the use of a reference vector of digraphs similar to Garcia’s idea. However, Young and Hammon chose Euclidean distance as the measure of similarity between claimant and reference vectors.

Many of the researchers in keystroke dynamics have adopted/adapted the methods described in these two patents. Most of the experiments conducted since 1990 involved storing digraph measurements into vectors and determining a way of measuring “distance” between a claimant and reference vector. The user whose reference vector that had the shortest distance from the claimant vector was the user that was identified as being the claimant.

## 2.2 1990-1999: Practical Keystroke Authentication and Neural Networks

After the first studies concluded that keystroke authentication was feasible, researchers begin designing experiments that would make keystroke dynamics a more practical tool. The two goals were to shorten the amount of typing input needed from the users and to further lower the False Alarm Rate (FAR) and Impostor Pass Rate (IPR) [JG]. Previous studies required users to submit typing samples as large as 537 characters [UW85]. The Rand report study had a FAR of 4% and IPR of 0% and was able to reduce both to 0% with some fine tuning of their metric [GLPS80]. However, when other researchers repeated the methods described in the Rand report, they were only able to achieve a FAR of 30% and a IPR of 17%[UW85]. It was clear that some of the previous research suffered from the problem of over-fitting.

In 1990, Joyce and Gupta set out to address some of the issues mentioned above [JG]. They stated that IPR should ideally be below 1.0% and an FAR of 5% and below was acceptable. Their experiment only used a person’s username, password, and 2 short sentences for sampling a user’s typing style. A FAR of 6.67% and IPR of below 1% was achieved by simply using the Euclidean distance measurement between reference vectors and claimant vectors. This was close to what Joyce and Gupta had hoped to achieve and they believed that while authentication may be hard to implement, keystroke dynamics could easily be implemented as a safety device for detecting intoxicated or tired users.

Brown and Rogers also decided to take a more practical approach in their 1994 study on keystroke authentication [BR94]. Their research was also the first to examine the use of neural networks as a method of classifying claimant vectors. The experiment only used typing samples that were 15 characters long and they were able to achieve FAR between 12.0% and 40.9%. They purposely chose to tune their metrics to have a 0% IPR because they argued that minimizing the number of intruders is far more important than annoying the user with false alarms. For comparison, they also tried using Euclidean distance and came to the conclusion that it did not perform any better than a neural network. Brown and Rogers favored the development of a neural network authentication mechanism because they believed that such an implementation would be trivial.

There are many points that we disagree with Brown and Rogers on. First, having an IPR of 0% is ideal but not practical if it results a FAR of 40.9%. The standards set by Joyce and Gupta were far more realistic [JG]. Additionally, there are several aspects of neural networks that make them less ideal for practical use. In the work described by Brown and Rogers, several different types of neural networks were trained with a set of authentic user typing samples and a large set of impostor typing samples. One neural network was then designed to take in a typing input and output a 0 for rejecting and 1 for accepting the input as valid. A second one was designed to output a number between 0.0 and 1.0, representing the likelihood that the input was valid. All of these designs are impractical because the cost of implementation is substantial. Training the neural network is costly, and so is the creation of a large set of impostor data. The addition of a user would also require additional retraining of the neural net. Finally, these neural nets were only performing classification and were not addressing the harder problem of authentication/identification.

Monrose and Rubin recognized the shortcomings of both neural networks and statistical/mathematical strategies [MR97]. They recommended that to mitigate the cost of constant retraining of neural networks, users can be broken up into smaller groups with one neural net for each group. Mathematical methods which require the storing of numerous reference profiles may suffer from long search times. Monrose and Rubin addressed this issue by clustering user profiles by typing speed. The two key contributions of their study was a) the idea of using keystroke durations as an additional measurement of typing behavior and b) the conclusion that certain people exhibited unique typing behaviors even when typing “free text”. Their overall results were average and they used implementations described in previous work.

Around the same time Monrose and Rubin were conducting their study, Obaidat and Sadoun were also conducting studies on keystroke dynamics [OS97]. Obaidat and Sadoun conducted numerous experiments comparing the effectiveness of neural networks versus mathematical methods. They also looked at the effectiveness of keystroke durations as an identifier. The experiment confirmed that keystroke durations were an useful measure and could potentially be better than keystroke latencies. The best results were when both keystroke durations and keystroke latencies were used. Obaidat and Sadoun also achieved a 0% error rate on one of their neural networks and a FAR of 1.9% and IPR of 0.7% using vector distance measurements. While these results are impressive, we are skeptical because the experiment only tested classification and re-testing of these methods may not yield the same results.

Robinson et al. confirmed the idea that keystroke duration times is a superior measure than keystroke latency times [RLCM98]. Their research also took a practical approach by collecting "real" typing data from several students typing in their login ids. They compared the effectiveness of three different statistical classifiers and the best was able to achieve a FAR of 10% and an IPR of 9%. We find these results to be quite impressive given the small amount of typing data they were working with.

## 2.3 2000-Present: New Ideas and Commercial Products

The most recent research in keystroke dynamics has led to the development of interesting new strategies and we are now seeing commercial products that are using keystroke dynamics as a supplemental form of authentication.

In 2000, researchers began looking into ways to make neural networks more practical. Cho et al. attempted to make neural networks classify users correctly without the need for a large set of impostor typing data [CHHK00]. The need for impostor data was seen as expensive and also unrealistic; Any person attempting to attack the authentication system would most likely not have submitted a sample of the their typing for the neural net to train on. When the neural networks receive typing data from a claimant who's data they have not trained on, the results are unpredictable. Cho et al. devised a neural network strategy that focused on "novelty detection". The neural network is trained on only authentic typing data and when a claimant vector was submitted, it would identify if the data was significantly different than the data it had trained on. Too many"novelties" would result in a rejection of the claimant. The researchers used short passwords for their typing samples and compared the effectiveness of using a statistical classifier (nearest-neighbor algorithm) versus their neural network classifier. The neural network outperformed the statistical classifier with a FAR of 4% and IPR of 0%.

Betchel, Serpen, and Brown in their 2002 research also used a neural network implementation that only relied on typing data from authentic users [BSB02]. While they did not improve over previous rates, their research furthered strengthen the idea that neural nets could be successful with out the need for impostor data. Their paper also recognized that previous neural networks studies which had achieved 100% success rates were flawed because of the large amount of typing samples taken from each user and fine capture resolutions of 0.0001 seconds. These attributes most likely allowed the neural networks to become extremely good at classifying but not as successful for authentication.

Besides new advancements in neural network implementations, new uses for keystroke dynamics were being tested in 2002. Monrose, Reiter, and Wetzel proposed a idea for hardening passwords using keystroke data [MRW02]. While this study does not directly relate to our study, it is an important study that shows how keystroke dynamics can definitely be used practically to aid in authentication.

A third study conducted in 2002 also contributed a new idea for improving keystroke dynamics. Bergadano, Gunetti, and Picardi suggested the idea of using a relative measure instead of an absolute measure [BGP02]. Not only did this study have a significant number of participants (154 volunteers), it also had impressive results of a FAR of 4% and IPR of 0.01%. Their explanation of the metric and rationale of methods convinced us that these results could be replicated. The strategy was to take the reference vector of trigraph timing measurements and then sort them in order of shortest to longest trigraph timing. A claimant vector is also sorted in the same method and then the vector disorder distance is calculated from the reference vector. Authentic claimant vectors consistently had significantly smaller distance measurements than impostor vectors. The rationale behind this method is that by sorting the vectors and computing vector disorder, absolute keystroke timings become less significant and relative keystroke timings become more important. For example, while users may not always type the trigraph “abc” consistently, they are likely to consistently type “abc” faster than certain trigraphs and slower than other trigraphs. Trigraphs were used because the researchers tested both digraphs and trigraphs and trigraphs yielded better results.

Gunetti and Picardi then continued to test this idea of relative measures and it’s effectiveness on free text analysis [GP05]. Free text analysis was considered to be an important area to study because previous research had always had participants submit typing samples in very controlled settings. For their study they used both relative and absolute measures to analyze the typing vectors. Relative measures was determined to be better than absolute measures and using both seemed to achieve the best results. With their 205 volunteers that submitted free text samples of roughly 800 characters, they reported FAR of less than 5% and IPR of less than 0.005%. The paper also carefully outlined the differences between classification, authentication, and identification and described how their metrics could be used to address all three problems. Based on the success of this study, we are convinced that users typing free text also exhibit unique typing styles. This information may be useful in implementing dynamic authentication.

The work done by Bergadano, Gunetti, and Picardi in 2002 and 2005 are the most successful and most well implemented experiments in our opinion. As far as we are aware of, no recent research has improved on their results. However, more recent research has now tried to improve the consistency of user typing. Hwang, Lee, and Cho argue that the quality of the typing samples used to create the reference profiles are more important than the quantity of typing samples [HLC06]. In their study, they tried using artificial pauses and cues in order to improve consistency and therefore improve keystroke authentication overall. We are not convinced that this idea might not be practical despite the low error rates.

Examining the current state of keystroke dynamics research as of the writing of this paper, we see that keystroke dynamics is already viable enough to be used in commercial products as a supplement to traditional password authentication. Biopassword, mentioned by several of the studies we reviewed, is now marketing such a product [CMD$^+$]. There is still a lot of debate on whether or not "global" typing patterns exist for users or if typing patterns only exhibit themselves in controlled situations. The most recent patent granted for keystroke dynamics was in April 2007 and the the authors of the patent argue that global patterns do not exist and that any viable keystroke authentication system must have controlled typing situations [PB07]. Meanwhile, new studies are now examining the feasibility of using keystroke dynamics on mobile phone devices [CF07] [KC07]. Initial results are promising with EER of around 12.8%.

# Chapter 3

# Data Collection

The keystroke logging system was implemented using Adobe Flex$^{®}$ Builder. A flash application was created and hosted on the web to give users a convenient and accessible way of submitting typing samples. Appendix B contains supplemental info on the details of this application.

## 3.1 Collecting Data

Due to the sensitive nature of keystroke logging, participants were first directed to an introductory page where they were told exactly what data was being logged. If they consented they clicked the continue button to launch the flash application. Users were asked to input their name and were then directed to an instructions page where they were given an overview on how the experiment was being run.

Part 1 of the data collection involved collecting user typing data when the text used is predefined. The writing samples that were used in this part of the data collection was a collection of 50 Homer Simpson quotes. A large amount of typing data needed to be recorded in order to create good reference profiles. However, this required users to be typing for a significant amount of time. We hoped to make the experience more bearable by having interesting/humorous writing samples. The typing samples totaled 4000 characters.

The data logged were: key that was pressed, time the key was depressed, and time the key was released. The time recorded was the amount of time passed since the start of the application and the unit of measurement was milliseconds. The data structures used to store this data were two vectors of tuples. One vector was for all key depressions and the other was for key releases. Tuples were in the form of (time,key-code).

Once a user finished typing, all the data that was logged was first inserted into a MySQL database and then an email was sent to notify us that a successful writing sample had just been inserted. If an error had occurred anywhere in this process, the error message would be caught and sent via email.

A second part of the data collection involved inviting a subset of the users back to submit a sample of free text. These users were asked to submit a 350 character typing sample about their favorite movie, food, or book. One of our criticisms of the previous studies in keystroke authentication is that many of the tests were too controlled in the data collection. Ideally a reference typing profile can be created from text A and a different text B will still exhibit the same typing behavior as text A. The point of the free text data collection was so we can test typing from an less controlled situation. The results from the free text analysis would also allow us to examine the possibility of "global" typing patterns.

A total of 70 users submitted complete typing samples and 11 of these users also submitted a free text sample.

## 3.2 Preprocessing Data

After all the data collection was done, we began preprocessing the typing data so it could be used for analysis.

Three separate data structures were created for use. Data structure 1 was the reference profiles, Data structure 2 was the claimant samples, and Data Structure 3 was the free text samples. Each vector stored user typing entries in the data structure that is illustrated in Figure 3.1.

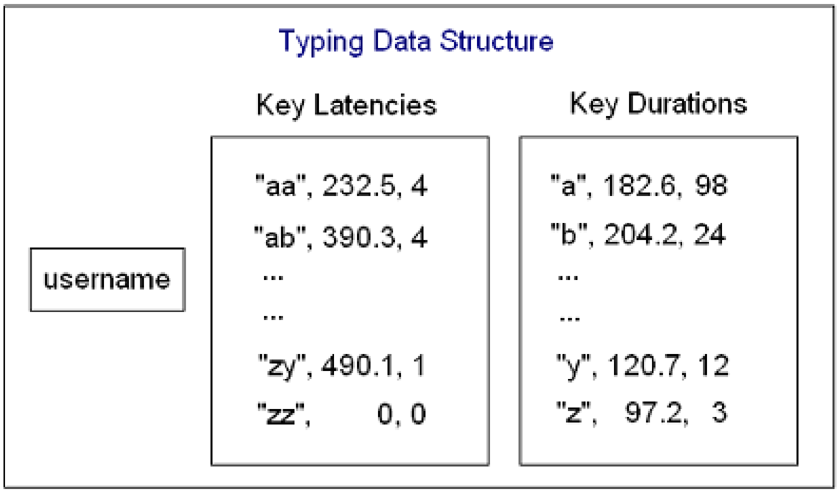


Figure 3.1: Schema of the Typing Data Structure

Looking at Figure 3.1 we see that for every user, we record their username and then have a profile vector of their key latency and key depression timings. The timing data is stored in the form of (X, timing, numberOfX) where X is the digraph or keystroke and timing is the average timing measurement of that keystroke. The third value represents the number of X timing measurements were used to calculate the average. By keeping track of this value, it allows us to easily update the timing measurements when we take in new typing samples and there is no need to keep track of separate typing samples.

For example, if we already have the timing data for the digraph “aa” as (“aa”, 232.5, 4) and we were to take in a new timing measurement for “aa”, we would update by performing the following calculation: $newAverageTiming = (232.5 * 4) + newTiming/(4 + 1)$. The new entry will then be (“aa”, newAverageTiming, 5).

For each user in our database, we divided up their typing data in half. The first 25 quotes (2126 characters) that they typed were used to populate the entries for the reference profile vector. The remaining 25 quotes (1874 characters) were used to populate entries for the claimant samples vector. Finally, the free text vector was populated with the free text typing samples. In populating the timing data, we did some minimal preprocessing to remove potentially invalid data. Similar to previous studies, we set a maximum digraph time so long timing measurements due to pauses in typing are not recorded. The maximum time we set was 500 milliseconds and this was based on the findings of Leggett and Williams[LW88]. Additionally, we set a minimum digraph time of 1 millisecond. This was because we noticed that we occasionally had timing measurements of 0 milliseconds for digraphs such as “cv”. We attribute this as a typo because the letters “c” and “v” are next to each other and the 0 millisecond timing measurement occurs when the user accidentally hits both keys at the same time when they meant to only press one of them.After we populated the vectors and preprocessed the data to remove invalid entries, we were ready to begin testing.

# Chapter 4

# Analysis Methods

Previous experiments concerning keystroke dynamics have found a large number of statistical strategies to be successful. While experiments differ on what statistical tools were used, the essential idea from all of them is the creation of a reference profile for every user and a method of comparing a submitted sample with a profile. The creation of the reference profile vector is described in Chapter 3. In this chapter, we describe the four analysis methods we use to compare claimant samples against the reference profile and the measure of success we used to rate each of the analysis methods. One thing to note is that in our discussion we only mention comparing digraph timings. Key latency comparisons are done the same way as digraph comparisons so we do not mention them in order to reduce redundancy.

## 4.1 The Standard Deviation Method

The standard deviation method we used is adapted from one of the studies by Umphress and Williams [UW85].

This method involves calculating the standard deviation of a person's digraph measurements. On receiving a claimant sample, each digraph from the claimant sample is compared to the corresponding digraph in the reference profile. If the timing measurement from the claimant is within 0.5 standard deviations of the reference timing, then the digraph is considered to be valid.

For classification, the claimant sample with the largest percentage of valid digraphs is the best match for the reference profile. For authentication, the threshold is the percentage of digraphs needed to authenticate. Some studies recommend that 60% is a good threshold that will yield good FAR and IPR [UW85]. The benefit of this method is that each user has their own standard deviation. Consistent typists will have a smaller standard deviation and therefore have a smaller chance of impostors achieving a high valid digraph percentage. The drawback is that it does not look at all of the typing data as a whole. An impostor could have drastically different digraph timings than the reference but still have 60% of the timings close to the reference.

## 4.2 The Euclidean Distance Method

The Euclidean distance method we used is adapted from the studies conducted by Brown and Rogers [BR94]. This method involves calculating the distance between the claimant and reference vectors. The Euclidean distance is the sum of the absolute values of the difference between every claimant digraph timing and every the corresponding reference digraph timing.

$$\sum_{n=0}^{numberOfDigraphs} |claimaintDigraph_n - referenceDigraph_n|$$

For classification, the claimant sample with the shortest Euclidean distance is the best match for the reference profile. For authentication, the threshold is the minimal distance a claimant sample is away from the reference in order to authenticate. The benefit of this method is that it takes in to account every digraph measurement. The drawback is that the same threshold is used for every user, regardless of their typing consistency.

## 4.3 The Ratio Method

The ratio method we used is adapted from a study by Gunetti and Picardi [GP05]. This method closely resembles the standard deviation method in that it attempts to determine the "validity" of a claimant digraph. Given a claimant digraph and a reference digraph, the test for validity is to check:

$$\frac{max(claimantTiming, referenceTiming)}{min(claimaintTiming, referenceTiming)} <= 1.25$$

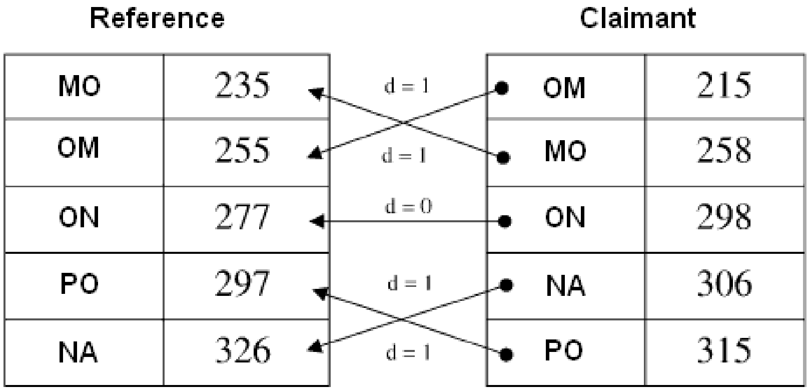
For classification, the claimant sample with the largest percentage of valid digraphs is the best match for the reference profile. For authentication, the threshold is the percentage of digraphs needed to authenticate. Similar to using the standard deviation method, the ratio method takes into account individual typing abilities. However, the drawback is that it does not analyze the typing data as a whole. An interesting note is that Gunetti and Picardi recommend the ratio method over the standard deviation method because they claim that calculating one standard deviation for a person's typing speed in general is not as significant as calculating the individual standard deviations for each digraph. However, in order to calculate a standard deviation for each digraph, each digraph needs to have multiple entries. The ratio method can be used on digraphs with only one measurement, thus taking advantage of as much data as possible.

## 4.4 The Vector Disorder Method

The vector disorder method we used is adapted from studies by Bergadano, Gunetti, and Picardi [BGP02][GP05].

This method claims to be the best out of all previous research. It introduces the novel idea of a relative measure vs. an absolute measure. The previous 3 metrics are absolute measures because they depend on the comparison of absolute timing measurements. Using the vector disorder method, we sort the claimant and reference profiles by the digraph timings. Once sorted, we calculate the vector disorder using the the following:

$$\sum_{n=0}^{numberOfDigraphs} |indexOf(claimaintDigraph_n) - indexOf(referenceDigraph_n)|$$



Reference

| | |
|---|---|
| MO | 235 |
| OM | 255 |
| ON | 277 |
| PO | 297 |
| NA | 326 |

Claimant

| | |
|---|---|
| OM | 215 |
| MO | 258 |
| ON | 298 |
| NA | 306 |
| PO | 315 |

Figure 4.1: A example of vector disorder between a reference and claimant for the typing sample "POMONA"

Figure 4.1 gives an visual example of how vector disorder is calculated. The benefit of this metric is that it is extremely robust against variance in typing. The argument being that while your absolute timings might change from time to time and when you switch keyboards, your relative timings remain consistent. You will ideally consistently type “th” faster than say “vz” or some other sort of pattern. The drawback of this metric is that by simply relying on relative measurements, you have a situation where an impostor typing twice as slowly as the reference profile still authenticate if the impostor mimics the relative timing correctly.

## 4.5 Measure of Success: FAR vs. IPR

The measure of success of keystroke dynamics has been essentially the same for every study. Every authentication system is tested by creating the reference profiles and then testing every users samples against every other user’s reference to simulate an impostor attack. Additionally, samples not used in the creation of an user’s reference profile is tested against the reference to simulate a valid login.

Each of these tests measure how likely the authentication system will reject a result when the person is actually the person who he/she claims to be and how likely it is to accept a result when the person is an impostor. Different studies have referred to these two errors as: Type I vs. Type II errors, False Acceptance Rate vs. False Reject Rate, and False Alarm Rate vs. Impostor Pass Rate. For this study, we refer to these errors as False Alarm Rate (FAR) and Impostor Pass Rate (IPR) since these are the terms used in the most current papers.

## 4.6 Thresholds

FAR and IPR values can be adjusted easily by changing the threshold for accepting and rejecting users. Different situations require different emphasis on FAR or IPR. For our experiment, we decided to graph out the FAR and IPR for each of the analysis methods across 1000 thresholds. These graphs are informative because they give a sense of how sensitive an analysis method is to threshold adjustment.

We will use the standard deviation method to illustrate our graphing method in more detail. Using the standard deviation method, the threshold of authentication is the percentage of valid digraphs needed to authenticate. To graph across 1000 thresholds, we would need to determine the min and

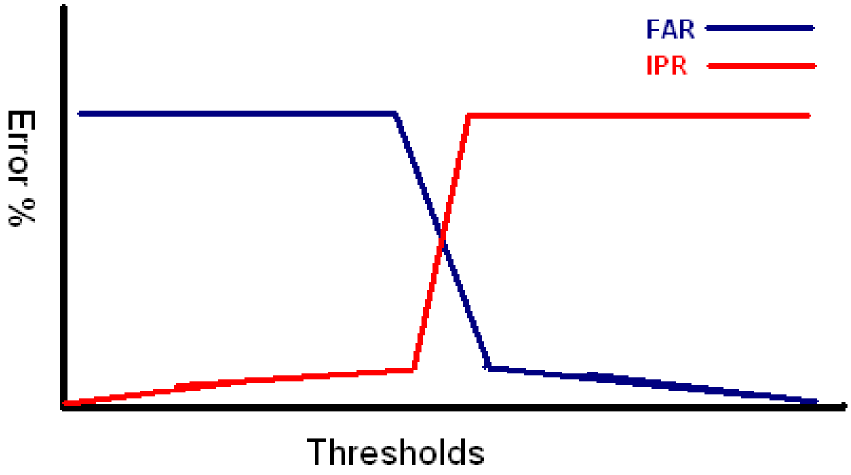


Figure 4.2: A graph of an analysis method that is sensitive to threshold adjustments

max threshold which would be 0% and 100% in this case. We then divide up the range between the min and max threshold into 1000 to get a threshold increment (0.1%). Then we calculate the FAR and IPR values for the standard deviation method starting with a threshold of 0% and incrementing the threshold 0.1% each time until we reach 100%. Next, plotting the FAR and IPR values will give us a graph that will resemble Figure 4.2 or Figure 4.3.

Ideally we want our graphs to look like Figure 4.3. Such a graph would indicate that that the error rate drops quickly and that both IPR and FAR stay low near the EER regardless of threshold changes. A bad analysis method would have a graph similar to Figure 4.2. This graph shows that slight changes in the threshold around the EER drastically affect the error rates. Such an analysis method would not be effective in a practical implementation.

The argument for why such a method would not be practical is as follows. If in our experiment, we determine that using 60% as the threshold for standard deviation gets us FAR and IPR of 0%. However we see that at the 59% and 61% thresholds, FAR and IPR jump up to 70%. What this indicates is that the distinction between an impostor and a valid user is incredibly small and given the variance of typing in real life, there is essentially no distinction between an impostor and a valid user. Our 0% would simply be due to the fact that we found the “magic number” threshold for the

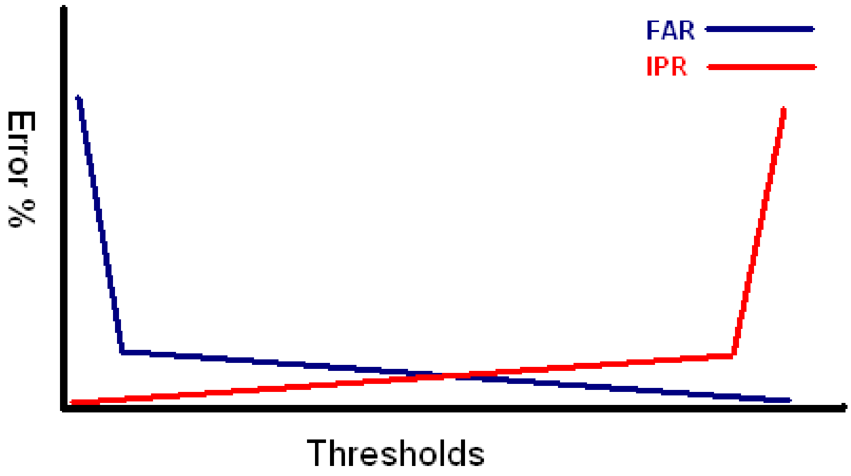


Figure 4.3: A graph of an analysis method that is *not* sensitive to threshold adjustments

*experimental* data but not a threshold that would be useable in a general setting. Alternatively, if we see that the error rate stays low from the 40% to 80% thresholds, then we could conclude that the 60% threshold would be acceptable in a practical implementation.

In order to make the graph comparisons fair, each of the metrics were graphed across 1000 thresholds with the min and max thresholds set to the threshold where IPR was 100% and FAR was 100% respectively.

# Chapter 5

# Results

In this chapter, we present the most relevant graphs for our experiment. The graphs presented in this chapter are scaled according to the method described in Chapter 4. The scales that were used can be found in Appendix C. General statistics about the typing samples can be found in Appendix D. Additional graphs using different scales can be found in Appendix E.

## 5.1 Authentication

The following comparisons were done only with key latencies using predefined text samples for authentication tests. Each graph depicted shows the IPR graph (starting from 100% going down to 0%) and the FAR graph (starting from 0% and going up to 100%). The IPR graph is always the smooth one while the FAR graph seems more like a step function. This is because given 70 users, we can simulate 4830 impostor attacks while only simulating 70 authentic login attempts.

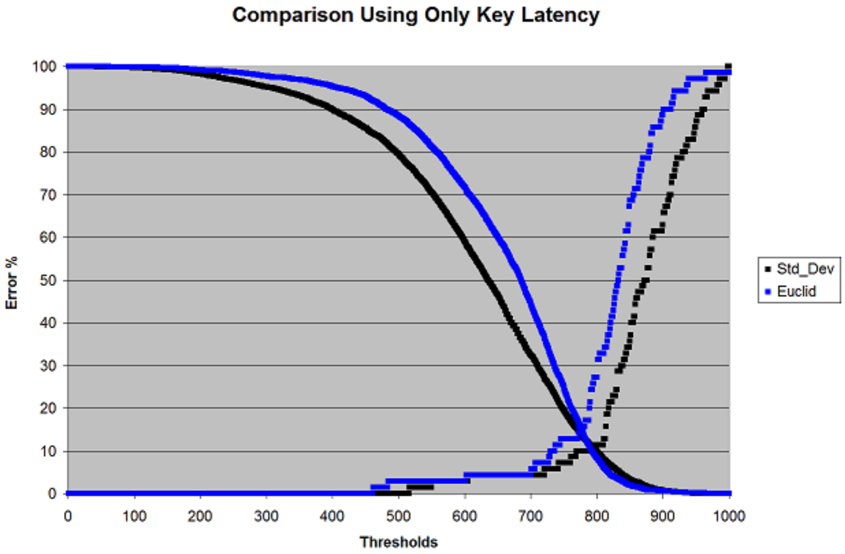


Figure 5.1: A graph of the standard deviation method vs. the Euclidean distance method.

We begin by comparing the first two metrics which are the standard deviation method and the Euclidean distance method. Looking at the graph in Figure 5.1, we see that the standard deviation method has an EER of around 10.5% and the Euclidean distance has an EER of 15.0%. Additionally, we see that the slope of the standard deviation graph is less steep around the EER, indicating that the standard deviation method is less susceptible to threshold adjustments.

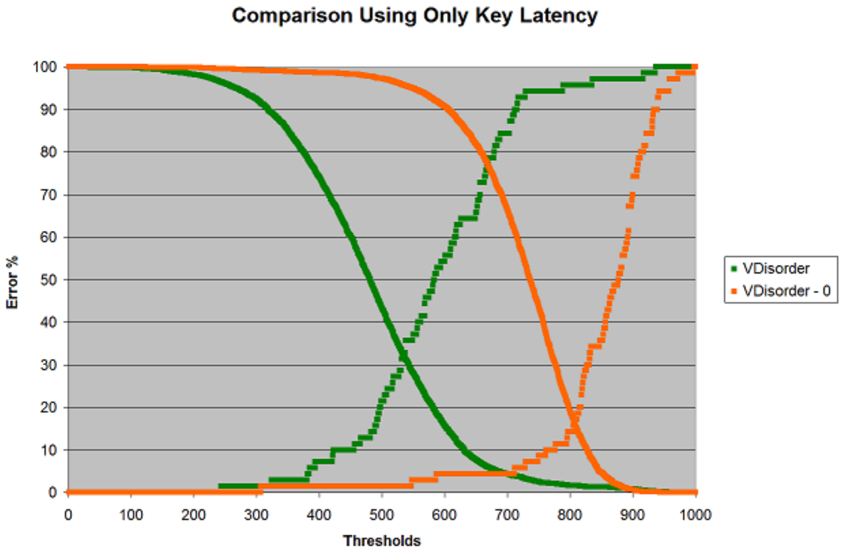


Figure 5.2: A graph of the vector disorder method with 0 entries and with 0 entries removed.

Moving on to the vector disorder method, we tested two ideas for the implementation. The research paper by Bergadano, Gunetti, and Picardi indicate that larger vectors will theoretically yield better results for the vector disorder method [BGP02]. One aspect of our data structure is that everyone is using large vectors of size 676. However, these vectors tend to end up with numerous zero entries since not every digraph is typed in a given typing sample. Leaving the zero entries would give us a larger vector but might also contaminate some of the vector disorder calculations. We graphed both methods to see which implementation was better. Looking at the graph in Figure 5.2, we see that removing the zero entries reduces the EER from around 31.5% to about 15.5%. Clearly for our data structure, removing the zero entries is the best option.

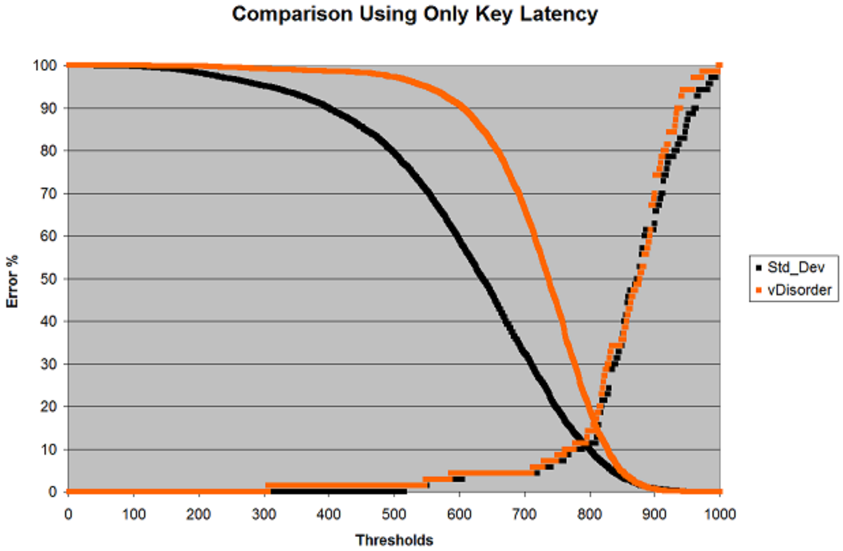


Figure 5.3: A graph of standard deviation method vs. the vector disorder method.

Figure 5.3 depicts the comparison of the vector disorder method versus the standard deviation method. Once again standard deviation is the superior metric with an EER of 10.5% compared to vector disorders 15.5%. The vector disorder method also has a steep slope around the EER which is surprising given that relative measures are suppose to be more robust than absolute measures. We discuss this result in more detail in Chapter 6.

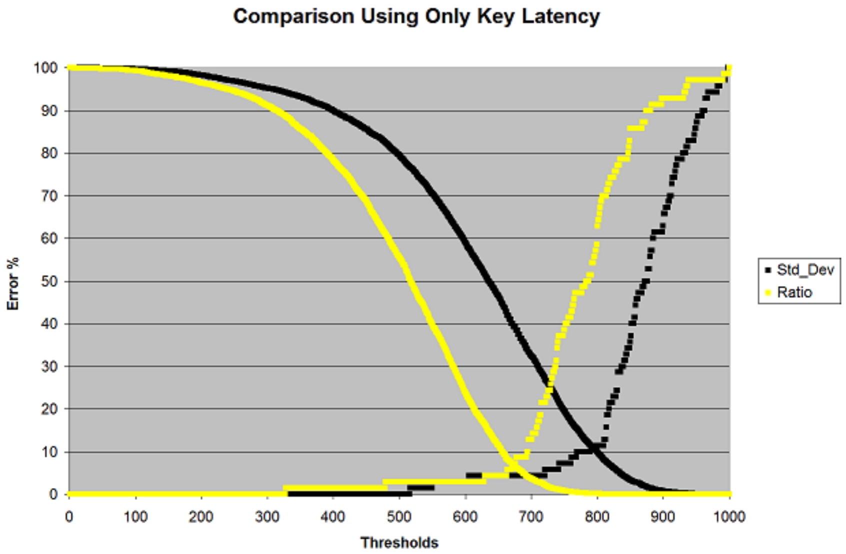


Figure 5.4: A graph of the standard deviation method vs. the ratio method.

Finally, we compare the standard deviation strategy against the ratio method in Figure 5.4. The ratio method outperforms the standard deviation method with an EER around 5.0%. Examining the data more closely, we determined that the best possible result we could achieve with the ratio method was a FAR of 4.29% and an IPR of 8.55%. The slope near the EER of the ratio graph is roughly the same as the standard deviation graph, leading us to the conclusion that the two metrics are equal in terms of sensitivity to threshold adjustments. After looking at each metric, we determined that the ratio method was the most successful strategy for keystroke authentication.

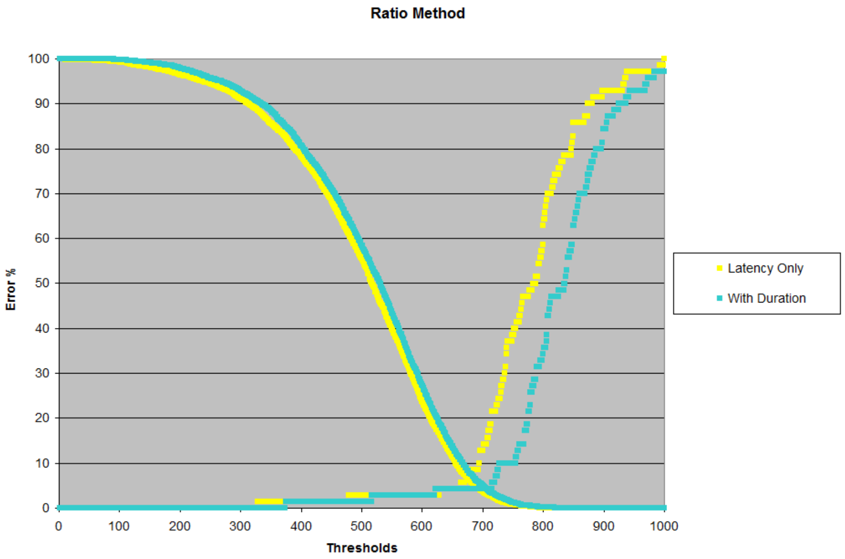


Figure 5.5: A graph of the ratio method using only key latencies versus key latencies and key durations

Once we determined the best authentication strategy, we looked into use of keystroke durations and free text analysis. Using keystroke latencies and keystroke durations we were able to improve the error rates of the ratio method slightly. Our best result was a FAR of 4.29% and an IPR of 3.37%. However, the improvement was not overly significant as can be seen in the Figure 5.5. The graphs for the other three metrics had similar results of only minor improvements with the addition of keystroke durations.

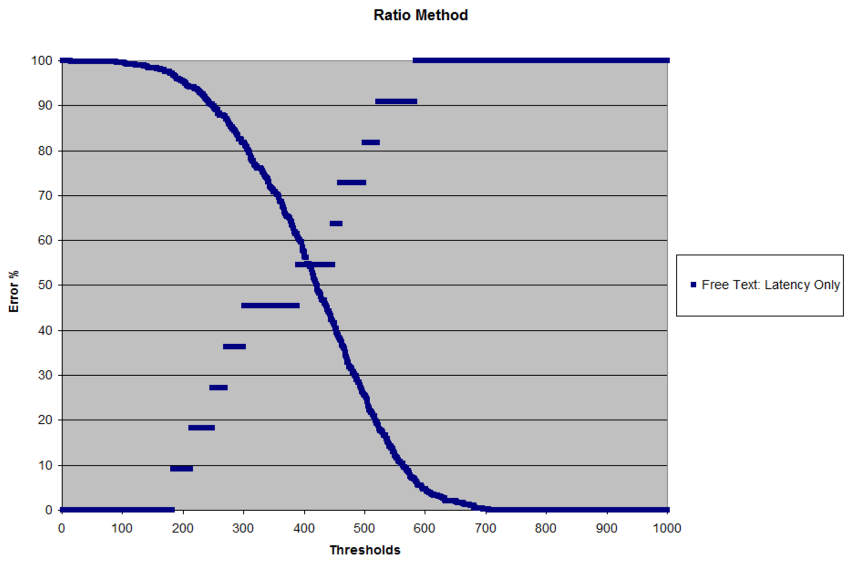


Figure 5.6: A graph of the ratio method on free text authentication.

Free text authentication results were far less successful than predefined text authentication. All the error rates were above 50%. The best result was once again the ratio method and the graph can be seen in Figure 5.6. We achieved an FAR of 54.55% with an IPR of 42.03% in the best case.

## 5.2 Classification

| Classification Results | | |
|---|---|---|
| | Predefined Text | Free Text |
| Std. Dev. | 44.29% | 72.72% |
| Euclidean | 35.72% | 63.64% |
| Ratio | 18.57% | 54.55% |
| vDisorder | 61.43% | 100% |
| vDisorder - 0 | 45.29% | 90.91% |

Table 5.1: Table of Classification Results

Table 5.1 shows the different error rates achieved for each statistical method. The excessively high error rates of the vector disorder metric without the removal zero entries further strengthens our claim that zero entries need to be removed in our implementation. The ratio method performed the best for classification of both predefined and free text samples. This result firmly confirms that the ratio method is the superior of the four metrics. One surprising result we saw in classification was that Euclidean distance proved to be a better classifier than the standard deviation metric. We explain our reasoning for this in Chapter 6

# Chapter 6

# Discussion of Results

The results that we presented in Chapter 5 show that we were able to achieve error rates on par with the 5% error rates of the studies we reviewed in Chapter 2. However, we did have some surprising results and some exceptionally high error rates for some cases. We will proceed to discuss these results further in this chapter.

## 6.1 Goals

We want to begin by discussing the four goals we had for this experiment. We definitely achieved our first goal of performing a fair comparison of several statistical strategies. Our second goal of examining "global" patterns was not as successful. Based on our research, we concluded that such patterns do not exist for the general user. A more detailed discussion of our conclusion can be found in the later section of this chapter. Our third goal of testing the effectiveness of our simplified data structure was very successful. In general, we found that our simplified reduction of the typing samples into our data structure still resulted in the low error rates we presented in Chapter 5. However, the data loss associated with our reduction may have caused the disappointing performance of the vector disorder method. We discuss this further in the next section. Finally, our fourth goal of emphasizing ideas useful for practical implementations of keystroke dynamics was fulfilled with the nature of data collection and our metric of graphing thresholds. Our data collection did not discriminate between users of varying typing proficiency because we believe that any practical authentication system needs to work for your general user. We also perform minimal pruning of the data and no user samples was completely removed. The method of graphing out thresholds to examine sensitivity to threshold adjustments turned out to be an informative metric that we believe future research should consider.

## 6.2 Vector Disorder

Our literature review of keystroke dynamics convinced us that the vector disorder would be the metric with the lowest error rates and the least sensitive to threshold adjustments. However, the opposite of this was true. After examining our implementation carefully, we believe that the cause of this is that because our data structure is sufficient for maintaining absolute timings but horrible for maintaining the integrity of relative timing information. For our experiment, we wanted to make a fair comparison so we made all the statistical methods to use a standard data structure. Given more time, we would want to redesign our data structure to be closer to the implementation described by Gunetti and Picardi [GP05].

## 6.3 Classification

In the authentication tests, the rankings of the methods from worst to best was: vector disorder, Euclidean distance, standard deviation, and ratio. In the classification tests, the rankings of Euclidean distance and standard deviation were swapped. We believe that this is because Euclidean distance takes into account every digraph measurement and therefore examines the typing sample as a whole. This is mentioned in Chapter 4, and we want to emphasize again that not looking at a typing sample as a whole can cause problems where 60% of a sample maybe similar but the other 40% is dramatically different. This is mostly likely what happened to the standard deviation classification method.

## 6.4 Keystroke Durations

Previous studies that have looked at using keystroke durations have found that duration timings are much more accurate for authentication than latency timings are. In our experiment we were not able to replicate this success when we used keystroke durations in addition to keystroke latencies. Comparing our study to previous experiments, we think this lack of improvement is caused by our data structure. When previous studies included keystroke duration timings, they effectively doubled the sized of the samples they were comparing. With our data structure, adding keystroke durations only adds 26 more entries to a latency vector of size 676. Since keystroke durations does not give us a significant amount of additional data, it makes sense that adding these timings would not significantly improve our results the way it did in previous studies.

## 6.5 Free Text: Global Patterns

All of the metrics we tested were not able to achieve good results when we tested free text authentication. This led us to conclude that "global" typing patterns do not exist for the average user. Careful examination of the graphs in Chapter 5 did make us think that "global" patterns may exist for *certain* individuals. Looking at Figure 5.4, we see that the ratio method graph is essentially the standard deviation graph shifted left and down. This make sense given the similarity of the two metrics and how they're trying to maximize number of valid digraphs. The ratio is simply a better definition of "valid". The almost identical slopes of the two metrics in this figure however hint at the fact that the same users may be the cause of the same errors. If this is the case, this tells us that certain users simply do not have consistent enough typing for keystroke dynamics to work successfully. The inclusion of these inconsistent users in our data set can skew our results to have higher error rates. Previous free text studies have achieved results of FAR and IPR of below 5% [GP05]. Our failure to replicate this success could be the side effect of us not controlling the quality of the typists that participated.

# Chapter 7

# Future Work

We are satisfied with the progress that was made in this experiment but we also have several ideas for future research. Not only would more research improve keystroke dynamics commercial products, any work done in keystroke dynamics can also be used to aid the research of other behavioral biometric research. Every behavioral biometric authentication scheme essentially is trying to find a way of representing a reference and comparing a claimant sample against the reference. The latest behavioral biometric that has been suggested is gait-based authentication[GSB07], where users can be identified and authenticated based on how they walk. We wish to emphasize that the following proposed ideas can easily be applied behavioral biometric research in general.

## 7.1 Improved Data Collection

One of the many problems is that there has been a lack of "real" typing data collected. Every experiment so far has always involved asking the user to type in a controlled situation. The closest to "real" data collection so far are the Robinson et al. study [RLCM98] that recorded login strings and the Gunetti and Picardi study on free text [GP05]. We believe that future experiments should look into having volunteers install keystroke loggers on their computers so that their every day use of computers can be logged. This would have two immediate benefits. One, the amount of data gathered would be significantly larger than any previous experiment and would also allow researchers to make claims about "global" patterns that exist in users regardless what typing situation they are in. Two, it would eliminate the user complaint factor. It's extremely difficult to get users to sit down and submit good typing data for long periods of time. Additionally, in practical situations, it is unlikely that users will be willing to spend 20 minutes creating a typing profile so they can authenticate. In our experiment, only 2 users told us that they enjoyed the experiment (they liked the typing sample choice) with the majority of users complaining about the length of the data collection when they've gotten through about half the experiment. This leads us to our next research idea of user acceptance.

## 7.2 User Acceptance

The idea proposed in the previous section about installing keystroke loggers may raise some privacy concerns from researchers. We should consider that if we can't convince volunteers in an experiment to install a keystroke logger, how likely is it that an average user would be willing to install a keystroke logging authentication system on their computers? To our knowledge, no study has been conducted on user attitudes towards a keystroke authentication system. Additionally we need to gauge how much annoyance an user can tolerate for authentication. What level of false alarms is tolerable by an user? How long of a typing sample are users willing to submit to authenticate? How long are users willing to spend to create the reference profiles? These are all important questions that need to be answered if keystroke dynamics authentication is to gain wide-spread acceptance.

## 7.3 Dynamic Authentication

The idea of dynamic authentication has been mentioned and discuss in several papers but in order to truly test and implement such a system, there needs to be long term keystroke logging of users. Current commercial products such as Biopassword [CMD$^+$] only perform static authentication. Any study that looks at dynamic authentication will need to adopt a similar data structure that we used in our experiment and have volunteers commit to the experiment for a long period of time.

## 7.4 Conclusion

In conclusion, this experiment has convinced us that keystroke dynamics is a viable technology that can definitely be improved on. While our results were not spectacular, the fact that we were able to replicate the results described in previous studies eliminates a lot of our earlier skepticism. However, we believe that keystroke dynamics is still in the theoretical stages and no where near practical success yet.

# Appendix A

# Glossary of Key Terms

**Authentication** Given a new typing sample X. *"X is claimed to belong to user U. The system must decide if this is true or false. X may belong to U, to another known user, or to someone else (whose typing habits are) completely unknown to the system."* [GP05]

**Claimant** Claimant Vector, Claimant Sample, etc ... The typing sample that we take in and compare to against a reference. It is a sample from someone "claiming" an identity and we need to verify this claim.

**Classification** Given a new typing sample X. *"X comes from one of the known users. The system must find who actually provided the sample."* [GP05]

**Digraph** A two keystroke combination. Sometimes used to refer to the timing of that keystroke combination. The timing of a digraph is defined as the time elapsed from the release of the key and the depression of the second key. See **Key Latency**

**EER** Equal Error Rate. The point where FAR and IPR are equal.

**FAR** False Alarm Rate. An error rate that represents how often a False Alarm occurs. A False Alarm is when a valid user logging in is rejected.

**Identification** Given a new typing sample X. *"X is presented to the system. The system has two possible answers: (a) X belongs to user U; or (b) X belongs to someone unknown. As in the case of authentication, X may, in fact, belong to one of the known users, or to someone unknown to the system."* [GP05]

**IPR** Impostor Pass Rate. An error rate that represent how often an Impostor Pass occurs. An Impostor Pass is when an impostor attempts to login and is accepted.

**Key Duration** The amount of time a key is depressed for. It is calculated as the time elapsed from when the key is first depressed to the time it is released.

**Keystroke Dynamics** The timing information that describes when keys are depressed and released. Sometimes also used to signify the idea of using the timing information for authentication/identification.

**Key Latency** The time elapsed from the release of the first key and the depression of the second key

**Over-Fitting** A situation where a method is modified to the point where it achieves near perfect results for the experimental data. However, this optimized method does not work successfully given a new set of data.

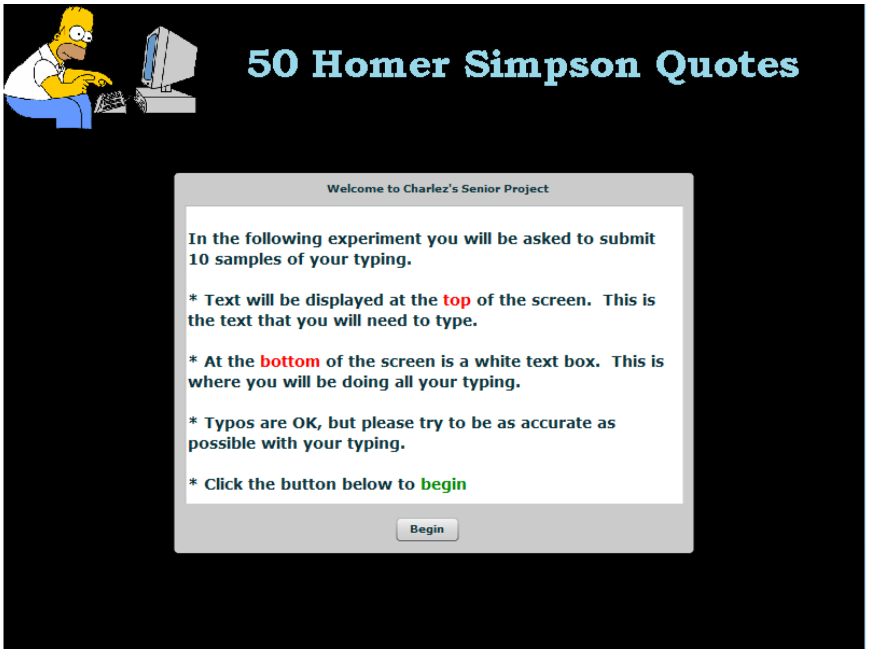
**Reference** Reference Vector, Reference Profile, etc . . . The sample stored on file as the "authentic" typing sample. The reference is submitted by the user initially and is used to determine the validity of a Claimant Sample. Validity is determined by how similar/close the Claimant is to the Reference. The closer the better.

**Trigraph** A three keystroke combination. The timing of trigraph is defined as the time elapsed from the release of the first key to the depression of the third key.

# Appendix B

# Data Collection: Supplemental

Some supplemental info about the data collection aspect of the experiment



Welcome to Charlez's Senior Project

In the following experiment you will be asked to submit 10 samples of your typing.

* Text will be displayed at the top of the screen. This is the text that you will need to type.

* At the bottom of the screen is a white text box. This is where you will be doing all your typing.

* Typos are OK, but please try to be as accurate as possible with your typing.

* Click the button below to begin

Figure B.1: A screenshot of the web application we implemented

The following is a small sample of what users were asked to type

- Operator. Give me the number for 911.
- Oh, so they have internet on computers now.
- Bart, with $10,000, we'd be millionaires. We could buy all kinds of useful things like...love.
- Just because I don't care doesn't mean I don't understand.
- I'm normally not a praying man, but if you're up there, please save me superman.
- You know, boys, a nuclear reactor is a lot like a woman. You just have to read the manual and press the right buttons.
- Lisa, if you don't like your job you don't strike. You just go in every day and do it really half-assed. That's the American way.
- When will I learn? The answer to life's problems aren't at the bottom of a bottle, they're on TV!
- Son, when you participate in sporting events, it's not whether you win or lose: it's how drunk you get.
- I'm going to the back seat of my car, with the woman I love, and I won't be back for ten minutes!

# Appendix C

# Scale of Graphs

These are the scales used for the graphs in Chapter 5

| Scales | | | |
|---|---:|---:|---:|
| | 100% IPR | 100% FAR | Threshold Increment |
| Std. Dev. | 0.074 | 0.705 | 0.000631 |
| Euclidean | 31155 | 7900 | -23.255 |
| Ratio | 0.083 | 0.703 | 0.000620 |
| vDisorder | 0.214 | 0.107 | -0.000107 |
| vDisorder - 0 | 0.694 | 0.240 | -0.000454 |

Table C.1: Scales for graphs

# Appendix D

# Database Statistics

The following is some statistical data about the digraph timings. Number of times a digraph was typed refers to the maximum number times that digraph was typed in one sample. The units of measure for the timing data is milliseconds.

| | |
|---|---:|
| Minimum Digraph Time: | 1 |
| Maximum Digraph Time: | 500 |
| REFERENCE PROFILE DATABASE STATISTICS | |
| Number of users in database: | 70 |
| Average Number of digraphs typed out of 676: | 317.542857143 |
| Digraph typed the most: | th |
| Number of times th was typed: | 54 |
| Average Number of digraphs users typed: | 1784.44285714 |
| Average Number of keystrokes typed out of 26: | 23.7714285714 |
| Average Keystroke Duration: | 121.399490048 |
| CLAIMANT SAMPLES DATABASE STATISTICS | |
| Number of users in database: | 70 |
| Average Number of digraphs typed out of 676: | 296.471428571 |
| Digraph typed the most: | th |
| Number of times th was typed: | 51 |
| Average Number of digraphs users typed: | 1501.2 |
| Average Number of keystrokes typed out of 26: | 23.8857142857 |
| Average Keystroke Duration: | 127.092522121 |
| FREE TEXT SAMPLES DATABASE STATISTICS | |
| Number of users in database: | 11 |
| Average Number of digraphs typed out of 676: | 152.0 |
| Digraph typed the most: | th |
| Number of times th was typed: | 26 |
| Average Number of digraphs users typed: | 333.727272727 |
| Average Number of keystrokes typed out of 26: | 21.7272727273 |
| Average Keystroke Duration: | 74.578198301 |

Table D.1: Database Statistics

# Appendix E

# Additional Graphs

The following graphs are on the scales of minimum possible score to maximum possible score. For example, with the Euclidean distance, the minimum possible distance is 0 and the maximum distance would be 676 multiplied by maximum allowed digraph time. The IPR graph is the smooth curve that starts at 100% and declines. The FAR graph is the dotted curve that starts at 0% and increases.

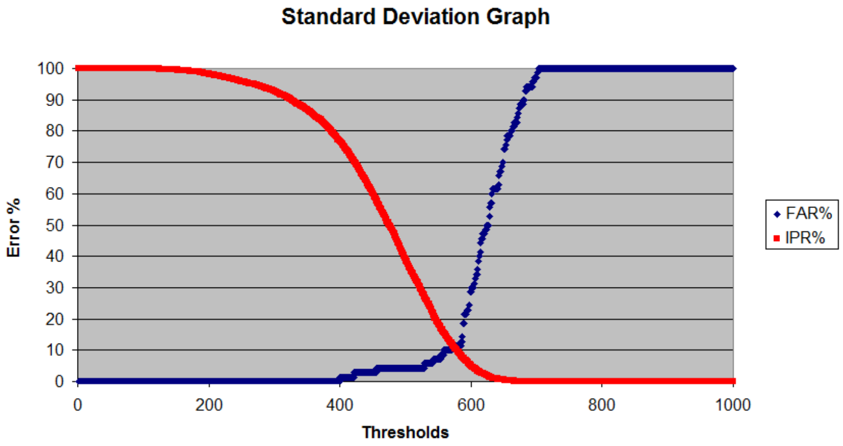


Figure E.1: A graph of the standard deviation method scaled from 0% to 100%

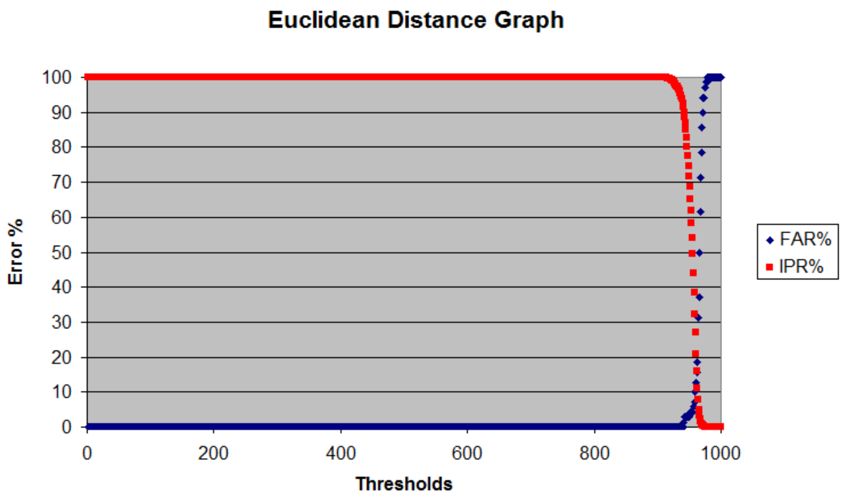


Figure E.2: A graph of the Euclidean distance method scaled from 0 to 338000

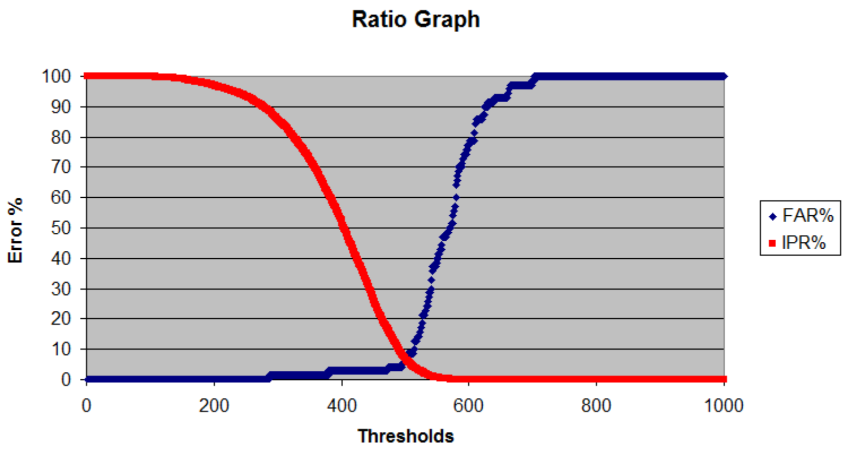


Figure E.3: A graph of the ratio method scaled from 0% to 100%

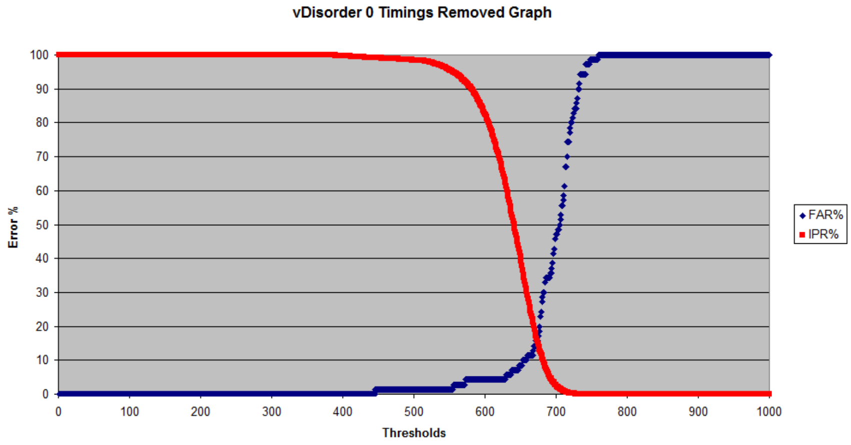


Figure E.4: A graph of the vector disorder method scaled from a normalized disorder of 1.0 to 0.0

# Bibliography

[BGP02] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. User authentication through keystroke dynamics. *ACM Trans. Inf. Syst. Secur.*, 5(4):367–397, 2002.

[Bis03] M. Bishop. *Computer Security: Art and Science.* Addison-Wesley Professional, 2003.

[BR94] M. Brown and S.J. Rogers. A practical approach to user authentication. *Computer Security Applications Conference, 1994. Proceedings., 10th Annual*, pages 108–116, 5-9 Dec 1994.

[BSB02] J. Bechtel, G. Serpen, and M. Brown. Passphrase authentication based on typing style through an ART 2 Neural Network. *International Journal of Computational Intelligence and Applications*, 2(2):131–152, 2002.

[CF07] N.L. Clarke and S.M. Furnell. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1):1–14, 2007.

[CHHK00] S. Cho, C. Han, D.H. Han, and H.I. Kim. Web-Based Keystroke Dynamics Identity Verification Using Neural Network. *Journal of Organizational Computing and Electronic Commerce*, 10(4):295–307, 2000.

[CMD$^+$] K. Code, S. Mani, D.M. D'Andrea, V.P. Images, LLC BioPassword, and P. Class. Method and apparatus for multi-model hybrid comparison system.

[Cou] F.F.I.E. Council. FFIEC guidance: Authentication in an Internet banking environment, Oct. 2005.

[Gar86] J.D. Garcia. Personal identification apparatus, November 4 1986. US Patent 4,621,334.

[GLPS80] R. Gaines, W. Lisowski, S. Press, and N. Shapiro. Authentication by Keystroke Timing: Some Preliminary Results. Technical report, Report R-256-NSF. Rand Corporation, 1980.

[GP05] Daniele Gunetti and Claudia Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, 2005.

[GSB07] D. Gafurov, E. Snekkenes, and P. Bours. Spoof Attacks on Gait Authentication System. *Information Forensics and Security, IEEE Transactions on*, 2(3 Part 2):491–502, 2007.

[HLC06] S. Hwang, H. Lee, and S. Cho. mproving Authentication Accuracy of Unfamiliar Passwords with Pauses and Cues for Keystroke Dynamics-Based Authentication. *WISI*, pages 73–78, 2006.

[JG] R. Joyce and G. Gupta. Identity Authentication Based on Keystroke Latencies.

[KC07] S. Karatzouni and N. Clarke. Keystroke Analysis for Thumb-based Keyboards on Mobile Devices. *International Federation for Information Processing Publications*, 232:253, 2007.

[LW88] J. Leggett and G. Williams. Verifying identity via keyboard characteristics. *Int. J. Man-Machine Studies*, 28(1):67–76, 1988.

[LWU89] J. Leggett, G. Williams, and D. Umphress. Verification of user identity via keystroke characteristics. *Human Factors in Management Information Systems*, 1989.

[MR97] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *CCS '97: Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56, New York, NY, USA, 1997. ACM.

[MRW02] F. Monrose, M.K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[OS97] M.S. Obaidat and B. Sadoun. Verification of computer users using keystroke dynamics. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, 27(2):261–269, Apr 1997.

[PB07] H.J. Postley and S.S. Bender. Key sequence rhythm recognition system and method, April 17 2007. US Patent 7,206,938.

[Pol97] D. Polemi. Biometric Techniques: Review and Evaluation of Biometric Techniques for Identification and Authentication. *Institute of Communication and Computer Systems, National Technical University of Athens, April*, 1997.

[RLCM98] J.A. Robinson, V.W. Liang, J.A.M. Chambers, and C.L. MacKenzie. Computer user verification using login string keystroke dynamics. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 28(2):236–241, Mar 1998.

[UW85] D. Umphress and G. Williams. Identity verification through keyboard characteristics. *INT. J. MAN MACH. STUD.*, 23(3):263–274, 1985.

[YH89] J.R. Young and R.W. Hammon. Method and apparatus for verifying an individual's identity, February 14 1989. US Patent 4,805,222.