2007

User habitation in keystroke dynamics based authentication

Qi Cheng
West Virginia University

User Habituation in Keystroke Dynamics Based Authentication

Qi Cheng

Thesis submitted to the
College of Engineering and Mineral Resourses
at West Virginia Universiy
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

Bojan Cukic, Ph.D., Chair
Arun Ross, Ph.D.
Natalia Schmid, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2007

Keyword: Keystroke Dynamics, Behavioral Biometrics, Password Hardening, User Habituation, Two-password Mechanism

Abstract

User Habituation in Keystroke Dynamics Based Authentication

Qi Cheng

Most computer systems use usernames and passwords for authentication and access control. For long, password security has been framed as a tradeoff between user experience and password security. Trading off one for the other appears to be an inevitable dilemma for single password based security applications. As a new biometric for authenticating access, keystroke dynamics offers great promises in hardening the password mechanism. Our research first investigate the keystroke dynamics based password security by conducting an incremental study on user's habituation process for keystroke dynamics analysis using two distinct types of passwords. The study shows that 1) long and complex passwords are more efficient to be employed in keystroke dynamics systems; and 2) there is a habituation and acclimation process before the user obtains a stable keystroke pattern and the system collects enough training data. Then, based on our findings, we propose a two passwords mechanism that attempts to strike the right balance over user experience and password security by adopting a conventional easy-to-memorize password followed by a long-and-complex phrase for keystroke dynamics verification. Analysis and experimental studies successfully demonstrate the effectiveness of our proposed approach.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

In modern society, the username/password scheme is widely used to control access to a resource. Especially in computer networking and related fields, users may require passwords for many purposes: logging in to computer accounts, retrieving email from servers, accessing databases, etc. Once the password is stolen, the intruder will have access to the private data and have full set of rights of the authorized user. To reduce the risk of lost confidentiality and integrity, an effective password protect scheme is required.

Keystroke dynamics, which is a new biometric technology, has been developed to enhance the security of username/password verification scheme. Keystroke dynamics, also referred to as typing rhythms, is behavioral in nature. User's habituation has been shown in the process of forming his/her unique typing pattern. However, there is no deep analysis of user's typing habituation in the current studies of keystroke dynamics. A study of user's habitation, which is the topic of our research, will be helpful for building a reliable keystroke dynamic system.

Another concern, also the subject of our research, is that the security is increased with a consequent loss of convenience for users using

username/password verification. A balance between security and convenience should be found in the deployment of keystroke dynamic system.

1.2 Goal

Based on previous work, the goal of this study is to provide evidence of user habituation in keystrokes while typing passwords. As the password protecting policy of keystroke dynamics system is different from that of traditional password verification system, another goal of the study is to analyze their differences and evaluate the application potential of keystroke dynamic systems.

1.3 Contribution

An incremental learning experiment described in this thesis provides evidence of user habituation when they type two kinds of passwords: short common English words and long random character string with shift-key behavior.

The study also discusses the vulnerabilities of long passwords and short passwords. Considering the characteristics of traditional password system and the requirements of keystroke dynamics system, our study introduces a two-password mechanism which takes advantage of both, and finds a balance between security and convenience.

1.4 Organization

The thesis consists of 6 chapters. Chapter 2 is a literature review of user authentication systems, performance measures used in our study, and

selected data mining algorithms. Chapter 3 gives an overview of password security systems and keystroke related studies and describes the experimental design in our previous research and discusses the experimental results. Chapter 4 focuses on the results of incremental learning experiments and the system using Random Forest for matching / classification while also providing trend lines of the performance. Chapter 5 offers considerations and limitations of traditional password systems, and then presents a way to combine keystroke dynamics with it to improve password security. Finally, Chapter 6 provides the conclusions and discusses the ways to further optimize achieved performance.

Chapter 2 Literature Review

2.1 Overview of User Authentication

Over time, organizations and systems have means of authentication, using voice recognition, fingerprint and iris matching, and other trusted meanings of identification. Authentication mechanisms use any of three qualities to confirm a user's identity [1] [2]:

1. What the user knows. Examples of what a user may know are passwords, PIN numbers, and pass phrases.

2. What the user has. Common examples that people have them recognizable are keys, a driver's license or a uniform.

3. Who the user is. The authentication methods, which are also called biometrics, are based on a physic characteristic of the user, such as fingerprint, the voice or the face.

Biometrics refers to technologies that are measurable physiological or behavioral characteristics which can be used to verify the identity of an individual.

Examples of physical (or physiological or biometric) characteristics include:

- Iris
- Fingerprint (including nail)
- Hand (including knuckle, palm, vascular)
- Face
- Voice
- Retina
- DNA
- Even Odor, Earlobe, Sweat pore, Lips

Examples of mostly behavioral characteristics include:

- Signature
- Keystroke (typing pattern)
- Voice
- Gait

To compare the performance of these types of biometrics, the following figure ranks each biometric based on seven categories as being low, medium, or high.

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystroke dynamics	L	L	L	M	L	M	M
Hand vein	M	M	M	M	M	M	H
Iris	H	H	M	L	H	L	H
Retina	L	L	L	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermogram	H	H	L	H	M	H	H
DNA	H	H	H	L	H	L	L

H=High, M=Medium, L=Low

Figure 2. 1: A Comparison of Biometrics [1]

The seven categories are:

- **Universality** describes how commonly a biometric is found in each individual.
- **Uniqueness** is how well the biometric separates one individual from another.
- **Permanence** measures how well a biometric resists aging.
- **Collect ability** explains how easy it is to acquire a biometric for measurement.
- **Performance** indicates the accuracy, speed, and robustness of the system capturing the biometric.
- **Acceptability** indicates the degree of approval of a technology by the public in everyday life.
- **Circumvention** is how easy it is to fool the authentication system.

As an emerging area with many opportunities for growth, there are concerns about biometrics. A sound trade-off between security and

privacy may be necessary; collective accountability/acceptability standards can only be enforced through common legislation. As biometrics technology matures, there will be an increasing interaction among the market, technology and the applications.

2.2 General principles of Biometric Systems

A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extraction a feature set from the acquired data, and comparing this feature set against the template set in the database.

A biometric recognition system can run in two different modes: identification or verification. Identification is the process of trying to find out a person's identity by examining a biometric pattern calculated from the person's biometric features.

In the identification case, the system is trained with the patterns of several persons. For each of the persons, a biometric template is calculated in this training stage. A pattern that is going to be identified is matched against every known template, yielding either a score or a distance describing the similarity between the pattern and the template. The system assigns the pattern to the person with the most similar biometric template. To prevent impostor patterns (in this case all patterns of persons not known by the system) from being correctly identified, the similarity has to exceed a certain level. If this level is not reached, the pattern is rejected.

In the verification case, a person's identity is claimed a priori. The pattern that is verified only is compared with the person's individual template. Similar to identification, it is checked whether the similarity between

pattern and template is sufficient to provide access to the secured system or area.

The similarity between a pattern and a biometric template is expressed by matching scores. The higher the score is, the higher is the similarity between them. In theory, client scores (scores of patterns from persons known by the system) should always be higher than the scores of impostors. If this would be true, a single threshold, that separates the two groups of scores, could be used to differ between clients and impostors [3].

Depending on the choice of the classification threshold, between all and none of the impostor patterns are falsely accepted by the system. The threshold depending fraction of the falsely accepted patterns divided by the number of all impostor patterns is called False Acceptance Rate (FAR).

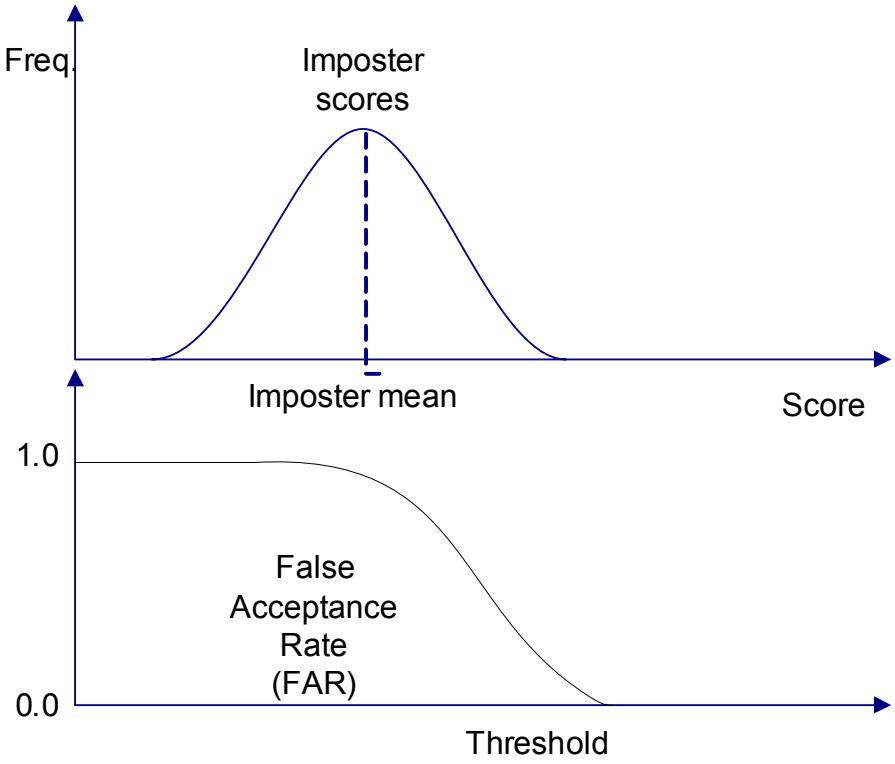

Figure 2. 2: Distribution of imposter scores and False Acceptance Rate

Similarly, the genuine pattern's scores vary around a certain mean value. If a classification threshold that is too high is applied to the classification scores, some of the client patterns are falsely rejected. The fraction of the number of rejected client patterns divided by the total number of client patterns is called False Rejection Rate (FRR).

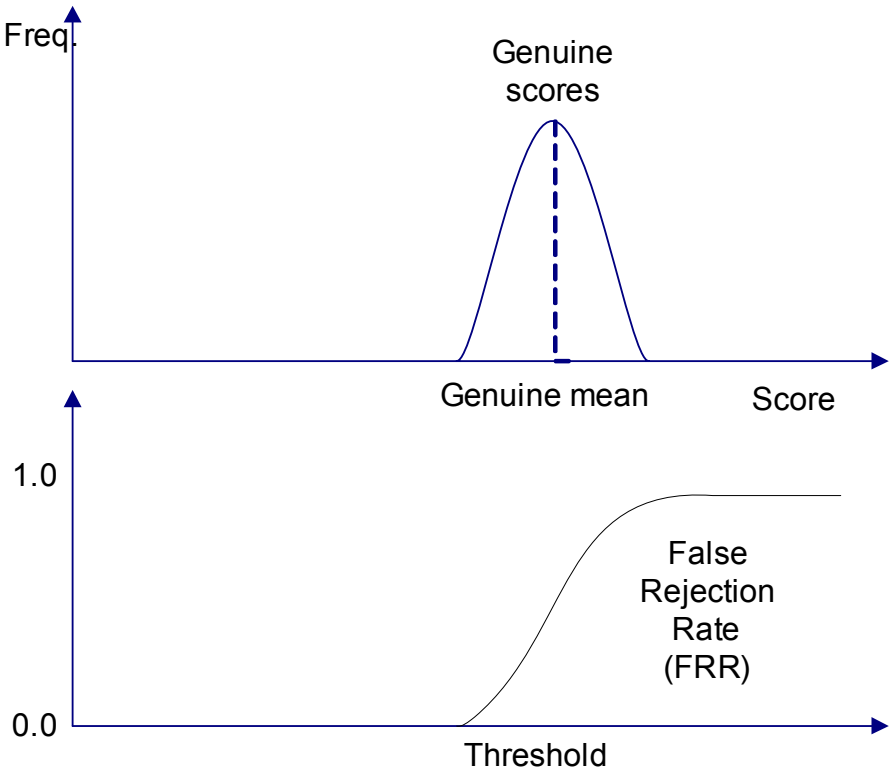

Figure 2. 3: Distribution of genuine scores and False Rejection Rate

If the score distributions overlap, the FAR and FRR intersect at a certain point. The value of the FAR and the FRR at this point, which is of course the same for both of them, is called the Equal Error Rate.

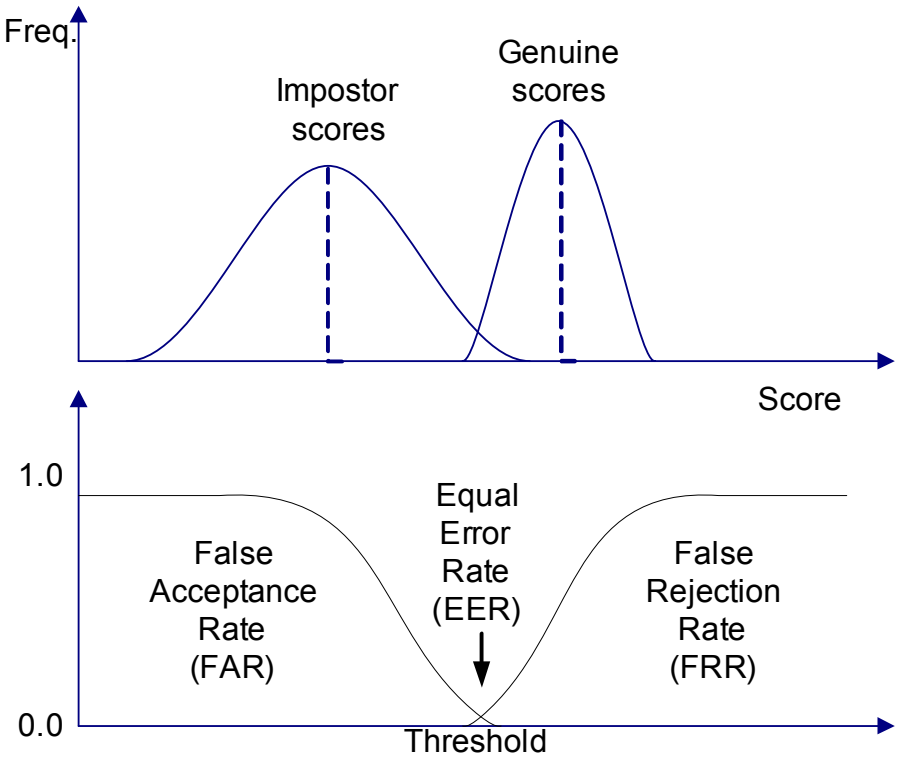

Figure 2. 4: The overlaps of distributions and FAR&FRR

In order to reach an effective comparison of different systems, a description independent of threshold scaling is required. One such example from the radar technology is the Receiver Operating Characteristic (ROC), which plots FRR values directly against FAR values, thereby eliminating threshold parameters. The ROC, like the FRR, can only take on values between 0 and 1 and is limited to values between 0 and 1 on the x axis (FAR).

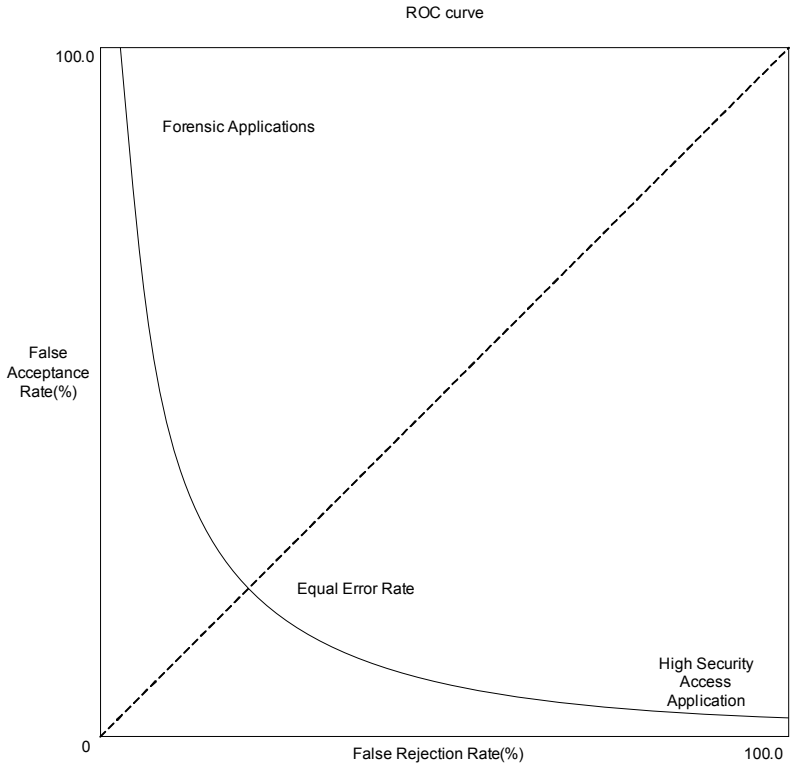

Figure 2. 5: FAR vs. FRR ROC Curve

2.3 Machine Learning and Random Forest

Machine Learning is an area of artificial intelligence concerned with the development of techniques which allow computers to change behavior in a way to improve future performance. Some parts of machine learning are closely related to data mining and statistics. Data Mining(DMM), also called Knowledge-Discovery in Databases(KDD) or Knowledge-Discovery and Data Ming[4], is data processing using sophisticated data search capabilities and statistical algorithms to discover patterns and correlations in large preexisting databases. Pattern recognition is also a branch of artificial intelligence [5]. It is the act of taking in raw data and taking an action based on the category of the data. These concepts were used within our keystroke analysis study to solve the two-classification problem (genuine, imposter).

The main algorithm used in our study is random Forest developed by Leo Breiman and Adele Cutler. In machine learning, a random forest is a classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. The method combines Breiman's "bagging" idea and Tin Kam Ho's "random subspace method" to construct a collection of decision trees with controlled variations[6][7][8].

Each tree is constructed using the following algorithm:

1. Let the number of training cases be N, and the number of variables in the classifier be M.

2. We are told the number m of input variables to be used to determine the decision at a node of the tree; m should be much less than M.

3. Choose a training set for this tree by choosing N times with replacement from all N available training cases. Use the rest of the cases to estimate the error of the tree, by prediction their classes.

4. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

For many data sets, random forest produces a highly accurate classifier. It has the ability to handle a very large number of input variables and it

includes a good method for estimating missing data. Even a large proportion of the data are missing, it maintains accuracy.

Random forest also provides an experimental way to detect variable interactions. It computes proximities between cases, useful for clustering, detecting outliers. The learning process in random forest is fast and it can balance error in class population unbalanced data sets.

Nowadays, random forest is widely used in micro array literature [9]. As it has several characteristics that make it ideal for data sets which have many noise predictive variables and many more variables than observations, it is also used for gene selection.

Chapter 3 Password Authentication Mechanism and Keystroke Dynamics

3.1 Password-Based Scheme

A big challenge for computer scientists is to protect partially shared or private resources from unauthorized users' accesses. The design of efficient and secure user authentication protocols is important. Among several suitable techniques over the years, password-based schemes are the most common due to their simplicity.

The password-based scheme is based on what the user knows and it seems to be secure if the user keeps his/her password a secret. However, a password can be compromised not only when the user accidentally discloses it, but also when the password is easy to guess.

Some attacks are referred to as dictionary attacks. Dictionary attack is a technique for defeating an authentication mechanism by try to determine its decryption key or pass phrase by searching through a large amount of possibilities. As most people have a tendency to choose passwords which are easy to remember and typically choose words taken from their native language, dictionary attacks succeed because they try possibilities which are most likely to be used by users.

Brute force attack is another method to defeat a password-based scheme. It works through all possible phrases to break the password. A theoretical feasibility of a brute force attack is recognized, but it would be computationally expensive to carry out. As the computation ability of modern computers grows, brute force attack can succeed in breaking the passwords with short lengths [10].

If a password is easy to guess, we call it weak or bad, while a hard-to-guess one is called good or strong. A password is weak if it can be guessed in a reasonable amount of time, while it is strong if the search requires unavailable resources in terms of time or space.

Some advices for choosing a good password are:

- Use at least 8 characters.
- Include a digit or punctuation.
- Use upper and lower case.
- Choose a phrase or combination of words to make the password easier to remember.
- Change password regularly and do not write it down.

To strengthen the security of password-based scheme, some solutions have been proposed, for example, one-time passwords approach and proactive password checking.

3.2 Keystroke Dynamics as a Biometric

The development of the biometric-based methods is strongly accelerated to satisfy the requests for controlled access to data-processing resources.

In most cases, these biometric-based methods may be expensive to implement. That is why we study the use of keyboard with is almost free and available on all computers, and use keystroke dynamics as a biometric method to enhance the reliability of password-based scheme.

Keystroke dynamics is the process of analyzing the typing patterns by monitoring the keyboard inputs in order to identify users based on his/her habitual typing rhythm. It has been proved that keystroke rhythm is a good sign of identity [11][12][13].

While a user is typing a string, key-down and key-up times are captured to drive features: duration of the key and keystroke latency. Duration of the key is the time interval between pressing and releasing the key and keystroke latency is the time between two keystrokes.

3.3 Related Previous Work

The first tests to differentiate people using the keystroke dynamics were carried out by Gaines et al. in 1980[14]. They investigated the possibility of using keystroke timings for authentication using a T-test. The T-test is a statistical tool used to check the assumption that two populations have the same standard deviation and average. They showed that it was possible to differentiate users' typing patterns. After that, many studies conducted experiments on this subject.

3.3.1 Feature Vector Collection

Several methods were used to extract feature vectors. One method is based on a measurement of the digraph latencies between reference strings, such as the timing vector of dimension (2n+1) in a password with n-characters [15]. For instance, in a password phrase "abc", the time

sequence vector {D1,L1,D2,L2,D3,L3,D4} contains the duration time of each keystroke and the latency time and the duration of the enter key.

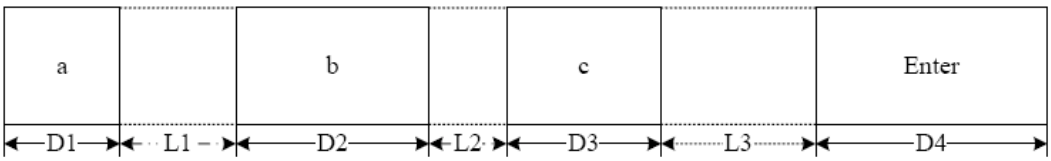

Figure 3. 1: The sequences of duration time and latency time

In [12] [16] [17], the authors used the two orthogonal components- total time the first key is pressed (i.e. keystroke duration), and the time between a key is released and the next key is pressed (i.e. keystroke latency.)

The second method is a classical method in the literature and it uses statistical data: the average and the standard deviation. A typical example is Leggett et al. used this method to authenticate users in [18] and [19]. The characteristics are extracted from the striking of a paragraph of text. During the creation of the profile, the average (μ) and the variance (σ^2) of each duration and latency time are calculated. Then a time o is declared valid if it is below ½*standard deviation (σ) from the average (μ). An observation is definitively validated if 60% of the times extracted during striking are valid. Other tests have been done on this method, in particular in [20]. The tests were carried on the definition of a difficulty degree α. So a time is considered as valid if: $|t- \mu| < \alpha * \sigma$.

Some feature sets are determined through factor analysis. Factor analysis seeks a lower dimensional representation that accounts for the correlation among features. This idea partitions the database of users into subsets whose in-class members are “similar” in typing rhythm over a particular

set of features and whose cross features are dissimilar in the corresponding sense[21]. For example, some users exhibit strong individualistic typing patterns for features in the set S={th, are, st, ion}, whereas some may be more distinctive over the features S={ere, on, wy}.

Some study is working in fact on discrete values associated with the numerical values of time [22]. For example, the time is classified into 5 classes: very short, short, average, long and very long. This method need to choose the threshold for the discretization. In a study of Sylvain Hocquet et al. [22], they realized a fusion of these methods, and obtained a significant improvement of performance with Z-score fusion.

3.3.2 Classification Methods

From traditional distance measurement methods to neural network techniques, algorithms employed in related keystroke studies are quite different. For example, Garcia [23] uses Mahalanobis distance function in their study; Young and Hammon[24] adopt Euclidean distance; neural Some network approaches have also been undertaken in [25],[32],[33] and [16]. Perceptron algorithm is used by Bleha and Obaidat in [26]. In [27], multiple machine learning algorithms were used.

3.3.3 Performance and Results

There is a relatively wide rage in performance over the two decades with published FAR ranging from 0-8% and FRR ranging from 0-45%. There are some commercial products on current market, such as BioPassword patented by Young[24]. Most of the studies improved the mechanism and performance of keystroke dynamic systems.

3.3.4 Typical Problems

The common problems that exist in the literature are:

1. The input target string cannot represent the user's typing pattern.
2. The number of samples is too small to obtain a good performance.
3. Feature vector sets chosen for identification purposes impose limitations on what can be achieved in the test.
4. Even the most efficient algorithms and classifiers have biases on the dataset.

3.4 Bartlow-Cukic Algorithm

3.4.1 Experimental Design and Data Collection

In the WVU experiment [28], each user was given two sets of username/password credential sequences. The username is of the form **Firstname.Lastname** with the first latter of each name capitalized. The first password was an eight letter lowercase English word taken from a cryptographic dictionary attack list, such as **computer** and **swimming**. The second password consisted of 12 random characters in a consistent pattern. The format of the pattern is:

SUUDLLLLDUUS

Where S is a special symbol, U is an uppercase letter, L is a lowercase letter and D is a digit. Examples of such passwords include **+JI5ftr8RE-** and **^DE2kum4WH?**.

For each sequence, key hold times and inter-key delays are collected in both the username and password. After some calculation, the following attributes were used to from a feature vector, as seen in Table 4.1.

1	userid	22	right_shift_hold_std
2	total_strokes	23	right_shift_hold_max
3	hold_avg	24	right_shift_hold_min
4	hold_std	25	right_shift_hold_total
5	hold_max	26	delay_avg
6	hold_min	27	delay_std
7	hold_total	28	delay_max
8	total_shifts	29	delay_min
9	shift_hold_avg	30	delay_total
10	shift_hold_std	31	left_shift_delay_avg
11	shift_hold_max	32	left_shift_delay_std
12	shift_hold_min	33	left_shift_delay_max
13	shift_hold_total	34	left_shift_delay_min
14	left_shifts	35	left_shift_delay_total
15	left_shift_hold_avg	36	right_shift_delay_avg
16	left_shift_hold_std	37	right_shift_delay_std
17	left_shift_hold_max	38	right_shift_delay_max
18	left_shift_hold_min	39	right_shift_delay_min
19	left_shift_hold_total	40	right_shift_delay_total
20	right_shifts	41	type {G=genuine, I=Imposter}
21	right_shift_hold_avg		

Table 3. 1: Feature Vector Collected for Each Input Sequence

The final database had a total of 53 users with 10,000 total input sequences. The demographics of the database represent a fairly diverse population. The gender split was approximately half and half, ages ranged from mid-teens to individuals in their early 60's. The typing ability of the population was also very diverse, ranging from typing beginners to individuals with professional keystroke experience.

3.4.2 Algorithmic Approach

The previous research demonstrated that Random Forests are superior in terms of overall performance, FAR, and FRR for the datasets tests than OneR, NaiveBayes, VotedPerceptron, LogitBoost and C5.0.

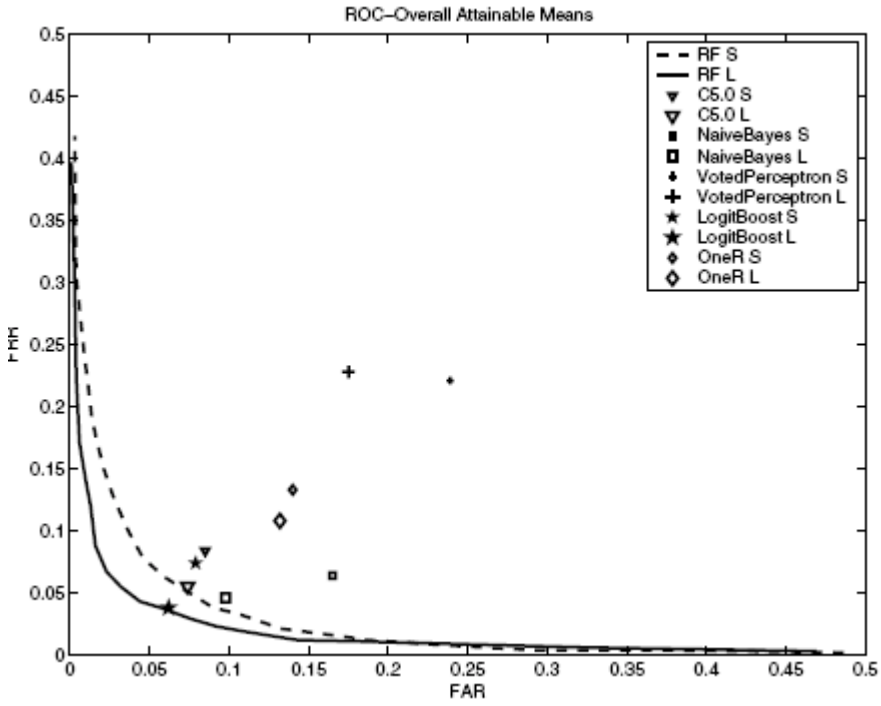

Figure 3. 2: Overall System Performance ROC Curve

Random Forests is an algorithm that constructs a set of decision trees randomly generated from a subset of the attributes in a feature vector. After construction of the forest of decision trees, a vote is taken across all individual trees to determine the class of a given input sequence.

3.4.3 Results and Conclusions

The system is capable of operating at various points on a traditional ROC curve depending on application specific security needs. A 1% False Accept Rate is attainable at a 14% False Reject Rate. An Equal Error Rate of 5% is suitable for systems requiring a relatively low security. As a username/password authentication scheme, the approach decreases the system penetration rate associated with compromised passwords by 95%-99%.

The results show that complex password sequences involving shift-key behavior offer a noticeable improvement in performance (classification accuracy, FAR and EER) over short password sequences that do not require the extensive use of the shift keys.

Chapter 4 Incremental Study

In [28], authors have demonstrated the performance of keystroke dynamics classification, proving that keystroke dynamics can be applied as an effective credential hardening mechanism. Unfortunately, the notion of personalization in user habituation brings issues caused by the diversity of typing abilities across different users, which range from the "hunt and peck" typists to individuals with professional training experience. It is imperative that we study the impact of such diversity over the keystroke dynamics. Furthermore, considering the evolving nature of the keystroke patterns over time, we designed an study on user habituation process and analyzed the data collected from the process.

4.1 User Habituation

As the passwords in our experiment are chosen by the computer, not every user was familiar with the typing sequences and had a smooth, quick and accurate typing style when they started using this system. Most beginners often hesitate between keystrokes and have to think about which key to press. However, while they were getting more and more familiar with their typing sequences, they were developing their typing skills and forming their unique typing patterns. Forming a unique typing pattern is a non-associative learning process. User might be unconscious

about his/her typing habit, but the underlying habituation occurs during the training process.

One concern is how long the process of creating a typing habit takes. In WVU experiment [28], we asked our users to practice the password 5-10 times before using the system. However, many, if not all users admitted to having neglected this instruction. Assuming most individuals were fairly familiar with typing their username, the collected trails should indicate the password acclimation process.

With the algorithm and datasets in Section 3.4, we are able to conduct the experiments with differing sizes of training sets. In the training sets, the number of genuine and imposter sequences is respectively increased from 6 to 30. The test set was defined from the remaining genuine and imposter sequences beyond the first 30. Using this system, the average test size was 58 and 59 sequences for short and long passwords respectively.

Our experiments were carried out using the Random Forest algorithm [8]. We attained the equal error rate for each user (30 users for long sequences and 33 users for short sequences), and then calculated the average equal error rates with differing number of samples in the training sets.

The average equal error rates across all users are listed in Table 4.1. The errors for most long passwords datasets turn out to be lower than those of short passwords by approximately 3%. On average, the long password sets produced 3.56% fewer errors than the short passwords set.

Number of Training Sequences	Average EER		
	Short	Long	Short-Long
6	0.135227	0.126579	0.00864765
9	0.14875	0.087469	0.06128095
12	0.121086	0.078887	0.04219854
15	0.101546	0.06745	0.03409582
18	0.092074	0.055666	0.03640771
21	0.085551	0.052098	0.03345294
24	0.084637	0.042157	0.04248027
27	0.074278	0.037469	0.03680865
30	0.07037	0.036888	0.03348222
Total Average	0.101502	0.064963	0.03653942

Table 4. 1: The average equal error rate (EER) of experiments with differing numbers of training sequences.

(The columns denoted as 'short' and 'long', respectively, indicate datasets trained with short passwords and long passwords. The column denoted as 'Short-long' indicates the difference in value of average EER between short passwords and long passwords.)

Figure 4.1 shows the influence of the number of training sequences used in the experiments. The two lines in Figure 4.1 represent the average EER's across all users who have entered a sufficient number of genuine and imposter sequences. As mentioned before, the training set was incrementally increased as seen on the x-axis in the figure (6, 9, 12, 15, etc…).

The average equal error rate drops significantly as the amount of training sequences increases. The maximum value of errors for short passwords is 14.9% when 9 sequences are used in the training set, while a 7.0% EER is achieved as the number of training sequences is increased to 30. For long password, the EER decreases from 12.7% to 3.7%.

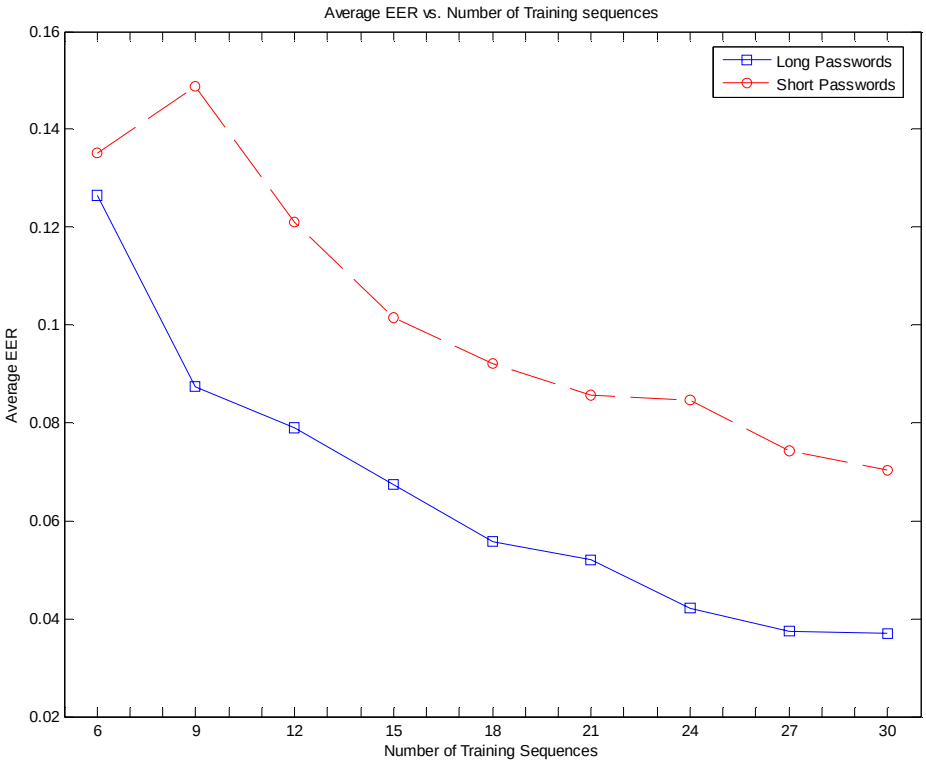

Figure 4. 1: User Habituation:

EER's achieved by incrementally training the Random Forests classification algorithm with password typing sequences

The following four box plots represent the distribution of equal error rates for each dataset.

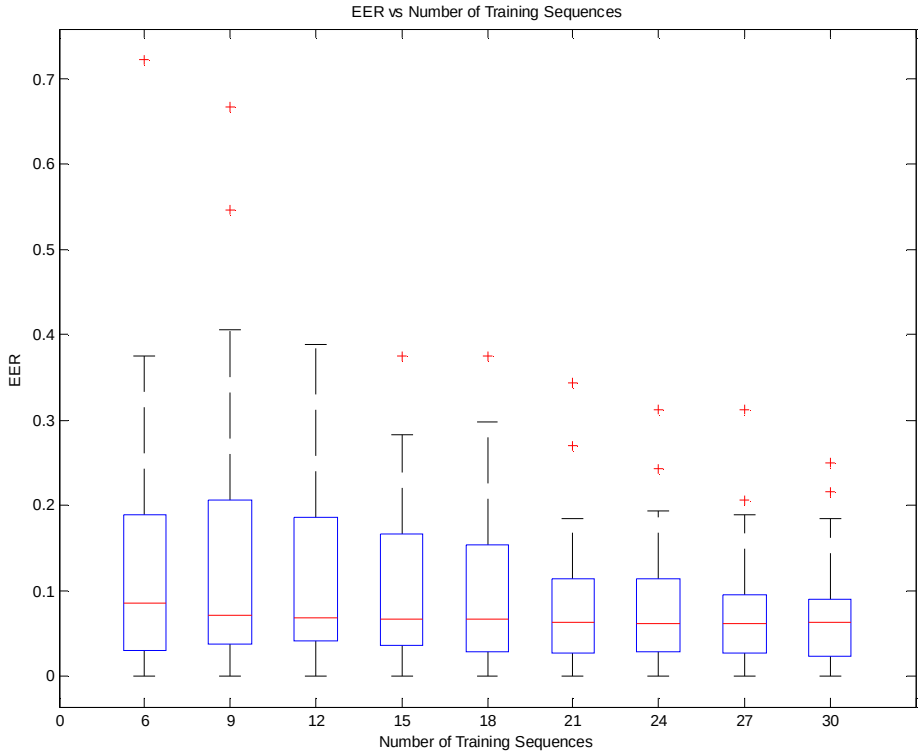

Figure 4. 2: Boxplot of EER vs. Number of Training Sequences for Short Passwords

As shown in Figure 4.2, in the short passwords datasets, these boxes have fairly similar median values. In each dataset, around half of the equal error rates are greater than 7%. The inter-quartile range (IQR) is decreasing as the number of training sequences increases, indicating reduced variability of equal error rates. In addition, the maximum values of equal error rates are closer to the median in the later periods, and each dataset has a minimum error rate of zero. This boxplot shows that the more the training sequences, the better the performance, but as the upper quartile decreases from one period to the next, the lower quartile has a limit value of 2%.

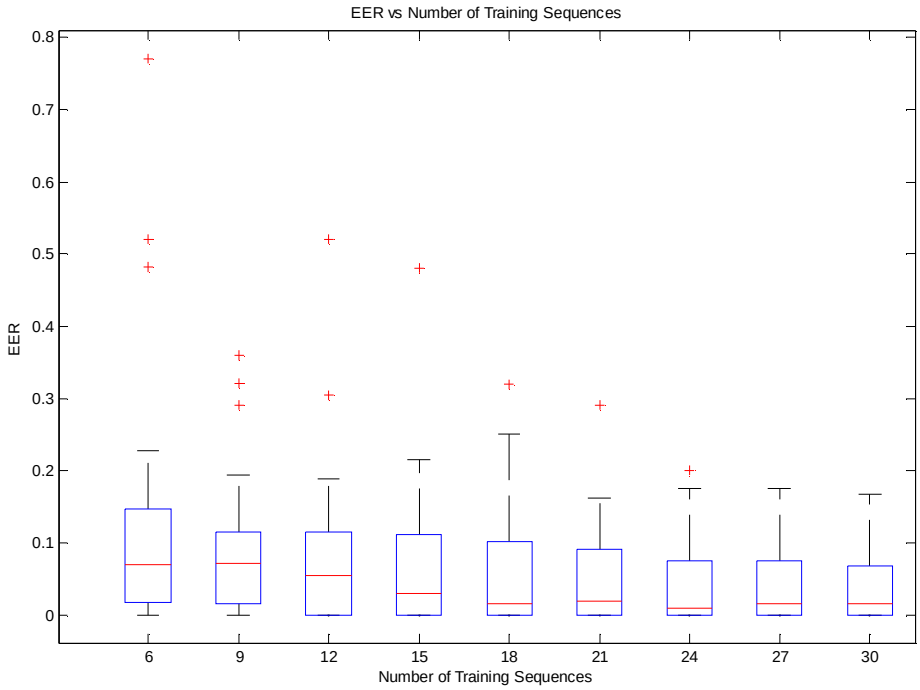

Figure 4. 3: Boxplot of EER vs. Number of Training Sequences for Long Passwords

Figure 4.3 shows the boxplots representing the long passwords dataset. Both the median value and the inter-quartile range are decreasing significantly as the size of training set increases. Each experiment has a minimum error rate of zero. Almost half of users (14 out of 30) have achieved an EER of 0% and all the EER's are less than 20% after the first 27 sequences are included into the training process.

4.2 Performance Prediction

To forecast the performance of the system, we add trend lines to the chart in Figure 4.1. As we can see in Figures 4.4 and 4.5, for long password, the trend line function is

$$y = 0.1325x^{-0.5719}, \quad (1)$$

and for short password, the trend line function is

$$y = 0.1597x^{-0.3405}, \quad (2)$$

where y is the equal error rate and x is the number of training typing sequences.

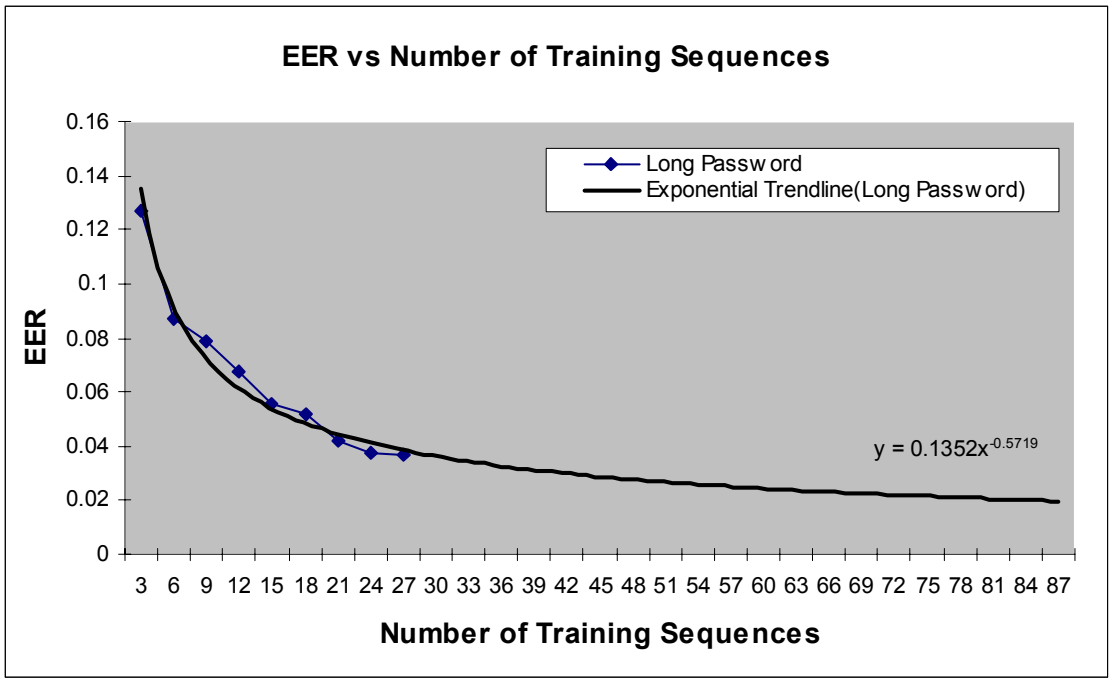

Figure 4. 4: Trendline of EER (Long Password)

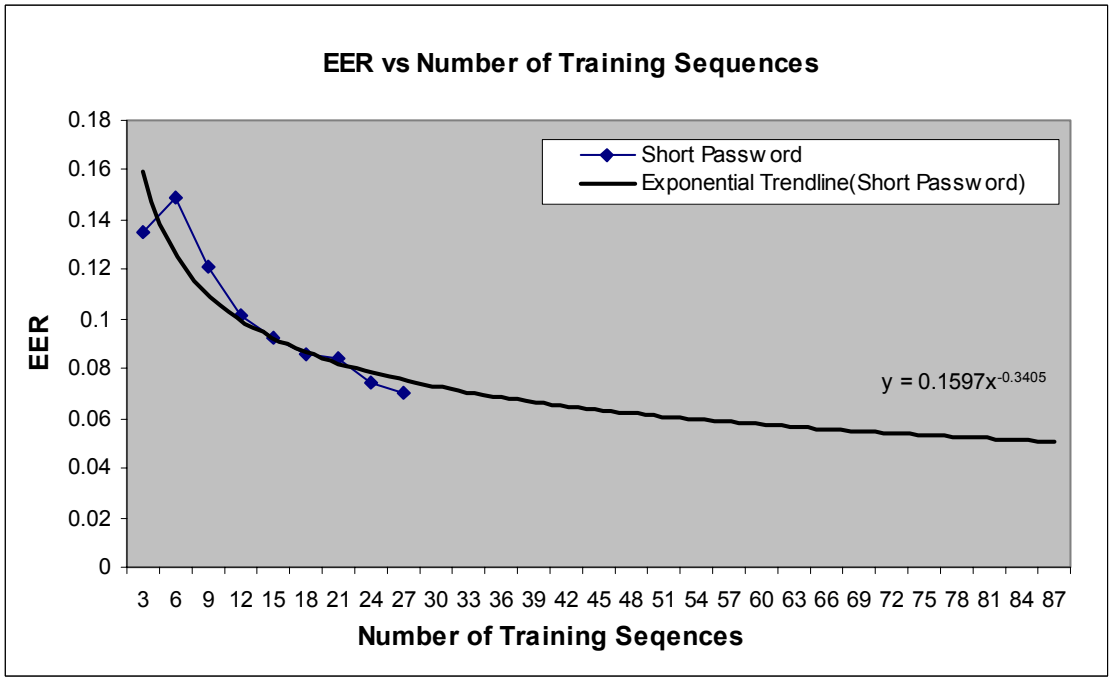

Figure 4. 5 Trendline of EER (Short Password)

Figure 4.6 shows the ROC curves for long password using different number of training sequences. The more training sequences used, the better the performance of keystroke dynamics authentication.

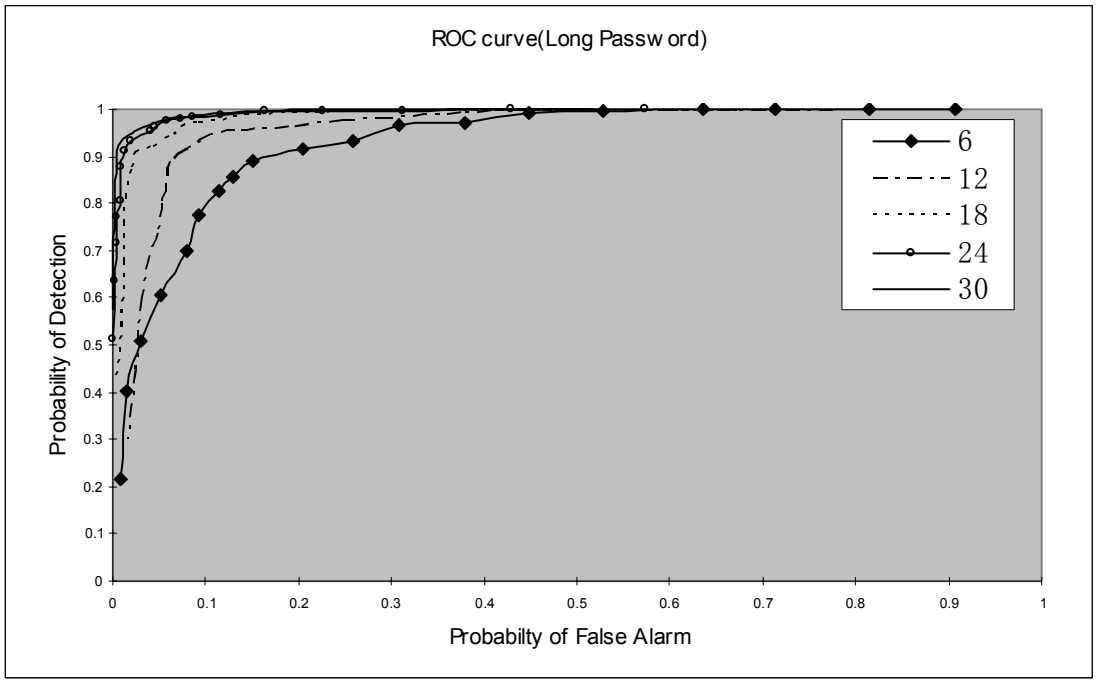

Figure 4. 6 ROC curves using different training sequences

Based on the false accept rates of the systems using increasingly long training sequences, we also generated the trendline(Figure 4.7) of FAR with a fixed FRR=0.01. The trendline function is

$$y = 0.5253x^{-0.7387}, \qquad (3)$$

where y is the false accept rate and x is the number of training sequences.

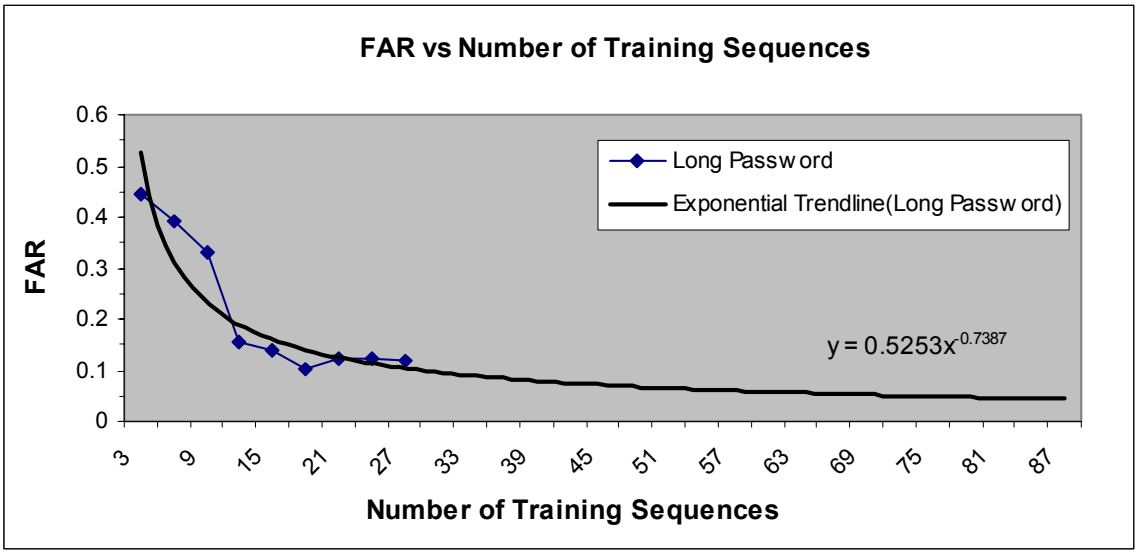

Figure 4. 7 Trendline of FAR(Long Password, FRR=0.01)

4.3 Discussion

In order to better understand the habituation process for keystroke dynamics classification, we conducted a chronologically incremental study and investigated the difference in performance between the eight-letter lowercase English passwords and the twelve-character randomly generated passwords which require shift-key behavior. Experimental results show that the performance obtained from the long and complex password sequences is dominating that from the short and simple password sequences.

What also should be noted is the trend of the decreasing error rate when more sequences are added into the training process. This provides an insightful perception of the habituation in keystroke dynamics. The habituation results give us confidence in the feasibility of deploying a practical and unsupervised keystroke dynamics system. Our study suggests that a user should choose a number of keystroke dynamics training sequences based on his/her application security specific needs. Moreover, we observed that with long shift-key passwords, the EER rates

begin to approach acceptable levels after only 10 sequences, which indicates a fairly robust password protection performance. However, an arguably low level of user-friendliness associated with long shift-key passwords definitely impedes their adoption in real-world applications. This motivates us in investigating further the security requirements and design for password-based authentication schemes and exploring new approaches to achieving high level security by leveraging the superior performance of keystroke dynamics with long and complex passwords.

Chapter 5 The Two Passwords Authentication Scheme

In this chapter, we investigate the password hardness measured by the probability of a password being cracked through guessing attacks of existing single password mechanisms. The trade-off between user's ability for memorizing the passwords and password security has become an inevitable dilemma for single password authentication systems. Based on our findings, we propose a novel two password mechanism to strengthen the password hardness.

5.1 Password Hardness

Password hardness is commonly used as a measurement of password security. In general, hardness can be viewed as the ability of a password to withstand the attacks, which mainly take the form of "repeated guessing". Thus, in the context of this study, we measure the password hardness by computing the password guessing attack metric in the form of a probability.

5.1.1. Probability of a Password Being Guessed (P)

According to U.S. Department of Defense password management guideline [29], the probability of a password being guessed can be estimated and managed by the following equation:

$$P = \frac{LR}{S}, \qquad (4)$$

where L is the maximum password lifetime, R is the login attempt rate, S is the size of the password space, and P is the probability that a password can be guessed through guessing attacks in its lifetime.

5.1.2. Password Space (S)

Password length and alphabet size are factors in computing the maximum password space requirement. The following equation expresses the size of the password space:

$$S = A^M , \quad (5)$$

where:

S = password space;

A = number of alphabet symbols;

M = password length.

From equation (4), we can see that as the lifetime of password increases, the probability of a password being guessed increases.

5.1.3. Password Login Attempt Rate (R)

The password login attempt rate is estimated as the number of attacks per time unit (e.g. per month or per second) to a password during its lifetime. It is also named as the "password guessing rate". Various data has been disclosed in slightly different forms measuring the login attempt rate. In 1988, Clifford Stoll uncovered a hacker using a dictionary attack on encrypted passwords, and estimated time to crack with common hardware was one month. In 2000, Paul Bobby published "Password Cracking Using Focused Dictionaries", achieving 48,000 guesses per second, while dictionary attacks only take seconds and full eight characters cracking needs 4765 years. In 2005, a G5 running at 2.7 GHz with a highly

optimized copy of John The Ripper hit 900,000 guesses per second and full eight characters cracking in this case take 200 years [30].

The magnitude of the password guessing rate is appalling. In order to avoid brute force attack [10], which is the most common and effective form of password guessing attacks, we can set limits on the number of login or other attempts before terminating the connection or process to control the guessing rate, as suggested in the password management guidelines by U.S. Department of Defense [29]. Thus, some password authentication procedures are intentionally made slow for the sake of forcefully decreasing the login attempt rate, given the fact that a legitimate user would rarely complain if the login process takes a slightly longer time, normally 1 to 2 seconds.

5.1.4. Password lifetime (L)

The greater the length of time during which a password is used for authentication purposes, the more opportunities exist for exposing it. In a useful password system, the probability of compromise of a password increases during its lifetime. The longer the same password is in use, the greater the likelihood that someone gets to know it or can guess it. For an initial period of time, this probability could be considered acceptably low. After a longer period of time, it would be considered unacceptably high. At the latter point, use of the old password should be considered suspect rather than a reliable proof of identity. By appropriately limiting the length of time during which a password can be used, the vulnerability of the password can remain acceptable. To protect against unknown threats, the U.S. Department of Defense recommends that the maximum lifetime of a password be no greater than 1 year.

Changing the password frequently adds another layer of security. In many computer systems, as part of the effort to improve computer security, it is required that a user should always change his/her password timely and periodically.

5.2 The Trade-off Dilemma

In general, the hardness of a password can be improved by increasing the complexity of the password and lengthening the password. However, the increasing complexity and length of passwords raises issues in password acceptance and user experience. Figure 5.1 illustrates the trade-off between the user experience and password security. Nowadays, as the security requirements on passwords become more and more demanding, the state of the art "hard" password would require a user to provide a password with no less than eight characters and a combination of capital cases, symbols, numbers, non-dictionary words, etc, all of which seems almost as complex as a randomly generated password. Yet at the same time, many users would admit that typing in such a password is as much a nuisance as creating one.

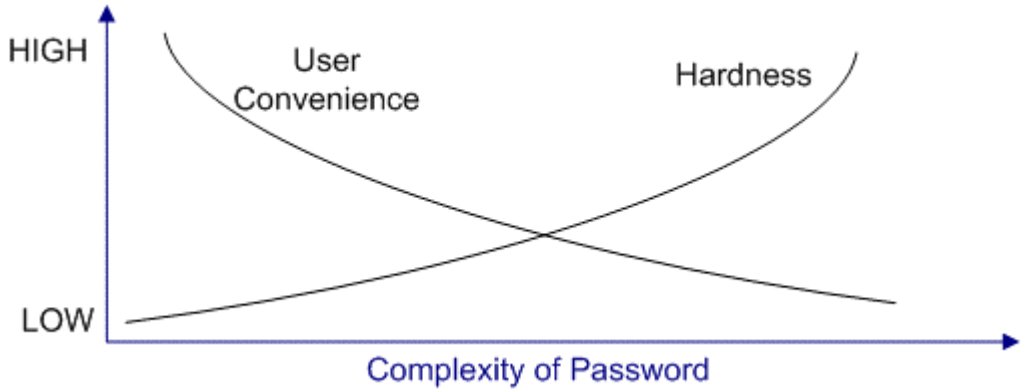

Figure 5. 1Trade-off between user experience and password security

Statistical studies also confirm the dilemma point. “Soft” passwords come with significant vulnerability and “hard” passwords come with a price in user experience. In his report [31], Daniel V. Klein gathered 25,000 Unix passwords, and found that 21 to 25 percent of password could be guessed, depending on the amount of effort put in [31]. Dictionary words accounted for 7.4 percent, common names for 4 percent, combinations of user and account name 2.7 percent, and so on down a list of less-probable choices such as words from science fiction (0.4 percent) and sports terms (0.2 percent). It seems that many users did not put much effort to create a reliable password. People have trouble remembering arbitrary digits and keystrokes, so they are willing to choose a password which is easy to remember and yet which is, in a sense, easy to be guessed by dictionary attack or brute force attack.

Some experts believe that we can improve security by restricting the password people can use. For example, use an automatic password generator to produce random passwords. Our results also show that the keystroke dynamics classification using short and simple passwords is not as effective as that using long and complex passwords. Therefore, we would prefer to use the password that at consists of 12 randomly generated characters. Unfortunately, such passwords will be extremely challenging for most users to memorize. Users might be compelled to write the password down in order to help themselves to remember it, whereby giving the attackers opportunities to physically steal the password. It is evident that with the single password mechanism, the trade-off dilemma is unavoidable.

5.3 Two Passwords Mechanism

Where single password mechanism fails, we find that two-password mechanism can succeed. We propose a two password mechanism system to take advantages of both simple and complex passwords, i.e. simple and short password is more user-friendly and complex and long password provides better security. Figure 5.2 depicts the two passwords mechanism.

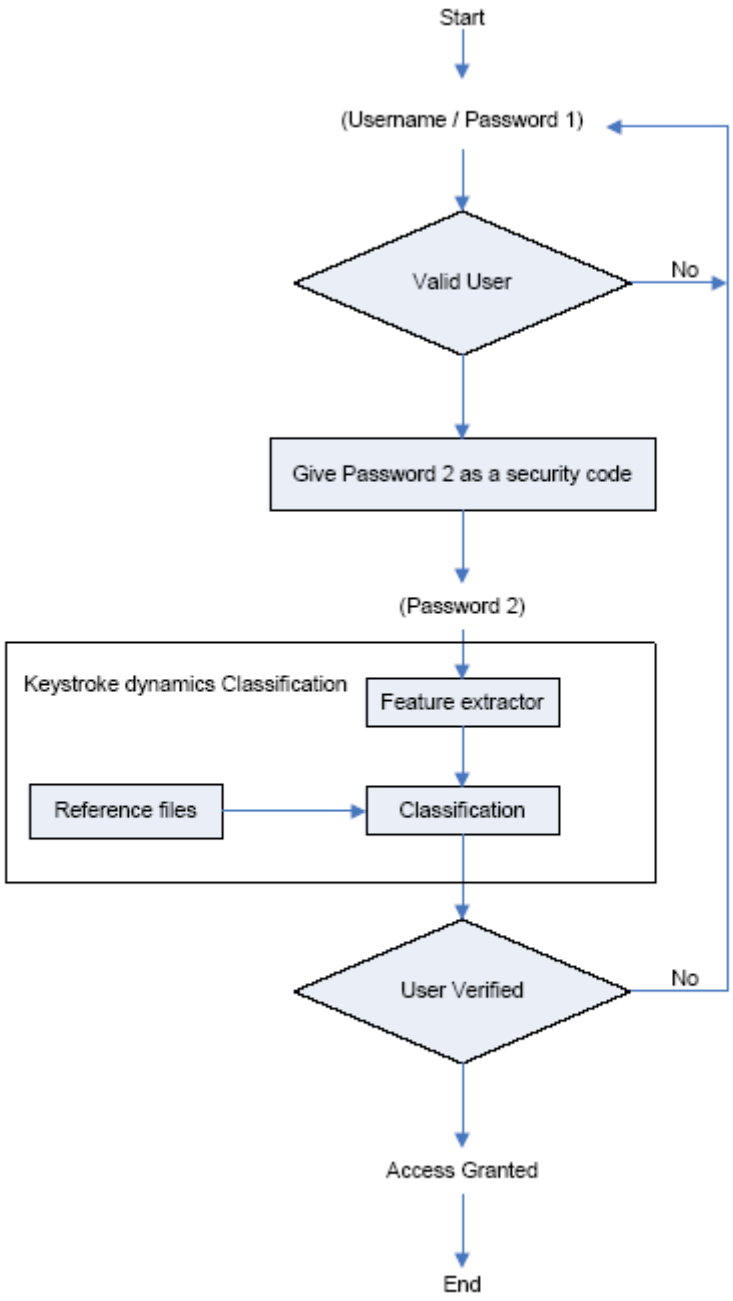

Figure 5. 2: Two Passwords Mechanism

As shown in the block diagram of the overall system in Figure 5.2, we propose two stages of user authentication. Stage one uses the traditional username/password scheme. Users login with their username and the first password, namely Password 1. After the user passes the first stage, the system will then provide a long sequence which contains upper and lower case characters and random symbols, called Password 2. The user types it and the key-up and key-down time is recorded. The system will use the data to analyze the user's keystroke dynamics behavior for verification. This completes the second stage of the two passwords mechanism.

The first stage is based on commonly used username/password scheme. The password is generated by users themselves when a new account is created. As mentioned in Section 5.2, the user-generated password is usually not very secure. The longer this password is used, the higher probability it will be cracked by guessing. Thus, it needs to be replaced when the probability of being cracked reaches the security lever. Figure 5.3 depicts the typical relationship between a password replacement policy and the probability of password cracking.

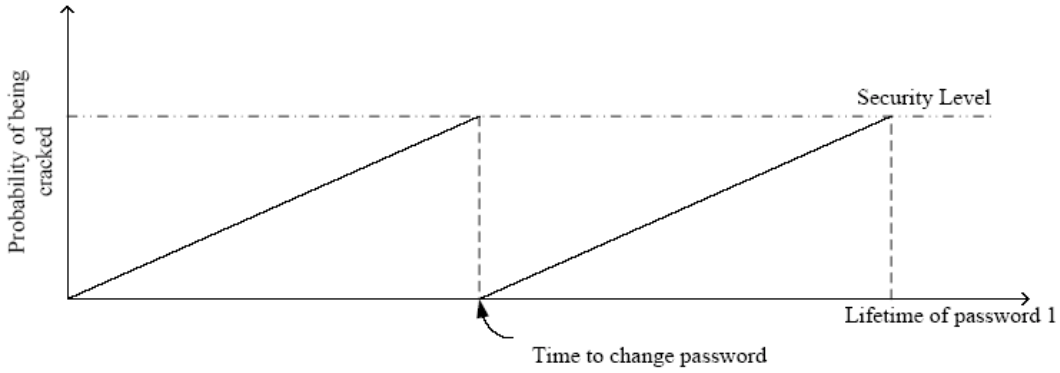

Figure 5. 3: Password replacement policy

In the second stage, the system would provide the user a long password, similar to the ones we used to test the user's keystroke behavior. To utilize keystroke dynamics effectively, this password would be set and changed less frequently. In Chapter 4, we demonstrated that the false accept rate of keystroke dynamic classification decreases as the users type the same password over and over again, as shown in Figure 5.4.

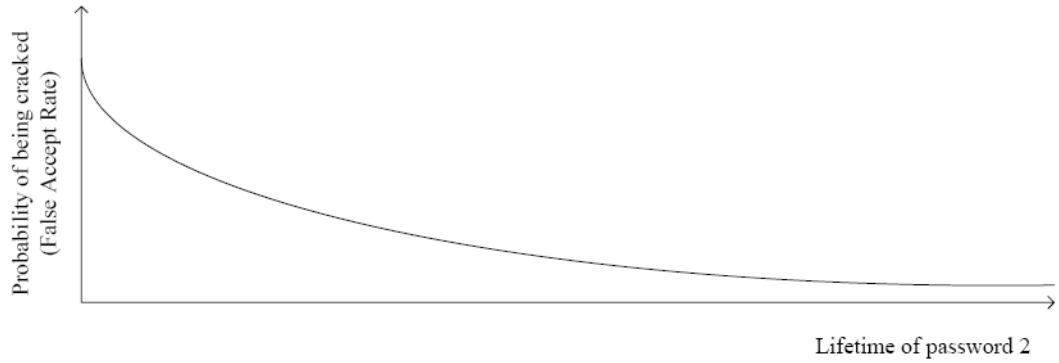

Figure 5. 4: False Accept Rate VS Lifetime of Password 2

In the proposed two-password authentication scheme, the probability of passing both verifications by an impostor depends on the probability of Password 1 being guessed and the false accept rate of keystroke analysis in Password 2. We can mathematically formulate this relationship as follows.

Prob (Impostors login) = Prob (Password 1 being guessed) * FAR

Due to its user friendliness, Password 1 might be changed frequently. Thus, a meaningful keystroke dynamics pattern would be hard to obtain from Password 1. In the light of this, users would need to type the first several sequences of Password 2 as keystroke training set, till FAR

reaches the desired security level. The steps to establish a reliable user habituation are as following.

1. To create a new account, the user generates his/her username and Password 1. Password 2 is provided by the system. Every time the user logins into the system, he/she is asked to enter username and Password 1 for verification. The keystroke information of typing Password 2 is collected by the system as model training.

2. After the system collects enough training data, the keystroke analysis verification process is turned on. Only the correct keystroke patterns will be accepted. Once the system finds any suspicious activity that tries to crack Password 2, it immediately locks the account and informs the genuine user to change Password 1. For better security, the new Password 2 is also prompted to the user. Afterwards, the system continues its learning and classification process using new data collected over the new Password 2.

3. To further reduce the probability of being guessed, it is required that Password 1 be changed once the probability of being guessed reaches a certain threshold value. This probability calculation would be continually updated by the system and it would guide Password 1 replacement process.

The model which indicates the authentication system resilience using the two-password scheme is shown in Figure 5.5.

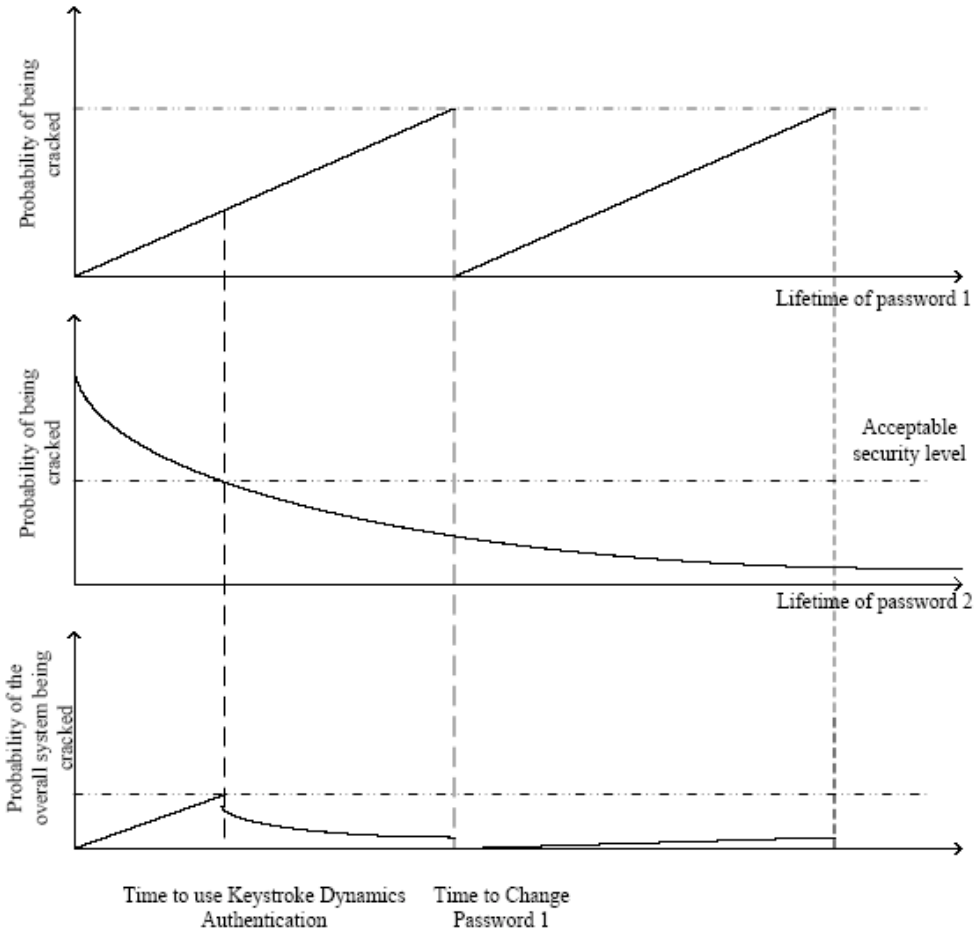

Figure 5. 5: Probability of the overall system being cracked

5.4 Experiments

Assume that most users would prefer to use short password which consist of 8 lower case letters as Password 1. The space of this password is then 26^8. Furthermore, assuming the password attacking rate is 1 attack per second, i.e. 3600*24 attacks per day, according to equation (4), the probability of this password being guessed can be computed as

$$P_1 = \frac{LR}{S} = \frac{3600 \times 24}{26^8} L . \qquad (6)$$

where:

P_1 = Probability of Password 1 being cracked;

L = Lifetime of Password 1(or existed time);

S = Password space;

R= Password attacking rate (1 attack per second).

Now suppose the user changes his/her password once per year. We compute P_1 using this frequency and then obtain the trend of P_1 over time as shown in Figure 5.6.

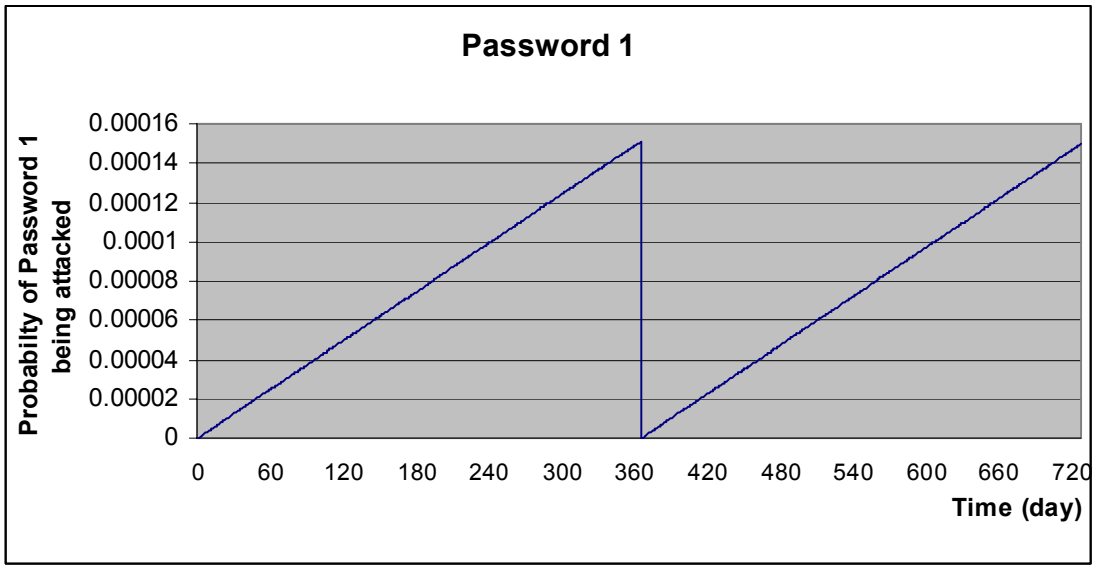

Figure 5. 6: Probability of being guessed vs. password lifetime (Password 1)

At the same time, let us assume the user types the long password, Password 2, once a day to let the system collect the keystroke information. From Chapter 5, we know the trend function of False Accept Rate of the long password is $P_2 = 0.5435(L \times r)^{-0.7798} = 0.5435L^{-0.7798}$ (7)

where:

P_2= Probability of Password 2 being cracked;

L= Lifetime of Password 2(or time since being generated);

r = The typing rate of Password 2(one time per day).

This trend function is depicted in Figure 5.7.

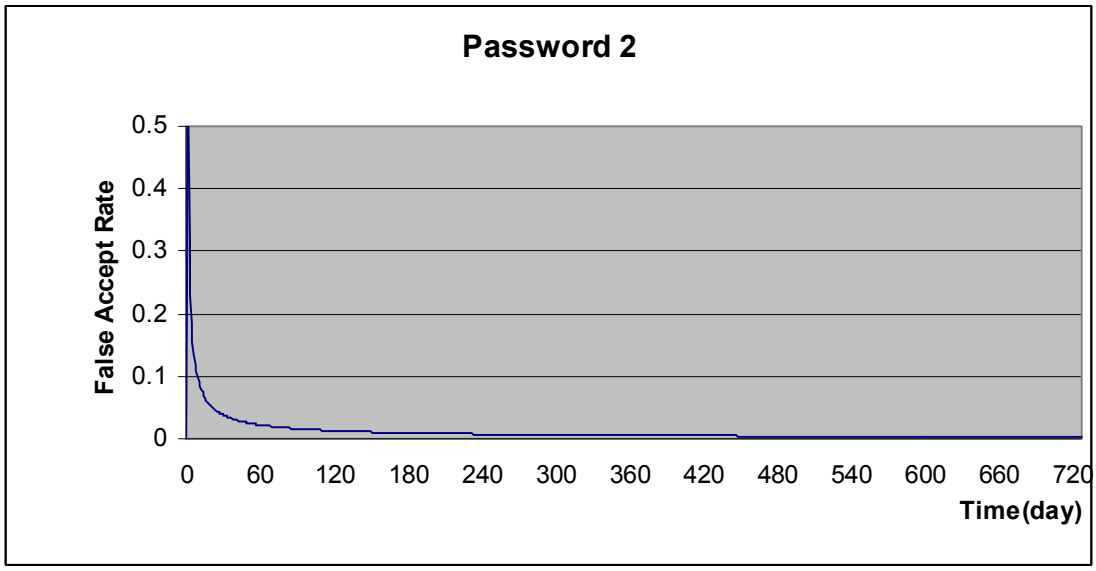

Figure 5. 7 FAR vs Password Lifetime (Password 2)

Thus, the probability of cracking the overall system is

$$P = P_1 \times P_2 = \frac{3600 * 24}{26^8} L \times 0.5435 L^{-0.7798}, \qquad (8)$$

Figure 5.8 shows the trend of P over time.

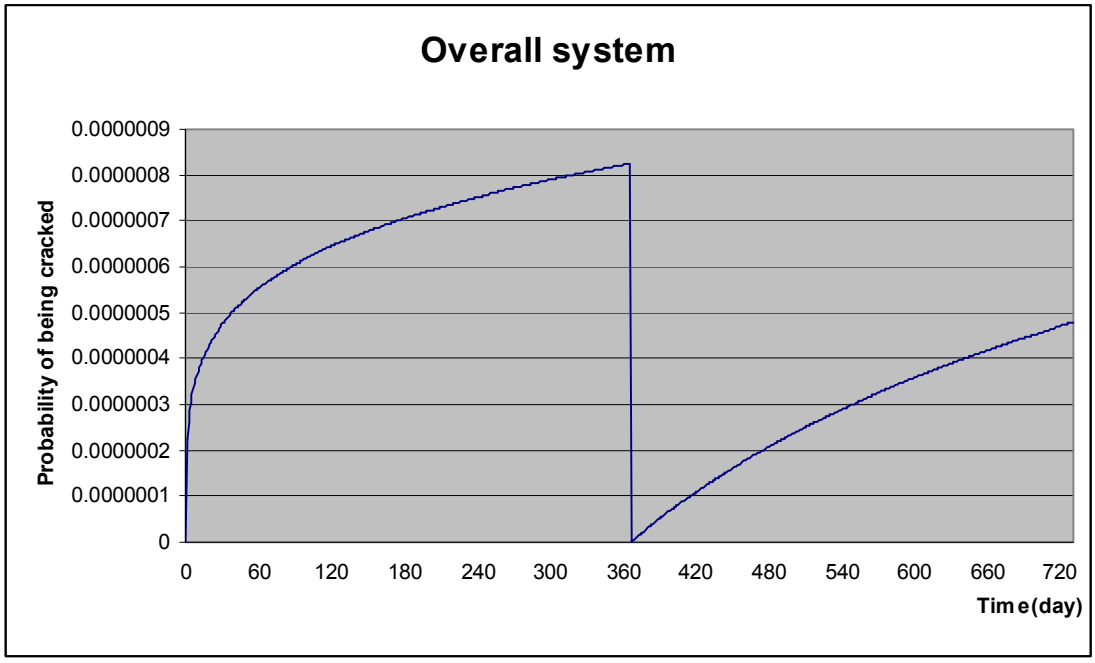

Figure 5. 8 Trendline of the probability of being cracked in the overall system (Password 1 + Password 2)

Based on this example, we can see that when a new user enrolls into the system, Password 2 that incorporates keystroke dynamics analysis might be vulnerable to attacks due to the acclimation process of user habituation. However, at this point of time, Password 1 is difficult to crack, which ensures the security of the system and protects Password 2 from attacks. Later on, although the probability of Password 1 being cracked would increase over time, the performance of the keystroke dynamic system reaches a relatively high security level and the hardness of Password 2 would be reinforced through continuous learning. This way, Password 1 and Password 2 complement each other, and a balance between user experience and password security is thus achieved by the two-password mechanism.

Chapter 6 Conclusions and Future Work

The increasingly high level of security demanded from password based user authentication systems aggravates the challenge of password protection. There is a wide agreement that increasing the complexity and length of a password will harden a password yet not necessarily be effective enough to survive attacks such as shoulder surfing, Trojan horses, and viruses. Keystroke dynamics has been proposed in previous studies to leverage user's typing patterns to fortify password security. As a promising biometric technology, keystroke dynamics provides an advanced layer of personalized security that has been proven to be able to successfully defeat many attacks, which conventional password protection mechanisms fail to vanquish.

The key to a reliable keystroke dynamic system is the user's habituation. An in-depth analysis of users' typing patterns as well as the forming process of such patterns are crucial to the performance of the keystroke dynamics based classifier that is built upon these patterns. We therefore designed a study analyze the typing habituation process. Based on the results obtained from this study, we investigated an extension of conventional single password mechanism. In an attempt to solve the trade-off dilemma between user experience and password security, we proposed a novel two passwords mechanism for advanced password protection incorporating keystroke dynamics.

The conclusions we draw from our study follow below.

6.1 Conclusions

We obtained the results of varying training set sizes over a wide range for two different types of passwords from our empirical study of habituation process, we used short and simple 8 letter words and long and complex 12 characters random phrases as pass phrases. The algorithm we used for classification is Breiman's Random Forest[8], which has demonstrated superior performance for keystroke dynamics classifications [28].

We first identified the impact of varying the training set size on the classification error rate. The equal error rate drops as we increase the amount of training data. Comparing the trend of error rates, the long password sequences performed significantly better across numerous tests with different training set sizes. This supports our conclusions that, 1) long and complex passwords are more effective when employed in keystroke dynamics systems; and 2) there is a habituation and acclimation process needed for the user to obtain a stable keystroke pattern.

Based on the insights we gained from the keystroke dynamics user habituation study, we proposed a two passwords mechanism that first adopts an easy-to-memorize password for password verification, followed by a fixed long and complex phrase that utilizes the features of the user's keystroke patterns for classification. The proposed mechanism attempts to improve password security without sacrificing much of user experience. Our analysis shows that the deployment of keystroke dynamics in this system should significantly decrease the probability of

authentication break-down. Experiments demonstrate that, with proper keystroke dynamics design, the two-password mechanism can significantly improve the usual login-password authentication.

6.2 Future Work

By its very nature, keystroke dynamics is less costly than other biometrics and can take place in a remote access point. The improvement of both performance and user experience in keystroke dynamics based password authentication offers a great opportunity to its wide adoption in real-world applications. We envision future improvements to be made in the following aspects.

Number of Samples: The number of samples collected in our experiment varies greatly. Some users did not provide sufficient samples to establish a reliable classification template. This can be problematic when the sample size is relatively small. In the future, we plan to address this issue by increasing our user population and the number of samples of each user.

Features: The features used in our experiment are mainly based on statistical data, such as the average time of key intervals and latency. The sequences of latency time and duration time of each string can also be considered as feature vectors in future work.

Mechanism: An adaptation mechanism could be performed to maintain and improve keystroke matching. Every time a successful authentication

is performed, we could replace the oldest samples from the training dataset with the newly obtained ones and rebuild the classification model.

Algorithm: It is well known that every algorithm has it bias. We envision to design and develop a hybrid approach or a combination of algorithms to avoid the bias problem and improve the classification performance. In a preliminary study, we developed a fusion model by combining different algorithms. The following is an example of the fusion model for one user. The algorithms used are Random Forest and Naïve Bayes with two different feature selection methods: Consistency Subset Evaluation and Correlation-based Feature Selection. From Figure 6.1, we can see the ROC curve of the 2 out of 3 decision level fusion system is significantly better than that of each single algorithm.

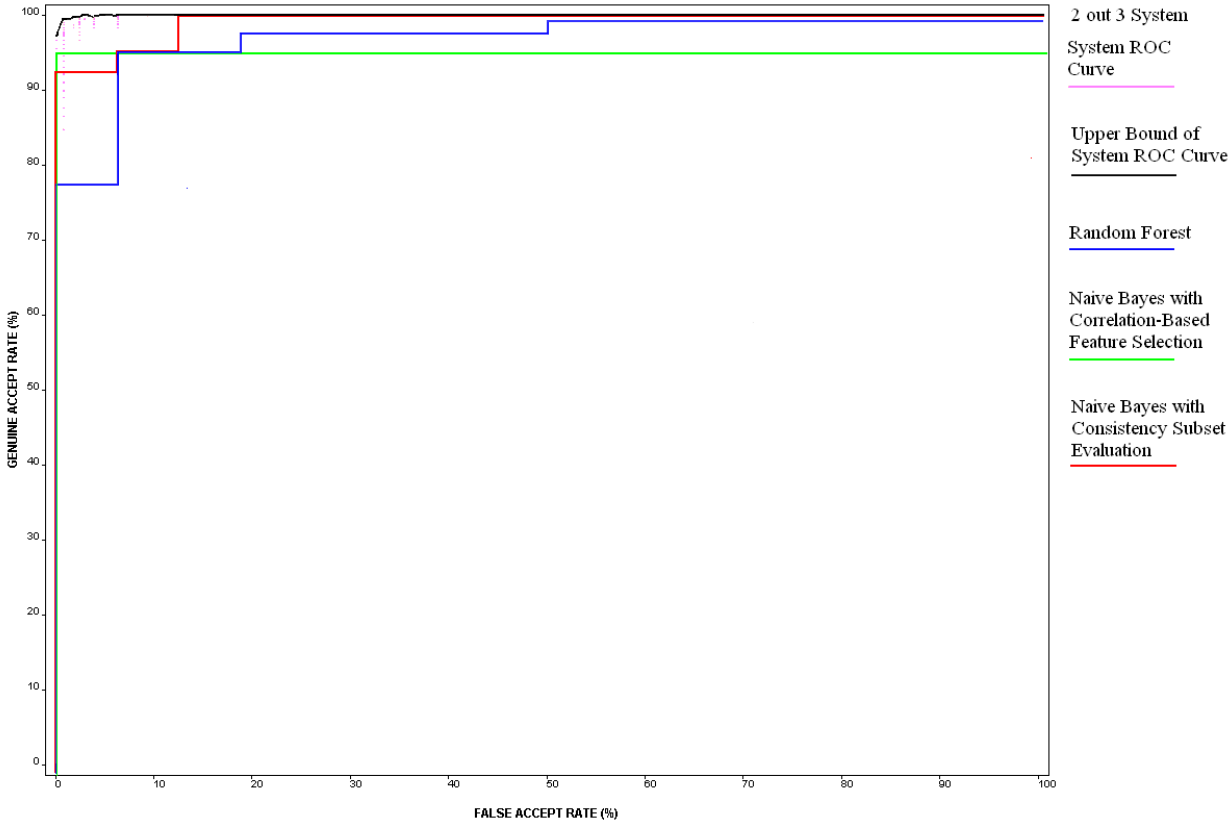

Figure 6. 1: An Example of Decision Level Fusion

Reference

[1]. Yun, Yau Wei. *The '123' of Biometric Technology*. 2003. Retrieved from on November 21, 2005 from the World Wide Web

[2]. Anil K. Jain, Arun Ross, and Samil Prabhakar. *An introduction to biometric recognition*. IEEE Transactions On Circuits And Systems For Video Technology, vol. 14, no. 1, pp. 4-21, 2004

[3]. http://www.bioid.com/sdk/docs/About_EER.htm

[4]. W. Frawley and G. Piatetsky-Shapiro and C. Matheus. *Knowledge Discovery in Databases: An Overview*. AI Magazine: pp. 213-228. ISSN 0738-4602, Fall 1992

[5]. Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer ISBN 0-387-31073-8, 2007

[6]. Ho, Tin Kam. *Random Decision Forest*. Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition, Montreal, Canada, August 14-18, 1995, 278-282,1995

[7]. Ho, Tin Kam. *The Random Subspace Method for Constructing Decision Forests*. IEEE Trans. on Pattern Analysis and Machine Intelligence 20 (8), 832-844, 1998

[8]. Breiman, Leo. *Random Forests*. Machine Learning 45 (1), 5-32, 2001

[9]. Ramon Diaz-Uriarte and Sara Alvarez. *Variable selection from random forests: application to gene expression data*. http://www.citebase.org/abstract?id=oai:arXiv.org:q-bio/0503025. 2005

[10]. Ciaramella, A., D'Arco, P., De Santis, A., and Galdi, C. *Neural Network Techniques for Proactive Password Checking*. IEEE Trans. Dependable Secur. Comput. 3, 4 (Oct. 2006), 327-339. DOI= http://dx.doi.org/10.1109/TDSC.2006.53

[11]. Rick Joyce and Gopal Gupta. *Identity authorization based on keystroke latencies*. Communicationas of the ACM, 33(2):168-176, Februray 1990

[12]. D. Mahar, R.Napier, M. Wagner, W. Laverty. R.Henderson, and M. Hiron. *Optimizing digraph-latency based biometric typist verification systems: inter and intra typists differences in digraph latency distributions*. Int. Journal of Human-Computer Studies, 43:59-592,1995

[13]. Fabian Monrose and Aviel Rubin. *Authentication via keystroke dynamics*. Fourth ACM Conference on Computer and Communications Security, pages

48-56, 1997

[14]. R Gaines, W Lisowksi, W Press, and S Shapiro. *Authentication by keystroke timing: Some preliminary results*. Rand Report R-256-NSF, The Rand Corporation, Santa Monica, CA, 1980.

[15]. Bleha, S., Slivinsky, C., and Hussien, B. 1990. *Computer-Access Security Systems Using Keystroke Dynamics*. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 12 (Dec. 1990), 1217-1222. DOI= http://dx.doi.org/10.1109/34.62613

[16]. Brown, M. and Rogers, S. J. 1993. *User identification via keystroke characteristics of typed names using neural networks*. Int. J. Man-Mach. Stud. 39, 6 (Dec. 1993), 999-1014. DOI= http://dx.doi.org/10.1006/imms.1993.1092

[17]. Fabian Monrose , Aviel Rubin. *Authentication via keystroke dynamics*. Proceedings of the 4th ACM conference on Computer and communications security, p.48-56, April 01-04, 1997, Zurich, Switzerland

[18]. Leggett, J., Williams, G., Usnick, M., and Longnecker, M. 1991. *Dynamic identity verification via keystroke characteristics*. Int. J. Man-Mach. Stud. 35, 6 (Nov. 1991), 859-870. DOI= http://dx.doi.org/10.1016/S0020-7373(05)80165-8

[19]. Leggett, J. and Williams, G. 1988. *Verifying identity via keystroke characteristics*. Int. J. Man-Mach. Stud. 28, 1 (Jan. 1988), 67-76.

[20]. O. Coltell, J. M. Badia, and G. Torres. *Biometric identi-fication system based in keyboard filtering*. In IEEE International Carnahan Conference on Security Technology, Madrid,Spain, 1999.

[21]. Edward Cureton. *Factor analysis, an applied approach. Erlbaum Associates*, Hillsdale, N.J.,1983

[22]. Hocquet, S., Ramel, J., and Cardot, H. 2005. *Fusion of Methods for Keystroke Dynamic Authentication*. In Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies (October 17 - 18, 2005). AUTOID. IEEE Computer Society, Washington, DC, 224-229. DOI= http://dx.doi.org/10.1109/AUTOID.2005.30

[23]. J Garcia. *Personal identification apparatus*. Patent 4,621,334, U.S. Patent and Trademark Office, Washington, D.C.,1986

[24]. J R Young and R W Hammon. *Method and apparatus for verifying an individuals identity*. Patent 4,805,222, U.S.Patent and Trademark Office, Washington, D.C., 1989

[25]. M S Obaidat and D T Macchairolo. *An on-line neural network system for computer access security*. IEEE Trans. Industrial Electronics, vol. 40, no. 2, pp. 235-241, Apr 1993

[26]. S Bleha and M S Obaidat. *Computer user verification using the perceptron*. IEEE Trans. Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 900-902, May 1993

[27]. M S Obaidat and B Sadoun. *Verification of computer users using keystroke dynamics*. IEEE Trans. Systems, Man, and Cybernetics, vol. 27, no. 2, pp. 261-269, Apr 1997

[28]. Bartlow, N. and Cukic, B. 2006. *Evaluating the Reliability of Credential Hardening through Keystroke Dynamics*. In Proceedings of the 17th international Symposium on Software Reliability Engineering (November 07 - 10, 2006). ISSRE. IEEE Computer Society, Washington, DC, 117-126. DOI= http://dx.doi.org/10.1109/ISSRE.2006.25

[29]. *U.S Department of defense password management guideline*. Network Security Library:: Policy & Standards. Feb 20, 2000

[30]. http://geodsoft.com/howto/password/password_research.htm

[31]. Klein, Daniel V.*"Foiling the cracker": A survey of, and Improvements to, Password Security*. Feb 22, 1991, http://www.klein.com/dvk/publications/passwd.pdf

[32]. Thomas J. Alexandre. *Biometrics on smartcards: an approach to keyboardbehavioral signature*. In Second Smart Card Research & Advanced ApplicationsConference, 1996

[33]. Bassam Hussien, Robert McLaren, and Saleh Bleha. *An application of fuzzyalgorithms in a computer access security system*. Pattern Recognition Letters,9:3943, 1989