Keystroke Dynamics Authentication: A Survey of Free-text Methods

Arwa Alsultan[1] and Kevin Warwick[2]

**[1]School of Systems Engineering, University of Reading
Reading, Berkshire, UK**

**[2]School of Systems Engineering, University of Reading
Reading, Berkshire, UK**

Abstract

Current computer systems depend greatly on authentication methods in order to provide sufficient protection to the data handled by these systems. Rather than using the common username and password scheme which suffers from many security and usability limitations, we investigate in this paper the use of keystroke dynamics as a more useable authentication alternative. We focus on the research done on free-text keystroke systems and its ability to provide continual identity verification during the whole time that the user is using the system.

Keywords: *free-text, keystroke dynamics, authentication, identification, performance, survey.*

1. Introduction

The use of computer systems has proliferated at an unforeseen rate. They are now used in almost all aspects of our lives. This is a strong reason to protect them against illegal intrusions. However many computer systems use the simple username/password scheme for authentication, even though it suffers from the security-usability trade-off dilemma. Passwords can be guessed using different methods such as social engineering, spyware, dictionary attack and mere brute force attacks. These are all reasons for the user to employ extreme measures to safeguard his/her computer by using long and complex passwords which are unfriendly and hard to memorize. It is therefore ideal to use an alternative authentication method that can be low-cost yet provide ease of use and transparency to the user in addition to security robustness.

Keystroke dynamics is a behavior biometric scheme that provides sturdy system protection while maintaining a high level of usability. In particular, using free-text keystrokes provides real-time identity verification by continuously monitoring the keyboard's activities. This is a very important, yet frequently ignored, part of the authentication process since it is fairly simple to establish a level of confidence about the user's identity at log-in time. However there is no guarantee that the user who was successfully authenticated is the same person who is still using the system. There is always a chance that the system was left unattended which is a golden opportunity for the attacker who is physically close to the machine to have access to it and, for example, alter some documents or send an e-mail on behalf of the original user.

In this method of authentication, it is not obligatory to memorize any text such as a password or a passphrase; instead authentication is conducted through finding the resemblance of the typing rhythm of a user, in a non-intrusive manner, regardless of the text typed.

One important fact in looking at research to date in free-text keystroke systems is that results from most studies are far from ideal, i.e. either the resulted accuracy is not satisfactory or it has a high accuracy level which was obtained under strictly controlled conditions, which is not at all representative of real-life situations. Thus, we aim in this paper to look at the various factors that might affect the authentication system performance in addition to covering the methods used for feature extraction and classification. Situations where free-text keystroke dynamics are best used are also discussed in this paper.

The rest of this paper proceeds as follows. Section two introduces keystroke dynamics theory and describes the differences between fixed-text and free-text systems. The third section lists some of the techniques followed for feature extraction while the section after that lists the methods used for classification. Performance measurement schemes are considered next. After that we list some of the factors affecting performance in free-text systems. A variety of applications that can benefit from free-text systems is given in the seventh section. Finally we discuss the level of protection that free-text systems can provide against some of the common security threats.

2. Keystroke Dynamics

Monitoring keystroke dynamics is considered to be an effortless behavioral based method for authenticating users which employs the person's typing patterns for validating his/her identity. As mentioned in [1], keystroke dynamics is "not what you type, but how you type." In this approach, the user types in text, as usual, without any kind of extra work to be done for authentication. Moreover, it only involves the user's own keyboard and no other external hardware. The original idea of using keystroke patterns for user identification purposes was originated from the idea of identifying the sender of Morse code on a telegraph machine, where operators have been able to identify the sender of a message by the rhythm, pace and syncopation of the received taps [2].

As early as 1980, researchers such as Gaines et al. [3] started to show interest in proving the hypothesis that typing patterns can be used as a mean of user authentication. Experiments were conducted to find typing patterns that can be used effectively for authentication. Results from these tests showed that the similarity between typing samples from the same person is high with respect to the time delays it takes the user when typing one key or two successive keys. All of this early research though was only concerned with keystrokes generated by typing fixed words.

It wasn't until 1995 when Shepherd et al. [4] showed interest in continuous authentication. In 1997, the first organized attempt to use free-text keystroke system was conducted by Monrose and Rubin [1] where both fixed-text and free-text were used. The overall performance was not encouraging for free-text giving only 23% correct classification while fixed-text produced about 90% correct classification. This shows the complexity of using free-text systems compared with the fixed-text systems. Nevertheless, free-text systems have gone a long way since that experiment and much better results have been obtained using more sophisticated techniques.

There are two main phases that a user has to go through in order to be authorized by keystroke dynamic systems; namely: the enrolment phase and the log-in phase. The first phase has to do with collecting data about the user such as username and password in addition to capturing the user's typing behavior. The system gathers the keystroke times and extracts the timing features to create a template for each user's typing behavior. This template, also referred to as a user's profile, is stored in a database in correspondence to the user's other details.

The second phase takes place whenever the user needs to actually use the system. At that time, the system collects the user's keystroke times and then extracts the timing features in the same manner pursued in the enrolment phase. After that, the system performs feature matching with the user's template which is stored in the database. Next, based on the results of the matching process, one of two actions will take place: granting access to the user if the two sets of data are sufficiently similar or denying access to the user otherwise.

Two types of keystroke systems are used and discussed in the literature; they are: fixed-text and free-text keystroke systems. Fixed-text, also referred to as static, obliges the users to use only a predefined text to produce the typing samples. The predefined text varies in the research done in this area in the way that some have utilized the same shared password for all users [5] and others used different fixed text for each user such as using the user's name [6] or log-in IDs [7]. The main function of the fixed-text systems is applying it at log-in time in order to verify the user's identity at the beginning of the session only. This is done by forcing the user to retype their password a number of times at the enrolment phase in order to determine the user's typing rhythm for that specific password. This is considered a critical usability issue because of the amount of burden it adds on the user; still, the user needs to memorize the predefined text. Generally speaking, fixed-text keystrokes are mainly used for password hardening.

Free-text systems, also known as dynamic, don't restrict users to a particular text; on the contrary, they are given complete freedom to use any text of any length without any constraints. Unlike fixed-text, free-text systems will continue to collect the keystrokes, after successfully passing the log-in session, throughout the whole time that the user is logged-in for the reason of assuring the identity of the user during the full duration of that session. In free-text systems, the user's typing pattern is typically monitored during several days where he/she is performing regular typing tasks such as writing e-mails or typing word documents i.e. the enrolment phase is long yet transparent to the user. Even though, free-text and fixed-text systems are quite similar in the way that they both utilize the key press and release times to build a user behavior profile, they clearly differ in the way that the system is trained and applied.

All keystroke dynamics studies involve conducting five main experiment parts in the following order: recruiting participants, requesting a typing task to be done by the participants, collecting the keystrokes timing data, obtaining timing features from the raw keystroke data, training the classifier using part of the keystroke data and using the other part for testing the classifier [8]. We will go through the previous mentioned stages in order to compare and contrast what has been done in this area as reported in the current literature.

3. Feature Extraction and Profile Creation

The manner in which user data is collected in free-text keystroke systems is quite different from that of fixed-text systems in the way that a user is normally monitored for a period of time, a number of days for example. From all the typing data collected during this time, the system infers the typing pattern that the user typically follows which will be then stored as the user profile. The time it takes to type single letters or combinations of letters i.e. di-graphs, tri-graphs, even longer combinations is considered in free-text keystroke systems, yet there is a condition for including a particular letter or combination of letters in the template. It has to be typed often enough during the enrolment phase which will cause its mean and standard deviation to be statistically sound [9].

This implies that it is not necessary to include all letters and letter combinations, typed during the enrolment phase, in the template. Therefore, much research includes a pre-processing stage for removing noise from the data set. Extreme duration or latency values, i.e. very small or very large outliers, are discarded; for example: only the durations and the latencies of keys for which the standard deviation was below a predefined value were added to the user's template in [10, 1] while minimum and maximum values were fixed for the latencies that were used in [11].

Timing features are basically calculated using the press and release times of every key the user types and then processed in a specific way before being stored in the user's profile. Different methods were followed to carry out this part of the system, as shown in Table1. Here we focus on some of the common methods used for feature extraction and profile creation in free-text keystroke systems.

First we go through some of the simple feature extraction techniques found in the literature. Profiles in [1, 23] consisted of the mean latency and standard deviation of each di-graph in addition to the mean duration and standard deviation of each individual key. While the profiles in [11, 10] only included the latencies' means and standard deviations for di-graphs that have occurred a minimum number of times. On the other hand, the down-down duration time of di-graphs was used in [12, 13, 14]. This was extended in [15] to include more n-graphs including di-graphs, tri-graphs, and other longer n-graphs.

Although, di-graph and tri-graph time has been used in plenty of research, Sim and Janakiraman [16] concluded from their several experiments that using di-graphs/tri-graphs is not a good discriminative between users when the actual typed words are not taken into consideration. This is because the context of the text that a particular letter is included in regulates the manner in which it is typed [9] i.e. the letter 't' has different duration in the word 'sentence' and 'question'. Therefore, di-graphs/tri-graphs are more effective for keystroke dynamics when using context-specific n-graphs.

A more structured feature extraction was followed in some research where the timing features were extracted for only a set of key pairs which helped to increase the number of the di-graphs that can be found and compared in both the training and the testing samples. This increases the stability of its mean and standard deviation, in addition to reducing the required computation time. This was done in [17] by dividing all keystrokes into four attributes: left hand side keys, right hand side keys, spacebar and backspace bar; then, creating 16 diagraphs using these attribute combinations.

A keyboard grouping technique was introduced in [18] for classifying the keys based on their location on the keyboard, which was divided into 8 sections; two left and right halves and then each half divided into 4 lines representing the rows of the keyboard. For example WM is represented as Left 2- Right 4. Moreover, only a fixed set of letters and two letter combinations were used in [9, 19]; these sets were chosen based on each letters frequency in the English language. Letters including E, A, T ... etc. and di-graphs including: AT, TH, HE … etc. are frequently found in English text, therefore, it is a good idea to use the mean and standard deviation of their duration and latencies in the user template which will increase its stability.

More complex features were also taken into consideration for the purpose of distinguishing users typing behavior. In addition to the usual key-press duration and di-graph's down-down (duration) and up-down (latency) times, other features were utilized in [19, 20, 21]; such as: typing speed, error rate, press-release ordering and the percentages of using special characters. Other features that capture the editing patterns of the user which includes the usage of specific keys i.e. Home, End, Backspace, Delete, Insert, shortcut keys , arrow keys … etc. were also used.

Another interesting feature was used in [22]; where all the commands executed in the first 10 minutes were collected. Although, it might seem irrelevant on first glimpse, an attacker is more likely to hurry to execute as many commands as he can on the victim's machine during the first few minutes. This shows an obvious change in the users habits which can be used to detect illegal intrusion.

4. Methods

After extracting the users' typing features and creating their profile templates has been completed, the classification process is performed to find the similarities and differences between the user's template stored at the enrolment phase and the sample provided during the session the system is being used. Similar to fixed-text systems, many methods have been used for classification in free-text keystroke systems; ranging from simple statistical methods to more complex pattern recognition and neural network algorithms. Moreover, an even more sophisticated combination of methods was used in some cases. This section highlights the major classification approaches used in the current literature. Please refer to Table1 for more details.

Simple statistical methods were used as a classification mean for typing behavior in several free-text keystroke systems studies. A variety of distance techniques have been used; Euclidean distance [18], weighted Euclidean distance [23], scaled Manhattan distance [9] and Bhattacharyya distance [24] were all utilized to find the level of similarity between samples. In addition, other statistical techniques were also used; decision trees were used in [22] while Kolmogorov-Smirnov Test (KS-test) was used in [13, 17].

One of the most cited free-text studies was that conducted by Gunetti and Picardi (P&G) [15] which depended on two measures, the first of which was the relative measure which was used to find the degree of disorder between the two samples. The second was the absolute measure which was used to calculate the absolute distance between the two samples. In both the relative and absolute measures only n-graphs occurring in both typing samples were considered. Even though the results were very good, the computational costs required to identify users was expensive because it needed to compare the test sample with all users' templates in the database which obviously makes it less scalable. Hu et al. [25] attempted to solve the scalability issue of P&G's method using the k-nearest classifier. In this approach, training samples were divided into clusters such that, every test sample was compared only with the samples of those users in the same cluster. Results for this modification revealed accuracy which compared well with that of P&G. Computation speed, on the other hand, proved to be 66.7% better.

A number of extensions have been carried out on P&G's method by Davoudi and Kabir. In [12] they combined P&G's method with a distance-calculating method that used histogram-based density estimation for each di-graph in order to find the probability density function of the di-graph's duration time. While in [26], they modified the relative distance in the P&G method by choosing the di-graph with the highest difference in duration between the two samples to compute the difference of its positions first. After that, it was removed from the two timing vectors, and then, the new vectors were sorted again. They also applied one further modification to P&G's method in [14] by adding a weight factor to the digraphs when computing the relative distance. This weight was defined as the ratio of the number of occurrences of this digraph and its standard deviation.

Table 1: Chronological list of free-text keystroke systems.

Study	*Features*	*Method*	*Subjects*	*Samples*	*Performance*
Monrose & Rubin [1]	Di-graph latency, key duration	Euclidian distance, probability score, weighted di-graph probability	31	-	23% accuracy
Gunetti & Ruffo [22]	Di-graph latency, executed commands	Decision tree	10	-	90% accuracy
Dowland et al. [11]	Di-graph latency	Mean, Standard deviation	4	-	50% accuracy
Gunetti & Picardi [15]	N-graph duration	Relative distance, absolute distance	205	765	0.005% FAR, 5% FRR
Villani et al. [20]	Di-graph latency & duration, key duration, typing speed, percentage of special characters, editing patterns	Euclidian distance, k-nearest neighbour	118	2360	99.8% - 44.2 % accuracy
Curtin et al. [19]	Di-graph latency & duration, key duration, typing speed, percentage of special characters, editing patterns	Euclidian distance, k-nearest neighbour	30	-	100% - 97% accuracy
Filho & Freire [30]	Di-graph latency	Simplified Markov chain model	15	150	41.6% - 12.7% EER
Janakiraman & Sim [24]	Di-graph latency, key duration	Bhattacharyya distance	22	-	100% - 70% accuracy
Buch et al. [34]	Di-graph latency & duration, percentage of special characters	Euclidian distance	36	650	100% - 98% accuracy
Hu et al. [25]	N-graph duration	Relative distance, absolute distance, k-nearest neighbour	36	36554	0.045% FAR, 0.005% FRR
Hempstalk et al. [21]	Di-graph latency, key duration, typing speed, error rate, P-R ordering	One-class classification	10	150	11.3% FAR, 20.4% FRR
Ahmed et al. [29]	Di-graph latency	Neural network	22	-	0.015% FAR, 4.82% FRR
Davoudi & Kabir [12]	Di-graph duration	Relative distance, absolute distance, histogram-based density estimation	21	315	0.015% FAR, 0.0025% FRR
Pilsung et al. [13]	Di-graph duration	Kolmogorov-smirnov Test	-	-	0.17% EER
Samura & Nishimura [23]	Di-graph latency & duration, key duration	Weighted Euclidian distance	112	-	67.5% - 81.2% accuracy
Bours & Barghouthi, [10]	Di-graph latency, key duration	Distance measure	25	-	79 – 348 keystrokes
Davoudi & Kabir [26]	Di-graph duration	Modified relative distance	21	315	0.08% FAR, 18.8% FRR
Davoudi & Kabir [14]	Di-graph duration	Weighted relative distance	21	315	0.07% FAR, 15.2% FRR
Park et al. [17]	Key-pair duration	Kolmogorov-smirnov Test	35	-	0.089% EER
Messerman et al. [38]	N-graph duration	Normalized relative distance	55	-	2.20% FAR, 1.84% FRR
Singh & Arya [18]	Key-pair latency	Euclidian distance	20	-	0.02% FAR, 0.04% FRR
Chantan et al. [28]	Di-graph duration	Bayes classifier	-	-	0% EER
Bakelman et al. [27]	Di-graph duration	K-nearest neighbour	20	200	4% EER
Bours [9]	Di-graph latency, key duration	Scaled Manhattan distance	25	-	182 keystrokes

Pattern recognition methods were also exploited in order to be used as a classification method for free-text keystroke authentication. For example: K-nearest neighbor was used in [27] and Bayes classifier was used in [28].

Ahmed et al. [29] used a feed forward multi-layer perceptron neural network system for the purpose of classifying users. Two neural networks were used; a behavior-modeling network and a detection network. The first used the di-graph's first and second keys press and release times to find the elapse time it took a user to press two successive keys. The second neural network used the di-graph's times and the matching output from the behavior-modeling network to estimate which user's typing patterns it represented.

5. Performance

Unfortunately, not only keystroke systems but all biometric authentication systems sometimes suffer from mistakes in the authentication decision. This is due to a number of reasons that has to do not only with the efficiency of the technique but also with the user himself or with his surroundings. First of all it is possible, yet not likely, that an imposter is mistakenly identified as the legitimate user if by chance the two persons typing patterns are close enough to the extent that the classification method fails to distinguish between them. Conversely, when one of the legitimate user's fingers slips off the keyboard and causes the typing pattern to change slightly, the user may not be successfully authenticated. Thus, it is important to have some metrics to exactly measure the error rate which will help to identify the performance level that can be expected and tolerated by that system's users.

A very simple way to measure the error rate was used in earlier studies; using the Accuracy measure which is the percentage of successfully authenticated attempts compared to the total number of completed attempts. This technique was adapted in [1, 27, 22].

The most frequently used error rates for inferring the performance of an authentication system are: the False Accept Rate (FAR), also referred to as the Imposter Pass Rate (IPR) and the False Reject Rate (FRR), also called the False Alarm Rate (FAR). The FAR is the percentage of impostors who have successfully gained access to the system while the FRR is the percentage of legitimate users who have been denied access to the system. These two error rates were used by the majority of free-test keystroke systems including [15, 21, 18].

Clearly, there is a trade-off between the FAR and FRR which can be controlled according to the level of security strictness required. FAR is required to be as low as possible in strictly secure applications while there is a compromise of having a higher FRR. Meanwhile, a higher FAR is acceptable in systems where security is not the major aim yet system usability has higher priority.

The other commonly used error rate is the Equal Error Rate (EER), also referred to as Cross-over Error Rate (CER), which is the value where FAR and FRR are equal. It was used in many methods such as [17, 30, 27] where lower EER values indicate a more secure system.

Due to the fact that free-text keystroke authentication is a continuous process, another metric which defines exactly how much time, in number of keystrokes, did it take the system to discover that an imposter had had access to the system has been proposed in some studies. This aims to detect the impostor as fast as possible, incorporating as few keystrokes as possible. This implies that an attacker would be detected before he can do more harm to the system. A penalty-reward technique was introduced in [9, 10] where a user was initially given the highest trust level prior to the user being successfully authenticated via a static authentication procedure. During the typing session, the user obtained a reward which he received in the form of an increase of his trust level when he typed in a manner sufficiently close to his typing template. Likewise, he obtained a penalty in the form of a decrease of his trust level when he typed in a manner far from his typing template. The system then locks-out a user if his/her trust level falls below a pre-determined threshold.

6. Factors Affecting Performance

There are many different performance measures used to determine the error rate in free-text keystroke systems, it is therefore often difficult to compare studies. This is also due to not having any form of standardization in the data collection process in these different experiments. Even though, the error rate in study A is lower than the error rate in study B, that does not necessarily mean that the method adapted in A is better than that used in B. Different factors may have a positive or a negative impact on the authentication process regardless of the actual method's functionality. Standardization of such factors requires information exchange amongst researchers which would offer an improved comparing mechanism between different algorithms. There are a lot of different factors to be considered in free-text keystroke systems; a detailed list of these factors is provided in this section.

Nevertheless, there are some solutions that can be used to standardize the factors involved. The first solution is using a widely available automated program for collecting data. A broad range of software is available commercially; for example: BehavioSec and KeystrokeID. Another solution involves the use of standardized databases which has been formerly created and published for the purpose of keystroke dynamics research. A list of some of the databases available online can be found in [31]. Using these solutions could not only standardize the data collection method, it could also decrease a duplication of effort among researchers.

6.1 Environment Controlling

There are two basic categories in the way experiments have been conducted in free-text keystroke studies.

Experiments have either been conducted in a controlled environment or in an uncontrolled one. In a controlled environment, users are asked to type on a specific machine which has built-in software for recording the keystrokes. Thus, the same external conditions are consistent for all users. The issue with this kind of arrangement is that it may not have the same characteristics as those encountered in realistic situations, therefore, the response may not be representative of a user's typical typing patterns.

In uncontrolled environments, on the other hand, users are asked to either download a program on their machines to collect their keystrokes [24, 30] or to use an online data collection form [15, 25]. This indicates that the data is collected wherever and whenever it is convenient to the user. Although, this method provides a realistic representation of normal circumstances for the user, each user's surroundings can be very different, which makes the data harder to analyze. This might be the reason for inconsistencies in the keystroke data provided by the users. A lot research done using free-text keystroke systems has so far been conducted in uncontrolled environments due to the desire to imitate the lifelike conditions of a real authentication system [15, 12, 28].

6.2 Keyboard Type

Using different brands of computer has a big impact on the user's typing pattern since the keyboard of different brands differs in key size and spacing between keys which is clearly a reason that users may type differently than normal [31]. Furthermore, different keyboards have different key pressing sensitivity levels which consequently may affect the timing data collected from the users. Using a laptop keyboard adds another variation which can also affect the typing behavior; because laptops provide the freedom of movement, users may use it in different positions such as on a bed or on a table.

Villani et al. [20] investigated the case of using different keyboards in free-text keystroke systems. One of their experiments was conducted using a desktop keyboard and another was performed on a lab top keyboard. A significant finding was produced in this study which can be summarized as: the system has a good chance of accurately identifying a user as long as he uses the same type of keyboard for training and testing. It is therefore important that researchers attempt to stick to using the same keyboard in order to maintain the same level of consistency throughout the data collecting process [19].

6.3 Entry Mode

Because free-text keystroke systems are used for long text, it makes more sense to allow the users to enter whatever text they prefer. Having said that, studies conducted have actually used two different methods for text entry in the experiments conducted for free-text authentication. The first technique allowed the users to type completely free text as they desired, such as: typing an e-mail or typing a report for work or an essay for school [15, 16]. The second approach required the users to type a specific long text from an article, in which the users needed to copy specific text into a section specified for text entry [19, 25].

In the research conducted by Villani et al. [20], participants were asked to be a part of several experiments with different conditions. One of these tests incorporated a copy-task in which the participants were asked to copy a predefined long text. Another included a free-text input where users were free to type-in arbitrary text. In this study, it was found that the accuracy of correctly authenticating a user decreased considerably when the user used different input modes in the training and testing phases. Moreover, it was also shown that the accuracy in free-text typing mode was higher than that in the copying-task mode. This can be explained by the frequent pauses that a user has to perform in order to look at the text during the copy-task which might cause the collected data to be inconsistent.

6.4 Text Length

One area that keystroke systems lack in is the amount of information that can be obtained. The only data that can be collected while the user is actually typing is the time each key is pressed and released, from which only little information can be inferred, including the time interval between each two consecutive keys and the duration time for each key press. In addition the data is often not stable since it changes based on the environment surrounding the experiment or based on the state of mind of the user at the time. As a result, to reduce the effect of such instabilities, much research has shown more interest in using short free-text [e.g. 28, 1, 18]. Realistically though, it is not enough to use short texts to analyze keystrokes since it does not offer an adequate amount of information to distinguish between users. Consequentially, using longer sample texts is considered a better alternative [11, 15, 16].

Moreover, Curtin et al. [19] provided evidence that using long-texts increases the chance of having more repetitions of the same di-graph in the training and testing samples which will, consequently, increase the stability of its mean and standard deviation significantly. The only problem with using long texts systems is that the training phase unavoidably needs more time. In their experiment, Curtin et al. investigated the accuracy of identifying users when typing long-texts under the condition that training and testing texts were different in length. The accuracy from different text/same length experiments was better than that from different text/different length experiments. Therefore, improving authentication accuracy can be achieved via standardization of the feature measurements i.e. the text size in this case.

6.5 User's Experience

The user's health and state of mind are a very crucial part of the authentication process using keystroke dynamics. The user's typing skills and level of comfort while using a keyboard are additional characteristics that have a clear impact on the user's typing behavior. The more skillful the user is, the more stable his/her fingers are located on the keyboard and the more familiar he is with the position of each character on the keyboard. This will result in a more consistent typing pattern all through.

Samura and Nishimura [23] conducted a study that examined keystroke dynamics for long free-texts. The experiment participants were divided into three groups based on their typing speed, specifically the number of letters typed in a 5 minute period. This study indicated that the best recognition accuracy was obtained from the group which typed fastest.

6.6 Monitoring Mode

A free-text keystroke system is a continual process of identity verification which is taking place during the course of the whole time a user is using the system. This can be done in either a continuous manner or a periodic manner. Continuous authenticating is done, in real time, every time a key is clicked on the keyboard [9]. Although this method provides strong imposter detection, it is computationally expensive. Periodic authentication, alternatively, is repeated every time a certain text is entered [24]. This is a less strict method, security wise, yet it is computationally cheaper. Moreover, waiting until a specific text is entered may cause the system to wait for long periods of time if this particular text does not occur frequently enough in the typed text; which will represent a security threat for the system.

A periodic verification scheme that included the use of interruptions was utilized in [27]. In this research, the identity of the user was only verified after text breaks e.g. user leaving the PC for a coffee break. The system only captured the first burst of input after each pause in order to analyze it. The method does though reduce the frequency of authentication checks which is a key reason for reducing the false alarm rate in addition to decreasing the computational cost.

6.7 Words Choice

As mentioned, some free-text systems depend on periodic authentication where the authentication process is actually performed every time a particular text is entered. It is clear that choosing a specific piece of text is crucial for training and testing the system. It might be thought that using familiar English words may realize more consistent typing patters. However this has been shown to be wrong by Janakiraman and Sim [24].

In their research, Janakiraman and Sim introduced a new “goodness” measure which was suggested to be used to calculate the universality, accuracy and expectancy of a word used for free-text keystroke authentication. Universality is a measure to identify if a word is one of the words commonly used by users or not. Accuracy measures how unique a word is. Lastly, expectance is used to calculate the average number of keystrokes typed before that word actually appears in the text. Unexpectedly, using the goodness measure, the result of this experiment revealed that non-English words, such as: ‘tmr’ which is an abbreviation of ‘tomorrow’ used in online chats, are better than English words for identification and verification purposes.

6.8 Number of Training Samples

When considering the training phase in fixed-text keystroke systems, it is hard to ignore the time required for training the system by retyping the password again and again. This is not an issue in free-text keystroke systems where the user’s data is collected while performing daily tasks. This implies that the free-text method is more practical in real life situations and easier to use since it causes less burden for the user. For example, 15 samples were collected from the participants over a two weeks period in [27]; each sample was 400 characters long of whatever the user needed to type at the time. This demonstrates that even though the samples were long, they were collected transparently to the user.

From the experiment results conducted by Gunetti et al. [32] it was found that the accuracy of the system generally escalated when the number of samples in the user’s profile was increased. Meanwhile an effective mechanism for profile enhancement was suggested in [18] where the user’s profile was expanded, during the typing session, by adding new key-pairs timing data attained from text entered by the user after being authenticated.

7. Applications

Although more than a quarter of a century has passed since keystroke authentication was first researched, it has not yet been applied much in the security field. In addition to the security that keystroke authentication systems can provide by locking-up the workstation when an imposter is detected at any point of time during which the system is used, a wide variety of other applications can also benefit from such authentication schemes. The applications, listed in this section, are examples of some situations where free-text keystroke authentication is more applicable than fixed-text systems.

7.1 Different Languages Authentication

Most of the work done on keystroke dynamics has concentrated on using the same language for training and testing the system. Gunetti et al. [32] though gave empirical proof that free-text typing patterns could be used to authenticate the user even when the test samples were written in different languages to that of the samples in the user’s profile. Evidently, this only works when the two languages share a significant number of di-graphs. So, languages like English and Italian which have largely the same alphabet can be used for this kind of authentication but English and Arabic, for example, cannot be used because they have a completely different set of letters.

The data used in this study was provided by Italian speakers each of whom provided two samples typed in Italian and another two samples typed in English. From the experimental results, about 10% mistakes in identification occurred when the test sample was in a language different to that of the user's profile. Better performance was obtained when the user's profile contained samples in both languages. The error rate was even smaller when the test sample was in the same language as that which dominated the samples in the user's profile. By experimenting with different combinations of template samples and test samples, it was clear that samples provided by the same person while typed in different languages were more similar than samples provided by different persons while typed in the same language. An average performance of 1.61% FRR and 3.23% FAR was achieved in total. Thus, keystroke authentication for different language texts, is possible, though more difficult than the case where all samples are in the same language.

7.2 Old Profile Authentication

Most of the studies conducted in the free-text keystroke authentication field have had only a few months gap between the time the training samples were collected and the time in which the test samples were gathered. Gunetti et al., however, showed in [33] that a typing profile could still be used to identify a user, even though, it has been created a long time before the test samples are provided and investigated. Their original experiment involved 30 participants whom were asked to provide 15 samples each. The samples consisted of whatever the users choose to type. One and a half years later, the same 30 volunteers were asked again to provide another two free-text samples. It was discovered that even after such a long period of time their keystroke dynamics system was still able to identify users with an average accuracy of a 1.67% FAR and a 11.67% FRR.

7.3 Intrusion Detection

The continual authentication scheme that the free-text method provides is a very effective intrusion detection method. It is mainly used to notice any warnings with regards to irregularities in the typing patterns of a specific user. Moreover, free-text keystroke systems are used for active monitoring of the system which can aid in finding any intrusion quickly and reliably. One important issue that has to be addressed here is the generation of false alarms in continual keystroke-based authentication systems. It might cause frequent and rapid system halts with much annoyance for the users when they falsely occur. Therefore, Gunetti et al. [32] suggested using it combined with other authentication methods in order to reduce the false alarm rate.

7.4 Online Marketing

Free-text keystroke systems can also be utilized for identifying users over the internet. This is done by capturing a user's typing patters on their first visit to the website and then it can be used to identify returning users [32]. This data can be used to determine user preferences and interests which can be employed for marketing purposes. This approach, on the other hand, has many privacy issues regarding the amount of information that users are happy to hand-in to the websites they visit.

7.5 Cybercrime Investigating

User tracking through typing patterns can also help in cybercrime and investigating illegal electronic movements of anonymous users. Using free-text keystroke schemes was suggested for network forensics in [29] through attacker profiling which is conducted by collecting his/her typing patterns when surfing websites on the internet. This profile, collected for each user, can be used as a digital fingerprint gathered from the cybercrime scenes. This is considered as passive fingerprinting because it can be created without the knowledge of the attacker which can be extremely beneficial in fraud or identity theft cases where attackers are completely oblivious that they are being monitored. The issue with such a digital fingerprint is that it must be built progressively which requires a lot of internet service providers to collaborate and work together in facing such threats.

7.6 Identification and Authentication

Keystroke dynamics systems are used for two different purposes. Firstly: identification, which is a way of determining the user's identity when no data is available about their identity beforehand. In this method, a test sample is matched with all the templates stored in a database. The system assigns the user to the identity of the person whose template is the most similar to the test sample. The second purpose is authentication which is used to verify the identity of the user. The user supplies his identity and the system takes on the responsibility of making sure that the user is who he/she claims to be. The test sample in this case is only compared with the user's template in the database.

The complexity of performing identification is clearly higher than that of authentication since it includes comparing the test sample with all available templates which may be a very large undertaking in large scale systems. Identification also requires a larger amount of data i.e. longer text. From the definition of both methods, fixed-text keystrokes system is used mostly for authentication since it employs a password that is considered a mean for providing the user's identity. Free-text keystrokes system, on the other hand, is used for both identification and authentication [e.g. 34, 15].

7.7 User's Emotion Detection

Since free-text keystroke systems gather a lot of data from the user during the whole time he/she is using the computer, this data can also be used to infer the emotional state that the user is going through during the typing process. This has been employed in [35] to determine what the user is feeling during every day free typing. Feelings like frustration, focus, anger, stress, relaxation, excitement and tiredness were derived from the user's typing behavior. Extracting the emotional state that the user is going through in a particular period of time that the user is using the system has many benefits for intelligent computers. It helps the system make the right decisions regarding the best interaction method to practice with the user. The issue with using keystrokes for user emotion detection is that it can cause an invasion into the user's typing experience. For example, in [35], the user was required to determine his emotion every 10 minutes in order to train the system to identify his emotions automatically.

8. Security Issues

In this section we discuss the security level that free-text keystroke authentication systems provide. A list of the most common threats is provided here along with the degree of safety that free-text keystroke systems deliver against these dangers [36].

1. Shoulder surfing and user mimicking: is an attack in which the attacker monitors the victim typing, during the typing process, in order to try imitating his/her typing behavior. Even though there is little possibility of an attacker successfully mimicking a user typing pattern in fixed-text keystrokes, it is even harder to do so in free-text systems. Since it requires the attacker to observe the user's behavior for the whole time the user is logged-in, it is very rare that an attacker can actually imitate all the aspects of the user's typing behavior.

2. Spyware: is software downloaded into the victims' computer without their consent which is used to record information about them. Spyware is perhaps the biggest threat to keystroke dynamic authentication systems because it can record exactly the time each key is pressed and released. This can be used by the attacker to simulate the legitimate user's typing behavior. Nevertheless, it is still a hard task for the attacker to undertake in the case of the huge amount of data that free-text systems need to analyze.

3. Social engineering: is manipulating the user in order to obtain his/her private information. Tricking the victim to reveal his typing pattern is though not possible using telephone calls or face to face meetings. Yet, phishing e-mails can be used to trick the user to type some text which can be used to extract the victim's typing patterns. But even then, the attacker has to get hold of a sufficient amount of keystrokes to be able to actually simulate the victim's free-text typing patterns.

4. Guessing: is trying to guess the way that a victim types. There are simply too many different ways that a user might normally follow when typing. Therefore, guessing the typing behavior of another person is almost impossible in free-text keystroke dynamics.

9. Conclusions

Free-text keystroke dynamics is a non-intrusive method, since it only uses the behavioral data that users convey during regular typing tasks. In addition to that, it is relatively inexpensive; the only required hardware is the keyboard. However the most important benefit that free-text keystroke systems provide is that the typing patterns can still be used for authenticating users even after the authentication phase has passed. In addition, free-text authentication provides a valuable balance between security and usability which is highly desirable in the businesses world.

One concern about free-text keystrokes is that it tends to be instable in the sense that it might be influenced by the user state or by environmental conditions. Indeed some level of instability might occur without any obvious cause. Therefore, free-text authentication is probably best used as a part of a multi-factor authentication scheme [28, 37] that provides a higher level of security.

Generally, it is obvious that keystroke dynamics works more accurately for fixed-text compared with free-text. Therefore, it might be a good practice for free-text tests to take into consideration the actual words that the user is typing, in addition to the key hold time the di-graph's duration and latency times.

Moreover, determining the best method to follow to achieve the best authentication accuracy is not a straightforward task. Due to the variation of conditions that might be affecting the study participants, environment or procedure, the comparison between two or more methods is not always accurate. Therefore, a standardization mechanism has to be established to assure that factors affecting performance are in agreement in all the studies and hence can be properly compared.

Lastly, it is clear that the idea of using keystroke dynamics is not only restricted to the traditional keyboard, it can be conveyed to many other mechanisms like ATM machines and cell phones, which will then provide better every day protection for the standard user.

References

[1] F. Monrose and A. Rubin, "Authentication via Keystroke Dynamics", in the Fourth ACM Conference on Computer and Communication Security, 1997.

[2] M. Karnan, M. Akila and N. Krishnaraj, "Biometric Personal Authentication Using Keystroke Dynamics: a Review", Applied Soft Computing, Vol. 11, No. 2, 2011, pp. 1565–1573.

[3] R. Gaines, W. Lisowski, S. Press and N. Shpiro, "Authentication by Keystroke Timing: some Preliminary Results", Rand Corporation, Rep. R-256-NSF, 1980.

[4] S. J. Shepherd, "Continuous Authentication by Analysis of Keyboard Typing Characteristics", in European Convention on Security and Detection, 1995, pp. 111-114.

[5] R. Giot, M. El-Abed, B. Hemery and C. Rosenberger, "Unconstrained Keystroke Dynamics Authentication with Shared Secret", Computers and Security, Vol. 30, 2011, pp. 427-445.

[6] J. Garcia, "Personal Identification Apparatus", U.S. Patent 4621334, 1986.

[7] M. S. Obaidat and B. Sadoun, "Verification of Computer Users Using Keystroke Dynamics", in IEEE Transactions on Systems, Man and Cybernetics - Part B: cybernetics, 1997, Vol. 27.

[8] K. S. Killourhy, "A Scientific Understanding of Keystroke Dynamics", PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, US, 2012.

[9] P. Bours, "Continuous Keystroke Dynamics: a Different Perspective Towards Biometric Evaluation", Information Security Technical Report, Vol. 17, 2012, pp. 36-43.

[10] P. Bours and H. Barghouthi, "Continuous Authentication Using Biometric Keystroke Dynamics", in The Norwegian Information Security Conference, 2009.

[11] P. S. Dowland, H. Singh and S. M. Furnell, "A Preliminary Investigation of User Authentication Using Continuous Keystroke Analysis", in the IFIP 8th Annual Working Conference on Information Security Management and Small Systems Security, 2001.

[12] H. Davoudi and E. Kabir, "A New Distance Measure for Fee Text Keystroke Authentication", in the14th International CSI Computer Conference, 2009, pp. 570-575.

[13] K. Pilsung, P. Jooseong, P. Sunghoon , Y. Joonha and C. Sungzoon, "Keystroke Dynamics Analysis Based on Long and Free Text", in Fall Korean Industrial Engineering Conference, 2009.

[14] H. Davoudi and E. Kabir, "Modification of the Relative Distance for Free Text Keystroke Authentication", in the 5th International Symposium on Telecommunications, 2010.

[15] D. Gunetti and C. Picardi, "Keystroke Analysis of Free Text", ACM Transactions on Information System Security, Vol.8, No.3, 2005, pp. 312-347.

[16] T. Sim and R. Janakiraman, "Are Digraphs Good for Free-text Keystroke Dynamics?", in the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp.1-6.

[17] S. Park, J. Park and S. Cho, "User Authentication Based on Keystroke Analysis of Long Free Texts with a Reduced Number of Features", in the Second International Conference on Communication Systems, Networks and Applications, 2010.

[18] S. Sing and K.V. Arya, "Key Classification: a New Approach in Free Text Keystroke Authentication System", in Third Pacific-Asia Conference on Circuits, Communications and System, 2011, pp. 1-5.

[19] M. Curtin , C. Tappert, M. Villani, G. Ngo, J. Simone, H. St. Fort, and S.-H. Cha, "Keystroke Biometric Recognition on Long-text Input: a Feasibility Study", in IMECS, 2006.

[20] M. Villani, C. Tappert, G. Ngo, J. Simone, H. St. Fort and S. Cha, "Keystroke Biometric Recognition Studies on Long-text Input Under Ideal and Application-oriented Conditions", in the IEEE Computer Society Workshop on Biometrics, 2006.

[21] K. Hempstalk, E. Frank and I. H. Witten, "One-class Classification by Combining Density and Class Probability Estimation", in the European Conference on Machine and Learning and Principles and Practice of Knowledge Discovery in Database, 2005, pp.505-519.

[22] D. Gunetti and G. Ruffo, "Intrusion Detection through Behavioral Data", in the Third International Symposium on Advances, 1999, pp. 383-394.

[23] T. Samura and H. Nishimura, "Keystroke Timing Analysis for Individual Identification in Japanese Free Text Typing", in ICROS-SICE International Joint Conference, 2009.

[24] R. Janakiraman and T. Sim, "Keystroke Dynamics in a General Setting," in Advances in Biometrics, 2007, Vol. 4642, pp. 584–593.

[25] J. Hu, D. Gingrich and A. Sentosa, "A K-nearest Neighbor Approach for User Authentication through Biometric Keystroke Dynamics", in IEEE International Conference on Communications, 2008, pp. 1556-1560.

[26] H. Davoudi, E. Kabir, "User Authentication Based on Free Text Keystroke Patterns", in the 3rd Joint Congress on Fuzzy and Intelligent Systems, 2010.

[27] N. Bakelman, J. V. Monaco, S. H. Cha, and C. C. Tappert, "Continual Keystroke Biometric Authentication on Short Bursts of Keyboard Input", in Pace University CSIS Research Day, 2012.

[28] C. Chantan, S. Sinthupinyo and T. Rungkasiri, "Improving Accuracy of Authentication Process via Short Free Text Using Bayesian Network", International Journal of Computer Science Issues, Vol. 9, No. 3, 2012.

[29] A. A. E. Ahmed, I. Traore and A. Almulhem, "Digital Fingerprinting Based on Keystroke Dynamics", in the Second International Symposium on Human Aspects of Information Security and Assurance, 2008.

[30] J. R. M. Filho and E. O. Freire, "On the Equalization of Keystroke Timing Histogram", Pattern Recognition Letters, Vol. 27, No.13, 2006, pp. 1440-1446.

[31] S. P. Banerjee and D. L. Woodard, "Biometric Authentication and Identification Using Keystroke Dynamics: a Survey", Journal of Pattern Recognition Research, Vol. 7, 2012, pp. 116-139.

[32] D. Gunetti, C. Picardi, and G. Ruffo, "Keystroke Analysis of Different Languages: a Case Study", in the Advances in Intelligent Data Analysis, 2005, Vol. 3646, pp. 133–144,

[33] D. Gunetti, C. Picardi, and G. Ruffo, "Dealing with Different Languages and Old Profiles in Keystroke Analysis of Free Text", in the Advances in Artificial Intelligence, 2005, Vol. 3673, pp. 347–358.

[34] T. Buch, A. Cotoranu, E. Jeskey, F. Tihon and M. Villani, "An Enhanced Keystroke Biometric System and Associated Studies", in Pace University CSIS Research Day, 2008.

[35] C. Epp, M. Lippold and R. L. Mandryk, "Identifying Emotional States Using Keystroke Dynamics", in the Conference on Human Factors in Computing Systems, 2011, pp. 715-724.

[36] D. Shanmugapriya and G. Padmavathi, "A Survey of Biometric Keystroke Dynamics: Approaches, Security and Challenges", International Journal of Computer Science and Information Security, Vol. 5, No. 1, 2009.

[37] O. S. Adeoye, "Evaluating the Performance of Two-factor Authentication Solution in the Banking Sector", International Journal of Computer Science Issues, Vol. 9, No. 2, 2012.

[38] A. Messerman, T. Mustafic, S. A. Camtepe and S. Albayrak, "Continuous and Non-intrusive Identity Verification in Real-time Environments Based on Free-text Keystroke Dynamics," in the International Joint Conference on Biometrics, 2011, pp.1-8.

Arwa Alsultan is pursuing a PhD degree in Computer Science from the School of Systems Engineering at the University of Reading, Reading, Berkshire, UK. She completed her Master's degree in Computer Science from the Computer and Information Science College at the King Saud University, Riyadh, SA in 2010. She works as a lecturer at the IT Department in the Computer and Information Science College at the King Saud University, Riyadh, SA.

Kevin Warwick is Professor of Cybernetics at the University of Reading. His research interests are in Artificial Intelligence, Robotics, Biomedical Engineering and Control Systems. He has D.Sc. degrees from both Imperial College London and the Czech Academy of Sciences. He has published over 500 research papers and is perhaps best known for his experimentation with implant technology.