10 KEYSTROKE DYNAMICS BASED AUTHENTICATION

M. S. Obaidat
Monmouth University
W. Long Branch, NJ
obaidat@monmouth.edu

B. Sadoun
Applied Science University
Amman, Jordan
bsadoun@aol.com

Abstract *This chapter deals with the applications of keystroke dynamics to authenticate/verify access to computer systems and networks. It presents our novel contribution to this area along with other related works. The use of computer systems and networks has spread at a rate completely unexpected a decade ago. Computer systems and network are being used in almost every aspect of our daily life. As a result, the security threats to computers and networks have also increased significantly. We give a background information including the goals of any security system for computers and networks, followed by types of security attacks on computers and networks. We present the applications of keystroke dynamics using interkey times and hold times as features to authenticate access to computer systems and networks.*

Keywords: *Keystroke dynamics, computer security, computer verification/authentication, interkey times, hold times, neural networks, pattern recognition, system identification.*

1. Introduction

Computer systems and networks are now used in almost all technical, industrial, and business applications. The dependence of people on computers has increased tremendously in recent years and many businesses rely heavily on the effective operations of their computer systems and networks. The total number of computer systems installed in most organizations has been increasing at a phenomenal rate. Corporations store sensitive information on manufacturing process, marketing, credit records, driving records, income tax, classified military data, and the like. There are

many other examples of sensitive information that if accessed by unauthorized users, may entail loss of money or releasing confidential information to unwanted parties [1-9].

Many incidents of computer security problems have been reported in the popular media [1]. Among these is the recent incident at Rice University where intruders were able to gain high level of access to the university computer systems which forced the administration to shut down the campus computer network and cut its link with the Internet for one week in order to resolve the problem. Other institutions such as Bard College of the University of Texas Health Science center reported similar breaches. Parker [10] reported that one basic problem with computer security is that the pace of the technology of data processing equipment has outstripped capability to protect the data and information from intentional misdeeds.

Attacks on computer systems and networks can be divided into active and passive attacks [11-12].

1. Active attacks: These attacks involve altering of data stream or the creation of a fraudulent stream. They can be divided into four subclasses: masquerade, replay, modification of messages, and denial of service. A masquerade occurs when one entity fakes to be a different entity. For example, authentication sequence can be collected and replayed after a valid authentication sequence has taken place. Replay involves the passive capture of data unit and its subsequent retransmission to construct an unapproved access. Modification of messages simply means that some portion of a genuine message is changed, or that messages are delayed or recorded, to produce an unauthorized result.

2. Passive attacks: These are inherently eavesdropping on, or snooping on, transmission. The goal of the attacker is to access information that is being transmitted. Here, there are two subclasses: release of message contents, and traffic analysis. In the first subclass, the attack occurs, for example, on an e-mail message, or a transferred file that may contain sensitive information. In traffic analysis, which is more sophisticated, the attacker could discover the location and identity of communicating hosts and could observe the frequency and length of encrypted messages being exchanged. Such information could be useful in guessing the nature of information/data.

Passive attacks are difficult to detect, however, measures are available to prevent them. On the other hand, it is difficult to prevent the occurrence of active attacks.

Computer security goals consist of maintaining three main characteristics: integrity, confidentiality, and availability [12]. These goals can overlap, and they can even be mutually exclusive. For example, strong protection of confidentiality can severely restrict availability to authorized parties.

1. Integrity: This characteristic means that the assets can be modified (e.g., substitution, deletion, or insertion) only by authorized parties or only in authorized ways. Integrity means different things in different contexts [12]. Among the meanings of integrity are precise, accurate, unmodified, consistent, and correct result. Three aspects of integrity are commonly recognized: (i)

authorized actions, (ii) separation and protection of resources, and (iii) error detection and correction.

2. Confidentiality: This is also called privacy or secrecy. It means that the computer and network systems are accessible only to authorized parties. The type of access can be read-only access; the privileges include viewing, printing, or even just knowing the existence of an object.

3. Availability: This term is also known by its opposite, denial of service. Here, the term means that assets are accessible to authorized parties. An authorized individual should not be prevented from accessing objects to which he/she has legitimate access. Availability applies both to data and service.

One major aspect of a multiuser computer system that can be a significant threat to security arises from access to remote terminals. Denning [13] states that the effectiveness of access control is based on two ideas: (1) user identification and (2) protection of the access rights of users. Protecting the access rights of users is generally done at the system level, by not allowing access permissions to be altered except by authorized "super-users". Denning [13] presents several cryptographic types of user authentication, in addition to password schemes. To properly identify a valid user, one or more of the following techniques are commonly used [1-8,14]:

- What the user knows or has memorized (password).
- What the user carries or possesses (e.g., a physical key).

User passwords are the most common means of identification, but they are subject to compromise, either by interception as the user types it, or by a direct attack. Hardware locks are secure, but there is no way for a computer system to know that the users who have logged on are really who they say they are.

A third method, using biometric characteristic such as the user's typing technique, was discounted by Walker as impractical [14]. However, more recent work by Obaidat et al. [1-8], Gaines et al. [22], Umphress and Williams [15], Leggett and Williams [25], Yong and Hammon, and Joyce and Gupta [21] has shown that a user can be identified based on his/her typing technique using traditional pattern recognition and neural network techniques. These research efforts in keystroke dynamics have focused on attributes like stream of interkey times (latency periods between keystrokes) and hold times (durations between the hit and release moments of key hold) to provide a unique feature/identifier/signature for authenticating an individual's identity.

2. Types of Security Attacks

The attacks on the security of a computer system or network can be characterized by viewing the function of the computer system/network as a provider of information. The possible attacks that may occur on a computer and networking system are as follows [11]:

1. Interruption: In this case, an asset of a system becomes unavailable, lost, or unusable due to alteration. Clearly, this is an attack on availability. Examples include vicious destruction of hardware devices, deletion of a program/data file, cutting of a communication link, disabling of a file management system, failure of an operating system function, etc.
2. Interception: This means that an unauthorized individual has gained access to an asset. Clearly, this is considered an attack on confidentiality. The consequences range from inconvenience to catastrophe. Examples include copying of data files/programs, and wiretapping to obtain data in a network. The unauthorized party could be a person, a program, or a computer system.
3. Modification: Here the unauthorized party not only accesses but also tampers with an asset. Clearly, this is an attack on integrity. Examples include changing values in a record or data file, altering a program so that it performs differently, and modifying the contents of messages being sent over a network. The modification can be done on the hardware configuration as well. Some cases of modification can be detected with simple schemes, but others may be more difficult if not impossible to detect.
4. Fabrication: Here an unauthorized party inserts counterfeit objects into the system. This is considered an attack on the authenticity of the computer system or network. The intruder may insert spurious transactions into the system, or add records to an existing data base. In some cases, these additions can be detected as forgeries, but if done skillfully, they are virtually indistinguishable from the real thing.

Computer networks, in particular, have security problems due to the following reasons [11-12]:

1. Sharing: Since resources and work are shared, more users have the potential to access networked systems than a single computer node.
2. Anonymity: An intruder can attack from thousands of kilometers away and thus, never have to touch the system attacked or come into contact with any of its managers or users.
3. Complexity of system: Operating systems tend to be very complex. Reliable security is not easy to implement on a large operating system, especially one not designed specifically for security. Designing a secure computer network is even more difficult since it combines two or more computer systems with possibly dissimilar operating systems.
4. Multiple points of attack: When a file physically exists on a remote host, the file may pass via many nodes in order to reach to the user over the computer network.
5. Unknown path: network users seldom have control on the routing paths of their own packets. Routes taken depend on many factors including load conditions and traffic patterns.

3. Predicting Human Characteristics

As early as the beginning of the 20th century, psychologists, and mathematicians have experimented with human actions. Psychologists have demonstrated that human actions are predictable in the performance of repetitive, and routine tasks [15]. In 1895, observation of telegraph operators showed that each operator had a distinctive pattern of keying messages over telegraph lines [16]. Furthermore, an operator often recognized who is typing on the keyboard and sending information simply by listening to the characteristic pattern of dots and dashes. Since the beginning of civilization, humans are able to recognize the person coming into a room from the sound of steps of the individual. Clearly, each person has a unique way of walking. Similarly, telegraph operators were able to find out who was sending message by just listening to the characteristics of dots and dashes.

Today, the telegraph keys have been replaced by other input/output devices such as keyboard and mouse. It has been established that keyboard characteristics are rich in cognitive qualities and hold promise as an individual identifier. Anyone sitting close to a typist or has an office next to a typist is usually able to recognize the typist by keystroke patterns.

Over many centuries, humans have relied on written signatures to verify the identity of an individual. It has been proven that human hand and its environment make written signatures difficult to forge. It has been shown [21] that the same neurophysiological factors that make written signature unique are also exhibited in an individual typing pattern. Once a computer user types on the keyboard of a computer, he/she leaves a digital signature in the form of keystroke latencies (elapsed time between keystrokes and hold times).

Human nature dictates that a person does not just sit before a computer and deluge the keyboard with a furious and continuous stream of non-stop data entry. Instead, the person types for a while, pauses to collect thoughts and ideas, pauses again to take a rest, continues typing, and so forth. In developing a scheme for identity verification, a common baseline must be established for determining which keystrokes characterize the individual's key pattern and which do not. Physiologists have studied human interface with computer systems and developed several models describing the interface to computers. One of the popular models is the keystroke-level model developed by Card et al. [17]. Their model describes the human-machine interaction during a session at a computer terminal. It was intended as a vehicle for the evaluation and comparison of competing designs for highly interactive programs. The keystroke level model summarizes the terminal session as follows:

$$T_t = T_a + T_e,$$

where T_t represents the duration of the terminal session; T_a represents the time required to assess the task, build mental representation of the functions to be performed, and choose a method for solving the problem; and T_e represents the time needed to execute all functions constituting the task.

Note that T_a varies according to the extent of the considered task, experience of the user, and understanding of the functions to be performed. Clearly, this term is not

quantifiable. Thus T_a cannot be used to characterize a person. On the other hand, T_e describes mechanical actions which itself can be expressed as:

$$T_e = T_k + T_m,$$

where T_k is the time to key in information and T_m is the time needed for mental preparation. Note that when interacting with a program, the user does not divide his actions into mental time followed by keystroke time. Instead, the two are intermixed.

Shaffer [18] has shown that when a typist is keying data, the brain acts as a buffer, which then outputs the text onto the keys of the keyboard. Average capacity of the buffer is about 6-8 characters in length [19]. Because of the limited size of the buffer, typists group symbols into smaller cognitive units and pause between each unit. Cooper [19] established that the typical pause points are between words as well as within words that are longer than 6-8 characters.

4. Applications of Keystroke Dynamics Using Interkey Times as Features

Although handwriting and typing are distinct manual skills, they both have measurable characteristics that are unique to those who perform the task [5,6]. Umphress and Williams [15] have conducted an experiment for keystroke characterization. They used two sets of inputs for user identification, namely, a reference profile and a test profile. Each keystroke was time-tagged to the nearest hundredth of a second and stored on a floppy disk. Another program was used to analyze the keystrokes and produced a database of reference profiles for each individual participating in the experiment. A third program was used to compare test profile keystrokes to reference profiles. Seventeen persons participated in that experiment. Each person was asked to take two typing tests. These tests were separated over several days. In the first test, the participants were asked to type about 1400 characters of prose. The second typing test, the test profile, consisted of 300 characters of prose. It was found that a high degree of correlation could be obtained if the same person typed both the reference and test profiles. Several medium confidence levels were assigned in cases where the typists of the profile differed. However, in most cases test profiles had low scores when the typists was not the same person who typed the reference profile.

Obaidat and his colleagues [3,5,6] described a method of identifying a user based on the typing technique of the user. The inter-character time intervals measured as the user types a known sequence of characters was used with traditional pattern recognition techniques to classify the users, with good verification results. By requiring the character sequence to be typed two times, and by using the shortest measurements of each trial, better results were obtained than if the user typed the sequence only once. The minimum-distance classifier provided the best classification accuracy. In order to obtain a better classification accuracy, their analysis considered the effect of the dimensionality reduction, and the number of classes in the identification system. The measurement vector is obtained by computing the real-time

durations between the characters entered in the password. Figure 10.1 shows the flow chart of the overall steps general recognition system.

Obaidat and Macchiarolo [4, 7-8] used some traditional neural network paradigms along with classical pattern recognition techniques for the classification/identification of computer users using as feature the interkey times of the keystroke dynamics. They considered six users in their work. The dataset used for the recognition of computer users is made up of the time intervals between successive keystrokes by users while typing a known sequence of characters (phrase). The participants in the experiment were asked to enter the same phrase, which was not visible during the process of typing; therefore, it was important to display the message on the monitor after entering it. The phrase was retyped by the participant if it was entered incorrectly. The time duration between keystrokes was then collected by using an IBM compatible PC-based data acquisition system which used Fortran and assembly language programming. The assembly language procedures make use of the software keyboard interrupt facility and provide the main program with the time duration between keystrokes. For example, if the password "OBAIDAT" were entered, then the assembly language program would compute the time duration between the letter pairs (O, B), (B, A), (A, I), (I, D), (D, A), and (A, T). An open period of time was given to the participants to conduct the experiment. This helped in averaging out the effect of uncorrelated sources of noise that could be introduced by instruments and participants. Furthermore, it helped to gather data that represent the different modes of the participants. A phrase that consists of 30 vector components was used first; however, only the first 15 vector components were used later since using the remaining vectors did not change the results. The data were collected from six different users over a six-week period. The total number of measurement vectors per user was 40. The raw data were arranged as follows:

- each pattern consisted of 15 values, which were the time durations in milliseconds between successive keystrokes of a known character sequence;
- there were 40 trials per user (class) (600 values per class), and
- there were six classes that were defined (3600 values total).

For training purposes, the raw data were separated into two parts: all of the odd-numbered patterns of each class, and all of the even-numbered patterns. In any given simulation run, only half of the data were used to form the training set. After each network was trained, the entire pattern set (24 patterns) was presented to the network for classification.

Several versions of the training data were created to investigate the network's ability to generalize, rather than to memorize the training set. The difference in the training pattern sets are: (a) whether the patterns are from the odd or even half of the raw data, and (b) the granularity of the training set which is defined by the number of raw patterns averaged to compose each training pattern. For example, if all raw patterns in a class are averaged together to form a single training pattern, the granularity is low. On the other hand, if no averages are used, i.e., all of the patterns are used in training, the granularity is high. Intuitively, when a higher granularity is

used for training then better classification performance should be obtained. Table 1 shows one example of a training set used [4].

During the investigation phase, various combinations of these patterns were created to test the learning abilities of the three different neural network paradigms. After experimentation determined the best neural network architecture for this application, the network was incorporated into an "on-line" system that would collect the character time intervals from users in real-time and perform a classification immediately. The simulators used to simulate the neural network paradigms were written using C programming language. Some critical timing functions were written in assembly language. The on-line computer security system consists of the following major tasks:

Data Input

The timing functions used the 8253 timer that is located in all IBM-PC compatible computers to measure the interkey time intervals. In the case of a PC-AT computer, a BIOS microsecond timing function [4] is used instead, as the 8253 timer outputs are not accessible. Similar schemes can be used for other computer platforms. In all platforms, a calibration subroutine is called before any timings are measured. The calibration routine first determines which timing method to use based on the computer type and then calibrates the timer using the time-of-the day clock. During actual timing of keystrokes, the routine gets each keystroke, stores it, and then begins timing, while waiting for the next keystroke. When the next keystroke occurs, it stores the time intervals and the key hit. This process is then repeated, and a second set of measurements is recorded. It has been shown that taking the lowest value of each interval, based on two sets of values, improves the classification accuracy.

Training

To train the neural network, a set of measurement vectors from each user class was required. These vectors are collected from each user and stored. When a sufficient number of vectors have been collected, they may be averaged and normalized to form a set of patterns that will be used to train the network. The number of pattern vectors is defined by the user of the program. The user can describe the network configuration to the program, and memory is allocated for the processing units (neurons), training pattern storage, and weight vector storage. Training consists of applying a pattern vector to the input, comparing the current output with the target output, and adjusting the weight values according to the training algorithm. When the error of the training vector set is reduced to a pre-defined threshold which is the total summed squared (TSS) error less than or equal to 0.01 in our work, training is stopped, and the entire network is saved to a disk file.

Classification

To run the program as an on-line classifier/identifier, the network is recalled from the file saved after training. Memory is allocated as needed, and the weight vectors are read from the file. The user is prompted to type the keyword phrase. The inter-

character intervals are stored, normalized using the user-selected normalization function (either the percentage of the largest value, unit-length vector, or none) and presented to the network inputs. The input values are propagated through the network, which has the same number of output units as there are defined user classes. The output unit which is strongly activated (above a user-defined threshold) represents the classification of the input measurement vector.

Normalization, Performance, and Incorporation

In an operational test, six user typed a 15-character phrase 20 times, each over a period of 6 weeks. The raw data were used to create pattern sets to train the network. Two types of normalization of data were investigated: unit length vector, and fraction of the largest element. The unit length vector normalization is obtained by dividing each element of the measurement vector by the total magnitude of the vector (square root of the sum of the square of each element's value). This proved to be unsatisfactory in that the vectors of different users were made more similar, and the network could not distinguish the difference between them during training. By dividing each element's value by the largest element value, the elements were simply rescaled into a range from 0 to 1. This is the range needed for the inputs of the neural network, while preserving the relative differences in the elements. To create the training patterns, two normalized vectors were averaged together to create each training pattern.

The training time of the network can be varied by adjusting the learning rate and momentum parameters. The learning rate is the fraction of the error value that is used to compute the weight adjustments. The momentum value is the fraction of the previous adjustment that is added into the current adjustment. After training is finished, each user tested the network. The overall accuracy was 97.8%.

The system can be easily incorporated into a computer security system. Initially, each user that is to have privileged access would be required to submit samples of his inter-character typing for a known phrase. These samples are acquired through the use of the data input module, and are kept by an administrator. The administrator then generates the training set, and configures a network using the training module. The network will have a number of inputs equal to the number of measurements in each vector, a number of hidden units, and a number of outputs equal to the number of users. The weight values are then determined through training, which could take place off line or as a background process. After training, the weights are stored and can be quickly recalled for on-line classification. When a user needed to be removed or added to the authorized list, the training set would have to be regenerated, however, the training module can automatically regenerate a training set from the existing and new sample data. Adding a user would require adding another output unit to the network, and the additional weights adjusted through training.

In practice, a user would identify himself by using the number assigned to his sample classification. He would then be asked to type the keyword phrase. His inter-character typing intervals would be collected and classified. If the user's number matches the class assigned by the classification system, then the user is granted access. If the classification does not match, several things could happen:

1. The user is denied access. This is the highest security level.

2. The user is granted access, after providing a higher-level password.
3. The user is granted only limited access.
4. The user is granted access, but a "warning" is signaled to the administrator, and the user's actions are intercepted for later analysis. This is considered the lowest security level.

There is a tradeoff to consider with any security system; the risk of security breach balanced with the user inconvenience.

Bleha and Obaidat [6] experimented with the Percepton algorithm as a classifier to verify the identity of computer users. By performing the real-time measurements of the time durations between keystroke entered in the user's password, data was collected from 10 valid users and 14 invalid users over a period of 8 weeks. The password used was the user's name. Decision functions were derived using half of the data (training data) to compute the weight vectors. The decision functions were applied to the remaining half of the data (testing data) to verify the users. An error of 9% in rejecting valid users, and an error of 8% in accepting invalid users were achieved. The percepton algorithm was found to be robust with respect to the choice of the initial weight vector.

Obaidat [13] evaluated the performance of five pattern recognition algorithms as applied to the identification of computer users using the time intervals between successive keystrokes created by users while typing a known sequence of characters. These algorithms are potential function, Bayes classifier, minimum distance and the cosine measure. A 100% accuracy was achieved when the potential function algorithm was used. The least successful algorithm was the cosine measure. Obaidat and Sadoun [2] evaluated the performance of a newly devised neural network scheme, called Hybrid-Sum-Of-Products (HSOP) [27] for computer users verification and other classification problems. They compared the performance of HSOP to the Sum-Of -Products and Backpropagation neural network paradigms. They found that HSOP performs better than the other two paradigms. In their work they used interkey time intervals between keystrokes while typing a known phrase.

5. Applications of Keystroke Dynamics using Hold Times as Features

Obaidat and Sadoun [1] verified computer users using hold times of keystroke dynamics as features to authenticate computer users. The participants in the experiment were asked to enter their login user ID during an eight-week period. The program collected key hit and key release times on an IBM compatible PC to the nearest 0.1 ms. The program was implemented as a *terminate and stay resident program* in an MS-DOS based environment. The standard keyboard interrupt handler was replaced by one that could sense the incoming keyboard scan codes and record them along with a time stamp. The program measures the time durations between the moment every key button is hit to the moment it is released. This procedure was performed for each letter of the user ID and for each participant. A scan code is generated for both the hit and release of any key.

The login routine was modified so that each time a login attempt was made, the timing vector of the assart (hit and release time) was stored for analysis. This procedure increases the dimensionality of an N character string to (2N-1). Such a high dimensionality can provide better discrimination even if the number of characters is not large. The login monitoring results were collected from 15 users who were given open period of time to conduct the experiment. Such approach averaged out the effects of fatigue and stress as well as the uncorrelated sources of noise. The forgery attempts of the 15 ID's used were collected from each of the 15 invalid users who attempted each of the 15 ID's 15 times. All attempted forgeries were collected in one session for each invalid user. Participants used the system interactively and the results were recorded. The interkey times were collected using the key interrupt facility. The average user ID length was seven characters. The data set was divided into two parts: the training part and testing part. Pattern recognition and neural network [28] techniques were used for the classification process. It was found that hold times are more effective than interkey times and the best identification performance was achieved by using both time measurements. An identification accuracy of 100% (zero false accept and zero false reject) was obtained when the combined hold times and interkey times-based approaches were considered as features using the fuzzy ARTMAP, radial basis function network (RBFN), and learning vector quantization (LVQ) neural network. Other neural network and classical pattern recognition algorithms such as backpropagation with sigmoid transfer function (BP, Sig), hybrid sum-of-products (HSOP), sum-of-products (SOP), potential function, and Bayes' decision rule also gave good accuracy.

The success of this approach was measured mainly in terms of false rejection rate (type I error) and false acceptance rate (type II error), cost of recognition system, and time to access identity verification. The two important measures considered in our work are type I error rate and type II error rate. The false rejection rate (type I error rate) of a verification system gives an indication of how often an authorized individual will not be properly recognized. Type II error describes how often an unauthorized individual will be mistakenly recognized and accepted by the system. It is generally more indicative of the level of a mechanism. This is due to the fact that it describes the degree to which the security measure may be breached by intruders. Type I error is important since it describes the amount of user frustration in using the security system. Our research results have shown that the most successful pattern recognition technique was the potential function followed by the Bayes' rule. The least successful algorithm was the cosine measure. The hold time-based verification/authentication scheme gave better accuracy than the interkey time-based scheme. When neural network paradigms were used for the classification process, it was found that the hold time-based verification/authentication scheme is superior to the interkey time-based scheme. Furthermore, the combined hold and interkey time-based approach gave the least misclassification error. The most successful neural network paradigms for the verification/authentication task are the LVQ, RBFN, and Fuzzy ARTMAP. They basically gave a zero misclassification error for both false acceptance rate and false rejection rate. Figures 10.2-10.7 illustrate these findings.

The average string length used in this recent work was just seven characters. In our previous work [3-8], we obtained lower classification accuracy with a password of 15 characters long. In all the experiments we conducted, it was observed that when

considering hold times alone we obtained better accuracy as compared when interkey times are considered as the only characterizing features. Clearly, hold times are more effective for identification than interkey times. Such results suggest that hold times may in general provide better characterization of the typing skills than the interkey times. Also, we found that the most successful neural network paradigm provides better authentication/verification accuracy than the best classical pattern recognition schemes.

One recent related work was conducted by Robinson et al. [20] in which the authors used key hold times to characterize typing style more effectively. They applied some traditional pattern recognition schemes for the classification procedure. They used hold times and interkey times as features and the best performance was obtained when the inductive learning classifier was used.

6. Conclusions

To conclude, keystroke dynamics are rich with individual mannerism and traits and they can be used to extract features that can be used to authenticate/verify access to computer systems and networks. The keystroke dynamics of a computer user's login string provide a characteristic pattern that can be used for verification of the user's identity. Keystroke patterns combined with other security schemes can provide a very powerful and effective means of authentication and verification of computer users. Neither our work nor any other work we are aware of has dealt with typographical errors. Further research into reliable methods for handling typographical errors is needed in order to make keystroke-based authentication systems non-irritating and widely accepted by the computing and network security community. Finally, it is found that artificial neural network paradigms are more successful than classical pattern recognition algorithms in the classification of users.

References

[1] M. S. Obaidat and B. Sadoun, "Verification of Computer users using Keystroke Dynamics," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 27, No. 2, pp. 261-269, April 1997.

[2] M. S. Obaidat and B. Sadoun, "An Evaluation Simulation Study of Neural Network Paradigm for Computer Users Identification," *Information Sciences Journal-Applications*, *Elsevier*, Vol. 102, No. 1-4, pp. 239-258, November 1997.

[3] M. S. Obaidat, "A Methodology for Improving Computer Access Security," *Computers & Security*, Vol. 12, pp. 657-662, 1993.

[4] 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, April 1993.

[5] S. Bleha and M. S. Obaidat, "Dimensionality Reduction and Feature Extraction Applications in Identifying Computer Users," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 21, No. 2, March/April 1991.

[6] S. Bleha and M.S. Obaidat, "Computer User Verification Using the Perceptron," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 23, N0. 3, pp. 900-902, May/June, 1993.

[7] M. S. Obaidat et al., "An Intelligent Neural Network System for Identifying Computer Users," In *Intelligent Engineering Systems through Artificial Neural Networks* (C. Dagli et al. editors), pp. 953-959, ASME Press, New York, 1991.
[8] M. S. Obaidat and D. T. Macchairolo, "A Multilayer Neural Network System for Computer Access Security," *IEEE Trans. on Systems, Man and Cybernetics*,Vol. 24, No. 5, pp. 806-813, May 1994.
[9] J. A. Adam, "Threats and Countermeasures," *IEEE Spectrum*, Vol. 29, No. 8, 21-28, August 1992.
[10] D. B. Parker, *Computer Security Management*, Reston Publishing Co., Reston, VA, 1981.
[11] W. Satllings, *Network and Internetwork Security*, Prentice Hall, Upper Saddle River, NJ, 1995.
[12] C. P. Pfleeger, *Security in Computing*, Prentice Hall, Upper Saddle River, NJ, 1997.
[13] D. Denning, *Cryptography and Data Security*, Addison-Wesley, Reading, MA, 1983.
[14] B. Walker, *Computer Security and Protection Structures*, Dowden, Hutchinson and Ross Inc., 1977.
[15] D. Umphress and G. Williams, "Identity Verification Through keyboard Characteristics," *International Journal Man-Machine Studies*, Vol. 23, pp. 263-273, Academic Press, 1985.
[16] W. L. Bryan and N. Halter, "Studies in the Physiology and Psychology of the Telegraphic Language," *The Psychology of Skill: Three Studies*, (E. H. Gardener and J. K. Gardner, editors), pp. 35-44, NY Time Co., NY 1973.
[17] S. Card, T. Moran, and A. Newell, "The Keystroke Level Model for User Performance Time with Interactive Systems," *Communications of ACM*, Vol. 23, pp. 396-410, 1980.
[18] L. H. Shaffer, "Latency Mechanisms in Transcription," *Attention and Performance*, Vol. IV, (S. Kornblum, editor), Academic Press, 1973.
[19] W. E. Cooper, *Cognitive Aspects of Skilled Typewriting*, pp. 29-32, Springer-Verlag, 1983.
[20] J. Robenson, V. Liang, J. Chambers, and C. MacKenzie, "Computer User Verification Using Login String Keystroke Dynamics," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 28, No. 2, pp. 236-244, March 1998.
[21] R. Joyce and G. Gupta, "Identity Authentication Based on keystroke Latencies," *Communications of ACM*, Vol. 33, No. 2, pp. 168-176, February 1990.
[22] R. Gaines, W. Lisowski, S. Press, and N. Shapiro, "Authentication by Keystroke Timing: Some Preliminary Results," *Rand Report, R-256*, NSF, Rand Corp., Santa Monica, CA, 1980.
[23] J. Garcia, "Personal Identification Apparatus," *Patent No. 4,621,334,* U.S. Patent and Trademark Office, Washington, DC, 1986.
[24] L. Leggett, G. Williams, and D. Umphress, "Verification of User Identity via Keyboard Characteristics," In *Human Factors in Management Information Systems*, (J. Carey, editor), Ablex, Norwood, NJ, 1988.
[25] L. Leggett and G. Williams, "Identity Verification Through Keyboard Characteristics," *International Journal Man-Machine Studies*, Vol. 23, No. 3, pp. 263-273, September 1985.
[26] J. R. Young and R. W. Hammon, "Method and Apparatus for Verifying and Individual's Identity," *Patent No. 4, 805,222*, U.S. Patent and Trademark Office, Washington, DC, 1989.
[27] M. S. Obaidat and B. Sadoun," HSOP: A Neural Network Paradigm and Its Applications," *Neural Computing & Applications Journal*, Springer, Vol. 2, No. 2, pp. 89-96, 1994.
[28] *NeuralWorks Reference Guide*, NeuralWare, Inc., Pittsburgh, PA, 1993.

Training Set			
File Name	*Patterns per Class*	*Raw Patterns per Average Pattern*	*Odd/Even Half*
all.pat	40	1	All
Odd20	1	20	Odd
Even20	1	20	Even
Odd5	4	5	Odd
Even5	4	5	Even
Odd2	10	2	Odd
Even2	10	2	Even

Table 10.1 An example of a training set used.

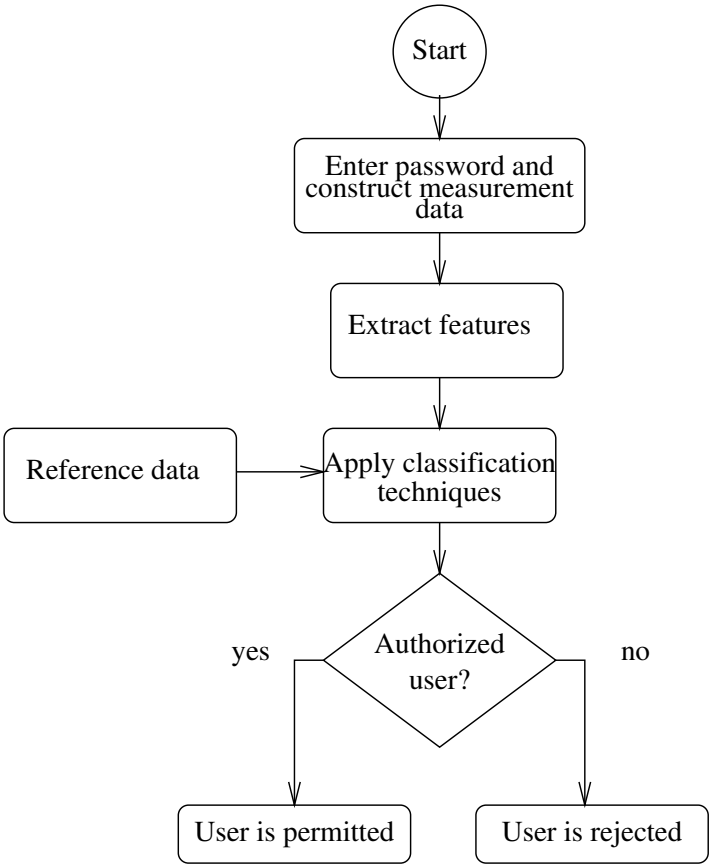

Figure 10.1 Flowchart of the overall steps of a computer verification system.

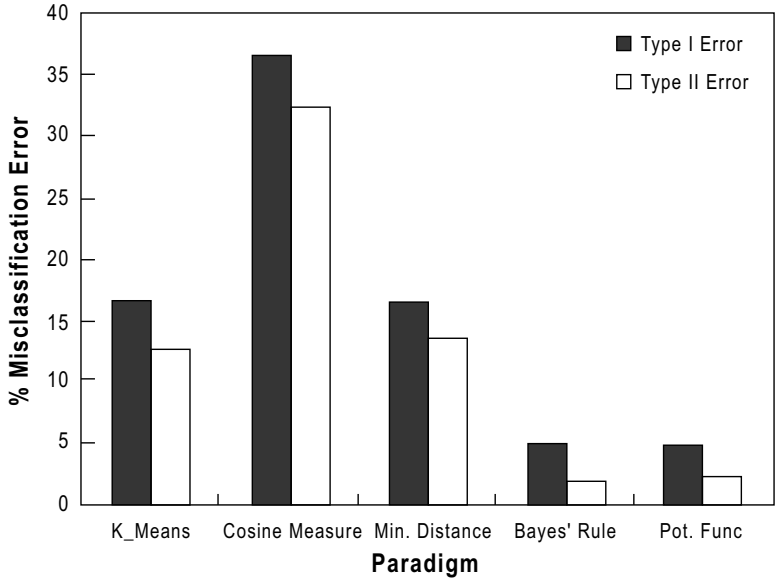

Figure 10.2 Interkey time-based results using pattern recognition techniques.

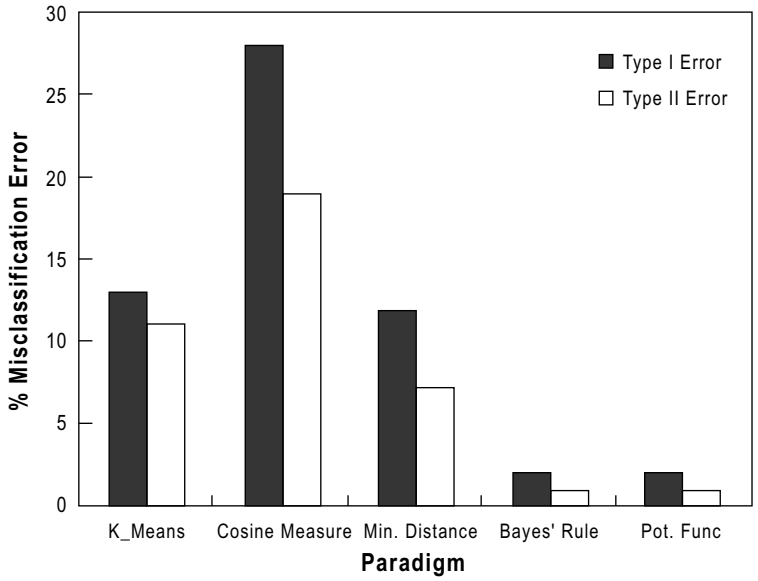

Figure 10.3 Hold time-based classification results using pattern recognition techniques.

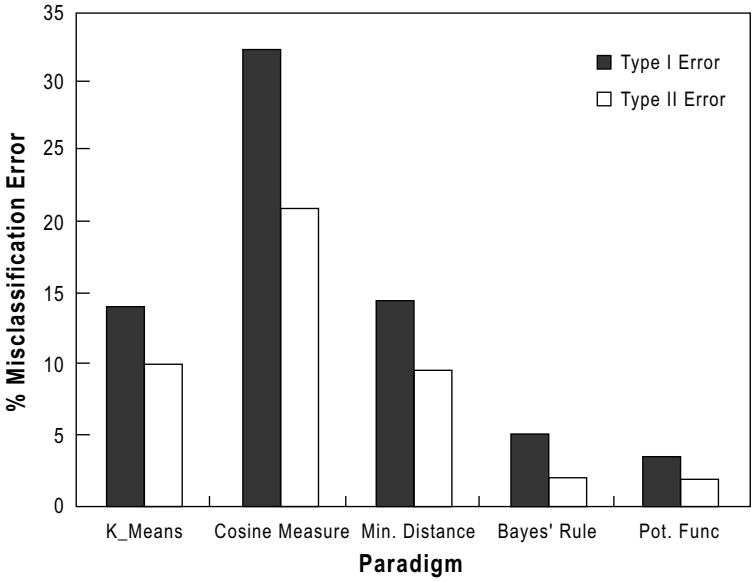

Figure 10.4 Combined interkey and hold time-based classification results using pattern recognition techniques.

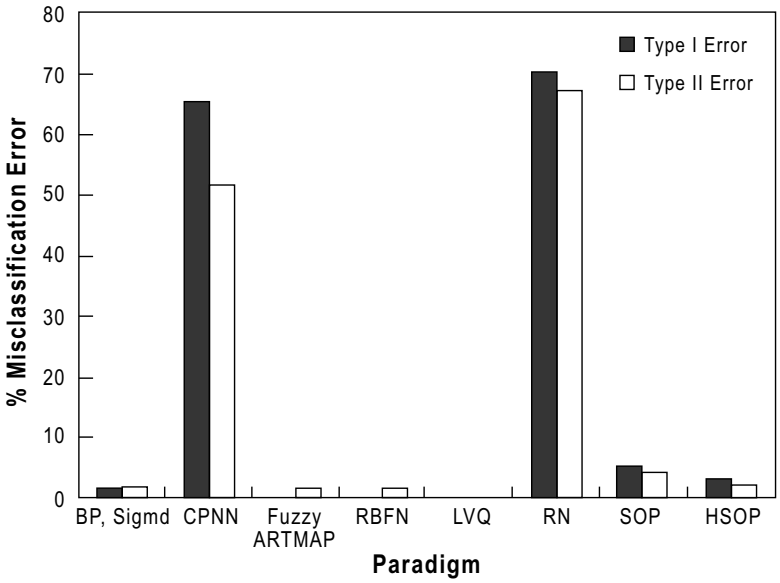

Figure 10.5 Interkey time based classification results using neural network techniques.

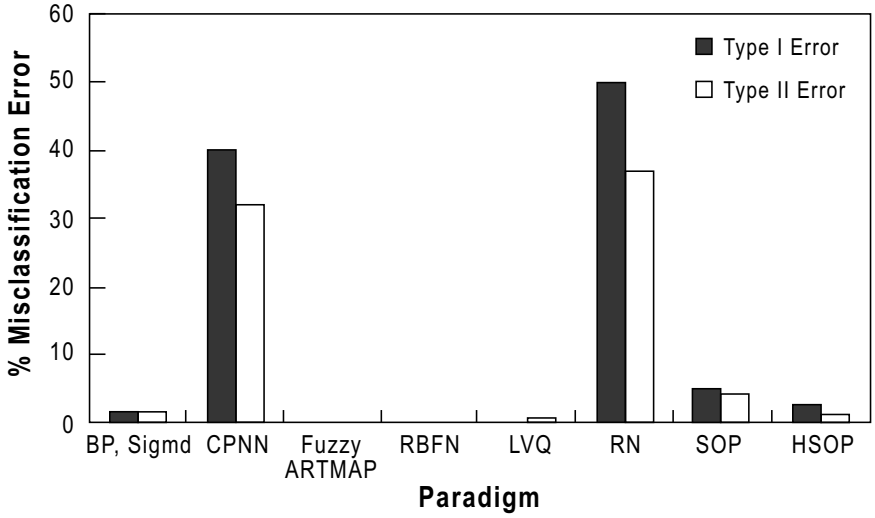

Figure 10.6 Hold time-based classification results using neural network techniques.

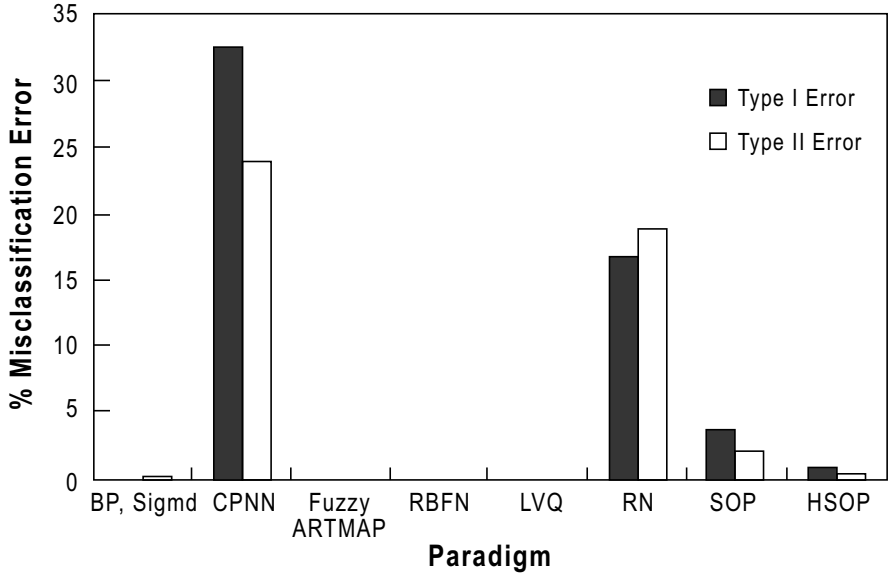

Figure 10.7 Combined interkey and hold time-based classification results using neural network techniques.