Journal of Organizational Computing and Electronic Commerce, Vol. 10, No. 4, pp. 295-307, 2000.

Web based Keystroke Dynamics Identity Verification using Neural Network

Sungzoon Cho, Chigeun Han, Dae Hee Han, Hyung-Il Kim

Abstract

Password typing is the most widely used identity verification method in World Wide Web based Electronic Commerce. Due to its simplicity, however, it is vulnerable to imposter attacks. Keystroke dynamics and password checking can be combined to result in a more secure verification system. We propose an autoassociator neural network that is trained with the timing vectors of the owner's keystroke dynamics and then used to discriminate between the owner and an imposter. An imposter typing the correct password can be detected with very high accuracy using the proposed approach. This approach can be effectively implemented by a Java applet and used in the World Wide Web.

I. INTRODUCTION

In recent developments in Electronic Commerce (EC), the World Wide Web (WWW) has emerged as the implementation platform of choice thanks to its simplicity and low cost. Since the WWW was developed as a means of sharing information among computers scattered around the world, it tends to lack sophisticated security methods. Thus, it is not suitable for certain commercial transactions that require a secure communication channel. Superior security measures need to be developed to further an even wider acceptance of the WWW as an EC platform.

User authentication is a particular aspect of security relevant to EC. It is concerned with verifying claimed identity. Several methods have been proposed for use on the WWW, such as user IDs and passwords, IP addresses, and message digest authentication [1]. Additional ideas such as channel-based, content-based and message-based methods all use hypertext transfer protocol (HTTP). Although the password approach is the most widely used, as well as being the simplest and least expensive tool, it has loopholes because people tend to choose as passwords such easy-to-guess words and/or numbers as the names of family members, birthdays, phone numbers, addresses, etc. The result is a security failure. Some other means should be devised which replaces or consolidates the password approach.

One approach that is both inexpensive and simple takes advantage of the uniqueness of keystroke dynamics. When a user types a word, for instance a password, the keystroke dynamics can be characterized by a “timing vector”, consisting of the duration of keystrokes and the time interval between them. A word of n characters followed by “Return” results in a timing vector of dimension $2n+1$. The owner's timing vectors are collected and used to build a model that discriminates between the owner and imposters. This idea is low cost and causes no user discomfort. Its only disadvantage has been a relatively low rate of accuracy. Previous studies have reported error rates much larger than 10%, which is practically unacceptable.

In this paper, we propose an autoassociator neural network model that reduces error rates significantly. Timing vectors from an

owner were collected and used to build a neural network model that outperformed a conventional Nearest Neighbor (*k*-NN) approach. Although experiments involving many more owners are required for practical use of this approach, the preliminary results are the best ever reported to the authors' knowledge. It is possible to implement this approach in the WWW environment using Java applets.

This paper is structured as follows. First, various system security measures are described. Then the typing dynamics based verification method is presented, followed by the results of previous research. The neural network based novelty detector is proposed, and then data collection and experimental results follow. After a Java applet implementation on the WWW is described, a summary and discussion of ongoing and future research issues conclude this paper.

II. System Security Measures

System security is usually maintained by system access control. User authentication is one way to achieve it. Table I lists several approaches which have been proposed and used [2].

TABLE I

USER AUTHETICATION APPROACHES

The ownership-based approaches are the oldest, but are vulnerable to loss or theft. The use of a password is the most popular approach in computer access security now thanks to simplicity, effectiveness, convenience and low cost. A user is expected to choose a hard-to-guess password and to change it frequently. Limitations related to memory and habit undermine both expectations, however, resulting in security breaks. The biometric-based approaches are free from loss, theft or memory problems. But they are not perfect, and involve two types of errors. False accept rate (FAR) denotes the rate that an imposter is allowed access. False reject rate (FRR) denotes the rate that the legitimate user is denied access. Various approaches can be quantitatively evaluated in terms of processing time, cost, and user acceptability as well as error rates [2] as shown in Table II. The retinal pattern based method is the most accurate, but the most expensive and least acceptable at the same time. The fingerprint method and the hand shape method suffer from similar problems.

TABLE II

COMPARISONS OF BIOMETRIC-BASED USER AUTHENTICATION METHODS

III. Typing Dynamics based User Verification

When one types a phrase or a password on a keyboard, the typing dynamics or timing pattern can be measured and used for identity verification. More specifically, a timing vector consists of the keystroke duration times interleaved with the keystroke interval times. If one types a password of seven characters, a 13-dimensional timing vector results which consists of seven keystroke duration times and six keystroke interval times. Two more elements can be added to the vector if ENTER key information is considered. Figure 1 shows the timing vector when “ABCD” is typed. An actual example of 15-dimensional timing vector from a seven character-long password is [120,60,120,90,120,60,150,-60,120,-30,120,-60,120,120,90,60,150]. The time unit is in milliseconds. Negative interval times result when the next key is pressed before a previous key is released.

Figure 1. Timing vector corresponding to “ABCD”

It has been shown that an individual has characteristic and distinctive typing dynamics. A pattern classifier can be built which distinguishes an individual's typing dynamics from those of others (see Figure 2). Combined with a simple password scheme, the typing dynamics based identity verification provides an additional layer of protection with a negligible increase in cost and processing time.

Figure 2. Typing Dynamics based Identity Verification

IV. PREVIOUS RESULTS

Here, we review typing dynamics based user verification methods that have been proposed in the past. All biometrics-based approaches have two types of errors, the false accept rate (FAR) and the false reject rate (FRR). Since one type of error can be reduced at the expense of the other, an appropriate middle point is usually used as a threshold based on the relative cost of the errors. A different choice of threshold results in different FAR and FRR values. In this paper, we employ a widely used error measure “FRR when FAR = 0”, i.e. FRR when we set the threshold such that FAR becomes zero.

In the past, a short character string such as a password was regarded as inadequate for user authentication [3]. A long string of 537 characters, for example, had to be employed to achieve 5.0% FAR and 5.5% FRR [4]. Only recently through the use of neural networks, a comparable performance of 12% to 21% was achieved using short strings such as real life names [5]. These error rates are still too high to be practically acceptable. In addition, the neural network was trained in advance not only with the owner's timing vectors but also with those of imposters. In real life situations, this is unacceptable because the owner's password has to be revealed to users at large. In the late 80's, two US patents were granted for statistical approaches [6],[7], but performance results are not available.

A lower error rate of 2.5% was obtained when the user identification problem was solved [8]. This problem involves finding out who, among several candidates, typed the password rather than determining whether the timing vector is that of the owner. The network had to be trained with the timing vectors of all candidates. Unfortunately, the result cannot be applied to the user authentication problem. Also, a 0% error rate was recently reported in user verification using 7 character-long login names [9]. However, negative examples (i.e., intruder's typing patterns) as well as positive examples (i.e., owner’s patterns) were used for training, and the training data set was much larger (6,300 positives and 112 negatives). Also, the training and test patterns were not chronologically separated.

V. AUTOASSOCIATIVE MLP NOVELTY DETECTOR

User authentication is challenging from a pattern classification viewpoint. It is a two class (owner vs. imposters) problem, yet the patterns from only one class, the owner’s are available in advance. Since there are millions of potential imposters, it is not practical to obtain enough patterns from all kinds of imposters. Also, it is not desirable to publicize one's password in order to collect potential imposters' timing vectors. The only solution is to build a model of the owner's keystroke dynamics and use this to detect imposters using some sort of a similarity measure. This type of problem has become known as a “partially exposed environment” [10] or “novelty detection”. A similar situation arises in fault diagnosis, while conditions are usually normal when information on abnormal

conditions is necessary. Usually, a model of normal conditions is built and then used to detect abnormality or novelty.

We propose to use a multilayer perceptron (MLP) for identity verification. An MLP (Figure 3) is a graphical computational model originally inspired by information processing in brains.

Figure 3. Multilayer Perceptron

A unit or node models a neuron while a connection or edge models a synapse. Units are located in three different layers, namely input, hidden and output. Each unit receives input from all the units in the previous layer, computes its output, and sends it to all the units in the next layer. Specifically, the output x_i of unit i is computed as follows.

$$x_i = f(\sum_j w_{ij})$$

where

$$f(z) = \frac{1}{1 + \exp(-z)}.$$

The value w_{ij} denotes the connection strength from unit j to unit i. The MLP is known to be a universal function approximator, i.e. able to fit any continuous function with a desired accuracy if there are enough training data available [11].

Given a set of input-output pairs, an MLP is to be found which maps the data, i.e. which produces the target output vector when presented with an input vector. This process of finding model parameters, w_{ij} in this case, is called learning. A general learning algorithm for MLP is known as backpropagation [12]. Since we want the network's output x_l^p (output value for l-th output variable for p-th input pattern) to be as close as possible to target value t_l^p, an error measure $E = \sum_p \sum_l (x_l^p - t_l^p)^2$ has to be minimized. When a gradient descent method is applied, we get the following iterative procedure for w_{ij}.

$$\Delta w_{ij} = -n \frac{\partial E}{\partial w_{ji}}$$
$$= \eta \delta_i x_j$$

where

$$\delta_i = \sum_k \delta_k w_{ki}.$$

Specifically, first, we randomly initialize the weight vector. Then, we repeat the procedure until a local minimum is reached, i.e. no more decrease in the error function. Several runs usually result in a local minimum whose quality is close to that of a global minimum in practice. Numerous applications of MLP trained with backpropagation have been reported [11].

An autoassociative MLP is an MLP where input vectors are also used as targets during training. The network is then forced to somehow encode the input vector in the hidden layer and then decode it back in the output layer. We use this model in identity verification as follows. The owner's patterns are used to train the network to become an autoassociator by employing a timing vector

as both an input and output. The MLP is trained to learn to encode certain properties only present in the owner's timing vectors at the hidden layer. When a previously unseen timing vector for the owner arrives, the network will output a vector that is reasonably close to the input. When an imposter's pattern arrives, however, the network will output a vector that is far from the input. We then use this property in distinguishing a pattern as either genuine or forged. That is, a timing vector *X* is classified as the owner's if, and only if,

$$\|X - M(X)\| < \varepsilon \tag{1}$$

where *M(X)* and ε represent the MLP's output for *X* and a threshold. Recently it was shown that an autoassociator with a nonlinear hidden transfer function such as a step or a sigmoid performs the necessary classification [13]. In particular, it was found that the trained autoassociator partition the input training patterns into several subsets, and that the input patterns in each subset are mapped to its center. Any test pattern that is far from a training pattern is also mapped to one of centers, thus the distance in Equation 1 becomes likely to be larger than ε. On the other hand, any test pattern that is not far from a training pattern will have a small distance, thus becoming correctly classified.

The proposed autoassociative MLP approach was compared with a more conventional Nearest Neighbor (NN) approach. When a new timing vector arrives, the average "distance" to the *k* closest training patterns is computed. If the average distance is smaller than a predetermined threshold, the timing vector is classified as that of the owner. Otherwise, it is classified as that of an imposter. The distance between two vectors is defined as $(\vec{x}-\vec{y})^T M(\vec{x}-\vec{y})$. In order to give more weight to those elements with a smaller variance, the inverse of a covariance matrix Σ can be used for *M*, which results in the so-called Mahalanobis distance. It has been shown that the Nearest Neighbor method's error is no larger than twice that of the optimal Bayesian classifier [14]. Thus, the method is widely used in practice.

VI. Experimental Results

A program was developed to measure keystroke duration times and interval times in X window environment on a Sun Sparcstation. A PC version was also developed, but was not used in this experiment. For instance, a password of 7 characters results in a timing vector of dimension 15 since the duration of striking the "Enter" key is also included. An example of a timing vector is [120, 60, 120, 90, 120, 60, 150, -60, 120, -30, 120, -60, 120, 120, 90, 60, 150] where each element was measured in milliseconds. A negative interval time results from a situation where the next key is stroked before the previous key is released.

A total of 25 subjects were asked to come up with a new password (see Table III). Each password was of length 7, but in general, it could be of any length. A longer password will simply require a neural network with more input and output units. Each subject or owner typed this password 150 to 400 times during a period of several days, and the last 75 timing vectors collected were set aside for testing. The remaining timing vectors were used to train the network. If any of its elements was larger than the upper 10%, however, the vector was classified as an outlier and discarded. Depending on the owner, 6 to 50% of the training vectors were discarded. There were four owners whose discard rates were higher than one third. A high discard rate implies that the owner did not consistently type the new password. Since we only consider experienced owners, we removed those four owners from the experiment. A total of 15 imposters were given all the 21 passwords and asked to type each password five times, resulting in 75 imposter test vectors for each password. Combined with the owner's 75 test vectors previously set aside, a total of 150 test vectors per password were obtained.

Table III shows, for all 21 owners, the respective password, the number of training patterns and the discard rate. Some passwords, such as 5, 6 and 8, are words in Hangul, the Korean alphabet. We simply show the corresponding English characters.

TABLE III

PASSWORD CHARACTERISTICS AND ERROR MEASURES FROM RESPECTIVE MODELS

Also shown in Table III are the error rates for the *k*-NN and MLP approaches. The error is the False Reject Rate (FRR) when the False Accept Rate (FAR) was reduced to zero. For *k*-NN, we tried 1, 2, and 3 as *k* values and obtained the best result with *k*=1, which is shown here. Each MLP contains the same number of hidden units as input units. All 21 MLPs were trained with a standard backpropagation algorithm, with a learning rate of 0.1 and a momentum term of 0.3, for 500 epochs. The proposed MLP approach clearly outperformed the *k*-NN. A perfect authentication was achieved for 13 owners. The worst performance was from owners 12 and 13, with an error rate of 4.0%. The average error rate was 1.0%. The paired comparison hypothesis test was performed with $H_0 : \mu_d = 0$ and $H_1 : \mu_d > 0$ where random variable *D* denotes the difference of error rates from two algorithms, i.e. $e_{1-NN} - e_{MLP}$. Assuming both error values are from a normal distribution, $\bar{d} / (s_d / \sqrt{n})$ follows a *t*-distribution with a degree of freedom of 20. Since the *t* statistic value of 3.656 is larger than 2.528 = $t_{0.01}$, H_0 is rejected with a 99% confidence. In conclusion, the superiority of MLP approach's performance is statistically significant.

The MLP approach's performance advantage is obvious in Figures 4 and 5. The histogram distributions are from owner 2. A total of 150 test patterns, one half from the owner and the other from 15 different imposters, were presented to two respective models. For 1-NN, the average distance from the nearest neighbor was computed. For MLP, the generalization error was computed. The resulting histograms show why the MLP gives perfect authentication (no overlap between owner and imposter test vector histograms) while the 1-NN shows 30% error (a significant overlap).

Figure 4. Histograms of average distance measure (1-NN). The better separated the owner and imposter populations are, the better the classification is.

Figure 5. Histograms of generalization error (MLP). The better separated the owner and imposter populations are, the better the classification is.

When the proposed neural network approach is to be used in actual applications, a few issues have to be resolved. First, how one obtains training data in a real situation? Obviously, a separate data-collecting module has to be built. During the data collection period, right after a new password is registered, the proposed identity verification can not be used. However, an ordinary level of security can be maintained with the conventional password security system. The length of the collection period can be dynamically determined by monitoring the variability of typing patterns. This data collection overhead is common among all dynamic biometrics based approaches including the online signature based identity verification.

Second, for each password or user, a separate MLP has to be constructed. Also, whenever a user changes his or her password, a new MLP has to be built. The problem of finding the appropriate number of hidden units, or model selection, is not straightforward.

Trial and error is usually employed. However, it is not much of a concern here since the autoassociative MLP used here employed the same number of hidden units as input and output units for all 21 cases. It was automatically set, thus the network building cost can be minimized.

Third, the performance measure we used in the paper is FRR when FAR is set to zero. Thus, we set threshold ε in Equation 1 such that FAR becomes zero. Then, we measured FRR with that specific threshold. In practice, however, ε has to be set to a fixed value. It controls the two types of errors. A large ε leads to a high FAR while a small value leads to a high FRR. An empirical investigation is necessary considering the nature of the application. Even some trial and error is unavoidable.

VII. WWW IMPLEMENTATION

Figure 6 shows a simplified diagram of how the proposed scheme can be implemented over a network in the WWW. There are three different ways to implement the typing dynamics, namely Plug-in, Active-X and Java applet. The Plug-in approach is expensive since actual implementation depends on the type of web browser and operating systems involved. The Active-X approach makes it possible to take advantage of already developed Component Object Model (COM) objects in Windows. The downside is that the resulting system operates only in a Windows environment. Finally, the Java applet approach is inexpensive since a single implementation works for different environments as long as a web browser (see Figure 7) supports Java Virtual Machine. Also, unlike the Plug-in approach, it is not necessary to install anything in a client machine in advance. When the Java applet code is loaded by the web browser security environment such as Secure Sockets Layers (SSL), it can use the additional package support to communicate with the CGI in the secure sockets layer because the standard Java package does not provide the SSL [15]. For these reasons, we have chosen the Java applet approach in this work. Comparison of the three methods is summarized in Table IV.

Figure 6. Client-server implementation structures in WWW

Figure 7. Security environment for the Java applet approach

TABLE IV

COMPARISON OF PLUG-IN, ACTIVE-X, AND JAVA APPLET

Conceptually, the Java applet approach works as follows. When a client tries to access a homepage, for example, say a firm's on-line shop, located in a server, the user types the already registered user ID. Then the server sends the client a Java applet code that can measure the user's password keystroke timing vector. Once the Java applet running in the client system gathers the user's keystroke timing vector, it sends it back to the server. Then the autoassociative neural network located in the server can verify whether the user is the person he or she claims to be. Since the code is programmed in Java, any client system that has a Java browser can be connected to the server.

More specifically, the procedure can be described in four steps (see Figure 8). First, the Java applet byte code stored in the web server is downloaded onto the web browser. Second, the applet receives user ID and password information (characters and timing vectors) through TextField object and send them back to Common Gateway Interface (CGI) which is the authentication module in the

web server. Third, if the password information is classified as authentic, a first page is sent back to the web browser along with a cookie. From then on, URL requests are issued through the cookie.

Figure 8. Step-by-step WWW implementation procedure

VIII. CONCLUSIONS

An MLP-based novelty detector is proposed for user authentication using keystroke dynamics. An autoassociative MLP is built from a set of timing vectors previously collected from the owner. When a new timing vector arrives, it is presented to the MLP, and output is computed. If the output is close enough to input, the input timing vector is classified as that of the owner. If not, it is classified as that of an imposter. The experimental results involving 21 skilled users show that the proposed approach is significantly more effective than the *k*-NN approach. For 13 owners, the MLP approach achieved perfect authentication. Among the rest, the worst performance was a 4% error rate. The overall average error rate was 1%. The preliminary result reported here is quite promising. The proposed approach was also implemented on the World Wide Web, proving that the scheme can be used for electronic commerce applications.

Further investigation is necessary in the following areas: First, many more experiments involving human subjects must be conducted. Such issues as typing inexperience, learning effects, and fatigue must also be considered. Second, we must investigate how to make the number of necessary training patterns as small as possible. Finally, a variety of preprocessing and feature extraction algorithms must be examined.

ACKNOWLEDGEMENTS

This research was supported by grants to the first author from the Brain Science and Engineering Research Program sponsored by the Korean Ministry of Science and Technology and Brain Korea 21 Project. The authors would like to thank two anonymous reviewers for valuable comments.

REFERENCES

[1] J. Park, S. Kang, and S. Park, "Trends in world wide web security technology", *Korea Information Science Society Review*, vol. 15, no. 4, pp. 37-44, 1997.

[2] D. Davis and W. Price, *Security for Computer Networks*, John Wiley & Sons, Inc., 1989.

[3] G. Forsen, M. Nelson, and R. Staron, "Personal attributes authentication techniques", in *Rome Air Development Center Report RADC-TR-77-1033*, A. Griffs, Ed., New York:RADC, 1977.

[4] J. Leggett, G. Williams, M. Usnick, and M. Longnecker, "Dynamic identity verification via keystroke characteristics", *International Journal of Man-Machine Studies*, vol. 35, pp. 859-870, 1991.

[5] M. Brown and S. J. Rogers, "User identification via keystroke characteristics of typed names using neural networks", *International Journal of Man-Machine Studies*, vol. 39, pp. 999-1014, 1993.

[6] J. Garcia, "Personal identification apparatus", *Patent No. 4,621,334*. U.S. Patent and Trademark Office, Washington D.C. 20231, 1986.

[7] J. Young and R. Hammon, "Method and apparatus for verifying an individual's identity", *Patent No. 4,805,222*. U.S. Patent and Trademark Office, Washington D.C. 20231, 1989.

[8] M. S. Obaidat and D. T. Macchairolo, "A multilayer neural system for computer access security", *IEEE Transactions on Systems, Man, and*

Cybernetics, vol. 24, no.5, pp. 803-816, May 1994.

[9] M. Obaidat and S. Sadoun, "Verification of computer users using keystroke dynamics", *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, vol. 27, no. 2, pp. 261-269, 1997.

[10] B. V. Dasarathy, "Recognition under partial exposure and imperfect supervision", in *Proceedings of the International Conference on Cybernetics and Society*, pp. 218-221, October 1979.

[11] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[12] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error backpropagation", in *Parallel Distributed Processing*, vol. 1, pp. 318-362, 1986.

[13] B. Hwang and S. Cho, "Output characteristics of autoassociative MLP and its application in novelty detection", in *Proceedings of Korea Information Science Society*, vol. 25, no. 11, pp. 581-583, 1998.

[14] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley, 1973.

[15] Phaos Company, "SSlava Toolkit Overview", http://www.phaos.com/products/sslavatk.ht, 1998.

Sungzoon Cho received the B.S. and M.S. degrees in Industrial Engineering from Seoul National University in 1983 and 1985, respectively. And, he received the M.S. and Ph.D degrees in Computer Science from University of Washington in 1997 and from University of Maryland in 1992, respectively. He has taught from 1993 to 1998 at Pohang University of Science and Technology in Pohang, Korea. He currently teaches at Industrial Engineering in Seoul National University, Seoul, Korea. His research interests are neural networks and machine learning algorithms for data mining.

Chigeun Han received the B.S. and M.S. degrees in Industrial Engineering from Seoul National University in 1983 and 1985, respectively. And, he received the M.S. and Ph.D. degrees in Computer Science from the Pennsylvania State University in 1988 and 1991, respectively. In 1992, he joined Computer Engineering Dept. in Kyung Hee University and is working as an associate professor in the department. His research interests are graph theory, combinatorial optimization, and network design.

DaeHee Han received the B.S. and M.S. degrees from the Pohang University of Science and Technology in 1995 and 1997, respectively. In 1997 he joined SK Telecom Co., Ltd. and now works as a research engineer. He is involved in developing a system for Short Message Service in the CDMA system. He is generally interested in finding the way the brain thinks and learns, and applying it to a computer system.

Hyung-Il Kim received the B.S. in Physics and M.S. in Computer Engineering from Kyung Hee University in 1994 and 1996, respectively. He is currently pursuing Ph.D. in Computer Engineering in the same university. His research interests are real-time OS, Fault-tolerant system, OS scheduling.

Journal of Organizational Computing and Electronic Commerce, Vol. 10, No. 4, pp. 295-307, 2000.

[1] S. Cho is with the Department of Industrial Engineering, Seoul National University, San 56-1, Shinrimdong, Seoul 151-742, Korea. Tel: +82-2-880-6275, Fax: +82-2-889-8560, E-mail: zoon@snu.ac.kr
[2] Department of Computer Engineering, Kyung Hee University, Sochonri, Kiheung, Yongin 449-701, Kyunggi, Korea. E-mail: cghan@nms.kyunghee.ac.kr
[3] SK Telecommunications, Hwaamdong, Yoosunggu, Daejon, 305-348, Korea. E-mail: handol@sktelecom.com
[4] Department of Computer Engineering, Kyung Hee University, Sochonri, Kiheung, Yongin 449-701, Kyunggi, Korea.

Journal of Organizational Computing and Electronic Commerce, Vol. 10, No. 4, pp. 295-307, 2000.

TABLE I

USER AUTHETICATION APPROACHES

Approach	Examples
Ownership-based	Key, Card
Knowledge-based	Password, PIN
Biometric-based	(static) Fingerprint, Hand Shape, Retinal Pattern (dynamic) Signature, Typing Dynamics

TABLE II

COMPARISONS OF BIOMETRIC-BASED USER AUTHENTICATION METHODS

Authentication Method	FAR (%)	FAR (%)	Time (sec)	Cost ($)	User Acceptability
Fingerprint	0.001	0.5	4	4,000	Low
Hand Shape	1	1 – 3	6	3,500	Medium
Retinal Pattern	0.0001	5	2	6,000	Low
Voice	0.1 - 2	0.25 – 5	3	1,000 – 1,500	High
Signature	3	0.7	5	700 – 1,300	High

TABLE III

PASSWORD CHARACTERISTICS AND ERROR MEASURES FROM RESPECTIVE MODELS

Owner ID	Password	Number of Training Patterns	Discard Rate	FRR when FAR = 0	
				1-NN	MLP
1	loveis.	207	0.21	22.7	2.7
2	i love 3	330	0.15	30.7	0.0
3	autumnman	111	0.10	0.0	0.0
4	90200jdg	164	0.10	5.3	0.0
5	rla sua	101	0.18	8.0	1.3
6	dhfpql.	232	0.08	17.3	2.7
7	love wjd	101	0.19	54.7	0.0
8	dltjdgml	151	0.14	0.0	0.0
9	dusru427	365	0.27	0.0	0.0
10	manseiii	86	0.25	60.0	1.3
11	rhkdwo	205	0.20	18.7	0.0
12	beaupowe	76	0.24	9.3	4.0
13	tdwnsl1	108	0.18	17.3	4.0
14	yuhwa1kk	388	0.12	0.0	0.0
15	anehwksu	319	0.10	10.7	0.0
16	tjddmswjd	337	0.10	33.3	0.0
17	drizzle	299	0.10	9.3	1.3
18	dlfjs wp	342	0.06	1.3	0.0
19	c.s.93/ksy	200	0.22	17.3	2.7
20	dirdhfmw	309	0.33	89.3	0.0
21	ahrfus88	260	0.20	5.3	0.0
Avg.		223	0.17	19.5	1.0
Min.		76	0.06	0.0	0.0
Max.		388	0.33	89.3	4.0

TABLE IV

COMPARISON OF PLUG-IN, ACTIVE-X, AND JAVA APPLET

	Plug-in	Active-X	Java applet
Platform	UNIX, Windows	Windows	UNIX, Windows
Language	C/C++, Basic	MFC	Java
Cost	High	Medium	Low
Prior additional Installation	Yes	No	No

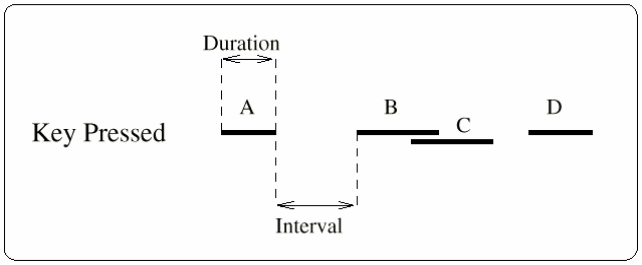

Figure 1. Timing vector corresponding to “ABCD”

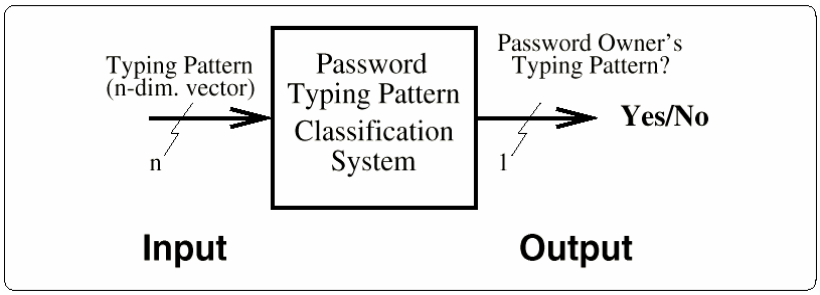

Figure 2. Typing Dynamics based Identity Verification

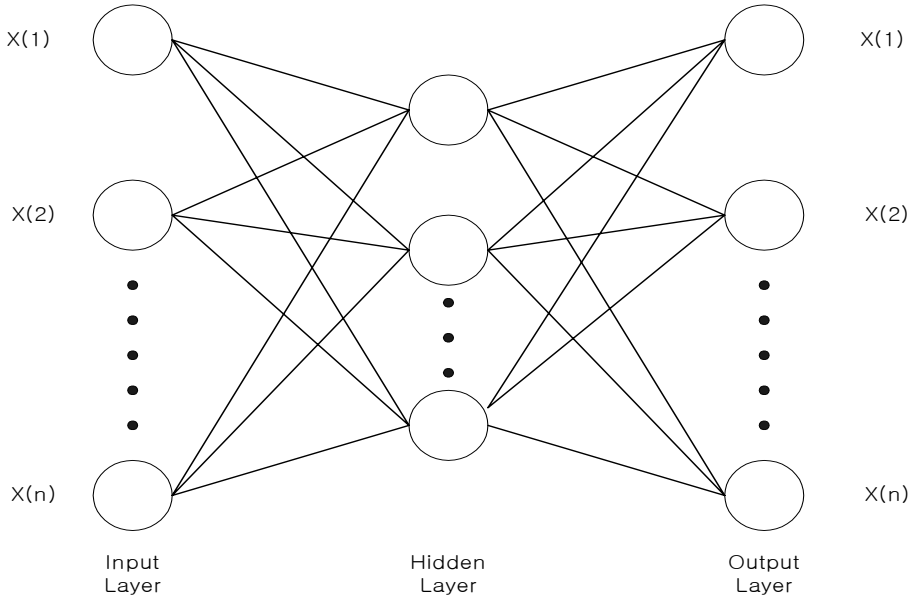

Figure 3. Multilayer Perceptron

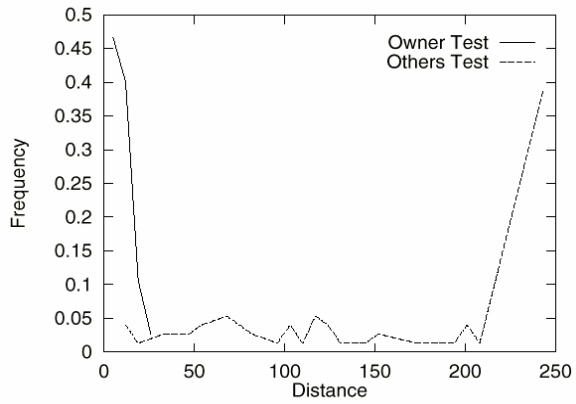

Figure 4. Histograms of average distance measure (1-NN). The better separated the owner and imposter populations are, the better the classification.

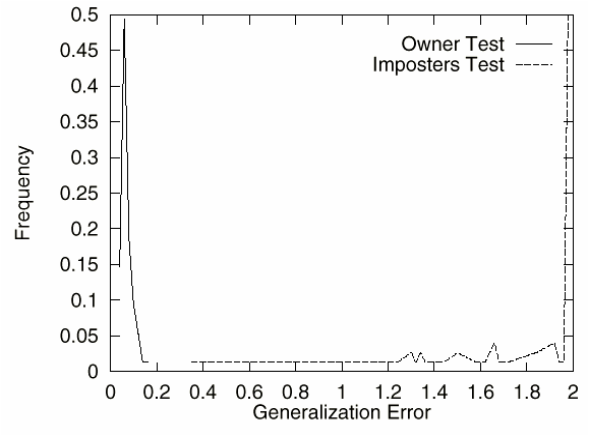

Figure 5. Histograms of generalization error (MLP). The better separated the owner and imposter populations are, the better the classification is.

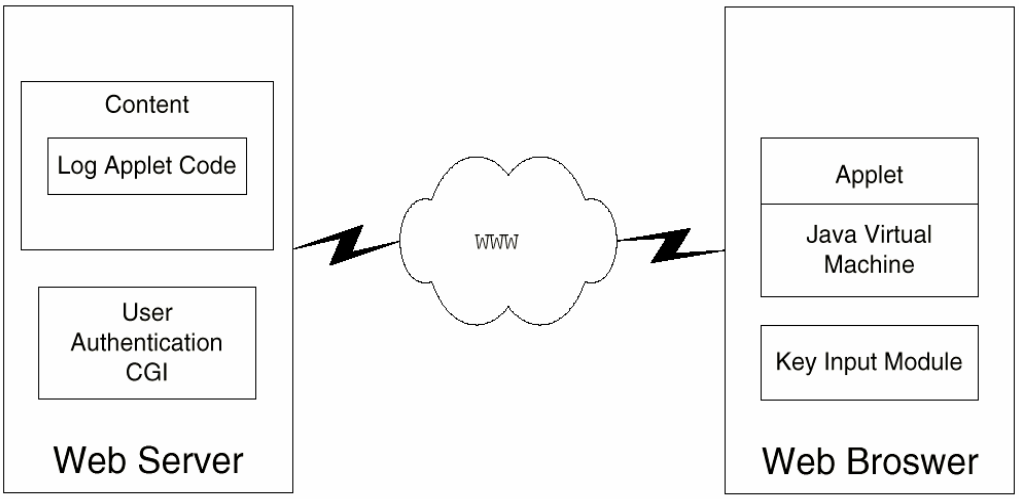

Figure 6. Client-server implementation structures in WWW

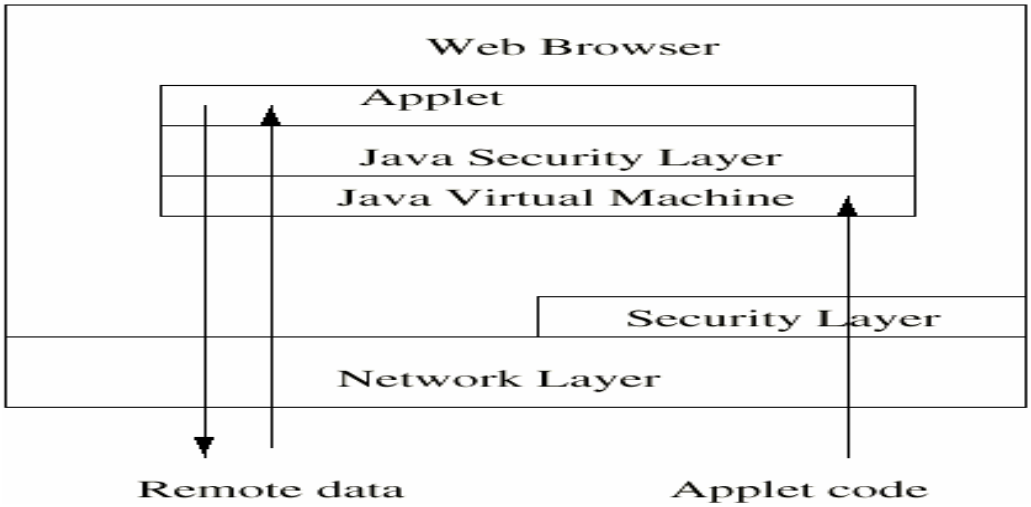

Figure 7. Security environment for the Java applet approach

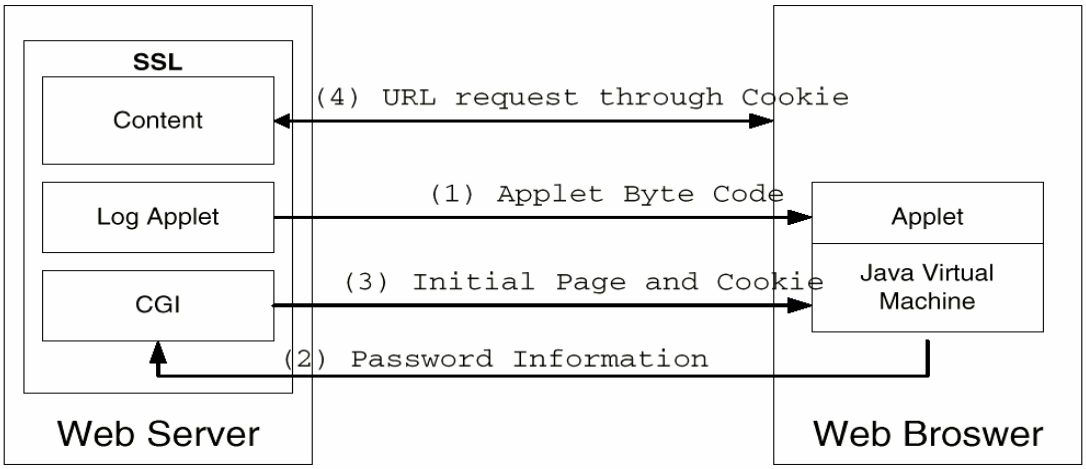

Figure 8. Step-by-step WWW implementation procedure

TABLE I

USER AUTHETICATION APPROACHES

TABLE II

COMPARISONS OF BIOMETRIC-BASED USER AUTHENTICATION METHODS

TABLE III

PASSWORD CHARACTERISTICS AND ERROR MEASURES FROM RESPECTIVE MODELS

TABLE IV

COMPARISON OF PLUG-IN, ACTIVE-X, AND JAVA APPLET