A Remote Password Authentication Scheme for Multi-Server Architecture Using Neural Networks *†

Li-Hua Li † Iuon-Chang Lin ‡ Min-Shiang Hwang †

Department of Information Management †
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@mail.cyut.edu.tw
Fax: 886-4-3742337

Department of Computer Science and ‡
Information Engineering,
National Chung Cheng University,
Chaiyi, Taiwan, R. O. C.

January 13, 2001

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC89-2213-E-324-001.

†Responsible for correspondence: Prof. Min-Shiang Hwang

A Remote Password Authentication Scheme for Multi-Server Architecture Using Neural Networks

Abstract

Conventional remote password authentication schemes allow a serviceable server to authenticate the legitimacy of a remote login user. However, these schemes are not used for multi-server architecture environments. In this paper, we present a remote password authentication scheme for multi-server environments. The password authentication system is a pattern classification system based on an artificial neural network. In this scheme, the users only remember user identity and password numbers to log in to various servers. Users can freely choose their password. Furthermore, the system is not required to maintain a verification table and can withstand the replay attack.

Keywords: Neural network, Password authentication, Remote login, Security.

1 INTRODUCTION

Recently, computer security has become an important issue. More and more systems have added control to the access process for avoiding illegitimate users reading sensitive information. Password authentication is one of the mechanisms that is widely used to authenticate a legitimate user. Conventional password authentication schemes are suited to solve the privacy and security problem for single servers under a client/server architecture. However, the use of computer networks and information technology has grown spectacularly. Many network architectures have become multi-server environments. In conventional password authentication schemes, a network user not only needs to log into various remote servers with repetitive registration, but also needs to remember the various user identities and passwords. Another problem in using the traditional password authentication method is that a server must maintain a password table that stores each user's ID and password. Therefore, the server requires extra memory space to store the password table. The table is shown in Figure 1(a) [10]. When a user logs into a computer, he/she types in the ID and password. The server searches the password table and checks if the password is legal. However, this method is dangerous. The password information table could be read or altered by an intruder. An intruder can also append a new ID and password into the table.

In order to avoid the security problem, some password authentication schemes have proposed as in [9, 10, 15, 16, 17, 19, 27, 29]. In their schemes, verification table is used to replace password table. The verification table is shown in Figure 1(b). These schemes often uses the techniques of one way hash function or encryption algorithm to encode the password, and stores the value of $F(pw)$ on the table to solve the security problem. However, the server must maintain and protect the table from being modified by an intruder.

In this paper, we propose an efficient remote password authentication scheme based on a neural network. This system identifies the legitimate user in real time using a pattern classification technique, and is applicable to a multi-server network architecture. In this classification system, the input pattern is the user password, and the output is the serviceable servers. Before describing the proposed scheme, we shall briefly review the related works on remote password authentication schemes. It is worthwhile to notice that the reviewed schemes have already solved the problems of remote password authentication in a single-server network architecture. However, these reviewed schemes are not applicable to a multi-server network architecture. Following this review, our proposed scheme is presented. The experimental results and the selection of an applicable structure for the neural network are discussed

ID_1	PW_1
ID_2	PW_2
ID_3	PW_3
. . .	. . .
ID_n	PW_n

(a) The password table

ID_1	$F(PW_1)$
ID_2	$F(PW_2)$
ID_3	$F(PW_3)$
. . .	. . .
ID_n	$F(PW_n)$

(b) The verification table

Figure 1: The password table and verification table architecture.

thereafter. Finally, our conclusions are presented in the last section of this paper.

2 Related Works

In this section, related works in the area of remote password authentication schemes are introduced. A typical remote password authentication system consists of two kinds of participants, a remote user and a serviceable server. The server authenticates the password of the login user and provides service for a legitimate login user. Typically, a remote password authentication protocol consists of three phases: (1) registration phase, (2) login phase, and (3) server authentication phase. A new user must register with the server first. After registration and authorization, the user can login to the server and obtain service.

The first password authentication scheme concept based on the one-way hash function was proposed by [9, 15, 17, 19, 29]. This system used a verification table to verify the validity of a login user. An intruder could modify the verification table stored in this computer system. In 1982, Denning [6] proposed a password scheme based on a signature scheme. This scheme, however, also required a verification table. These schemes cannot withstand the replay attack.

In 1990, the password authentication scheme based on a neural network was initially proposed. The related works [3, 22, 23] proposed an effective pattern recognition techniques for identifying users. The user types his password and the time interval between each character stroke is collected. The intercharacter time is treated as an input vector. Using the neural network to classify a particular user, latter, Anagun and Cin [1] proposed a scheme for a multiple user environment. In their scheme, the input vector is collected in a manner similar to the above method. The system presents each user with a password of different lengths.

Recently, new remote password authentication schemes [4, 11, 12, 13, 14, 28, 30] that can withstand the replay attack and permit users to choose and change their passwords freely have been proposed. However, these schemes are not adaptable to multi-server environments. It is inefficient to login various remote servers with repetitive registration and retain numerous user IDs and passwords. To amend these problems, we propose a new authentication scheme that uses a neural network and no verification table for password authentication in a multiple server system.

3 A New Authentication Scheme Using Neural Networks

In this section, we propose a remote password authentication scheme that utilizes a neural network. This scheme is designed for a multi-server environment. In the password authentication process, there are three participants: the login users, the various servers, and a system administrator (SA). In our scheme, each legitimate user holds only a user identity and its corresponding password.

The system administrator recognizes a user through the neural network. The neural network is trained and the weights are stored in each server. The process can be divided into three phases: (1) the registration phase, (2) the login phase, and (3) the authentication phase. Before logging in to a server, a new user must register some information first to become a legitimate user. The registration phase only performed once. Each legitimate obtains a valid user identity and password. Later, the user types his identity and password to login to any one of the servers. In the authentication phase, the servers validate the legitimacy of the remote login user.

In the remainder of this paper, U_1, U_2, $\cdots$, U_n stands for n users and Ser_1, Ser_2, $\cdots$, Ser_m stand for the m various servers. ID_i and PW_i stand for the user identity and password of U_i. Initially, SA chooses a large public prime p and a 56-bit secret key k. $E_k()/D_k()$ define the DES-like encryption/decryption function using the secret key k, where DES-like is a symmetrical data encryption standard. It encrypts plaintext in 64 bits and outputs the ciphertext in 64 bits. For more details about the DES-like algorithm, refer to [18, 25]. In the following, the proposed remote password authentication scheme for a multi-server architecture is described.

The registration phase:

In the registration phase, the user must first register with the server. Assume that the new user U_i is granted registration only from some certain servers. The steps of the registration phase are as follows.

1. We allow the new user U_i to choose a login password PW_i freely. The user delivers the PW_i to SA in a secure manner. SA then computes the user identity ID_i that satisfies

$$ID_i = E_k(PW_i). \tag{1}$$

   Without the key k, no one can compute the ID_i.

2. In this step, SA adds the training pattern of the new user to reconstruct the network. The network architecture consists of three layers: the input layer, the hidden layer, and the output layer. The input units are the password characters and the value v. The password characters can be an English word or a numeral. In addition, the value v is related to the expected output. The output represents the serviceable servers. If the system has m servers, the number of output units is m. The training pattern includes the user's password, v, and the expected output value. If the user receives the privilege of service from Ser_i, the ith unit of the expected output value denotes 1. For example, a system has six servers and a new user can login to Ser_1 and Ser_5. The input is shown in Figure 2. The "$KgiKaXXD$" is the new user's password. The value v is 17, which is the binary number for the expected output 010001.

   Then, SA collects the entire registered user's training pattern as the training set for the neural network. Before training, according to the mapping table, as shown in Table 1, each input character is mapped into a value that ranges from 0 to 62. Then SA normalizes the input value that ranges from 0 to 1. Once the training process is completed by the SA. SA sends ID_i and v to user U_i, and stores the networks weights and the secret key k in each server. The network model and the details of the training steps are described in the next section. The registration phase is complete up to this point.

The login phase:

In this phase, assume that a legitimate user wants to login to server Ser_j. The login phase is performed using the following steps.

1. The user obtains a time sequence T, which is like a time-stamp.

The training pattern															
Input										The expected output					
password								*V*		*Ser_6*	*Ser_5*	*Ser_4*	*Ser_3*	*Ser_2*	*Ser_1*
K	g	i	K	a	X	X	D	1	7	0	1	0	0	0	1

Figure 2: The training pattern.

character	null	a	b	c	d	e	f	g	h	I
mapping	0	1	2	3	4	5	6	7	8	9
character	j	k	l	m	n	o	p	q	r	s
mapping	10	11	12	13	14	15	16	17	18	19
character	t	u	v	w	x	y	z	A	B	C
mapping	20	21	22	23	24	25	26	27	28	29
character	D	E	F	G	H	I	J	K	L	M
mapping	30	31	32	33	34	35	36	37	38	39
character	N	O	P	Q	R	S6	T	U	V	W
mapping	40	41	42	43	44	45	46	47	48	49
character	X	Y	Z	0	1	2	3	4	5	6
mapping	50	51	52	53	54	55	56	57	58	59
character	7	8	9							
mapping	60	61	62							

Table 1: The mapping table.

2. Afterward, the user computes W_i from the equation

$$W_i \quad = \quad PW_i^T \mod p. \tag{2}$$

Then, the user delivers the ID_i, W_i, v, and T to the login server Ser_j.

The authentication phase:

In the authentication phase, the server receives W_i, ID_i, v, and T at the time T'. The server performs the following tasks to authenticate the user's login request.

1. The server checks the correctness of the timestamp first. If the time interval between T and T' is greater than $\triangle T$, the server rejects the login request. Let $\triangle T$ denote the expected legal time interval for transmission delay between the login terminal and the system servers.

2. If the timestamp T is within the valid period, the Ser_j decrypts the ciphertext (ID_i) from the equation

$$PW_i = D_k(ID_i). \tag{3}$$

Then Ser_j obtains the user password PW_i and the server verifies if the following equation holds,

$$W_i = PW_j^T \mod p. \tag{4}$$

3. If the previous verification holds, the server authenticates that ID_i and PW_i are valid. Then the server authenticates the service privilege granted by the server. To provide the proper service the server normalizes the password and sends these values as input to derive the output values from the neural network. The outputs obtained represent the privileges that user can receive from the allowed servers. To transfer the output value into a binary number, we can check whether v holds. If the jth output unit is the desired output that approaches 1, the server accepts the user for login. The procedures for privilege authentication utilizing the neural network are shown in Figure 3. The detail on how the network works are described in the next section.

In our scheme, the system does not require password nor verification tables. Each server only stores the same and fixed set of network weights and a secret key k. Each login user only has one pair, PW and ID, to log in various servers in this system. Therefore, it is suitable for multi-server environment. Furthermore, our scheme validates the legitimacy of a remote login user in real time and can withstand the replay attack.

4 Experimentation

In the investigation phase, we used the same training pattern to test the learning abilities and the performance of three neural network models. The three

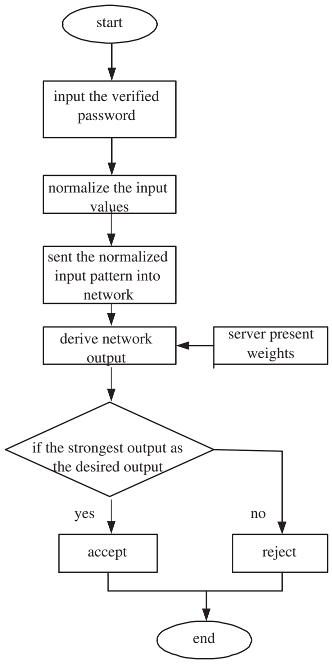

Figure 3: The procedures for privilege authentication.

models are as follows:

- Back-Propagation Network (BPN),
- Sum-of-Product Network,
- Hybrid Sum-of-Product Network.

All of these three models are supervised learning models. Each model consists of three layers: the input layer, the hidden layer, and the output layer. In order to compare the differences among these three models, the working environment should be defined in advance. These three models were run on AMD K6-based PCs. The programs used on these networks were implemented in C language. We assumed the multi-server system had 6 servers $(Ser_1, Ser_2, \cdots, Ser_6)$, and 100 users $(U_1, U_2, \cdots, U_{100})$. Each user password consisted of 8 characters. The server normalized value v into two login values. Therefore, the neural network architecture had 10 input units in the input layer, 24 processing units in the hidden layer and 6 output units in the output layer.

Assume that the password of U_1 is "$KgiKaXXD$" and he/she is granted registration only from servers Ser_1, and Ser_5. Therefore, the expected output values are $(0, 1, 0, 0, 0, 1)$. Before training the network, the password is normalized.

Normalization:

The password "$KgiKaXXD$" is divided into characters, then according to the mapping table, as shown in Table 1, the characters are mapped into [37, 7, 9, 37, 1, 24, 24, 30]. The values are then normalized into [0.592, 0.112, 0.144, 0.592, 0.016, 0.8, 0.8, 0.48]. In addition, the normalization value v becomes [0.1, 0.7]. Therefore, the training pattern for U_1 is [0.592, 0.112, 0.144, 0.592, 0.016, 0.8, 0.8, 0.48, 0.1, 0.7, 0, 1, 0, 0, 0, 1]. Table 2 illustrates each user's password and the corresponding normalization results from this experiment.

The Architecture:

In this experiment, the same training set, as shown in Table 2, was used to compare the accuracy and performance of these three architectures.

The Back-Propagation Network is one of the most well known types of neural networks. The BPN architecture is shown in Figure 4. The processing units between the layers are fully connected and the input value from each unit is the sum of the previous layer's output values multiplied by a weight vector. The value of these weights is modified by the training patterns. The BPN algorithm can be found in [24, 26]. There are 384 weights used by the BPN network architecture.

The Sum-of-Products network architecture [21, 22, 23] is shown in Figure 5, which is an additive scheme of the BPN network [24]. Each unit is the sum of the output values from the previous layer multiplied by a weight value.

User	Password	Training Set															
		Normalized Password								V		Expected output					
U_1	KgiKaXXD	0.592	0.112	0.144	0.592	0.016	0.8	0.8	0.48	0.1	0.7	0	1	0	0	0	1
U_2	coYLbq6s	0.048	0.24	0.816	0.608	0.032	0.272	0.944	0.304	0.1	0	0	0	1	0	1	0
U_3	t9zh8JCR	0.32	0.992	0.416	0.128	0.976	0.576	0.464	0.704	0.1	0.3	0	0	1	1	0	1
U_4	JxMR7pHo	0.576	0.384	0.624	0.704	0.96	0.256	0.544	0.24	0.1	0.8	0	1	0	0	1	0
U_5	lV9IneiU	0.192	0.768	0.992	0.56	0.224	0.08	0.144	0.752	0.3	0.4	1	0	0	0	1	0
U_6	LmGFjDMd	0.608	0.208	0.528	0.512	0.16	0.48	0.624	0.064	0.5	0.6	1	1	1	0	0	0
U_7	Qsji63jH	0.688	0.304	0.16	0.144	0.944	0.896	0.16	0.544	0.2	0.8	0	1	1	1	0	0
U_8	D8eXRi31	0.48	0.976	0.08	0.8	0.704	0.144	0.896	0.864	0.1	0.1	0	0	1	0	1	1
U_9	1ezhQamk	0.864	0.08	0.416	0.128	0.688	0.016	0.208	0.176	0	0.7	0	0	0	1	1	1
U_10	MQcF7Y6O	0.624	0.688	0.048	0.512	0.96	0.816	0.944	0.656	0.5	0.9	1	1	1	0	1	1
U_11	eXFZJPjy	0.08	0.8	0.512	0.832	0.576	0.672	0.16	0.4	0.2	0.6	0	1	1	0	1	0
U_12	LH5hHeRT	0.608	0.544	0.928	0.128	0.544	0.08	0.704	0.736	0.3	0.2	1	0	0	0	0	0
U_13	qnMAyAkE	0.272	0.224	0.624	0.432	0.4	0.432	0.176	0.496	0.5	0.9	1	1	1	0	1	1
U_14	fSgWPIT3	0.096	0.72	0.112	0.784	0.672	0.56	0.736	0.896	0.6	0.2	1	1	1	1	1	0
U_15	6cYAhEdk	0.944	0.048	0.816	0.432	0.128	0.496	0.064	0.176	0	0.8	0	0	1	0	0	0
									. . . . . .								
U_88	NmzviSj7	0.64	0.208	0.416	0.352	0.144	0.72	0.16	0.96	0.3	0.4	1	0	0	0	1	0
U_89	clvk0fbb	0.048	0.192	0.352	0.176	0.848	0.096	0.032	0.032	0.2	0.6	0	1	1	0	1	0
U_90	rUiD3YZp	0.288	0.752	0.144	0.48	0.896	0.816	0.832	0.256	0.4	0	1	0	1	0	0	0
U_91	1QrSAsdr	0.864	0.688	0.288	0.72	0.432	0.304	0.064	0.288	0	0.4	0	0	0	1	0	0
U_92	b9nuiXpG	0.032	0.992	0.224	0.336	0.144	0.8	0.256	0.528	0	0.2	0	0	0	0	1	0
U_93	NsAC6HuM	0.64	0.304	0.432	0.464	0.944	0.544	0.336	0.624	0.5	0.4	1	1	0	1	1	0
U_94	Xvo2qiJU	0.8	0.352	0.24	0.88	0.272	0.144	0.576	0.752	0	0.3	0	0	0	0	1	1
U_95	Zc7fT8Ri	0.832	0.048	0.96	0.096	0.736	0.976	0.704	0.144	0.1	0.9	0	1	0	0	1	1
U_96	o5F4tZGk	0.24	0.928	0.512	0.912	0.32	0.832	0.528	0.176	0	0.8	0	0	1	0	0	0
U_97	O2Zad1jH	0.656	0.88	0.832	0.016	0.064	0.864	0.16	0.544	0.2	0.6	0	1	1	0	1	0
U_98	b5OAW9rp	0.032	0.928	0.656	0.432	0.784	0.992	0.288	0.256	0.5	0.4	1	1	0	1	1	0
U_99	FQxX5dPN	0.512	0.688	0.384	0.8	0.928	0.064	0.672	0.64	0.2	0.1	0	1	0	1	0	1
U_100	JWgztJkj	0.576	0.784	0.112	0.416	0.32	0.576	0.176	0.16	0.5	0.8	1	1	1	0	1	0

Table 2: The training set illustration.

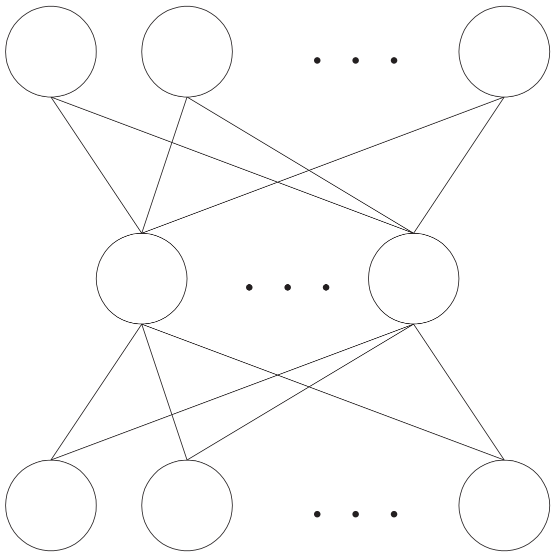

Figure 4: The back-propagation-network architecture.

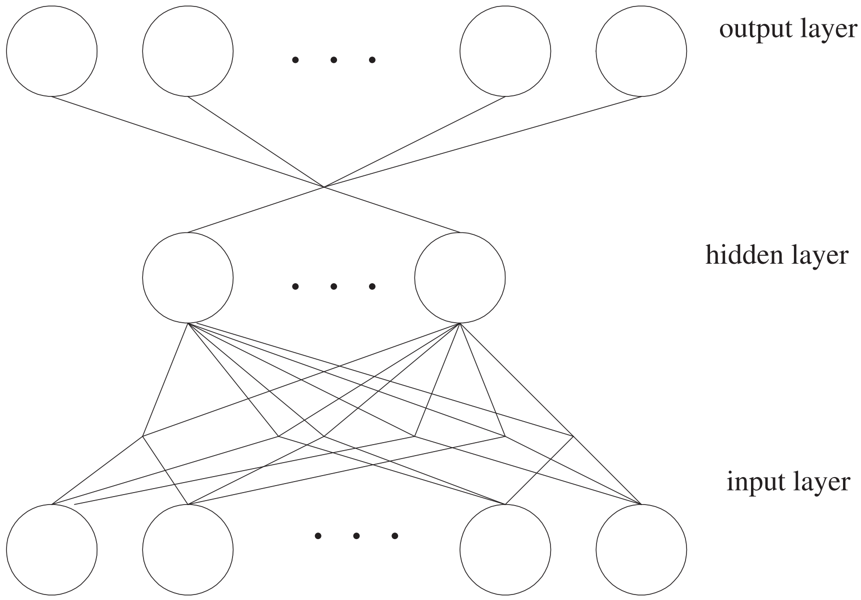

Figure 5: The sum-of-product network architecture.

For example, the 10 input units have 45 $(= 9 + 8 + \cdots + 1)$ two unit combinations. We can compute the number of combinations using this equation: $n(n-1)/2$, where n is the number of units. In this experiment, the Sum-of-Product network consisted of 10 input units, 24 hidden units, and 6 output units. Therefore, this network implies 2736 weights.

The number of interconnecting weights grows faster than the BPN network. In our password authentication scheme, the number of input units is fixed, but the number of output units increases when a server is added into the system. Therefore, if the number of servers increases, the number of interconnect weights grows rapidly. A large number of weights require more memory space and computation time. In order to improve this weakness, we propose a new Hybrid Sum-of-Products network architecture, which is similar to that described in Obaidat et al. [22]. The Hybrid Sum-of-Product network architecture is shown in Figure 6. The hidden layer connects to the output layer using the standard BPN architecture and the input layer is connected to the hidden layer using the Sum-of Product network architecture. With this

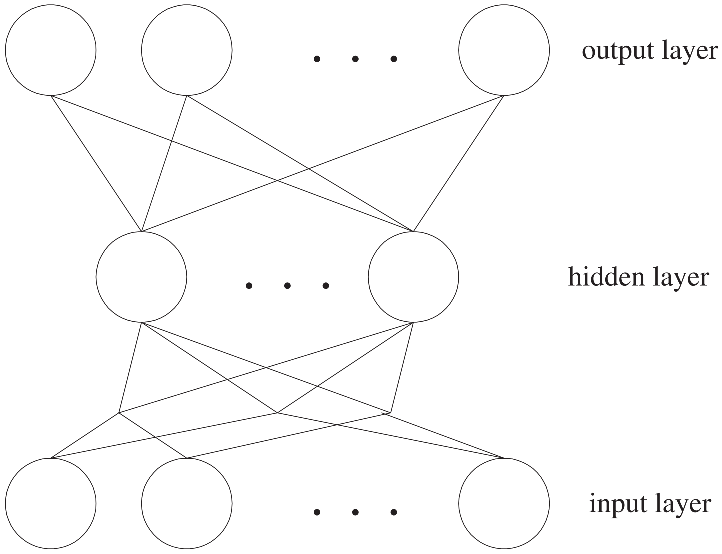

Figure 6: The hybrid sum-of-product network architecture.

network model, the network needs 1224 weights in the experiment.

Three types of network architecture for testing the password authentication scheme are presented. The cost and performance of these architectures will be compared.

Training:

Table 2 shows the training pattern for the neural network. In this table, each training pattern has 16 values. The numbers 1 - 8 are used to normalize the password. The numbers 9 - 10 are the normalization values for v, and the last numbers 11 - 16 are expected values. In this experiment, we assumed that the system had 100 users and 6 servers. Each training pattern has 16 values. The training set has a total of 100 patterns. The training set is then input to train the network.

In the training process, the weights are randomly assigned and the learning rate is 0.5. The transfer function is the Sigmoid function. The value is between 0 to 1. When the Sum of Squared Error (SSE) reaches its minimum or the error has not changed, the training is stopped. Once the training is completed, the weights are stored in the neural network in each server. In addition, the system threshold is designed to have a greater limit.

Classification:

After the training phase, the servers store the network weights and define the threshold, which is the minimum error for the all classification outputs for all legitimate users. When a user delivers his ID, W, and v to login to a particular server in step 2 of the authentication phase, a wrong ID and PW was rejected first. If the pair, ID and PW, are valid, the server will input the password normalization and v to compute the classification output using the neural network. Each user's classification output is shown in Table 3. The output result is must strongly represent a particular server. If the output error is less than the threshold, the server accepts the login request. For example, we define the system error threshold at 0.1. When a user U_1 wants login to server Ser_1, the server computes the classification result through the classification network. The classification output is $[0.00, 1.00, 0.00, 0.00, 0.00, 0.99]$, all units are less than the error threshold. The server then computes the binary number 010001 and checks if the value v is equal to 17. If this holds, the server accepts the login request. Otherwise, the server rejects the login request.

Accuracy and Performance Analysis:

In this experiment, we used the same training set to test the three types of neural network architectures. In all of these architectures, the number of input and output units are the same. The number of units in the hidden layer may be different. In these neural networks, it was difficult for the Sum of Squared Error (SSE) algorithm to reach it's minimum. The classification output approached the expected output as close as possible. A comparison of these three architectures is shown in Table 4. A smaller SSE can be obtained

User	*The classification output*					
U_1	0.000000	1.000000	0.000000	0.000000	0.000001	0.999843
U_2	0.000000	0.000000	1.000000	0.000000	1.000000	0.000000
U_3	0.000000	0.024476	0.999981	0.999998	0.000000	0.999898
U_4	0.000000	0.999867	0.000000	0.000000	1.000000	1.000000
U_5	1.000000	0.000000	0.000000	0.000000	0.999992	0.000000
U_6	0.999983	1.000000	0.999982	0.000000	0.000262	0.000000
U_7	0.000000	1.000000	1.000000	0.961495	0.000002	0.000013
U_8	0.000000	0.000000	1.000000	0.000000	0.999649	1.000000
U_9	0.000000	0.000000	0.000000	1.000000	0.931571	1.000000
U_10	1.000000	1.000000	1.000000	0.000000	1.000000	0.997464
U_11	0.000000	0.999948	0.997851	1.000000	0.000000	0.015538
U_12	1.000000	0.000168	0.000000	1.000000	0.000005	0.000000
U_13	1.000000	1.000000	0.961591	0.000000	0.999669	1.000000
U_14	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000
U_15	0.000000	0.105541	1.000000	0.000000	0.000000	0.000000
			⋮			
U_89	0.000000	0.992993	1.000000	0.000000	0.999999	0.000000
U_90	0.993012	0.000000	1.000000	0.000000	0.000010	0.000424
U_91	0.000000	0.000000	0.000000	0.999901	0.000000	0.000383
U_92	0.000000	0.000005	0.000000	0.000000	1.000000	0.000000
U_93	1.000000	0.987097	0.000000	1.000000	0.999580	0.000000
U_94	0.000000	0.004576	0.000000	0.000000	1.000000	1.000000
U_95	0.000000	1.000000	0.000000	0.000000	1.000000	1.000000
U_96	0.000000	0.000003	0.999853	0.000000	0.000008	0.000853
U_97	0.000000	1.000000	0.996084	0.000000	1.000000	0.000083
U_98	1.000000	0.999018	0.048888	1.000000	1.000000	0.000000
U_99	0.000000	0.928396	0.000000	1.000000	0.155087	1.000000
U_100	0.996604	1.000000	1.000000	0.000090	1.000000	0.000000

Table 3: The classification output for all users.

using the Sun-of-Product network. However, there are greater computation and storage costs. The Hybrid Sum-of-Product network is more suitable for a 100-user multi-server system.

In authentication scheme accuracy, the wrong ID and PW can be checked in step 2 of the authentication phase. It is impossible to login a wrong ID and PW into a certain server. The cases in which the ID and PW are legitimate, but the login user does not have the privilege to log in certain server, the user

	BPN		**Hybrid S-O-P**		**S-O-P**	
SSE	8.19	7.84	4.478	4.89	5.19	4.33
training times	1324 sec	1652 sec	599 sec	1839 sec	1946 sec	679 sec
# of units in hidden layer	24	48	24	32	20	24
# of the weights	384	768	1224	1632	2040	2736

Table 4: Comparison of the three network architectures.

will present a value v' to the login server. The server computes the classification output and checks if this value equals v'. The server will not grant privileges to that user for this server, although server accepted the login request. We tested 200 pairs (PW, v') for login. Out of 200 attempts, no classification output equaled the value v'. These users were not accepted for login to the server. It is very difficult to input v' and the output binary number equals the value v'. Therefore, in our scheme the error rate was nearly 0%.

5 Security Analysis

In this section, we discuss the security analysis of the proposed scheme in password authentication using the neural network. There are several possible attacks in the conventional remote password authentication scheme. In the following, we discuss some key properties of the proposed scheme and examine the security of our scheme.

Secrecy:

In our scheme, the secret key k of the system and the password PW_i of the user U_i must be kept secret. The parameters ID_i, v, and a large prime p are published. In our scheme, the ID is a ciphertext that using the DES-like algorithm to encrypt the password. It is not feasible to obtain the password without the system key k. Furthermore, a login user delivers ID , W, and v to the login server in login phase. The message W is $(PW^T \bmod p)$ and the information of PW can not be revealed by W. It gets its security from the difficulty of calculating descrete logarithms [7, 8]. Therefore, we can confirm the password secrecy.

Non-forgability:

Assume that an intruder wants to forge a legal user password to login to a server. The intruder forges a pair (ID, PW), and a timestamp T. However, if equation (4) does not hold, the server will reject the login request. A legal user wants to access a non-serviceable server. The user delivers the correct ID and PW, but delivers an incorrect value v. The server can resist the attack using the neural network. The server can check if the v is equal to the binary number of the classification output.

Replay Resistance:

To resist the replay attack, our scheme uses the timestamp concept. When an intruder replays a previously intercepted login message to masquerade as a legal user, he/she must pass the test in step 1 of the authentication phase. The intruder must change T into a new time T^* such that $(T'' - T^*) \leq \Delta T$. T'' is the timestamp given at the time the server receives the illegal login message. Once T is changed, the value of W also changes. However, the intruder can not solve the valid PW. This is the property of non-forgability. The intruder will fail the test in the authentication phase. Therefore, the proposed scheme is secure against the replay attack.

The proposed scheme detects replays of the login message by testing $(T' - T) \geq \Delta T$. However, this test may occur under two conditions. The first is if ΔT is too small, the server denies legitimate authentication requests. This will result in a possible network delay or the loss of clock synchronization. The other condition is if ΔT is too large, the server can not resist the replay attack. Thus, it is important here to choose an appropriate ΔT. One of the solutions is to use a replay detection buffer, which stores the clock synchronization. The server can change the ΔT according to the traffic load and clock synchronization. In the proposed scheme, the timestamping concept is susceptible to denial-of-service [2] at the authentication phase.

6 Conclusion

In this paper, we proposed a remote password authentication scheme based on a neural network. In this scheme, the server does not store or maintain password or verification table. The server only stores the weights of the classification network. According this network, the server can authenticate the validity of the login user in real time. The main advantage of this scheme is that it is applicable to both multi-user and multi-server networks. The system users can freely choose their password and the servers are required to retain only the pair user ID and password. The user can login to various servers without repetitive registration with each server. The password authentication scheme can prevent the replay attack, the intruder cannot obtain a login password through the open network and replay the password to login to a server.

Acknowledgements

The authors wish to thank many anonymous referees for their suggestions to improve this paper. Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC89-2213-E-324-001.

References

[1] A. S. Anagun and I. Cin, "A neural network based computer access security system for multiple users," *23rd International Conference on Computers and Industrial Engineering*, vol. 35, pp. 351–354, 1998.

[2] R. K. Bauer, T. A. Berson, and R. J. Feiertag, "A key distribution protocol using event markers," *ACM transactions on Computer Systems*, vol. 1, pp. 249–255, 1983.

[3] S. Bleha and M. S. Obaidat, "Dimensionality reduction and feature extraction applications in identifying computer users," *IEEE Transaction on System, Man, and Cybernetics*, vol. 21, pp. 452–456, Mar./Apr. 1991.

[4] C. C. Chang, S. M. Tsu, and C. Y. Chen, "Remote scheme for password authentication based on theory of quadratic residues," *Computer Communications*, vol. 18, pp. 936–942, Dec. 1995.

[5] C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," *IEE Proceedings-E*, vol. 138, no. 3, pp. 165–168, 1991.

[6] D. E. R. Denning *Cryptogryaphy and data security*, 1982. Addision-Wesley, Massachusetts.

[7] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, pp. 644–654, 1976.

[8] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[9] A. J. Evans, W. Kantrowiz, and E. Weiss, "A user authentication scheme not requiring secrecy in the computer," *Communications of the ACM*, vol. 17, pp. 437–442, 1974.

[10] G. Horng, "Password authentication without using password table," *Information Processing Letters*, vol. 55, pp. 247–250, 1995.

[11] M. S. Hwang, "A remote password authentication scheme based on the digital signature method," *International Journal of Computer Mathematics*, vol. 70, pp. 657–666, 1998.

[12] M. S. Hwang, "Cryptanalysis of remote login authentication scheme," *Computer Communications*, vol. 22, no. 8, pp. 742-744, 1999.

[13] M. S. Hwang, L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, 2000.

[14] M. S. Hwang, C. C. Lee, Y. L. Tang, "An improvement of SPLICE/AS in WIDE against guessing attack," *accepted and to appear in International Journal of Informatica.*

[15] T. J. Hwang, "Password authentication using public-key encryption," *IEEE Proceddings International Carnahan Conference Security Technology*, pp. 141–144, 1983.

[16] H. T. Liaw, "Password authentication using triangles and straight lines," *Computers Math. Applic.*, vol. 30, no. 9, pp. 63–71, 1995.

[17] R. Morris and K. Thompson, "Password security: a case history," *Communications of the ACM*, vol. 22, pp. 594–597, 1979.

[18] National Institute of Standard and NIST FIPS PUB 46-2 Technology, "Data encryption standard," *U.S. Department of Commerce*, Dec. 1993.

[19] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, pp. 993–999, 1978.

[20] P. G. Neumann, "Risks of passwords," *Communications of the ACM*, vol. 37, p. 126, 1994.

[21] M. S. Obaidat, D. T. Macchiarolo, and S. Bleha, "An intelligent neural network system for identifying computer users," *Intelligent Engineering Systems Through Artificial Neural Networks*, pp. 953–959, 1991.

[22] M. S. Obaidat and D. T. Macchiarolo, “An on-line neural network system for computer access security,” *IEEE Transaction on Industrial Electronics*, vol. 40, no. 2, pp. 235–242, 1993.

[23] M. S. Obaidat and D. T. Macchiarolo, “A multilayer neural network system for computer access security,” *IEEE Transaction on System, Man, and Cybernetics*, vol. 24, pp. 806–813, May 1994.

[24] M. Roth, “Survey of neural network technology for automatic target recognition,” *IEEE Transaction on Neural Networks*, vol. 1, pp. 28–43, Mar. 1990.

[25] B. Schneier, *Applied Cryptography.* Second edition, 1996.

[26] B. Soucek, “Neural and concurrent real-time system,” *New York: wiley*, 1989.

[27] M. Udi, “A simple scheme to make passwords based on one-way function much harder to crack,” *Computers & Security*, vol. 15, no. 2, pp. 171–176, 1996.

[28] S. J. Wang and J. F. Chang, “Smart card based secure password authentication scheme,” *Computers and Security*, vol. 15, no. 3, pp. 231–237, 1996.

[29] M. V. Wilkes *Time Sharing Computer Systems*, 1975. Macdonald.

[30] T. C. Wu and H. S. Sung, “Authentication passwords over an insecure channel,” *Computers & Security*, vol. 15, no. 5, pp. 431–439, 1996.

(a) (b) (c)

Figure 7: (a)LI-Hua Li (b)Iuon-Chang Lin (c)Min-Shiang Hwang

BIOGRAPHICAL SKETCHES

Li-Hua Li was born in February 16, 1965. She received her B.S. degree in Mathematics from Providence University, Taiwan, in 1985. After she got her B.S. degree, she joined the Flight Test Department of Aeronautical Industrial Development Center (AIDC), a research center under Chung-Shan Institute of Science & Technology (CSIST) in Taiwan, as a research assistant from 1986 to 1989. She received her M.S. and Doctoral degree of Computer Science from the University of Alabama in 1991 and 1995, respectively. Dr. Li came to the Department of Information Management (IM), Chaoyang University of Technology (CYUT), Taiwan, as an associate professor in August 1995. She was appointed as the head of the department from Aug. 1997 to July 1999 and now she is the director of the CYUT Library. Her major research area is in neural network applications, fuzzy applications, decision making, and expert systems. Recently, she joins the Electronic Commerce (EC) research group in CYUT to study related topics of EC security.

Iuon-Chang Lin received the B.S. in Computer and Information Sciences from Tung Hai University, Taichung, Taiwan, Republic of China, in 1998; the M.S. in Information Management from Chaoyang University of Technology,Taiwan, in 2000. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He is currently professor and chairman of the Department of Information Management, Chaoyang University of Technology, Taiwan, ROC. He is a member of IEEE, ACM, and Chinese Information Security Association. Dr. Hwang's current research interests include database and data security, cryptography, image compression, and mobile communications.