A User Authentication System Using Back-Propagation Network *

Iuon-Chang Lin† Hsia-Hung Ou‡ 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.

August 6, 2002

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-004.

†Responsible for correspondence: Prof. Min-Shiang Hwang

A User Authentication System Using Back-Propagation Network

Abstract

Information security is a critical issue with all information systems. One of the key points in information security is user authentication. Password is widely used to authenticate a legitimate user. In conventional password authentication schemes, a server must maintain a password table that stores each user's ID and password. In order to ensure the system's security, the system has to protect the password table from being tampered with by an intruder. The method is very dangerous. In this paper, we use the technique of Back-Propagation Network to design a user authentication scheme. The proposed scheme does not store or maintain a password table and the users can freely choose their username and password. Furthermore, if any user wants to change his/her password, the system can retrain the BPN efficiently.

Keywords: Back- propagation network, information security, neural network, security, user authentication

1 Introduction

Information system security management has become a very important issue. With the dramatic increase of all types of information systems and the explosive use of the widespread Internet as a major avenue for business and educational information exchanges, protecting information and information systems from unlawful access, information theft and information system interruption or destruction has become more and more critical. The coming of *information criminals* has brought the following security threats for information systems [22].

- System invasion by illegal users
- Deliberate system compromise by legal users
- Information intercepted and illegally modified
- Other software or system corruption

These security threats can affect the secret data of business enterprises and personal privacy as well as cause damage or less to online sales or denial of information services. Therefore, how to avoid illegal users from invading the computer system is an important part of information security.

Many methods are used to identify the user's identity such as password, fingerprint, typing sequence, etc. Among them, password-based user authentication is a method that is widely used to authenticate a legitimate user. Generally speaking, password authentication system involves a password table that contains the usernames and passwords of authorized users. When requesting information services, the user must first input his/her username and password. The system looks through the password table for a matching name and password. If a match is found, the user is granted access to the service. Therefore, the password table plays an important role. However, the method is dangerous. The chief defect in this system is that usernames and passwords are stored in the system. The password table could be read or altered by an intruder. An intruder can also append a new ID and password into the table. Therefore, the server not only requires extra memory space to store the password table but also requires more secure mechanisms to protect the password table.

In order to deal with the secure problem. A lot of researches have been aimed at replacing the password table. Most of these methods use a public-key cryptographic system as the solution. However, the computation of public-key cryptosystem is time consumed. So far, many user authentication schemes have

been proposed. Generally, they can be categorized into password authentication, symmetric cryptography, public-key cryptography, and authentication certificate. The defects of the four categories are summarized as follows.

1. Password identification: In on-line user systems, the user directly inputs his/her ID and password. However, this method has two defects:
   - the password can be easily stolen in the transfer process;
   - the system must store a password table, which increases the load on security maintenance and management [6].
2. Symmetrical cryptography: The symmetrical password technique, such as the Kerberos technique [17], requires a third party for identification before the user can access the system. The third party is the weakness in this process.
3. Public-key cryptography: The public-key cryptography system in the RSA [19] does the identification. The user uses his/her private key to sign the status, while the server uses a public key to encrypt the status and identify the user. The defect is the extensive computation time.
4. Authentication certificate: The user has previously obtained a certificate for identification. The defect is that a third party must issue this certificate and the user must have a storage device (such as a Smart Card or IC Card) to store the certificate.

In this paper, we propose an efficient user authentication system using a Back-Propagation Network (BPN) [13]. The memory characteristic of the BPN is used to recall the password information in the network, replacing the traditional password table. Furthermore, the system allows users to freely choose their IDs and passwords. The rest of this paper is organized as follows. Before describing the proposed method, we shall briefly review the related

works on user authentication schemes in the next section. In Section 3, we shall describe the model and the proposed scheme. The experimental results and the security analysis of our scheme will be discussed in Section 4. Finally, our conclusions will be in the last section of this paper.

2 Related Works

In this section, available mechanisms and systems will be introduced. A typical password authentication system consists of two kinds of participants, a remote user and a serviceable server. Before providing service for a login user, the server has to authenticate the legitimacy of the login user. Maintaining a password table is a simplest and most straightforward method to identify a legal user. This table records each user's ID and the corresponding password. However, the user's password may be tampered with either by intruders or insiders. The method is very dangerous.

To solve this security problem, many schemes use a verification table to prevent the password from leaking out [5, 9, 15, 16, 23]. The verification table stores the IDs and the corresponding $F(PW)s$ of the authorized users, where $F(\cdot)$ can be a one-way hash function or an encryption algorithm. However, even through the user can keep the password to him-/herself, the verification table still might be tampered with by an intruder. Therefore, the system has to maintain and protect the verification table carefully.

Recently, some password authentication schemes without verification tables have been proposed [7, 8, 10, 11, 21]. However, these schemes require more computational and communication traffic overhead. In additional, some schemes [2, 3] do not allow the user to choose the identity and password freely.

Neural network is a powerful technique for applying in user authentication systems. In some related works [1, 18], the efficient pattern recognition techniques for identifying users are proposed. The user types his password and the

time interval between each character stroke is collected. The intercharacter time is treated as an input vector. Then, the server trains a neural network and uses the neural network to classify a particular user.

Recently, Li et al. [12] proposed a new user authentication scheme using neural networks. The main advantages of this scheme are that it is suitable for multi-server architecture and also without any password table or verification table. However, when the number of the usernames, passwords, and login servers increases or requires modification, the scheme must retrain. It is not efficient. In this paper, we propose a password authentication scheme based on back-propagation network. This system not only can identify the authorized user but also can retrain the neural network efficient. In addition, the users are allowed to choose their IDs and passwords freely.

3 The Proposed Scheme

3.1 The System Model

The plan here is to use neural network to generate and memorize the identification parameters. The Back-Propagation Network (BPN) is one of the most well known types of neural network. Many different models of BPN are proposed such as Sum-of-Product network, Hybrid Sum-of-Product network. As the result of [12], the typical BPN requires less number of weights. The BPN algorithm can be found in [20]. The architecture of BPN is shown in Figure 1. It is basically composed of the Input Layer, Hidden Layer, and Output Layer. 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 BPN must be trained with a set of training pattern. The training pattern must include both the known input and expected output respectively in the input layer and output layer. Training provides the network parameters

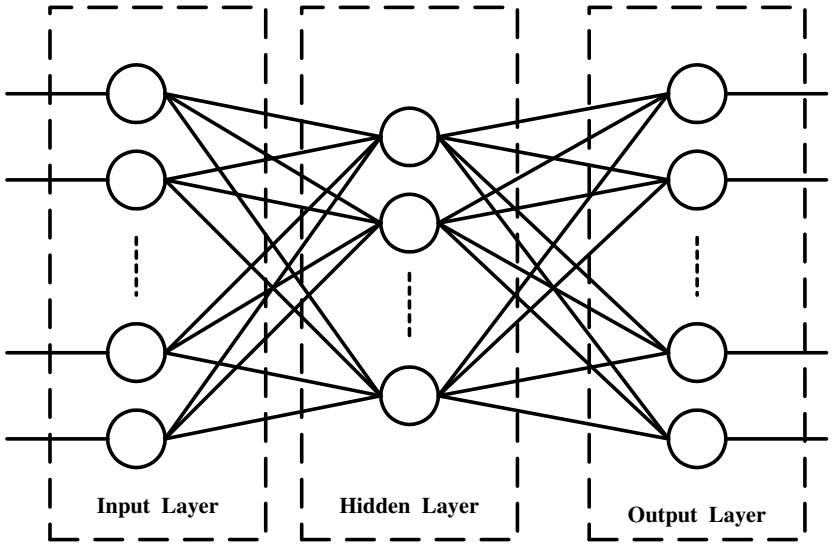

Figure 1: The basic architecture of BPN

and weight values. The value of these weights are modified by the training patterns. When the weight values have been calculated, the expected output can be produced as using known input values are entered. This is the basic theorem of the BPN, and our scheme follows the same route. Since the BPN is capable of recalling and identify user information, it can be used to identify the validity of a user.

3.2 Our Proposed Scheme

A user authentication scheme can be divided into three phases: the user registration phase, user login phase, and user authentication phase. A user must register at the computer system to become an authorized user. Each user only goes through this process once. When a registered user wants to log into the computer system, the user types his/her ID and password in the login phase. In the user authentication phase, the system validates the legitimacy of the login user. The details of the user authentication system is described as follows.

- **The User Registration Phase:**

The training pattern															
Input								The expected output							
username								***password***							
A	b	r	a	h	a	m	e	1	2	3	e	d	u	t	w

Figure 2: The training pattern

1. The user chooses a login username and password freely, which can be either English letters or numerals. This data is sent to the system administrator (SA) in a secure way.

2. The SA collects all registration information as the training set for training the BPN. The training pattern is shown in Figure 2. The input is the username and the expected output is the user's password. Since the range of the input and output value is 0 to 1, the system has to normalize the username and password before training the BPN. Therefore, we add an encoding mechanism to the system to normalize the training pattern. The encoding mechanism maps each character into ASCII . When the SA receives the username and password, it will divide the username and password into characters and transform each character into a 7-bit binary code. For example, suppose the username is Tom. The ASCII code for "Tom" is 84 111 109, and the binary code is 1010100 1101111 1101101. The reason why we decide use a 7-bit binary code is that we assume the application system could only accept 127 ASCII characters.

3. After encoding the username and the password, the train pattern is passed through the hash mechanism. The hash mechanism can take in an arbitrary-length input and turn it into a fixed-length output [4, 14]. Furthermore, the hash mechanism has the three features:1)

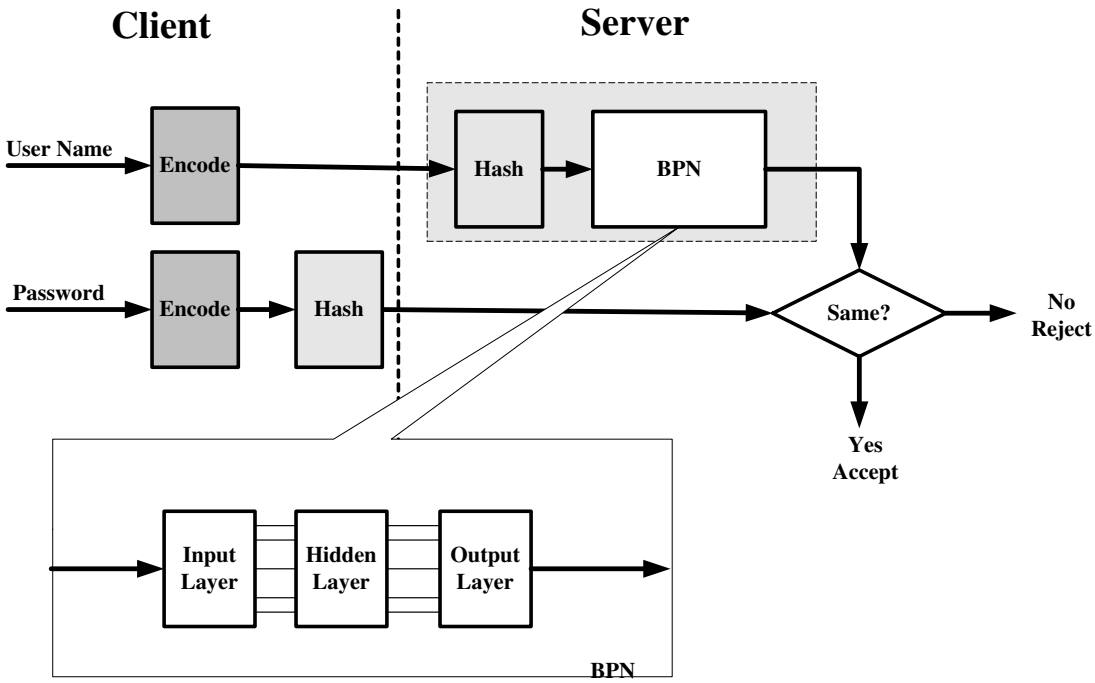

Figure 3: The processes of the login and user authentication phases

Given an input, it is easy to compute the output through the hash mechanism; 2) Obtained an output, it is difficult to derive the input; 3) Given an input, it is difficult to find another input. Finally, the hashed username and password are either 0 or 1.

4. The SA takes the hashed username as the input and the hashed password as the expected output to train the BPN. When the training process is completed by SA, the SA stores the network weights in the server.

- **The Login Phase:**

When a legitimate user wants to log into the computer system, the user must input both the username and the password. Then, the client site encodes the username and the corresponding password, producing a hash value in accordance with the encoded username automatically. Therefore, the login request includes the encoded username and the hashed password. Then, the login request is sent to the server. Figure 3 illus-

trates the processes of the login and user authentication phases in our proposed scheme.

- **The User Authentication Phase:**

  When the server receives the login request, it uses the BPN to authenticate the legality of the login user. The authentication process is described as follows.

  1. First, the server hashes the encoded username.
  2. The hashed username is the input. The server calculates the output via the BPN.
  3. The server compares the output and the hashed password received after. If they are identical, the user passes identification. If not, this user is rejected as an illegal user.

4 Experimentation

In our scheme, the train model is BPN that is a supervised learning model. This model consists of three layers: the input layer, the hidden layer, and the output layer. The training data: username for input and password for expected output. The training set of the experiment are as shown in Table 1. In this experiment, we assumed the user authentication system had 200 users. Each username and password consisted eight characters and transformed each character into 7-bit binary code. Therefore, the BPN architecture had 56 input units in the input layer, 120 processing units in the hidden layer, and 56 output units in the output layer. The system was run on AMD K6II-300 PCs with the RAM of 64 M.

Table 1: The Test Training Data

Username	Password	Username	Password	Username	Password

Abrahame	123edutw	Addison	4816747	Adam	aaron
alec	MTV	Anand	4866447	Andrzej	MTV
albert	3377	Antorun	12345678	barry	barboy
Bella	dell	Bishop	ANS	Boebert	99gpw
Brussea	qAzX	Cesare	Taiwan	Chaum	chair
Chawla	7653	Cifford	werwet	colin	callhome
Corradi	Sexy	Cremon	ccc123	Cybenko	7799123
Darnell	R2D2	Damianos	City	Daniela	password
david	935	Dennis	8814605	Dhaval	maggie
Dominic	Filter	Donny	windows	EA95611	6812
edith	earth	eric	3323000	Ebank	fcic
Egbert	ORTE	Emmanuel	1NGhgtre	eyeQuR	978df
Felix	TOYOTA	Franz	1199aaa	freddy	friend
Fritz	element	Fuenfr	3236754	Gabriel	ggg999
Gatot	137946	Gavalas	Gussic	Gennady	ansi11
George	flower	Ghanbar	9999	Gleeson	gogo
Gray	weqqzd	Green	Wood	Guan	gloss
Guido	9999	Hansoth	Banla	Hartmut	Golder
Hohl	Hotel1	Holding	any	Holger	Kevin
Hugo	DES327	Hylton	hyper	Itabashi	111111
Ingemar	Jissly	Isidore	Fdgh	Isabel	8rstm
JaeYi	9089rt	JanLee	December	jane	4856600
Jatin	BigApple	jean	3323000	Jepsen	111111
Jeremy	bigman	JinHong	November	Jimyuan	start
JiRen	324667	Jessica	JASNIC	Johan	sentrans
JoonLee	lam0	Jorge	coco	Joseph	ABC
judy	neural	kaiKin	ISO9002	Kare	2000
Karnker	g5g6d	Kristin	BBig	Katsuya	zxcvbnm
Kazuhiko	small	Keith	44388591	Ken	flybird
Korba	5658	Kelly	Tony	Kunkel	MANN
Kurfess	kills	Larry	OKI8W	Lauvset	last
lily	4032	Leopold	Lee	Louise	48rcd
Luis	BlueWay	maggie	friday	Mahony	87dfjkl
Manheim	nmfg235	Marco	Internet	Marques	66585
Marzul	lomoqw1	Masse	09fg5f	Matasz	eriter
Medvin	sky	Mickun	aszero	Mike	month
Mitsuru	Linux	Mizuno	xzoisf	Mladen	Multi
Mogath	ds978es	Mohamme	Red Hat	Montan	motoro
Moriyama74	mkvk0	Moura	4598fd	Myeong	Japan
Naldurg	kkkkk	Neeran	Mobile	Neumann	CNuke
Noemi	only one	Nicola	bubu324	Nydia	31w3A
Norma	b124	Okamoto	qmnioZ	Oliver	palapala
Orazio	where	Oshima	llooppqq	OREO	COOK
Paciorek	96578	Pagurek	Park	Paolo	christle
Patrick	duncan	Pedersen	878324	Peine	Table

Peller	paLA	Prasad	jackson	Pulia	999111
Prudence	joline	Question	12345678	Quintina	lktjs
Queena	9731	Rahul	Camilla	Renee	amaei
Rebecca	885tink	Richaard	4856600	Robat	ioio64
Roger	Small666	Rossum	555FFF	Roberta	username
Sunder	quesT	Saurab	Motolola	Seung	Sentra
Shah	shall	Silva	aAaAaA	Someya	m91t5
Stefan	9a9b	Stockton	STOCK	Suzanne	thankyou
Sumit	Seminas	Susilo	Studiv	Tadanori	fotoshop
Tiffany	8996489	Takashi	JavaApi	Taococ	colee
Tardo	xyz	Tatsuaki	VISUAL	Theoph	quality
Thomas	discopub	TinQian	Ford	Tomar	66fffg
TomLee	water989	Tomoya	Storage	Torben	60min
Theresa	cyutms	Valente	weliw	Valerie	Network
Vitek	volume	Vogler	gold963	Vouk	qwqwqw
VuAnh	systems	Walsh	8dj4s	Winifred	9DoS6
Warsaw	PPP	Watanabe	TaBolO	WCZexe	taco99
Weissman	786dv	William	database	WongMS	8Cegg
Xudong	aaaa530	Xaviera	csccuedu	YangGH	slsl
Yiling	Paper	Yvonne	Mbetter	Yutaka	Chanel24
Yuuichi	datamini	Zhaoyu	acho56	ChZero	000ert
Zhung	889412	ZingCG	qsechay	Zyang	popsecu
zzHwang	1829iods				

4.1 Accuracy and Performance Analysis

The accuracy of the proposed scheme is good. We use the same training patterns to test the trained network. In our test, if we input the right username, the output is the corresponding hashed password. In contrary, if we input a wrong username, the output from the BPN is never equal to the corresponding hashed password. The case in that the login user does not have the privilege to log in the server. Thus the user inputs the wrong username and password and the system can check its wrong. The error probability is zero.

The efficiency of the proposed scheme is also good. In the training process, the system uses 257 minutes to train the BPN. The training time is long, because the input is digital data (0 or 1) and there is no relativity within the training pattern, which means that convergence is not easy. Although the

training time is long, we only need to perform the training process once when in initiating the system. After completing the training process, the system can efficiently to authenticate the identity of the login user. In this process, unlike public key cryptography that requires exponential computing, our system only requires simple multiplication and addition to produce the result. The BPN time complexity is $O(1)$. Thus, when a user wants to log into the server to obtain service, the user authentication system can quickly response the result that accept or reject the user's request. The frequency of the user authentication phase is high. Therefore, the proposed scheme can reduce more computational overhead than public key based system. Furthermore, the proposed scheme can apply to many applications that require real time response or low computational capability machine, such as using in user authentication for mobile cell.

When a user's privileges are cancelled, the network must be retrained. Fortunately, the training for deleting a user is much simpler than the training of the initial BPN. We can set some particular characters in its password (for example: #$%K) and then directly use the original network parameter to train the network (Learning Time). The same thing can be done when any user wants to change his/her password or system increases new users. This way, when a user tries to log into the system with an out-of-date username or password, then the system will reject the user.

4.2 Security Analysis

Only the legal user knows both correct the username and the password, and only the correct username and password can help the user make it through the authentication. In the user authentication phase, the username and the password are processed by a hash function. So, if any malicious attacker intercepts the identification message, he/she can only get the current hashed username and password. Since the hash function is a one-way function, the

attacker cannot turn the hashed data back to the original password. One thing that makes our method most distinct from others is that even our neural parameter can be open to the public without affecting the security. The hash layer makes our model secure. If an attacker gets the neural parameter, he/she can try inputting any username and get the password from the output end. But this password is not the original password because it is hashed and cannot be reversed by the same hash function. Therefore, our proposed method can completely replace the password table.

4.3 Comparisons

As we know, most of the existed schemes [2, 3, 5, 7, 8, 9, 10, 11, 15, 16, 21, 23] for user authentication are using cryptography technologies. The security of these technologies is usually based on difficult to factor a large numbers or calculate discrete logarithms in a finite field. However, in these schemes, they usually require modular exponential operations. The operation is inefficient and time-consuming. In addition, some schemes [5, 9, 15, 16, 23] still required password table or verification table, and some schemes [2, 3] the user cannot choose the username and password freely.

In our prior work [12], we proposed a new password authentication scheme using neural network. The outstanding contributions of this scheme are that it is suitable for multi-server architecture and can deal with the all drawbacks in the existed schemes. However, when the system servers required modification, the initial neural network must be retrained. Therefore, this scheme is suitable for the applications that the number of servers is fixed or low frequency to change.

In this scheme, when system increases new users or any user changes his/her password, the processes is more efficient than prior scheme. Different from prior scheme, this scheme does not retrain the initial network. It directly uses the original network parameters to train the network. The processes only

require learning time. Although this scheme is only suitable for single server architecture, it also can meet the all requirements of user authentication.

5 Conclusions

The user authentication system in a traditional network must maintain a password table in the server. In contrast, our method employs the BPN to recall the username and password. This method can safely produce a user name and password without any lookup table stored in the server.

The advantages of our method are as follows.

- Sparing the password table. It can resolve the problems associated with information leaks from the password table.
- Computing quickly. Only simple multiplication and addition are needed to produce the result instead of requiring exponential computing as public key cryptography does. The BPN time complexity is $O(1)$.
- When new data have to be added to the password table, the search time for the increased records increases, and yet this search time is minimal in our proposed system.
- An increase in the user number enlarges the storage space while the number of weight values is stationary.
- No third party is necessary in the identification process.
- The error probability is zero.

References

[1] 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, March 1991.

[2] C. C. Chang, R. J. Hwang, and J. B. Daniel, "Using smart cards to authenticate passwords," in *IEEE International Carnahan Conference on Security Technology*, pp. 154–156, 1993.

[3] C. C. Chang and S. J. Hwang, "Using smart cards to authenticate remote passwords," *Computers and Mathematics with Applications*, vol. 26, no. 7, pp. 19–27, 1993.

[4] I. B. Damgard, "A design principle for hash functions," in *Advances in Cryptology, CRYPTO'89*, pp. 416–427, 1989.

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

[6] W. Ford, "Security techniques for network management," in *Advanced Communications and Applications for High Speed Networks*, pp. 133–149, 1992.

[7] Min-Shiang Hwang, "Cryptanalysis of remote login authentication scheme," *Computer Communications*, vol. 22, no. 8, pp. 742–744, 1999.

[8] Min-Shiang Hwang, "A remote password authentication scheme based on the digital signature method," *International Journal of Computer Mathematics*, vol. 70, pp. 657–666, 1999.

[9] Min-Shiang Hwang, Cheng-Chi Lee, and Yuan-Liang Tang, "An improvement of SPLICE/AS in WIDE against guessing attack," *International Journal of Informatica*, vol. 12, no. 2, pp. 297–302, 2001.

[10] Min-Shiang Hwang and 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.

[11] J. K. Jan and Y. Y. Chen, "'paramita wisdom' password authentication scheme without verification tables," *The Journal of Systems and Software*, vol. 42, pp. 45–57, 1998.

[12] Li-Hua Li, Iuon-Chang Lin, and Min-Shiang Hwang "A remote password authentication scheme for Multiserver Architecture Using Neurnal Networks," *IEEE Transactions on Neurnal Networks*, vol. 12, no. 6, pp. 1498-1504, November 2001.

[13] R. P. Lippman, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, pp. 4–22, Apr. 1987.

[14] R. C. Merkle, "A fast software one-way hash function," *Journal of Cryptography*, vol. 3, no. 1, pp. 43–58, 1990.

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

[16] 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, Dec. 1978.

[17] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, 1994.

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

[19] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

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

[21] K. Singh, "On improvements to password security," *Operating System Review*, vol. 19, pp. 53–60, Jan. 1985.

[22] B. C. Soh and T. S. Dillon, "Setting optimal intrusion-detection thresholds," *Computers & Security*, vol. 14, pp. 621–631, 1995.

[23] 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.