A User Authentication System Using Back-Propagation Network

Iuon-Chang Lin[†] Hsia-Hung Ou[‡] Min-Shiang Hwang[†]

Department of Management Information System[†]
National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.
Email: {iclin;mshwang}@nchu.edu.tw
Fax: 886-4-22857173

Department of Computer Science [‡]
National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.

Identifying Number: NC ms 2896

September 19, 2012

[†]Responsible for correspondence: Prof. Min-Shiang Hwang

A User Authentication System Using Back-Propagation Network

Abstract

Information security has been a critical issue in the field of information systems. One of the key points in the security of computer system is how to identify the authorization of users. Password-based user authentication is widely used to authenticate a legitimate user in current system. In conventional password-based use authentication schemes, a system has to maintain a password or verification table which stores the information of users' IDs and passwords. Somehow, even the one-way hash functions and encryption algorithms are applied to prevent the passwords from being disclosed; the password or verification table is still vulnerable. In order to solve the vulnerable problem, in this paper, we apply the technique of Back-Propagation Network instead of the functions of password and verification table. Our proposed scheme offer better capability to work out the security problems that occurred in the systems of using password table and verification table. Furthermore, our scheme also provides each user to freely choose his/her username and password.

Keywords: Back-propagation network, information security, neural network, one-way hash function, user authentication

1 Introduction

The security management of information system has become a very important issue in now days. With the rapid 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

ID_1	PW_1
ID_2	PW_2
ID_3	PW_3
⋮	⋮
ID_n	PW_n

Figure 1: The password table

information systems from unlawful access, information theft, and information system interruption or destruction has faced a more critical condition. The coming of *information criminals* has brought the following security threats for information systems [24].

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

Therefore, in order to prevent an illegal user from invading the computer system, a user would need to provide an identity to a system as a proof of being legitimate user before he/she logs into the system. So far, there are many methods proposed to identify the legitimacy of each login user such as password, fingerprint, typing sequence, and so on [1, 21]. Among them, password-based user authentication scheme is the most widely-used and inexpensive mechanism.

A straightforward implementation of password-based user authentication is that the system keeps each authorized user's username, ID, and the corresponding password, PW, in table. The table is shown in Figure 1. When a user wants to login the computer system, he/she keys in his/her identity, ID,

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

Figure 2: The verification table

and password, PW, in response to the system's request. Then, the system looks through the password table for a matching name and password. If a match is found, the user is granted to login to the computer system. However, the plain password table directly stored in the computer system may present a potential threat to the system that is the passwords may be read or altered by an intruder. Therefore, the system requires an additional burden on the system for managing the password table.

In order to deal with the secure problem, verification table is utilized to replace password table [5, 9, 18, 19, 26]. The approach of using verification table is to transform the passwords into some test patterns, $F(PW)s$, through one-way hash functions [4, 16, 17] or encryption algorithms [11], and then store these test patterns as a public verification table as shown in Figure 2 [14]. When a user submits his/her ID and Pw to login to the computer system, the system first applies the same one-way function, $F()$, to the submitted password, and then checks the result according to the corresponding entry in the verification table. The verification table could not be kept secretly, because an intruder cannot decipher the original passwords from what is stored in the table [25]. The security of this technique is based on the cryptographic one-way hash functions and encryption algorithms. It has been widely used in UNIX system [18].

Nevertheless, the technique still has some shortcomings. An intruder is still able to append a forged $(ID, F(PW))$ pair to the verification table or replace someone's $F(PW)$ with another one. For instance, in Figure 2, user 3 may replace user 2's $F(PW_2)$ with $F(PW_3)$, then user 3 can forge the user 2 to login to the system.

Instead of using the verification table, we propose an alternative approach to overcome the security problem in verification table by applying the Back-Propagation Network (BPN) [15]. The memory characteristic of the BPN is applied to recall the password information. Compared with previously proposed schemes, which engages with a password table or verification table, the proposed scheme offers more security and allows users to freely choose their IDs and PWs.

The rest of this paper is organized as follows. In Section 2, we shall describe the model and the proposed scheme. The results of experiment results and the security analysis of our scheme will be discussed in Section 3. Finally, our conclusions will be in the last section of this paper.

2 The Proposed Scheme

2.1 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 [14], the typical BPN requires less number of weights. The BPN algorithm can be found in [23]. The architecture of BPN is shown in Figure 3. 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

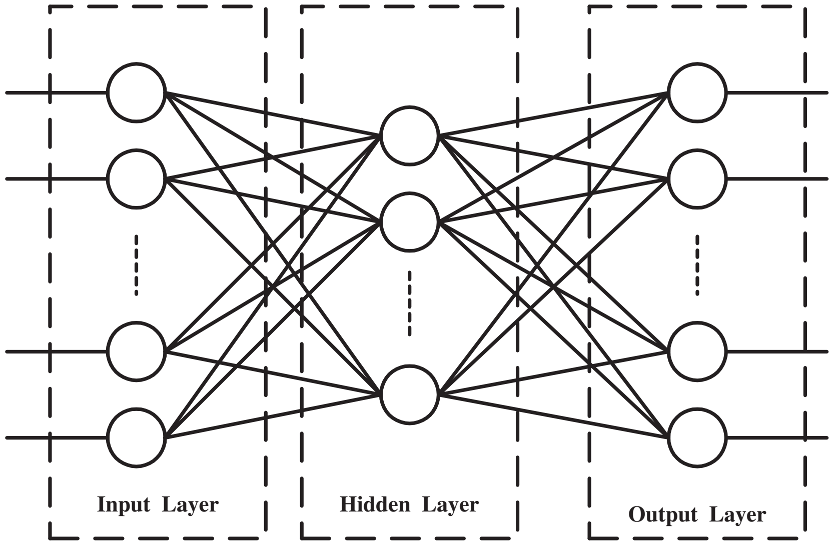

Figure 3: The basic architecture of BPN

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 and weight values. The values of these weights are modified by the training patterns. When the weight values are calculated, the expected output could be produced as entering the known input values. 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.

2.2 Our Proposed Scheme

A user authentication scheme can be divided into three phases: the user registration phase, user login phase, and user authentication phase. First, in order to be an authorized user, each person has to register in the system by giving his/her personal information, and this action would be taken only once. Afterward, while a user would like to use the system, he/she only needs to login with ID and password that he/she offered previously. In the user au-

The training pattern															
Input								The expected output							
hashed username								***hashed password***							
A	b	r	a	h	a	m	e	1	3	u	7	w	u	q	9

Figure 4: The training pattern

thentication phase, the system would validate the legitimacy of the login user. The details of the password-based user authentication system are described as follows.

- **The User Registration Phase:**

  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 4. The input is the username and the expected output is the user's hashed 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.

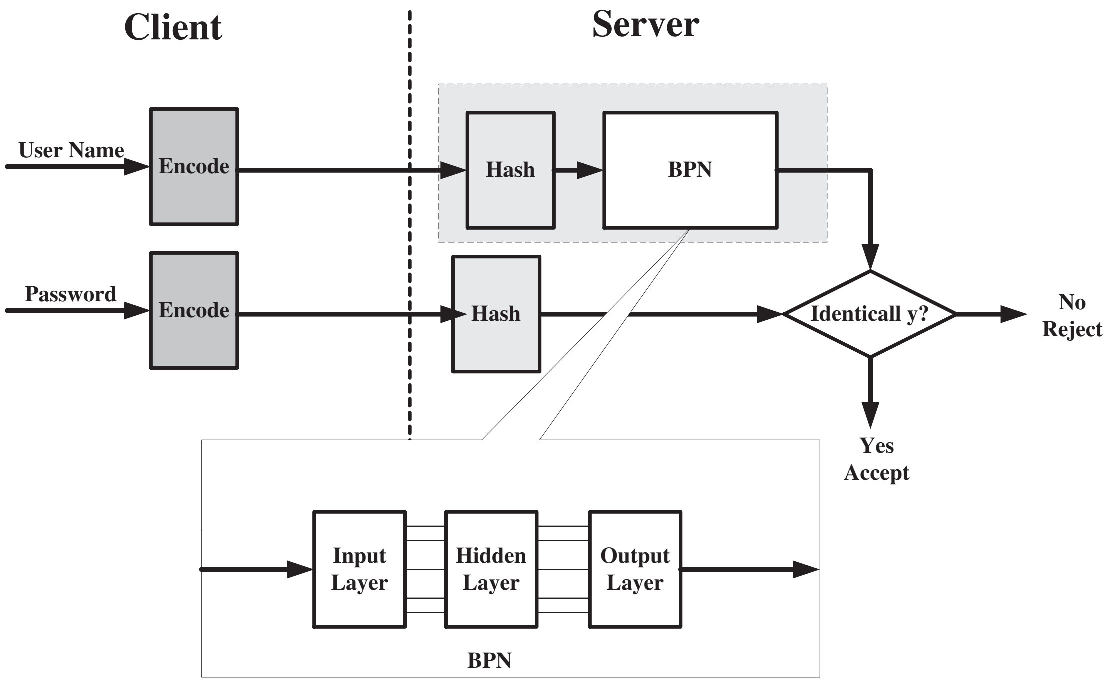

Figure 5: The processes of the login and user authentication phases

3. After encoding the username and the password, a one-way hash function is applied to them and the results is the train pattern. Generally, one-way hash function has the three features: (1) Given an input, it is easy to compute the output through the hash function; (2) Given an output, it is difficult to derive the input; (3) Given an input, it is difficult to find another input that has the same output [4, 17].

4. The SA takes the hashed usernames as the inputs and the hashed passwords as the excepted outputs to train the BPN. When the training process is completed, the SA stores the network weights in the system.

- **The Login Phase:**

  When a legitimate user wants to log into the computer system, the user has to input both the username and the password. Then, the client site encodes the username and the corresponding password, and then produces an encoded username and password automatically. Finally, the login request would be sent to the server, which includes the the encoded username and password. Figure 5 illustrates 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 applies the same one-way hash function and the trained BPN to authenticate the legality of the login user. The authentication process is described as follows.

  1. First, the server hashes the encoded username and password.
  2. Input the hashed username, then the server products an output through the trained BPN.
  3. The server compares the output with the hashed password. If the comparing result is positive, the login user is recognized as an authorized user. If the result is negative, the login user is rejected as an illegal user.

3 Experimental Results and Discussions

In our scheme, the training model "Back-Propagation Network (BPN)" is a supervised learning model. This model consists of three layers: the input layer, the hidden layer, and the output layer. The training pattern: hashed username for input and hashed password for expected output. The training set of the experiment is shown in Table 1. In this experiment, we assumed the user authentication system had 200 users. Each username and password is consisted of eight characters, and it 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. Note that each input and output unit is digital data (0 or 1). The training processing could be stopped when the sum of squared error (SSE) reaches its minimum or the error has not changed. 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				

3.1 Accuracy and Performance Analysis

The accuracy of the proposed scheme would be turned out good because the trained BPN makes its output approached the excepted output as close as possible. Furthermore, a threshold is defined by the system after the training phase, which is the minimum range of error tolerance. When a user submit a pair, (ID, PW), to the system, the received ID and PW in wrong formats would be rejected first. Then the system inputs the hashed username to the trained BPN. If the errors in all output units are less than the threshold, the system accepts the output result and uses it to verify the received hashed password. For a simple example, the system defines the threshold at 0.1. If the output result is $[0.01, 0.03, 1.00, 0.06, 0.94, 0.99]$, the system would accept the output result as $[0, 0, 1, 0, 1, 1]$. In contrary, if the output result is $[0.11, 0.03, 0.96, 0.96, 0.94, 0.99]$, the system cannot accept the output result because the error in the first unit is larger than threshold.

In the investigation phase, we first use the same training patterns to test the trained network. In the test, if we input the right username, the output is the corresponding hashed password. The reason is that each registered username and password has been used in training of BPN, so the error could be limited to less the threshold. In contrary, if we randomly input a unauthorized username (we test 100 attempts), the output from the BPN is never accepted and equal to the existed hashed passwords. In other words, the unauthorized login user would never be able to log into the computer system.

The performance of our proposed scheme is discussed as follows. In the training process, the system spends 257 minutes on training the BPN. The reason to set up such a long training time is that the input is digital data (0 or 1), and the convergence is too hard to be related within the training pattern. Note that the usernames and passwords in our test training data are distinct because if two or more usernames have the same password, the training BPN will be difficult to converge. Thus, the system has to avoid users selecting the same username and password.

After completing the training process, the system can be used to authenticate the identity of the login user efficiently. In this process, unlike public key cryptography that requires exponential computing, our system would only request simple multiplication and addition to produce the result. The BPN time complexity is $O(1)$. Thus, when a user wants to log into a computer system, the system could quickly response the result, which might accept or reject the user's request.

The main drawback of this scheme is that we have to spend a long time on training the BPN network. However, the whole training process needs to be performed in condition of initial processing, system user increase, or any password changed by users. The main advantage of this scheme is that computational overhead could be largely reduced in the authentication phase. Generally, the frequency of authenticating the legitimacy of login users is higher than new user increase or password changed. Therefore, the scheme could be used to the applications that require real time response or low computational capability machine, such as the utilization of user authentication for mobile phone.

3.2 Security Analysis

In our proposed scheme, the username and the password are encoded into test patterns through one-way hash functions [4, 17]. The security of our proposed scheme is based on the difficulty of inverting the test pattern to original password. Therefore, an intruder cannot easily derive the secret password from the test pattern even he/she knows the weights of the trained BPN. This feature is similar to UNIX system in which the verification table (/etc/password) could be public, the weights of trained BPN in our proposed scheme also could be public. Therefore, only the registered user knows both correct the username and the corresponding password, and only the correct username and password can help an authorized user make it pass the authentication.

A potential attack in our proposed scheme is that an intruder may get the weights of the trained BPN and try all possible passwords to verify them until a match occurs. Since the attack can be done off-line, we cannot limit the number of attempts. Such attack is called guess attack [25], and it has become a major threat to the security of UNIX system especially the users usually tend to select easy-to-remember password [13], which neither UNIX system nor our proposed scheme could avoid. An efficient way to prevent the guess attack is to select a "strong password", e.g. the password must contain at least two alphabetic characters and at least one numeric or special character. Furthermore, in [25], a simple approach was presented to make the guessing passwords based on one-way hash functions 100 to 1000 times harder. In this approach, two additional salts, one public and one secret, are appended to each password, and then we encode it into a test pattern through an one-way function, such that $test\ pattern = F(password\&secretsalt\&publicsalt)$. The public salt is stored in the system, and the secret one is discarded by the system after use. The approach is independent of the one-way hash function used and does not change the system model. Therefore, it could be directly used in our proposed scheme to make guessing attack much more difficult.

The proposed scheme is used a trained BPN to replace the verification table to verify whether the submitted hashed password is identical to the stored hashed password or not. If the verification table can be modified, an intruder can easily append a forged pair $(ID, F(PW))$ to the table. However, in the condition of employing the weights of trained BPN, if an intruder wants to log into the system successfully, he/she has to collect collect each user's ID and hashed PW to retrain the BPN or change the weights in order to match the username with the excepted output. Since each user's username and password is combined in the trained BPN, adding a forged user to the BPN is harder than adding a forged entry to the verification table. Therefore, the proposed scheme provides greater security than using verification table. Furthermore, we also could directly combine our scheme with Manber's simple approach [25].

3.3 Comparisons

As we know, most of the existed schemes [2, 3, 5, 7, 8, 9, 10, 12, 18, 19, 25] for user authentication are using cryptography technologies. The security of these technologies is usually based on the level of difficulty to factor a large numbers or calculate discrete logarithms in a finite field. As the modular exponential operations are usually required in these schemes, they are actually inefficient and time-consuming. In addition, some schemes [5, 9, 18, 19, 25] employed password table or verification table, and some schemes [2, 3] the user cannot choose the username and password freely.

In this scheme, we proposed a new password authentication scheme by using neural network. The contributions of this scheme is that it can deal with the drawbacks in the schemes that use verification table. Although this scheme requires more overheads to train and retrain the network, it also can meet the all requirements of user authentication.

4 Conclusions

The user authentication system in a traditional computer network has to maintain a password table or verification table in the server. In contrast, our method employs the BPN to recall the relationship of username and password. This method can easily produce the corresponding hashed password according to the input username, and it could be used to replace the password table or verification table stored in the system. Although the system still needs to store the parameters of the trained network, the stored parameters do not leak any sensitive information out. The advantages of our method are as follows.

- Instead of password table or verification table, the proposed scheme can prevent an intruder from adding a forged $(ID, F(PW))$ to the network.
- 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 $O(1)$.
- The proposed scheme offers all the users in the system to select their

username and password freely.

- No third party is necessary in the authentication process.

Besides, the long training time is the major weakness in our proposed scheme. Finding a solution to reduce the training time will be another work in the future.

Acknowledgements

The authors wish to thank many anonymous referees for their suggestions to improve this paper.

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

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

[9] M. S. Hwang, C. C. Lee, and Y. L. Tang, "An improvement of SPLICE/AS in WIDE against guessing attack," *International Journal of Informatica*, vol. 12, no. 2, pp. 297–302, 2001.

[10] M.-S. 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] ISO/IEC 9797, "Data cryptographic techniques-Data integrity mechanism using a cryptographic check functionemploying a block cipher algorithm," *Internal Organization for Standardization*.

[12] 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.

[13] D. L. Jobush and A. E. Oldehoeft, "A survey of password mechanisms: weakness and potential improvements," *Computers & Security*, vol. 8, pp. 587–604, 1989.

[14] L. H. Li, I. C. Lin, and M. S. 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.

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

[16] R. C. Merkle, "One way hash function and DES," *Advances in Cryptology-CRYPTO'89*, pp. 428–446, 1989.

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

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

[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, Dec. 1978.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] U. Manber, "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.

[26] C. Y. Yang, C. C. Lee, and S. Y. Hsiao, "Man-in-the-middle attack on the authentication of the user from the remote autonomous object," *International Journal of Network Security*, vol. 1, no. 1, pp. 22–24, 2005.