Proceedings of the 13th IASTED International Conference
Software Engineering and Applications (SEA 2009)
November 2 - 4, 2009 Cambridge, MA, USA

A FAST AND SECURE SOFTWARE SOLUTION [SS7.0] THAT COUNTERS SHOULDER SURFING ATTACK

Special International Student
Divyans Mahansaria, Samarpan Shyam, Anup Samuel, Ravi Teja
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge
USA
divyansmahansaria@hotmail.com

ABSTRACT
Shoulder Surfing is a direct observation technique, such as looking over someone’s shoulder to trap the information. It is relatively easy to stand next to someone and watch, what data the user types as an information to authenticate himself\herself to enter into a particular system. Often users are unaware of the presence of any external device, which may be placed in order to trap the user and obtain valuable information, such as, passwords, while the user types through. In order to control the loss in authentication information due to Shoulder Surfing we have developed a novel software solution. It could be used in computers, ATM machines etc. According to our software, typing the password information even in front of others will not lead to Shoulder Surfing and not even allow others to understand the password quickly and grasp the authenticated information. This paper deals an overview of Shoulder Surfing with a direct plunge into the various aspects related to our software solution [SS7.0] and outperforms well in highly confidential situations. It is an upgraded version with extra security and is also much faster than the previous version. It has an inbuilt encryption feature for passwords which is based on RSA security algorithm. We have showcased the mathematical and performance analysis of our software solution.

KEY WORDS
Shoulder Surfing, Authentication, Security, SS7.0, Shoulder Attacker

1. Introduction

Identity theft refers to fraudulence that involves stealing money or getting other benefits by pretending to be someone else. The person whose identity is used faces various consequences when held responsible for the perpetrator's actions. The person who is truly concerned about identity should certainly make them selves familiar with Shoulder Surfing. In reality, this terminology is used to describe one of the many ways criminals obtain the personal information they need, to commit identity theft. In this paper we provide a highly secure password entry solution which is resistant against Shoulder Surfing. In the first section of the paper we provide a brief description about the concept of Shoulder Surfing along with few tips to reduce Shoulder Surfing which can be implemented by a casual user. Second section covers some of the related research works that have already been carried out in the area of Shoulder Surfing. We then move on to our software solution. The software solution developed by us employs an encryption feature which is useful in preventing other forms of attack other than Shoulder Surfing. The performance in mathematical and experimental analysis pertaining to it has been briefed in section 4. In section 5 we show the simulation of the developed software and then list utilities & certain constraints associated with it in subsequent sections.

1.1 What is Shoulder Surfing?

Shoulder Surfing is using direct observation techniques, such as, looking over someone's shoulder, to get information. Shoulder Surfing is an effective way to get information be it in a user’s home while he works on his personal computer or in a public place which is more prone to Shoulder Surfing attack. Shoulder Surfing can also be done long distance with the aid of binoculars or other vision-enhancing devices. The increase in number of laptop and personal digital assistant (PDA) usage has greatly increased the danger of unauthorized observation of authentication procedures. The users have become more prone to password theft due to such kind of sneaking. Especially when the users are moving around it is difficult for them to keep a strict vigilance on their surroundings. They could be easily trapped be someone who is viewing the traveler’s authentication information.
One should remain cautious of his/her surroundings if he/she is authenticating by the traditional authentication methods prone to Shoulder Surfing.

1.2 Proposed methods of Reducing Shoulder Surfing on a Small Scale

Shoulder Surfing certainly is not the most technical form of identity theft, but many have used this method to commit major fraud. The first step in the prevention of it is in understanding that this problem does exist. There are certain precautions which may be taken by the user on a small scale while authenticating in any system, that are presently not using any prevention techniques to control Shoulder Surfing. Shielding keypad from view by using body or cupping by hand while typing passwords - is obviously one such method. One should experiment and create best password. It is advisable to use mixtures of numbers and letters rather than single, simple words for passwords. One should never carry important letters or statements from banks or building societies.

These documents, along with credit or debit card, can be a treat to any, would-be robber. One should always remember to dispose of the receipts carefully after completing an ATM transaction. It is not a direct solution to Shoulder Surfing, but doing so can be a bit handy when it comes to protecting customers from revealing their personal information to strangers (probably a Shoulder Surfer).

2. Related work

There have been some countermeasures used in a few products to prevent peeping attack. Few research proposals pertaining to it have also been proposed. But a fully functional solution which could be widely used in several applications in order to control Shoulder Surfing has not been deployed yet.

One of the schemes proposed is that of Pass faces [1].

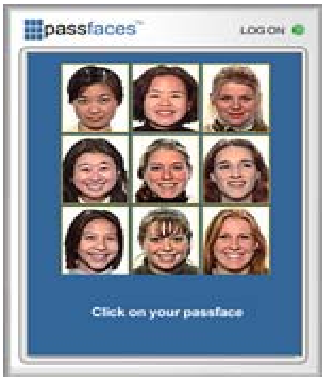


Figure 1. Pass faces of Web Access Components

It is a challenge response scheme. A user chooses a set of images as his password. While authentication a user needs to select the chosen images in the serial order of his selection. When one picture is selected a new set of images for subsequent selection appears. In this method a user can authenticate by going through several rounds of image selection (which is actually equivalent to the password length). This method is prone to Shoulder Surfing attack because one can easily view the position of the mouse cursor while authentication and the picture can be noted. A scheme similar to this has been proposed by S.Bindu etal [2]. Here the Pass faces are arranged in a similar fashion and challenge response scheme is carried out. A user enters the coordinates of a particular Pass face rather than choosing it directly.

Wiedenback etal [3] describes a graphical password entry scheme using convex hull method towards Shoulder Surfing attacks.

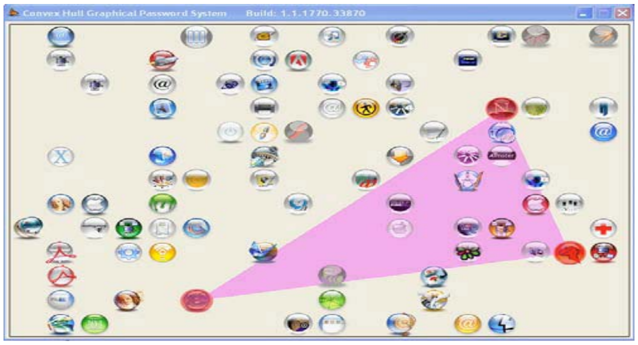


Figure 2. Example of a convex hull

A user needs to recognize pass-objects and click inside the Convex hull formed by all the pass-objects. In order to make the password hard to guess large number of objects can be used but it will make the display very crowded and the objects almost indistinguishable, but using fewer objects may lead to a smaller password space, since the resulting convex hull can be large.

Our software solution employs a different scheme than the already existing methods. It is neither graphical in nature nor based on a challenge response scheme. It is one that could be deployed in the usual real time systems. It eliminates the unnecessary complexity. An encryption technique is also inbuilt in it which takes care of other forms of attack apart from shoulder surfing. We have also used the concept of most frequently occurring alphabets in English to make the password entry faster. Thus it offers a complete security while authentication.

3. Our Solution

In the proposed scheme, users create an authentication account initially. Information regarding a particular user such as his/her User Name and Password is taken. For security reasons, we propose that the chosen password should be at least 7 characters in length and maximum length is 20 characters. This information is, then, stored in a Database. Any number of users can register in the system and a complete database which contains the entire information about the user along with their Usernames & Passwords can be formed. Most importantly, this database is hidden from the user and only accessible to the system programmer or administrator of the particular system.

Let us suppose that, at a later point of time, someone wants to logon to a system (here system need not be a standalone one, a user could perform remote login too) which contains the information about several users who have already registered and have the right to use the system. The incoming user will be asked to enter his authentication information, Username & Password as is usually done for a secured system. We have an interactive screen where, as usual, the username & password need to be entered. The username will be entered in the usual fashion as is done in most computer systems. But the trick lies while entering the password. The software uses an inbuilt novel technique to make the users enter their password.

As the cursor is clicked on the password field a popup box appears. It contains an 8*6 order matrix (i.e. 8 rows and 6 columns). The rows are numbered using the numbers 1 to 8 and columns are numbered using the numbers from 1 to 6. The elements of the matrix will be a randomly generated set of alphabets, numerals and symbols without repetition of any alphabet, numerals and symbols in the matrix.

The English alphabets have varying relative frequencies among each other. The relative frequency of various alphabets of English as per LEWA00 is shown in Fig.3 [10]. In the first row of the matrix we randomly arrange the most frequently occurring alphabets of English as per Fig.3. Therefore we choose E, T, A, O, I and N as the first row elements of the matrix.

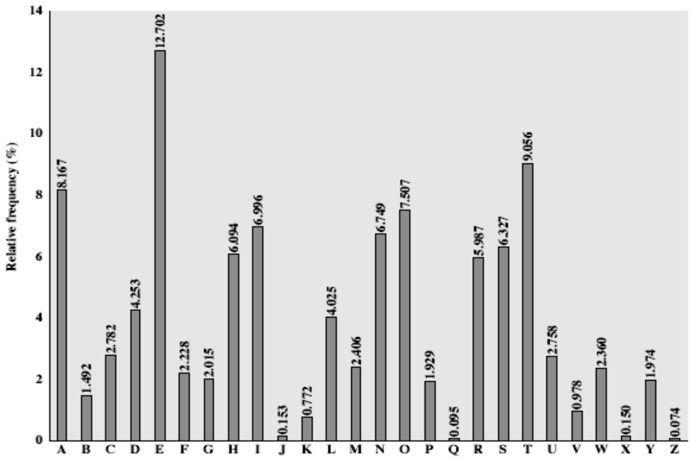


Figure 3. English Letter Frequencies

We have chosen this because it will be easier and quicker for users to scan the matrix to find the alphabets pertaining to the password.

In the next five rows, the remaining 30 alphabets and numbers are randomly arranged.

We provide the option to include symbols in password as well. It will make the password more secure. For this we have selected 12 commonly used symbols for passwords.

These symbols are randomized in the last two rows. Again they are not mixed with the alphanumerical characters to facilitate faster and easier scanning of the matrix by the user. Thus three different randomizations are carried here one each for 1st row, 2nd – 6th row and 7th – 8th row.

The entire 26 English alphabets, 10 numerals (0-9) and 12 chosen symbols fill the matrix. Now instead of entering the actual password the user uses a novel phenomenon. In the ‘password field’, the user will enter the positions of the constituent characters (alphabets, numerals or symbols) of his password, from the given matrix, for the initial characters except for the last three characters of his password. For the last three characters of the password the user will enter the usual characters of the password without using positions from the matrix.

Let us suppose, for example, the password corresponding to the username “SECURITY” is “1DEI*2DTA#3”. In this case, the user enters “SECURITY” in the ‘username field’. In the password field, the user will enter the position of the character ‘1’ followed by the positions of the characters ‘D’, ‘E’, ‘I’, ‘*’, ‘2’, ‘D’ and ‘T’. As the position, the user will enter first the row number of a particular element, then just after that the column number of that particular element. Let us assume we have the matrix for a particular iteration as:-

	1	2	3	4	5	6
1	A	O	E	N	I	T
2	9	2	H	J	K	Q
3	6	0	Y	S	3	L
4	U	V	7	B	W	P
5	C	M	8	4	R	F
6	X	Z	D	1	G	5
7	!	@	#	$	%	:
8	&	*	(	)	?	“

Figure 4. SS concept

In this matrix, the positions that the user has to enter corresponding to the password “1DEI*2DTA#3” can be calculated as:-

For ‘1’ – position is 64 (i.e. 6th row & 4th column), for ‘D’ – position is 63 (i.e. 6st row & 3rd column), for ‘E’ – position is 13 (i.e. 1st row & 3rd column), for ‘I’ – position is 15 (i.e. 1st row & 5th column), for ‘*’ – position is 82 (i.e. 8st row & 2nd column), for ‘2’ – position is 22 (i.e. 2nd row & 2nd column), for ‘D’ – position is 63 (i.e. 6th row & 3rd column), for ‘T’ – position is 16 (i.e. 1st row & 6th column).

Thus, the user will enter “6463131582226316” as his password position. Now for the last three characters of the password the user will enter the password characters as usual. Thus, instead of entering the password “1DEI*2DTA#3”, the user will be entering “6463131582226316A#3” as his password. The Matrix will generate random elements for each new login (i.e. for every subsequent authentication the matrix elements are going to dynamically change).Now, let us suppose that the matrix for new login changes to the one as show in fig.5.

	1	2	3	4	5	6
1	E	O	A	I	T	N
2	0	V	5	Y	S	9
3	F	3	X	7	B	6
4	L	W	J	K	Q	U
5	4	R	Z	D	1	C
6	M	P	8	G	2	H
7	:	@	*	$	?	&
8	!	#	(	)	%	“

Figure 5. SS concept

In this matrix the position for the same password “1DEI*2DTA#3” can be calculated as:-
For ‘1’ – position is 55 (i.e. 5th row & 5th column), for ‘D’ – position is 54 (i.e. 5th row & 4th column), for ‘E’ – position is 11 (i.e. 1st row & 1st column), for ‘I’ – position is 14 (i.e. 1st row & 4th column), for ‘*’ – position is 73(i.e. 7th row & 3rd column), for ‘2’ – position is 65 (i.e. 6th row & 5th column), for ‘D’ – position is 54 (i.e. 5th row & 4th column), for ‘T’ – position is 15 (i.e. 1st row & 5th column).
Thus, the new position of initial characters (excepting last three) is “5554111473655415”. Thus, instead of entering the password “1DEI*2DTA#3”, the user will be entering “5554111473655415A#3” as his password which is quite different from the previous logon password.

4. Performance and Mathematical Analysis

We have assigned the password length as minimum of 7 characters and maximum of 20 characters. Thus, total possible combinations of choosing a password of length ‘L’ is
C = ((48) ^ L)
Where C is equal to the number of combinations &
7 <= L <= 20
Thus for password length equal to 7 characters we have total choice of
((48) ^ 7) = Approx. (10^ 11) ways
& for password length equal to 20 characters we have total choice of
((48) ^ 20) = Approx. (10^ 33) ways
Thus we can see that we have a wide range of combinations for selecting the password. Thus it will be very difficult for an unauthorized person to enter into a system by merely guessing a password of another user.
The 6 most frequently occurring elements can be arranged in first row of the matrix in (6!) (‘!’ represent factorial) ways. The elements in the other 5 rows can be arranged in (30!) ways. The last two rows containing symbols can be arranged in (12!) ways. Thus the entire matrix can be arranged in:
(6!) * (30!) * (12!) Ways
= (720) * (10^32) * (10^8) Ways (approx.)
= (10^42) Ways (approx.)

It shows that we have a wide range of arrangements possible in the matrix and thus making it very difficult to break the security.
In order to check the average time taken by the users to type the password using our scheme we chose 14 different passwords varying in length from 7 to 20. Each of these 14 different passwords had a minimum of two alphabets, two numbers and two symbols. The password was a complex one. We analyzed the time taken by 100 users to key in these 14 chosen passwords. Based on our analysis we obtained the curve as shown in fig.6.

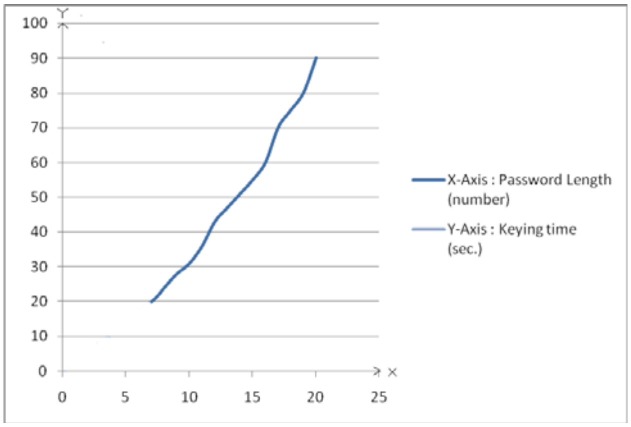


Figure 6. Password length Vs Keying time (Using SS7.0)

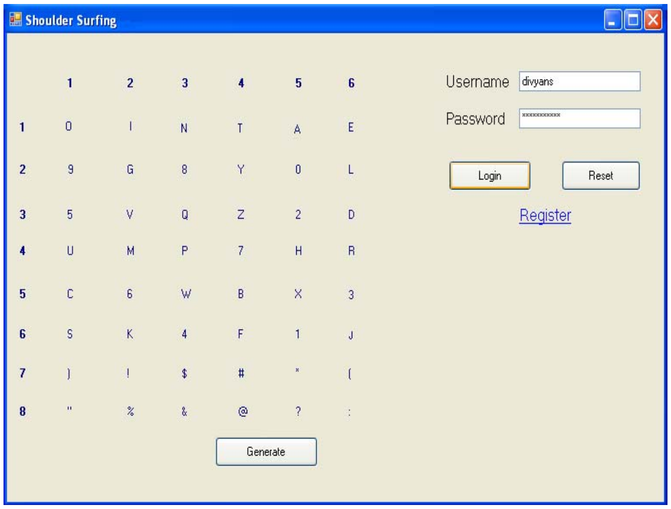
This curve shows that on an average a user takes around 20 seconds to key in the password of length 7 (which is the minimum password length set by us) and around 90 seconds to key in the password of length 20 (which is the maximum password length set by us). Thus, one could even use a password of length as long as 20 which take just about 90 seconds to type. Such type of long password could be used for high end applications such as in military etc.

5. Simulation

Step 1: A user starts the system to logon.

Figure 7. User Logon

Step 2: In order to logon a user initially requires authenticating himself/herself by providing user name and password. There will be two fields: one containing user name & the other containing password. The user name is entered, as usual. While entering the password, a new scheme is being applied. As is evident from fig.8, a randomized matrix of alphanumeric characters and symbols is present. The user has to find out the corresponding position of his/her password and provide the positions as his/her password in the password field (excepting for the last three elements of the password which the user enters as usual). Then 'Submit' is clicked for verification of the correctness of the provided username & password.



Shoulder Surfing

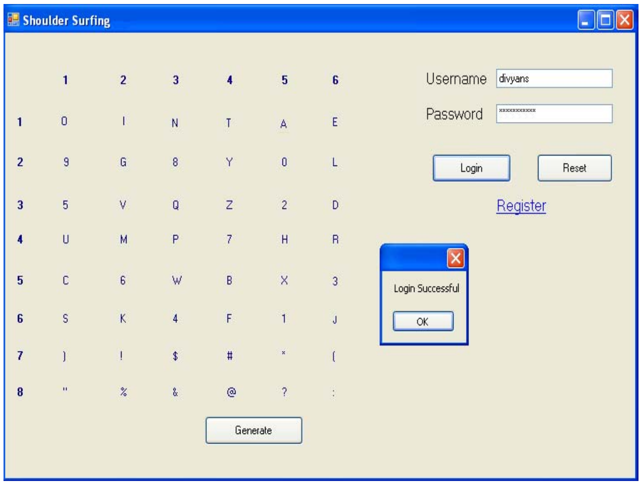
	1	2	3	4	5	6
1	O	I	N	T	A	E
2	9	G	8	Y	0	L
3	5	V	Q	Z	2	D
4	U	M	P	7	H	R
5	C	6	W	B	X	3
6	S	K	4	F	1	J
7	)	!	$	#	*	(
8	"	%	&	@	?	:

Username divyans

Password ***********

Figure 8. Authentication

Step 3: A comparison between the entered information for authentication and the already existing user names & passwords in the database is made. If there exits such an account, then the login becomes successful.



Shoulder Surfing

	1	2	3	4	5	6
1	O	I	N	T	A	E
2	9	G	8	Y	0	L
3	5	V	Q	Z	2	D
4	U	M	P	7	H	R
5	C	6	W	B	X	3
6	S	K	4	F	1	J
7	)	!	$	#	*	(
8	"	%	&	@	?	:

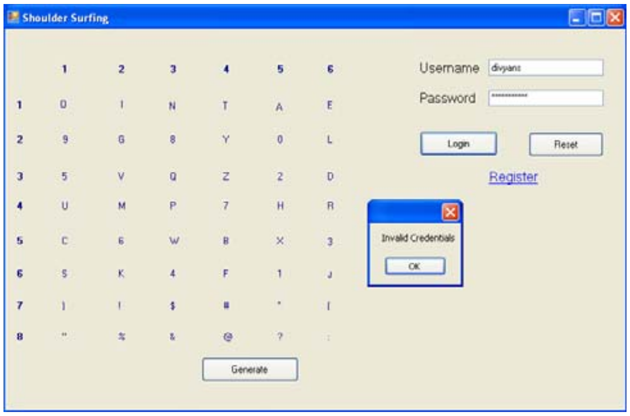
Username divyans

Password ***********

Login Successful

Figure 9. Login Successful snapshot

However, if the authentication information is incorrect, then the login fails & authentication information may again be retaken from the user



Shoulder Surfing

	1	2	3	4	5	6
1	O	I	N	T	A	E
2	9	G	8	Y	0	L
3	5	V	Q	Z	2	D
4	U	M	P	7	H	R
5	C	6	W	B	X	3
6	S	K	4	F	1	J
7	)	!	$	#	*	(
8	"	%	&	@	?	:

Username divyans

Password ***********

Invalid Credentials

Figure 10. Login Failure snapshot

6. Utilities of Using our proposed Software

Let there exits a sneaker who tries to figure out the authentication details of a user through Shoulder Surfing.
If a system deploys our software solution, the sneaker's efforts would go in vain. A sneaker can either look onto the keyboard or look at the screen at a time. If he looks onto the keyboard then what he will get to see is a false authentication password of the user. Suppose the one who sees the password tries to keep the password in mind and waits for the user to leave the system and then reenters the positions entered by the user previously, his effort will go in vain because after each login the positions of the elements in the matrix are dynamically changed. We can also avoid the loss of passwords which could have been obtained otherwise through the use of binoculars, closed-circuit television cameras or other vision-enhancing devices that a shoulder attacker may use in order to trap a user.
Yet another effective advantage of using it is that it involves figuring out positions. It has been biologically proven that such type of mental exercise improves one's cognition. It does not involve cases like covering ourselves and the machines with a cloth which is highly unprofessional, more time consuming as well as dangerous. We have inculcated RSA encryption in the software. The encryption and decryption process is carried out automatically without the user's involvement. Thus, the sending of the password to a remote database to check for its correctness particularly in a networked environment will not cause any problem. The havoc of loss of passwords through illegal tapping of messages during its transportation can be eliminated through this.
We have divided the matrix into three parts. The first row contains most frequently occurring alphabets of English.
The last two rows contain only symbols. The other elements are arranged in second row to sixth row. This helps in figuring out the elements of the user's password in a quicker and easier way.
From the discussions we can see that our proposed software solution could be a novel solution in controlling Shoulder Surfing in particular.

7. Certain Constraints in Using this Software

While it is by far suitable in controlling Shoulder Surfing, it does suffer certain drawbacks. The proposed mechanism of password entry is a new one so users need to be educated about the new password entry method (although the methodology is very simple). The login time will increase than usual. But keeping the high performance and other benefits in mind, we can compensate on the time taken for initial logon.

8. Conclusion

Password theft protection is of vital concern. Unfortunately, today's standard methods for password input are subject to a variety of attacks based on observation, from casual sneaking (Shoulder Surfing), to many other forms of attacks. The method presented by us can be very useful in controlling "Shoulder Surfing". Our software solution can be used while initial login after booting of a computer, during authentication which may be required before using particular software, opening important documents etc. It can be integrated with any email service provider. It could also be used in websites wherever a user name & password is initially required for authentication. Thus it covers a wide domain of services in a computer system where initial authentication is a must. With necessary changes the same scheme can also be employed to ATM Machines and other forms of electronic devices which requires authentication before giving access to its users. Thus we see that the scheme employed in our software solution finds its usage in a wide variety of different applications.

Acknowledgement

We would like to thank the department of EECS of Massachusetts Institute of Technology (MIT); Cambridge, USA for helping us in the presentation and publication of our work.

References

[1] Real User Corporation: Passfaces. www.passfaces.com

[2] S.Bindu, Raj Mohammed "A Novel Cognition based graphical Authentication Scheme which is resistant to shoulder surfing attack", *Proceedings ICIP 08*, I.K. International, Bangalore, August, 2008.

[3] S.Wiedenbeck, J.Waters, L.Sobrado, and J.C.Birget, "Design and Evaluation of a Shoulder-Surfing Resistant Graphical Password Scheme", *Proc. of Advanced Visual Interface (AVI2006)*, pp.23-26, May (2006).

[4] Behzad Malek, Mauricio Orozco and Abdulmotaleb El Saddik "Novel Shoulder-Surfing Resistant Haptic-based Graphical Password" *Proc. of the EuroHaptics 2006 conference*, July 3-6 Paris, France

[5] Bogdan Hoanca, Kenrick Mock, "Screen Oriented technique for reducing the incidence of shoulder surfing". *Security and Management 2005*: 334-340.

[6] Manu Kumar, Tal Garfinkel, Dan Boneh, Terry Winograd,"Reducing Shoulder-surfing by Using Gazebased Password Entry". SOUPS '07: *Proceedings of the 3rd symposium on Usable privacy and security, July 2007*, Publisher: ACM.

[7] Tetsuji TAKADA "fakePointer: An authentication scheme for improving Security against Peeping attacks using video Cameras". *UBICOMM08*, Sept. 29-Oct. 4 2008 Page(s):395 – 400.Publisher – IEEE Computer Society.

[8] "Graphical Passwords: A Survey" Xiaoyuan Suo Ying Zhu G. Scott. Owen. *Proceedings of the 21st Annual Computer Security Applications Conference*, 463 - 472, 2005.

[9] William Stallings "Cryptography and Network Security", 4th Edition. Publisher–Pearson Education Inc.

[10] Lewand, R. "Cryptological Mathematics". Washington, DC: Mathematical Association of America, 2000.