Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 48 (2012) 9563-9571

ISSN: 2229-712X

Random pattern based flexible user interface for an effective secured authentication protocol

R. Sujatha[1] and G. Arumugam[2]

[1]SSE Project, Department of Computer Science, Madurai Kamaraj University, Madurai, India.
[2]Department of Computer Science, Madurai Kamaraj University, Madurai, India.

ARTICLE INFO

Article history:
Received: 31 May 2012;
Received in revised form:
28 June 2012;
Accepted: 23 July 2012;

Keywords
Confidentiality,
Security, Server,
Authentication Protocol,
Random Patterns,
Flexible User Interface based
Authentication System and
Authentication.

ABSTRACT

The development and maintenance of user interface software is surely a challenging task. Organizations who use secured authentication system tolerate no leakage. Cryptographic primitives are useful tools but security of these primitives does not guarantee security of the system. Using patterns for authentication is a system that provides patterns as passwords to the users. Users tend to choose their passwords through random art generation which can be captured by malicious users by video capturing or photo clicking. In lieu of the traditional password based system, several attempts had been reported in literature about authentication schemes which are successful in increasing the strength of the system against some of the known attacks. In this paper, Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol (SAP-RP) is presented. With iterations of random patterns, the users enter different passwords (which are based on the original images selected by them) for every login attempt that is converted to hashed value. It is compared with registered hashed value stored in the database which ensures confidentiality and authentication in the network plane using server and flexible user interface based authentication mechanism.

Introduction

Designing [1,3,12] flexible and dynamically configurable user interfaces is challenging. It is unlikely that developers will come up with a solution for all problems that is appropriate for all users. Users often like to customise their interfaces, as evident from different preferences in office and desktop designs. The system should provide flexibility not only to the individuals but also across different groups. A groupware infrastructure is defined by three dimensions (usability 1st (1999)): a) Communication (pushing or pulling information out into an organization, distributed environment, networking etc.), b) Collaboration (using shared information and building shared understanding), and c) Coordination (concurrency support and latecomer support).

The development and maintenance of user interface software is challenging. The interface development environments provide facilities that allow individual components within an interface to be constructed without recourse to programming. But the behaviour of user interfaces is generally implemented by complex, hand crafted software systems. The design patterns can be used to provide an organisational framework for interface software. But still it is the case that user interface software is intrinsically complex. Changing an existing interface to reflect changing requirements and to take account of user feedback is a laborious and often somewhat ad-hoc process.

Protected Substantiation System is relevant for a computer system to process information with different sensitivities (i.e. classification of information at different levels) to permit simultaneous access by users with different security clearance and to prevent users from obtaining access to information for which they lack authorization. Secured Authentication System has two goals: First goal is to prevent unauthorized personnel from accessing information. Second goal is to prevent unauthorized personnel from declassifying information.

The solution presented here offers flexibility both at the group level and at the application level. At the group level, the shared workspace can be adapted by loading collaboration-specific that incorporate the collaboration and coordination dimensions into the user interface. At the application level, the user can choose between multiple image modalities to interact with the application.

In this paper, we propose Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol. In this authentication protocol, users choose their passwords from the flexible user interface patterns. User has to enter random patterns as their passwords for the chosen flexible user interface patterns. At every sequence of iteration these random patterns are varied and associated with hidden characters (assigned to patterns) at run time. Every registered user has hashed value and it should be compared with generated hashed value at login time, to see if it matches then authentication granted to the user otherwise denied. It overcomes the identified drawbacks of existing systems. The attacks on existing model embedded in encrypted sessions are detected as monitoring the processes taking part in the systems is integrated. The new system incorporates flexibility in mechanisms to provide security. Hence the inside information is protected, and also the outside attacks are prevented. To establish this, a server with flexible user interface authentication mechanism is used. Types of attacks [7, 8, 9, 10, 11] proscribed in the proposed system are Shoulder Surfing, Brute Force attack, Dictionary attack,

Tele:
E-mail addresses: sujamurali72@gmail.com

Keyloggers attack, Man-In-The-Middle attack and Database Server Compromise attack.

In section 2, related works are discussed with their drawbacks.

Section 3 discusses the overview of Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol methodology.

In section 4 implementation details related to the system are presented. Conclusion is given in section 5.

Related Work

Graphical Password: [12] Usable Graphical Password Prototype is a system that provides graphical passwords to the users to choose their passwords. Users choose their passwords in the form of entering the image positions at the specified columns in the login phase by clicking the images for authentication. This makes malicious users to hack the password easily through video capturing and shoulder surfing attacks.

Authentication Using Graphical Passwords: [13] Basic Results is a system that provides graphical passwords to the users for authentication. Users tend to choose their passwords by clicking pixel positions from a scene in a selected order. In this system if the users have forgotten their pixel password they have to do more attempts to choose the password. The same process can be applicable for malicious users. Pixel positions are tedious to choose as passwords. The process can be captured through video by hackers to acquire the password.

A User Study [14] Using Images for Authentication is a system that provides images as passwords to the users. Users tend to choose their passwords through random art generation issued by the system. Through video capturing or photo clicking the password generation can be captured by malicious users. It processes with more capacity to store the images. It takes more time for authentication process because of high pixel resolutions. Main demerit of this system is that the server needs to store the seeds of the portfolio images of each user in plain text. That makes hackers to acquire the generation of the passwords easily.

Passface is a technique developed by [15] Real User Corporation is a system that provides images as passwords to the users. The user is asked to choose four images of human faces from a face database as their password. During authentication user has to choose correct face from chosen faces in each stage. Drawback of this system is that most users tend to choose faces of people from the same race. It makes the Passface password predictable.

A Password scheme [16] strongly resistant to spyware is the system that provides images as passwords to the end users. The user will be asked to enter the alphanumeric code as password for the chosen images. The user has to assign his / her own code for the images, and that code is assigned to images as passwords. Normal text password makes users to forget details very often. This system also makes users to remember the alphanumeric code for the chosen images. It makes users again suffer the same trouble that they had in normal text password systems.

Enhanced authentication mechanism [1] using multilevel security model is the system that applies multilevel security. Any sensitive application includes confidential and secret information that must be used effectively in complicated and authenticated procedures. Using five levels of authentication methods with a set of privileges assigned, each user has to surpass 50% of every level to get the privileges rights. During authentication the information was hacked from the network plane using network analyser tool. Leakage of information occurred in three levels while transmitting answers with username and multiple questions methods.

In Improving [2] text password through persuasion, users enter their passwords with visibility. Users tend to choose their passwords in a simple manner by entering visibility method. It makes the hacker to know by the shoulder-surfing process.

Pass Pattern System [5] is a Pattern-Based User Authentication Scheme, where data can be hacked from the database through database compromise server attack.

There are several attempts reported in literature about authentication schemes in lieu of the traditional Password-based system. Each attempt is successful in increasing the strength of the system against some of the known attacks. They are either computationally intensive or they require additional hardware/software in the infrastructure. In this section we review the current attempts, identify the gaps and emphasize the motivation for developing Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol.

The proposed SAP-RP incorporates the essence of flexible user interface using random patterns in various environment systems. It is robust against attacks such as brute force, shoulder surfing, social engineering, database server compromise attack and Man-In-The-Middle attacks.

Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol

This system involves the use of authentication mechanism and a server that minimizes the hacking by the attackers. It monitors the clock cycle process effectively. Two processes are involved in this system. They are a) Authentication using Random Patterns with Flexible User Interface and b) Security Questions Authentication using server.

Authentication using Random Patterns with Flexible User Interface

This is a text-based authentication system using flexible user interface patterns, based on the premise that ‘humans are [6] good at identifying, remembering and recollecting picture patterns than text patterns’. This vision is incorporated in this system to authenticate the users in a memorized way.

The concept of authentication using images [18] can be modified with any kind of random pattern based flexible user interface patterns to avoid specific scenarios. More flexible user interface patterns are generated using this system and one can apply these patterns to various scenarios to provide secured authentication.

The authentication process implemented in the system is detailed in Figure 1. The authentication is done in two levels.

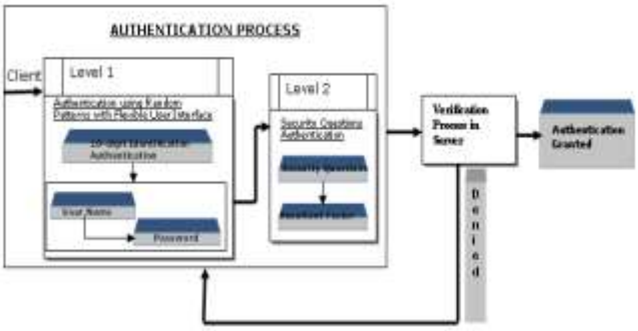
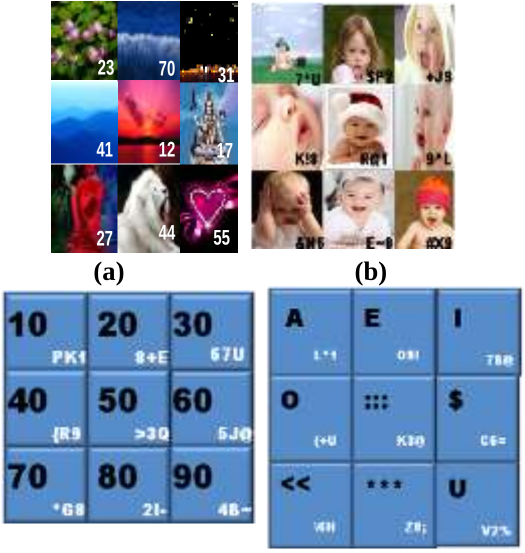

AUTHENTICATION PROCESS
Client
Level 1
Authentication using Random Patterns with Flexible User Interface
Authentication
User Name
Password
Level 2
Security Questions Authentication
Security Questions
Verification Process in Server
Authentication Granted
Denied

Figure 1: Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol Flow Diagram

In the first level, the client gets authenticated using username and password provided in the flexible user interface patterns. For providing the password, the client has to enter the random patterns displayed in the screen. It is illustrated in Figure 2.

For example, if the client chooses the password pattern as red rose, white lion and pink heart from Figure 2.a, then the random patterns 274455 should be entered in a selected order at password column. If the client chooses the password pattern as blue hat baby, red hat baby and orange hat baby from Figure 2.b, then the random patterns 7*UR@1#X9 should be entered in selected order in password column. If, suppose the client chooses the password pattern as 20, 50 and 90 from Figure 2.c, then the random patterns 8+E>3Q4B~ should be entered in selected order in password column.



(a) (b)

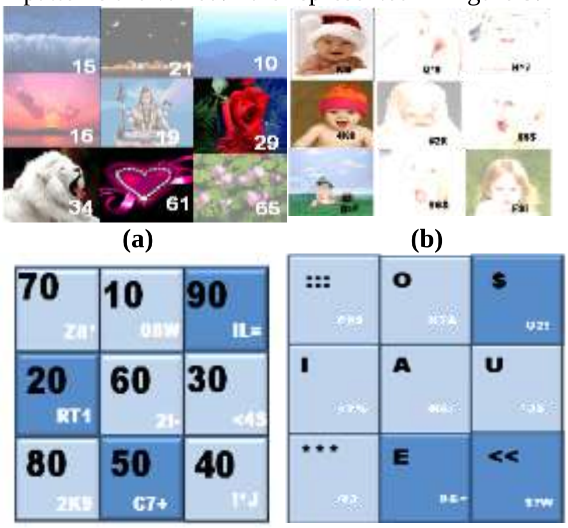
(c) (d)

Figure 2: A sample Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol.

If the client chooses the password pattern as E$<< from Figure 2.d, then the random patterns 09!C6=!6H should be entered in selected order in the password column.

While confirming password, random patterns are varied. The user has to re-enter password by giving different random patterns according to the password pattern chosen. In every login attempt, both password patterns and random patterns are represented as dynamic arrangements. Due to this setup no one would be able to read or guess the mechanism involved.

For every authentication, the patterns are shuffled and random patterns are varied. It is represented in Figure 3.



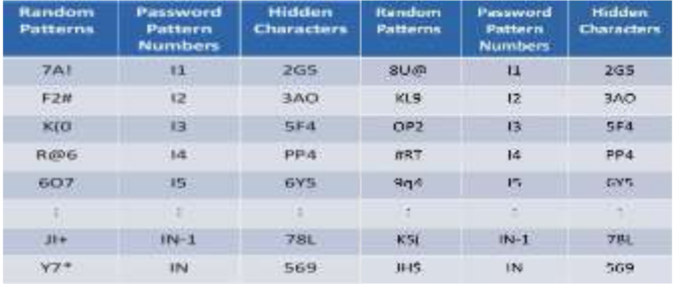
(a) (b)

(c) (d)

Figure 3: A sample shuffling mechanism of Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol.

The client has to confirm his / her password by enter the random patterns' according to the selected order given at the time of registration. According to the selection made during registration, the client has to enter random patterns as 293461 for Figure 3.a, for the password pattern red rose, white lion and pink heart. For Figure 3.b, the client has to enter random patterns as @=PK!84K8 for password pattern blue hat baby, red hat baby and orange hat baby. For Figure 3.c, the client has to enter random patterns as RT1C7+IL= for password pattern 205090. For Figure 3.d, the client has to enter random patterns as 0&-U2!$?W for password pattern E$<<.

During registration and login times, the password patterns displayed are shuffled. The random patterns are varied. Mapping associated between random patterns, password pattern numbers and hidden characters are represented in Figure 4. Each password pattern is mapped with a random pattern with hidden character as represented in Pattern-Map table for first iteration in Figure 4.a. On the next iteration, random patterns are varied and mapped with password pattern numbers. Hidden characters are represented in Figure 4.b.



Random Patterns	Password Pattern Numbers	Hidden Characters	Random Patterns	Password Pattern Numbers	Hidden Characters
7A!	I1	2G5	8U@	I1	2G5
F2#	I2	3AO	KL$	I2	3AO
K(O	I3	5F4	OP2	I3	5F4
R@6	I4	PP4	#RT	I4	PP4
6O7	I5	6Y5	9q^	I5	6Y5
⋮	⋮	⋮	⋮	⋮	⋮
JI+	IN-1	78L	K5(	IN-1	78L
Y7*	IN	569	JH$	IN	569

Figure 4.a: Iteration 1 **Figure 4.b: Iteration 2**

Figure 4: 4.a) Varied Random Patterns, Password Pattern Numbers and Hidden characters in Iteration 1 associated with mapping and 4.b) Varied Random Patterns, Password Pattern Numbers and Hidden characters in Iteration 2 associated with mapping.

Iteration 1:

Random Pattern Number Password Pattern Hidden Character

7A! → I1 → 2G5

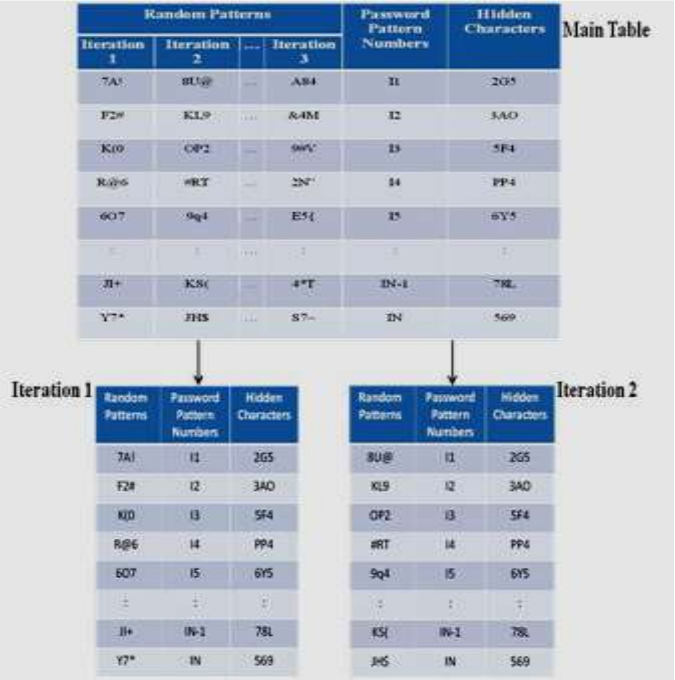
Iteration 2:

Random Pattern Number Password Pattern Hidden Character

8U@ → I1 →2G5

In both the iterations, random pattern is varied. It is associated with password pattern number and hidden character. These association processes persist for the entire password pattern present in the scenario. Instead of comparing password patterns, the associated random numbers and hidden characters are compared. It serves as user friendly for the end-user and machine friendly for the system, by reducing the comparison time by using numbers rather than password patterns. Hidden characters assigned to each password pattern are taken as password (according to the user choice in selected order). It is converted to a hash value and then stored in the database for validation.

At every time of login, the password patterns and varied random patterns are displayed on the screen according to the sequence of iterations. Each time the random patterns are generated dynamically and assigned to the password patterns as represented for two iterations in Figure 5.



Main Table

Random Patterns				Password Pattern Numbers	Hidden Characters
Iteration 1	Iteration 2	…	Iteration 3		
7A!	8U@	…	A84	I1	2G5
F2#	KL9	…	&4M	I2	3AO
K(0	OP2	…	9#V	I3	5F4
R@6	#RT	…	2N"	I4	PP4
6O7	9q4	…	E5{	I5	6Y5
:	:	…	:	:	:
JI+	KS(	…	4*T	IN-1	78L
Y7*	JHS	…	S7~	IN	569

Iteration 1

Random Patterns	Password Pattern Numbers	Hidden Characters
7A!	I1	2G5
F2#	I2	3AO
K(0	I3	5F4
R@6	I4	PP4
6O7	I5	6Y5
:	:	:
JI+	IN-1	78L
Y7*	IN	569

Iteration 2

Random Patterns	Password Pattern Numbers	Hidden Characters
8U@	I1	2G5
KL9	I2	3AO
OP2	I3	5F4
#RT	I4	PP4
9q4	I5	6Y5
:	:	:
KS(	IN-1	78L
JHS	IN	569

Figure 5: Password Pattern Numbers, Hidden Characters and Varied Random Patterns associated with main table in two iterations.

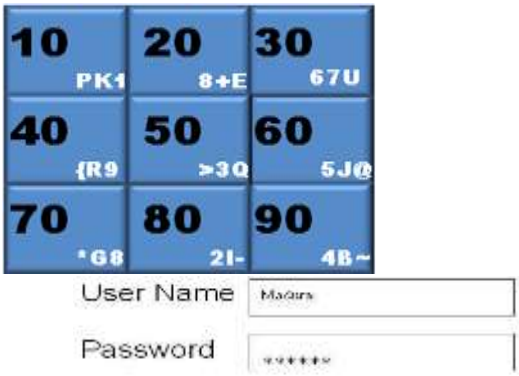
The password pattern positions are generated and displayed using permutation sequences. For n password patterns n! sequences are generated. They will be used randomly for every attempt of registration or login. The random patterns are generated dynamically. It will be assigned to the password pattern sequences at the time of every registration or login time. If the random number digit is n then n! sequences are generated for random patterns. A mapping mechanism which validates the random patterns with hidden characters is represented in Table 1. In the Table 1 the n value has 3 digits, then 999! sequences are generated and assigned to the password patterns.

The digits include alphabets (26), numbers (0-9) and special characters (32), which create 8320! (26 x 10 x 32) combinations of characters occurred for display. According to the number of digits taken, possible combinations of characters are framed and utilized in the display system. While entering random patterns in the password area, it will be hidden. Bullet marks will be displayed. Here the first level system is represented as a sample 3 x 3 matrix set of password patterns.

The user has to choose password as patterns. He / she should provide the random patterns in the password area. It is illustrated in Figure 6.

Table 1. A Sample password pattern map mechanism for SAP-RP

Password Pattern Numbers	Hidden Characters	Random Patterns			
		Iteration 1	Iteration 2	…	Iteration N!
BI1	1A	7A!	5H/	…	A84
BI2	2G	F2#	;8P	…	&4M
BI3	25	K(0	9+E	…	9#V
BI4	1C	*5L	U2=	…	2N"
BI5	2P	%I9	1^R	…	E5{
BI6	37	R@6	W3)	…	B3>
BI7	9L	6O3	?X5	…	4:T
:	:	:	:	…	:
BIN	5P	Y8$	0Q|	…	S7~



User Name Madurai

Password ******

Figure 6: A sample 3 x 3 Random Pattern based Flexible User Interface for an effective Secured Authentication Protocol.

User Registration Steps

a. After entering username, the user has to enter the password.

b. To enter password, the user has to choose password patterns as their password. Let us consider the username as "Madurai" and password patterns as "SSEMKU1234" shown in Figure 7.

c. Now the user has to enter their corresponding random patterns as password in the password area as A&4A&4A#1A$7B8+@7L*U4P-4L*5N#0. The entered random patterns are hidden, and bullets displayed in the password area.

d. The user has to remember the chosen password pattern order.

e. During confirmation of password the password patterns are shuffled. The random patterns varied are represented in Figure 8.

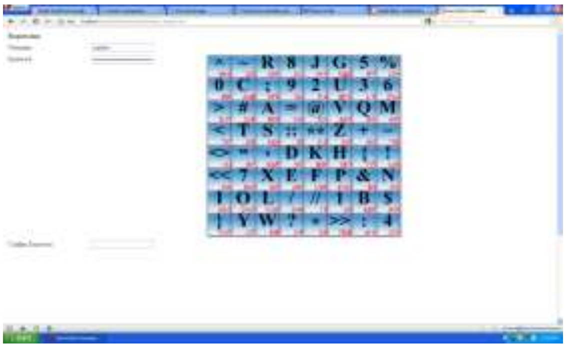


Figure 7: Shuffled Password Patterns with Varied Random Patterns for User Registration.

f. Now the user have to enter his / her password as G!7G!7?P8F9+U9!0#N0V%)K56T%B8^ as confirmation password in the password area.

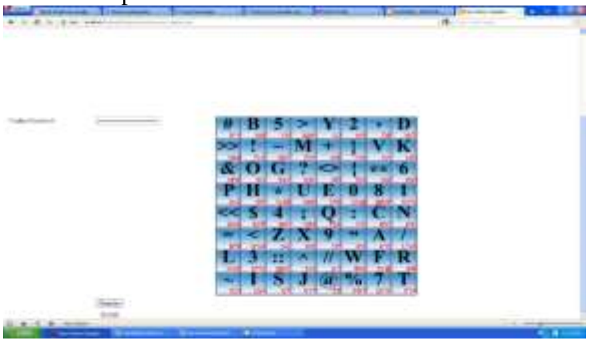


Figure 8: Confirmation of Password in User Registration using Shuffled Password Patterns with Varied Random Patterns.

g. Hidden characters are assigned to each password patterns. Using that hidden character a hash value is generated. It is stored in the database server.

User Login Steps

a. During login phase again, the password patterns are shuffled and random patterns varied as represented in Figure 9.

b. According to the password chosen by the user at the time of registration, he/she has to enter random patterns displayed as 6K*6K*G2)5)G-W20J@5T)3#W3C-@7Y at the time of login.

c. The password given at login time is taken and a hash value generated.

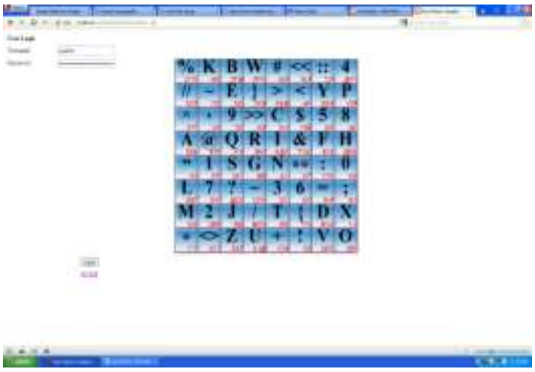


Figure 9: User Login - Shuffled Password Patterns with Varied Random Patterns.

d. Every time of login, the password patterns are shuffled and the random patterns varied.

Authentication Verification Steps

a. During registration phase – for each password pattern there will be a hidden character. Using that a generated hash value present in the database is taken for validation.

b. During login phase – Using the user password given at login time, a hash value is generated. It is compared with the value present in the database server.

c. If both values are matched, then authentication is guaranteed to the end, user otherwise denied.

During login time, the client has to give the 10-digit identification number as a first level of security to get the login page. At the time of login, client's registered hash value password is verified with the hash value password generated at the time of login. If both the passwords are equal, then authentication is granted otherwise is denied.

The client can select the password patterns on some sequences, familiar to him/her. Due to shuffling mechanism, this method reduces the guess ability of the persons who are related to the clients. During entry of password, only bullets appear in the password area which avoids the shoulder surfing attacks.

When sending random patterns in the network plane, it will be converted into a computed ascii value, so that Man-In-The-Middle attack is prohibited.

Using this mapping mechanism, the shuffling process of password patterns and index numbers are generated. The password patterns are validated only by using the hidden characters and random patterns. It reduces the time complexity of comparing with password patterns.

The password pattern positions are generated using permutation sequences. Let A = {I1, I2, I3}, be arranged in 3! ways as,

[I1] [I2] [I3], [I1] [I3] [I2], [I2] [I1] [I3], [I2] [I3] [I1], [I3] [I1] [I2], [I3] [I2] [I1]

For n password patterns n! sequences are generated and it will be used randomly for every attempt of registration or login.

Security Advantages of SAP-RP

Security of patterns or images in the existing scenario is freely hacked by malicious users in Right-click option method.

Right-click option method is a popular trick method and will not deter anyone who knows how to deal with it. In this method the user can right-click the image to save it. This works in all JavaScript – enabled browsers. If a malicious user is tries to copy any image or random patterns through right click option from the web page by online, it is allowed in the existing system by disabling the java script.

The way of disabling the java script, disables all right-clicking; it is very annoying for visitors who genuinely want to right-click in the page. Thus securing of patterns using this method makes malicious user to acquire the patterns easily by disable the java script option.

To avoid acquirement of patterns from the webpage, securities of patterns are done through cloaking method in SAP-RP system.

Cloaking Method

Cloaking method is incorporated in this SAP-RP system is to avoid acquirement of patterns from the webpage. A lesser-known nifty trick is to cloak patterns behind a transparent GIF. Place the original pattern on the page in a table or layer, and then place a transparent GIF pattern the same size over the top.

When a malicious user right-click the pattern he / she will save the transparent GIF and get only blank pattern, not the original one. This method is incorporated in this system and malicious users will get only the transparent GIF if they try to save the pattern in their folders. It is represented in Figure 11.

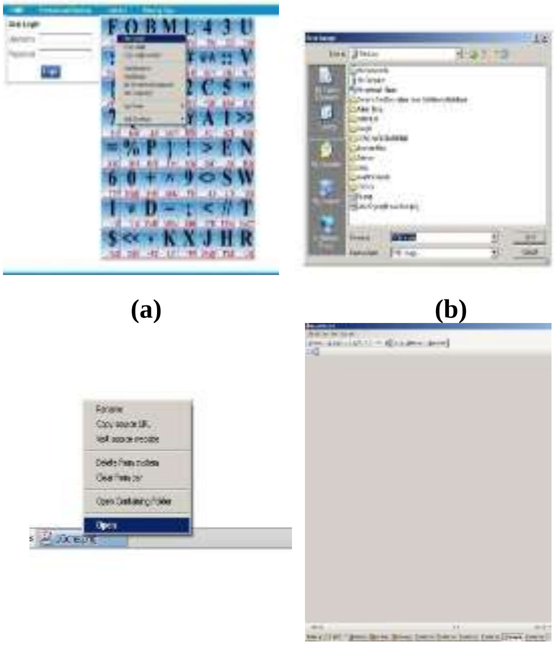


Figure 11 : Cloaking Method. (a) Malicious user try to save the pattern from the web page. (b) Save it in 1Clone(1).jpg file. (c) Try to open in the 1Clone(1).jpg file. (d) A transparent pattern is appeared.

Thus securing of random patterns is imparted using these methods.

Protocol Analysis of SAP-RP

SAP-RP system is used as text-based authentication by giving password pattern in the FUI. It registers the password from the user. It verifies the same process in registration and login. Notations used in this system are given in Table II.

Table 2. Notations of Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol

Symbol	Meaning
E	$\{S_i / 1 \le i \le n!\}$ where S_i is a sequence of n password patterns.
M_o	0^n = the all 0's string of length n.
$M_o(n)$	$0^{n-e} \parallel b_{e-1}, b_{e-2},\dots b_1, b_0$ where $b_{e-1}, \dots b_1, b_0$ is $b_r \in \{0, 1, 2, 3 \dots 9$, A-Z, Special Characters$\}$, $0 \le r \le e-1$
K	Set of sequences formed out of $M_o(n)$ Define $\phi_p : N \rightarrow \{M_o(n)\}$, N – set of Natural numbers, such that ϕ_p is a sequence on $M_o(n)\}$
K	$\{\phi_p / p = 1, 2, \dots, n!\}$
E_t	A transformation belonging to K with $t \in E$. Each E_t maps sequence in K to an n-image sequence in E.
h	A publicly-known one-way hash function from $\{0,1\}^*$ to $\{0,1\}^n$
n	A fixed positive integer which serves as a security parameter.

Algorithm (i) : Key generation (User Registration)

Description: Each entity A, selects a password pattern key scheme E, generates n random secret strings, and creates a set of validation parameters.

Each entity A should do the following:

1. *Select a text password key scheme using password patterns E.*
2. *Generate n random secret strings $k_1, k_2, \dots k_n \in K$, each of length n.*
3. *Compute $y_i \leftarrow E_{Ki}(M_o(n))$, $1 \le i \le n$.*
4. *A's secret key:$(y_1, y_2, \dots y_n)$.*

Algorithm (ii) : Password pattern key generation and verification (User Login)

Description: Entity A signs a message m of arbitrary length, message verification is interactive with A.

1. Password pattern key generation. Entity A should do the following:

a. *Compute: $x \leftarrow h(m)$.*

b. *Compute: $s_i \leftarrow E_{Ki}(x)$, $1 \le i \le n$.*

c. *A's key for m: $m \leftarrow (s_1, s_2, \dots s_n)$.*

2. Password pattern key verification. To verify A's key $(s_1, s_2, \dots s_n)$ on m, B does the following:

a. *A's authentic secret key is given as input: $(y_1, y_2, \dots y_n)$.*

b. *Compute: $x \leftarrow h(m)$.*

c. *Select: n distinct random secret numbers r_j, $1 \le r_j \le n$, for $1 \le j \le n$.*

d. *Request from: A the keys k_{rj}, $1 \le j \le n$.*

e. *Verify authenticity of the received keys by computing:*

f. *$z_j \leftarrow E_{krj} (M_o(r_j))$ and checking that $z_j \leftarrow y_{rj}$, for each $1 \le j \le n$.*

g. *Verify: $s_{rj} \leftarrow E_{krj} (x)$, $1 \le j \le n$.*

Time Complexity of SAP-RP

In SAP-RP the time complexity is said to be described asymptotically, i.e., as the input size goes to infinity.

By determining the number of steps per execution (s/e) of the statement and the total number of times (i.e., frequency) each statement is executed for SAP-RP Algorithm (i) is represented in Table III and Algorithm (ii) is shown in Table IV.

Table 3. Performance Analysis – Time Complexity for SAP-RP (Algorithm (i))

Step No.	Statement	s/e	Frequency	Total Steps
1	*Select a text password key scheme using password pattern E*	1	-	1
2	*Generate n random secret strings $k_1, k_2, \dots k_n \in K$, each of length n*	1	n	n
3	*Compute $y_i = E_{ki}(M_o(n))$, $1 \le i \le n$*	1	n	n
4	*A's secret key is $(y_1, y_2, \dots y_n)$*	1	1	1
	Total			O(n)

Table 4. Performance Analysis – Time Complexity for SAP-RP (Algorithm (ii))

Step No.	Statement	s/e	Frequency	Total Steps
1. a	*Compute h(m)*	1	-	1
1. b	*Compute $s_i = E_{ki}(h(m))$, $1 \le i \le n$*	1	n	n
1. c	*A's key for m is $(s_1, s_2, \dots s_n)$*	1	1	1
2.a	*Obtain A's authentic secret key $(y_1, y_2, \dots y_n)$*	1	1	1
2.b	*Compute h(m)*	1	-	1
2.c	*Select n distinct random secret numbers r_j, $1 \le r_j \le n$, for $1 \le j \le n$*	1	n	n
2.d	*Request from A the keys k_{rj}, $1 \le j \le n$*	1	n	n
2.e	*Verify the authenticity of the received keys by computing $z_j = E_{krj} (M_o(r_j))$ and checking that $z_j = y_{rj}$, for each $1 \le j \le n$*	1	n	n
2.f	*Verify that $s_{rj} = E_{krj} (h(m))$, $1 \le j \le n$*	1	n	n
	Total			O(n)

Since an algorithm may take a different amount of time even on inputs of the same size, the most commonly used measure of time complexity, is the maximum amount of time taken on any input of size n. Time complexities are classified by the nature of the function T(n). Hence this SAP-RP is dealt with linear time algorithm due to the instance T(n) = O(n).

Security Potency of Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol:

In general, several attacks are possible on an authentication system. For any authentication system, the hacker can attack at least at three places: they are server, client and the communication link.

The attack on server includes Brute force attack, Dictionary attack and compromising the server as a whole.

- At the client, the possible attacks are key logging and shoulder surfing.
- Finally on the communication link, the possible attack is Man-In-The-Middle attack. It can be done using packet sniffers.[5].

In terms of the data being passed from the user to the server the data stored in the secured server is comparable with the classical Password-based authentication system. In both cases, user sends the username and a password. This will be compared with the registry in the database. But because of the dynamic nature of password selection system, SAP-RP is more secure than ordinary password-based scheme to attacks such as Brute force, Dictionary attack, Keylogger, Shoulder surfing and Server database compromise attack. The best known solution for such attacks is to use cryptography protocols at the server or on the communication link.

In this we analyse the impact of the four attacks mentioned here on SAP-RP. On analysing Brute force attack - I in SAP-RP, if the hacker wants to guess the password, the probability of success will be $1/(64^4)$ = 5.96046E-08 (Since there are unlimited password patterns, 64 password patterns are taken as sample). If the guess is wrong, probability of success will remain the same for the next guess. It is because; the password will change with every attempt.

Hence, the probability of success for every attempt = $1/64^n$

The other way of doing Brute force attack - II is to try all combinations of positions. For example, if we consider a 8x8 Random Pattern setup, there will be 64^n (if selection of patterns includes reuse of patterns) or ${}^{64}P_n$ (without reuse of patterns) different patterns of length n.

Eqn. I.a
Number of possible
Password patterns =
Eqn. I.b

$$\left\{ \begin{array}{c} (N^2)^{\,n} \\ \hline \\ \dfrac{(N^2)!}{(N^2-n)!} \end{array} \right.$$

Number of possible password patterns for the size of N x N matrix with re-use of password patterns as passwords $(N^2)^{\,n}$ is illustrated in Equation I.a and without re-use of password patterns as passwords $(N^2)!/(N^2-n)!$. It is stated in Equation I.b. The password pattern access varies in two forms as, i) N^2 password patterns with increase in password length and ii) Password length is n with increase in password patterns.

i) N^2 password patterns with increase in password length:

Let us consider N^2 password patterns and let n be a length of the password.

The total number of passwords available are given by } $C1 = (N^2)^n = (N^{2n})$

Now, increase the password length by one, then } $C2 = (N^2)^{n+1} = (N^{2n}.N^2)$

$C2 – C1 = (N^{2n}.N^2) – (N^{2n})$

$C3 = N^{2n} (N^2 – 1)$ Eqn. II

Thus, we provide N^{2n} passwords which are exponential in nature.

ii)Password length is n with increase in password patterns:

Let us consider N^2 password patterns and let n be a length of the password.

The total number of passwords available are given by } $C1 = (N^2)^n = (N^{2n})$

Now, increase the number of password patterns by the password patterns order of one, then } $C2 = (N + 1)^2$

$C3 = C2 – C1 = (N + 1)^{2n} – (N^{2n})$ Eqn. III

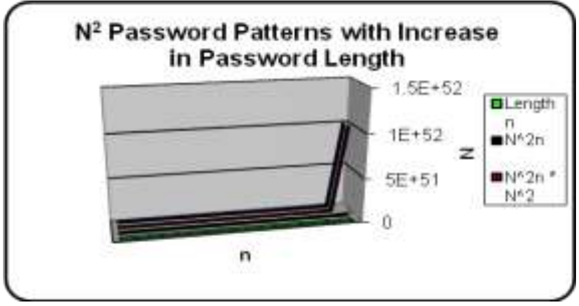
Thus, once again, the total number of passwords is exponential in nature.

Hence, the Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol provides very wide range of passwords. Evaluation of N^2 password patterns with increase in password length is represented in Table V and password length is n with increase in password patterns as represented in Table VI.

Table 5. Evaluation of N^2 password patterns with Increase in Password Length

Password Patterns – N^2 and Length of the Password – n N = 10 and N^2 = 100					
	N^{2n}	$N^{2n}.\ N^2$			$N^2(n+1)$
n	C1	C2	C2 – C1	N^{2n} * (N^2- 1)	C3
8	1E+16	1E+18	9.9E+17	9.9E+17	1E+18
9	1E+18	1E+20	9.9E+19	9.9E+19	1E+20
10	1E+20	1E+22	9.9E+21	9.9E+21	1E+22
11	1E+22	1E+24	9.9E+23	9.9E+23	1E+24
12	1E+24	1E+26	9.9E+25	9.9E+25	1E+26
13	1E+26	1E+28	9.9E+27	9.9E+27	1E+28
14	1E+28	1E+30	9.9E+29	9.9E+29	1E+30
15	1E+30	1E+32	9.9E+31	9.9E+31	1E+32
16	1E+32	1E+34	9.9E+33	9.9E+33	1E+34
17	1E+34	1E+36	9.9E+35	9.9E+35	1E+36
18	1E+36	1E+38	9.9E+37	9.9E+37	1E+38
19	1E+38	1E+40	9.9E+39	9.9E+39	1E+40
20	1E+40	1E+42	9.9E+41	9.9E+41	1E+42
21	1E+42	1E+44	9.9E+43	9.9E+43	1E+44
22	1E+44	1E+46	9.9E+45	9.9E+45	1E+46
23	1E+46	1E+48	9.9E+47	9.9E+47	1E+48
24	1E+48	1E+50	9.9E+49	9.9E+49	1E+50
25	1E+50	1E+52	9.9E+51	9.9E+51	1E+52

In SAP-RP, the user passwords are exponential in nature as represented through Graph 1 by increasing the length of the password and through Graph 2 by increasing the length of the password patterns. N represents the size of the Password Patterns and n represents the length of the password.

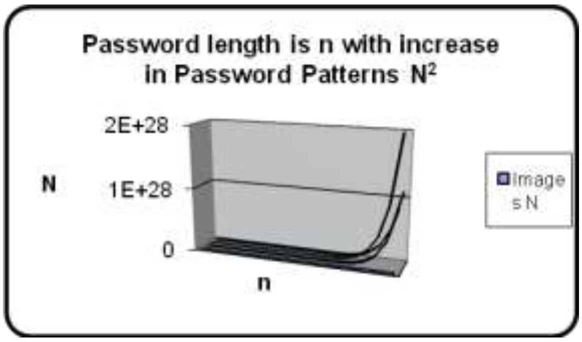


Graph 1: Password Patterns created are exponential by increasing the Password Length.

Analysing Dictionary attack in SAP-RP, commonly used password patterns with client guess sequences can be possible (if random patterns are static). However, here the password pattern changes randomly on every presentation or session; it approaches the behaviour of one-time pad.

Table 6. Password Length is n with Increase in N^2 Password Patterns

Password Patterns – N^2 , Length of the Password – n and n = 10					
n	N^2	N^{2n} C1	$(N+1)^{2n}$ C2	C2 – C1	$(N+1)^{2n\text{-}}N^{2n}$
8	36	3.656E+15	7.979E+16	7.6136E+16	7.61361E+16
9	49	7.979E+16	1.152E+18	1.0731E+18	1.07313E+18
10	64	1.152E+18	1.215E+19	1.1004E+19	1.10047E+19
11	81	1.215E+19	1E+20	8.7842E+19	8.78423E+19
12	100	1E+20	6.727E+20	5.7275E+20	5.7275E+20
13	121	6.727E+20	3.833E+21	3.1610E+21	3.16101E+21
14	144	3.833E+21	1.900E+22	1.5171E+22	1.51712E+22
15	169	1.900E+22	8.366E+22	6.4663E+22	6.46633E+22
16	196	8.366E+22	3.325E+23	2.4885E+23	2.48857E+23
17	225	3.325E+23	1.208E+24	8.764E+23	8.764E+23
18	256	1.208E+24	4.064E+24	2.8553E+24	2.85531E+24
19	289	4.064E+24	1.274E+25	8.684E+24	8.684E+24
20	324	1.274E+25	3.759E+25	2.4841E+25	2.48417E+25
21	361	3.759E+25	1.048E+26	6.7267E+25	6.72676E+25
22	400	1.048E+26	2.782E+26	1.7336E+26	1.73361E+26
23	441	2.782E+26	7.054E+26	4.2721E+26	4.27211E+26
24	484	7.054E+26	1.716E+27	1.0107E+27	1.01073E+27
25	529	1.716E+27	4.019E+27	2.3038E+27	2.30383E+27



Graph 2: Password Patterns generated are exponential by increasing the Password Patterns.

SAP-RP, being a dynamic password system, is not vulnerable to keyloggers. Even if the hacker gets the password of the client of a SAP-RP system, this password cannot be reused by the hacker to login to the system, because of the dynamic nature of the Password Pattern system.

Shoulder surfing can be done easily on the password system, just by seeing the keys that the user is typing. But to decode the password in SAP-RP, the hacker has to see both the key sequence and Password patterns and do a mapping before user submits the page. So shoulder surfing is of little or no use in SAP-RP as compared to a password-based system.

In the case of Man-in-the-middle attack, the attackers are not able to get original messages in SAP-RP, because the password patterns and random patterns changed dynamically on every presentation or session. Comparing these attacks with existing systems and it is represented in Graph 3.

ES1 – Pass Pattern System
ES2 – Improving Text Password through Persuasion
ES3 – Enhanced Authentication Mechanism using Multilevel Security Model
ES4 – Graphical Password: Usable Graphical Password Prototype
ES5 – Authentication Using Graphical Passwords: Basic Results
ES6 – A User Study Using Images for Authentication
ES7 – Passface: Real User Corporation
ES8 – A Password Scheme Strongly Resistant to Spyware
SAP-RP – Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol

The images used for password selection can be of any kind. Depending on the application it can be varied. For sample discussion nature patterns were used. For implementation characters, numbers and special characters were used as patterns. Two digits and three digits random patterns were used in implementation. In compact display applications two digit random patterns were preferred and in large display applications three digit random patterns were preferred to mystify hackers.

Security Questions Authentication

In second level the client gets authenticated using security questions. A 10-digit number is issued to the client at the time of registration. The client has to answer three security questions; the results are encrypted with a 10-digit number. A resultant factor is passed over the network plane for validation to the server.

Encryption Process

- Three security questions queried (s1, s2, and s3).
- Ascii value evaluated for two security questions.(a1 and a2)
- Bitwise operation is performed,

– sum1=(a1 & a2) | s3

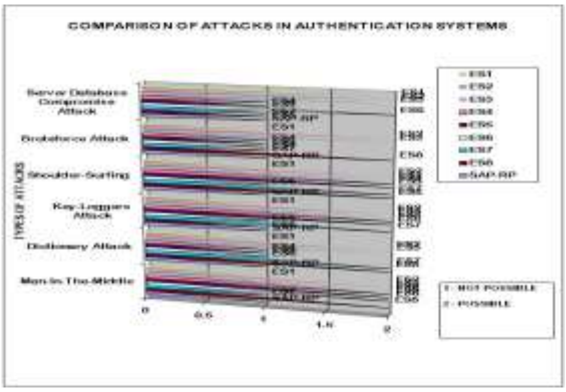
- resultant factor (sum2) = sum1 ⊕ id
- Ascii value of resultant factor (sum2) send to verifier.

Verification Process

• During Client registration; a shared 10-digit key (id) and resultant factor (sum2) issued to server.

• Authentication process: achieved result (sum3) of client ⊕ resultant factor (sum2).

• Authentication granted – a shared 10-digit key (id) generated. If not, then authentication is denied.



Graph 3: Comparison of Attacks in Authentication Systems using Existing and Proposed System.

The server decrypts the resultant factor and gets the registration number of the client. After passing Random Patterns level and Security questions authentication level, the client gets authenticated.

Analysis and Implementation

This system overcomes all the drawbacks of existing system. It provides confidentiality and authentication using random patterns with flexible user interface. This system is implemented both in single client and multiple clients with server.

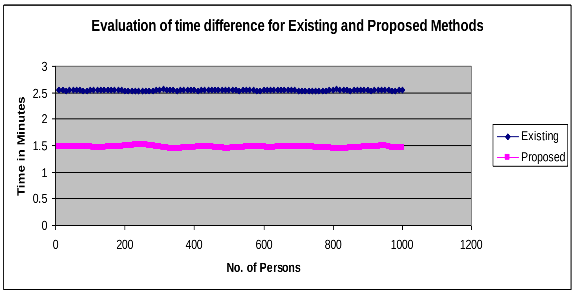
In this system, three levels of authentication are integrated with single server to authenticate clients. No tiresome methods are used in this system. It does not irritate the client. No leakage of information is possible in this system which avoids the Man-In-The-Middle attack.

Hashing of user password generated through HMAC(MD5). The HMAC process mixes a secret key with the message data, hashes the result with the hash function, mixes that hash value with the secret key again, and then applies the hash function a second time. MD5 is a cryptographic hash algorithm developed at RSA Laboratories. HMAC(MD5) accepts keys of any size, and produces a hash sequence 128 bits in length.

Sample representation of HMAC_MD5() is,
HMAC_MD5(“10-digit identification number”, “A&4A&4A#1A$7B8+@7L*U4P-4L*5N#0”) = 0x 80070713463e7749b90c24911e27.

Cryptographic hash function MD5 is used in the calculation of HMAC; resulting MAC algorithm is termed HMAC_MD5 accordingly. The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output length in bits, and on the size and quality of the cryptographic key.[17]. Thus reversing the hash value of MD5 will not lead to acquire the password pattern given by the client.

Both existing and proposed systems were implemented and the time difference is evaluated. It is represented in Graph 4.



Graph 4: Evaluation of time difference for existing and proposed methods.

The total pattern position sequences generated for 9 patterns are 362880 and increase in number of patterns persuades to increase in pattern sequences in factorials.

Conclusion

A novel system is developed and implemented using the Random Pattern based Flexible User Interface for an Effective Secured Authentication Protocol for authentication. This system is more simple and easy for all kind of end-users to remember the password patterns, even when the user has more number of passwords. Due to the shuffling and variation mechanisms involved in the system it makes the malicious users unable to hack the information from the network plane. In order to improve the confidence in security system, detecting intrusions in those systems plays a vital role, as the security system design is not always perfect. This system overcomes the problem encountered in existing systems and ensures the confidentiality and authentication when a transaction is held.

Acknowledgement

This paper is part of SSE Project funded through a National Technical Research Organization, New Delhi is gratefully acknowledged.

References

[1] Abdulameer Hussain, “Enhanced Authentication Mechanism Using Multilevel Security Model”, Faculty of Science and Information Technology, Zarka Private University, Jordan, International Arab Journal of e-Technology, Vol. 1, No.2, June 2009.

[2] Alain Forget, Sonia Chiasson, P.C. van Oorschot, Robert Biddle, “Improving text passwords through persuasion”, School of Computer Science, Human Oriented Technology Lab, Carleton University, Ottawa, Canada, {aforget, chiasson, paulv}@scs.carleton.ca, robert_biddle@carleton.ca. Symposium on Usable Privacy and Security (SOUPS) 2008, July 23-25, 2008, Pittsburgh,, PA, USA.

[3] Atul Kahate, Cryptography and network security, The Tata Mc-Graw Hill publications.

[4] Bruice Schneier, Applied Cryptography, Protocols, Alogrithms and Source Code in C, Second Edition, Published by JOHN WILEY and SONS, Reprint 2007.

[5] T. Rakesh Kumar and S.V. Raghavan, PassPattern System (PPS): A Pattern-Based User Authentication Scheme, NSL, Department of Computer Science and Engineering, IITM, Chennai, India.

[6] R. N. Shepard, C.:Recognition memory for words, sentences and pictures, Journal of verbal Learning and verbal Behavior, vol. 6, pp. 153—163 (1967).

[7] William Stallings, Cryptography and network Security principles and practices, 2006 by pearson education, Inc.

[8] http://www.computerhope.com/jargon/n/bruteforc.htm - Definition referred.

[9] http://www.tech-faq.com/brute-force-attack.html - Definition referred.

[10] http://searchsecurity.techtarget.com/definition/dictionary-attack - Definition referred.

[11] http://www.anotherwindowsblog.com/2010/09/attack-of-the-keyloggers.html - Definition referred.

[12] Ali Mohamed Eljetlawi and Norafida Bt. Ithnin, “Graphical Password: Usable Graphical Password Prototype”, Journal of International Commercial Law and Technology Volume 4, Issue 4 (2009).

[13] Susan Wiedenbeck, Jean-Camille Birget and Alex Brodskiy, Nasir Memon, “Authentication Using Graphical Passwords: Basic Results”, Proceedings of the 11th Human-Computer Interaction International (2005), Key: citeulike: 8209689.

[14] Rachna Dhamija and Adrian Perrig, “Deja Vu: A User Study Using Images for Authentication”, SSYM’00 Proceedings of the 9th Conference on USENIX Security, Symposium, Volume 9, USENIX Association Berkeley, CA, USA c 2000, ACM Digital Library.

[15] Real User, www.realuser.com last accessed in June 2005.

[16] D. Hong, S. Man, B. Hawes, and M. Mathews, “A password scheme strongly resistant to spyware”, in Proceedings of International Conference on security and management. Las Vergas, NV, 2004.

[17] en.wikipedia.org/wiki/HMAC – Definition referred.

[18] R. Sujatha and G. Arumugam, “Secured Authentication Protocol System using Images”, International Journal of Computer Science and Information Security, Vol. 8, No. 8, 2010, pp 110-116.

Authors

Dr. G. Arumugam received his post-graduate degree in Applied mathematics from PSG College of Technology, Coimbatore and Ph.D degree from University of Piere and Marie curie, Paris, France in 1987. He is now the Chair Person, School of Physics and Prof. & Head of Computer Science department at Madurai Kamaraj University, Madurai, TamilNadu, South India. He is an active researcher in databases, data mining, Bioinformatics and mobile computing and has published more than 50 papers in journals and conference proceedings. Email: gurusamyarumgam@gmail.com

Ms. R. Sujatha received her post-graduate degree in Computer Science from Madurai Kamaraj University, Madurai and M.Phil degree from Alagappa University, Karaikudi. She is now working as Research Associate in a SSE Project in Department of Computer Science at Madurai Kamaraj University, Madurai, TamilNadu, India. She is a researcher in the area of Authentication and Network Security and has published papers in journals and presented in workshops. Email: sujamurali72@gmail.com.