ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

Shoulder Surfing Resistance Graphical Password Autentication

[1] V.D.Janani, [2] V.Anwar, [3] M .Ravi Sankar
[1] Asst, Prof Dept Of Cse, Biher, Chennai, Tamil Nadu, India.
[2] [3] Student, Dept Of Cse, Biher, Chennai, Tamil Nadu, India.

Abstract: ***With the development of cloud computing, Data security becomes more and more important in cloud computing. This paper analyses the basic problem of cloud computing data security. Since Cloud Computing share distributed resources via network in the open environment thus it makes security problems. In this method, we use Grid selection and Pass Point algorithms. This algorithm process on the image to get the pixel points and the points were used as a secret key for the file uploaded by the user. These techniques were different from the usual technique like Text Based Password or randomized key authorization has some difficulties that users have tended to write passwords down manually.***

Keywords: Graphical password, password, co-ordinates, CaRP, Captcha, Dictionary attack, password guessing attack, security primitive.

I. INTRODUCTION

Using hard AI (Artificial Intelligence) problems for security, initially proposed in, is an exciting new paradigm. Under this paradigm, the most notable primitive invented is Captcha, which distinguishes human users from computers by presenting a challenge, i.e., a puzzle, beyond the capability of computers but easy for humans. Captcha is now a standard Internet security technique to protect online email and other services from being abused by bots. However, this new paradigm has achieved just a limited success as compared with the cryptographic primitives based on hard math problems and their wide applications. Is it possible to create any new security primitive based on hard AI problems this is a challenging and interesting open problem.

In this paper, we introduce a new security primitive based on hard AI problems, namely, a novel family of graphical password systems integrating Captcha technology, which we call CaRP (Captcha as gRaphical Passwords). CaRP is click-based graphical passwords, where a sequence of clicks on an image is used to derive a password. Unlike other click-based graphical passwords, images used in CaRP are Captcha challenges, and a new CaRP image is generated for every login attempt. The notion of CaRP is simple but generic. CaRP can have multiple instantiations. In theory, any Captcha scheme relying on multiple-object classification can be converted to a CaRP scheme. We present exemplary CaRPs built on both text Captcha and image-recognition Captcha. One of them is a text CaRP wherein a password is a sequence of characters like a text password, but entered by clicking the right character sequence on CaRP images. CaRP offers protection against online dictionary attacks on passwords, which have been for long time a major security threat for various online services. This threat is widespread and considered as a top cyber security risk. Defense against online dictionary attacks is a more subtle problem than it might appear. Intuitive countermeasures such as throttling logon attempts do not work well for two reasons:

1) It causes denial-of-service attacks (which were exploited to lock highest bidders out in final minutes of eBay auctions and incurs expensive helpdesk costs for account reactivation.

2) It is vulnerable to global password attacks whereby adversaries intend to break into any account rather than a specific one, and thus try each password candidate on multiple accounts and ensure that the number of trials on each account is below the threshold to avoid triggering account lockout.

II. EXPERIMENTAL DETAILS

2.1 SCREEN SHOTS REGISTER FORM

REGISTER FORM

LOGIN FORM

UPLOAD FILES

DOWNLOAD FILES

RESPONSE PAGE

III. ALG0RITHM SPECIFICATION

3.1 Grid Selection Algorithm

Grid selection algorithm is also a pure recall based authentication technique. It overcomes the disadvantages of DAS system i.e. with respect to password space and stroke count. The user is required to select a small region from a large rectangular grid. This region gets zoom in on selection, and then he is required to draw the password pattern.

Advantages

Larger password space as compared to DAS Algorithm

pseudo code

```
For a grid of size R × C
Track best scores: bestR, bestC
Keep caches rCache, cCache
Init:
    rCache[0] ← 1
    cCache[0] ← 1
for R_H from 1 to R
    bestR ← 0
    for R_L from 1 to R_H
        for C_H from 1 to C
            bestC ← 0
            for C_L from 1 to C_H
                s ← max{score(0, R_L, R_H, C_L, C_H), score(1, R_L, R_H, C_L, C_H)}
                bestC ← max{bestC, cCache[C_L − 1] · s}
            end for
            end for
        end for
        bestR ← max{bestR, rCache[R_L − 1] · bestC}
    end for
    rCache[R_H] ← bestR
end for
return bestR
```

3.2 PASSPOINT ALGORITHAM

Pass Point Algorithm is a cued recall based technique. The system allows any natural image to be used which should be rich enough to have many possible click points. The role of the image is just to provide a hint to the user which helps in remembering the click points. During login, the click points should be selected in the same order as in

Registration phase inside some adjustable tolerable distance. The tolerable distance can be set by the system say within 0.25 cm from the actual click point.

Advantages

User had to click on the predefined image at predefined region. Pass Point algorithm overcomes this by selecting any natural
image and having click points as possible which make the system more secure.

Pseudo code

```
Require: that G = (V, L) is a undirected graph.

function FindConnectedSets(G, n_max)
    S_all ← ∅
    for i in V do
        S ← {i}
        V_- ← {j ∈ V | j ≤ i}
        V_+ ← {j ∈ N(i) | j > i}
        SubFind(G, n_max, S_all, S, V_-, V_+)
    return S_all

function SubFind(G, n_max, S_all, S, V_-, V_+)
    S_all ← S_all ∪ {S}
    if |S| = n_max then return
    for i in V_+ do
        S' ← S ∪ {i}
        V'_- ← V_- ∪ {j ∈ V_+ | j ≤ i}
        V'_+ ← {j ∈ V_+ | j > i} ∪ {j ∈ N(i) | j ∉ V'_-}
        SubFind(G, n_max, S_all, S', V'_-, V'_+)
```

IV. IMPLEMANTATION

4.1 ARCITECTURE DIAGRAM

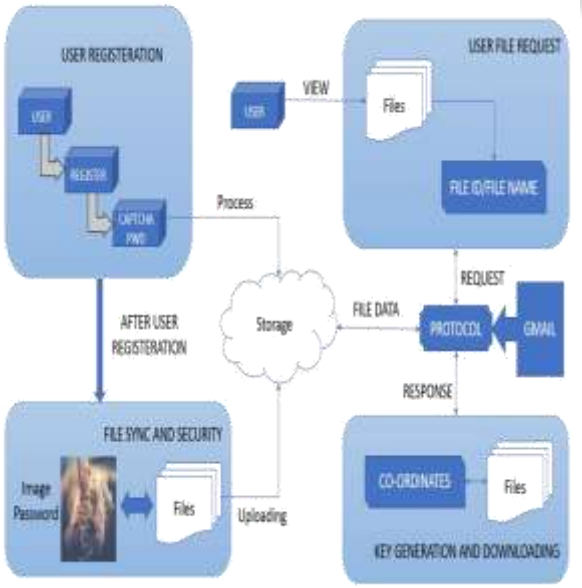

Fig 4.1

4.2 LIST OF MODULES

[1] User registration

[2] File sync and security

[3] User file request

[4] Key generation and downloading

4.3 MODULES DISCRIPTION

1. USER REGISTERATION

In this module, Users are having authentication and security to access the cloud. For security, captcha technique is implemented to access the detail which is presented in the Image system. Each time Users have to enter, registered captcha text and password for accessing the account.

2. FILE SYNC AND SECURITY

The user can start up the server after system is opened. Then the user can upload the file to the storage with the key to access it. The key process is done with Grid selection and Pass Point algorithm. By clicking particular point at the given image, the position of the image pixel is taken as X & Y Co-ordinates as key. These co-ordinates were assigns as X1, Y1 and by clicking on different position 2nd coordinates were assigned as X2, Y2.In this a password guess tested in an unsuccessful trial is determined than traditional approaches.

3. USER FILES REQUEST

The request process is done through protocol and key is send to an authorized user through mail. By this process key is shared and the file is view/downloaded by the other user with the key given by the data owner.

KEY GENERATION AND DOWNLOADING

The number of undetermined password guesses decreases with more trials, leading to a better chance of finding the password. To counter guessing attacks, traditional
approaches in designing graphical passwords aim at increasing the effective password space to make passwords harder to guess and thus require more trials, the password can always be found by a brute force attack. In this paper, we distinguish Key generated through algorithm than the traditional approaches, and key request process is done through the protocol with the other user to access the files.

V. DATA FLOW DIAGRAM

TABLE I. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

TABLE II. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

TABLE III. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

TABLE IV. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

VI. UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Supporthigherleveldevelopment
concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices

VII. USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Fig : 7.1

7.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

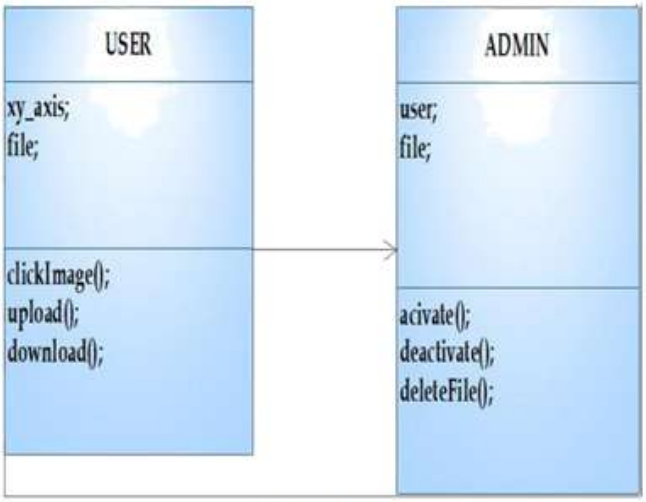

Fig : 7.2

7.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

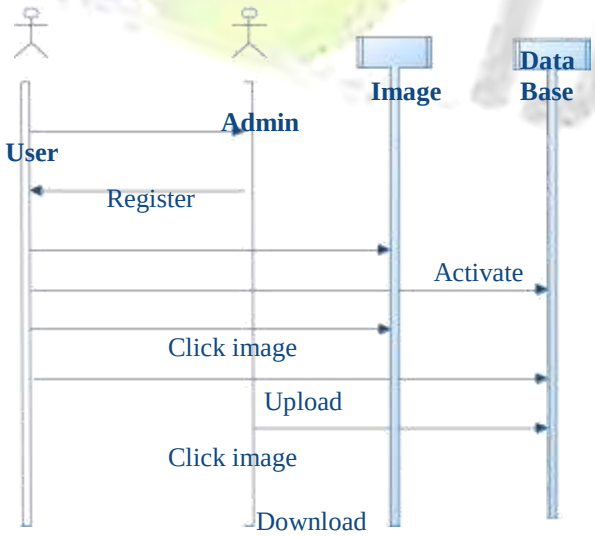

Fig : 7.3

VIII. SYSTEM TESTING

TESTING PROCESS

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

and/or a finished product it is the a process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

8.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application
.it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

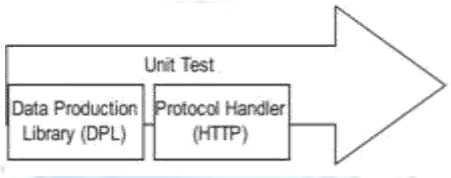

Fig: 8.1 Unit Testing

8.2 Integration Testing

Integration tests are designed to test
integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3 Functional Testing

Function altests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation and user manuals.

Functional testing is centered on the following items:

Valid Input is used to identified classes of valid input must be accepted.

Invalid Input is used to identified classes of invalid input must be rejected.

Functions is used to identified functions must be exercised.

Output is used to identify classes of application outputs.

Systems/Procedures is used to interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive
Processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 3, Issue 4, April 2018

8.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level

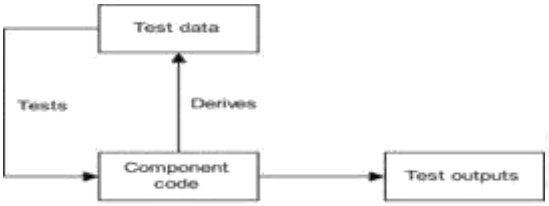

Fig :8.5 White box Testing **8.6 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without

Considering how the software works.

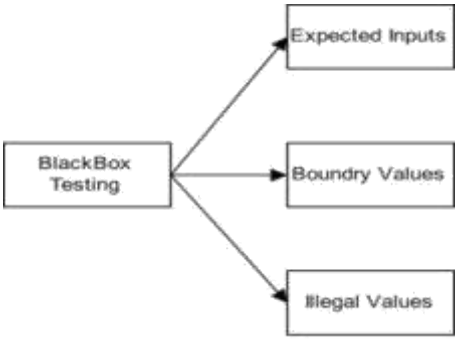

Fig : 8.6 Black box Testing

IX. FUTURE ENHANCEMENT

The past decade has seen a growing interest in using graphical passwords as an alternative to the traditional text-based passwords .Although the main argument for graphical passwords is that people are better at memorizing graphical passwords than text-based passwords, the existing user studies are very limited and there is not yet convincing

evidence to support this argument. We have proposed CaRP, a new security primitive relying on unsolved hard AI problems. CaRPis both a Captcha and a graphical password scheme. The notion of CaRP introduces a new family of graphical passwords, which adopts a new approach to counter online guessing attacks.

A new CaRP image, which is also a Captcha challenge, is used for every login attempt to make trials of an online guessing attack computationally independent of each other. A password of CaRP can be found only probabilistically by automatic online guessing attacks including brute force attacks, a desired security property that other graphical password schemes lack. Hotspots in CaRP images can no longer be exploited
o mount automatic online guessing attacks, an inherent vulnerability in many graphical password systems. CaRP forces adversaries to resort to signicantly less efficient and much more costly human based attacks.

The results of our experiments show that the future research should concentrate on improving the login time and memorability. When a user inputs the corresponding substrings which belong to different CAPTCHAs, the time gap is longer than the time between two characters in one substring. So a method for narrowing the time gap in the entering process and reduction of the impact of users choice trend on security, provide other areas for future research. The CbPA-protocols described require a user to solve a Captcha challenge in addition to inputting a password under certain conditions. For example, the scheme described applies a Captcha challenge when the number of failed login attempts has reached a threshold for an account. A small threshold is applied for failed login attempts from unknown machines but a large threshold is applied for failed attempts from known machines on which a successful login occurred within a given time frame.

X. CONCLUSION

We have proposed CaRP, a new security primitive relying on unsolved hard AI problems. CaRP is both a Captcha and a graphical password scheme. The notion of CaRP introduces a new family of graphical passwords, which adopts a new approach to counter online guessing attacks: a new CaRP image, which is also a Captcha challenge, is used for every login attempt to make trials of an online guessing attack computationally independent of each other. A password of CaRP can be found only *probabilistically* by automatic online guessing attacks including brute-force attacks, a desired security property that other graphical password schemes lack. Hotspots in CaRP images can no longer be exploited to mount automatic online guessing attacks, an inherent vulnerability in many graphical password systems. CaRP forces adversaries to resort to significantly less efficient and much more costly human-based attacks. In addition to offering protection from online guessing attacks, CaRP is also resistant to Captcha relay attacks, and, if combined with dual-view technologies, shoulder-surfing attacks. CaRP can also help reduce spam emails sent from a Web email service.

Our usability study of two CaRP schemes we have implemented is encouraging. For example, more participants considered Animal Grid and Click Text easier to use than PassPoints and a combination of text password and Captcha. Both Animal Grid and Click Text had better password memo ability than the conventional text passwords. On the other hand, the usability of CaRP can be further improved by using images of different levels of difficulty based on the login history of the user and the machine used to log in. The optimal tradeoff between security and usability remains an open question for CaRP, and further studies are needed to refine CaRP for actual deployments.

Like Captcha, CaRP utilizes unsolved AI problems. However, a password is much more valuable to attackers than a free email account that Captcha is typically used to protect. Therefore there are more incentives for attackers to hack CaRP than Captcha. That is, more efforts will be attracted to the following win-win game by CaRP than ordinary Captcha: If attackers succeed, they contribute to improving AI by providing solutions to open problems such as segmenting 2D texts. Otherwise, our system stays secure, contributing to practical security. As a framework, CaRP does not rely on any specific Captcha scheme. When one Captcha scheme is broken, a new and more secure one may appear and be converted to a CaRP scheme. Overall, our work is one step forward in the paradigm of using hard AI problems for security. Of reasonable security and usability and practical applications, CaRP has good potential for refinements, which call for useful future work. More importantly, we expect CaRP to inspire new inventions of such AI based security primitives.

REFERENCES

[1]. R.Biddle, S.Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," ACM Comput. Surveys, vol. 44, no. 4, 2012.
[2].I.Jermyn, A.Mayer, F.Monrose, M.Reiter, and A.Rubin, "The design and analysis of graphical passwords," in Proc. 8th USENIX Security Symp., 1999, pp. 1–15.
[3].H.Tao and C.Adams, "Pass-Go: A proposal to improve the usability of graphical passwords," Int. J. Netw. Security, vol. 7, no. 2, pp. 273–292, 2008. [4].S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical password system," Int. J. HCI, vol. 63, pp. 102–127, Jul. 2005.
[5].P.C.van Oorschot and J. Thorpe, "On predictive models and userdrawn graphical passwords," *ACM Trans. Inf. Syst. Security*, vol. 10, no. 4, pp. 1–33, 2008.
[6]. K. Golofit, "Click passwords under investigation," in *Proc. ESORICS*, 2007, pp. 343–358.
[7]. A. E. Dirik, N. Memon, and J.-C. Birget, "Modeling user choice in the passpoints graphical password scheme," in Proc. Symp. Usable Privacy Security, 2007, pp. 20–28.
[8]. J. Thorpe and P. C. van Oorschot, "Human-seeded attacks and exploiting hot
[9]. spots ingraphical passwords," in Proc. USENIX Security, 2007, pp. 103–118.
[9]. P. C. van Oorschot, A.Salehi-Abari, and J. Thorpe, "Purely automated attacks on passpoints-style graphical passwords," IEEE Trans. Inf. Forensics Security, vol. 5, no. 3, pp. 393–405, Sep. 2010.
[10]. P. C. van Oorschot and J. Thorpe, "Exploiting predictability in clickbased graphical passwords," J. Comput. Security, vol. 19, no. 4, pp. 669–702, 2011.